Zachodniopomorski Uniwersytet Technologiczny
w Szczecinie
Wydział Informatyki

mgr inż. Gerard Wawrzyniak

Wykorzystanie łańcuchów bloków zintegrowanych z formularzami elektronicznymi w realizacji bezpiecznych transakcji

Rozprawa doktorska

Promotor:

dr hab. inż. Imed El Fray prof. ZUT

Szczecin, 2023

Spis treści

1. Wstęp **6**

2. Historia – pierwsze transakcje i dokumenty **8**
2.1. Pierwsze transakcje . . . 8
2.2. Pismo, dokument, transakcja . . . 9
2.3. Rozwój dokumentu i umów-formularzy . . . 14
2.4. Wiarygodność pism – pieczęcie i podpisy . . . 15
2.5. Podsumowanie . . . 18

3. Transakcje **19**
3.1. Definicje transakcji . . . 19
3.2. Informacja w transakcji . . . 20
3.3. Dojrzałość usług elektronicznych . . . 21
3.4. Przetwarzanie i transfer informacji w transakcji . . . 22
3.5. Dokumenty elektroniczne . . . 25
3.6. Formularz elektroniczny . . . 27
3.7. Podsumowanie . . . 29

4. Interoperacyjność **30**
4.1. Obszary interoperacyjności . . . 30
4.2. Podsumowanie – Interoperacyjność . . . 32

5. Bezpieczeństwo **33**
5.1. Podpis elektroniczny . . . 33
5.2. Podpis XML . . . 34
5.2.1. Struktura podpisu XML . . . 34
5.2.2. Cechy podpisu XML . . . 36
5.2.3. Przetwarzanie podpisu XML . . . 36
5.3. Podpisy XML w formularzu elektronicznym . . . 39
5.4. Podatności XML . . . 39
5.5. Podatność *Referencji* w podpisie XML . . . 42
5.6. Podsumowanie . . . 43

6. Łańcuchy bloków **45**
6.1. Systemy zcentralizowane, zdecentralizowane i rozproszone . . . 48
6.1.1. Systemy scentralizowane (centralne) . . . 48
6.1.2. Systemy zdecentralizowane . . . 49
6.1.3. Systemy rozproszone . . . 49
6.2. Koncepcja łańcucha bloków . . . 50
6.3. Zalety i problemy łańcuchów bloków . . . 58

6.3.1. Zalety łańcuchów bloków . . . 58
6.3.2. Problemy związane ze stosowaniem łańcuchów bloków . . . 60
6.4. Zastosowania łańcuchów bloków . . . 61
6.5. Wspomaganie realizacji transakcji . . . 62
6.5.1. System przetwarzania transakcji . . . 63
6.5.2. Uwierzytelnianie zapisów medycznych . . . 63
6.5.3. Infrastruktura klucza publicznego . . . 66
6.6. Podsumowanie . . . 71

7. Podsumowanie 73

8. Cel i hipoteza badawcza 76
8.1. Cel badawczy . . . 76
8.2. Hipoteza badawcza . . . 76

9. Formularz elektroniczny 77
9.1. Geneza i założenia . . . 77
9.2. Struktura formularza . . . 78
9.2.1. Atrybuty statyczne . . . 81
9.2.2. Atrybuty dynamiczne . . . 82
9.2.3. Typy przechowywanych danych . . . 83
9.2.4. Definiowanie akcji (działań) . . . 84
9.2.5. Bezpieczeństwo formularza – podpisy . . . 85
9.2.6. Wyrażenia . . . 91
9.3. Trójwarstwowa natura formularza . . . 93
9.3.1. Warstwa danych . . . 93
9.3.2. Warstwa prezentacji . . . 93
9.3.3. Warstwa logiki . . . 94
9.4. Oprogramowanie . . . 95
9.4.1. Aplikacja użytkownika . . . 96
9.4.2. Obsługa podpisów elektronicznych . . . 98
9.4.3. Komponenty wspomagające . . . 100
9.5. Podsumowanie . . . 101

10. Integracja formularzy elektronicznych z łańcuchami bloków 103
10.1. Integracja w oparciu o usługi sieciowe . . . 103
10.2. Integracja poprzez rozszerzenie wbudowanych funkcji formularza elektronicznego . . . 104
10.3. Integracja poprzez wykorzystanie uniwersalnego rozszerzania funkcjonalności formularza . . . 107
10.4. Wykorzystanie łańcuchów bloków do weryfikacji certyfikatów . . . 111
10.5. Podsumowanie . . . 113

11. Transakcje wspomagane formularzem elektronicznym 114
11.1. Cykl życia formularza elektronicznego . . . 114
11.2. Sterowanie realizacją transakcji . . . 115
11.3. Bezpieczeństwo transakcji . . . 120
11.3.1. Podpisy w transakcji . . . 120
11.3.2. Zabezpieczenie protokołu SOAP . . . 123

11.3.3. Bezpieczna poczta elektroniczna 124
11.3.4. Podsumowanie 125
11.4. Rodzaje wspomaganych transakcji 125
11.4.1. Transakcja prosta 125
11.4.2. Transakcja złożona 126
11.4.3. Integracja formularza z systemami 129
11.5. Łańcuchy bloków w transakcjach wspomaganych formularzami elektronicznymi 130
11.6. Interoperacyjność formularzy elektronicznych 133
11.7. Podsumowanie 133

12. Wnioski – realizacja celu i weryfikacja hipotezy badawczej 134
12.1. Model transakcji rozproszonych 137
12.2. Weryfikacja hipotezy badawczej 138

13. Kierunki przyszłych badań 140
13.1. Łańcuchy bloków a formularz elektroniczny 140
13.2. Formularz elektroniczny jako blok łańcucha 140
13.3. Formularz elektroniczny a inteligentny kontrakt 142
13.4. Potencjał formularzy elektronicznych 143
13.4.1. Wysoki poziom dojrzałości i potencjał interoperacyjności 144
13.4.2. Integracja z różnymi technologiami komunikacyjnymi 145

Słownik pojęć 146

Spis rysunków 153

Spis tabel 154

Spis wydruków 155

Literatura 156

A. Składnia formularza elektronicznego 171
A.1. Ogólna struktura formularza 172
A.2. Graficzne komponenty formularza 175
A.3. Dynamiczne atrybuty komponentów formularza 183
A.4. Sytem podpisów 187
A.5. Wyrażenia 190
A.6. Wyrażenia logiczne (`booleanExpression`) 191
A.7. Wyrażenia napisów (`stringExpression`) 196
A.8. Wyrażenia dziesiętne (`decimalExpression`) 201
A.9. Wyrażenia całkowite (`integerExpression`) i „liczbowe” (`numberExpression`) 203
A.10. Wyrażenia daty (`dateExpression`) 205
A.11. Wyrażenia binarne (`mimeExpression`) 207
A.12. Wyrażenia działań - akcji (`actionExpression`) 210
A.13. Wyrażenia dla obsługi kolorów (`colorExpression`) 217
A.14. Wyrażenia dla obsługi obramowań (`borderExpression`) 218
A.15. Wyrażenia dla obsługi czcionki `<fontExpression>` i komponentów `<itemExpression>` 219

A.16. Wyrażenia dla dostępu do klucza `(keyInfoExpression)` i podpisu XML `(xmlDSigExpression)` 221
A.17. Wyrażenia dla dostępu do dokumentu XML *(xmlDocumentExpression)* . . 222

1. Wstęp

Sprawna realizacja transakcji, rozumianych jako wymiana różnego rodzaju dóbr, zawsze była i nadal jest warunkiem rozwoju społeczeństw oraz gospodarek. Ważnym czynnikiem wpływających na tę sprawność jest jakość i sposób obsługi informacji, która zawsze towarzyszy realizacji transakcji. Informacja ta, aby mogła skutecznie wspomagać transakcje i budować wzajemne zaufanie stron, musi być bezpieczna, czyli autentyczna, inetgralna, wiarygodna, trwała, a jej pochodzenie niezaprzeczalne. Informacja o takich cechach jest dokumentem.

Rozwój technologiczny, a szczególnie powstanie technik elektronicznych i informatycznych, otworzyło nowe możliwości dla sprawnego tworzenia, gromadzenia, przetwarzania i przechowywania informacji (dokumentów). Na przestrzeni lat ogólne wymagania dotyczące cech dokumentu nie uległy zmianie, a dokumenty elektroniczne muszą posiadać podstawowe cechy dokumentu tradycyjnego. Ponadto dokumenty elektroniczne posiadają cechy pliku co sprawia, że mogą być one obsługiwane z wykorzystaniem środków elektronicznych. W związku z tym formularze elektroniczne, będące dokumentami elektronicznymi, stanowią użyteczne narzędzie bezpiecznego wspomagania realizacji transakcji, które włącza w ten proces ludzi i ich decyzje.

Formularze elektroniczne funkcjonując w środowisku wirtualnym mogą być przetwarzane przez zróżnicowane technologicznie systemy (aplikacje, usługi), dlatego muszą one spełniać wysokie wymagania dotyczące interoperacyjności w zakresie formatów danych, standardów i konwencji przetwarzania dokumentów oraz elastycznego korzystania z podpisów elektronicznych z zastosowaniem standardowych metod i narzędzi, które zapewniają wysoki poziom bezpieczeństwa realizowanych transakcji. Spełnienie tych wymagań zapewnia trójwarstwowa (warstwa danych, warstwa prezentacji, warstwa logiki) natura zaproponowanego w pracy formularza elektronicznego. Dzięki warstwie logiki możliwe jest sterowanie przebiegiem realizacji transakcji w sposób zapisany w strukturze formularza. Natomiast rozszerzenie funkcjonalności warstwy logiki o mechanizmy komunikacji i obsługi rejestrów rozproszonych pozwala na realizację transakcji w środowisku całkowicie rozproszonym w sposób zintegrowany z łańcuchem bloków. Formularz taki może bezpiecznie wspomagać realizację transakcji w rozproszonym topologicznie i zróżnicowanym technologicznie środowisku.

Celem pracy jest **opracowanie modelu transakcji rozproszonych sterowanych formularzami elektronicznymi**, a sformułowana hipoteza badawcza to **zastosowanie formularza elektronicznego z własną logiką umożliwi realizację transakcji rozproszonych w heterogenicznym środowisku bez używania infrastruktury serwerowej i udziału scentralizowanej infrastruktury PKI**.

Praca składa się z 13 rozdziałów. Rozdział 1 to wstęp. Rozdziały 2-7 opracowano na podstawie przeglądu literatury. W rozdziale 1 zaprezentowano historię transakcji ze szczególnym uwzględnieniem form ich dokumentowania – od tokenów, pisma klinowego, poprzez pieczęcie, do odcisków palców i podpisu odręcznego. Rozdział 3 zawiera definicje transakcji oraz dokumentu elektronicznego i formularza elektronicznego z uwzględnieniem ich roli w osiąganiu wysokich poziomów dojrzałości usług elektronicznych. W rozdziale 4

przedstawiono problematykę interoperacyjności formularzy elektronicznych rozumianą jako zdolność osób i usług zaangażowanych w transakcje do wymiany i wykorzystania informacji dla osiągnięcia określonego celu. W kolejnym rozdziale (5) skoncentrowano się na analizie sposobów wykorzystania podpisów elektronicznych jako narzędzia zabezpieczającego formularz elektroniczny w różnych obszarach. W rozdziale tym omówiono również podatności podpisów XML, a szczególnie podatność referencji w podpisie XMLdSig na manipulację zawartością podpisu. W rozdziale 6 opisano koncepcję i technologię łańcuchów bloków, a także przedstawiono ich zalety oraz wady. Zaprezentowano również przykłady zastosowania łańcuchów bloków (oraz zestawienie ich głównych cech) do realizowania usług PKI w oparciu o rejestry rozproszone. Rozdział 7 zawiera podsumowanie rozdziałów 2-6 i stanowi uzasadnienie dla sformułowanego w pracy celu i hipotezy badawczej, które zamieszczono w rozdział 8.

Treści zawarte w rozdziałach 9-11 stanowią podstawę do realizacji celu pracy oraz weryfikacji hipotezy badawczej. W rozdziale 9 zaprezentowano szczegółową koncepcję formularza elektronicznego wraz z przykładami jego implementacji. Podkreślono rolę jego trójwarstwowej natury z warstwami danych, prezentacji i logiki zbudowanej na wyrażeniach określonego typu. Za pomocą tych wyrażeń możliwe jest definiowanie dynamicznych atrybutów komponentów formularzy, a przez to dostosowanie ich wyglądu do aktualnego stanu realizacji transakcji i sterowania nią. W rozdziale tym pokazano również sposób implementacji systemu podpisów w formularzu, który zapewnia jego bezpieczeństwo w różnych obszarach. Rozdział 10 poświęcono na zaprezentowanie sposobów integracji formularzy elektronicznych z łańcuchami bloków ze szczególnym uwzględnieniem ich zróżnicowania. W rozdziale tym zaprezentowano również ogólny schemat weryfikacji statusu certyfikatów podpisów elektronicznych ze wskazaniem dwóch typów tej weryfikacji. Pierwsza to korzystanie ze standardowych usług infrastruktury klucza publicznego, a druga z wykorzystaniem technologii łańcuchów bloków dedykowanych dla konkretnego systemu. W rozdziale 11 przedstawiono sposoby realizacji transakcji z użyciem formularzy elektronicznych. Zademonstrowano wykorzystanie formularza do sterowania realizacją transakcji o różnym stopniu złożoności oraz omówiono mechanizmy zabezpieczeń transakcji zarówno w obszarze formularza, jak i mechanizmów komunikacji oraz integracji. Przedstawiono również implementacje praktycznych przykładów integracji formularzy elektronicznych z produkcyjnymi systemami obiegu dokumentów, koncepcję integracji logiki formularza z systemami łańcuchów bloków oraz sposoby zapewnienia interoperacyjności w różnych obszarach formularza.

W rozdziale 12 wykazano, że cel pracy został osiągnięty, a sformułowana hipoteza badawcza została pozytywnie zweryfikowana.

Pracę kończy rozdział 13, w którym skoncentrowano się przede wszystkim na potencjalnych możliwościach dalszego rozwoju prac badawczych i inżynierskich w zakresie stosowania oraz rozszerzania funkcjonalności formularzy elektronicznych.

2. Historia – pierwsze transakcje i dokumenty

2.1. Pierwsze transakcje

Transakcje rozumiane jako wymiana dóbr były warunkiem powstania i rozwoju cywilizacji. Towarzyszą one ludziom od czasu kiedy przeszli oni od łowiectwa, zbieractwa i koczowniczego trybu życia do rolnictwa, hodowli i osiadłego trybu życia w okresie neolitu[1] na obszarze tzw. „Żyznego Półksiężyca" (rys. 2.1).

Rys. 2.1: Obszar „Żyznego Półksiężyca" (źródło: domena publiczna)

Okres ten nazywany jest „Rewolucją Neolityczną". To wtedy powstały pierwsze osady tzw. protomiasta[2]. Wynikiem tej „Rewolucji" jest również powstanie rzemiosła i podział pracy między rolnikami, rybakami a rzemieślinikami.

Olbrzymi rozwój gospodarczy świata starożytnego, wzrost liczby ludności, zwłaszcza w Azji[3], i towarzyszący mu wzrost wielkości dochodu „narodowego"[4], objawiał się przede wszystkim rozwojem działalności gospodarczej i ekspansją handlu czemu towarzyszył rozwój sposobów prowadzenia transakcji. W rozliczeniach zaczęto posługiwać się szlachetnymi kruszczami – srebrem i złotem. Wtedy powstał także kodeks Hammurabiego, w którym dużo miejsca poświecono między innymi problemom handlowym.

---

[1] Neolit (epoka kamienia gładzonego) jest to młodsza epoka kamienia (epoka kamienia gładzonego), która najwcześniej, czyli na początku IX tysiąclecia p.n.e., rozpoczęła się na obszarze „Żyznego Półksiężyca" (Izrael, Syria, Palestyna). Później neolit rozpoczął się na takich obszarach jak: Turcja, Irak i Iran (około 8 tys. lat p.n.e.), Chiny (7,5 tys. lat p.n.e.), Pakistan i Europa Południowo-Wschodnia (7 tys. lat p.n.e.), Dolina Gangesu (6,2 tys. lat p.n.e), Europa Środkowa (5,5 tys. lat p.n.e.), Europa Północna (4,5 tys. lat p.n.e.), Ameryka Środkowa (5 - 4,5 tys. lat p.n.e., tzw. okres archaiczny).

[2] Protomiasta były to osady przypominające miasta, których mieszkańcy zajmowali się uprawą roli, jednak część zaczęła zajmować się rzemiosłem. Pierwszym protomiastem było Jerycho w Palestynie.

[3] W czasach antycznych (przed rokiem 1500 p.n.e.) w Azji zamieszkiwało ponad 75% ludności świata, w Europie około 15%, w Afryce około 7%, a resztę świata zamieszkiwało około 3% ludności.

[4] Według danych statystycznych wielkość dochodu w epoce starożytnej wynosiła w przybliżeniu około 100 mld dolarów (według siły nabywczej z 1990 roku), co stanowiło ponad 400 dolarów rocznie na mieszkańca.

W okresie tzw. „Starego Państwa”[5] istotne miejsce w rozwoju gospodarczym ówczesnego świata zaczął zajmować Egipt, który utrzymywał rozległe kontakty międzynarodowe obejmujące między innymi: Nubię, Syrię, Palestynę, Kretę, Cypr, Azję Mniejszą, Mezopotamię.

Kolejnym państwem, które miało duży udział w rozwoju handlu dalekosiężnego, było państwo fenickie, które weszło do historii nie tylko poprzez wynalezienie pisma alfabetycznego, ale również dzięki upowszechnieniu pieniądza[6]. Państwa-miasta fenickie, powstałe około 2000 lat p.n.e. na wschodnim wybrzeżu Morza Śródziemnego, zdominowały handel w Basenie Morza Śródziemnego i Morza Czarnego do X wieku p.n.e.

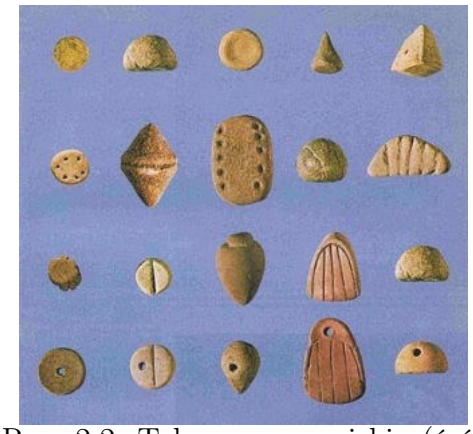
Z rozwojem gospodarki w skali ówczesnego świata ściśle związany był rozwój cywilizacji chińskiej i indyjskiej, a na terenie Europy rozwój gospodarczy był związany z Grecją i Rzymem. Rzymianie zbudowali wielkie imperium, które było dobrze zorganizowane i administrowane oraz oparte na doskonałym systemie prawnym. Na jego obszarze – oprócz rozwoju rolnictwa i rzemiosła – pojawiło się finansowanie robót publicznych związanych między innymi z budową dróg. To dzięki stworzeniu sieci dróg nastąpił rozkwit rzemiosła w prowincjach rzymskich i rozwój handlu z tymi prowincjami. Natomiast handel zewnętrzny Cesarstwa Rzymskiego obejmował Galię, Nadrenię, Germanię i Skandynawię oraz ludy zamieszkujące Europę Środkowo-Wchodnią (szlak bursztynowy). W schyłkowym okresie Cesarstwa, handel objął również Półwysep Arabski, Egipt, Babilonię, oraz Chiny.

Z powyższych rozważań wynika, że zaawansowane systemy organizacji społeczeństw, działaności gospodarczej oraz wymiany dóbr w skali dalekosiężnej (międzynarodowej) były dobrze znane już w starożytności, a gwałtowny rozwój gospodarczy byłby niemożliwy bez realizacji określonych transakcji wraz z ich dokumentowaniem.

2.2. Pismo, dokument, transakcja

Wspomniane w podrozdziale 2.1 rozwój gospodarczy, realizacja zaawansowanych przedsięwzięć (współpracy) i rozwój handlu wymagały stosowania bardziej dojrzałych niż komunikacja werbalna środków utrwalania i wymiany informacji oraz sposobów realizacji transakcji.

Jeżeli przyjmiemy, że podstawowymi środkami porozumiewania się (wymiany informacji) w obrębie określonej społeczności są mowa i pismo, to z punktu realizacji transakcji, to właśnie pismo odgrywa pierwszoplanową rolę. Do najwcześniejszych rodzajów pisma należy zaliczyć pismo sumeryjskie powstałe na bazie tzw. tokenów (rys. 2.2), które można uznać za pierwsze formy zorganizowanej rachunkowości. System rozliczeń oparty na tokenach stosowany był w państwie sumeryjskim od około 8 tys. lat p.n.e. Jego celem był zapis informacji o przekazywanym towarze, dlatego tokeny miały różne kształty (kulki, dyski, czworościany, stożki, walce), które odpowiadały konkretnemu przechowywanemu towarowi



Rys. 2.2: Tokeny sumeryjskie (źródło: domena publiczna)

[5] „Stare Państwo” jest to okres w dziejach starożytnego Egiptu obejmujący lata około 2675 - 2170 p.n.e., przypadający na rządy III - VI dynastii.

[6] Dzięki pracom wykopaliskowym wiadomo, że w V wieku p.n.e. istniało mennictwo fenickie, a pierwsze monety mogły być wybite przez fenickiego lub greckiego przedsiębiorcę w celach reklamowych.

lub hodowanemu zwierzęciu. Wraz z rozwojem rzemiosła, coraz większą liczbą wytwarzanych produktów i w konsekwencji z rozwojem handlu, powstawały lepsze i sprawniejsze sposoby administrownia dobrami. To z kolei doprowadziło do powstania coraz bardziej skomplikowanego systemu tokenów o bardzo zróżnicowanych kształtach. System tokenów przetrwał kilka tysięcy lat.

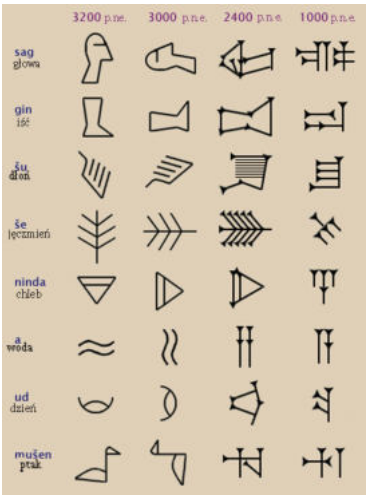
Rys. 2.3: Bulla sumeryjska (źródło: domena publiczna)

Kolejnym etapem w rozwoju pisma sumeryjskiego, związanym z rozwojem handlu (wzrost odległości i udział pośredników), było powstanie tzw. „bulli" (rys. 2.3), czyli glinianych pojemników (kopert), w których zamykano pewną liczbę tokenów oznaczających rodzaj i liczbę towarów będących przedmiotem transakcji. Każdy pojemnik był opatrzony odciskiem pieczęci właściciela przesyłki, który był odpowiednikiem dzisiejszego podpisu. Pieczęć potwierdzała autentyczność przesyłki. Odbiorca mógł rozbić bullę i sprawdzić czy liczba towarów określonych przez tokeny zgadza się z przesyłką. Z czasem na powierzchni pojemników zaczęto umieszczać odciski tokenów, a do odwzorowania nacięć i wyżłobień używano rylca.

Stworzyło to możliwość przeczytania i zinterpretowania dokumentu bez konieczności rozbijania bulli. W momencie, gdy na powierzchni bulli pojawiły się znaki z informacją o jej zawartości, umieszczanie tokenów w jej środku okazało się zbędne. W ten sposób powstały pierwsze tabliczki z odciśniętymi i wyrytymi znakami, które przypominały bulle – miały podobne rozmiary i owalny kształt (rys. 2.4).

Rys. 2.4: Pismo sumeryjskie (źródło: domena publiczna)

Taka forma pisma piktograficznego była kontynuacją pomysłu zapoczątkowanego przez tokeny. Pismo to wykorzystywano do zapisów o charakterze gospodarczym.



Rys. 2.5: Ewolucja pisma sumeryjskiego (źródło: domena publiczna)

Z czasem okazało się jednak, że rysowanie piktogramu odpowiadającego jednemu pojęciu było niepraktyczne i wówczas podjęto próby zapisu dźwięków, czyli pisania wyrazów w oparciu o fonetykę. Oznaczało to odejście od zapisywania obrazów i przejście do używania abstrakcyjnych symboli reprezentujących dźwięki.

Dalsza ewolucja pisma sumeryjskiego poprzez dążenie do jego uproszczenia doprowadziła do tego, że piktogramy wyszły z użycia, gdyż były stopniowo zastępowane kreskami o układzie poziomym, ukośnym i pionowym (rys. 2.5).

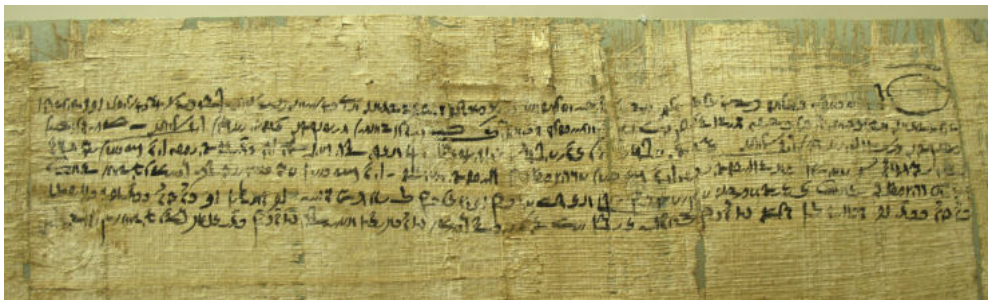
Tak powstało pismo klinowe, które w formie dojrzałej tzw. wyrazowo-sylabowej posiadało bardzo dużą liczbę znaków i było trudne do opanowania. Sumeryjskie pismo klinowe nigdy nie

przekształciło się w alfabet i było oparte na sylabach, a nie na dźwiękach. Pismo klinowe było używane nie tylko w Sumerze, ale również na obszarach sąsiednich przez ludy zamieszkujące Azję Zachodnią.

Pismo klinowe było używane przede wszystkim do zapisu informacji dotyczących działalności gospodarczej i handlu oraz do tworzenia pism urzędowych i prawniczych. Przykładem dokumentu napisanego pismem klinowym jest rozkaz o zwrocie zbiegłej niewolnicy z Tejszebaini z odciskiem pieczęci urzędnika w państwie Urartu[7] (rys. 2.6). Ponadto pismem klinowym zapisywano informacje o obrzędach religijnych, pisano powieści, prywatne listy i wywody naukowe. Niektóre „dokumenty" pisane pismem klinowym posiadały tzw. „kolofony" (podpisy), które zawierały podstawowe informacje podawane przez pisarza na końcu tabliczki: datę (moment napisania), imię piszącego, nazwę miasta, w którym powstała tabliczka.

Rys. 2.6: Dokument z Urartu (źródło: domena publiczna)

Istotną rolę w rozwoju piśmiennictwa odegrało pismo starożytnego Egiptu[8]. Wraz ze zjednoczeniem Egiptu w VII wieku p.n.e. zaczęła się upowszechniać demotyka (rys. 2.7). Najstarsze zapisy w tym piśmie powstały w VII wieku p.n.e., natomiast ostatnie w 476 roku n.e. Było to pismo codziennego użytku, wykorzystywane przede wszystkim do celów administracyjnych i handlowych oraz do pisania korespondencji i sporządzania aktów prawnych, a także do sporządzania podręcznych notatek.



Rys. 2.7: Papirus z czasów Ptolemeuszów, zawierający tekst umowy spisanej demotyką (źródło: domena publiczna)

Pierwszym pismem stosującym zapis alfabetyczny było pismo fenickie[9] (rys. 2.8) po-

[7] Urartu jest to starożytne państwo na Wyżynie Armeńskiej w granicach obecnej Armenii, Turcji i Iranu, istniejące od około XIII do VI wieku p.n.e.

[8] Starożytny Egipt jest to okres cywilizacji egipskiej trwający od okresu predynastycznego (około 3,5 tys. lat p.n.e.) do utworzenia z Egiptu prowincji rzymskiej w czasach panowania Oktawiana Augusta w VII wieku n.e.

[9] Fenicja jest to starożytna kraina na wschodnim wybrzeżu Morza Śródziemnego (dzisiejszy Liban, zachodnia Syria i Izrael). Fenicja, chociaż nie posiadała bogactw naturalnych, była jednym z najlepiej rozwiniętych gospodarczo krajów starożytności. Najważniejszą rolę gospodarczą odgrywało rozwinięte

wstałe w XI wieku p.n.e., z którego wywodzi się większość obecnie używanych systemów pisma[10].

Rys. 2.8: Alfabet fenicki (źródło: domena publiczna)

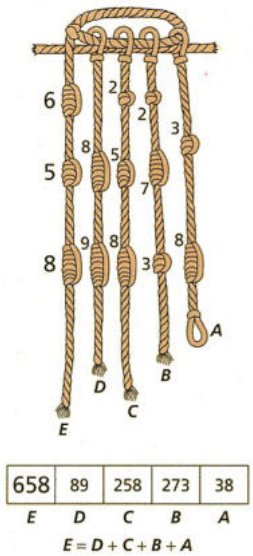
Pismo fenickie składało się z 22 znaków – spółgłosek, które po uzupełnieniu samogłoskami, stało się podstawą alfabetu łacińskiego. Proces powstawania i upraszczania, a następnie upowszchnienia alfabetu fenickiego, zakończył się w XIV-XIII wieku p.n.e. W tym czasie tabliczki gliniane zastąpiono takimi materiałami jak pergamin, skóra, papirus itp.

Alfabet grecki, który wywodzi się z alfabetu fenickiego (poprzez dodanie samogłosek), stanowi jak się wydaje, ostatni etap kształtowania się współczesmego systemu pisma. Powstanie alfabetu było warunkiem powstania języków pisanych.

Historia oraz okoliczności powstawania i rozwoju różnych systemów pism pokazują, że ten rozwój był zawsze ściśle związany z rozwojem państw, kultur i cywilizacji, który z kolei nie byłby możliwy bez istnienia języków pisanych - zaczynając od tokenów, poprzez pisma klinowe i hieroglificzne, a kończąc na pismach alfabetycznych. Wszystkie one pozwalały na prowadzenie różnorodnych zapisów związanych z działalnością handlową, gospodarczą, administracyjną, prawną oraz naukową, kulturalną i religijną.

Oprócz zaprezentowanych powyżej systemów pisma niealfabetycznego i alfabetycznego, które rozwinęły się na obszarze Bliskiego Wschodu i Basenu Morza Śródziemnego, istniało wiele innych wykorzystywanych w praktyce systemów pisma. Przykładem może być tzw. pismo węzełkowe („metoda węzełkowa") stosowane przez Indian prekomlumbijskiej Ameryki Południowej, chińskie i tybetańskie ludy starożytne oraz plemiona z Riukiu[11], wysp polinezyjskich, Afryki Środkowej i Zachodniej Kalifornii, a także z południowego Peru. Na Wyspach Salomona, Karolinach, Palau[12] do dziś przekazuje się wiadomości za pomocą sznurków z węzełkami i pętelkami.

Pismo węzełkowe było zbiorem, wykonanych z bawełny albo włosia lamy lub alpaki, kolorowych sznurków ze supełkami i nosiło nazwę „kipu" (rys. 2.9). Służyło ono do zapamiętywania liczb (najprawdopodobniej w syste-



Rys. 2.9: Kipu (źródło: domena publiczna)

rzemiosło i rolnictwo, a najważniejszym źródłem dochodu był handel dalekomorski oraz eksport towarów luksusowych.

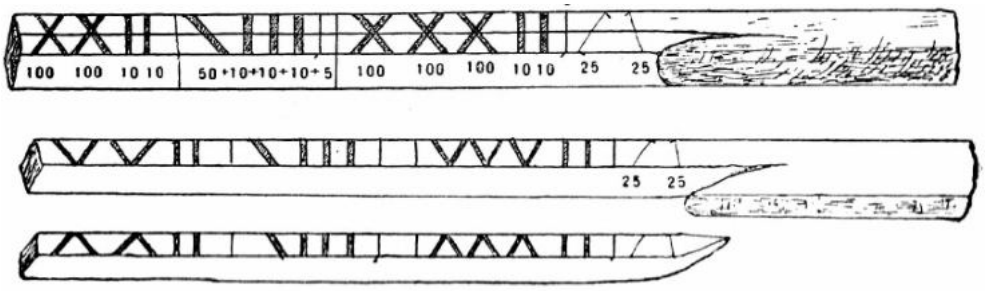
[10] Z pisma feinickiego wykształciły się między innymi: pismo greckie, alfabet łaciński, cyrylica, pismo hebrajskie, arabskie, etiopskie, gruzińske i wiele innych. Spośród współczesnych systemów pism jedynie pismo chińskie, i japońskie nie pochodzą od alfabetu fenickiego.

[11] Riukiu jest to archipelag rozciągający się pomiędzy Tajwanem a Kiusiu należący do Japonii.

[12] Palau jest to wyspiarskie państwo samorządowe stowarzyszone ze Stanami Zjednoczonymi, położone na Oceanie Spokojnym w Oceanii i Mikronezjii około 1160 km na południe od Guam, około 800 km na wschód od Filipin i 3200 km na wschód od Tokio.

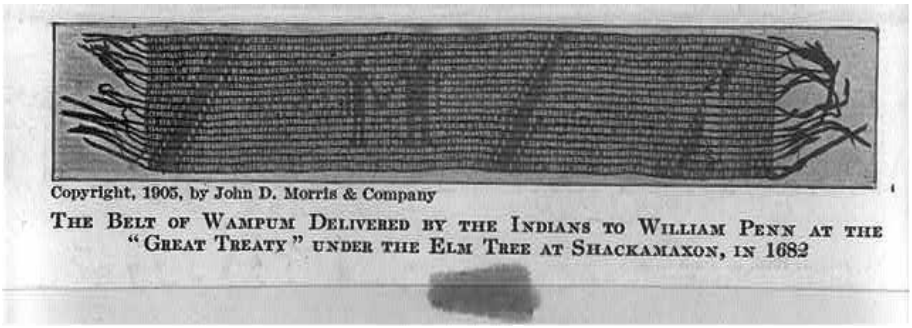
mie dziesiętnym) oraz innych informacji. Mogło przechowywać także dane nienumeryczne (nazwy geograficzne, imię twórcy, właściciela lub przedmiot obliczeń), co czyniłoby z niego pierwotny rodzaj pisma. Używali go członkowie klasy panującej oraz specjalni rachmistrze. Istniało kilka odmian tej metody „zapisywania informacji", które umożliwiały ewidencję ludności, obliczanie podatków, zapisywanie stanów magazynowych, składanie sprawozdań, przechowywanie danych historycznych i kalendarzowych. W początkowym okresie podboju Ameryki Południowej „kipu" było wykorzystywane nawet przez hiszpańskich kolonizatorów do ustalania zobowiązań podatkowych i rozstrzygania sporów majątkowych. Istnieją dowody, że „kipu" było stosowane w okresie około 5000 lat p.n.e., a więc byłoby to jedno z najstarszych pism obok pisma klinowego z Mezopotamii.

Inną metodą należącą do tej samej grupy co pismo węzełkowe (tzw. metody mnemotechniczne) jest karbowany kij - „rabosz" (rys. 2.10) stosowany przez pierwotne ludy Australii, Ameryki Północnej, Afryki Zachodniej, Chin, Mongolii, Azji Południowo-Wschodniej i Skandynawii. Metoda ta znana była również w Polsce (karbowy, rabosz), Rosji (birki), Anglii (tally-sticks) i we Włoszech.



Rys. 2.10: Dwudzielny rabosz z miasta Gilian (Serbia) zapisany cyframi (źródło: domena publiczna)

W średniowiecznej Europie rabosz był kawałkiem drewna z wykonanymi nacięciami. System rozliczeń oparty na raboszu był rozpowszechniony na południu i północy słowiańszczyzny, a także w części Azji, gdzie korzystano zniego od średnoiowiecza do XIX wieku. System ten wykorzywano do rozliczeń podatkowych, rejestrowania informacji o pożyczkach oraz tradycyjnego rozliczania umów w gospodarce pasterskiej. W przypadku stosowania rabosza do rozliczeń przecinano go wzdłużnie na dwa elementy po jednym dla każdej ze stron, a wysokość kontraktu była widoczna dopiero po ich złożeniu.



Rys. 2.11: Wampum traktatowy, który otrzymał od Indian William Penn w 1682 roku (źródło: domena publiczna)

Ostatnim przykładem systemu zapisu z grupy metod mnemotechnicznych może być wampum – rodzaj pasa z koralików i przewierconych muszli splecionych w określony wzór

lub naszywanych na skórze (rys. 2.11). Był on stosowany przez Indian wschodniego wybrzeża USA i służył do utrwalania oraz potwierdzania ważnych wydarzeń politycznych na przykład traktatów takich jak ten, który doprowadził do powstania Ligi Irokezów[13]. Taki pas został użyty do ustanowienia traktatu przez Williama Penn'a[14]. Wampum był również wykorzystywany jako środek płatniczy.

2.3. Rozwój dokumentu i umów-formularzy

Wraz z rozwojem handlu i systemów pisma pojawiły się pierwsze akty praktyki prawnej. W Mezopotamii – w pierwszej połowie trzeciego tysiąclecia p.n.e. – powstały Kodeks Ur-Nammu i Kodeks Hammurabiego. Natomiast w drugim tysiącleciu p.n.e. ustanowiono akty prawne regulujące relacje gospodarcze – przyładem takiego aktu jest „edykt przywracający sprawiedliwość", czyli anulowanie umowy pożyczki zawartej pod przymusem ekonomicznym i jej skutków.

Wśród pierwszych dokumentów prawnych dominowały teksty dotyczące prawa rodzinnego (akty adopcji, kontrakty małżeńskie), spadkowego (testamenty, działy spadku) i zobowiązań (umowy sprzedaży, pożyczki, najmu i dzierżawy). Warto nadmienić, że podobnie jak dzisiaj, w starożytnym Bliskim Wschodzie nie spisywano umów dotyczących drobnych, bieżących spraw życia codziennego – umowy te zawierano ustnie i niezwłocznie wykonywano, nie istniała więc potrzeba sporządzania dokumentu. Natomiast, gdy w grę wchodziły wartościowe ruchomości (niewolnicy, bydło, łodzie, prebendy świątynne[15]) i nieruchomości, umowę zawartą ustnie w obecności świadków spisywano, a powstały dokument miał służyć w przyszłości jako dowód.

Pierwsze dokumenty, które można uznać za umowy sprzedaży, pochodzą z przełomu czwartego i trzeciego tysiąclecia p.n.e. Są one wykute w kamieniu i zaczynają się od określenia powierzchni pola nabywanego przez wskazaną imiennie osobę, następnie wymienia się działki składające się na to pole oraz ich właścicieli. Z czasem tabliczki kamienne zostały wyparte przez tabliczki gliniane (najstarsze z nich pochodzą z 2600-2500 lat p.n.e.), które są lakoniczne i wkazują jedynie cenę oraz cechy przedmiotu sprzedaży, zbywcę lub zbywców jako osoby otrzymujące cenę, nabywcę, który ją wypłaca, listę świadków obejmującą osoby prywatne i urzędników, a niekiedy także „eponima"[16]. Później, taki „okrojony formularz" ulegał rozszerzaniu i zróżnicowaniu. Zaczęto spisywać umowy z punktu widzenia nabywcy, o którym mówiono, że kupił przedmiot sprzedaży. Pojawiły się również znaczące różnice pomiędzy umowami w zależności od przedmiotu umowy (nieruchomości i rzeczy ruchome). W umowach dotyczących nieruchomości zaczęto umieszczać klauzule niepodważalności umowy i zaprzysiężone zobowiązanie stron, że nie będą próbowały wycofywać się z zawartej transakcji. Umieszczano też rękojmię za wady prawne. Zawarciu umo-

[13] Liga Irokezów była to liga plemion Indian Ameryki Północnej zamieszkujących północną część obecnych stanów Nowy Jork, Pensylwania i Maine, niemal cały stan Vermont oraz południowe obszary kanadyjskich prowincji Quebec i Ontario.

[14] Wiliam Penn (14.10.1644 - 30.07.1718) był to angielski kwakier, założyciel Pensylwanii. W październiku 1682 r. Penn pożeglował w górę rzeki Delaware, by po raz pierwszy obejrzeć swoją kolonię. W tym samym roku zawarł swój pierwszy traktat z Indianami z plemion Lenápe reprezentowanymi przez wodza Tamanenda.

[15] Prebenda jest to uposażenie duchownych niepołączone zasadniczo ze sprawowaniem obowiązków duszpasterskich.

[16] Eponim jest to słowo wywodzące się ze starożytności, kiedy do datowania wydarzeń używano imion urzędników eponimicznych. Jego odpowiednikami są: „limmu" w starożytnej Asyrii, „Ervchont" w starożytnych Atenach, „Efor" w Sparcie oraz imiona konsulów w Rzymie.

wy towarzyszyła czynność symboliczna (rytuał) polegająca na „wbiciu gwoździa w ścianę i rozsmarowanie oleju transakcji". Oznaczało to przeniesienie własności oraz stanowiło jego upublicznienie. Często nabywca wyprawiał uroczystą ucztę dla zbywcy i świadków w celu przypieczętowania transakcji, aby zapamiętali oni fakt zawarcia umowy, co było już wtedy elementem zabezpieczającym transakcję oprócz zobowiązania lub przysięgi stron o niepodważalności umowy. Na przestrzeni lat można zauważyć rosnące znaczenie słowa pisanego w prawie i malejące znaczenie czynności symbolicznych. Świadczy to o wzroście rangi bezpieczeństwa stron przy zawieraniu transakcji, które początkowo było oparte na rytuałach, a później na faktach, czyli spisanych dokumentach.

W kolejnych wiekach zmiany treści umowy („formularza") były niewielkie i polegały głównie na jej dostosowaniu do aktualnego stanu zwyczajów prowadzenia handlu. Umowy w dalszym ciągu były sporządzane według typowego formularza oraz zawierały klauzulę o zamknięciu transakcji, czyli o przeniesieniu własności i uiszczeniu zapłaty. Nadal dużą wagę przywiązywano do niepodważalności treści zawartej umowy, dlatego umowę zabezpieczano przysięgą i stosowano kary cielesne lub finansowe w przypadku jej podważania. Ostatnim elemenem umów w dalszym ciągu były lista świadków oraz data zawarcia umowy.

W II tysiącleciu p.n.e. w Asyrii pojawił się w umowach obowiązek przeprowadzenia przez zbywcę procedury publicznego ogłoszenia zamiaru zbycia nieruchomości. Po upływie okresu zgłaszania roszczeń nabywca otrzymywał królewską gwarancję prawa własności. Transakcję odnotowywano na tabliczce sporządzonej w trzech egzemplarzach: dla specjalnego urzędnika, dla władz miejskich i dla nabywcy. Wtedy też w formularzach (umowach) kary za podważanie transakcji zastąpiła klauzula stwierdzająca, że w razie pojawienia się jakichkolwiek roszczeń, dowodem w sprawie będzie tabliczka zawierająca treść umowy. Na obszarach podporządkowanych Hetytom w dokumentach spisanych w obecności króla zawarta była „urzędowa" gwarancja niepodważalności transakcji, a królewska pieczęć zastępowała świadków. W przypadku umów sporządzanych bez obecności władcy konieczni byli świadkowie, a strony zobowiązywały się do niepodważania w przyszłości zawartej umowy.

W zachowanych dokumentach brak jest informacji o transakcjach zawieranych na kredyt lub z odroczoną płatnością. Dlatego przyjmuje się, że sprzedaż miała charakter „sprzedaży za gotówkę", w której spełnienie świadczenia przez jedną ze stron rodziło podobny obowiązek drugiej strony. Jeśli powstała konieczność skredytowania części ceny, korzystano z osobnej umowy pożyczki, a w dokumencie sprzedaży fikcyjnie odnotowywano zapłatę całości, gdyż dopiero w momencie zapłaty pełnej kwoty następowało przejście prawa własności. W ten sposób rozwiązano konieczność zapewnienia atomowości (niepodzielności) transakcji we współczesnym rozumieniu tego pojęcia.

W przedstawionym procesie formowania się transakcji (realizacji umów) można zauważyć wiele „nowoczesnych" elementów bardzo przypominających współczesne regulacje prawne i praktykę działania. Na tej podstawie można stwierdzić, że rozwój transakcji, również w sferze bezpieczeństwa, był uwarunkowany nie tylko zastosowaniem pisma, ale również stosowaniem ustalonych „standardowych" procedur – rytuałów.

2.4. Wiarygodność pism – pieczęcie i podpisy

Z rozważań przedstawionych w podrozdziale 2.3 wynika, że istotnym elementem zapewniającym bezpieczeństwo transakcji, była wiarygodność dokumentów towarzyszących

tej transakcji. Dlatego na końcu dokumentów umieszczano listy świadków zawieranej transakcji, strony składały przysięgi, a także autoryzowano transakcje pieczęcią władcy.

Pieczęć uznaje się za znak własnościowy i rozpoznawczy osoby fizycznej lub prawnej stanowiący zabezpieczenie nienaruszalności dokumentu lub przedmiotu.

Rys. 2.12: Starożytne pieczęcie - a) w formie cylindra, b) w postaci pierścienia/sygnetu (źródło: domena publiczna)

Pierwsze pieczęcie wyrabiano już w epokach kamienia, miedzi, brązu i metalu. Już 3000 lat p.n.e. gliniane naczynia, pochodzące z trzeciego tysiąclecia p.n.e., były oznakowane na obwodzie w celu potwierdzenia ich oryginalności. Jednocześnie to oznakowanie umożliwiało rozpoznanie osoby lub regionu. Starożytne pieczęcie występowały w dwóch formach: sygnetu (pierścienia) oraz cylindra (rys. 2.12, 2.13). Używano ich do odciskania poprzez stemplowanie lub walcowanie w mokrej glinie, z której wykonane były dokumenty, słoje lub bryły. Każda pieczęć musiała być unikalna i indywidualna jak sama osoba oraz wystarczająco wyraźna, aby mogła służyć jako rozpoznawalny podpis jej właściciela. Taki sposób identyfikacji był używany w Galilei od 2000 roku p.n.e. do V wieku n.e.

W starożytności pieczęci używano głównie jako znaku własnościowego i do stwierdzenia nienaruszalności przedmiotów (np. grobowców). Również traktaty pokojowe, świadectwa sprzedaży ziemi, transakcje handlowe, adopcje, małżeństwa, były uwierzytelniane przy użyciu pieczęci, czyli starożytnej wersji podpisu.

Rys. 2.13: Cylindryczna pieczęć z Mezopotamii (źródło: domena publiczna)

W średniowieczu pieczęcie przejęli barbarzyńcy. Używali ich, podobnie jak ich poprzednicy, do gwarantowania nienaruszalności korespondencji i przedmiotów oraz jako znaku rozpoznawczo-własnościowego. Na zachodzie Europy od XI wieku pieczęcie stały się pełnoprawnym środkiem legalizacji dokumentu, a europejscy władcy wykorzystywali je do znakowania dekretów (potwierdzenie zawarcia umowy lub wypowiedzenie wojny).

Uszkodzenie lub brak pieczęci powodowały uznanie dokumentu za nieważny. W tym samym czasie w Chinach stemple były wykonywane z kamienia i wykorzystywano je do potwierdzania autentyczności podpisywanych dokumentów. W miarę upowszechniania pisma pieczęcie wykorzystywano do zamykania korespondencji i przedmiotów w celu kontroli ich nienaruszalności.

Rys. 2.14: Bulla papieża Klemensa VI z 1342 roku (źródło: domena publiczna)

Od XII wieku papieże, do znakowania ważnych dokumentów, wykorzystywali pieczęć metalową tzw. „Pieczęć Rybaka” (rys. 2.14). W 1166 roku papież Aleksander II wydał oficjalny dekret, w którym nadał pieczęciom rangę środka legalizacji dokumentów. Aby uniknąć jej podrobienia po śmierci papieża pieczęć jest niszczona.

Z powyższego wynika, że przez wiele tysiącleci pieczęć była jedynym środkiem uwierzytelniania dokumentów, a podpis nie był stosowany, chociaż pismo znane było od czasów sumeryjskich.

Podpis jest to językowy znak graficzny utrwalony w dokumencie własnoręcznie przez podpisującego. Wskazuje on imię, nazwisko albo przezwisko, lub znak „X”. Podpis umożliwia identyfikację podpisującego i może pełnić rolę pieczęci.

Pierwszy podpis w znanej postaci historycznej został złożony dopiero 1069 roku w średniowiecznej Hiszpanii przez szlachcica „El Cida”. W 1677 roku parlament Anglii w „Statucie Oszustw” (Statute of Frauds) uznał podpis za oficjalne upoważnienie do zawierania umów. To zapoczątkowało powszechne użycie podpisu. Każdy, kto potrafił czytać i pisać bez względu na to kim był, zaczął podpisywać się swoim nazwiskiem pod umowami.

Rys. 2.15: Indyjski dokument prawny z 1952 roku, w którym odcisk palca traktowany jest jako podpis (źródło: domena publiczna)

Tradycyjną funkcją podpisu jest trwałe umieszczenie na dokumencie jednoznacznej, osobistej i niezaprzeczalnej identyfikacji danej osoby jako fizycznego dowodu osobistego poświadczenia treści całego dokumentu lub jego określonej części. Podpisy mogą być poświadczone oraz zarejestrowane w obecności notariusza i wówczas mają dodatkową moc prawną. Na dokumentach prawnych niepiśmienny sygnatariusz może postawić znak (często „X" lub inny spersonalizowany symbol), ale wtedy dokument musi być kontrasygnowany przez piśmiennego świadka. W niektórych krajach niepiśmienni składają odcisk kciuka na dokumentach prawnych zamiast pisemnego podpisu (rys. 2.15).

Pieczęć razem z podpisem stała się środkiem legalizacji dokumentów. Obecnie pieczęcie są używane przede wszystkim przez władze państwowe, a w dalszej kolejności przez osoby prywatne, i pełnią funkcję wyłącznie znaku wierzytelności dokumentu. Wynika z tego, że w dzisiejszych czasach to podpis stał się podstawowym sposobem uwierzytelniania wszelkiego rodzaju dokumentów.

W XX wieku postęp technologiczny zrewolucjonizował podpis. W latach 80-tych XX wieku do przesyłania podpisanych dokumentów powszechnie wykorzystywano faks. Niedługo później pojawiły się pierwsze e-podpisy. W 2000 r. podpisano Electronic Signatures in Global and National Commerce Act, który uznał e-podpisy za prawnie wiążące.

2.5. Podsumowanie

Zaprezentowane w rozdziale 2 rozważania pokazują, że transakcja jako zorganizowana wymiana dóbr pomiędzy ludźmi, powstała w czasach, kiedy człowiek zaczął prowadzić osiadły tryb życia i głównym źródłem utrzymania stało się rolnictwo oraz hodowla. Można zatem uznać, że transakcje są fundamentem współczesnej gospodarki, gdyż były i są immamentnie związane z rozwojem wszystkich cywilizacji na przestrzeni dziejów niezależnie od czasu i lokalizacji geograficznej. Ekspansja handlu nie byłaby możliwa bez zawierania i realizowania transakcji zgodnie z ustaloną procedurą. Podobnie rozwój państwowości od pierwszych organizacji pańsw cywilizacji sumenryjskej, poprzez starożytny Egipt, Rzym, i państwa średniowieczne, aż do państw z czasów współczesnych, był i jest oparty na transakcjach, czyli zorganizowanej i bezpiecznej wymianie dóbr. Należy podkreślić, że wymieniane dobra nie muszą być tylko materialne, ale mogą być również niematerialne i za takie dobro niematerialne należy uznać informację. Jednak informacja nie ma większej wartości, jeżeli nie jest informacją bezpieczną, czyli pewną i wiarygodną. Znaczenie bezpiecznej informacji rosło w miarę jak wymiana odbywała się na coraz większe odległości i była realizowana przez pośredników. Prowadzenie handlu w taki sposób (duże odległości i funkcjonowanie pośredników) wymagało stosowania trwałych form zapisywania i przenoszenia informacji. Już podczas realizacji najstarszych transakcji przywiązywano dużą wagę do ich bezpieczeństwa. Zapewnienie bezpieczeństwa informacji podczas realizacji pierwszych transakcji było jedną z głównych przyczyn powstania pierwszego pisma, kiedy to tokeny, umieszczane w bulli zabezpieczone odciskami stron transakcji, przekształciły się w tabliczki zapisywane pismem klinowym. Zapisy te były zabezpieczane poprzez wykonywanie odcików pieczęci cylindrycznych. Już wtedy pojawiły się swoiste podpisy - „kolofony", zawierające informacje pozwalające na identyfikację autora i miejsca, w którym powstał dany dokument.

3. Transakcje

Jak pokazano w rozdziale 2, transakcje towarzyszyły człowiekowi od kiedy znane są historyczne ślady jego działalności. To własnie rozwój działalności gospodarczej warunkował rozwój cywilizacyjny ludzkości, a działalność gospodarcza zawsze była immamentnie związana z transakcjami. Nie wnikając w to, czy transakcje były podstawą działań gospodarczych, czy też działania gospodarcze warunkowały powstawanie transakcji, należy stwierdzić, że transakcje były podstawą funkcjonowania gospodarki i nadal są jej fundamentem. Ze względu na fundamentalne znaczenie transakcji w działalności gospodarczej, należy zastanowić się nad tym czym jest transakcja i dokonać próby jej zdefiniowania.

3.1. Definicje transakcji

W różnych pracach w bardzo podobny sposób zostało opisane i zdefiniowane pojęcie transakcji ([213, 221, 72, 1, 191, 74, 193, 210, 192, 21]).

Na ich podstawie można przyjać, że transakcja jest to szereg czynności, działań jednej, dwóch lub większej liczby stron (transakcje jednostronne, dwustronne lub wielostronne), zmierzających do osiągnięcia określonych celów. Mogą to być cele handlowe, gospodarcze, społeczne lub prawne. Przedmiotem transakcji może być stworzenie, zmiana lub transfer dóbr materialnych lub niematerialnych (prawnych), a także realizacja usługi, nawiązanie współpracy, wymiana towarów, wymiana aktywów finansowych, informacji, usług, wiedzy (patenty).

Natomiast w pracy [136] zauważono, że *transakcja to działanie lub czynność komunikacyjna z udziałem dwóch stron lub rzeczy, które wzajemnie na siebie wpływają lub oddziałują.*

Pojęcie *transakcji* jest również popularne w technologiach informatycznych, a zwłaszcza bazach danych. Według [51] *transakcje* są: *jednym z podstawowych pojęć współczesnych baz danych. Umożliwiają one współbieżny dostęp do zawartości bazy danych dostarczając niezbędnych mechanizmów synchronizacji. Istotą transakcji jest integrowanie kilku operacji w jedną niepodzielną całość.*

W opisie transakcji (informatycznych) [214] oprócz definicji: *„Transakcją w dziedzinie informatyki jest zbiór operacji na bazie danych, które stanowią w istocie pewną całość i jako takie powinny być wykonane wszystkie lub żadna z nich”*, wskazano te istotne cechy transakcji, które mogą mieć zastosowanie również do transakcji biznesowych. Cechami tymi są przede wszystkim zasady ACID (atomicity, consistency, isolation, durability – niepodzielność, spójność, izolacja, trwałość).

Warto wspomnieć, że w pracy [89] transakcji przypisano przymiot skończoności, podkreślając, że transakcją jest to wykonana, zrealizowana, zakończona umowa, co skutkuje osiągnięciem celu umowy.

Przykładem transakcji może być transakcja bankowa jaką jest przelew. W trakcie tej transakcji muszą być wykonane dwie operacje (czynności) – zabranie pieniędzy z jednego konta oraz dopisanie ich do drugiego konta. W przypadku niepowodzenia żadna z tych

operacji nie może być zatwierdzona. W dziedzinie informatyki przykładem transakcji może być transakcja w systemie plików.

Biorąc pod uwagę powyższe rozważania dotyczące sposobów definiowania transakcji oraz zgodnie z wcześniejszą pracą autora [207] można stwierdzić, że:

— transakcja jest zorganizowanym, zaplanowanym ciągiem działań (czynności) wynikających z relacji wiążących strony (umowa, kontrakt, przepisy prawa),
— przedmiotem transakcji może być transfer dóbr materialnych bądź niematerialnych, a także usługi lub współpraca,
— szczególnym przedmiotem transakcji może być informacja lub wiedza,
— cele transakcji mogą mieć różny charakter (np. handlowy, gospodarczy, społeczny, prawny),
— w transakcji uczestniczy co najmniej jedna strona, ale najczęściej jest ich więcej,
— realizacji każdej transakcji towarzyszy przetwarzanie informacji w różnej postaci,
— realizacja transakcji może wymagać skorzystania z innych transakcji pełniących rolę usługi.

Uwzględniając wymienione powyżej najważniejsze cechy charakteryzujące transakcje sformułowano następującą definicję tego pojęcia:

> ***Transakcja jest ustaloną formalnie sekwencją działań wynikającą z ustaleń umownych lub innych relacji wiążących jedną lub większą liczbę stron. Działania te mają na celu osiągnięcie zamierzonych celów (gospodarczych, handlowych, społecznych, prawnych). Transakcje są związane z dowolną działalnością, a przedmiotem transakcji mogą być zarówno dobra materialne, jak i niematerialne.***

Zawarte w podanej definicji cechy transakcji są niezależne zarówno od formy informacji, jak i sposobów jej przetwarzania. Forma informacji oraz sposoby jej przetwarzania będą przedmiotem rozważań w kolejnych rozdziałach rozprawy ze względu na szczególne relacje jakie wiążą informacje z transakcjami.

3.2. Informacja w transakcji

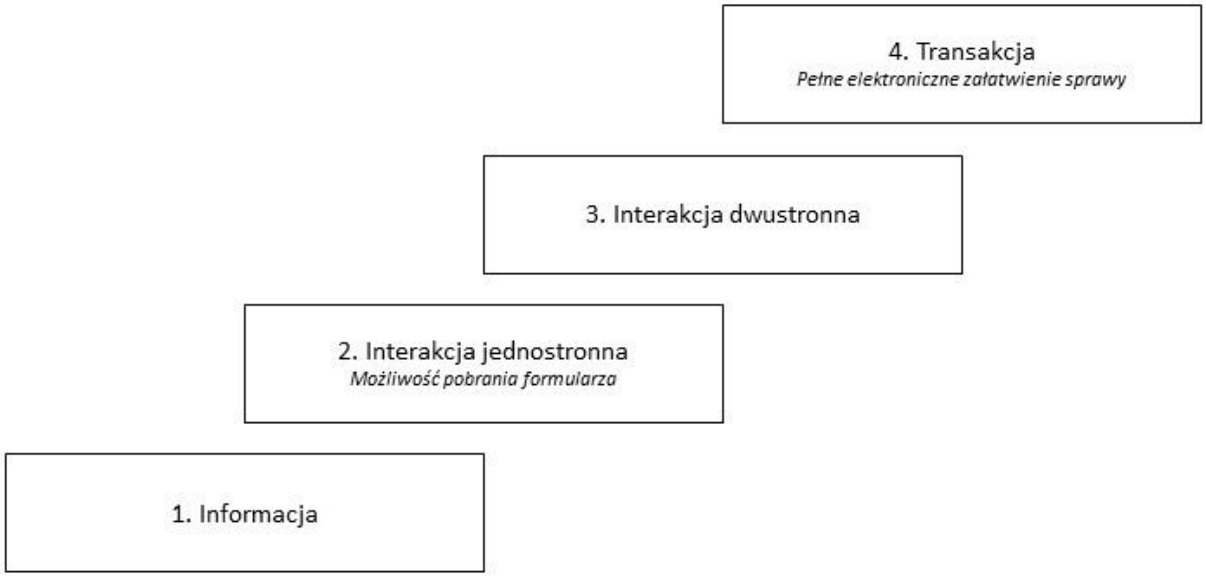
W dotychczasowych rozważaniach wielokrotnie stwierdzono (rozdział 2 oraz podrozdział 3.1), że wymiana informacji nieodłącznie towarzyszy wymianie dóbr, czyli realizacji transakcji. Na przestrzeni dziejów, a zwłaszcza w ostatnim czasie, znaczenie informacji rosło. Informacja sama w sobie stała się przedmiotem transakcji jako tzw. „dobro niematerialne" (prawne) na równi z tzw. „dobrami materialnymi". Przykładami takich transakcji mogą być patenty, wiedza, świadectwa udziałów itd. Również w obszarze wymiany kapitału, zwykle towarzyszącego realizacji transakcji, rola informacji ciągle rosła. O ile na początku dziejów „kapitałem" były dobra wymieniane w ramach tzw. handlu wymiennego, to poprzez wymianę kruszców szlachetnych, a potem monet jako równoważników złota i srebra, zaczęto stosować dokumenty (noty bankowe – banknoty), czyli bezpieczną informację o posiadanych w banku zasobach kruszców. W końcu banknoty i monety zostały zastąpione informacją elektroniczną, a wymiana kapitału jest realizowana poprzez wymianę danych pomiędzy systemami i kontami informatycznymi.

Obecnie rola informacji w realizacji transakcji jest niepodważalna i uwidacznia się w dwóch obszarach – po pierwsze jako dane i parametry realizowanej transakcji na różnych etapach, a po drugie jako przedmiot transakcji. Dlatego też w niniejszej pracy informację, a w szczególności informację elekrtoniczną, oraz jej rolę w realizacji transakcji uznano za

kluczową, a za podstawowe miejsce jej zapisywania (utrwalania) należy uznać dokumenty elektroniczne. Pojawienie się tych dokumentów oraz związane z tym możliwości ich przesyłania i przetwarzania dały z jednej strony możliwość świadczenia usług elektronicznych, które możemy utożsamić z transakcjami, a z drugiej strony postawiły nowe wymagania techniczne. Przydatność konkretnych implementacji realizacji transakcji wspomaganych elektronicznie w kontekście realizacji elektronicznych usług publicznych może być oceniona za pomocą taksonomii tzw. *Dojrzałości Usług Elektronicznych*.

3.3. Dojrzałość usług elektronicznych

Na zlecenie Komisji Europejskiej firma Cap Gemini Ernst and Young opracowała czterostopniową skalę oceny postępów inicjatywy e-Europe [36]. Jej zadaniem była ocena stopnia dostępności usług online dla obywateli i biznesu. Rysunek 3.1 pokazuje kolejne poziomy dojrzałości usług elektronicznych [204].



Rys. 3.1: Poziomy dojrzałości usług elektronicznych według Cap Gemini (źródło: [204])

Poszczególne poziomy, przedstawione na rysunku 3.1, oznaczają stopień zaawansowania usługi:

0 - usługa nie jest wspomagana elektronicznie – brak publicznie dostępnej strony internetowej lub strona ta nie spełnia wymagań określonych dla poziomów 1-4,

1 - informacja – opublikowano stronę internetową z informacjami o świadczonych usługach oraz opisem sposobu uruchomienia procedury,

2 - interakcja jednostronna – możliwe jest pobranie (download) formularza realizacji usługi, a następnie (np. w późniejszym terminie) wydrukowanie, wypełnienie i podpisanie go przez interesanta oraz dostarczenie go w celu uruchomienia realizacji usługi,

3 - interakcja dwustronna – istnieje możliwość wypełnienia „formularza" na stronie internetowej lub elektronicznego złożenia wniosku albo żądania, przy czym wymagane jest przetwarzanie „formularzy" z uwzględnieniem uwierzytelnienia,

4 - transakcja – możliwe jest pełne elektroniczne załatwienie sprawy łącznie z podejmowaniem decyzji, informowaniem zainteresowanych stron, potwierdzaniem, dostawą i płatnością.

Osiągnięcie najwyższego poziomu realizacji usługi nie zawsze oznacza wykonanie usługi na poziomie czwartym, czyli na poziomie transakcji. Okazuje się, że czasami realizacja usług na najwyższym poziomie wymaga osiągnięcia zaledwie poziomu drugiego, czyli informacji jednostronnej - na przykład dostarczenie informacji do urzędu nie musi wymagać podjęcia decyzji przez urzędnika. Dlatego kluczowym aspektem rozwoju usług elektronicznych jest dostępność usług publicznych dla obywateli i biznesu. Niemniej jednak należy zauważyć, że osiągnięcie trzeciego a zwłaszcza czwartego poziomu dojrzałości, wymaga:

— zapewnienia autentyczności, integralności i niezaprzeczalności pochodzenia wymienianych informacji, czyli środków skuteczności prawnej na poziomie wymaganym dla konkretnej transakcji (dla różnych transakcji wymagany może być różny poziom bezpieczeństwa informacji), oraz
— automatyzacji realizacji usługi, czyli automatyzacji przetwarzania informacji związanej z realizacją usługi.

Jak wspomniano powyżej wymagania dotyczące poziomu bezpieczeństwa informacji mają charakter relatywny i zależą od rodzaju transakcji. Dla prostych transakcji o niewielkich konsekwencjach, będących skutkiem nadużyć wynikających z niedotrzymania wysokich standardów bezpieczeństwa, nie ma konieczności spełnienia wszystkich celów (wymogów) bezpieczeństwa, a nawet nie trzeba w ich przypadku korzystać z bezpiecznej informacji. Przykładem może być transakcja zakupu taniego produktu w sklepie. Natomiast transakcje charakteryzujące się dużą wrażliwością i kosztownymi konsekwencjami nadużyć wymagają korzystania z bezpiecznej oraz skutecznej prawnie informacji. Dojrzałość usługi (transakcji) elektronicznej powinna być zatem rozważana z kilku punktów widzenia dotyczących:

— możliwości stosowania zaawansowanych środków przesyłania i przetwarzania informacji, w tym integracji z dziedzinowymi systemami wewnętrznymi użytkowników,
— możliwości zapewnienia bezpieczeństwa informacji na poziomie wymaganym dla realizacji określonej transakcji,
— możliwości wykorzystania informacji w rozproszonym środowisku heterogenicznym technologicznie.

3.4. Przetwarzanie i transfer informacji w transakcji

Wymiana i przetwarzanie informacji towarzyszących różnym procesom, na przykład gospodarczym lub administracyjnym, odbywa się na wiele sposobów – od ustaleń ustnych poprzez wykorzystanie listów, faksów, poczty elektronicznej, aż do dokumentu elektronicznego.

W pracy [207] autor sformułował i przedstawił cechy nowoczesnego dokumentu elektronicznego, który może być w szerokim zakresie wykorzystany w realizacji transakcji z uwzględnieniem integracji z różnymi technologiami komunikacji. Cechy te to:

A niezaprzeczalność pochodzenia (patrz def. 16), integralność (patrz def. 13) i autentyczność – wiarygodność (patrz def. 29), czyli cechy budujące skuteczność prawną (bezpieczeństwo prawne) dokumentu,

B autonomiczność, która oznacza, że dokument stanowi pojednyńczy, zamknięty jako całość samodzielny, niezależny byt (zamykający dane i sposób prezentaji w jednym dokumencie), co daje możliwość korzystania z dokumentu bez konieczności dostępu do sieci,

C zrozumiałość dla ludzi, czyli wizualizacja oparta na uniwersalnych, powszechnie znanych zasadach stosowanych w dokumentach tradycyjnych,

D interoperacyjność, czyli możliwość interpretacji przez różne podmioty zaangażowane w realizację transakcji, co z kolei zapewnia łatwość automatycznego przetwarzania i wymiany danych z różnymi systemami informacyjnymi,

E możliwość definiowania i realizacji logiki – reguł poprawności, reguł obliczania i prezentowania wartości, reguł dynamicznej prezentacji obiektów z użyciem wbudowanych funkcji kryptograficznych i funkcji infrastruktury klucza publicznego, oraz wbudowanych mechanizmów komunikacji (poczta elektroniczna, integracja z protokołami usług sieciowych),

F możliwość łatwej integracji z nowopowstającymi technologiami takimi jak: kryptowaluty, łańcuchy bloków, rozproszone organizacje autonomiczne, protokoły inicjalizacji sesji (SIP - Session Initialisation Protocol).

Ponadto w wyżej wymienionej publikacji zaprezentowano popularne formaty oraz metody, czyli technologie, wymiany informacji wspomagające realizację transakcji (pliki w formacie PDF [54, 55]), pliki w formacie popularnych pakietów biurowych (np. MS Office [2], LibreOffice [118]), poczta elektroniczna – SMTP [104], faks, zcentralizowane systemy informatyczne, oraz oceniono je z punktu widzenia wymienionych powyżej cech (od A do E) dokumentu elektronicznego. Wyniki tej oceny przedstawiono w tabeli 3.1.

Technologie	Cechy					
	A	B	C	D	E	F
Pliki w formacie PDF	✓	✓	✓	✗	✗	✗
Pliki w formacie pakietów biurowych	✗	✓	✓	✗	✗	✗
Poczta elektroniczna	✗	✓	✓	✗	✗	✗
Faks	✗	✓	✓	✗	✗	✗
Scentralizowane systemy informatyczne	✓	✗	✗	✗	✗	✓

Tabela 3.1: Technologie wymiany informacji a cechy nowoczesnego dokumentu elektronicznego (źródło: [207])

Zaprezentowane w tabeli 3.1 przykłady technologii wymiany informacji nie oferują wyrafinowanych i zaawansowanych technik zabezpieczania oraz przetwarzania informacji. Jednak są one powszechnie stosowane przy realizacji prostych transakcji. Realizacja tych transakcji zachodzi pomiędzy stronami darzącymi się nawzajem zaufaniem, a ich wartość jest zazwyczaj stosunkowo niewielka. W związku z tym ryzyko wystąpienia pomyłek lub nadużyć jest nieduże, a konsekwencje błędów są akceptowalne przez strony (patrz podrozdział 3.3). Ponadto można zauważyć, że rozwiązania te mają zastosowanie dla mniej znaczących transakcji, które są realizowane bezpośrednio pomiędzy stronami bez korzystania z bardziej scentralizowanych systemów informatycznych, a więc w zdecentralizowanej sieci transakcji. W przypadku poważniejszych transakcji, które są wrażliwe na manipulacje, konieczne jest zastosowanie odpowiednich mechanizmów zabezpieczających transakcje. Pomocne w tym przypadku jest korzystanie z mniej lub bardziej scentralizowanych systemów dedykowanych właśnie dla takich transakcji.

Z powyższych rozważań wynika, że wspomaganie realizacji transakcji lub inaczej mówiąc usług, może być zaimplementowane w postaci systemów informatycznych. Można wskazać dwa typy takich rozwiązań: scentralizowane systemy dziedzinowe i usługi siecio-

we w architekturze SOA (Service Oriented Architecture), przy czym podstawą realizacji nowoczesnych transakcji jest wykorzystanie SOA [200], [165].

Nowoczesne implementacje oprogramowania wspomagającego realizację transakcji tworzone są zgodnie z paradygmatem architektury zorientowanej na usługi (Service Oriented Architecture - SOA), który doczekał się wielu opracowań (np. [11, 77, 150, 224, 226, 78, 150, 224]). W tych implementacjach podstawą wymiany komunikatów podczas realizacji usługi jest zwykle protokół SOAP ([139, 134, 141, 100]), a podstawowym formatem danych jest XML ([26, 27, 127, 189]). Natomiast szczegóły implementacji usługi są publikowane w sformalizowanej postaci za pomocą dedykowanego języka opisu usług sieciowych (Web Service Description Language - WSDL) [121, 211, 178]), który jest podstawą do opracowania interakcji z usługą.

Obecnie transakcje realizowane są w dedykowanych środowiskach systemów informatycznych zaimplementowanych w różnych technologiach (np. webowych) [91]. W systemach tych użytkownicy są rejestrowani, a następnie uwierzytelniani. Skuteczność prawna transakcji jest zapewniona poprzez rejestrację poszczególnych działań uwierzytelnionego użytkownika przy założeniu, że system jest zaufany. Może to prowadzić do sytuacji, w której osoba uczestnicząca w wielu transakcjach musi być zarejestrowana w wielu systemach i dynamicznie przełączać się między systemami dziedzinowymi. Niektóre systemy rozwiązują ten problem poprzez zastosowanie systemów jednokrotnego logowania [83, 202, 95, 66]. Innym sposobem na zmniejszenie liczby systemów realizujących transakcje jest integracja wielu usług w większe systemy zwane platformami (np. polska platforma kontaktu z obywatelami i urzędami ePUAP (elektroniczna Platforma Usług Administracji Publicznej) [73]. Budowanie większych, bardziej złożonych rozwiązań wymaga współpracy pomiędzy różnymi systemami, a integracja ta jest zwykle realizowana z wykorzystaniem architektury zorientowanej na usługi (SOA) [32]. W tym podejściu usługi te stanowią interfejs dostępu do usług świadczonych przez poszczególne systemy.

Z punktu widzenia złożoności systemu prostsze wydaje się rozwiązanie oparte na wymianie tradycyjnych dokumentów. Użytkownik składa oświadczenie woli (wniosek) w formie formularza posiadającego wszystkie cechy dokumentu i to oświadczenie jest podstawą do wykonania poszczególnych kroków transakcji. W tym tradycyjnym podejściu nie jest wymagana rejestracja uczestników transakcji w systemie. Cały proces transakcyjny opiera się na wiarygodnych dokumentach jako nośnikach informacji, które są podstawą podejmowanych działań. Z tego samego powodu zastosowanie dokumentu elektronicznego, będącego odpowiednikiem – metaforą dokumentu papierowego, umożliwia budowę systemów informatycznych wspomagających realizację transakcji, które odzwierciedlają rzeczywiste procesy. Systemy takie są prostsze, gdyż na przykład nie trzeba utrzymywać kont użytkowników – interakcja i wymiana informacji pomiędzy systemem a użytkownikiem lub innymi systemami odbywa się z wykorzystaniem dokumentów, które stanowią bezpieczną formę dzięki zastosowaniu podpisów elektronicznych zapewniających cechy informacji wymienione w podrozdziale 3.4 punkt A. Ponadto, takie podejście do wspomagania realizacji transakcji pozwala uniknąć konieczności implementowania interakcji interfejsów użytkownika, ponieważ środkiem gromadzenia i pezentowania informacji są formularze zabezpieczone podpisami elektronicznymi. Formularze elektroniczne są łatwiejsze do zaakceptowania przez człowieka, gdyż odtwarzają znane stereotypy zachowań, które występują w przypadku realizacji procesów z wykorzystaniem tradycyjnych dokumentów znanych przez użytkownika, a jednocześnie dzięki ich elektronicznej formie mogą być w łatwy sposób przetwarzane i interpretowane w automatycznych systemach informatycznych.

3.5. Dokumenty elektroniczne

Aby informacja mogła być skutecznie i powszechnie stosowana w realizacji transakcji musi spełniać stosowne wymogi poziomu dojrzałości (podrozdział 3.3) i bezpieczeństwa. Należy do nich zaliczyć: niezaprzeczalność, autentyczność-wiarygodność, interoperacyjność (omówione w podrozdziale 3.4) oraz trwałość rozumianą jako możliwość przechowywania informacji na trwałym, odpornym na upływ czasu nośniku [204].Takie cechy są specyficznymi cechami ogólnie rozumianego dokumentu, który może być zdefiniowany jako (patrz def. 5):

> ***każdy przedmiot, który może być wykorzystany jako przedmiot prawa potwierdzając określony stosunek prawny, okoliczności lub fakty jakiejś sprawy.***

Szczególnym typem dokumentu jest dokument elektroniczny, którego pojawienie się jest konsekwencją powstania elektronicznych technik obliczeniowych. Dokumentem elektronicznym jest (patrz def. 8):

> ***dokument w postaci pliku (patrz def. 18) tekstowego, graficznego, muzycznego, filmowego lub mieszanego będącego wynikiem pracy z danym programem komputerowym, dający się zapisać, a następnie odczytać. Posiada cechy dokumentu, czyli potwierdza prawdziwość zaistnienia danego wydarzenia, okoliczności, zjawiska oraz cechy pliku. Może być więc dowodem w procedurach prawnych.***

Podobną definicję można znaleźć między innymi w publikacjach [28] i [1], przy czym w tej ostaniej zapisano dodatkowo, że dokument może stanowić zapis, czyli *konkretną część zarejestrowanej informacji wytworzonej, zebranej lub otrzymanej w trakcie rozpoczynania, prowadzenia lub kończenia działania, która zawiera wystarczającą treść, kontekst i strukturę, aby dostarczyć dowód tego działania.*

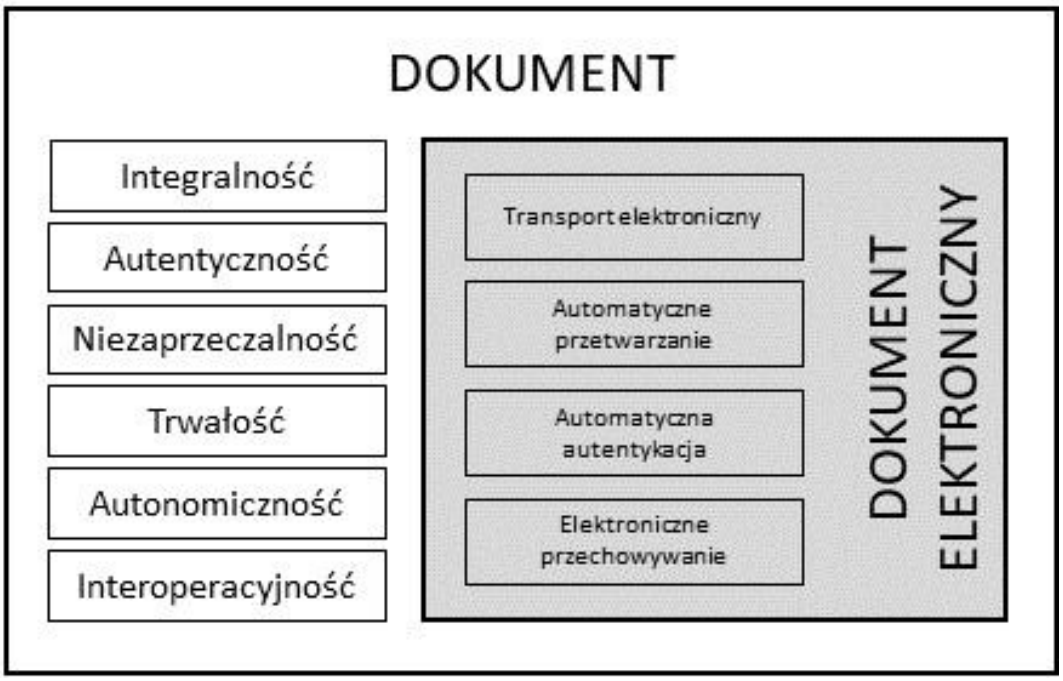
Z kolei zgodnie z *„Ustawą z dnia 17 lutego 2005 r. o informatyzacji działalności podmiotów realizujących zadania publiczne”* [196] dokumentem elektronicznym jest (patrz def. 9) *stanowiący odrębną całość znaczeniową zbiór danych uporządkowanych w określonej strukturze wewnętrznej i zapisany na informatycznym nośniku danych* (patrz def. 12).

Wynika z tego, że dokument elektroniczny jest dokumentem przydatnym – z prawnego punktu widzenia – w realizacji dowolnych transakcji. Ponadto poprzez fakt, że jest plikiem danych rozumianym jako *uporządkowany, posiadający określoną składnię-strukturę i skończony zbiór danych stanowiących dla użytkownika całość*, dokument elektroniczny daje szereg nowych możliwości i korzyści w stosunku do klasycznych form dokumentu. Do najważniejszych zaliczyć można przede wszystkim możliwość jego automatycznego przetwarzania oraz elektronicznego, niematerialnego przesyłania [156]. Możliwości te wprowadziły nową jakość w realizacji transakcji zwłaszcza w zakresie czasu przesyłania i przetwarzania, a także bezpieczeństwa.

Dokument elektroniczny, oprócz cech dokumentu tradycyjnego, posiada dodatkowe cechy specyficzne tylko dla niego, takie jak:

— możliwość transportu z wykorzystaniem środków elektronicznych – eliminacja fizycznego transportu dokumentu od nadawcy do odbiorcy,
— możliwość automatycznego przetwarzania (odczyt, interpretacja, przetwarzanie i zapis danych dokumentu),
— możliwość przechowywania dokumentów w zasobach elektronicznych – nie jest konieczne korzystanie z fizycznych archiwów dokumentów,

— automatyczne przetwarzanie – możliwość tworzenia i weryfikacji cech bezpieczeństwa. Ogólne cechy dokumentu i specyficzne cechy dokumentu elektronicznego zaprezentowano na rysunku 3.2.



Rys. 3.2: Dokument (ogólnie) a dokument elektroniczny (źródło: [204])

Oprócz omówionych powyżej cech dokumentu elektronicznego, stanowiących zbiór jego najważniejszych właściwości jako dokumentu i jako pliku, istotne są cechy „użytkowe", warunkujące chęć użytkowników do stosowania konkretnego rozwiązania. Aby dokument elektroniczny był powszechnie i skutecznie stosowany powinien posiadać następujące cechy:

— możliwość stosowania niezależnie od stopnia zaawansowania technologii informatycznych wykorzystywanych przez użytkowników,
— niezależność formatu dokumentu od branży lub sektora gospodarki,
— neutralność technologiczną formatu dokumentu i oprogramowania,
— łatwość integracji z różnymi systemami użytkownika,
— autonomiczność,
— możliwość interpretacji dokumentu zarówno automatycznie, jak i przez człowieka (koncepcja takiego dokumentu jest nazywana „dokumentem semantycznym" i została przedstawiona w artykule [144]).

W dużej część powyższe cechy użytkowe są związane z wymogami dotyczącymi szeroko rozumianej interoperacyjności, która związana jest ze zdolnością do przetwarzania informacji zapisanej w dokumencie przez różne systemy informatyczne oraz łączenia usług realizowanych przez różne systemy informatyczne.

Dokumenty elektroniczne, które zgodnie z przedstawioną definicją, są plikami zawierającymi „uporządkowany zbiór danych", podobnie jak pliki posiadają określony format, czyli inaczej składnię albo strukturę. Tak więc możliwość ich przetwarzania jest uwarunkowana znajomoscią tego formatu. Istnieje wiele formatów plików i można je systematyzować według różnych kryteriów (binarne, tekstowe, wykonywalne, systemowe, ukryte, do odczytu itd.). Przykładowe formaty plików, które mogą być wykorzystane podczas realizacji trans-

akcji, zaprezentowano w podrozdziale 3.4. Jednak bez względu na format pliku składnie dokumentów zawierają informacje o:

1. położeniu danych w strukturze dokumentu – tzw. „warstwa danych”,
2. sposobie wyświetlania oraz znaczeniu danych – tzw. „warstwa prezentacji”.

Jeżeli plik (dokument) jest przetwarzany automatycznie, bez konieczności interakcji z człowiekiem, wówczas wystarczające jest zastosowanie pliku jednowarstwowego, zawierającego jedynie struktury danych. W takim pliku zarówno struktura, jak i znaczenie danych, są zapisane w jego specyfikacji, która stanowi podstawę do implementowania logiki interpretacji i przetwarzania pliku zgodnie z wymaganiami transakcji.

Jeżeli z kolei dokument, w trakcie jego przetwarzania ma być przeczytany i interpretowany przez człowieka, wówczas sposób prezentacji danych, czyli ich znaczenie, jest elementem drugiej warstwy, czyli warstwy prezentacji opisanej w specyfikacji formatu.

3.6. Formularz elektroniczny

W poprzednim podrozdziale zaprezentowano dokument elektroniczny jako nośnik bezpiecznej informacji w transakcji. Dokument taki nie jest przeznaczony do aktywnej interakcji z człowiekiem lub systemem. Człowiek lub logika systemu mogą jedynie odczytać i zinterpretować dane w dokumencie oraz zeweryfikować atrybuty bezpiczeństwa.

Dokumentem, dającym możliwość aktywnej interakcji z człowiekiem lub maszyną w celu wprowadzania kolejnych danych związanych z realizacją kolejnych kroków transakcji, jest formularz elektroniczny.

Zanim zostanie przedstawione pojęcie formularza elektronicznego warto przypomnieć, że formularz powstał już w czasach antycznych (patrz rozdział 1). W warstwie prezentacji odzwierciedlał on treść transakcji, która była ustalona („ustandaryzowana”) dla konkretnego typu transakcji i należało ją uzupełnić o dane (parametry) tej konkretnej transakcji (warstwa danych). Co do istoty pojęcie formularza, które ukształtowało się w czasach prehistorycznych, obowiązuje do dzisiaj i można je zdefiniować następująco (patrz def. 10)

> ***druk urzędowy z miejscami wolnymi, w które wpisuje się wymagane i właściwe dla danego typu pisma informacje. Czyli jest to wzór dokumentu z rubrykami do wypełnienia. Aby formularz mógł być dowodem - dokumentem (patrz def. 5), musi spełniać określone wymagania formalne i merytoryczne.***

Współczesne formularze, będące dokumentami elektronicznymi, dają ponadto możliwość realizacji transakcji wieloetapowej z udziałem różnych stron lub realizacją innych „cząstkowych” transakcji. W związku z tym współczesny formularz elektroniczny powinien odzwierciedlać możliwość partycypacji wielu stron w transakcji, a więc powinna istnieć możliwość „dopisywania” danych i uwierzytelniania ich przez różne strony transakcji w miarę wykonywania kolejnych kroków. Natomiast w zakresie bezpieczeństwa powinna istnieć możliwość umieszczenia w formularzu wielu podpisów, obejmujących różne części formularza przez różne strony transakcji. Sposób zapewnienia wskazanych wyżej oczekiwań w zakresie bezpieczeństwa dokumentów z wyrafinowanym wykorzystaniem podpisu elektronicznego jest przedmiotem rozważań przedstawionych w rozdziale 5 z jednoczenym zapewnieniem odpowiedniego poziomu interoperacyjności omówionej w rozdziale 4.

Jeżeli przyjmiemy, że formularz – zgodnie z jego definicją podaną wyżej – jest to druk urzędowy z miejscami przeznaczonymi na wprowadzanie danych, to dokument elektro-

niczny, który w swojej strukturze posiada miejsca do wprowadzania danych, jest również formularzem. Inaczej mówiąc formularz elektroniczny jest to dokument elektroniczny zapewniający interakcję z użytkownikiem lub systemem w warstwie danych. Interakcja ta polega na wpisywaniu, przez aplikację użytkownika lub aplikację serwerową, danych w miejsca dla nich przeznaczone. Można więc stwierdzić, że formularze elektroniczne mogą być przeznaczone do:

1. interakcji tylko z systemami automatycznymi i nie muszą mieć wówczas warstwy prezentacji, oraz
2. aktywnej interakcji z człowiekiem (wprowadzanie danych, uruchamianie procedur bezpieczeństwa) z użyciem aplikacji użytkownika.

W związku z powyższym koncepcja formularza elektronicznego powinna uwzględniać:

1. specyfikację składni formularza rozumianej jako opis tej składni z zapewnieniem możliwości łatwego dopisywania i odczytywania danych z dokumentu,
2. oprogramowanie użytkowników formularza, którego zadaniem jest odczyt, interpretacja i prezentacja formularza na podstawie pliku o składni zgodnej ze składnią formularza oraz wprowadzanie danych do formularza,
3. możliwość generowania i pełnego weryfikowania podpisów elektronicznych przez strony transakcji.

Ze względu na fakt, że realizacja współczesnych transakcji przebiega w rozproszonym i zróżnicowanym środowisku technologicznym, w którym korzysta się z rozmaitych protokołów wymiany informacji i usług o charakterze transakcyjnym, to formularze elektroniczne powinny uwzględnić konieczność integracji ze środowiskiem technicznym oraz usługami sieciowymi. Dzięki temu możliwe będzie implementowanie transakcji sterowanych logiką formularza z udziałem wielu stron (ludzi i usług automatycznych) poprzez zastosowanie logiki formularza stanowiącej trzecią warstwę dokumentu elektronicznego. W tej warstwie, uwzględnionej w składni formularza, definiowane są na przykład:
— dynamiczne aspekty prezentacji formularza stosownie do stanu realizowanej transakcji,
— reguły poprawności relacji pomiędzy wartościami pól formularza,
— mechanizmy tzw. podpowiedzi i pomocy dla użytkowników,
— definicje relacji pomiędzy polami z podpisami w formularzu.

Istotnym aspektem koncepcji formularza elektronicznego z warstwą logiki jest możliwość zaimplementowania w formularzu wielu podpisów obejmujących różne części formularza odzwierciedlających realizację transakcji. Podpisy takie składane przez strony (ludzie lub maszyny) biorące udział w transakcji umożliwiają zabezpieczenie informacji w obszarach zgodnych z aktualnym etapem realizowanego procesu.

Jednak podpisy elektroniczne w trójwarstwowej koncepcji formularza elektronicznego nie są ograniczone do podpisów stron biorących udział w realizowanej transakcji. W celu uniknięcia zagrożeń wynikających z podatności opisanych w rozdziale 5, niezbędne jest takie zabezpieczenie wszystkich warstw formularza – warstwy danych, prezentacji i logiki, które uniemożliwi złośliwą ingerencję w dane oraz sposób wyświetlania i „działania” formularza. Sposób zabezpieczenia poszczególnych rodzajów podpisów w formularzu jest przedstawiony szerzej w podrozdziale 5.3.

Taki zakres zabezpieczenia sprawia, że formularz staje się bezpieczny w szerokim sensie jako aktywny element realizacji transakcji, ponieważ dzięki trzeciej warstwie jest zintegrowany z usługami sieciowymi (np. w architekturze SOAP, REST), wykorzystuje istniejące środki transportu informacji (SMTP, POP3) i włącza ludzi jako aktywnych uczestników

transakcji. Z powyższego wynika, że trzecia warstwa formularza elektronicznego stanowi kluczowy element zapewnienia interoperacyjności na wysokim poziomie. Koncepcja takiego podejścia do implementacji formularza została zaprezentowana przez autora rozprawy na przykład w pracach [204, 205, 207].

3.7. Podsumowanie

W podrozdziale 3.1 zdefiniowano pojęcie tranakcji (patrz również def. 28 w słowniku) rozumianej jako formalnie określony sposób działania. Jej realizacja wymaga stosowania bezpiecznej i skutecznej prawnie wymiany trwałej informacji co opisano w podrozdziale 3.2. Z kolei odpowiedni poziom bezpieczeństwa informacji pozwala na realizację usług (transakcji) elektronicznych na oczekiwanym poziomie dojrzałości omówionym w podrozdziale 3.3. Informacja spełniająca określone cechy jest dokumentem, a jego szczególną formą jest dokument elektroniczny. Dokument ten, jak wspomniano w podrozdziałach 3.4 i 3.5, jest przede wszystkim nośnikiem bezpiecznej informacji. Ponadto służy on do gromadzenia, przetwarzania, przechowywania i udostępniania informacji z wykorzystaniem technik elektronicznych, co wymaga zapewnienia odpowiedniego poziomu interoperacyjności, o której mowa w rozdziale 4. W podrozdziale 3.6 opisano formularz elektroniczny, który posiada wszystkie cechy dokumentu, w tym również dokumentu elektronicznego. Formularz ten do realizacji transakcji włącza nie tylko ludzi, ale również automatyczne systemy informatyczne. Zadaniem tego formularza, jak każdego innego, jest bezpieczne gromadzenie i przechowywanie informacji o transakcjach. Posiada on trójwarstwową naturę, którą tworzą: warstwa prezentacji, warstwa danych oraz warstwa logiki. Warstwy te opisano w podrozdziale 3.6

Istotną cechą formularza elektronicznego, decydującą o jego przydatności we wspomaganiu realizacji transakcji z udziałem wielu stron – zarówno ludzi, jak i systemów informatycznych – jest system podpisów elektronicznych. Zadaniem podpisów jest zabezpieczanie formularza i transakcji w wielu aspektach – od zabezpieczania zasobów informacyjnych skojarzonych z formularzem, poprzez zabezpieczanie samej struktury formularza, po zabezpieczanie danych wprowadzanych do formularza przez strony transakcji. Podpisy w formularzu jako podpisy w formacie XML zostaną opisane w podrozdziale 5.2, a ich wykorzystanie w formularzu w podrozdziale 5.3.

Przedstawiona w tym podrozdziale koncepcja formularza elektronicznego nadaje nowy wymiar interoperacyjności systemów, procesów i organizacji biorących udział w transakcji. Problematyka interoperacyjność w różnych obszarach oraz związane z nią wymogi są bardziej szczegółowo przedstawione w rozdziale 4 pracy, natomiast zagadnienia związane z bezpieczeństwem w kontekście zastosownych podpisów elektronicznych, a także pewnych innych zagrożeń, opisano w rozdziale 5.

4. Interoperacyjność

Wśród wielu definicji interoperacyjności [45, 60, 198, 88, 90] istnieje zgodność co do jej istoty. W odniesieniu do realizacji transakcji interoperacyjność można zdefiniować jako:

> ***zdolność osób i usług zaangażowanych w transakcje do wymiany i wykorzystania informacji dla osiągnięcia określonego celu.***

Ponadto, według [45], interoperacyjność ***obejmuje również zdolność do transgranicznej wymiany danych*** (w ramach jednolitego rynku pomiędzy podmiotami z państw członkowskich UE).

Omówione w podrozdziale 3.4 sposoby implementacji usługi elektronicznej wspomagającej realizację transakcji (XML, XSD, SOAP, WSDL [139, 134, 141, 100, 121, 211]) pozwalają na budowanie rozwiązań charakteryzujących się dużymi możliwościami interoperacyjności (integracja wielu systemów) [178]. To właśnie odpowiedni poziom interoperacyjności jest warunkiem koniecznym do szerokiego wykorzystania proponowanego formularza elektronicznego, który umożliwia integrowanie – w ramach transakcji elektronicznych – różnych usług sieciowych. Tak więc formularz elektroniczny, oprócz gromadzenia danych i decyzji podejmowanych przez uczestników transakcji w sposób prawnie skuteczny, integruje wymagane usługi sieciowe, które z natury są luźno (lub wcale) powiązane. Ta cecha formularza (interoperacyjność) jest bardzo istotna, gdyż w obecnych czasach wykorzystanie usług sieciowych [10, 200] jest podstawą realizacji nowoczesnych transakcji.

4.1. Obszary interoperacyjności

Omawiając interoperacyjność transakcji z wykorzystaniem formularza elektronicznego nie można pominąć jej rodzajów. Wielu autorów jest zgodnych w tej kwestii [59, 45, 198] i wyróżniają oni następujące jej rodzaje:

— **techniczna:** obejmuje zagadnienia techniczne związane z komputerami, interfejsami odpowiedzialnymi za komunikację maszyna-maszyna,
— **składniowa:** związana jest z formatami danych używanymi do tworzenia wiadomości (HTML, XML, ASN.1),
— **semantyczna:** odnosi się do precyzyjnego znaczenia i rozumienia informacji wymienianych przez aplikacje wcześniej do tego nieprzygotowane,
— **organizacyjna:** dotyczy modelowania procesów biznesowych, dopasowywania architektury informacji do celów organizacyjnych oraz wspierania biznesu w zakresie współpracy.

W artykule [165], wspomnianą wyżej klasyfikację rodzajów interoperacyjności, autorzy nazywają „poziomami” (levels) i na ich podstawie określają wytyczne dotyczące interoperacyjności e-biznesu w następujących aspektach:

1. **Interoperacyjność danych** (także [32, 149, 171]) – jest to cecha danych, która umożliwia ich powszechną dostępność, wykorzystanie i zrozumienie przez wszystkie strony

transakcji. Stronami transakcji mogą być zarówno ludzie, jak i systemy komputerowe (maszyny). Termin interoperacyjność danych powinien odnosić się do semantycznej reprezentacji danych, standaryzacji danych, uzgadniania schematów, a także tłumaczenia i transferu danych.

2. **Interoperacyjność systemów oprogramowania** (także [125, 133]) – jest to zdolność systemów lub produktów organizacji do współpracy z systemami lub produktami innych organizacji. Obszar ten obejmuje m. in. interoperacyjność usług sieciowych w paradygmacie architektury zorientowanej na usługi (SOA). Podejście to traktuje funkcjonalność jako zbiór interoperacyjnych usług, które mogą być wykorzystywane przez wiele odrębnych systemów w wielu obszarach lub transakcjach. Usługi sieciowe mogą reprezentować zarówno nowe aplikacje, jak i „opakowywać" istniejące, stare systemy w celu udostępnienia ich w sieci. Innym rodzajem interoperacyjności systemów jest interoperacyjność w warstwie oprogramowania pośredniczącego, tzn. zdolność do łączenia komponentów oprogramowania lub osób z ich aplikacjami.
3. **Interoperacyjność tożsamości elektronicznej** (także [18, 170, 209])- - odnosi się do zdolności różnych systemów identyfikacji tożsamości w obrębie współpracujących organizacji lub pomiędzy nimi do współpracy w celu automatycznego uwierzytelniania i autoryzacji podmiotów oraz przypisywania im ról i uprawnień w zakresie bezpieczeństwa, niezależnie od tego, skąd pochodzą. Interoperacyjność ta obejmuje interoperacyjność podpisów elektronicznych, czyli wzajemne uznawanie podpisów elektronicznych pomiędzy krajami oraz różnorodność techniczną w rozumieniu atrybutów, walidacji, formatów i algorytmów [145], [163].
4. **Interoperacyjność usług** (także [101]) – to zdolność przedsiębiorstwa do dynamicznego korzystania ze złożonych usług pochodzących ze źródeł zewnętrznych, takich jak partner biznesowy (strona transakcji) lub dostawca usług internetowych. Ten rodzaj interoperacyjności obejmuje m. in. orkiestrację lub kompozycję procesu (wykonanie transakcji) z istniejących usług. Określa ona interakcje zewnętrzne i wewnętrzne. Interakcje zewnętrzne obejmują połączenia z innymi usługami i komunikaty przychodzące, natomiast interakcje wewnętrzne obejmują przetwarzanie danych i obliczenia. Ten typ interoperacyjności obejmuje również choreografię usług opisującą interakcję usług z perspektywy zewnętrznego obserwatora.
5. **Interoperacyjność obiektów** (także [197]) – związana jest z komunikacją sieciową i współpracą obiektów (np. RFID, interoperacyjność urządzeń, komunikacja obiektów i urządzeń inteligentnych).
6. **Interoperacyjność kulturowa** (także [42, 157]) – może dotyczyć m. in. grup międzynarodowych posługujących się różnymi językami, a w węższym znaczeniu jest to interoperacyjność językowa (lingwistyczna).
7. **Interoperacyjność procesów** – jest to zdolność do dopasowania procesów różnych podmiotów w celu wymiany danych.
8. **Interoperacyjność reguł** – jest to zdolność podmiotów do dopasowania swoich procesów biznesowych do reguł prawnych w celu przeprowadzania zgodnych z prawem transakcji, które są również zgodne z wewnętrznymi regułami biznesowymi. Obejmuje modelowanie reguł, ujednolicanie reguł biznesowych i prawnych oraz wdrażanie reguł.

Interoperacyjność procesów i reguł w warstwie technicznej jest zapewniona podczas projektowania obsługi transakcji w oparciu o ustalenia techniczne między stronami transakcji. Uzgodnienia techniczne są wynikiem uzgodnień organizacyjnych i prawnych, przy czym ustalenia te (organizacyjne) określają poziom i dojrzałość interoperacyjności na

poziomie organizacji, ale wykraczają poza rozważania techniczne będące przedmiotem niniejszych rozważań.

Przedstawione „wytyczne” stanowią wskazówki zarówno do zdefiniowania koncepcji wykorzystania formularza elektronicznego w realizacji transakcji, jak również do oceny tego rozwiązania z punktu widzenia interoperacyjności. Spełnienie tych wytycznych w odniesieniu do formularza elektronicznego stosowanego w różnych środowiskach zapewnia wysoki poziom interoperacyjności, tj. możliwości integracji z różnymi systemami oraz akceptacji użytkownika - człowieka, czyli strony uczestniczącej w transakcji.

4.2. Podsumowanie – Interoperacyjność

Podsumowując powyższe rozważania należy stwierdzić, że uwzględnienie wymogów dotyczących interoperacyjności dokumentów elektronicznych, a zwłaszcza formularzy elektronicznych, daje możliwość umiejscowienia formularza elektronicznego jako punktu centralnego realizacji transakcji, integrującego różne usługi sieciowe z różnych branż z realizowaną transakcją. Ponadto wykorzystanie takiego formularza umożliwia budowanie rozwiązań transgranicznych (wielojęzykowych), a koncepcja dokumentu trójwarstwowego umożliwia wykorzystanie w realizowanych transakcjach najnowszych i wschodzących technologii.

Warunkiem szerokiego i przede wszystkim bezpiecznego korzystania z rozwiązań – dokumentów wspomagających realizację transakcji w heterogenicznym środowisku jes konieczność zastosowania interoperacyjnych środków ochrony informacji. Ich format powinien umożliwić zabezpieczanie (podpisywanie) dokumentów w wieloetapowych transakcjach, w których bierze udział wiele stron – zarówno ludzie, jak i maszyny. Zagadnienia związane z podpisami elektronicznymi, spełniającymi wymienione postulaty, są przedsawione w rozdziale 5.

5. Bezpieczeństwo

Bezpieczeństwo transakcji – w rozważaniach zawartych w niniejszej rozprawie – jest tożsame z bezpieczeństwem informacji towarzyszącej lub będącej przedmiotem transakcji. Dlatego też nie będą w pracy rozważane inne aspekty bezpieczeństwa transakcji jak na przykład fizyczne bezpieczeństwo zasobów związanych z transakcją.

W związku z powyższym w pracy przyjęto, że bezpieczeństwo informacji oznacza bezpieczeństwo z punktu widzenia skuteczności prawnej [212, 74, 1]. Należy to rozumieć w ten sposób, że na pewnym etapie realizacji procesu biznesowego istnieje potrzeba zapewnienia bezpieczeństwa wymienianych informacji, przynajmniej na potrzeby postępowania sądowego, gdyż mogą wystąpić różnice w interpretacji zdarzeń i zobowiązań podjętych przez strony. Taką możliwość daje dokument, który niezależnie od swojej formy (implementacji technicznej), musi zapewniać osiągnięcie celów bezpieczeństwa wskazanych w [135] i w pracach [3, 204, 181]. W literaturze (np. w pracach [2, 104, 69]) zaproponowano wiele schematów kryptograficznych, które pozwalają zapewnić niezaprzeczalność i autentyczność informacji. Większość z nich wykorzystuje podpisy cyfrowe i/lub algorytmy funkcji skrótu w celu spełnienia tych właściwości.

Ze względu na różnorodność rodzajów transakcji – od „drobnych" operacji gospodarczych, w których cała wymiana informacji jest ograniczona do ustaleń i umów „ustnych" do transakcji skomplikowanych związanych z dobrami o znacznej wartości i wymagających spełnienia zaawansowanych uwarunkowań prawnych – nie jest możliwe ustalenie jednego spójnego zbioru wymagań dotyczących sposobów zapewnienia bezpieczeństwa informacji we wszystkich typach transakcji. Dlatego dla każdego typu transakcji należy zapewnić realizację takich celów bezpieczeństwa informacji jaka jest wymagana dla danego typu transakcji. Jest to szczególnie istotne dla informacji elektronicznych wspomagających realizację transakcji. W kontekście implementacji tych rozwiązań dostosowanie odpowiednich, skutecznych i bezpiecznych prawnie środków (np. podpis elektroniczny) powinno być przedmiotem analizy i projektu rozwiązania dostosowanego do konkretnego typu transakcji.

5.1. Podpis elektroniczny

Podpis jest jednym z najważniejszych narzędzi wykorzystywanych do zabezpieczania informacji związanej z transakcją [135]. Stosowanie podpisu jest kluczową metodą zapewnienia niezaprzeczalności, uwierzytelniania, identyfikacji, poświadczenia i innych cech świadczących o bezpieczeństwie informacji. Jest on środkiem zapewniającym informacji cechy dokumentu [14, 17, 109, 173, 116, 33, 41, 99, 105, 166, 31, 33, 166, 177].

Na przestrzeni lat powstawały kolejne formaty podpisu elektronicznego o różnych możliwościach (PKCS#7 [97], CMS [87] lub zaawansowany format CADES [56] i PAdES [58]) dedykowane dokumentom. W wymienionych formatach podpisów treść podpisu jest definiowana arbitralnie (sztywno określona przez specyfikacje formatu), co stanowi problem dla interoperacyjności rozumianej jako możliwość elastycznego dopasowania treści podpi-

su do różnych typów dokumentów wspierających realizację transakcji w heterogenicznym rozproszonym środowisku z udziałem wielu stron (np. podpis powinien obejmować różne części dokumentu). Rozwiązaniem tego problemu było pojawienie się formatu XML [26] jako środka implementacji różnych typów dokumentów w sposób interoperacyjny, z możliwością definiowania składni, co umożliwiło implementację różnych typów dokumentów wspierających transakcje. Rozszerzenie koncepcji XML o specyfikację podpisu elektronicznego XML [26] dodatkowo pozwoliło na umieszczanie w podpisie XML różnego rodzaju informacji rozszerzających sam podpis (jako odpowiednik CADES). Istotną różnicą pomiędzy standardowymi formatami podpisów (w których treść podpisu musi być zgodna ze specyfikacją standardu podpisu) a podpisem XMLdSig jest możliwość zamieszczenia w jednym podpisie wielu informacji (np. XAdES [47]), a w szczególności wielu różnych podpisywanych treści.

5.2. Podpis XML

5.2.1. Struktura podpisu XML

Zgodnie z definicją podaną na przykład w [31] podpis elektroniczny jest wynikiem operacji kryptografii asymetrycznej i funkcji skrótu wykonanych na elektronicznej reprezentacji podpisywanej treści (podpisywanym dokumencie).

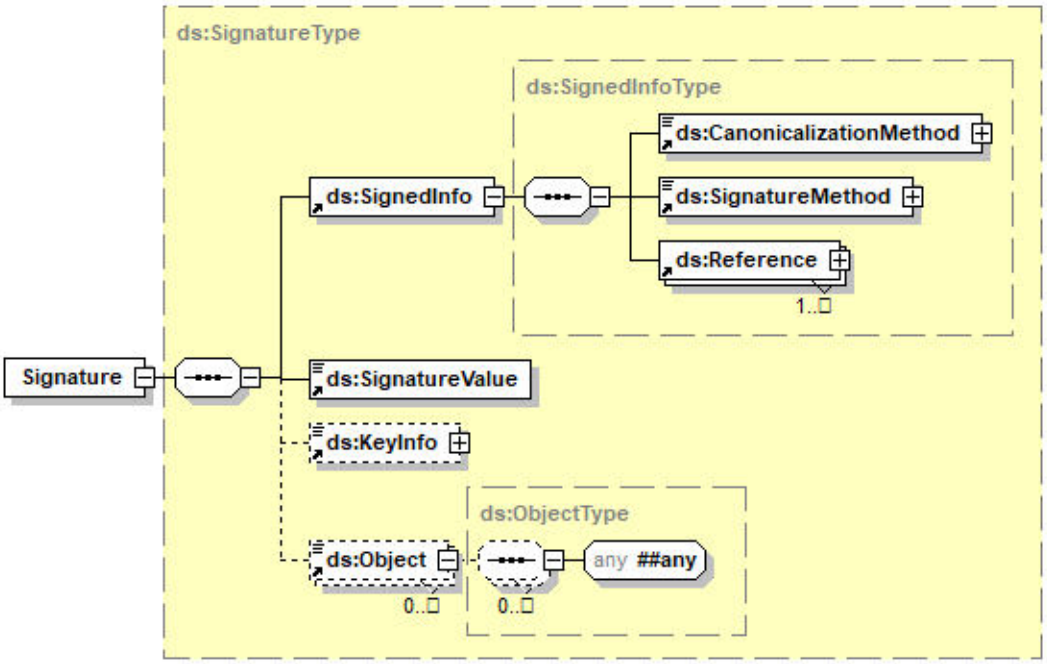
Podpis XML (o akronimie XMLdSig) jest jednym z wielu formatów podpisu cyfrowego, o których była mowa w podrozdziale 5.1. Jest on strukturą w formacie XML [26] o składni zdefiniowanej z użyciem schematu XML (XML Schema – XSD) [127], [189]. Struktura (składnia) oraz znaczenie poszczególnych elementów struktury podpisu XML są szczegółowo opisane w rekomendacji W3C [168]. Rekomendacja ta wprowadza stosowanie publicznie dostępnych algorytmów kryptografii asymetrycznej, funkcji skrótu, kanonikalizacji czy kodów uwierzytelniania wiadomości HMAC [108]. Ponadto definiuje również zestaw algorytmów do zaimplementowania oraz umożliwia stosowanie nowo publikowanych lub definiowanych przez użytkownika algorytmów [33]. Cechą wyróżniającą podpis XML według [176] jest jego elastyczność oraz fakt, że może być osadzony w dokumencie XML. Charakterystyka tych cech jest szczegółowo opisana w literaturze na przykład w [99, 14, 17, 173].

Zgodnie ze specyfikacją W3C [168] podpis XML składa się z czterech części strukturalnie stanowiących elementy XML:

1. `<SignedInfo>` – element, w którym określona jest metoda konstruowania podpisu. W tym elemencie podpisujący wskazuje:
   a) w elemencie `<CanonicalisationMethod>` – algorytm kanonikalizacji zawartości podpisu,
   b) w elemencie `<SignatureMethod>` – sposób obliczania wartości podpisu, na który składają się algorytm obliczania wartości funkcji skrótu oraz algorytm szyfrowania wartości skrótu,
   c) sekwencję elementów `<Reference>` – która definiuje zwartość podpisu. Zawartość ta otrzymywana jest w wyniku wykonywania transformacji opisanych w `<Reference>` na treściach wskazywanych w `<Reference>`. W tych elementach wskazywane są również algorytmy funkcji skrótów oraz wynik obliczenia wartości funkcji skrótów na wyżej omówionych treściach.

2. `<SignatureValue>` – element, w którym zapisana jest wartość zaszyfrowanego skrótu – jest to wartość podpisu. Wartość ta obliczana jest z wykorzystaniem algorytmów (funkcji skrótu i szyfrowania) wskazanych w omówionym wyżej elemencie `<SignedInfo>`.
3. `<KeyInfo>` – element, który ma zawierać informacje o danych niezbędnych do pobrania klucza publicznego lub innych informacji niezbędnych do weryfikacji podpisu i tożsamości osoby podpisującej, takich jak: klucz publiczny, certyfikat klucza publicznego lub inne informacje pozwalające na uzyskanie tego klucza.
4. `<Object>` – element opcjonalny, w którym może być umieszczona dowolna treść zgodnie ze specyficznymi wymaganiami stron transakcji na przykład zgodnie z XADES [47].

Przedstawiona struktura podpisu jest szeroko omawiana i interpretowana w literaturze na przykład w [176, 177, 166]. Graficzna reprezentacja struktury podpisu XML zgodnie ze specyfikacją (Rekomendacją W3C) XMLdSig jest przedstawiona na rysunku 5.1.

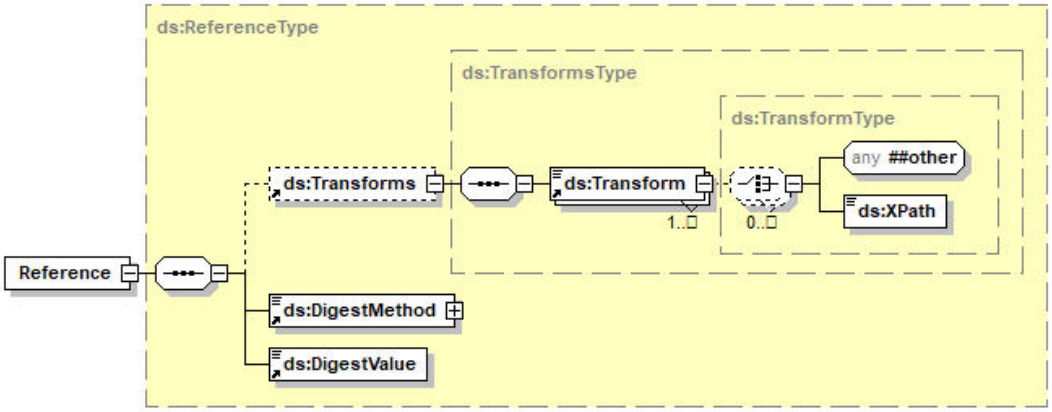


Rys. 5.1: Struktura podpisu XMLdSig (na podstawie [168], z oprogramowaniem XMLSpy)

Struktura elastycznej definicji zawartości podpisu jest zapisywana/przechowywana w elemencie `<Reference>` elementu `<SignedInfo>` (punkt 1 powyżej) i składa się z:

1. atrybutu `@URI`, który specyfikuje wskazującego podpisywany zasób,
2. elementów `<Transform>` zawierających opisy transformacji kolejnych zawartości podpisu,
3. elementu `<DigestMethod>` określającego algorytm obliczania wartości funkcji skrótu z przetransformowanej zawartości,
4. elementu `<DigestValue>` zawierającego obliczoną wartość funkcji skrótu z transformowanego zasobu.

Na rysunku 5.2 zaprezentowano strukturę zawartości elementu `<Reference>` podpisu XML.



Rys. 5.2: Struktura zawartości elementu `<Reference>` podpisu XMLdSig (na podstawie [168], z oprogramowaniem XMLSpy)

5.2.2. Cechy podpisu XML

Podstawową zaletą podpisu XML jest możliwość elastycznego definiowania treści podpisu (w elementach `<Reference>` elementu `<SignedInfo>`). Pozwala ona na tworzenie następujących podstawowych typów podpisu w zależności od rodzaju relacji pomiędzy podpisem i podpisywaną zawartością:

- **podpis otoczony** (*enveloped*), który znajduje się wewnątrz struktury XML dokumentu,
- **podpis otaczający** (*enveloping*) zawierający podpisaną treść w elemencie `<Object>`,
- **podpis odłączony** (*detached*), który podpisuje dane poza dokumentem XML zawierającym podpis (zwykle stosowany w przypadku podpisów plików, które nie są dokumentami XML).

Na rysunku 5.3 przedstawiono omówione relacje pomiędzy podpisami i dokumentami dla poszczególnych typów podpisów.

Ponadto ważnymi cechami podpisu XML według [99, 109] są:

1. możliwość podpisywania fragmentów dokumentów (tzw. podpis częściowy) z wykorzystaniem mechanizmów XPath [180], oraz
2. korzystanie z podpisów wielokrotnych (możliwość opatrywania dokumentu wieloma podpisami) z wykorzystaniem `@URI` (Uniform Resource Identifier) .

Przykładem wykorzystania przedstawionych cech podpisu XML może być karta historii choroby pacjenta prowadzona przez różnych lekarzy odpowiedzialnych za poszczególne zapisy (rozpoznania, zalecenia itp.) [14]. W tym przypadku każdy nowo dodany zapis musi być podpisany indywidualnie przez konkretnego lekarza.

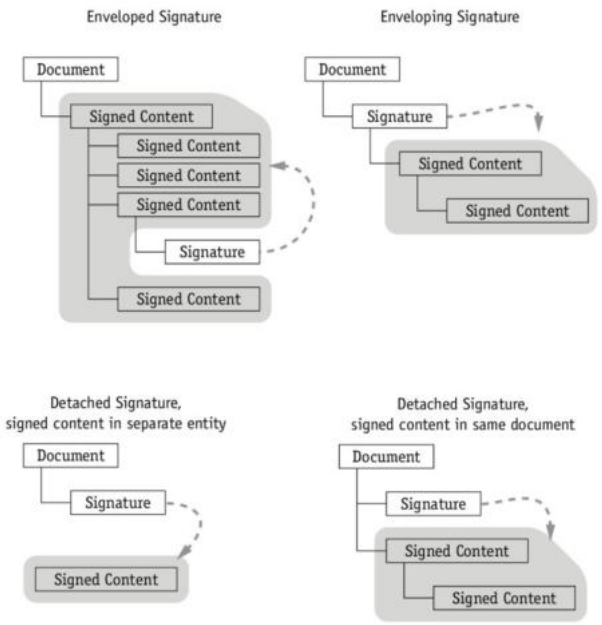
Autor rozprawy zgadza się z opiniami autorów przedstawionymi w [177], że najważniejsze cechy podpisu XML to:

— **możliwość osadzenia podpisu w dokumencie XML,** oraz
— **elastyczność podpisu (możliwość dowolnego modelowania jego zawartości).**

Szczegółowe opisy tych dwóch cech znajdują się w literaturze [14, 99, 166, 173].

5.2.3. Przetwarzanie podpisu XML

Sposób przetwarzania podpisu XML wynika z jego specyfikacji [168]. Jednak różni autorzy [41, 17, 111, 122, 153, 161, 166, 177, 177] w odmienny sposób podkreślają czynności,



Rys. 5.3: Podpisy: odłączony, otoczony i otaczający (źródło: domena publiczna)

które mogą być wykonane w różny sposób. Pełny, a zarazem zwięzły sposób przetwarzania podpisów XML, który nie stoi w sprzeczności z rozwiązaniami zaprezentowanymi w wymienionych powyżej pozycjach literatury, przedstawiono w [116]. W tej pracy przetwarzanie podpisów XML jest opisane jako procesy (tworzenie i weryfikacja) składające z następujących czynności:

Tworzenie (generowanie) podpisu (rysunek 5.4a)

1. tworzenie referencji
   — lokalizowanie i pobieranie zasobów za pomocą `@URI`,
   — wykonywanie transformacji opisanej w elemencie `<Transform>` na referowanych obiektach danych,
   — obliczenie wartości funkcji skrótu (przedstawionego w elemencie `<DigestMethod>`) i zapisywanie jej w elemencie `<DigestValue>`;
2. tworzenie podpisu
   — wykonanie algorytmu kanonikalizacji elementu `<SignedInfo>` (zawiera element `<Reference>`),
   — wykonanie algorytmu skrótu na elemencie (kanonikalizowanym) `<SignedInfo>` w celu obliczenia wartości skrótu,
   — zaszyfrowanie wartości skrótu za pomocą klucza prywatnego – uzyskana wartość podpisu jest zapisywana w elemencie `<SignatureValue>`.

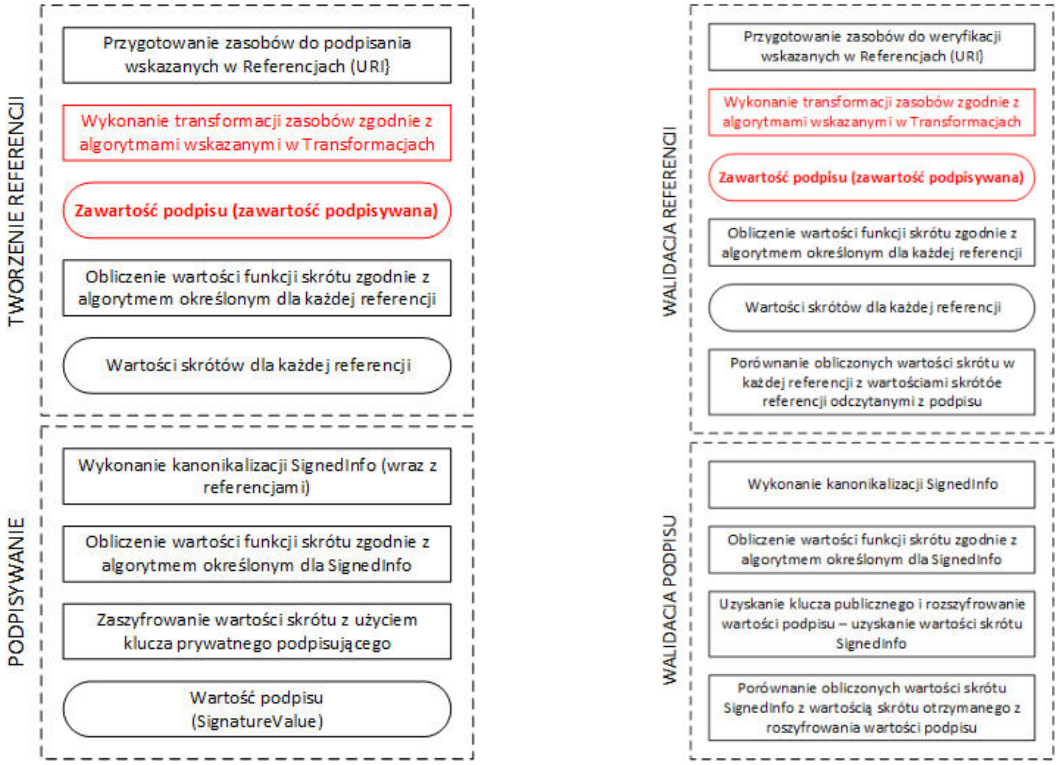
Weryfikacja podpisu (rysunek 5.4b)

1. weryfikacja referencji
   — wykonanie transformacji dokumentu zgodnie z definicją zapisaną w elemencie `<Transform>`,
   — wykonanie obliczeń wartości funkcji skrótu, aby otrzymać wartość funkcji skrótu dokumentu przetransformowanego,
   — porównanie obliczonej wartości funkcji skrótu z wartościami odczytami z elementów `<Reference>`;

2. weryfikacja podpisu
   — odczytanie informacji o kluczu publicznym z elementu `<KeyInfo>` podpisu,
   — obliczenie wartości funkcji skrótu w elemencie `<SignedInfo>` z wykorzystaniem algorytmu wskazanego w podpisie,
   — rozszyfrowanie wartości podpisu `<SignatureValue>` z wykorzystaniem klucza publicznego podmiotu podpisującego oraz algorytmu wskazanego w elemencie `<SignatureMethod>`,
   — porównanie obliczonej wartości podpisu z wartością odczytaną z elementu `<SignatureValue>` podpisu.

Należy zauważyć, że w propozycjach weryfikacji podpisu (w zakresie integralności) autorzy [41, 17, 111, 166, 177, 116] proponują, aby:

1. obliczyć (uzyskać) zawartość każdej referencji zgodnie z jej specyfikacją i obliczyć skrót tych zawartości,
2. obliczyć wartość funkcji skrótu dla elementu `<SignedInfo>`, który zawiera wartości funkcji skrótu dla wszystkich referencji,
3. odczytać wartość podpisu `<SignatureValue>` i odszyfrować ją przy użyciu klucza publicznego podpisującego (uzyskanego (odczytanego) z elementu `<KeyInfo>`),
4. porównać obliczone (punkt 2) i odszyfrowane (punkt 3) wartości funkcji skrótów (kolor czerwony na rysunku 5.4), przy czym wynik porównania *równy* lub *różny* jest równoznaczny z wynikiem weryfikacji integralności – odpowiednio *poprawny* lub *niepoprawny*.



(a) Tworzenie podpisu (b) Weryfikacja podpisu

Rys. 5.4: Schematy przetwarzania podpisu XMLdSig (źródło: [116])

5.3. Podpisy XML w formularzu elektronicznym

Jak napisano w podrozdziale 3.6, jedną z kluczowych cech formularza elektronicznego, decydującą o jego przydatności we wspomaganiu realizacji transakcji, jest pełne zabezpieczenie jego integralności jako dokumentu. Konieczne jest przy tym wskazanie niezaprzeczalnej odpowiedzialności w różnych aspektach jego funkcjonowania. Dlatego w strukturze formularza przewidziano:

podpisy zasobów zewnętrznych – są to opcjonalne podpisy, których zawartość stanowią dowolne opublikowane zasoby zewnętrzne wskazane przez ich `@URI`, które są w relacji logicznej lub prawnej z formularzem (np. instrukcje, opisy, regulacje prawne związane z wspomaganą transakcją);

podpisy autoryzujące formularz – są to opcjonalne podpisy, których zawartość obejmuje:
- — całą strukturę formularza, czyli wszystkie jego warstwy (prezentacja i logika) z wyłączeniem warstwy danych, oraz
- — kontrasygnaty podpisów zasobów zewnętrznych.

Podpisy te wskazują odpowiedzialność właścicieli/gestorów formularza, czyli organizacji odpowiedzialnych za jego poprawne stosowanie w kontekście realizacji transakcji, za którą odpowiadają. Mogą one również wskazywać na projektantów i twórców formularza odpowiedzialnych za poprawną jego implementację w aspekcie techniczym. Zawartość tego podpisu, obejmująca strukturę formularza (bez warstwy danych) razem z podpisami zasobów zewnętrznych oraz podpisami autoryzującymi, stanowi tzw. „wzór formularza”;

podpis zabezpieczający formularz – jest to jeden, obowiązkowy podpis, którego zawartością jest „wzór formularza”. Pełni on dwie funkcje polegające na zabezpieczaniu formularza przed jakąkolwiek modyfikacją jego wzoru oraz zabezpieczaniu praw autorskich właściciela rozwiązania poprzez możliwość łatwej weryfikacji tego „podpisu zabezpieczającego”, który dzięki temu można również nazwać „podpisem licencyjnym”;

podpisy użytkowników (stron transakcji) – są to podpisy stron biorących udział w transakcji. Mogą być nimi zarówno ludzie, jak i maszyny. Zawartość podpisu obejmuje „wzór formularza”, kontrasygnatę podpisu zabezpieczającego oraz te fragmenty dokumentu z warstwy danych, które są przyporządkowane do użytkownika (strony transakcji) biorącego udział w transakcji zgodnie z zaprojektowanymi i zaimplementowanymi w warstwie logiki wymogami tej transakcji.

Możliwości selektywnego podpisywania wskazanych fragmentów struktury XML, które odzwierciedlają zawartości omówionych wyżej podpisów, stwarza jedynie podpis w formacie XMLdSig opisany w podrozdziale 5.2 dzięki jego szczególnym cechom opisanym w podrozdziale 5.2.2.

5.4. Podatności XML

Stosowanie podpisu w formacie XML, oprócz znaczących korzyści wynikających z jego wykorzystania (opisane w podrozdziale 5.2.2), może stwarzać zagrożenia będące efektem nieostrożnego, niekompetentnego lub nieuczciwego korzystania z tego podpisu. Należy zdawać sobie sprawę, że istnieją zagrożenia, które nie są bezpośrednio związane z kryptografią, ale wynikają ze specyficznej natury dokumentów XML i podpisów XMLdSig. W pracach [132, 86] przedstawiono kilka ataków na podpisy XML i szyfrowanie, na przykład „odmowa usługi” przez *Entity Expansion* – parsowanie DTD (Data Type Definition) i

przetwarzanie elementów referencji prowadzi do wielopoziomowego rozszerzania (wielopoziomowej ekspansji) dokumentu oraz obciążenia pamięci i procesora.

Inną grupą ataków są ataki polegające na wstrzykiwaniu kodu do samego dokumentu (np. wiadomości SOAP [134, 141, 100]) lub elementów podpisu, i dotyczą one przetwarzania poprawnie zbudowanych i weryfikowanych podpisów. W dokumencie pojawia się kod (treść), który nie jest treścią podpisu (nie jest objęty podpisem), ale jest przetwarzany i może prowadzić do błędnej interpretacji treści dokumentu w przeświadczeniu o poprawności podpisu.

Również inni autorzy opisują różnorodne ataki na format dokumentów i podpisów XML. Na przykład w [111] opisano atak na podpis w komunikacie (wiadomości) SOAP związany z przetwarzaniem tego typu wiadomości. Problemem ataków na podpis w strukturze SOAP zajmują się także [79, 8]. W pracy [8] przedstawiono przegląd ataków typu CIA (Code Injection Attacks) które są oparte na tzw. atakach „*cross-site scripting*" [1], oraz metodę i algorytm przeciwdziałania tym atakom o nazwie „Gathering Multiple Signatures Approach" (GMSA). Metoda ta oparta jest na przetwarzaniu danych o złośliwych i poprawnych adresach URL zgromadzonych w bazach danych, a przetwarzanie danych wykorzystuje algorytmy sztucznej inteligencji.

Autorzy [132] pokazują, że podpisany obiekt wskazany przez `@URI` może zostać przeniesiony z zachowaniem poprawnego wyniku weryfikacji integralności podpisu i umożliwieniem wstawienia w to miejsce złośliwych danych. Gdy położenie danych jest istotne dla interpretacji ich znaczenia, to może ono być wykorzystane przez atakującego do uzyskania nieautoryzowanego dostępu do chronionych zasobów. W ten sposób inne dane są weryfikowane, a inne przetwarzane. Atak taki można zdefiniować jako wstrzyknięcie nieautoryzowanych danych do podpisanego dokumentu XML, z zachowaniem integralności i autentyczności podpisu. W szczególności ten rodzaj ataku opisano w odniesieniu do podpisu w strukturze wiadomości SOAP [139, 134, 141, 100], gdzie do przeprowadzenia ataku można wykorzystać wskazywanie referencji poprzez `@URI` w połączeniu z przetwarzaniem specyficznym dla SOAP. Ataki opisane w [132] występują w szczególnych przypadkach związanych z referencjami w podpisie, gdzie istnienie elementu w określonym miejscu w dokumencie jest opcjonalne, a nieoczekiwana obecność elementu w innym miejscu jest tolerowana. Jako środek zaradczy proponuje się stosowanie wyrażeń XPath w referencji, gdy zwykłe (z użyciem atrybutu) wskazanie w odnośniku jest nieodpowiednie. Oprócz środków zaradczych wykorzystujących XPath, ataki tego typu można wykrywać za pomocą polityki bezpieczeństwa, która zapobiega używaniu podpisanych i niezrozumiałych elementów (cecha protokołu SOAP). Niestety ten sposób ogranicza pełne wykorzystanie możliwości przetwarzania oferowane przez protokół SOAP.

Podobne zagrożenie zaobserwowano również w pracy [94]. Jako środek zaradczy autorzy proponują ograniczenie możliwości wystąpienia podpisywanego elementu do jednoznacznie określonego miejsca w dokumencie, co może prowadzić do zmniejszenia podatności dokumentu na ataki, a tym samym do zwiększenia jego bezpieczeństwa. Jednym ze sposobów realizacji takiej koncepcji może być wykorzystanie definicji XML Schema [127, 189]. Jednakże w pracy[94] podkreślono, że sposób ten nie gwarantuje bezpieczeństwa, ale jest ważnym elementem zabezpieczenia struktur XML odpornych na atak XSW (Signature Wrapping Attack).

[1] CSS - Cross Site Scripting: Ataki Cross-Site Scripting (XSS) są rodzajem iniekcji, w których złośliwe skrypty są wstrzykiwane do zaufanych stron internetowych lub dokumentów. Ataki XSS mają miejsce, gdy atakujący wykorzystuje aplikację do wysłania złośliwego kodu, zazwyczaj w formie skryptu po stronie przeglądarki lub aplikacji przetwarzającej, do innego użytkownika końcowego

Podobnie w pracy [126] zauważono, że atak XML Signature Wrapping (XSW) jest jednym z najpoważniejszych zagrożeń. Stwierdzono jednak, że zaproponowane do tej pory środki zaradcze, zwykle zapewniają ochronę tylko w określonych przypadkach i wciąż brakuje ogólnego, kompleksowego podejścia. Jako środek zaradczy autorzy tej pracy [126] proponują holistyczne i zintegrowane podejście polegające na wykorzystaniu dedykowanej architektury oraz dedykowanej biblioteki podpisów XML, które pozwalają na bezpieczną i skuteczną ochronę danych XML zgodnie ze standardami. Proponowane rozwiązanie jest dedykowane dla komunikatów XML w strukturach komunikatów SOAP i wykorzystuje cechy tego protokołu.

Również w pracy [107], analogicznie jak u innych autorów, zwrócono uwagę na podobny słaby punkt w stosowaniu podpisów XML do zabezpieczania wiadomości SOAP. Jako środek zaradczy autorzy proponują, aby nie używać atrybutu `@Id` lub wyrażenia XPath do wskazywania podpisanego elementu (w ramach wiadomości SOAP), ale zamiast tego podpisywać całą wiadomość SOAP.

Z kolei w pracy [79] zauważono, że w usługach sieciowych, pomimo stosowania standardów związanych z WS-Security[2], pojawiają się pewne ataki na komunikaty SOAP, które prowadzą do istotnych błędów w zabezpieczeniach. Jako przykład takiego ataku autorzy przytaczają pracę [132], podkreślając, że atak taki może nastąpić, ponieważ podpis XML związany z obiektem w dokumencie XML nie zależy od lokalizacji obiektu w dokumencie. Ponadto model rozszerzeń SOAP ma niewielkie ograniczenia dotyczące obecności elementu `<Headers>` oraz elementów wewnątrz komunikatu SOAP. Autorzy proponują wykrywanie ataku XSW poprzez zliczanie liczby wystąpień każdego węzła w komunikacie SOAP oraz przedstawiają algorytm zliczania węzłów w komunikatach SOAP i porównywania liczby węzłów wynikającej ze specyfikacji *referencji* w podpisie.

Innym rodzajem zagrożenia, które zostało omówione w pracy [162], jest zagrożenie związane z możliwością przeniesienia oryginalnej wiadomości spoza elementu `<Body>` i wstrzyknięcia złośliwej treści w miejsce oryginalnej wiadomości. Jako zabezpieczenie przed tym zagrożeniem autorzy proponują nowy proces weryfikacji podpisu wykorzystujący długość elementów `<Headers>` i `<Body>`, element `<Timestamp>` oraz odległość pomiędzy atrybutami `@id` w elementach `<Headers>` i `<Body>`.

W pracy [9], poświęconej zagrożeniom bezpieczeństwa w chmurze obliczeniowej, autorzy przedstawiają, oprócz zagadnień bezpieczeństwa przeglądarek i blokady dostawców, problem ataku XSW. Wymienione zostały w tej pracy cztery kategorie takich ataków: proste ataki kontekstowe przodków, opcjonalne ataki kontekstowe elementów, ataki na kontekst rodzeństwa oraz starszy kontekst rodzeństwa. W zakresie tych ataków autorzy proponują wykorzystanie koncepcji WS-Security i szyfrowania *end-to-end* dla komunikatów SOAP. W ten sposób atakujący nie będzie mógł pobrać i odczytać zaszyfrowanej wiadomości w celu jej modyfikacji.

Natomiast w pracy [111] autorzy zauważają, że w kontekście dokumentów XML zastosowanie SSL nie rozwiązuje w pełni kwestii integralności oraz niezaprzeczalności pochodzenia danych i nawet podpisany komunikat SOAP jest podatny na przechwycenie oraz późniejszą manipulację jego treścią. Według [132] wiadomość SOAP chroniona podpisem cyfrowym XML może zostać sfałszowana bez wpływu na poprawność podpisu. Autorzy proponują metodę wykrywania ataku w podpisanym żądaniu (usłudze sieciowej) z wykorzystaniem dodatkowego „tokena pozycji" jako dodatku do podpisanych elementów XML. Wówczas atak może zostać wykryty podczas weryfikacji podpisu poprzez porównanie obliczonego „hasha żądania" z „hashem treści" podpisanej w żądaniu.

[2] WS-Security są to rozszerzenia protokołu SOAP publikowane przez OASIS.

Z analizy zaprezentowanej powyżej literatury wynika, że jednym z podstawowych standardów budowania usług sieciowych jest protokół SOAP, oparty na formacie XML i zabezpieczony elektronicznym podpisem XML. Zastosowanie podpisu w ramach protokołu SOAP wykorzystuje wskazanie podpisywanej treści przez referencję, która z kolei wskazuje na wartość atrybutu *id* podpisywanego elementu. Takie podejście rodzi zagrożenie polegające na możliwości „przeniesienia" podpisanego elementu w inne miejsce (w ramach dokumentu XML) i wstrzyknięcia w jego miejsce sfałszowanej treści. Prowadzi to do weryfikacji podpisu z wykorzystaniem innej (prawdziwej) treści niż przetwarzanie dokumentu w ramach transakcji (sfałszowana treść).

Autorzy omawianych w tym podrozdziale prac proponują różne metody zapobiegania atakowi XSW w odniesieniu do wiadomości SOAP, ale problem ten może dotyczyć również innych wiadomości niż SOAP, a więc w zasadzie każdego dokumentu XML.

5.5. Podatność *Referencji* w podpisie XML

Przetwarzanie (tworzenie i weryfikacja) podpisu XML różni się od przedstawionych w podrozdziale 5.1 „tradycyjnych" podpisów elektronicznych, w których podpisywana treść nie jest wskazana w podpisie i wynika bezpośrednio lub pośrednio ze specyfikacji standardów lub wymagań dla tego typu podpisów.

Dla podpisu XML autorzy prac [176, 166, 41, 116, 17, 111] w obszarze weryfikacji integralności przedstawiają procedurę zgodną z omówionym w podrozdziale 5.2.3 referencyjnym schematem walidacji. Należy zwrócić uwagę, że przedstawiona w tych pracach metoda weryfikacji:

— po pierwsze ogranicza się do formalnej weryfikacji integralności treści poszczególnych referencji uzyskanych w wyniku wykonania transformacji referencji (specyfikacji referencji – element `<Reference>`),
— po drugie weryfikacja podpisu nie uwzględnia poprawności specyfikacji (zapisanej w `<Reference>`), tzn. nie sprawdza czy specyfikacja nie definiuje treści, która nie jest uzgodniona i nieadekwatna do wymogów realizowanej transakcji.

W opinii autora, przedstawionej w pracy [206], podejście do weryfikacji podpisu w zaprezentowany powyżej sposób, bez weryfikacji poprawności definicji zawartości podpisu (punkt drugi) stwarza zagrożenie. Powstaje ono w momencie składania podpisu (celowe działanie podpisującego). Atakujący (podpisujący) może wskazać w referencji dowolną treść, która jest nieistotna z punktu widzenia realizowanej transakcji. Prowadzi to do możliwości dowolnej modyfikacji treści istotnych z punktu widzenia realizowanej transakcji bez wpływu na ważność podpisu podczas jego weryfikacji. Automatyczne przetwarzanie dokumentów prowadzi do przetwarzania nieprawidłowej treści pomimo pozytywnego wyniku weryfikacji podpisu, co w konsekwencji prowadzi do nieoczekiwanego przebiegu realizacji transakcji. Problem ten został opisany przez autora w pracy [206] oraz rozwinięty w odniesieniu do formularzy elektronicznych w pracy [208]. W pracach tych zaproponowano, żeby nie kwestionując konieczności wykrywania ataków typu XSW (opisanych w podrozdziale 5.4), zabezpieczać dokument przed innymi nadużyciami polegającymi na podawaniu w podpisie XML niebezpiecznych referencji. Te niebezpieczne referencje powodują skutki podobne do XSW, czyli możliwość modyfikacji rzekomo „podpisanej" treści. Takie zabezpieczenie powinno bazować na weryfikacji poprawności specyfikacji referencji w podpisie. Oznacza to konieczność sprawdzania, czy referencje w podpisie wskazują na te elementy dokumentu XML (nie tylko SOAP), które są zgodne z ustaleniami stron przy projektowaniu sposobu i formatów wymiany danych dla konkretnej transakcji.

Reasumując należy stwierdzić, że pełna weryfikacja podpisu XML powinna obejmować:
— „standardową” kryptograficzną weryfikację integralności dokumentu z uwzględnieniem weryfikacji wartości funkcji skrótów zapisanych w elementach `<Reference>`, oraz
— weryfikację poprawności definicji zawartości podpisywanej treści zapisanej w elemencie `<Reference>`.

W tym celu autor zaporoponował, aby w procesie tworzenia i weryfikowania podpisów korzystać z tzw *szablonu podpisu*, czyli dokumentu, który zawiera sposób:
— definiowania treści podpisu (*referencji*) poza strukturą podpisu, czyli tworzenie tzw. szablonu podpisu, przy czym szablon ten powinien być wiarygodnym dokumentem tworzonym w trakcie realizacji automatycznej obsługi transakcji,
— automatycznego budowania podpisu na podstawie szablonu,
— weryfikacji podpisu na podstawie referencji w podpisie oraz referencji zbudowanych w oparciu o szablon – wystąpienie różnych wartości skrótów oznacza naruszenie podpisu.

5.6. Podsumowanie

Podpis XML w stosunku do istniejących wcześniej formatów podpisów elektronicznych, o których wspomniano w tym rozdziale, wykorzystujących zaawansowane techniki kryptograficzne, wniósł przede wszystkim:

elastyczność rozumianą jako łatwość dostosowania podpisu do wymogów różnego rodzaju transakcji, przede wszystkim poprzez możliwość modelowania jego zawartości, czyli włączenie do zawartości podpisu (objęcie podpisem) różnych rozdzielnych fragmentów dokumentu, a także dokumentów zewnętrznych. Ta cecha stanowi podstawę modelowania bezpieczeństwa formularzy elektronicznych poprzez definiowanie systemu podpisów, który opisano w podrozdziale 3.6, wykorzystując możliwości podpisu XML omówione w podrozdziale 5.2;

interoperacyjność, czyli możliwości łatwej implementacji logiki tworzenia i weryfikowania podpisów elektronicznych przez różne strony transakcji, w różnych środowiskach. Interoperacyjność ta może być rozważana w różnych „rodzajach interopercyjności” (np. techniczna, składniowa, semantyczna, organizacyjna) oraz w różnych aspektach takich jak interoperacyjność danych, systemów oprogramowania, tożsamości elektronicznej, co zostało opisane w podrozdziale 4.1.

Jednak należy podkreślić, że nieuważne lub nieumiejętne stosowanie w dokumentach XML podpisów w formacie XML może prowadzić do niepożądanych konsekwencji. Cechy podpisu XML z jednej strony dają duże korzyści w zakresie bezpieczeństwa i interoperacyjności, ale z drugiej strony są źródłem zagrożeń opisanych w podrozdziale 5.4. Omówione tam podatności są związane z protokołem SOAP, natomiast w podrozdziale 5.5 autor rozprawy wskazał na bardziej uniwersalną podatność, związaną nie tylko z protokołem SOAP, ale ogólnie z podpisem XML w dokumencie XML i dotyczącą definiowania zawartości podpisu w tzw. referencji. Jednocześnie zaproponowano przeciwdziałanie tej podatności polegające na wykorzystaniu tzw. *szablonu podpisu* w postaci bezpiecznego dokumentu zawierającego opis sposobu tworzenia i weryfikowania podpisu XML. W ten sposób, czyli wykorzystując *szblon podpisu* oraz system podpisów w formularzu (podrozdział 3.6), można osiągnąć bardzo wysoki poziom bezpieczeństwa podpisów w formularzach elektronicznych z zachowaniem ich interoperacyjności.

Na zakończenie należy stwierdzić, że osiągnięcie wyższych poziomów dojrzałości usług elektronicznych, które zostały opisane w podrozdziałach 3.3 i 3.4, wymaga stosowania zaawansowanych i bardziej lub mniej zcentralizowanych systemów. Dlatego też w kolejnym

6 rozdziale zaprezentowano koncepcję łańuchów bloków, które powinny zapewnić rozwiązanie tego problemu, czyli budowanie bezpiecznych rozwiązań wspomagających realizację transakcji w całkowicie rozproszonym środowisku.

6. Łańcuchy bloków

Łańcuchy bloków stały się bardzo popularne w ostatnich latach. W związku z tym pojawiły się liczne publikacje na ich temat. Większość autorów tych publikacji ([158, 119, 12, 48, 216, 174, 68, 147, 80, 223, 124, 12, 174]) podaje, że koncepcja łańcuchów bloków została stworzona i opublikowana po raz pierwszy przez Satoshi Nakamoto, którego tożsamość nie jest znana, w artykule „***Bitcoin: A Peer-to-Peer Electronic Cash System***" [143]. Artykuł ten jest artykułem przełomowym w omawianym obszarze, chociaż nigdy nie został przesłany do tradycyjnego recenzowanego czasopisma. W rzeczywistości Nakamoto nie jest autorem koncepcji łańcuchów bloków. Jak podaje [182] twórcami koncepcji uwierzytelniania dokumentów cyfrowych, opartej na pomyśle łańcuchów bloków i dającej pewność, że wersja dokumentu nie została zmieniona w stosunku do oryginału, byli naukowcy **Stuart Haber** i **Scott Stornetta** z Real Bellcore Labs (rys. 6.1).

Rys. 6.1: Stuart Haber i Scott Stornetta – twórcy koncepcji łańcucha bloków (źródło: [137])

Naukowcy ci byli bliscy udowodnienia, że kwestia zaufania w niezaufanym środowisku sprawia, że problem uwierzytelnienia jest nierozwiązywalny. Wynika to z tego, że jeżeli zawsze potrzebna jest niezależna osoba lub organ, które mają zweryfikować autentyczność dokumentu, to one również mogą brać udział w zmowie. Jednak autorzy zdali sobie sprawę, że gdyby trzeba było dodawać zaufane strony ręczące za uczciwość pozostałych uczestników, to lista rozszerzałaby się w nieskończoność – potrzebny byłby cały świat i to wciąż mogłoby być za mało. Zauważyli oni również, że jeżeli ten problem odwrócić i stworzyć system powiązanych ze sobą dokumentów, w którym wszyscy byliby świadkami jego poprawności, to problem jest rozwiązany. Ponadto zaobserwowali oni, że użycie czasu

jako uzgodnionego konstruktu[1] społecznego, umożliwia potwierdzenie, że dokument istniał w określonej formie w określonym czasie. Wyniki swoich przemyśleń i prac opublikowali w artykule „***How to Timestamp a Digital Document***" [81] w 1990 roku. Wyniki te i przemyślenia stały się kluczową częścią „*Bitcoin: A Peer-to-Peer Electronic Cash System*" [143] i oprogramowania wspomagającego *blockchain Bitcoin*. Jak powiedzieli Stuart Haber i Scott Stornetta w wywiadzie [137]: „*... Jedyną rozsądną rzeczą jest stworzenie systemu, w którym (1) istnieje sposób na zarejestrowanie lub certyfikowanie pliku. Wynikiem tego procesu (rejestracji przyp. autora) musi być (2) jakiś dodatkowy plik. Nazwaliśmy go „certyfikatem znacznika czasu", ale można go również nazwać „dowodem rejestracji".Trzecią częścią systemu jest to, że musi istnieć (3) sposób sprawdzenia tych rzeczy (certyfikatu zncznika czasu lub dowodu rejestracji przyp. autora)...*" Do generowania „dodatkowych plików" wykorzystuje się funkcje skrótu [81], które są również podstawą działania systemu *Bitcoin*. Ponadto autorzy [81] stwierdzają, że osiągnięciem Satoshi było ograniczenie ich pomysłu do obszaru pieniędzy po rozszerzeniu go o *sieć wydobywczą*.

Jednak zanim doszło do upowszechnienia i rosnącej popularności łańcuchów bloków, w 1992 roku technologię tę wzbogacono o drzewo skrótów (drzewo hash, drzewo Merkle), co zwiększyło wydajność tego rozwiązania. Koncepcja rozproszonej księgi transakcji umożliwiającej dostęp do danych finansowych zaufania publicznego, danych bankowości internetowej, danych handlu on-line i elektronicznego, została opublikowana w 2000 roku w pracy [106]. Według tej koncepcji każdy powiernik (użytkownik) księgi jest niezależny od innych (maksymalnie rozproszony system).

W 1998 roku Nick Szabo zaprojektował, a w 2005 roku opublikował w [184] (również [183]) koncepcję zdecentralizowanej waluty cyfrowej nazwanej „bit gold". Waluta ta nigdy nie została zaimplementowana, ale została nazwana „prekursorem architektury *Bitcoin'a*". Koncepcja tej waluty zapewniała sposób na weryfikację i znakowanie czasem nowych monet – jeżeli większość stron nie zgodziła się na przyjęcie nowych rozwiązań, nie mogła rozpocząć pracy nad kolejnym zadaniem [194, 185]. Jak powiedział Nick Szabo *„Starałem się jak najwierniej naśladować w cyberprzestrzeni cechy bezpieczeństwa i zaufania złota, a głównym z nich jest to, że nie zależy ono od organu centralnego"* [154].

W tym samym 1998 roku Wei Dai opracował wczesną propozycję *„anonimowego rozproszonego systemu waluty"* – B-money [49, 50], która została opublikowana na liście mailingowej *cipherpunk*[2] w listopadzie tego roku. Koncepcja ta upadła po opublikowaniu białej listy (*„white list"*), a odniesienia do tej koncepcji można znaleźć w propozycji Satoshi Nakamoto (*Bitcoin*).

Celem działania wspomnianej grupy *Cipherpunk* było stworzenie cyfrowego pieniądza, którego wartość byłaby niezależna od organizacji, która ten pieniądz wydaje. Z kolei koncepcje Nick'a Szabo (Bit Gold) i Wei Dai (B-money) dążyły do stworzenia cyfrowej gotówki o ograniczonej podaży i trudnej do generowania poprzez zdefiniowanie waluty w kategoriach *proof-of-work*. To doprowadziło do powstania prototypu systemu RPoW

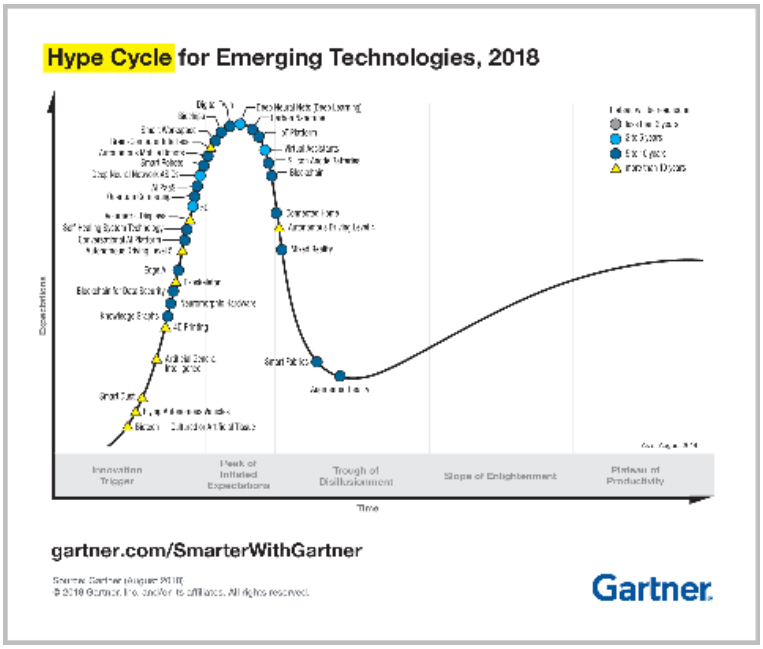
[1] Konstrukt jest to logiczna struktura powstała w czyimś umyśle; sposób postrzegania, konstruowania lub interpretowania zdarzeń; pewna abstrakcyjna, logiczna całość złożona z danych elementów; wytwór umysłu mający spójny i często przy tym skomplikowany układ.

[2] Cipherpunk - działacze propagujący powszechne stosowanie silnej kryptografii jako drogi do zmian społecznych i politycznych. Pierwotnie tworzyli oni nieformalną grupę komunikującą się za pośrednictwem list dyskusyjnych za cel stawiającą osiągnięcie prywatności i bezpieczeństwa poprzez aktywne wykorzystanie kryptografii. Grupa ta została humorystycznie nazwana „cypherpunks". Pochodzi ona od słów szyfr („cypher") oraz „cyberpunk". „Cyberpunk" jest odmianą fantastyki naukowej, skupiającej się na negatywnych konsekwencjach funkcjonowania ludzi w środowisku zaawansowanej techniki konputerowej i informatycznej.

(Reusable Proof of Work) opracowanego przez Hall'a Finney'a w 2004 roku [62]. System ten pozwalał na wymianę tokenów podpisanych z użyciem klucza RSA i funkcji skrótu. Tokeny te są wynikiem pewnych obliczeń i stanowią dowód ich wykonania. RPoW, chociaż nigdy nie wyszedł poza fazę prototypu, był wczesnym krokiem w rozwoju cyfrowej gotówki i wyrafinowanym prekursorem *Bitcoin'a*.

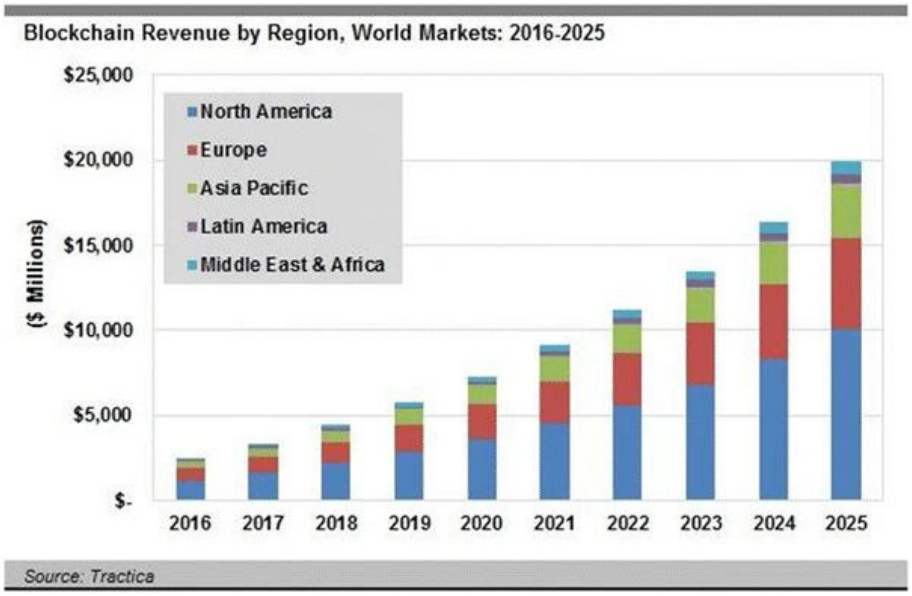
Dopiero 9 listopada 2008 roku opublikowano „białą księgę" [143, 195] (*white paper*[3]) pierwszego rozwiązania o nazwie *Bitcoin* łączącego wymienione wyżej koncepcje. Raport ten został opracowany przez osobę lub grupę osób podpisującą się pseudonimem Satoshi Nakamoto. Natomiast uruchomienie publicznej sieci *Bitcoin* oraz emisja pierwszego bloku miała miejsce 3 stycznia 2009 roku [5], a więc 19 lat od pierwszej praktycznej implementacji łańcucha bloków dokonanej przez Stuart'a Haber'a i Scott'a Stornetta. Ponadto warto podkreślić, że powstanie kryptowaluty *Bitcoin* było poprzedzone licznymi pracami i rozwiązaniami o charakterze prototypów, a nawet istnieją spekulacje o tym, że pod tożsamością „Satoshi Nakamoto" kryje się sam Nick Szabo.

Od tego czasu technologia łańcuchów bloków zdobywa coraz większą popularność i znajduje się na czele powstających (wzrastających) technologii [67] zajmując szóste miejsce w Hype Cycle dla rozwijających się technologii (rys. 6.2). Zgodnie z tym cyklem Gartner przewiduje, że realne i przewidywalne korzyści z zastosowania technologii łańcuchów bloków pojawią się i zostaną szeroko uznane w okresie 5-10 lat licząc od 2018 roku. Przewidywania te są zgodne z opinią innych autorów. Na przykład autor pracy [61] stwierdza, że zastosowanie łańcuchów bloków rośnie z każdym dniem, a w pracy [146] zauważono wyraźny wzrost przychodów z technologii łańcuchów bloków – ponad czterokrotny od 2016 roku do 2025 roku. Prognozowany wzrost przychodów z technologii łańcuchów bloków zaprezentowano na (rys. 6.3).



Rys. 6.2: Pozycja technologii łańcuchów bloków wśród wzrastających technologii (źródło: [67])

[3] White paper – biała księga jest to raport wykonany przez rząd, instytucję lub organizację, zawierający analizę jakiegoś produktu, usługi lub programu.



Rys. 6.3: Wzrost przychodów z technologii łańcuchów bloków (źródło: [146])

W pracach [23, 39] i podobnie w pracy [71] podkreślono, że łańcuch bloków stał się najbardziej obiecującą technologią w różnych branżach. W tych pracach skupiono się przede wszystkim na dziedzinie zarządzania łańcuchami dostaw twierdząc, że technologia łańcuchów bloków może doprowadzić do znacznego ulepszenia łańcucha dostaw.

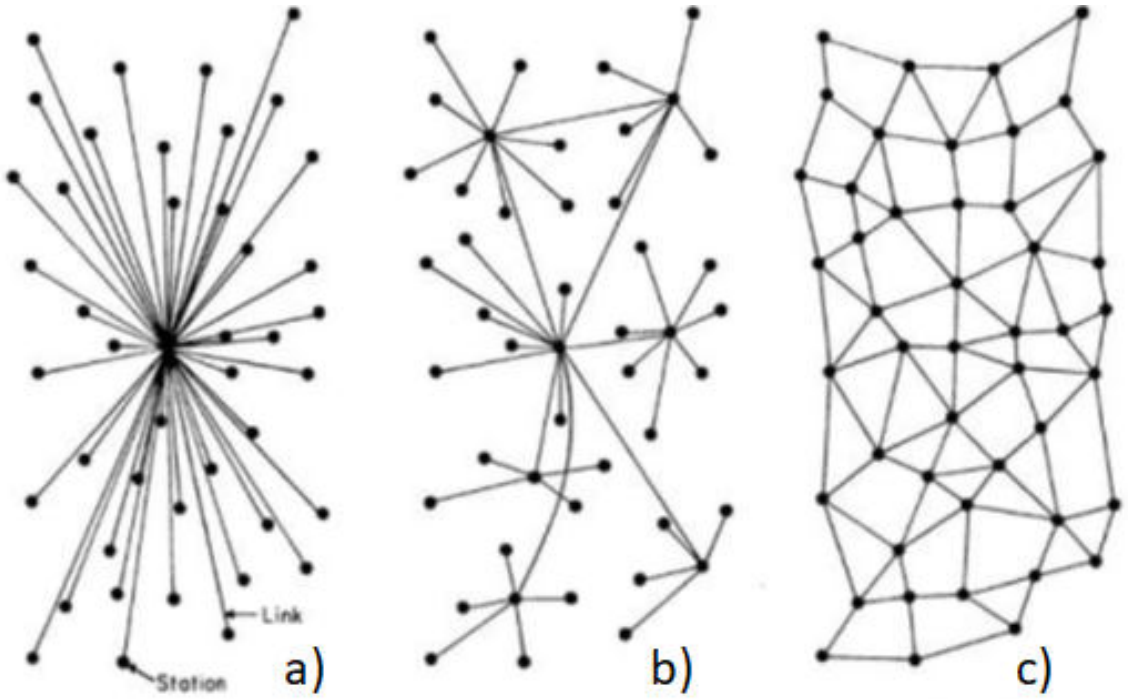
Tak duża popularność technologii łańcuchów bloków, a co za tym idzie tak znaczny wzrost przychodów z jej stosowania, jest między innymi konsekwencją jednej z podstawych zalet łańcuchów bloków (omówione w podrozdziale 6.3.1) jaką jest możliwość ich wykorzystania w budowie informatycznych systemów rozproszonych. Systemy te stają się coraz bardziej atrakcyjne wśród powszechnie stosowanych dotychczas rozwiązań zcentralizowanych i zdecentralizowanych. Dlatego w kolejnym podrozdziale zostaną przedstawione główne cechy tych systemów, czyli systemu zcentralizowanego, zdecentralizowanego i rozproszonego.

6.1. Systemy zcentralizowane, zdecentralizowane i rozproszone

Ze względu na szczególną rolę jaką mogą pełnić systemy rozproszone wykorzystujące łańcuchy bloków, w tym podrozdziale zaprezentowano koncepcję podziału systemów na scentralizowane, zdecentralizowane i rozproszone (rys. 6.4). Koncepcja ta jest szczegółowo omówiona w pracy [53]. Co prawda zawarte w tej pracy rozważania dotyczą systemów w rozumieniu ogólnym z ukierunkowaniem na systemy społeczne, to logika tego rozumowania może być odniesiona również do systemów informacyjnych. Warto nadmienić, że te rozważania są zgodne z prezentowaną przez autora rozprawy koncepcją informacyjnego wspomagania realizacji transakcji w rozproszonym środowisku.

6.1.1. Systemy scentralizowane (centralne)

Systemy scentralizowane są stosowane od dawna na przykład w systemach opartych na architekturze master i slaves. Systemy te bezpośrednio kontrolują działanie poszczególnych jednostek i przepływ informacji z jednego centrum. Wszystkie jednostki (reprezentowane przez węzły na rysunku 6.4a) w tym systemie są bezpośrednio zależne od



Rys. 6.4: Topologia systemów a) scentralizowany, b) zdecentralizowany, c) rozproszony (źródło: domena pubiczna, Paul Baran 1964)

centralnej władzy i dotyczy to zarówno wysyłania informacji, jak i otrzymywania poleceń. Jeżeli przyjmiemy, że jednostkami tego systemu są na przykład samorządy lokalne, to są one bezpośrenio kontrolowane przez władzę centralną i mają zakaz koordynowania działań oraz współdziałania między sobą. Zamiast tego, każdy z nich jest zobowiązany do wykonywania rozkazów z centrum.

6.1.2. Systemy zdecentralizowane

Kolejną koncepcją jest koncepcja systemu zdecentralizowanego, który dał początek hierarchicznym systemom cywilnym i wojskowym, a tym samym powstaniu imperiów. W tym systemie władza polityczna jest zdecentralizowana w porządku hierarchicznym w taki sposób, że pomiędzy węzłami centralnym i lokalnym znajdują się władze średniego szczebla. W tym systemie jedna władza kontroluje inne bezpośrednio pod nią podległe i jest równocześnie kontrolowana przez włądzę, która znajduje się bezpośrednio nad nią. W ten sposób władza centralna może kontrolować cały system. Schemat działania systemu zdecentralizowanego zaprezentowano na rysunku 6.4b. W systemie zdecentralizowanym - analogicznie jak w systemie scentralizowanym - samorządy lokalne nie mogą koordynować działań i współpracować między sobą. System zdecentralizowany znany jest również jako system warstwowy lub system hierarchiczny.

6.1.3. Systemy rozproszone

System rozproszony pojawił się w czasach nowożytnych jako narzędzie dla poszukujących wolności, ale militarnie mniej potężnych podmiotów, które dawało możliwość sprostania zagrożeniom ze strony imperium. Chociaż był to najstarszy system, to jego występowanie zostało zlikwidowane przez nowoczesne imperia, które uznały, że system

zdecentralizowany jest dla nich korzystniejszy, gdyż dzięki niemu można było realizować zasadę „dziel i rządź”.

System rozproszony nie ma centrum, a więc wszystkie węzły w tym systemie są połączone w sieć na zasadzie równości, niezależności i współpracy. Węzły najniższego poziomu (władze) mogą łączyć się w sieć z sąsiednimi węzłami za pomocą wspólnie uzgodnionych protokołów, budując w ten sposób silną sieć, która może być wielokrotnie bardziej odporna niż systemy scentralizowane lub zdecentralizowane. W przeciwieństwie do systemu scentralizowanego, który może zostać zniszczony poprzez zniszczenie niewielkiej liczby węzłów znajdujących się wyżej w hierarchii, odporność systemu rozproszonego wzrasta wraz ze wzrostem liczby uczestników. Jest to największa zaleta tego systemu oprócz braku przymusu zależności funkcjonalnej od centrum. Zaleta ta skutkuje tym, że samorządy lokalne są suwerenne w koordynacji działań i współpracy między sobą. Schemat działania systemu rozproszonego pokazano na rysunku 6.4c.

System rozproszony znany jest również jako system bezwarstwowy lub system bezhierarchiczny.

6.2. Koncepcja łańcucha bloków

Różne topologie systemów informatycznych, a zwłaszcza systemy zdecentralizowane i rozproszone stawiają wymogi, które z powodzeniem mogą spwłnić łańcuchy bloków. Zastosowanie łańcuchów bloków jest atrakcyjne zwłaszcza w zakresie wspomagania realizacji transakcji rozproszonych. Wówczas łańcuchy bloków stanowią rozdzaj rozproszonej bazy danych i stąd są nazywane „rejestrami rozproszonymi”. Stwierdzenie to jest zgodne z opinią autorów pracy [131], którzy twierdzą, że *„blockchain jest niczym więcej niż zwykłą strukturą danych z rozproszoną wielowersyjną kontrolą współbieżności”*.

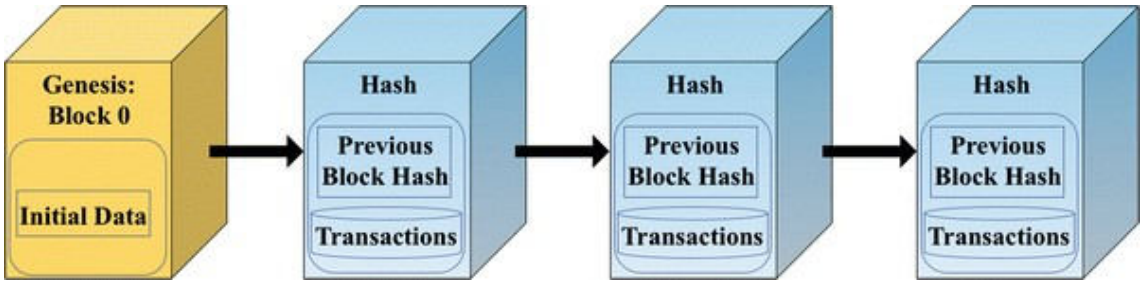
W wąskim znaczeniu [124] łańcuch bloków jest rodzajem struktury danych łączącej bloki danych w łańcuch z zachowaniem porządku chronologicznego. Struktura ta umożliwa przechowanie prostych, sekwencyjnie powiązanych danych, które mogą być weryfikowane w ramach systemu. Kombinacja bloków (łańcuch bloków) stanowi zdecentralizowany rejestr (księgę), który dzięki kryptografii jest odporny na manipulacje i jest niemożliwy do sfałszowania. W szerszym, bardziej ogólnym ujęciu, technologia łańcuchów bloku jest zdecentralizowaną infrastrukturą z paradygmatem obliczeń rozproszonych, który wykorzystuje kryptograficzne, łańcuchowe struktury blokowe do weryfikacji i przechowywania danych oraz zawiera rozproszone algorytmy konsesusu węzłów do generowania i aktualizacji danych[4]. Łańcuch bloków jest ukształtowany przez wszystkie bloki zawierające informacje o transakcjach w pewnym okresie czasu, uporządkowane chronologicznie.

Struktura łańcucha bloków

Jak zdefiniowano w pracy [80] Blockchain to powiązana jednokierunkowa lista niezmiennych, odpornych na manipulacje bloków, które są przechowywane w każdym uczestniczącym węźle. Każdy blok rejestruje zbiór transakcji i związane z nimi metadane. Wszystkie transakcje w łańcuchu bloków działają na identycznych danych rejestru przechowywanych w każdym węźle. Na rysunku 6.5 przedstawiono strukturę prostego łańcucha bloków.

Blok jest przyjmowany do łańcucha tylko wtedy, gdy wszyscy uczestnicy (węzły) osiągną konsensus co do kolejności i zawartości bloku [80]. W najprostszej postaci blok jest

[4] Jeżeli zaprezentowany opis łańcucha bloków zostanie rozszerzony o zautomatyzowany kod skryptowy do programowania i manipulowania danymi, wówczas mówimy o tzw. *inteligentnych kontraktach.*



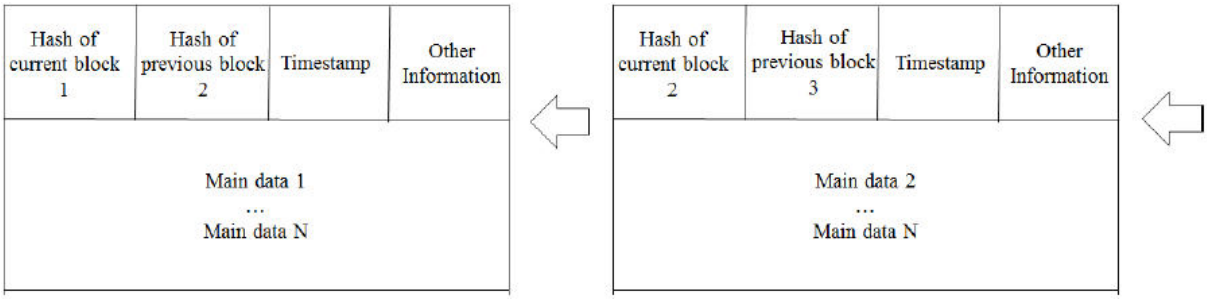
Rys. 6.5: Podstawowy schemat łańcucha bloków (źródło: [80])

zaszyfrowanym agregatem-zbiorem transakcji i związanych z nimi metadanymi, a jego istnienie daje pewność, że transakcje zostały wykonane i zweryfikowane. Jest to zagwarantowane dzięki procedurze konsensusu przyjęcia bloku do łańcucha (kilka algorytmów konsensusu wskazano w nstępnym rozdziale 6.2), a modyfikacja przyjętego bloku jest niemożliwa, ponieważ wymaga dokonania zmian wszystkich poprzednich bloków.

Każdy blok składa się z danych i nagłówka zawierającego wartość funkcji skrótu obliczoną dla poprzedniego bloku odzwierciedlającą relację pomiędzy blokami. Wraz z pojawiającymi się transakcjami powiększa się zbiór struktur danych (liczba bloków) składających się na rejestr (księgę). W pracy [119] podano ogólną strukturę łańcucha bloków, która zawiera:

— *dane główne*, które w zależności od zastosowania łańcuchy bloków mogą zawierać zapisy transakcji, zapisy rozliczeń bankowych, zapisy umów lub danych IoT (Internet of Things),
— *skrót bieżący bloku* jest to wartość funkcji skrótu policzona dla transakcji zapisanych w bloku z wykorzystaniem drzewa Merkle, co zmniejsza transmisję danych oraz zasoby obliczeniowe,
— *skrót poprzedniego bloku* jest to skrót z ostatniego (dotychczas) bloku w łańcuchu,
— *znacznik czasu* jest to czas wygenerowania bloku,
— *inne informacje*, czyli na przykład podpis bloku lub inne dane definiowane przez użytkownika.

Na rysunku 6.6 zaprezentowano schemat zawartości bloków w łancuchu. Ciało bloku składa się z licznika transakcji i samej transakcji. Maksymalna liczba transakcji zawartych w bloku zależy od wielkości bloku i wielkości każdej transakcji. Łańcuch bloków wykorzystuje mechanizm kryptografii asymetrycznej do walidacji i uwierzytelnienia transakcji.



Rys. 6.6: Łańcuch bloków oraz struktura bloków (źródło: [119])

Działanie łańcucha bloków

Każdy nowo utworzony blok jest dołączany do istniejącego łańcucha bloków. Pierwszym, początkowym blokiem łańcucha jest tzw. blok *genesis* [52], [80]. Ma on numer 0, a jego zawartość jest „twardo zakodowana" w aplikacji łańcucha bloków. Wszystkie inne bloki są kolejno dołączane do poprzenich, dlatego łańcuch rośnie w miarę dodawania nowych bloków. Decentralizacja systemów opartych na łańcuchu bloków, jest zapewniona poprzez możliwość przechowywania i utrzymywania pełnej kopii łańcucha przez wszystkich uczestników (wszystkie węzły).

Wprowadzenie nowego bloku do łańcucha bloków wymaga realizacji następujących etapów [119, 146]:

1. wysłanie transakcji do sieci P2P[5] złożonej z komputerów zwanych węzłami,
2. walidacja transakcji (wiadomości) przez węzły odbiorcze sieci i zapisanie poprawnych transakcji do bloku,
3. wykonanie przez wszystkie węzły algorytmu konsensusu dla bloku na przykład dowód wykonania pracy (PoW), dowód udziału (PoS) lub inny,
4. dopisanie poprawnego bloku do łańcucha w sposób trwały i niezmienny co skutkuje rozszerzeniem łańcucha.

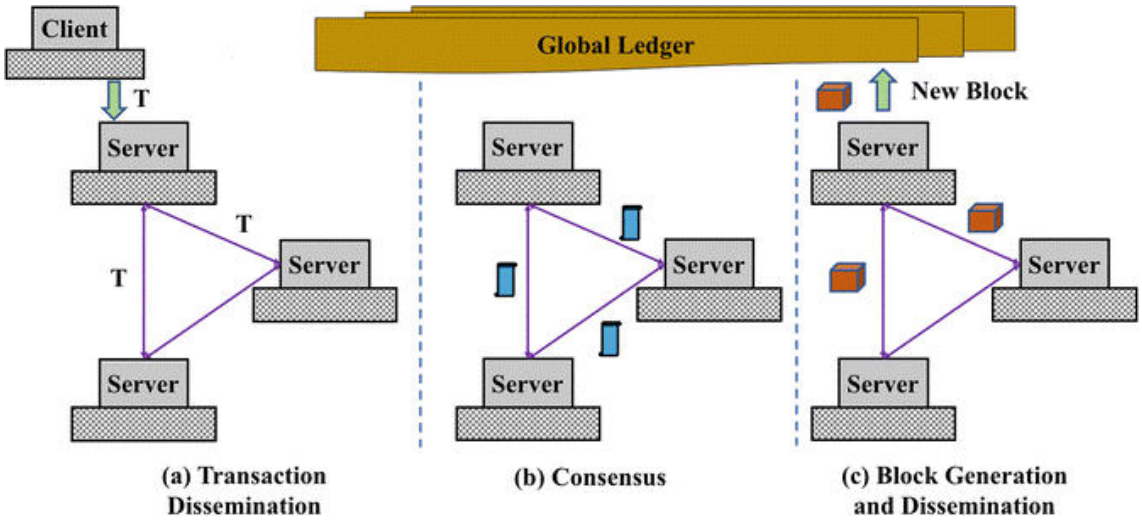
Zaprezenowany proces jest zgodny z działaniem łańuchów bloków opisanym w [80]. Proces ten pokazano na rysunku 6.7. Ilustruje on trzy główne fazy realizowane przez każdą aplikację łańcucha bloków w celu utworzenia nowego bloku. Klient przekazuje żądanie transakcyjne do jednego z serwerów. Serwer ten rozsyła żądanie do wszystkich pozostałych serwerów. Faza ta jest określana jako rozpowszechnianie transakcji. Gdy wszystkie serwery mają kopię żądania klienta, inicjują protokół konsensusu. Wybór bazowego protokołu konsensusu (algorytmu konsesnusu) wpływa na złożoność czasową i zużycie zasobów. Zwycięzca fazy konsensusu generuje następny blok i przesyła go do wszystkich pozostałych serwerów. Ta transmisja jest równoważna z zakończeniem wpisu nowego bloku do globalnej księgi rozproszonej.

W pracy [147] stwierdzono, że powszechnie stosowanym mechanizmem komunikacji między węzłami sieci łańcuchów bloku jest *peer-to-peer*. Z powodu pewnych ograniczeń związanych z tym podejściem coraz cześciej pojawiają się łańcuchy bloków oparte na „chmurze". Charakteryzują się one dużą elstycznością zakresu zastosowań – od pojedyńczego repozytorium dla łańcucha bloków do wielu funkcji w systemach o szerokiej dostępności. Zazwyczaj zdecentralizowane systemy publiczne są oparte na sieciach *peer-to-peer*, a rozwiązania zcentralizowane na „chmurze".

Sposoby osiągania konsensusu

Jak zauważono w pracach [131, 147] w publicznym systemie zdecentralizowanym krytycznym problemem związanym z przekazywaniem danych w sieci jest odpowiedź na pytanie: Jak zweryfikować autentyczność i aktualność informacji otrzymanych z sieci? W publicznym łańcuchu bloków ta weryfikacja jest zapewniona przez konsensus sieciowy.

[5] Peer to peer (P2P) jest to sieć komputerowa, w której wszystkie urządzenia są równe w hierarchii. Oznacza to, że nie istnieje centralny administrator zarządzający pozostałymi komputerami – każde urządzenie działa jako host z identycznymi uprawnieniami. W procesie wymiany informacji pełni on jednocześnie funkcję klienta pobierającego zasoby cyfrowe oraz funkcję serwera, który je udostępnia.



Rys. 6.7: Główne fazy wprowadzania transakcji do łańcucha (źródło: [80])

Taki mechanizm może być stosowany w prywatnych łańcuchach bloków poprzez wykorzystanie *crowdsourcingu*[6].

Funkcja konsensusu jest mechanizmem, który sprawia, że wszystkie węzły łańcucha bloków osiągają zgodę odnośnie transakcji. Istnieje więc pewność, że ostatni blok został dodany do łańcucha poprawnie z gwarancją, że zachowana wiadomość jest niezmieniona i nie dojdzie do ataku „fork attack". Ponadto funkcja ta zapewnia ochronę przed atakiem złośliwym. Inaczej mówiąc algorytm rozproszonego konsensusu pozwala wypracować wspólną decyzję dotyczącą przyjęcia bloku do łańcucha, która jest akceptowana przez większość węzłów [114, 75]. Istnieje wiele mechanizmów konsensusu. Do najbardziej popularnych zaliczyć można:

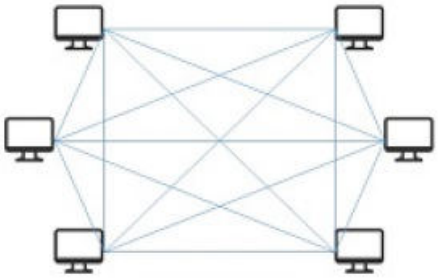
— *dowód wykonania pracy* PoW (Proof of Work) – w celu zaakceptowania bloku przez użytkowników sieci (węzły) kopacze muszą wykonać dowód pracy, który dotyczy wszystkich danych w bloku. Trudność tej pracy jest dopasowana w taki sposób, aby ograniczyć tempo generowania nowych bloków do 10 minut na blok. Ze względu na bardzo małe prawdopodobieństwo pomyślnego wygenerowania dowodu wykonania pracy nie można przewidzieć, który komputer będzie w stanie wygenerować kolejny blok [92], [143];
— *dowód udziału* PoS (Proof of Stake) – o ile dowód wykonania pracy powoduje duże straty energii i mocy obliczeniowej, to dowód udziału nie wymaga kosztownej mocy obliczeniowej. PoS zapewnia również zwiększoną ochronę przed złośliwymi atakami w sieci [103]. Ta dodatkowa ochrona pochodzi z dwóch źródeł:
  — dużego kosztu wykonania ataku,
  — zmniejszenia zachęty do ataku, gdyż napastnik musiałby posiadać większość wszystkich zasobów;
— inne istotne algorytmy takie jak na przykład: *Proof-of-Authority* [151], *Practical Byzantine Fault Tolerance* [38, 35].

[6] Crowdsoursing – jest to proces, w ramach którego organizacja przeprowadza outsourcing zadań wykonywanych tradycyjnie przez pracowników do niezidentyfikowanej, zwykle bardzo szerokiej grupy ludzi w formie otwartego zapytania (ang. crowd „tłum", sourcing „pozyskiwanie, zaopatrywanie się").

Rodzaje łańcuchów bloków

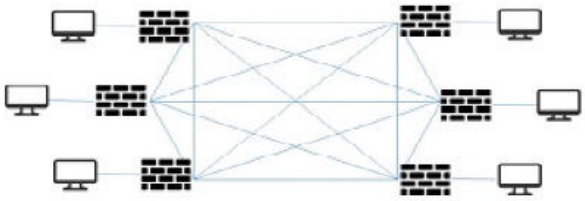
Można wyróżnić kilka sposobów klasyfikacji (taksonomii) łańcuchów bloków. Najbardziej rozpowszechnionym jest ich podział według sposobu regulacji, czyli kontroli dostępu. Zgodnie z tym kryterium wyróżnia się:

1. *publiczne łańcuchy bloków*, które nie wymagają zgody (zatwierdzenia) do przyłączenia się do bloku. Każdy może publikować i walidować transakcje (np. Bitcoin [22] lub Ethereum [57]). Topologia (struktura) sieci publicznego łańcucha bloków została zaprezetnowana na rysunku 6.8.



Rys. 6.8: Topologia sieci publicznego łańcucha bloków (żródło: [119])

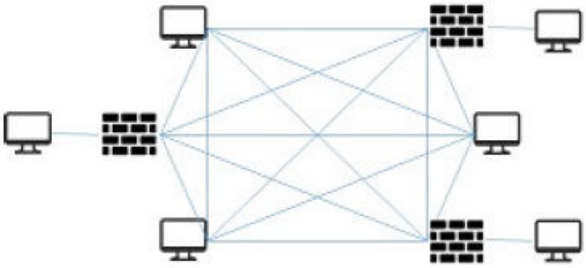
2. *prywatne łańcuchy bloków*, w których udział w łańcuchu bloków jest regulowany przez właściciela decydującego o takich kwestiach jak: wynagradzanie za kopanie, uprawnienia dostępu do sieci, lub wskazanie węzłów uprawnionych do walidacji łańcucha. Przykłady zastosowania prywatnych łańcuchów bloków omówiono w pracy [70] na przykład Ripple [167] lub CAPER [12]. Należy zauważyć, że istnienie jednego właściciela w przypadku łańcuchów prywatnych oznacza, że nie są one zdecentralizowane. Na rysunku 6.9 zaprezentowano topologię (strukturę) sieci prywatnego łańcucha bloków.



Rys. 6.9: Topologia sieci prywatnego łańcucha bloków (żródło: [119])

3. *konsorcjalne lub federacyjne łańcuchy bloków*, będące własnością grupy właścicieli, którzy posiadają ograniczony dostęp i zakres działań. Algorytm konsensusu jest zwykle wykonywany przez wybraną grupę użytkowników, co zwiększa poufność transakcji i przyśpiesza jej walidację. Przykładem takiego typu łańcucha jest Hyperledger - Fabric [64]. Ten rodzaj łańcuchów jest stosowany w instytucjach finansowych (np. Corda [43]), sektorze energetycznym [63] i ubezpieczeniach (np. B3 Technologies [188]). Struktura sieci konsorcjalnego łańcucha bloków jest zaprezentowana na rysunku 6.10.

Nieco szerszą i nowocześniejszą klasyfikacją łańcuchów bloków jest klasyfikacja zaprezentowana w pracach [159, 80]. Topologię łańcuchów bloków zaprezentoaną w tych pracach przedstawiono na rysunku 6.11. Na pierwszym miejscu znalazły się w niej *publiczne łańcuchy bloków* (Bitcoin [143], Ethereum [215]), które pozwalają każdemu węzłowi uczestniczyć w procesie konsensusu i każdy węzeł może wygenerować następny ważny



Rys. 6.10: Topologia sieci konsorcjalnego łańcucha bloków (źródło: [119])

blok. Na ostatniej pozycji znalazły się *prywatne łańuchy bloków*, ponieważ pozwalają tylko niektórym węzłom brać udział w procesie konsensusu i tylko podzbiór tych węzłów może wygenerować następny blok.

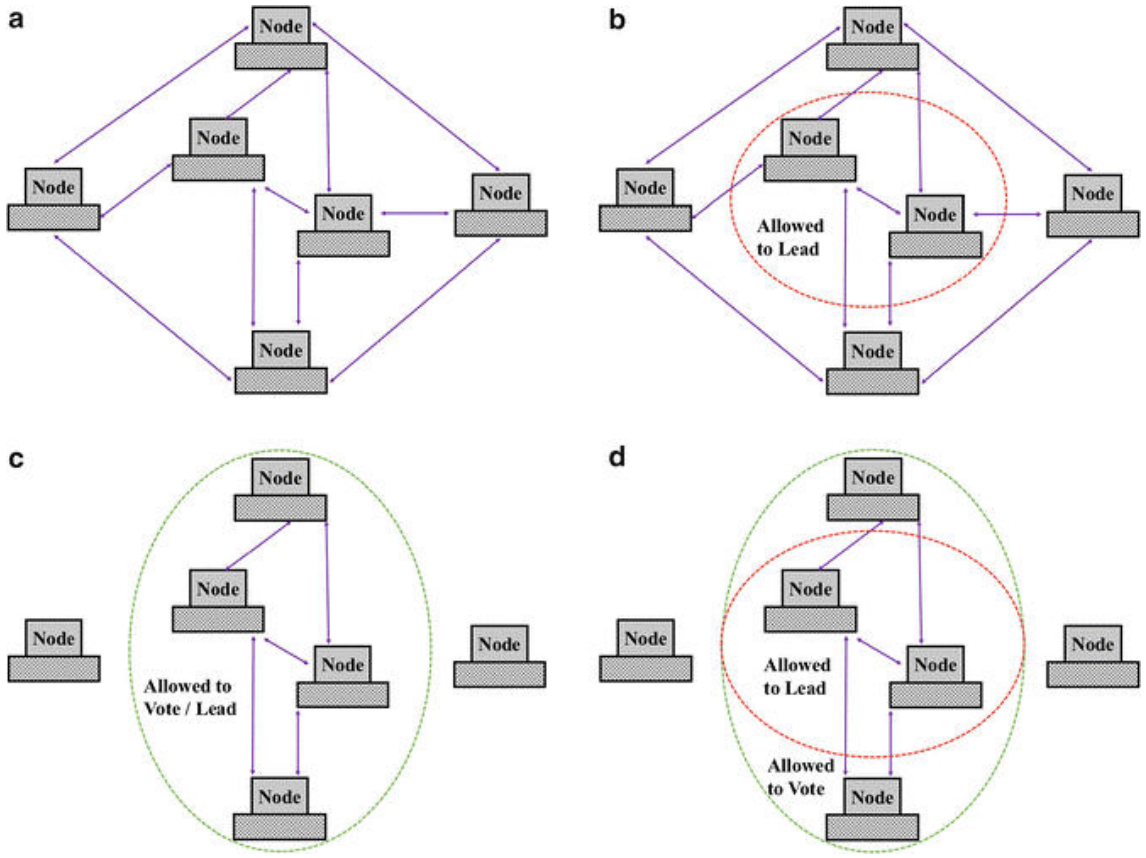
Prywatne łańcuchy bloków określają wstępnie wybrane węzły, które są uprawnione do korzystania z rejestru (księgi), co wynika ze zobowiązań i rozliczeń węzła w realizowanych transakcjach. W tym rodzaju łańcucha zbiór uprawnionych węzłów jest ograniczony przez co dostęp do informacji o transakcjach jest ograniczony. Podział na publiczne i prywatne łańcuchy bloków z reguły odpowiada podziałowi na łańcuchy zcentralizowane i zdecentralizowane. Zazwyczaj łańcuchy prywatne są scentralizowane, a publiczne zdecentralizowane. W łańcuchu zdecentralizowanym (takim jak *Bitcoin*) każdy uczestnik transakcji ma takie same uprawnienia, a w łańcuchu zcentralizowanym uprawnienia (dostępy, widoki) są określane przez centralny organ.

Zgodnie ze spostrzeżeniami w pracy [93] (przytoczonej w [147]) *„jedyne rozróżnienie między publicznym i prywatnym łańcuchem bloków jest związane z tym komu wolno uczestniczyć w sieci, wykonywać protokół konsensusu i utrzymywać wspólną księgę (rejestr)”*. W odróżnieniu od prywatnych, publiczne łańcuchy bloków pozwalają na publicznie widoczne transakcje i zdolność do ich generowania przez wszystkich potencjalnych uczestników. Każdy może zobaczyć lub wysłać transakcje i uczestniczyć w procesie jej realizacji, a informacje o transakcjach są szeroko dostępne dla uczestników sieci w otoczeniu publicznym.

Hybrydowe łańcuchy bloków (np. kryptowaluta Ripple [175]) znajdują się pomiędzy tymi dwoma skrajnościami. Pozwalają one każdemu węzłowi brać udział w procesie osiągania konsensusu, ale tylko niektóre, wyznaczone węzły mogą tworzyć następny blok. Kolejne, łańcuchy typu *Permissioned* są ograniczonym wariantem publicznych łańcuchów bloków. Rejestrują one kilka węzłów jako członków z równymi prawami, czyli każdy węzeł jako część systemu może utworzyć blok. Pozwalają one na odzwierciedlenie wewnętrznej pracy dowolnej organizacji czy grup mediów społecznościowych.

Zastosowania poszczególnych typów łańcuchów bloków(zgodnych z wyżej omówioną klasyfikacją) pokazano na rysunku 6.12 w wymiarach łańcuchów *publicznych-prywatnych* i *zdecentralizowanych-zcentralizowanych*. Pokazano na nim, że

— klasycznym *publicznym*, *zdecentralizowanym* łańcuchem bloków jest *Bitcoin* (lewa dolna ćwiartka),
— *zdecentralizowany*, *prywatny* łańcuch bloków jest zarządzany przez tłum (*crowd*), który głosuje nad tym, kto otrzymuje prawa dostępu (lewa górna ćwiartka),
— korporacje, które obawiają się, że nieuprawnione podmioty będą miały dostęp do ich transakcji, będą stosowały systemy *prywatne*, a dostęp będzie zarządzany w sposób *zcentralizowany* przez centralne biuro (prawa górna ćwiartka),
— analiza rządowych stron internetowych sugeruje, że zastosowanie łańcucha bloków mo-



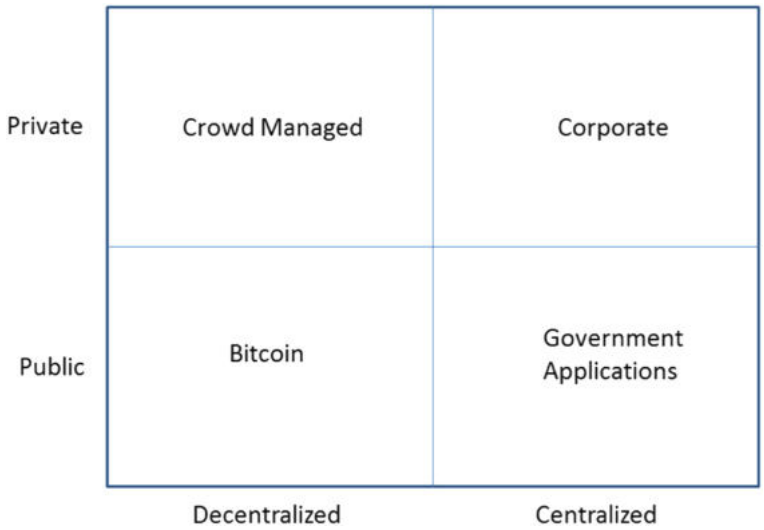
Rys. 6.11: Topologie łańcuchów bloków: (a) publiczny, (b) hybrydowy, (c) *permissioned*, (d) prywatny (źródło: [80])

głoby być *publiczne*, ale na pewnym poziomie rząd mógłby zarządzać dostępem do aplikacji w sposób *zcentralizowany*, zwłaszcza w przypadku potencjalnie destrukcyjnych użytkowników (dolna prawa ćwiartka).

Wykorzystanie różnych architektur Blockchain do gromadzenia i przetwarzania transakcji w różnych obszarach (np. w księgowości, audycie, łańcuchu dostaw) zaprezentowano w pracy [147]. W tej pracy przedstawiono również możliwość wykorzystania publicznych łańcuchów bloków opartych na komunikacji *peer-to-preer*, a przede wszystkim opartych na chmurze w konfiguracji prywatnej. Ponadto omówiono możliwość wykorzystania łańcucha bloków w odmianie konsorcjalnej do wspomagania realizacji transakcji w innych technologiach, takich jak hurtownie danych i bazy danych.

Oprócz omówionych powyżej kryteriów łańcuchy bloków można podzielić uwzględniając:

1. **rodzaj uprawnień:**
   a) *łańcuchy bez kontroli uprawnień*, czyli wszyscy użytkownicy mogą wykonywać te same operacje na łańcuchu, a więc zarządzanie uprawnieniami nie jest wymagane. Najbardziej znanymi przykładami są *Bitcoin* i *Ethereum*, ale są również inne takie jak: Litecoin [120], Monero [140] lub Zcash [222],
   b) *łańcuchy z kontrolą uprawnień* pozwalające na kontrolę, którzy użytkownicy mogą wykonywać transakcję. Przykładami takich łańcuchów są Monax [4] i Multichain [76];
2. **rodzaj zachęty:**



Rys. 6.12: Zastosowania łańcuchów bloków według ich typów: *prywatny-publiczny*, *zdecentralizowany-zcentralizowanego* (źródło: [147])

a) *łańcuchy bloków z tokenem*, w których transakcja i zachęta są oparte na tokenach wymienianych lub dawanych uczestnikom (np. *Bitcoin*, *Ethereum*),
b) *łańcuchy bloków bez tokenów* nie polegające na walucie wirtualnej (np. Hyperledger – Fabric, RootStock [113]);

3. **tryb pracy:**
   a) *zorientowane na logikę*, które pozwalają na wykonanie pewnej logiki. Najbardziej popularnymi implementacjami są inteligentne umowy, ale mogą być również wykorzystane do uruchomienia innych aplikacji. Przykładami są Ethereum, Hyperledger - Fabric, NXT [169] i Counterparty [46],
   b) *zorientowane na transakcje* mające na celu wyłącznie śledzenie zasobów cyfrowych na przykład Bitcoin, Monero, Ripple.

Topologia i rodzaj łańcuchów a bezpieczeństwo

Rodzaj systemu (publiczny-prywatny oraz centralny-rozproszony) ma istotny wpływ na bezpieczeństwo realizacji wspomaganych transakcji. Problematyka ujawniania informacji finansowych i publicznie udostępnianych baz danych jest dyskutowana od lat 70 XX wieku [152], a współczesne ekosystemy biznesowe wymagają dzielenia się informacjami i wymianą danych w celu poprawy wydajności i efektywności. Można więc zaobserwować konflikt między przejrzystością i poufnością – im więcej informacji jest udostępnianych tym bardziej przejrzyste jest przedsiębiorstwo, ale również wzrasta możliwość naruszenia tajemnicy handlowej i poufności. Jest to istotny problem współczesnego bisnesu [203] i odzwierciedla on dylemat *„współpraca kontra konkurencja”*.

Problematyka ta była również zauważona w pracy [20], a autorzy pracy [203] twierdzą, że stosowanie technologii łańcuchów bloków ma potencjał, aby znacząco wpłynąć zwłaszcza na dziedziny rachunkowości i audytu. Według nich technologia ta *zapewnia metodę współdzielenia bazy danych pomiędzy uczestnikami, nawet, jeśli sobie nie ufają, oraz tworzy rynek do przekazywania aktywów w oparciu o sieć peer-to-peer bez centralnego organu.*

Jednak w odniesieniu do łańcuchów bloków problem istnieje i jest związany z tym, że im więcej węzłów dodaje się do sieci, tym bardziej wiarygodne są dane i tym mniej poufny

jest łańcuch bloków. Ta cecha doprowadziła do rozwoju prywatnych łańcuchów bloków, w których tylko uprawnione strony mogą czytać zapisy i tworzyć transakcje. Prywatny łańcuch bloków zapewnia zamknięte bezpieczne środowisko kosztem przejrzystości danych i udziału społeczeństwa, co z kolei może ograniczyć jego odporność na manipulacje ponieważ menadżerowie mają pełną kontrolę nad prywatnym łańcuchem bloków. Oznacza to, że odporność prywatnego łańcucha bloków na manipulacje nie może być zagwarantowana, gdy kierownictwo jest w stanie manipulować danymi transakcji. Zatem dylemat przyjęcia określonego rodzaju łańcucha bloków polega na znalezieniu kompromisu pomiędzy prywatnym łańcuchem o zmniejszonej odporności na manipulacje a publicznym narażonym na ryzyko naruszenia poufności.

Reasumując – jak zauważono w [203] publicznie dostępne informacje w publicznych łańcuchach bloków stanowią potencjał dla działań wywiadu gospodarczego. Dlatego w miarę postępów w stosowaniu łańcuchów bloków będzie rosło zinteresowanie w ustanowieniu asymetrii informacji na temat realizowanych transakcji co doprowadzi do szerszego korzystania z łańuchów prywatnych.

6.3. Zalety i problemy łańcuchów bloków

6.3.1. Zalety łańcuchów bloków

Zalety łańcuchów bloków zostały szybko dostrzeżone i znalazły szerokie uznanie, co z kolei przyczyniło się do ich zastosowania w wielu obszarach na całym świecie. Na przykład w pracach [124, 119, 146, 146] autorzy wymieniają następujące zalety:

1. **Decentralizacja** – łańcuch bloków składa się z wielu współpracujących węzłów kształtujących sieć *end-to-end*, która nie posiada urządzenia lub organizacji zarządzającej. Prawa i obowiązki wszystkich węzłów są równe, a dane wymieniane między nimi są weryfikowane z wykorzystaniem technologii podpisu bez wzajemnego zaufania. Tak długo jak wymiana danych jest prowadzona zgodnie z ustalonymi zasadami, węzły nie mogą oszukiwać innych węzłów, a uszkodzenie lub utrata któregoś z węzłów nie będzie miała wpływu na działanie całego systemu. Zdecentralizowana struktura systemu łańcucha bloków poprawia efektywność działania systemu i zmniejsza koszty jego działania. Innymi słowy łańcuch bloków nie bazuje na jednym centralnym węźle, ale dane są zapisywane, przechowywane i aktualizowane przez wszystkie węzły.
2. **Otwartość i przejrzystość (transparentność)** – łańcuch bloków jest otwarty z wyjątkiem zaszyfrowanych prywatnych informacji stron transakcji. Każdy może brać udział w sieci łańcucha bloków, może przeszukiwać dane w łańcuchu oraz tworzyć współpracujące aplikacje wykorzystujące publiczny interfejs. Każde urządzenie może być węzłem, a każdy węzeł posiada kompletną kopię bazy danych rejestru rozproszonego. W ten sposób zapewniona jest wysoka transparentność całego systemu.
3. **Bezpieczeństwo** – łańcuchy bloków wykorzystują czysto matematyczne podejście do ustanowienia relacji zaufania oraz szyfrowania/deszyfrowania danych transakcji w oparciu o algorytmy kluczy asymetrycznych. Po dodaniu każdej nowej transakcji do łańcucha bloków wykonywany jest skomplikowany algorytm obliczania wartości skrótu z danych transakcji, tożsamości kontrahenta oraz historycznych danych z transakcji. Celem wykonania tego algorytmu jest zapewnienie poprawności i bezpieczeństwa transakcji. Każdy węzeł przechowuje wszystkie dane transakcji, a atakujący nie mogą

zmienić historycznych danych transakcji. W ten sposób wszystkie węzły mogą bezpiecznie wymieniać dane w niezaufanym środowisku. Jednocześnie następuje zmiana zaufania do ludzi w zaufanie do maszyn, które jest odporne na złośliwe lub nieuczciwe działanie innych osób.

4. **Odporność na manipulacje (niezmienność)** – zweryfikowane dane dodane do łańcucha bloków są na stałe przechowywane w tym łańcuchu. Próby dokonania zmiany danych w jednym węźle nie mają wpływu na dane przechowywane we wszystkich innych węzłach pod warunkiem, że więcej niż 51% węzłów w systemie jest w tym samym czasie prawidłowo kontrolowanych. Przez to łańcuch bloków jest niezykle stabilny i niezawodny jeżeli chodzi o bezpieczeństwo danych. Każda transakcja jest kryptograficznie powiązana z transakcją poprzedzającą i następującą w łańcuchu, co umożliwia prześledzenie całej historii transakcji.
5. **Autonomiczność** – reguły działania i informacje o danych w łańcuchu są publicznie dostępne, a wymiana danych między węzłami przebiega zgodnie z ustalonym algorytmem, w związku z tym w odniesieniu do wymienianych danych nie jest wymagane wzajemne zaufanie. Zasada konsensusu pozwala na przekazywanie i aktualizowanie danych przez każdy węzeł, a więc wszyscy mogą ufać systemowi i nikt nie może w niego ingerować.
6. **Anonimowość (poufność)** – technologie łańcuchów bloków rozwiązują problem zaufania w taki sposób, że transfer danych (lub nawet transakcja) może być anonimowy i jedyną wymaganą (znaną) informacją jest adres osoby (klucz publiczny) w łańcuchu bloków.
7. **Równość praw** – w pracach [146, 48] podkreślono, że ważną cechą łańcuchów bloków jest równość praw węzłów, a najważniejszą jest to, że interakcje pomiędzy węzłami w sieci zapewniają osiągnięcie zaufania, przy czym zaufanie jest oparte na samej sieci, a nie trzeciej stronie (zaufania).

Wśród licznych korzyści ze stosowania łańcuchów bloków, które wynikają po części z omówionych wyżej zalet, wymienia się [146]:

1. bezpośrednie wydzielenie, bez centralnego administratora, bazy danych,
2. zastąpienie zcentralizowanej logiki aplikacji przez zasady łańcucha bloków na przykład własny dowód poprawności danych i autoryzacji,
3. kontrolowanie danych (informacji i transakcji) wyłącznie przez upoważnionych użytkowników,
4. wysoką jakość danych, czyli ich kompletność, spójność, terminowość, dokładność i dostępność,
5. wytrzymałość, niezawodność i trwałość zapewnioną przez sieci zdecentralizowane, ponieważ łańcuch bloków nie ma scentralizowanego punktu awarii i jest bardziej odporny na złośliwe ataki,
6. integralność procesu polegającą na tym, że użytkownicy mają pewność, że transakcje zostaną wykonane zgodnie z poleceniami protokołu, co eliminuje potrzebę korzystania z usług zaufanej strony trzeciej,
7. przejrzystość i niezmienność zmian - każda zmiana jest publicznie widoczna dla wszystkich stron, zapewniając transparentność, a wszystkie transakcje są niezmienne, co oznacza, że nie mogą być zmieniane ani usuwane,
8. uproszczenie ekosystemu poprzez dodawanie wszystkich transakcji do jednej księgi publicznej, co zmniejsza bałagan i komplikacje związane z wieloma księgami,

9. skrócenie czasu realizacji trankcji i zapewnienie ciągłości działania systemu przez 24 godziny na dobę, 7 dni w tygodniu,
10. niższe koszty transakcyjne dzięki wyeliminowaniu osób trzecich, pośredników i kosztów ogólnych wymiany aktywów. Przez to łańcuchy bloków stanowią solidną i możliwą do zweryfikowania alternatywę dla tradycyjnych stosów własnościowych za ułamek kosztów,
11. ułatwnienie zarządzania systemem udostępniania zapisów, a więc umożliwienie różnym podmiotom (np. klientom, powiernikom i organom regulacyjnym) dostępu do ich własnych kopii współdzielonego systemu zapisów,
12. szybsze rozliczanie i rozrachunek transakcji, co ułatwia przejście z przetwarzania wsadowego na rozliczanie i rozrachunek bieżacy,
13. możliwość tworzenia samooopisujących się transakcji elektronicznych. Takie inteligentne umowy mogą wykorzystywać język programowania łańcucha bloków do tworzenia transakcji zależnych od kontekstu – na przykład przeniesienie *swap* ryzyka kredytowego może być wypłacone automatycznie zgodnie z wcześniej ustaloną logiką, która obserwuje strumień danych rynkowych,
14. zaangażowanie udziałowców i współpraca, na które zwrócono uwagę w pracy [39].

W pracy [124] autorzy prezentują przykład korzyści wynikających z zastosowania technologii łańcuchów bloków do systemów monitorujących pasma radiowe VHF/UHF. Stwierdzają oni, że decentralizacja, wysoki poziom bezpieczeństwa oraz ochrona prywatności pozwalają skutecznie rozwiązać problemy związane zarówno z utratą komunikacji, jak i wysokimi kosztami oraz niewydolnością systemów scentralizowanych, a także uniknąć masowej utraty danych lub wycieków spowodowanych przez organizacje scentralizowane w wyniku ataku lub niewłaściwego zarządzania. Ponadto zauważają oni, że dane w łańcuchu bloków są oznakowane czasem, weryfikowane i zapisywane przez węzły w oparciu o konsenus. Nie mogą być więc naruszone lub sfałszowane dzięki czemu dane z rejestratorów stacji radiowych mogą być przechowywane w sposób trwały i bezpieczny, a istnienie danych i pewność autentyczności można zawsze łatwo udowodnić.

6.3.2. Problemy związane ze stosowaniem łańcuchów bloków

Pomimo dużej i stale rosnącej popularności w zakresie stosowania łańcuchów bloków wielu autorów zwraca uwagę na problemy związane z tą technologią. W pracy [84] wskazano problem interoperacyjności polegający na ograniczeniach wynikających z konieczności stosowania odrębnych struktur bloków dla szerokiej gamy implementowanych rozwiązań. Z kolei w pracy [119] opisano takie wyzwania jak: atak większości, problem z rozgałęzieniami, problem skali, problem czasochłonności potwierdzania danych, brak regulacji prawnych związanych ze stosowaniem łańcuchów bloków oraz czas i koszt migracji z istniejących systemów. Natomiast autorzy pracy [146] podają szereg wyzwań związanych przede wszystkim z bezpieczeństwem, a także z wydajnością, na którą składają się dodatkowe, w stosunku do klasycznych baz danych, czynności takie jak: weryfikacja podpisu, algorytm konsensusu, konieczność wykonania operacji konsensusu przez wszystkie węzły sieci. Realizacja tych czynności prowadzi do dużego zużycia energii, które dodatkowo wzrasta ze względu na konieczność wykonania czynności „kopania”.

W pracy [48] zauważono, że rewolucyjna natura łańcuchów bloków już wywołała problemy bezpieczeństwa, a technologia łańcuchów bloków ciągle znajduje się w fazie badawczej. Ponadto w tej pracy w usytematyzowany sposób zaprezentowano problemy związane ze stosowaniem technologii łańcuchow bloków i wymieniono 4 grupy problemów

takie jak: ograniczenia techniczne, potencjalne ryzyka związane ze stosowaniem kryptografii, zagrożenia wynikające z otwartości i dostępności łańcuchów bloków oraz problemy z zarządzaniem bezpieczeństwem, samoorganizacją i anonimowością.

6.4. Zastosowania łańcuchów bloków

Technologia łańcuchów bloków, a dokładniej mówiąc zalety i korzyści jakie przynosi jej stosowanie, sprawia, że znalazła ona olbrzymie zastosowanie w wielu obszarach. W pracy [146] przewiduje się, że technologia łańcuchów bloków pozytywnie wpływa i będzie wpływać w przyszłości na różne dziedziny takie jak: środowisko inteligentnego miasta, bankowość i płatności, przewidywania demograficzne, zapisy medyczne, głosowanie, usługi cyfrowe w działaniach rządowych, usługi finansowe i wiele innych. Rozwój zastosowań łańcuchów bloków objął tak dużo obszarów, że one same stały się przedmiotem badań, czego przykładem może być praca [138].

W pracy [112] zaprezentowano rozbudowaną ośmiowymiarową taksonomię zastosowań łańcuchów bloków obejmującą dwadzieścia charakterystyk technicznych i pięć przypadków zastosowań. Taksonomię tę przedstawia tabela na rysunku 6.13.

W pracy [48] przedstawiono inny przykład klasyfikacji zastosowań technologii łańcuchów bloku (rys. 6.14) zwracając uwagę na reprezentatywne scenariusze wykorzystania łańcuchów bloków w cyberbezpieczeństwie, a dokładniej w zakresie:
— zdecentralizowanego bezpiecznego DNS,
— bezkluczowej infrastruktury podpisu (Keyless Signature Infratructure),
— bezpiecznego przechowywania danych (bezpiecznego dostępu do danych wrażliwych).

Z kolei w pracy [119] zaprezentowano nasepującą klasyfikację zastosowań łańcuchów bloków:
— *usługi finansowe*: zarządzanie aktywami, roszczenia ubezpieczeniowe, przetwarzanie operacji finansowych, płatności transgraniczne,
— *inteligentna administracja (Smart Government)*: paszport elektroniczny, metryki urodzenia, świadectwa ślubne, identyfikacja osobista, inteligentne społeczeństwo *(Smart Community)*,
— *inteligentna służba zdrowia (Smart Health Care)*: prowadzenie osobistej dokumentacji medycznej, kontrola dostępu, zarządzanie opieką zdrowotną, przetwarzanie ubezpieczeń,
— *Internet Rzeczy (IoT)*: inteligentne urządzenia *(Smart Appliances)*, „czujniki" łańcucha dostaw *(Supply Chain Sensors)*,
— *waluta cyfrowa (np. Bitcoin)* – poprzez wykorzystanie technik kryptograficznych możliwy jest transfer funduszy bez konieczności polegania na banku centralnym,
— *inteligentne umowy (Smart Contracts np. Ethereum)* – umowy cyfrowe, które kontrolują zasoby cyfrowe użytkownika formułując jego prawa i obowiązki. Nie są to tylko procedury komputerowe, ale mogą być one postrzegane jako strony kontraktu (umowy) odpowiadające za przetwarzanie i przechowywanie otrzymywanych wiadomości (danych) pochodzących również z otoczenia zewnętrznego,
— inne: ochrona praw intelektualnych, śledzenie łańcuchów dostaw, certyfikacja tożsamości, płatności międzynarodowe, internet rzeczy, poufność w leczeniu pacjentów lub przewidywania rynkowe.

A TAXONOMY OF BLOCKCHAIN APPLICATIONS		Reading access		Writing access		Main consensus mechanism				Anonymity level			Event handling			Data exchange type		Encryption			History retention	
		Pr	Pu	P	U	W	S	B	SD	A	P	I	N	F	C	T	C	U	P	T	W	R
Financial transactions	Anonymous cryptocurrencies		X		X	X				X			X			X				X	X	
	Cryptocurrencies		X		X	X					X		X			X		X			X	
	Wealth storage & micro-payments		X		X		X				X		X			X		X			X	
	Financial services		X	X		X					X		X			X		X			X	
	Energy-efficient financial services		X	X			X				X		X			X		X			X	
	Enterprise global and micro-financial transactions	X			X			X				X	X			X			X		X	
	Global centrally issued financial instruments	X		X					X			X	X			X			X		X	
Smart contracts	Smart contracts		X		X	X					X				X	X		X			X	
	Testing of smart contracts	X			X	X					X				X	X		X			X	
	Energy-efficient smart contracts		X		X		X				X				X	X		X			X	
	Testing of energy-efficient smart contracts	X			X		X				X				X	X		X			X	
	Community smart contracts		X	X		X					X				X	X		X			X	
	Energy-efficient community smart contracts		X	X			X				X				X	X		X			X	
	Enterprise smart contracts	X			X			X				X			X	X			X		X	
	Global agreements between institutions	X		X					X			X			X	X			X		X	
Data management	Global authentication and ownership		X		X	X					X			X		X		X			X	
	Sharing economies		X	X		X						X		X		X		X			X	
	Enterprise asset management	X			X			X				X		X		X			X		X	
Storage	Open access publishing		X	X		X						X		X			X	X			X	
	Content preview		X	X		X						X		X			X		X		X	
	Decentralized storage		X	X		X						X		X			X			X	X	
Communication	Broadcasting		X		X	X						X		X			X	X				X
	Discussion Forum		X	X		X						X		X			X	X				X
	IoT communication	X			X			X				X		X			X	X				X
Ranking	Global reputation & rating		X	X		X						X		X		X		X				X

LEGEND

X – characteristics belong to an application case
Reading access
Pr – Private: only authorized members of a limited community can read blockchain
Pu – Public: everybody can read a blockchain
Writing access
P – Permissioned: a user should be authorized to validate transactions
U – Unpermissioned: a user can validate transactions without authorization
Main consensus mechanism
W – Proof-of-work: consensus for secure blockchain updating is achieved by Proof-of-Work
S – Proof-of-stake: consensus for secure blockchain updating is achieved by Proof-of-Stake
B – Practical Byzantine fault tolerance: secure blockchain updating is achieved by agreements of trusted nodes
SD – Self-developed mechanism: consensus for secure blockchain updating is achieved by self-developed mechanism
Anonymity level
A – Anonymous: users do not have to provide any data for working with blockchain
P – Pseudonymous: users can work with a blockchain under a pseudonym
I – Identifiable: users should provide personal data to work with a blockchain

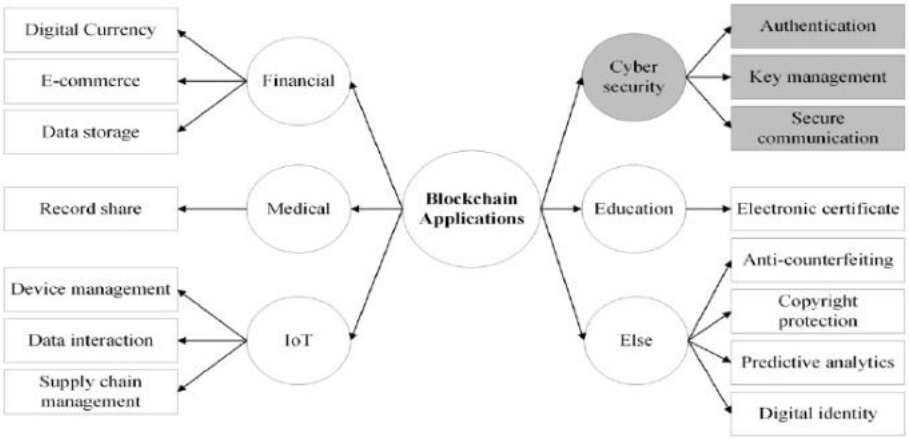
Event handling
No – No: blockchain does not support any events
F – Fixed: blockchain supports built-in events
C – Custom: blockchain supports processing of events created by user
Data exchange type
T – Transaction: logs of actions executed are exchanged among users and recorded on a blockchain
C – Content: digital assets are exchanged among users and recorded on a blockchain
Encryption
U – Unencrypted: all data on a blockchain is unencrypted
P – Partially-encrypted: data on a blockchain is partially encrypted
T – Totally-encrypted: all data on a blockchain is encrypted
History retention
W – Whole: blockchain keeps whole transaction history from a genesis block
R – Recent updates: blockchain keeps only recent updates of the transaction history

Rys. 6.13: Taksonomia zastosowań łańcuchów bloków (źródło: [112])

6.5. Wspomaganie realizacji transakcji

W poprzednim podrozdziale omówiono szeroki wachlarz zastosowań łańcuchów bloków, w których informacje o transakacjach stanowią elementarne dane w bloku. Transakcja w łańcuchu bloków jest taka sama jak każda transakcja rozproszona lub transakcja OLTP[7] [44]. Zwykle aplikacje łańcuchów bloków (takie jak Bitcoin) składają się z transakcji, które reprezentują wymianę pomiędzy dwoma podmiotami. Każda ważna transakcja jest zapisywana w bloku, który w celu zwiększenia wydajności może zawierać wiele transakcji. Aby zapewnić integralność danych o poszczególnych transakcjach w bloku, stosowane są

[7] OLTP - *Online Transaction Processing* jest to rodzaj przetwarzania danych, polegający na jednoczesnym wykonywaniu szeregu transakcji na przykład bankowości internetowej, zakupów, składania zamówień lub wysyłania wiadomości tekstowych.



Rys. 6.14: Obszary zastosowań łańcuchów bloków (źródło: [48])

drzewa skrótów Merkle oraz silne algorytmy kryptograficzne takie jak wartości funkcji skrótu [102], co czyni potencjalny atak niewykonalnym.

W kolejnych podrozdziałach przedstawiono kilka przykładów zastostosowania łańcuchów bloków. Spośród wielu, wybrano przykłady dotyczące: systemu wspomagającego operacje monitorowania zdarzeń ekonomicznych w przedsiębiorstwie produkcyjnym, systemu uwierzytelniania zapisów medycznych oraz systemy realizacji zadań Infrastruktury Klucza Publicznego.

6.5.1. System przetwarzania transakcji

W pracy [203] zaprezentowano projekt systemu przetwarzania transakcji TPS (Transaction Processing System) oparty na łańcuchu bloków. System ten jest dedykowany dla ciągłego monitorowania w czasie rzeczywistym ([164, 219]) i zapobiegania oszustwom w rachunkowości. System, oprócz technologii łańcuchów bloków, wykorzystuje również *dowód zerowej wiedzy* oraz *szyfrowanie homomorficzne*. Na rysunku 6.15a pokazano cykl realizacji zamówienia, w którym dostawca wysyła tokeny[8] „surowce" (np. drewno) do firmy produkcyjnej M, co wywołuje cztery wpisy w dzienniku dla dwóch podmiotów gospodarczych.

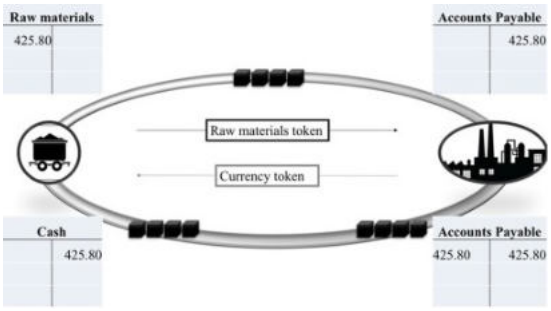
Z kolei na rysunku 6.15b pokazano cykl Zamówienie-Gotówka, w którym firma M sprzedaje gotowe wyroby co wywołuje cztery wpisy w dzienniku dla dwóch podmiotów gospodarczych. Oprócz transakcji między firmami (organizacjami), Bb-TPS może być również platformą komunikacyjną, wspomagającą realizację transakcji wewnątrz firmy, czyli pomiędzy jej oddziałami.

Na rysunku 6.15c pokazano przykład przedsiębiorstwa, którego filie, tworzą kompletny łańcuch dostaw w przemyśle (na przykład od poszukiwania i rafinacji ropy naftowej do sprzedaży detalicznej gazu). System przetwarzania transakcji zbudowany w oparciu o łańcuchy bloków łączy centrale i filie oraz rejestruje transakcje wewnątrzorganizacyjne.

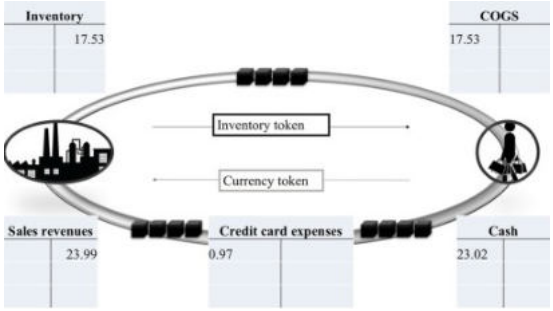
6.5.2. Uwierzytelnianie zapisów medycznych

W pracy [142], poświęconej problematyce zabezpieczania zapisów medycznych z wykorzystaniem *infrastruktury podpisu bezkluczowego opartej na łańcuchu bloków w chmurze*,

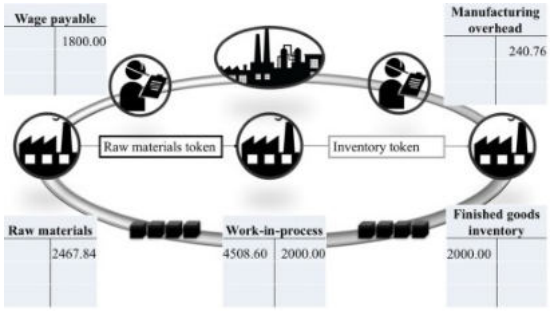
[8] Tokenizacja – jest to proces wiązania klasy aktywów świata rzeczywistego z tokenem nazywany jest tokenizacją aktywów



(a) Cykl „Zamówienie-Płatność”



(b) Cykl „Zamówienie-Gotówka”



(c) Cykl „Zamówienie-Gotówka”

Rys. 6.15: System przetwarzania transakcji TPS (źródło: [203])

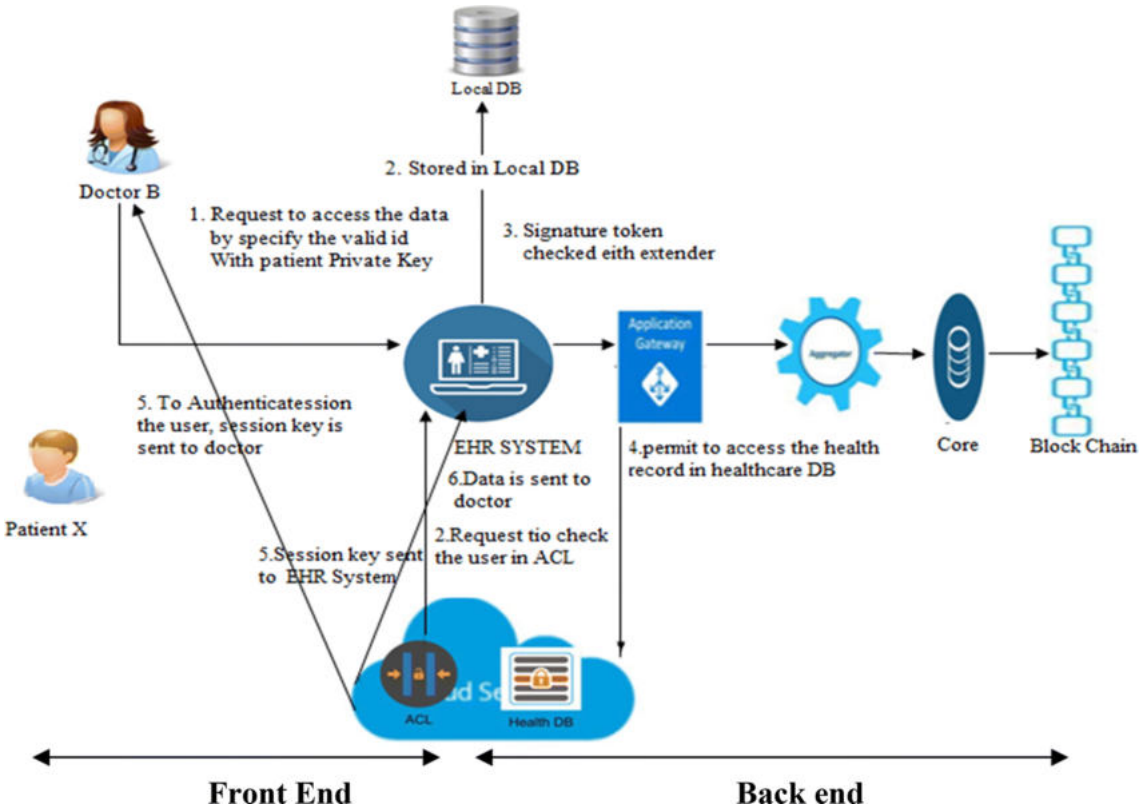
podkreślono konieczność zabezpieczania tych zapisów zwłaszcza pod kątem ich integralności. Scentralizowane przechowywanie danych jest podatne na cyberataki i dotyczy to również danych o zdrowiu. Autorzy omawianej pracy ([142]) twierdzą, że z tego powodu konieczne jest zaprojektowanie systemu wykorzystującego chmurę, który zapewni uwierzytelnienie, a także zapewni integralność zapisów zdrowotnych. Infrastruktura podpisu bez klucza zastosowana w proponowanym systemie do zapewnienia tajności podpisów cyfrowych zapewnia również aspekty uwierzytelniania. Ponadto integralność danych jest zapewniona przez proponowaną technologię łańcuchów bloków. Autorzy twierdzą, że czas odpowiedzi proponowanego systemu opartego na łańcuchu bloków jest prawie 50% krószy od technik konwencjonalnych, a koszt przechowywania jest o około 20% krótszy. Jednak zapewnienie integralności dokumentacji zdrowotnej pozostaje poważnym problemem [85] podobnie jak zapewnienie prywatności (poufności) [225]. Aby zagwarantować poufność dokumentacji medycznej zaproponowano, aby zapisy udostępniać na podstawie weryfikowania atrybutów i ich wartości. Opierając się na założeniu, że zdecentralizowana technologia łańcuchów bloków zapewnia dostępność oraz bezpieczeństwo danych, autorzy pracy [142] zaproponowali nową architekturę nazwaną *ProvChain* [117], która wspomaga przechowywanie, dostęp i walidację danych. System oparty na Blockchain z wykorzystaniem struktury drzewa Merkle, zabezpiecza zapisy i pomaga w utrzymaniu danych z zachowaniem prywatności. Z kolei do walidacji danych wykorzystywana jest koncepcja znacznika czasu.

W odróżnieniu od konwencjonalnych technik infrastruktury klucza publicznego, które zależą od kluczy w kryptografii asymetrycznej [25], [172], podpis w infrastrukturze bezkluczowej KSI (Keyless Signature Infrastructure) opiera się na funkcji skrótu, która zapewnia bezpieczeństwo wykorzystującemu ją systemowi. Podpis bezkluczowy KSI zapewnia poufność, bezpieczeństwo i integralność, a także przejrzystośćc danych (według autorów [142]).

Podpisane dane są przechowywane w łańcuchu bloków, aby później zostały zweryfikowane z uwzględnieniem podpisu i podpisanego czasu. Ponieważ KSI nie ma kluczy tak jak w *kryptografii Publicznej*, nie może on być skompromitowany. Jednak podpis bezkluczowy wykorzystuje funkcję skrótu, która zapewnia integralność danych. Należy zauważyć, że KSI:

— jest obiektem *offline* opartym na czasie rzeczywistym, ponieważ nie wymaga żadnego połączenia sieciowego dla swojego działania,
— nie wymaga, aby jakakolwiek strona trzecia utrzymywała dane związane z KSI, a samo rozwiązanie jest wysoce skalowalne,
— umożliwia weryfikację danych nawet po ich przekazaniu (przeniesieniu w dowolne miejsce) z jednoczesnym zapewnieniem dostępności tych danych.

Środowisko (*framework*) infrastruktury podpisu bezkluczowego opartego na łańcuchu bloków (*KSIBC*) realizuje sekwencję działań, które zapewniają bezpieczeństwo i integralność



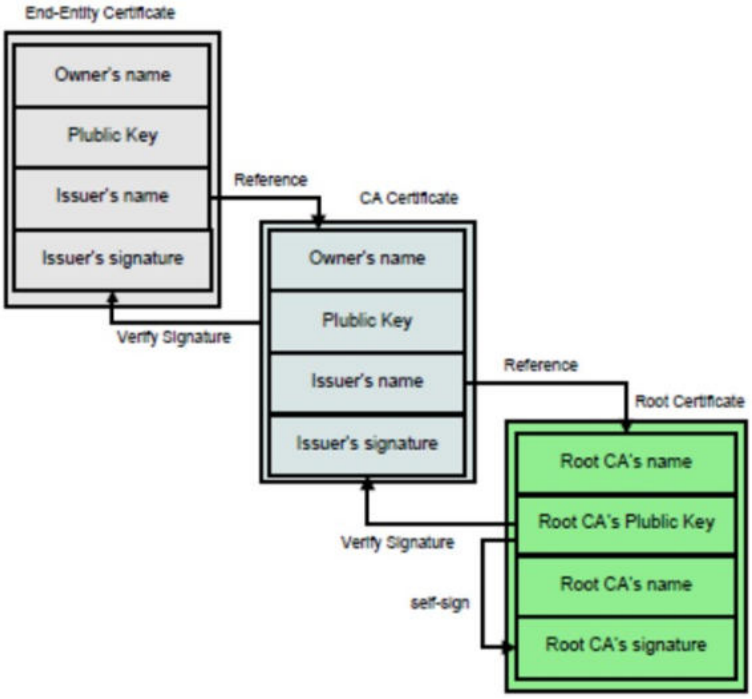
Rys. 6.16: Platforma uwierzytelniania danych medycznych (źródło: [142])

danych medycznych. W schemacie systemu, pokazanym na rysunku 6.16, ujęto *Front-end* realizujący komunikację interesariuszy procesu medycznego z systemem oraz *Back-end*, w którym realizowany jest proces zabezpieczania danych opieki zdrowotnej za pomocą infrastruktury bezkluczowej i technologii łańcuchów bloków.

Jak twierdzą autorzy [142] łańcuch bloków z KSI jest pierwszą i jedyną platformą używaną do sprawdzania integralności danych. KSI jest specjalnie wykorzystywany do uwierzytelniania i weryfikacji integralności zapisów medycznych. Ponadto wprowadza się infrastrukturę podpisu bez klucza wraz z łańcuchem bloków w celu zabezpieczenia integralności danych pacjenta. KSI Blockchain zapewnia bezpieczne uwierzytelnianie bez pomocy zaufanych organów.

6.5.3. Infrastruktura klucza publicznego

Infrastruktura Klucza Publicznego PKI dostarcza środków dla zapewnienia wiarygodności tożsamości w internecie, poprzez określenie zasad i procedur wydawania, zarządzania, zatwierdzania i dystrybucji certyfikatów cyfrowych w celu bezpiecznego stosowania szyfrowania z użyciem klucza publicznego [220]. Funkcjonowanie systemów PKI jest oparte na Urzędach Certyfikacji CA (Certtification Authority), które wydają certyfikaty zgodne ze standardem X.509, wiążące tożsamość subskrybenta (podmiot, właściciel pary kluczy) z kluczem publicznym komplementarnym z posiadanym kluczem prywatnym podmiotu. Aby certyfikat mógł być uznany i akceptowany przez ogół użytkowników musi być on wydany przez uznany Urząd Główny (Root CA) lub Urząd Podległy (Subordinate CA) wystawcy głównemu. Tożsamość urzędów podległych jest gwarantowana poprzez certyfikaty X.509 wydawane kolejno przez urzędy nadrzędne urzędom podległym - od Urzędu Głównego, poprzez Urzędy Podległe do certyfikatu podmiotu końcowego (End Entity Certificate). Certyfikaty tworzą ciąg o dowolnej[9] długości nazywany *łańcuchem zaufania* lub *ścieżką zaufania*. Na rysunku 6.17 pokazano ścieżkę zaufania, której początek wyznacza certyfikat podmiotu końcowego (End Entity Certificate), a koniec certyfikat wystawcy głównego (*Root certificate*). Oprócz weryfikacji tożsamości podmiotu, faktu posiadania klucza prywatnego komplementarnego do klucza publicznego i wydawania certyfikatu, Urzędy Certyfikacji publikują w postaci List Certyfikatów Unieważnionych CRL (Certificate Revocation List) informacje o zmianie statusu tych certyfikatów: zawieszenie, odwieszenie oraz unieważnienie.



Rys. 6.17: Ścieżka (łańcuch) zaufania (źródło: [218])

Systemy PKI poprzez ich centralizację stanowią pojedyńcze punkty awarii (Single Point Of Failure), co w połączeniu z możliwością wystąpienia błędów lub uchybień w funkcjonowaniu CA może prowadzić do poważnych konsekwencji w postaci wydania błędnie

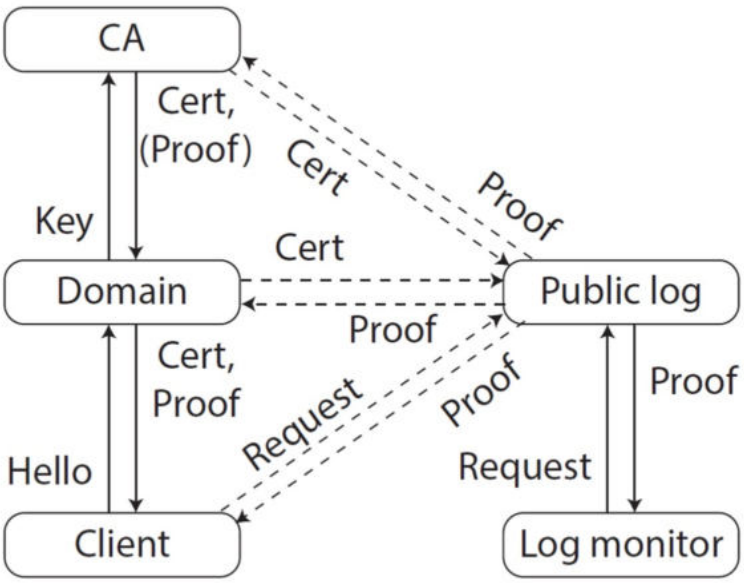
[9] Uwaga: standard X509v3 posiada rozszerzenie o nazwie *Basic Constraints* i to rozszerzenie może ograniczać maksymalną głębokość poprawnego łańcucha certyfikatów [24].

autoryzowanych certyfikatów [13]. Przykładem głośnego incydentu jest wykrycie ponad 500 fałszywych certyfikatów dla domen o wysokim statusie wydanych z naruszeniem zasad bezpieczeństwa przez holenderski urząd certyfikacji DigiNotar [160].

Innym przykładem może być DigiCert Sdn. Bhd., malezyjski pośredni urząd certyfikacji, który błędnie wydał 22 słabe certyfikaty SSL. W rezultacie główne przeglądarki musiały wycofać swoje zaufanie do wszystkich certyfikatów wydanych przez DigiCert Sdn. Bhd. Problemy z naruszeniem certyfikatu wystąpiły również w przypadku TrustWave, dużego amerykańskiego urzędu certyfikacji, który przyznał, że wystawił jednemu ze swoich klientów błędne certyfikaty pośrednie (dla pośrednich urzędów certyfikacji). Chociaż TrustWave wycofało certyfikat i stwierdziło, że nie będzie już wydawać klientom certyfikatów podrzędnych, to jednak sprawa ta ilustruje, jak łatwo jest popełnić błąd w urzędzie certyfikacji i jak poważne mogą być konsekwencje tych błędów [218].

Aby rozwiązać problemy SSL/TLS PKI, zaproponowno dwa podejścia: schemat oparty na logach (*log-based PKI*) oraz zdecentralizowane sieci weryfikacji *peer-to-peer* znane jako Web of Trust (WoT)

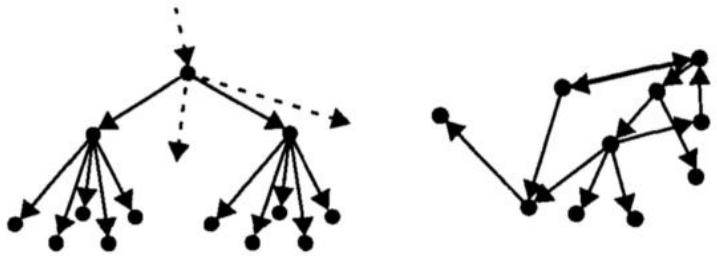
Ideą stanowiącą podstawę podejścia opartego na logach ***log-based PKI*** jest wykorzystanie wysoko dostępnych publicznych serwerów logów, które monitorują i publikują certyfikaty wydane przez urząd certfikacji. Publiczne logi gwarantują przejrzystość poprzez zapewnienie, że tylko publicznie zarejestrowane certyfikaty są akceptowane i zaufane przez klientów końcowych. Na rysunku 6.18 pokazano architekturę systemu PKI wykorzystującego system logowania. Stąd każde niewłaściwe zachowanie urzędu certfikacji zostanie wykryte przez użytkowników i serwery. Istnieją również inne propozycje mające na celu rozszerzenie cech schematu *Log-based PKI* o wsparcie dla unieważnienia certyfikatu i obsługę błędów. Niestety PKI oparte na logach nadal mają kilka problemów związanych na przykład z unieważnianiem certyfikatów, co zostało opisane w pracach [129, 130].



Rys. 6.18: Ogólna architektura modelu PKI opartego na logach. Linie ciągłe oznaczają informacje przesyłane podczas procesu wydawania i walidacji certyfikatów, linie przerywane - możliwe sposoby rejestracji certyfikatów w logu oraz sposoby uzyskiwania dowodów z logów dla certyfikatu (źródło: [130])

Podejście ***Web of Trust (WoT)*** jest całkowicie zdecentralizowane, w którym użytkownicy wskazują innych użytkowników jako zaufanych poprzez podpisanie ich certyfi-

katów klucza publicznego. Następnie certyfikat jest zaufany (akceptowany) przez stronę trzecią. W ten sposób użytkownik wchodzi w posiadanie certyfikatu zawierającego jego klucz publiczny i podpisy cyfrowe podmiotów, które uznały go za godnego zaufania. System ten daje korzyści wynikajce z jego rozproszonej natury, ponieważ usuwa wszelkie centralne punkty awarii. Jednakże utrudnia on nowym lub zdalnym użytkownikom dołączenie do sieci, ponieważ istniejący członek WoT musi spotkać się z nowym użytkownikiem osobiście, aby jego tożsamość została zweryfikowana, a klucz publiczny podpisany po raz pierwszy. Ponadto, w przeciwieństwie do podejścia opartego na urzędach certyfikacji, WoT nie ma prostego sposobu na unieważnienie klucza i certyfikatu. Użytkownik jest ograniczony do wybrania innego użytkownika, który będzie jego „wyznaczonym unieważniaczem" (*revoker*) z prawem do unieważnienia certyfikatu. Publikacja zmiany statusów certfikatu praktycznie polega na okresowym przesyłaniu list odwołań do przeglądarek. Metoda ta powoduje, że nieważne certyfikaty są zaufane do czasu, gdy listy odwołań przeglądarek nie otrzymają kolejnej aktualizacji [220]. Porównanie topologii klasycznej sieci PKI z siecią Web of Trust pokazano na rysunktu 6.19.



Rys. 6.19: Porównanie klasycznej Infrasrtuktury Klucza Publicznego (lewy rysunek) z rozwiązaniem Web of Trust (prawy rysunek) (źródło: [37])

Technologia łańcuchów bloków zapewnia wartościowe cechy bezpieczeństwa w odniesieniu do systemów PKI. Są to przede wszystkim: możliwość unieważnienia certyfikatu, eliminacja centralnych punktów awarii (*cetral-point-of-failura*) oraz wiarygodny zapis transakcji. Na przykład dzięki szybkiemu unieważnianiu certyfikatów, funkcjonalność PKI oparta na łańcuchu bloków może natychmiast odizolować zainfekowany urząd certyfikacji i wydane przez niego certyfikaty bez konieczności czekania do następnej aktualizacji list unieważnień certyfikatów CRL. Ponadto zastosowanie łańucha bloków w naturalny sposób zapewnia właściwość transparentności certyfikatów CT (Certificate Transparency) zaproponowaną przez Google [15].

Zastosowania łańcuchów bloków w budowie systemów PKI były przedmiotem wielu prac. Na przykład w pracy [7] zaproponowano rozwiązanie o nazwie *Blockstack* wykorzystujące łańcuch bloków *Bitcoin* w celu rejestracji nazw, które zą powiązane z kluczami publicznymi. Podobne rozwiązanie, wykorzystujące wiązanie nazw z kluczami publicznymi, pokazano w pracy [65] z wykorzystaniem platformy *Certcoin*. Innym przykładem jest system PKI wymieniony w pracy [218], podobny do *Blockstack*, który oparty jest na łańcuchu bloków i zrealizowany na platformie *Emercoin*. *Emercoin* jest publicznym łańcuchem bloków zbliżonym pod względem architektury do Bitcoina, w którym zastosowano hybrydowy konsensus *Proof-of-Work* i *Proof-of-Stake* w zależności od dostępności mocy wydobywczych. Nie posiada on jednak inteligentnych kontraktów i przechowuje wartości skrótów certyfikatów w łańcuchu bloków. Oznacza to, że weryfikacja certyfikatów (osiąganie konsensusu) nie jest rozproszona i zależy wyłącznie od logiki (kodu) wykonywanej poza systemem łańucha bloków.

Autorzy wszystkich wymienionych wyżej publikacji proponują czyste podejście oparte na łańcuchu bloków. W pracy [218] zaproponowano zastosowanie standardu certyfikatu X.509v3 z wykorzystaniem rozszerzeń [24] dla informacji związanych z łańcuchem bloków. Dzięki temu certyfikat X.509 może być walidowany za pomocą klasycznego łańcucha zaufania opartego na urzędzie certyfikacji lub PKI opartym na łańcuchu bloków. Jak twierdzą autorzy jest to pierwsza propozycja wykorzystująca *„certyfikat hybrydowy”*. Jeżeli chodzi o to określenie, to trzeba zauważyć, że faktycznie jest to standardowa struktura certyfikatu X.509, w którym wykorzystano rozszerzenia zgodne z tym standardem.

W pracy [218] zaprezentowano propozycję platformy zarządzania infrastrukturą ***PKI na bazie łańcucha bloków***. Proponowana platforma umożliwia odwołanie certyfikatu, co jest rzeczywistym problemem w tradycyjnych systemach PKI. Ponadto nie jest możliwe usunięcie informacji z łańcucha bloków – jedynie nadrzędny urząd certyfikacji może zaznaczyć certyfikat wydany przez siebie jako „odwołany”. Dzięki temu wszelkie niewłaściwe zachowania urzędu certyfikacji w zakresie unieważniania certyfikatów będą śledzone i zauważane przez wszystkie inne podmioty.

W omawianym rozwiązaniu ([218]) wykorzystano nastepujące rozszerzenia certyfikatu X.509 do przechowania informacji:

***Subject Key Identifier*:** identyfikator klucza wystawcy – jest to tożsamość właściciela certyfikatu;

Blochain name: nazwa platformy łańcucha bloków;

***CA key identifier*:** identyfikator klucza urzędu certyfikacji, jeżeli jest to certyfikat urzędu certytfikacji, wówczas to pole jest puste;

***Issuer CA identifier*:** identyfikator wystawcy urzędu certyfikacji (adres inteligentnego kontraktu CA, który wydał ten certyfikat). Umożliwia on walidatorowi znalezienie w łańcuchu bloków kontraktów inteligentnych nadrzędnego CA i sprawdzenie, czy certyfikat o odpowiadającym mu skrócie został wydany i nie został unieważniony. Dla certyfikatów głównych pole to jest puste;

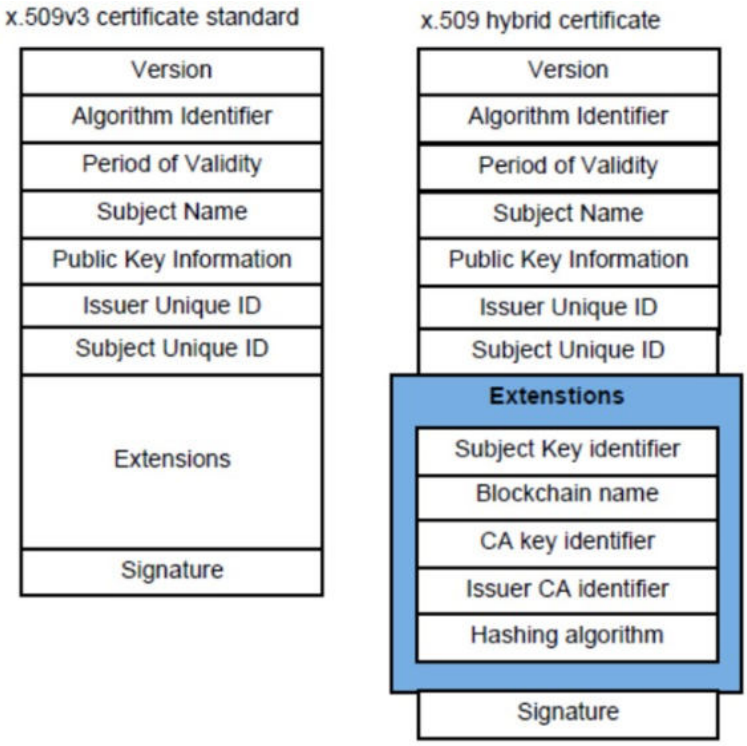
Hashing algorithm: algorytm skrótu zawierający informacje dotyczące algorytmu funkcji skrótu, który został użyty do obliczenia skrótu certyfikatu wczytanego do łańcucha bloków.

Na rysunku 6.20 pokazano zawartość „standardowego” certyfikatu X.509 oraz tzw. „ceryfikatu hybrydowego”.

Zgodnie z „klasycznym PKI”, w prezentowanym rozwiązaniu, na podstawie omówionych rozszerzeń X.509, można wyróżnić trzy rodzaje certyfikatów: certyfikat główny (*Root-CA*), ceryfikaty podrzędne (*Sub-CA*) oraz certyfikaty podmiotów (*Enduser*). W tabeli 6.1 pokazano te certyfikaty oraz relacje pomiędzy nimi. W pierwszym wierszu przedstawiono Root CA, w którym certyfikat jest wystawiany i podpisywany przez niego samego, a więc nie ma danych o *Issuer CA*, co pokazuje *ID Issuer CA* w piątej kolumnie (tj. 0x00000000). Certyfikat *Sub-CA* znajduje się w drugim wierszu i został on wydany przez *Root-CA*, a *ID Issuer-CA* wskazuje na *Root-CA*. W ostatnim wierszu znajduje się certyfikat użytkownika końcowego wystawiony przez *Sub-CA*.

W omawianym systemie PKI dane są przechowywane w dedykowanych inteligentnych kontraktach i zawierają one: tablicę z wartościami skrótów wydanych certyfikatów, opcjonalnie datę wygasnięcia ważności certyfikatu oraz inne dane techniczne. Ponadto do kontraktu dodawane są certyfikaty wystawców, co daje możliwość sprawdzenia całego drzewa certfikacji od liści do korzenia.

W warstwie dynamiki systemu, to znaczy realizacji działań, zaimplementowano naste-



Rys. 6.20: „Standardowy” i „hybrydowy” certyfikat X.509 (źródło: [218])

Certyfikat	Wydany przez	Wydany dla	ID kontraktu CA	ID CA wystawcy
RootCA	RootCA	RootCA	0x1234xxxx	0x0000000
SubCA	RootCA	SubCA	0x5631xxxx	0x0000000
EndUser	SubCA	EndUser	-	0x5631xxxx

Tabela 6.1: Hierarchia certyfikatów „hybrydowych” w łańcuchu bloków (źródło: [218])

pujące funkcjonalności PKI: zarejestrowanie certyfikatu, unieważnienie certyfikatu, stworzenie kontraktu, aktualizacja kontraktu, walidacja certyfikatów.

W tabeli 6.2 zestawiono przykłady różnych implementacji systemów infrastruktury klucza publiczngo opartych na łańcuchach bloków. Wynika z niej, że istnieją liczne rozwiązania w tym zakresie różniące się między sobą: typami, platformami na których je uruchomiono, modelami zaufania i konsensusu oraz sposobami przechowywania danych.

Tabela 6.2: Porównanie wybranych cech systemów PKI opartych na łańcuchach bloków (źródło: [148])

PKI	Kluczowe cechy	Typ łańcucha bloków	Platforma łańcucha bloków	Certyfikat	Model zaufania	Model konsensusu	Magazyn OFF	Magazyn ON	Złożoność czasowa
PA-PKI [199]	Identyfikacja i weryfikacja CA	z kontrolą[a]	Ethereum	niestandardowy	WoT	PBFT	Prywatny	Skrót	$O(n)$
BLOCK-CAM [201]	Krzyżowa weryfikacja CA	konsorcjalny z kontrolą[a]	Ethereum	X.509 v3	Hierarchiczny	–	Dane publiczne	Skrót + dane	$O(n^3)$
BC-TRUST [82]	Uwierzytelnianie	bez kontroli[b]	Ethereum	niestandardowy	WoT	–	Dane publiczne	Skrót + dane	–
BLOCK-PGP [217]	Kontrola dostępu do odwołań certyfikatów	z kontrolą[a]	Ethereum	X.509	WoT	PoW	Dane publiczne	–	–

kontynuacja tabeli na następnej stronie

Tabela 6.2 – kontynuacja tabeli z poprzedniej strony

PKI	Kluczowe cechy	Typ łańcucha bloków	Platforma łańcucha bloków	Certyfikat	Model zaufania	Model konsensusu	magazyn OFF	magazyn ON	Złożoność czasowa
PB-PKI [123]	Rejestr publiczny	bez kontroli[b]	niestandardowy	niestandardowy	WoT	–	Dane prywatne	Skrót	$O(n)$
TTA- SC [6]	Automatyzacja procesu identyfikacji błędnie skonfigurowanej strony CA	bez kontroli[b]	Ethereum	X.509	Hierarchiczny	–	Dane publiczne	Skrót + dane	$O(n)$
CERT-CHAIN [40]	Zaufanie CA zbudowane przy użyciu DCBF (Dual Counting Bloom Filter)	bez kontroli[b]	niestandardowy	X.509	Hierarchiczny	w oparciu o stopień niezawodności	Dane publiczne	Skrót	$O(n^2 log(n))$
CERT-LEDGER [110]	Transparentność certyfikatu	bez kontroli[b]	Ethereum	X.509	Hierarchiczny	PBFT	Dane publiczne	Skrót	$O(log(n))$
DB- PKI [190]	CA	bez kontroli[b]	niestandardowy	niestandardowy	WoT	PBFT	Dane publiczne	Skrót	$O(n^2)$
IKP [128]	Wiarygodne CA	bez kontroli[b]	Ethereum	X.509	Hierarchiczny	–	Publiczny	Skrót + dane	$O(nlog(n))$
FLY-CLIENT [34]	Lekki klient weryfikacji transakcji	z kontrola[a]	Ethereum	niestandardowy	Hierarchiczny	PoS	Publiczny	–	$O(log(n))$
BLOCK-QUICK [115]	Lekki klient weryfikacji transakcji	z kontrolą[a]	Ethereum	niestandardowy	WoT	PoPoW	Publiczny	–	$O(n)$

[a] w sieciach z kontrolą (*permissioned*) organizacja lub społeczność zarządzająca łańcuchem wskazuje użytkowników do walidacji bloków w ekosystemie, a niektóre z organizacji również kontrolują dostęp standardowych użytkowników
[b] w sieci bez kontroli dostępu (uprawnień) (*permissionless*) każdy może uczestniczyć i zostać walidatorem. Sieci blockchain bez uprawnień nie posiadają kontroli dostępu

Z powyższego zestawienia oraz przeglądu literatury (np. [218, 16, 15, 123, 19, 179, 96, 155, 187]) wynika, że prowadzone są intensywne prace oraz istnieją liczne propozycje systemów realizujących zadania systemów infrastruktury klucza publicznego. Należy zauważyć, że wykorzystanie technologii rozproszonych rejesetrów w systemach PKI na razie ogranicza się do prac studialnych związanych z konkretną technologią. Brakuje ustandaryzowanych, interoperacyjnych rozwiązań szeroko udostępniających podstawowe usługi PKI.

6.6. Podsumowanie

Rosnące zainteresowanie łańcuchami bloków jest spowodowane korzyściami jakie one dają w zastosowaniach w systemach rozproszonych. Zdecentralizowana natura łańcuchów bloków umożliwia i ułatwia implementację systemów wspomagania realizacji transakcji rozproszonych. Strukturę łańcuchów bloków oraz zasady ich funkcjonowania opisano w podrozdziale 6.2. Natomiast ich zalety, które przedstawiono w podrozdziale 6.3.1, pomimo wad opisanych w podrozdziale 6.3.2, spowodowały, że po pojawieniu się łańcuchów bloków zaczęły coraz liczniej powstawać nowe rozwiązania wspomagające rozproszone transakcje. Rozwiązania te nie wyeliminują centralnych systemów informatycznych, które wspomagają scentralizowane aktywności biznesowe i administracyjne, ale będą z nimi konkurować, zajmując obszary działalności pozostałych (rozproszonych) transakcji. W podrozdziale 6.1 przedstawiono klasyfikację i naturę systemów centralnych (6.1.1), zdecentralizowanych (6.1.2) i rozproszonych (6.1.3). Różne topologie sieci łańcuchów bloków oraz zróżnicowane stopnie centralizacji, opisane w podrozdziale 6.2, adekwatnie odzwierciedlają stopień scentralizowania-rozproszenia systemów, co ułatwia dobór topologii i stopnia scentralizowania łańucha bloków do natury konkretnej implementacji systemu wspomagającego realizację transakcji.

Potwierdzeniem tych szerokich możliwości są przykłady zastosowań łańcuchów bloków zaprezentowane w podrozdziale 6.5. Na szczególną uwagę zasługuje, przedstawiony w podrozdziale 6.5.2, przykład wykorzystania technologii łańcuchów bloków do uwierzytelniania zapisów medycznych. Jeszcze bardziej interesujący jest przykład zbudowania systemu Infrastruktury Klucza Publicznego w oparciu o łańcuch bloków, który zaprezentowano w podrozdziale 6.5.3. W systemie tym standardowe rozszerzenia certyfikatu X.509 wykorzystano do zapisu danych o łańcuchu bloków, które zawierają informacje o statusie certyfikatu i jego zmianach. Takie rozwiązanie eliminuje konieczność korzystania z centralnego systemu publikującego informacje o certyfikatach. System taki, będący pojedyńczym punktem awarii (*Single Point Of Failure*), zostaje zastąpiony przez rozproszony, odporny na ataki system oparty na łańcuchach bloków. Rozwiązanie to jest atrakcyjne ze względu na możliwość dalszego korzystania ze znanych i sprawdzonych struktur X.509, a jednocześnie ze wzgledu na korzystanie ze skrajnie zdecentralizowanej, bezpiecznej infrastruktury publikowania informacji o statusie certyfikatów. Pokazano również liczne implementacje infrastruktury PKI oparte na technologii łańcuchów bloków. Na obecnym etapie rozwoju zastosowań tych rozwiązań brak jest standardowych protokołów obsługi PKI, co stanowi pewną przeszkodę w ich stosowaniu ze względu na konieczność implementacji dedykowanych komponentów obsługi łańcuchów bloków.

7. Podsumowanie

Pojęcie transakcji może być definiowane na wiele różnych sposobów. Przegląd niektórych definicji zaprezentowano w podrozdziale 3.1, a przyjętą w rozprawie definicję transakcji podano również w słowniku (patrz def. 28).

Istotnym elementem towarzyszącym każdej transakcji jest informacja. Na tę informację składają się dane zawiązane z realizowaną transakcją, a bezpieczeństwo tych danych ma istotny wpływ na bezpieczeństwo samej transakcji. Informacja, oprócz funkcji nośnika danych skojarzonych z transakcją, może być również przedmiotem transakcji posiadającym określoną wartość rynkową, a nawet może bezpośrednio reprezentować wartość waluty (pieniądz elektroniczny). Obie te funkcje opisano w podrozdziale 3.2. Aby informacja rozumiana zarówno jako dane o transakcji, jak i przedmiot transakcji, mogła być stosowana skutecznie, bezpiecznie i efektywnie, musi posiadać cechy stanowiące o jej bezpieczeństwie takie jak niezaprzeczalność pochodzenia (patrz def. 16), integralność (patrz def. 13) i autentyczność-wiarygodność (patrz def. 29), a także interoperacyjność (patrz def. 14), której poświęcono rozdział 4. Cechy te opisano w podrozdziale 3.4. Zarówno definicje, jak i cechy dokumentów elektronicznych, zostały szczegółowo przedstawione w podrozdziale 3.5.

Na przestrzeni dziejów wykształciło się wiele rodzajów transakcji, które były przede wszystkim związane z gospodarczą działalnością człowieka i od początku zauważono konieczność ochrony transakcji w sferze wymiany informacji, co było przedmiotem rozważań w rozdziale 2. W rozdziale tym prześledzono w jaki sposób na przestrzeni wieków zmieniały się sposoby zabezpieczania stron transakcji – od tokenów sumeryjskich (podrozdział 2.2), poprzez różne formy pisma i techniki utrwalania informacji, do pieczęci i podpisów (podrozdział 2.4). Często zawarciu umowy towarzyszyły czynności rytualne, których zadaniem było również zapewnienie bezpieczeństwa transakcji. Już wówczas poziom zabezpieczenia transakcji dostosowywano do jej znaczenia – na przykład drobne transakcje związane z życiem codziennym nie były „zabezpieczane” w żaden sposób.

W rozdziale 3 zwrócono uwagę na różnorodność rodzajów transakcji, która jest przyczyną różnych wymagań dotyczących bezpieczeństwa informacji. Transakcje o niewielkiej wartości nie wymagają wyrafinowanych zabezpieczeń w przeciwieństwie do transakcji wartościowych. Dlatego właśnie te transakcje wykorzystują dokumenty jako bezpieczną postać informacji. Współczesne dokumenty w postaci elektronicznej, jak opisano w podrozdziale 3.2, dają możliwość automatycznego przetwarzania w heterogenicznym, rozproszonym środowisku technologicznym, jak pokazano w podrozdziale 3.4, co umożliwia realizację transakcji na najwyższych poziomach dojrzałości zaprezentowanych w podrozdziale 3.3. Wydaje się, że formularz elektroniczny, zaprezentowany w podrozdziale 3.6, stanowiąc swoisty interfejs pomiędzy ludźmi i systemami informatycznymi oraz będący jednocześnie bezpiecznym dokumentem elektronicznym, omówionym w podrozdziale 3.5, jest skutecznym i wydajnym środkiem gromadzenia, przetwarzania i wymiany informacji podczas realizacji transakcji. Zauważono, że technicznie zróżnicowane możliwości korzystania z formularzy elektronicznych wymagają zapewnienia wysokiego poziomu interoperacyjności w różnych obszarach – od interoperacyjności w aspekcie interoperacyjności danych,

tożsamości elektronicznej, usług, po interoperacyjność reguł, a nawet interoperacyjności kulturowej jak opisano w rozdziale 4. Jednym z podstawowych założeń dla dokumentów i formularzy umożliwiających zapewnienie tej szerokiej interoperacyjności jest przyjecie formatu XML jako podstawowego formatu dokumentów, którego przetwarzanie jest łatwe ze względu na proste reguły i powszechnie dostępne oprogramowanie.

Jak wspomniano powyżej jednym z najbardziej istotnych aspektów realizacji transakcji jest zapewnienie bezpiecznych środków wymiany informacji stosownie do potrzeb wynikających z charakteru transakcji, co zostało opisane w rozdziale 5. W odniesieniu do informacji elektronicznej, a konkretnie dokumentu elektronicznego, takim skutecznym środkiem zabezpieczającym jest podpis elektroniczny. W podrozdziale 5.1 wspomniano, że istnieje wiele formatów podpisu elektronicznego. Jednak ze względu na konieczność zapewnienia wysokiego poziomu interoperacyjności podpisu, rozumianej jako łatwość jego definiowania i przetwarzania (tworzenia i weryfikowania – podrozdział 5.2.3), a także naturalnej integracji z podpisywanymi dokumentami, oczywistym jest stosowanie podpisu w formacie XML omówionego w podrozdziale 5.2.

W obszarze stosowania podpisów XML zaobserwowano istnienie pewnych podatności, opisanych w podrozdziale 5.4. Podatności te dotyczą przetwarzania wiadomości SOAP i są związane ze strukturą podpisu XMLdSig opisaną w podrozdziale 5.2.1. Szczególną podatnością, związaną z zaawansowanym korzystaniem z podpisu w dokumentach oraz formularzach opartych na formacie XML i podpisywaniem fragmentów dokumentów, jest podatność opisana w podrozdziale 5.5. Polega ona na braku kontroli podpisywanych treści (fragmentów dokumenów) i w związku z tym możliwości modyfikacji tych treści bez naruszenia ważności podpisu XML. Istnienie tej podatności powoduje konieczność kontrolowania definicji zawartości podpisu zarówno podczas jego tworzenia, jak i weryfikacji.

Jak wielokrotnie stwierdzono istnieje olbrzymia różnorodność realizowanych transakcji. Z punktu widzenia stopnia zcentralizowania można wskazać transakcje o mniejszym lub większym stopniu scentralizowania – od scentralizowanych po rozproszone. Odpowiednio można również wskazać systemy wspomagające realizacje tych transakcji – od systemów zcentralizowanych, poprzez systemy zdecentralizowane, po systemy rozproszone, które omówiono odpowiednio w podrozdziałach 6.1.1, 6.1.2 i 6.1.3.

Obecnie większość rozwiązań systemów wspomagających realizację transakcji jest dedykowana dla transakcji zcentralizowanych i zdecentralizowanych. Są to systemy, w których na różne sposoby zapewniono bezpieczeństwo transakcji – najczęściej poprzez logowanie do konta systemu i rejestrowanie poszczególnych kroków transacji. Innym sposobem jest wykorzystanie mechanizmu jednokrotnego logowania (Single Sign On). O ile takie systemy w mniejszym lub większym stopniu zabezpieczają transakcje, o tyle brak jest zaawansowanych środków wspomagania realizacji transakcji w środowisku całkowicie rozproszonym. Powszechnie stosowane technologie w zakresie transakcji rozproszonych nie zapewniają oczekiwanych cech informacji towarzyszącej realizacji transakcji i mogą być one z powodzeniem stosowane tam, gdzie oczekiwany poziom bezpieczeństwa nie jest zbyt wysoki (transakcje o małej wartości). Aby umożliwić realizację transakcji o znacznych wartościach w całkowicie rozproszonym środowisku należy zapewnić możliwość korzystania z bezpiecznych form wymiany informacji jaką daje dokument i formularz elektroniczny oraz całkowicie rozproszone i bezpieczne sposoby przechowywania informacji. Taki sposób przechowywania i udostępniania różnego rodzaju dokumentów dają łańcuchy bloków.

Ze względu na zalety łańcuchów bloków, omówione w podrozdziale 6.3.1, znalazły one szerokie zastosowania, które przedstawiono w podrozdziałach 6.4 i 6.5. Różnoroność typów łańcuchów bloków, opisana w podrozdziale 6.2, zarówno w wymiarze *publiczny-prywatny*,

jak i stopnia centralizacji *centralne-rozproszone*, stwarza szerokie możliwości dostosowania rodzaju łańcucha do wymogów stawianych przez implementację konkretnego systemu z uwzględnienim stopnia jego dostępności oraz stopnia jego scentralizowania.

Zastosowania łańcuchów bloków mogą obejmować nie tylko wspomaganie realizacji typowych transakcji gospodarczych, jak przedstawiono w podrozdziale 6.5.1, ale również uwierzytelnianie różnego rodzaju zapisów, którego przykład podano w podrozdziale 6.5.2. Łańuchy bloków można nawet zastosować w implementacji lub w rozrzerzeniu systemów Infrastruktury Klucza Publicznego, jak opisano w podrozdziale 6.5.3.

Istnienie formularza elektronicznego o naturze trójwarstwowej (wizualizacja, dane i logika) umożliwa atrakcyjne i praktyczne zintegrowanie formularzy z łańcuchami bloków. W połączeniu tym formularz byłby „interfejsem" do łańcuchów bloków. Jego zadaniem byłoby wizualizowanie danych (zgodnie z logiką formularza) oraz interakcja (wymiana danych) formularza z łańcuchami bloków (zgodnie z logiką formularza). Jednym z efektów takiej integracji mogłaby być próba zabezpieczenia transakcji bez korzystania ze scentralizowanej infrastruktury PKI (patrz podsumowanie w podrozdziale 6.6). W ten sposób formularz sterując realizacją usługi zintegrowanej z transakcjami w łańcuchach bloków byłby elementem integrującym różne usługi z łańcuchami bloków, niezależnie od ich typu.

Biorąc pod uwagę powyższe rozważania można stwierdzić, że dokument elektroniczny w postaci formularza o wysokim poziomie interoperacyjności i bezpieczeństwa, poprzez włączenie do jego struktury warstwy logiki, może być skutecznie wykorzystany do realizacji transakcji w zróżnicowanym środowisku. W szczególności formularz taki powinien zapewnić:

— wysoki poziom bezpieczeństwa poprzez bezpieczne użycie podpisu elektronicznego charakteryzującego się interoperacyjnością,
— zaawansowaną interoperacyjność, która ma nie tylko umożliwić automatyczne przetwarzania dokumentu, ale także używanie formularza przez człowieka,
— implementację mechanizmów integracji z systemami zewnętrznymi - na przykład poprzez wykorzystanie interfejsów do usług sieciowych, poprzez wykorzystanie koncepcji warstwy logiki formularza (trzecia warstwa), oraz
— współpracę z łańcuchami bloków, co ma umożliwić wspomaganie realizacji transakcji w całkowicie rozproszonym środowisku bez korzystania z infrastruktury serwerowej, a także
— implementację wspomagania realizacji transakcji w zakresie sterowania jej przebiegiem.

Konstatacją powyższego są:
— przedstawiony w podrozdziale 8.1 *Cel Badawczy* oraz
— *Hipoteza Badawcza* przedstawiona w podrozdziale 8.2.

8. Cel i hipoteza badawcza

8.1. Cel badawczy

Opracowanie modelu transakcji rozproszonych sterowanych formularzami elektronicznymi.

8.2. Hipoteza badawcza

Zastosowanie formularza elektronicznego z własną logiką umożliwi realizację transakcji rozproszonych w heterogenicznym środowisku bez używania infrastruktury serwerowej i udziału scentralizowanej infrastruktury PKI.

9. Formularz elektroniczny

Formularz elektroniczny jest centralnym podmiotem pracy, a jego cechy są przedmiotem badań i weryfikacji hipotezy badawczej w kontekście realizacji transakcji w rozproszonym, heterogenicznym środowisku. Poniżej przedstawiono zaprojektowane przez autora podejście do tworzenia formularza elektronicznego, zaczynając od genezy jego powstania, poprzez podstawowe założenia, po opis jego struktury (formatu, składni) oraz funkcjonowanie w otoczeniu.

9.1. Geneza i założenia

Geneza: Formularz elektroniczny powstał jako propozycja rozwiązania dającego możliwość wspomagania usług (transakcji) elektronicznych na wyższych poziomach dojrzałości opisanych w podrozdziale 3.3 i przedstawionych na rysunku 3.1 jako poziom 3 i 4. Rozwiązanie to przewyższa istniejące technologie (omówione w podrozdziale 3.4) o możliwość zapewnienia skuteczności prawnej (bezpieczeństwa), a także możliwość eliminacji kosztownych, zcentralizowanych i dedykowanych rozwiązań sieciowych. Jednocześnie daje ono możliwość integracji z usługami sieciowymi w architekturze SOA, poczty elektronicznej itp.

Założenia: techniczna implementacja przedstawionej idei formularza została oparta na następujących założeniach:

— Formularz jest dokumentem elektronicznym, czyli posiada wszystkie cechy dokumentu elektronicznego omówione w podrozdziale 3.5 (oraz w pracach [204, 207]). Dzięki temu może on być bezpiecznie i skutecznie wykorzystywany w realizacji transakcji na wysokim poziomie dojrzałości usług elektronicznych (*interakcja dwustronna i transakcja*).

— Formularz elektroniczny musi posiadać wszystkie cechy tradycyjnego formularza papierowego, a w szczególności odzwierciedlać jego postać graficzną oraz możliwość wykorzystania formularza do wspomagania realizacji transakcji, w której bierze udział wiele stron (ludzie i usługi automatyczne).

— Formularz powinien być zrozumiały dla ludzi (stron transakcji), a jednocześnie musi być łatwy do interpretacji i przetwarzania przez rozmaite systemy informatyczne (wysoki poziom interoperacyjności). W ten sposób stanowi on swoisty interfejs użytkownika dla systemów informatycznych umożliwiający bezpieczne włączenie w realizację transakcji zarówno ludzi, jak i systemów informatycznych.

— Formularz powinien zawierać informacje o sposobie jego prezentacji tak, aby czytelnie odzwierciedlać stan realizacji transakcji stosownie do stanu danych w polach formularza.

— Formularz powinien zapewniać bezpieczeństwo w zakresie integralności i wiarygodności różnych, również rozdzielnych części formularza, odzwierciedlających etapy realizacji transakcji przez uczestniczące strony.

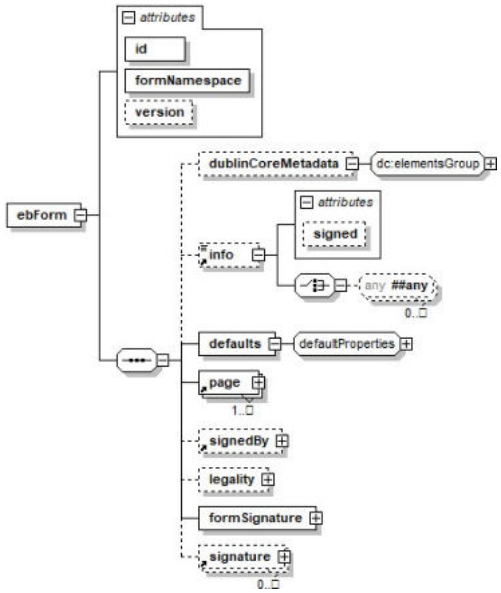
— Formularz powinien zapewnić, oprócz bezpieczeństwa danych, również bezpieczeństwo jego struktury (wzoru formularza), które może być rozumiane jako odpowiedzialność właścicieli i twórców za jego stworzenie oraz poprawne funkcjonowanie.
— W celu realizacji wymiany danych formularz powinien zawierać opis i wartości parametrów wykorzystywanych w mechanizmach komunikacji ze środowiskiem biznesowym, poprzez wykorzystanie integracji z usługami sieciowymi – np. SMTP lub SOAP.
— Formularz powinien umożliwić definiowanie logiki o możliwość obsługi specjalizowanych protokołów wysokiego poziomu takich jak: obsługa łańcuchów bloków i kryptowalut.
— Rozwiązania techniczne zastosowane w implementacji formularza i usług przez niego realizowanych muszą korzystać ze standardowych, powszechnie uznanych, komponentów i rozwiązań unikając niszowych, mało znanych i wąsko specjalizowanych rozwiązań.
— Powinna istnieć możliwość dalszej rozbudowy funkcjonalności logiki formularza o funkcje (akcje, działania) i sterowanie ich wykonaniem. Umożliwia to automatyczne wykonywanie zadań wchodzących w skład realizowanej transakcji w sposób „zaprogramowany” i zapisany w składni formularza.
— W przyszłości formularz powinien stanowić podstawę do budowania rozwiązań wspomagających technologię FinTech (łańcuchy bloków, kryptowaluty, inteligentne kontrakty itp.)

9.2. Struktura formularza

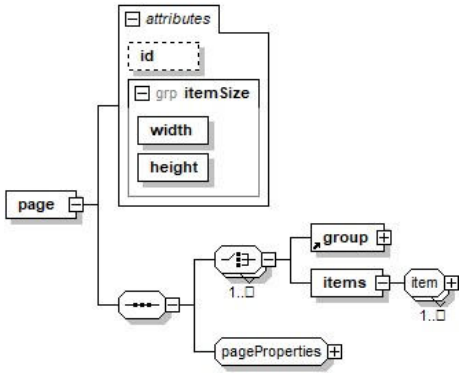
Aby zrealizować określone powyżej założenia i wymagania formularz elektroniczny musi posiadać określoną strukturę, którą autor pracy zaprojektował. Strukturę tę zaprezentowano na rysunku 9.1 (zgodnie ze schematami składni formularza zaprezentowanymi w podrozdziale A.1 Załącznika). Składa się ona z następujących elementów (rysunek 9.1a):

— ogólnych atrybutów formularza takich jak: identyfikator formularza, przestrzeń nazw i wersja składni,
— metadanych formularza zgodnych ze specyfikacją Dublin Core Metadata `<dublinCoreMetadata>`[1] (element opcjonalny),
— opcjonalnego elementu `<info>` o dowolnej składni, który zawiera informacje wymagane przez właścicieli transakcji wspomaganych przez formularz,
— obowiązkowego elementu `<defaults>`, który zawiera sekwencję struktur opisujących domyślne wartości atrybutów komponentów formularza. Wartości te są użyte wtedy, gdy nie są one zdefiniowane dla tego komponentu lub grupy komponentów (schemat struktury tego elementu oraz metadanych pokazano w Załączniku A na rysunku A.1–4),
— sekwencji struktur odzwierciedlających strony formularza `<page>`, co pokazano na rysunku A.1–3 w Załączniku A, na których są rozmieszczone opcjonalnie zgrupowane komponenty graficzne pokazane na rysunku 9.1,

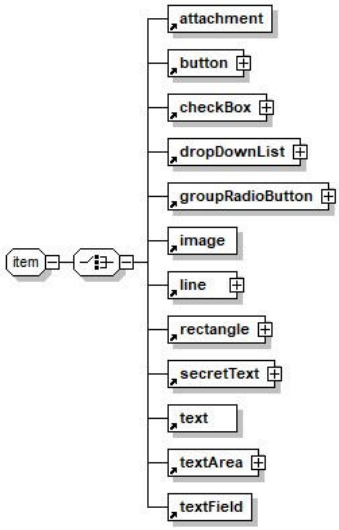
[1] Dublin Core Metadata (DC) jest to ogólny standard metadanych do opisu zasobów internetowych przyjęty jako standard ISO 15836-2003. DC definiuje 15 prostych elementów: data (`<date>`), format (`<format>`), identyfikator (`<identifier>`), język (`<language>`), opis (`<description>`), powiązanie (`<relation>`), prawa (`<rights>`), rodzaj (`<type>`), temat (`<subject>`), twórca (`<creator>`), tytuł (`<title>`), współtwórca (`<contributor>`), wydawca (`<publisher>`), zasięg (`<coverage>`), źródło (`<source>`).



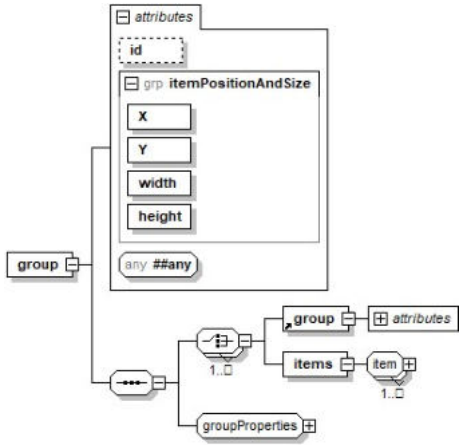
(a) Formularz



(b) Strona



(c) Komponenty graficzne



(d) Grupa

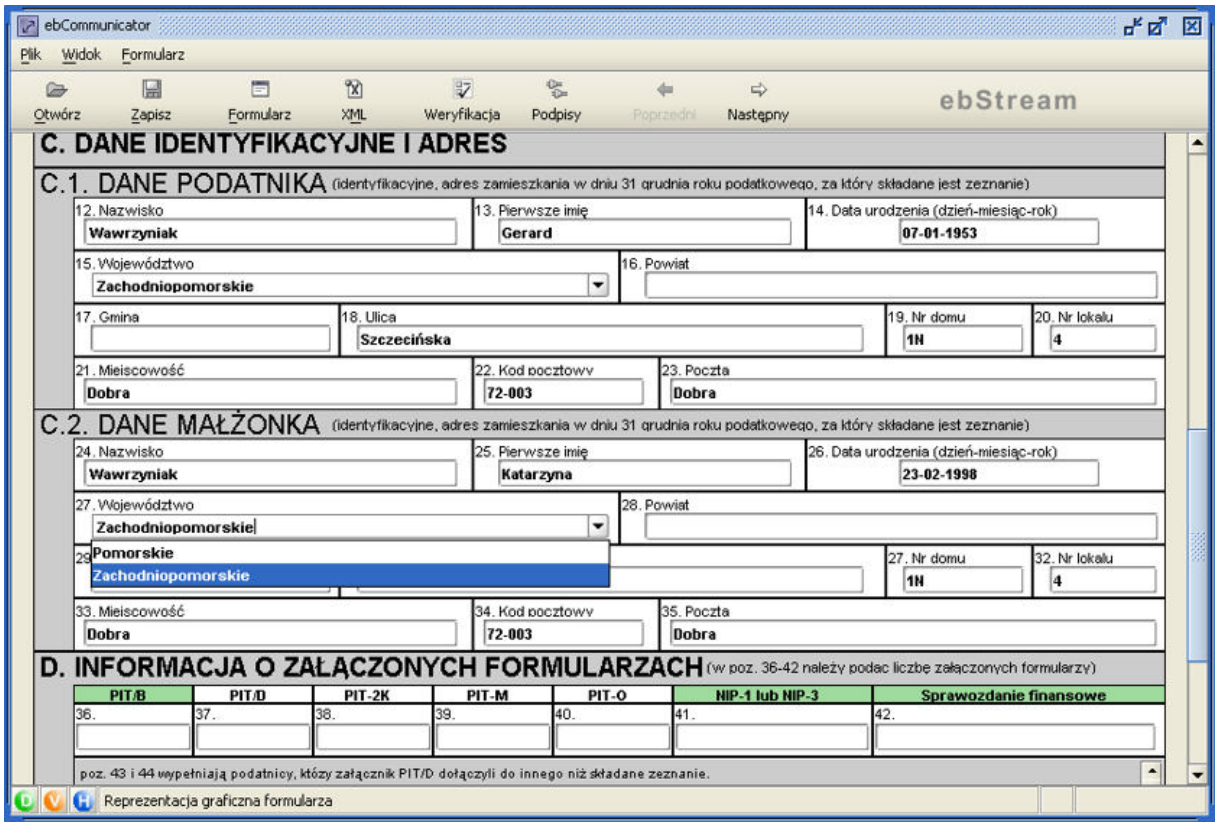
Rys. 9.1: Główne elementy struktury formularza (Załącznik rys. A.1–1, A.1–2)

— opcjonalnego elementu `<signedBy>` zawierającego co najmniej jednoelementową sekwencję elementów `<signer>` wskazujących identyfikatory podpisów podpisujących formularz,
— sekwencji czterech typów podpisów:
  1. opcjonalne podpisy środowiska prawnego transakcji,
  2. opcjonalne podpisy projektantów, wykonawców i organizacji będących właścicielami formularzy,
  3. obowiązkowy podpis zabezpieczający całą strukturę formularza,
  4. opcjonalne podpisy użytkowników formularza (stron transakcji).

Strona formularza, która odzwierciedla stronę tradycyjnego papierowego formularza, składa się z (rysunek 9.1b):
— ogólnych atrybutów strony: identyfikatora oraz wymiarów (szerokość i wysokość),

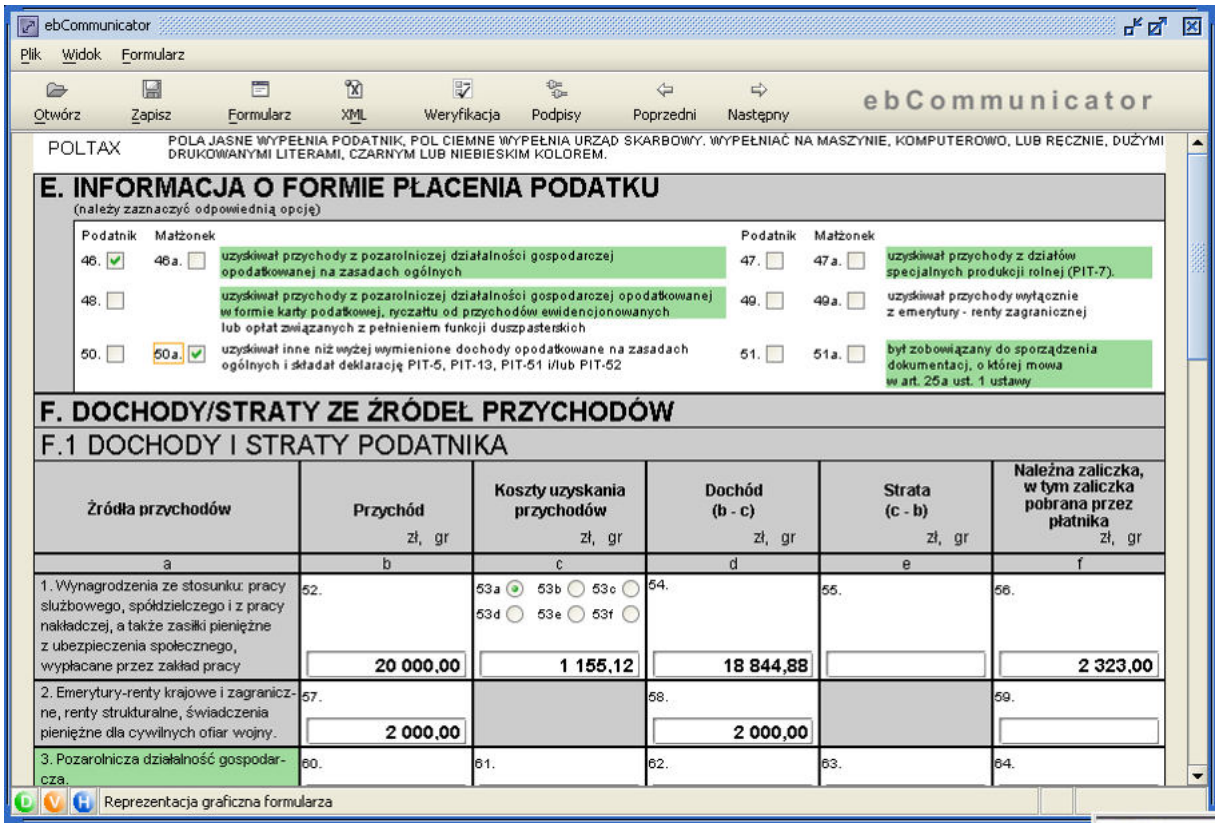
— sekwencji komponentów graficznych rozmieszczonych na stronie o atrybutach zdefiniowanych w ramach struktury każdego komponentu lub wskazanych w wartościach *<defaults>* w strukturze formularza,
— opcjonalnych grup zawierających (grupujących) komponenty graficzne oraz mogących zawierać inne grupy,
— definicji atrybutów strony, a w przypadku braku definicji, wartości atrybutów dla strony.

Komponenty graficzne formularza (rysunek 9.1c) są rozmieszczone na stronach i ewentualnie grupowane w grupach. Zaliczamy do nich: załącznik, przycisk, pole wyboru, lista rozwijalna, przyciski opcji, obraz, tekst, pole tekstowe, pole tajne oraz obszar tekstu. Struktura elementów graficznych formularza oraz ich atrybuty są przedstawione w postaci schematów w Załączniku A, rozdział A.2. Zadaniem komponentów graficznych jest odzwierciedlenie wizualnego aspektu dokumentu oraz gromadzenie, przechowywanie danych i weryfikacja ich poprawności. Komponenty graficzne mogą być grupowane tak, aby umożliwić definiowanie atrybutów dla zgrupowanych elementów, w tym również reguł poprawności. Istotnym elementem definiowania komponentów i grupy komponentów jest przyporządkowanie podpisu lub podpisów określonych użytkowników do poszczególnych komponentów lub ich grupy. Koncepcja grupowania komponentów graficznych ułatwia budowanie formularza, upraszcza jego odzwierciedlenie w strukturze formularza, a to z kolei ułatwia odzwierciedlenie wymogów wspomaganych transakcji.



Rys. 9.2: Komponenty graficzne formularza – napisy, listy rozwijalne, pola tekstowe

Na rysunkach 9.2 i 9.3 pokazano fragmenty formularza PIT-36. Z rysunków wynika, że elementy graficzne oraz ich atrybuty pozwalają na wierne odwzorowanie formularzy tradycyjnych, a jednocześnie wykorzystują specyficzne dla postaci elektronicznej komponenty, takie jak na przykład: lista rozwijalna, pola tekstowe, pola wyboru, przyciski



Rys. 9.3: Komponenty graficzne formularza – pola wyboru, przyciski opcji

opcji oraz pola z obliczeniami wartości dla bieżącego stanu danych w formularzu. Sposób wyświetlania i stan komponentów graficznych jest określany przez atrybuty statyczne oraz dynamiczne. Atrybuty te są podstawą do budowania warstwy logiki formularza (zostanie ona opisana w podrozdziale 9.3.3), dlatego poniżej zostaną one szczegółowo omówione.

9.2.1. Atrybuty statyczne

Atrybuty statyczne charakteryzują się tym, że ich wartości są definiowane podczas projektowania i budowy formularza oraz nie mogą one zmieniać swoich wartości podczas używania formularza w realizowanych transakcjach, czyli nie zależą od stanu danych wprowadzonych do formularza. Zestawienie komponentów graficznych ze wskazaniem ich atrybutów statycznych zamieszczono w tabeli 9.1.

Tabela 9.1: Statyczne atrybuty komponentów

Atrybuty statyczne	Komponenty													
	strona	grupa	napis	pole tekstowe	pole tajne	obszar tekstu	lista rozwijalna	przycisk wyboru	przyciski opcji	załącznik	obraz	przycisk	linia	prostokąt
identyfikator (opcjonalny)	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓
położenie (x, y)	✗	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✗	✓

kontynuacja tabeli na następnej stronie

Tabela 9.1 – kontynuacja tabeli z poprzedniej strony

Atrybuty statyczne	Komponenty													
	strona	grupa	napis	pole tekstowe	pole tajne	obszar tekstu	lista rozwijalna	przycisk wyboru	przyciski opcji	załącznik	obraz	przycisk	linia	prostokąt
wielkość (szer., wys.)	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✗	✓
poziome położenie tekstu	✗	✗	✓	✗	✓	✗	✗	✓	✗	✓	✓	✓	✗	✗
poziome wyrównanie tekstu	✗	✗	✓	✓	✓	✗	✗	✓	✗	✓	✓	✓	✗	✗
pionowe położenie tekstu	✗	✗	✓	✗	✗	✗	✗	✓	✗	✓	✓	✓	✗	✗
pionowe wyrównanie tekstu	✗	✗	✓	✗	✗	✗	✗	✓	✗	✓	✓	✓	✗	✗
algorytm funkcji skrótu	✗	✗	✗	✗	✓	✗	✗	✗	✗	✗	✗	✗	✗	✗
tekst pomocy	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✗	✗
tekst podpowiedzi	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✗	✗
wskaznie podpisujących	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✗	✗
atrybut (##any)	✗	✓	✗	✗	✗	✗	✗	✗	✗	✗	✗	✗	✗	✗
maksymalna długość tekstu	✗	✗	✗	✗	✗	✓	✗	✗	✗	✗	✗	✗	✗	✗
rozszerzenia nazw plików	✗	✗	✗	✗	✗	✗	✗	✗	✗	✓	✓	✗	✗	✗
nazwa pliku	✗	✗	✗	✗	✗	✗	✗	✗	✗	✗	✓	✗	✗	✗
współrzędne punktów	✗	✗	✗	✗	✗	✗	✗	✗	✗	✗	✗	✗	✓	✗

9.2.2. Atrybuty dynamiczne

Atrybuty dynamiczne są to atrybuty, których wartość zależy od stanu danych (pól) w formularzu zgodnie ze zdefiniowaną logiką. Pozwala to na zaprojektowanie formularza w taki sposób, aby prezentacja formularza i właściwości jego komponentów graficznych (dostępność, edytowalność, kolory, reguły poprawności) były adekwatne do stanu danych w formularzu zgodnie z wymogami wspomaganej przez formularz transakcji. Dzięki temu formularz razem z jego komponentami może nie tylko odzwierciedlać aktualny, zmieniający się stan realizacji transakcji, ale również umożliwia zachowanie poprawności wprowadzanych danych poprzez wykorzystanie dedykowanego atrybutu dynamicznego `<isValid>`. W tabeli 9.2 zaprezentowano komponenty graficzne i ich atrybuty dynamiczne. Atrybuty dynamiczne w składni komponentów formularza pokazano w Załączniku A na rysunku A.1–1 (na poziomie ogólnej struktury formularza) oraz na rysunkach w podrozdziale A.2 (poszczególne komponenty graficzne).

Tabela 9.2: Dynamiczne atrybuty komponentów

Atrybuty dynamiczne	Komponenty													
	strona	grupa	napis	pole tekstowe	pole tajne	obszar tekstu	lista rozwijalna	przycisk wyboru	przyciski opcji	załącznik	obraz	przycisk	linia	prostokąt
tło	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✗	✗
obramowanie	✓	✓	✓	✓	✓	✓	✓	✗	✓	✓	✓	✓	✗	✗
czy edytowalny	✗	✗	✗	✓	✓	✓	✗	✓	✗	✗	✗	✗	✗	✗

Tabela 9.2 – kontynuacja tabeli z poprzedniej strony

Atrybuty dynamiczne	Komponenty													
	strona	grupa	napis	pole tekstowe	pole tajne	obszar tekstu	lista rozwijalna	przycisk wyboru	przyciski opcji	załącznik	obraz	przycisk	linia	prostokąt
czy aktywny	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✗
czy widoczny	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✗	✗
reguły poprawności	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✗	✗
napis dynamiczny	✗	✗	✓	✗	✗	✗	✗	✓	✗	✓	✓	✓	✗	✗
ikonka	✗	✗	✓	✗	✗	✗	✗	✓	✗	✓	✓	✓	✗	✗
wartość domyślna	✗	✗	✗	✓	✗	✓	✓	✓	✓	✗	✓	✗	✗	✗
wartość parametryzacji form.	✗	✗		✓	✗	✓	✓	✓	✓	✓	✓	✗	✗	✗
formatowanie wyświetlania	✗	✗		✓	✗	✗	✗	✗	✗	✗	✗	✗	✗	✗
czcionka	✗	✗	✗	✓	✓	✓	✓	✓	✗	✓	✓	✓	✗	✗
kolor czcionki (foreground)							✓	✓	✗	✓	✓	✓	✗	✗
kolor	✗	✗	✗	✗	✗	✗	✗	✗	✗	✓	✗	✗	✓	✓
grubość	✗	✗	✗	✗	✗	✗	✗	✗	✗	✗	✗	✗	✓	✓
kolor podświetlenia (sel. text)	✗	✗	✗	✓	✓	✓	✗	✗	✗	✗	✗	✗	✗	✗
kolor wyboru (selection)	✗	✗	✗	✓	✓	✓	✗	✗	✗	✗	✗	✗	✗	✗
kolor tekstu nieaktywnego	✗	✗	✗	✓	✓	✓	✗	✗	✗	✗	✗	✗	✗	✗
definiowanie akcji	✗	✗	✗	✗	✗	✗	✗	✗	✗	✗	✗	✓	✗	✗

Znaczenie poszczególnych atrybutów dynamicznych ze wskazaniem struktur ich definicji jest przedstawione w podrozdziale A.3 Załącznika A. Natomiast na rysunku 9.4 pokazano użycie takich komponentów formularza jak obraz i przycisk, a także użycie dynamicznych atrybutów pól formularza i napisów do formatowania danych wprowadzanych i wyświetlanych.

9.2.3. Typy przechowywanych danych

Podstawowym zadaniem formularza jest gromadzenie i przechowywanie danych w dedykowanych komponentach graficznych. Typy przechowywanych danych oraz komponenty, w których są one gromadzone, przedstawiono w tabeli 9.3. Natomiast miejsce przechowywanych danych w strukturze formularza zaprezentowano w Załączniku A, na rysunku A.6–1b dla typu logicznego, A.7–1b dla tekstu, A.8–1b dla wartości dziesiętnych, A.9–1b dla wartości całkowitych, A.10–1b dla wartości daty, A.11–1b dla obiektów binarnych.

Tabela 9.3: Typy danych przechowywanych w komponentach

Typ danych	Komponenty													
	strona	grupa	napis	pole tekstowe	pole tajne	obszar tekstu	lista rozwijalna	przycisk wyboru	przyciski opcji	załącznik	obraz	przycisk	linia	prostokąt
tekst	✗	✗	✗	✓	✓	✓	✓	✗	✗	✗	✗	✗	✗	✗

kontynuacja tabeli na następnej stronie

Tabela 9.3 – kontynuacja tabeli z poprzedniej strony

Typ danych	Komponenty													
	strona	grupa	napis	pole tekstowe	pole tajne	obszar tekstu	lista rozwijalna	przycisk wyboru	przyciski opcji	załącznik	obraz	przycisk	linia	prostokąt
wartość dziesiętna	✗	✗	✗	✓	✗	✗	✗	✗	✗	✗	✗	✗	✗	✗
wartość całkowita	✗	✗	✗	✓	✗	✗	✗	✗	✓	✗	✗	✗	✗	✗
wartość logiczna	✗	✗	✗	✓	✗	✗	✗	✓	✗	✗	✗	✗	✗	✗
data - czas	✗	✗	✗	✓	✗	✗	✗	✗	✗	✗	✗	✗	✗	✗
obiekty binarne	✗	✗	✗	✗	✗	✗	✗	✗	✗	✓	✓	✗	✗	✗

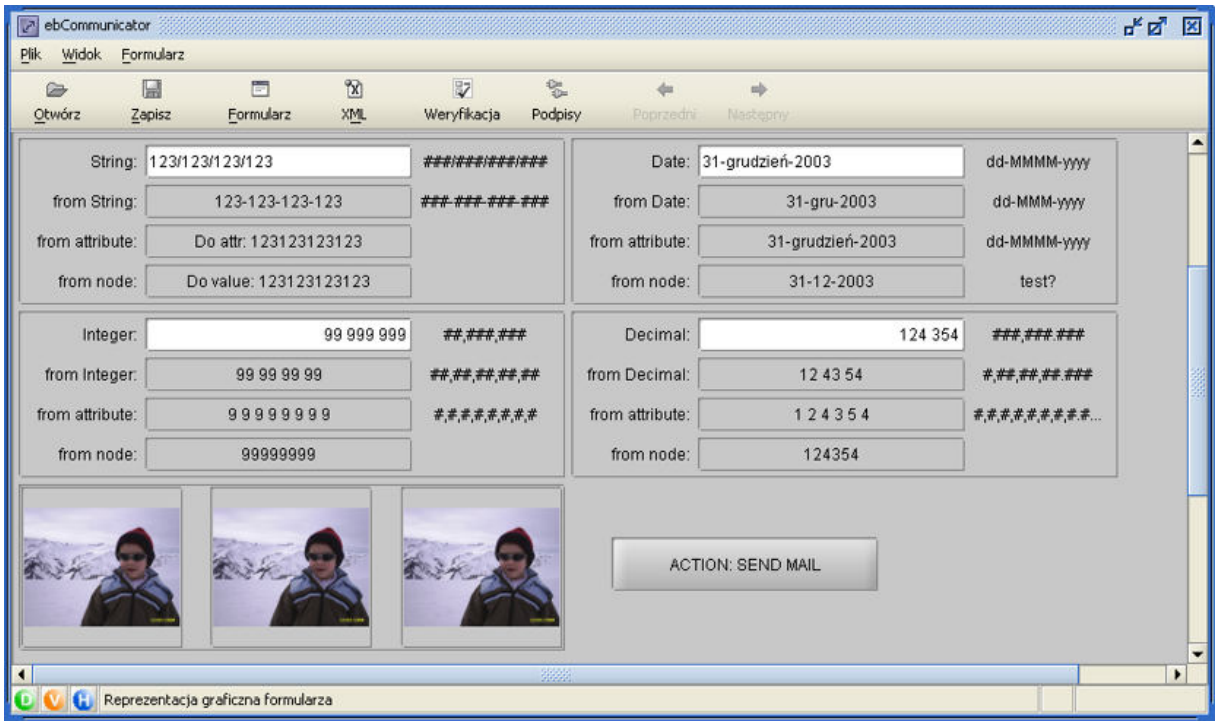
Wskazane w tabeli 9.3 komponenty dedykowane do przechowywania danych określonych typów ułatwiają kontrolę ich poprawności i sposobu ich interpretowania.

9.2.4. Definiowanie akcji (działań)

Nowym elementem, specyficznym dla formularza elektronicznego, jest możliwość korzystania z działań („akcji") definiowanych dla komponentu graficznego *Przycisk* (`<Button>`). Działania te są wykonywane po naciśnięciu tego komponentu przez użytkownika. Dzięki temu są one mocnym narzędziem pozwalającym wpływać na przebieg realizowanej transakcji. Pozwala to także budować formularze, które są intuicyjne, łatwe do interpretacji i użytkowania. W tabeli 9.4 zestawiono wszystkie akcje ze wskazaniem schematu ich struktury.

Tabela 9.4: Zestawienie „akcji" formularza ze wskazaniem schemtów struktur

Akcja	Schemat w Załączniku A
działanie puste	A.12–2
wywołanie komendy systemu operacyjnego	A.12–3
wyświetlenie strony formularza o wskazanym numerze	A.12–4
zapisanie formularza na dysku	A.12–5
otwarcie wskazanego pliku	A.12–6
otwarcie formularza	A.12–7
wyświetlenie obiektu binarnego	A.12–8
wykonanie wskazanego podpisu	A.12–9
usunięcie wskazanego podpisu	A.12–10
wyświetlenie informacji o podpisie	A.12–11
wyświetlenie informacji o certyfikacie	A.12–12
wyświetlenie deklaracji podpisu	A.12–13
wykonanie GUUID	A.12–14
wprowadzenie do komponentu wartości domyślnej	A.12–15
weryfikacja poprawności danych w komponencie	A.12–16
wykonanie akcji wskazanych przez użytkownika	A.12–17
generowanie certyfikatu X.509	A.12–18a
generowanie listy CRL	A.12–19
zbudowanie i wysłanie mail'a	A.12–20
wywołanie usługi SOAP	A.12–21



Rys. 9.4: Komponenty graficzne formularza – formatowanie danych wprowadzanych i wyświetlanych, umieszczanie obrazów, przycisk uruchamiający „akcję”

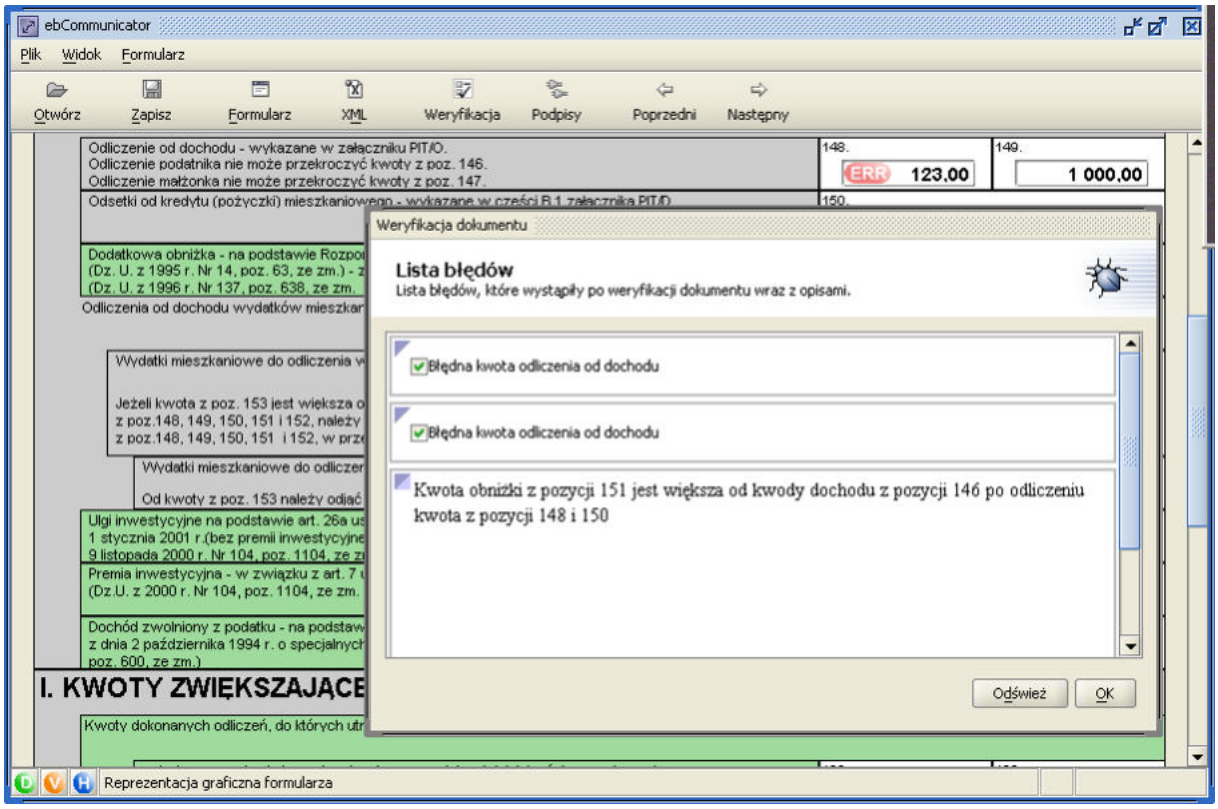
Na rysunku 9.5 zaprezentowano na przykładowym formularzu efekt wykonania „akcji” polegającej na wyświetleniu wyniku weryfikacji poprawności danych w komponentach, które są wskazane w „akcji” przyporządkowanej do naciśniętego przycisku.

Wśród „akcji” na szczególną uwagę zasługują działania zewnętrzne takie jak wysłanie i odebranie wiadomości SOAP oraz wysłanie wiadomości poczty elektronicznej, gdyż są one podstawą do budowania aktywnych/dynamicznych relacji ze światem zewnętrznym. Składnia wyrażeń dla tych „akcji” umożliwia zbudowanie sekwencji czynności, których argumentami są wyrażenia dowolnego typu, odwołujące się do pól formularza.

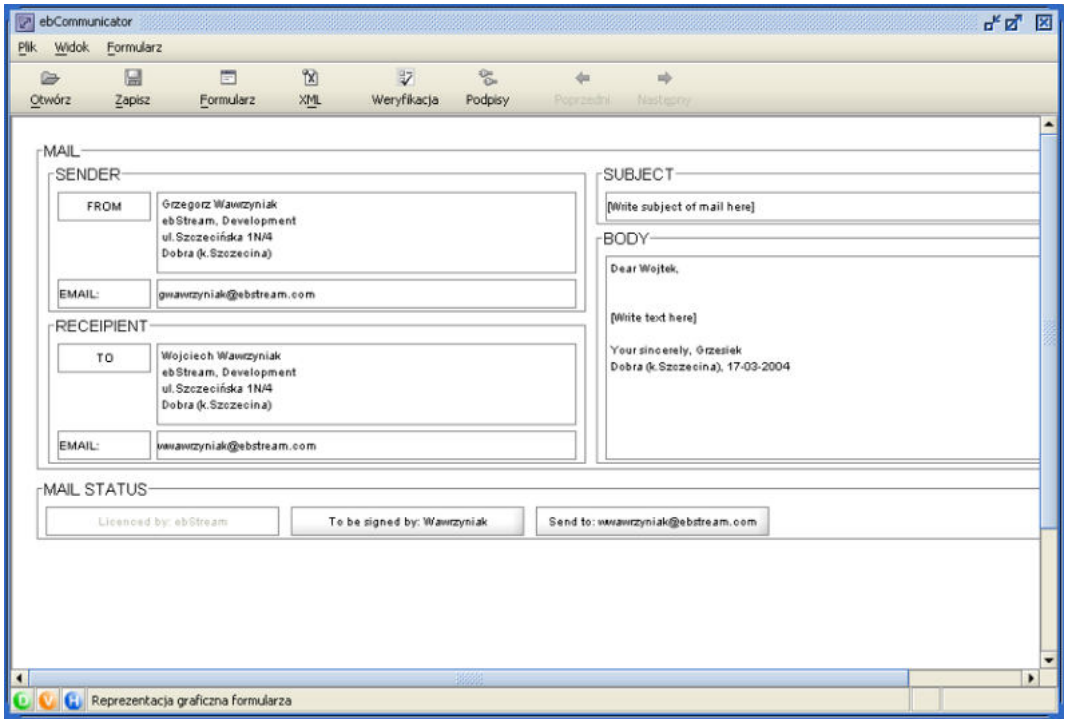
Komponenty graficzne formularza elektronicznego, możliwość definiowania atrybutów dynamicznych oraz możliwość definiowania „akcji” pozwalają na tworzenie bardziej zaawansowanych, „zaprogramowanych” formularzy. Przykładem takiego formularza jest formularz przedstawiony na rysunku 9.6, który jest przeznaczony do zbudowania i wysłania zaszyfrowanej oraz podpisanej wiadomości poczty elektronicznej z użyciem „akcji” przyporządkowanej dla *Przycisku* i definiowanej w strukturze `<mail>`. Struktura definicji tej „akcji” jest pokazana na rysunku A.12–20 w Załączniku A.

9.2.5. Bezpieczeństwo formularza – podpisy

Z założenia formularz elektroniczny posiada wszystkie cechy dokumentu. To z kolei implikuje wymóg zapewnienia bezpieczeństwa danych poprzez zastosowanie podpisów elektronicznych. Ponieważ formularz elektroniczny ma odzwierciedlać tradycyjny formularz papierowy, należy zapewnić możliwość podpisywania różnych części formularza przez różne strony biorące udział w realizacji transakcji. W związku z tym podpis zastosowany w formularzu musi:



Rys. 9.5: Weryfikacja poprawności danych w formularzu w wyniku naciśnięcia przycisku



Rys. 9.6: Budowanie wiadomości pocztowej – umieszczanie załączników, wykorzystanie funkcji tekstowych, grupowanie komponentów, dynamiczne wartości atrybutów (w przykładzie załącznikami są certyfikaty klucza publicznego oraz funkcje tekstowe na tych certyfikatach)

1. **Zapewnić możliwości kształtowania takiego zakresu informacji o podpisie, który zapewni optymalny poziom zaufania do podpisanego dokumentu i dostosowanie tych informacji do wymogów realizowanej transakcji**
   W formularzu istnieją cztery typy podpisów (Załącznik A rys. A.4). W celu zapewnienia wygodnego i skutecznego korzystania z podpisów dla każdego z nich określono zbiór atrybutów, które wymieniono w tabeli 9.5. Atrybuty te pozwalają na spełnienie dodatkowych, nienormowanych wymagań organizacji realizującej (realizujących) określoną transakcję. Dane zapisane w atrybutach, chociaż dotyczą podpisu, nie zawsze są elementami struktury podpisu w rozumieniu specyfikacji XMLdSig (W3C). Informacjami (atrybutami) o podpisie są:
   — *identyfikator* – identyfikuje strukturę podpisu w formularzu,
   — *nazwa* – nazwa podpisu pokazywana użytkownikom przez aplikację,
   — *URL zasobu* – adres podpisywanego zasobu odłączonego,
   — *usuwalnosc* – atrybut opcjonalny informujący o tym, czy funkcja usuwania podpisu za pomocą aplikacji jest dostępna,
   — *wymogi certyfikatu* – opcjonalna informacja o tym, jaki certyfikat ma być wykorzystany do generowania i weryfikowania podpisu („wymagany kwalifikowany", „pożądany kwalifikowany", „dowolny"),
   — *typ filtrowania* – opcjonalne wskazanie sposobu selekcji węzłów do podpisu („XPath" – wykorzystanie wyrażeń XPath do selekcji węzłów do podpisu, „ebForm" – wykorzystanie algorytmu dedykowanego do wykonywania podpisów zaimplementowanego w aplikacji użytkownika),
   — *szablon referencji* – opcjonalny atrybut zawierający wyrażenie, które ma być użyte do tworzenia i weryfikacji podpisu w celu sprawdzenia poprawności referencji w podpisie XML,
   — *deklaracja podpisu* – opcjonalny tekst zawierający formalną informację o roli podpisu w formularzu, który jest wyświetlany w aplikacji użytkownika,
   — *data i czas podpisu* – opcjonalne wskazanie dołączenia do podpisu czasu wykonania podpisu na podstawie danych otrzymanych z serwera sieciowego protokołu czasu (NTP),
   — *lista kontrasygnujacych* – opcjonalny element zawierający sekwencję referencji do innych podpisów w formularzu kontrasygnujących podpis,
   — *dane biometryczne* – opcjonalna informacja o konieczności dołączenia do podpisu danych biometrycznych o podpisie odręcznym i obrazie osoby wskazanej w certyfikacie podpisu,
   — *informacje o kluczu* – opcjonalna informacja o kluczu publicznym podpisującego w celu niezaprzeczalnego wskazania osób uprawnionych do wykonania podpisu.

Tabela 9.5: Atrybuty podpisów

Atrybuty	Podpisy			
	zasobów	zatwierdzające	formularza	użytkowników
schemat w Załączniku A	A.4–2	A.4–3	A.4–4	A.4–5
identyfikator	✓	✓	✓	✓

kontynuacja tabeli na następnej stronie

Tabela 9.5 – kontynuacja tabeli z poprzedniej strony

Atrybuty	Podpisy			
	zasobów	zatwierdzające	formularza	użytkowników
nazwa	✓	✓	✓	✓
URL zasobu	✓	✗	✗	✗
usuwalność	✓	✓	✓	✓
wymogi certyfikatu	✓	✓	✓	✓
typ filtrowania	✗	✓	✓	✓
szablon referencji	✓	✓	✓	✓
deklaracja podpisu	✓	✓	✓	✓
data i czas podpisu	✓	✓	✓	✓
lista kontrasygnujących	✗	✗	✗	✓
dane biometryczne	✗	✗	✗	✓
informacja o kluczu	✓	✓	✓	✓

Podczas definiowania atrybutów podpisów można wskazać uprawnienia do wykonania podpisów poprzez zdefiniowanie ograniczeń dla cech certyfikatu klucza publicznego jaki ma być użyty do tworzenia i weryfikacji podpisu. Funkcjonalność aplikacji użytkownika powinna uniemożliwić wykonanie podpisu z użyciem certyfikatu nie spełniającego warunków zdefiniowanych w formularzu. Poprzez porównanie cech certyfikatu z ograniczeniami zdefiniowanymi dla danego podpisu w logice formularza możliwe jest łatwe wykrycie nadużycia w tym obszarze.

2. **Zapewnić bezpieczeństwo całej struktury formularza tak, aby uniemożliwić modyfikację jego struktury nie objętej podpisami użytkowników, a jednocześnie zapewnić niezaprzeczalność autorstwa lub własności formularza**
   Bezpieczeństwo formularza z wykorzystaniem technologii podpisów elektronicznych jest zapewniane w takich obszarach jak:
   Środowisko prawne – poprzez wskazanie zasobów sieciowych (teksty instrukcji, procedur, rozporządzeń ustaw itd.) i objęcie ich podpisem „odłączonym". W formularzu podpis obejmujący ten obszar określono jako **podpis zasobów**. Jego zawartością jest wskazany zasób i jest on wykonywany przez projektanta lub właściciela formularza.
   Środowisko organizacyjne wspomagające transakcję – poprzez wskazanie (na etapie projektowania) i podpisanie (na etapie wdrażania) całej struktury formularza z uwzględnieniem *kontrasygnaty podpisów zasobów* o ile istnieją. Są to **podpisy zatwierdzające**. Podpisy te nie obejmują tych miejsc w strukturze, w których są gromadzone i przechowywane dane.
   Struktura formularza – poprzez wykorzystanie tzw. **podpisu formularza**, który zabezpiecza przed manipulacjami w strukturze formularza. Zawartość tego podpisu (podpisywana treść) jest taka sama jak *podpisów zatwierdzających*, ale dodatkowo kontrasygnuje istniejące podpisy (podpisy zasobów, podpisy zatwierdzające).
3. **Zapewnić możliwość objęcia podpisem różnych, nawet rozłącznych, fragmentów struktury formularza oraz kontrasygnowania podpisów**
   Miejsce podpisów w strukturze formularza pokazano na rysunku A.4–1 w Załączniku A. Wśród statycznych atrybutów komponentów (tabela 9.1) znajduje się atrybut

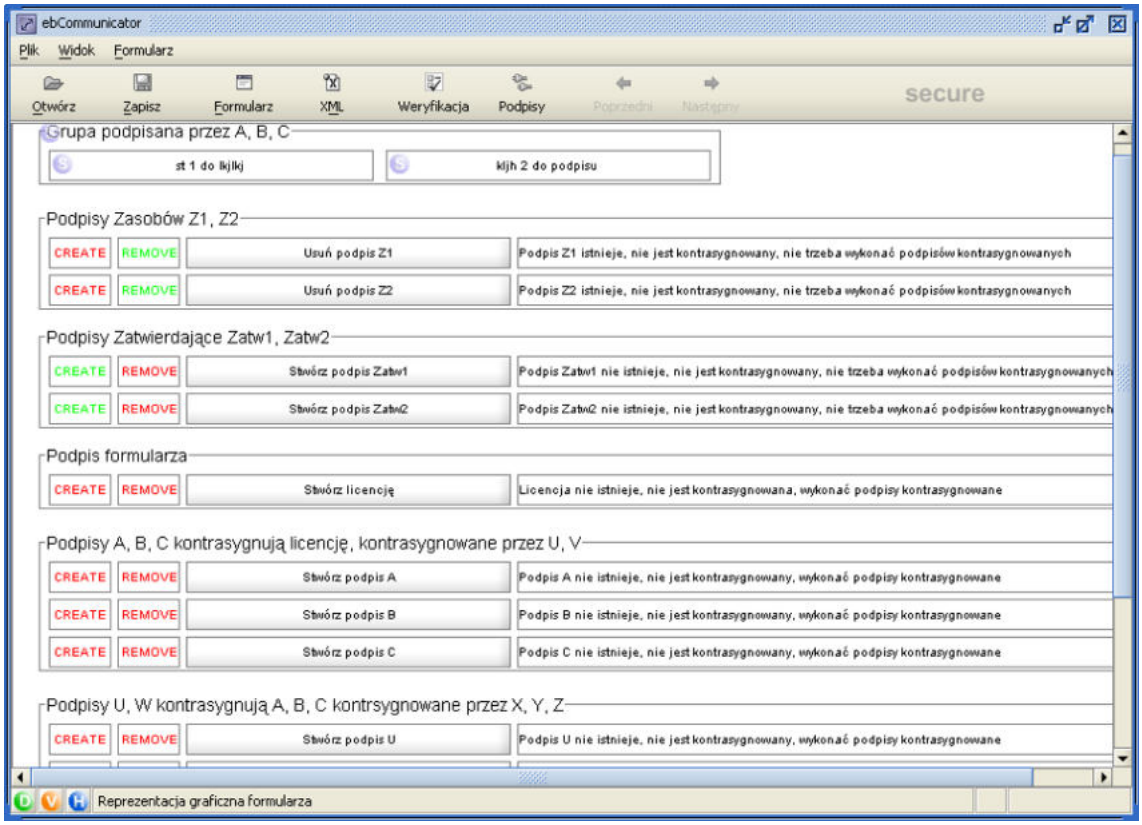
`<signedBy>`, posiadający strukturę pokazaną na rysunku A.3–20 w Załączniku A. Jest to atrybut wszystkich komponentów przedstawionych w tabeli 9.6 i zawiera w strukturach *signer* sekwencję wskazań do podpisów użytkowników.

Tabela 9.6: Referencje do podpisów w komponentach

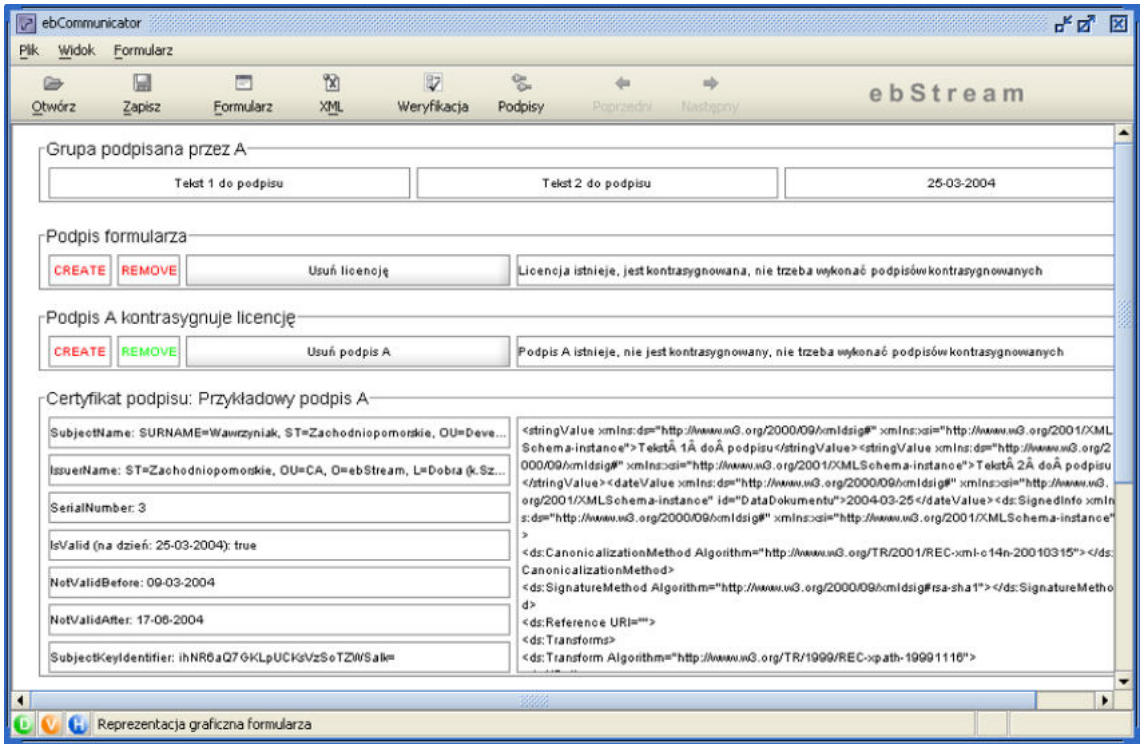
Komponent	Schemat w Załączniku A
formularz	9.1a
strona	A.4–1
grupa	A.1–3
przycisk	A.2–1b
przycisk wyboru	A.2–2b
lista rozwijalna	A.2–3b
obszar tekstu	A.2–4b
pole danych	A.2–6f
przycisk opcji	A.2–7d
grupa przycisków opcji	A.2–7b
obraz	A.2–8d
załącznik	A.2–9d

Dzięki temu powstaje możliwość podpisywania różnych części struktury formularza przez wiele wskazanych podpisów, a każdy podpis może w swojej treści zawierać wiele wskazanych komponentów. Jest to jedna z podstawowych własności formularza elektronicznego, która pozwala na podpisywanie różnych części formularza przez różnych użytkowników (stron transakcji). Podpisy te są nazwane **Podpisami użytkowników**. Ponadto każdy podpis użytkownika posiada atrybut `<countersignedBy>`, na który składa się, podobnie jak w przypadku komponentów graficznych, sekwencja wskazań na podpisy kontrasygnujące.

Zaprezentowane podejście do stosowania podpisów elektronicznych umożliwiło projektowanie i tworzenie formularzy elektronicznych odzwierciedlających rzeczywiste potrzeby realizacji transakcji. Na rysunkach 9.7 i 9.8 pokazano formularze, których zadaniem jest prezentacja możliwości związanych z obsługą podpisów.



Rys. 9.7: Przykład wykorzystania w formularzu interaktywnej obsługi podpisu funkcji i akcji związanych z podpisem



Rys. 9.8: Użycie funkcji operujących na certyfikatach podpisu oraz prezentacja zawartości podpisanych danych

Podpisy te są generowane przez aplikację stworzoną do korzystania z formularza na podstawie zależności zapisanych w jego logice. Budowanie struktury podpisu uwzględnia stworzenie definicji zawartości podpisu (Signature Content) z pomocą wyrażenia XPath. Na wydruku 9.1 zaprezentowano element `<XPath>` zawierający wygenerowane automatycznie wyrażenie XPath definiujące zawartość podpisu. W tym przykładzie pokazano wyrażenie XPath dla podpisu identyfikowanego jako *MANAGER_SIGNATURE* (zaznaczone jako *bold*, *underline*).

```
<ds:Reference URI="">
<ds:Transforms>
  <ds:Transform Algorithm="http://www.w3.org/TR/1999/REC-xpath-19991116">
      <ds:XPath>
(ancestor::*/parent::value or ancestor-or-self::ds:Signature/ancestor::
signature//parent::ebForm or ancestor::content/ancestor::attachment or
ancestor::*/ancestor::*/ancestor::mailStatus or ancestor::*/ancestor::
value/ancestor::soap-RPC or ancestor::body/ancestor::request or ancestor::
header/ancestor::request or ancestor::body/ancestor::response)or
(ancestor::*/parent::value/ancestor-or-self::*/signedBy/signer[text()=
'PodpisInteresanta'] or ancestor-or-self::ds:Signature/ancestor::
signature/countersignedBy/signer[text()='MANAGER_SIGNATURE'] )
      </ds:XPath>
  </ds:Transform>
  <ds:Transform
    Algorithm="http://www.w3.org/TR/2001/REC-xml-c14n-20010315"/>
  </ds:Transforms>
  <ds:DigestMethod Algorithm="http://www.w3.org/2000/09/xmldsig#sha1"/>
<ds:DigestValue>iWpdLjTj80sBb79diU9J50HHTc4=</ds:DigestValue>
</ds:Reference>
```

Wydruk 9.1: Przykład elementu `<Reference>` w strukturze podpisu XMLdSig z elementem XPath wyboru fragmentów do podpisu

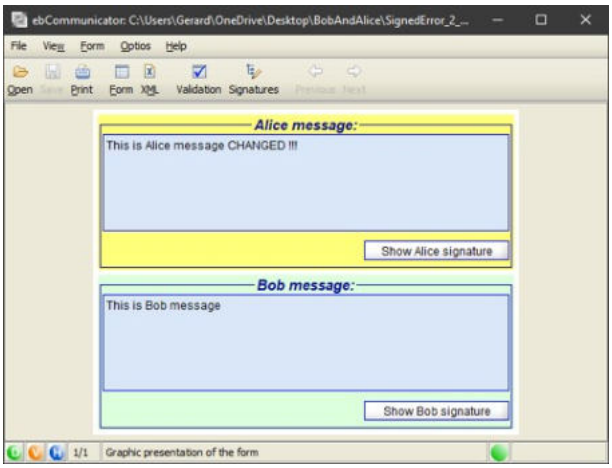
Istotnym atrybutem podpisu, wymienionym w tabeli 9.5, jest tzw. *Szablon Podpisu*. Atrybut ten przechowuje poprawną wartość wyrażenia XPath definiującego zawartość danego podpisu. Szablon ten jest wykorzystywany do eliminacji podatności (omówionej w podrozdziale 5.5), która polega na możliwości manipulacji „pozornie podpisanej" treści poprzez zmianę definicji referencji w podpisie. W artykule [206] autor zaproponował zastosowanie *Szablonu Podpisu* w postaci bezpiecznego (podpisanego) dokumentu, aby na jego podstawie tworzyć i bezpiecznie weryfikować podpisy XML. Natomiast w pracy [208] autor przedstawił odmianę *Szablonu Podpisu*, który jest integralną częścią formularza jako opcjonalny atrybut jego podpisów jak pokazano w tabeli 9.5.

9.2.6. Wyrażenia

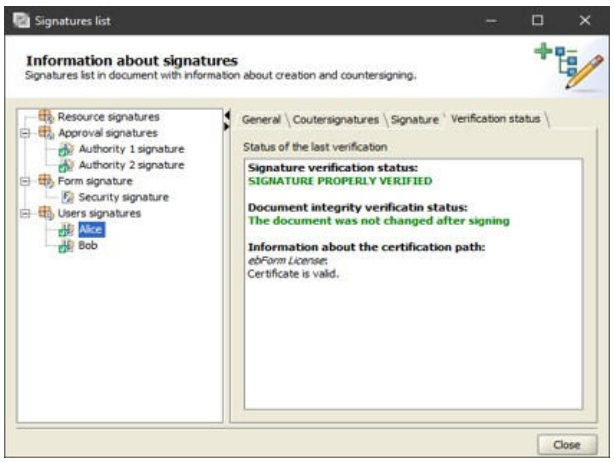
Wyrażenia są to struktury w składni formularza, które udostępniają (wskazują) wartości określonego typu, wykorzystywane do definiowania aktualnych wartości atrybutów w komponentach graficznych oraz aktualnych wartości argumentów (parametrów) akcji i funkcji różnego typu. W zależności od charakteru wskazywanych obiektów zdefiniowano wyrażenia następujących typów: tekst, wartość logiczna, wartość dziesiętna, wartość całkowita, data, obiekty binarne, a także kolory, czcionki, rodzaj obramowania komponentów itd.

Na rysunku 9.11 zaprezentowano koncepcję struktury wyrażenia dowolnego typu w graficznej postaci schematu XML. Jak pokazano na rysunku wyrażenie (`(ExpressionOfType)`) wskazuje na rezultat jako:

1. Element określonego typu `<ElementOfType>` lub referencja do elementu określonego typu `<ReferenceToExpressionOfType>`,
2. Funkcję określonego typu `FunctionOfType`,

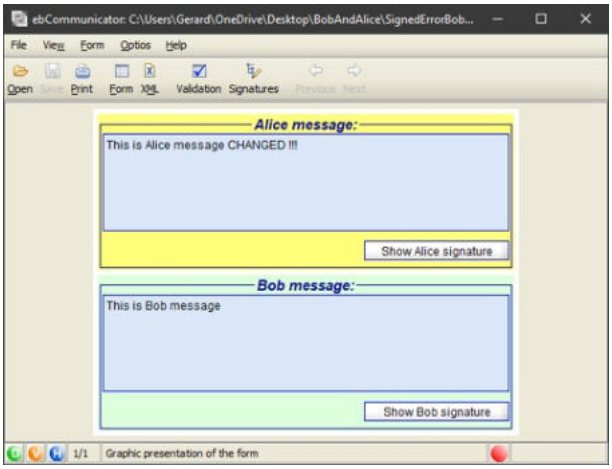


(a) Formularz ze zmanipulowanym polem *Alice message* z informacją o pozytywnym wyniku weryfikacji podpisów

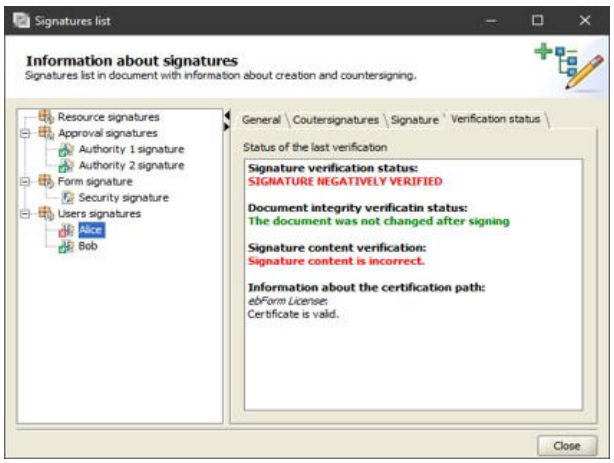


(b) Okienko aplikacji z pozytywnym wynikiem weryfikacji podpisu 9.9a pomimo zmanipulowania danych

Rys. 9.9: Obsługa podpisów w aplikacji użytkownika – informacje o podpisach i zastosowanie szablonu podpisu (źródło: [208])



(a) Formularz jak na rys. 9.9a z informacją o negatywnym wyniku weryfikacji podpisów



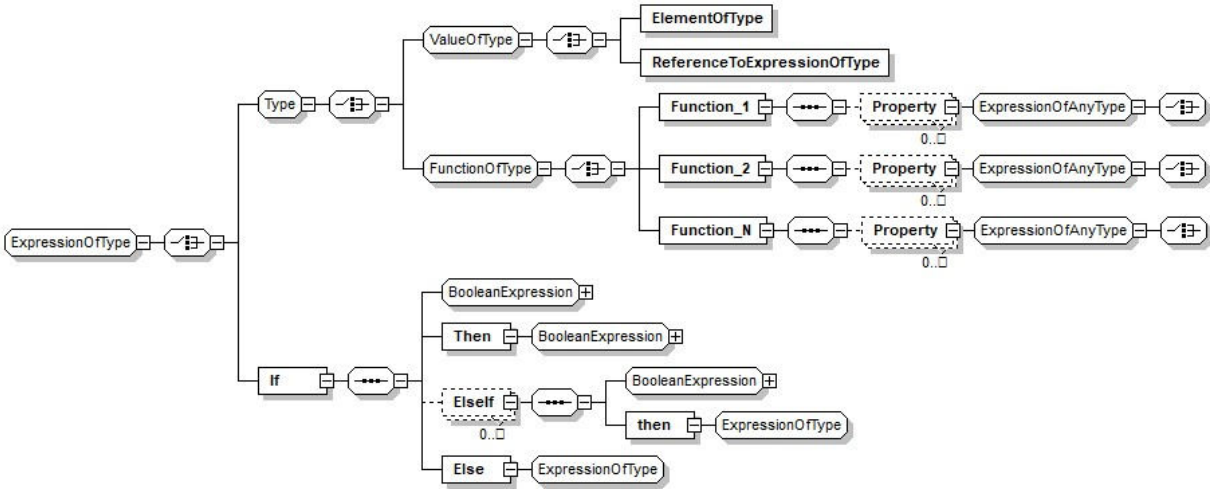
(b) Okienko aplikacji z negatywnym wynikiem weryfikacji podpisu 9.10a z użyciem *szablonu*

Rys. 9.10: Obsługa podpisów w aplikacji użytkownika – informacje o podpisach i zastosowanie szablonu podpisu (źródło: [208])

3. Wyrażenie logiczne o dowolnym stopniu zagnieżdżenia udostępniające wyniki wyrażeń w zależności od wyniku wyrażenia logicznego *if-then-else*.

Za pomocą wyrażeń i funkcji różnych typów możliwe jest realizowanie takich zadań jak:

— generowanie par kluczy, certyfikatów, list certyfikatów unieważnionych *CRL* z wykorzystaniem `(actionExpression)` oraz komponentu `<Attachment>`,
— obsługa danych biometrycznych jako predefiniowanych atrybutów certyfikatów X.509 wyrażonych w funkcjach opartych na wyrażeniach binarnych,



Rys. 9.11: Koncepcja struktury wyrażeń

— wyświetlanie jako obrazów kodów kreskowych jedno- i dwuwymiarowych reprezentujących wartości zbudowane za pomocą wyrażeń tekstowych.

Zgodnie ze schematem na rysunku 9.11, oprócz prostego udostępnienia obiektu lub wartości, koncepcja *wyrażeń* przewiduje możliwość zdefiniowania tego dostępu w sposób warunkowy zależny od aktualnej wartości wyrażenia logicznego wskazanego w elemencie `<if>`.

Definicje struktury wyrażeń różnego typu są przedstawione w podrozdziale A.5 Załącznika na rysunkach A.6–1b – wartości logiczne, A.7–1b – dane tekstowe, A.8–1b – wartości dziesiętne, A.9–1b – wartości całkowite, A.10–1b – daty, A.11–1b – obiekty binarne (*Base64*).

9.3. Trójwarstwowa natura formularza

W zaprojektowanym formularzu elektronicznym wyróżniono trzy warstwy: warstwę danych, warstwę prezentacji i warstwę logiki.

9.3.1. Warstwa danych

Warstwa danych służy do przechowywania danych i posiada ją każdy dokument zarówno tradycyjny, jak i elektroniczny. Jeżeli dokument funkcjonuje jedynie w przestrzeni wirtualnej, bez udziału człowieka w jego przetwarzaniu, wówczas do korzystania z dokumentu (jego przetwarzania) wystarczy specyfikacja jego struktury (składni) oraz znaczenia poszczególnych elementów tej struktury (semantyka).

9.3.2. Warstwa prezentacji

Jeżeli dokument, opisany w podrozdziale 9.3.1, ma być interpretowany przez człowieka, to jego specyfikacja musi definiować sposób jego prezentacji. Specyfikacja pełni funkcję opisu znaczenia danych w dokumencie. Opis ten może być zapisany w odrębnej strukturze (np. transformacja XSLT) lub może być integralną częścią struktury dokumentu. Specyficzny typ dokumentu jakim jest formularz umożliwia wprowadzanie danych w interakcji z użytkownikiem, ich przechowywanie oraz zawiera opis sposobu ich prezentowania – łączy zatem warstwę danych z warstwą prezentacji i elementami interakcji.

9.3.3. Warstwa logiki

Koncepcja warstwy logiki bazuje na możliwości definiowania wyrażeń, czyli dynamicznych relacji pomiędzy akcjami, atrybutami komponentów graficznych (np. wartości obliczane i domyślne, teksty pomocy dla użytkownika, reguły poprawności, inne atrybuty) a stanem danych w formularzu. Ponadto warstwa logiki zawiera zależności pomiędzy fragmentami formularza i podpisami stron transakcji.

Wyrażenia

Wyrażenia, omówione w podrozdziale 9.2.6, mają za zadanie udostępnianie wartości różnych typów danych (tekst, data, wartości numeryczne, logiczne, binaria) wykorzystywanych w komponentach.

Akcje

Akcje, omówione w podrozdziale 9.2.4, umożliwiają wskazanie procedur wykonywanych w efekcie naciśnięcia komponentu graficznego *Przycisk* (`<Button>`). Wskazanie akcji jaka ma być wykonana jest wyrażeniem, dzięki czemu działanie zależy od stanu danych w formularzu i decyzji użytkownika. Wyróżniamy:

1. działania wewnętrzne związane z formularzem (podpisywanie, walidacja, przejście do strony, wykonanie lub usunięcie podpisu, weryfikacja podpisu),
2. działania zewnętrzne związane z komunikacją z otoczeniem (wymiana wiadomości SOAP, zbudowanie, szyfrowanie, podpisanie i wysłanie wiadomości pocztowej do listy odbiorców).

Atrybuty komponentów

Atrybuty komponentów, zaprezentowane w tabelach 9.1 i 9.2, mają za zadanie dostosowanie sposobu wyświetlania komponentu do stanu danych pól formularza zgodnie z wymogami specyficznymi dla transakcji.

Wartości obliczane i domyślne

Wartości te są zaliczane do atrybutów komponentów graficznych. Wartości obliczane są definiowane dla komponentu graficznego `<Text>` z wykorzystaniem wyrażenia tekstowego. Natomiast wartości domyślne są wprowadzane do komponentu w wyniku uruchomienia akcji `<update>` po naciśnięciu komponentu *Przycisk*.

Teksty pomoc dla użytkownika

Są to teksty pomocy i podpowiedzi związane z konkretnym komponentem graficznym jako jego atrybut statyczny. Są one wyświetlane przez aplikację użytkownika.

Reguły poprawności

Reguły poprawności stanowią pomoc dla użytkownika oraz są funkcją lub „akcją” formularza, która może być wykorzystana do określenia dynamicznych wartości innych atrybutów. Ich definicja uwzględnia aktualny stan danych zapisanych w formularzu.

System podpisów

System podpisów, których typy zostały omówione w podrozdziale 9.2.5, stanowią kompleksową ochronę całej struktury formularza (podpisy zatwierdzające, podpis formularza), jego poszczególnych fragmentów (podpisy użytkowników) oraz zewnętrznych zasobów dokumentacji związanej z formularzem.

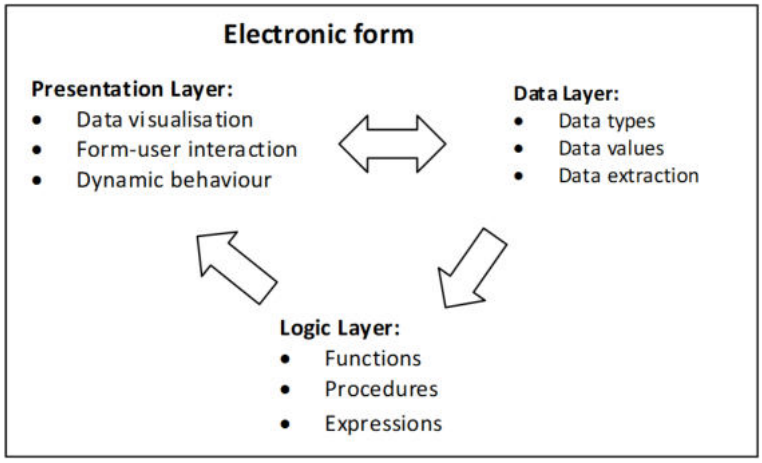
Reasumując można stwierdzić, że warstwa logiki dała nowe w stosunku do tradycyjnych rozwiązań możliwości związane z:

1. wprowadzeniem funkcji wykorzystywanych do budowania logiki dokumentu, które umożliwiają:
   — automatyczną walidację danych w formularzu w zależności od wartości innych danych,
   — obliczanie i prezentację wartości obliczanych, które są wyznaczane na podstawie aktualnych danych w formularzu,
   — obliczanie wartości domyślnych wskazanych danych,
   — dynamiczną, w zależności od stanu danych, modyfikację wartości atrybutów komponentów graficznych,
   — definiowanie podpisów elektronicznych, które odzwierciedlają relacje pomiędzy podpisem i podpisywanymi fragmentami formularza,
   — definiowanie tekstów pomocy i podpowiedzi dla poszczególnych komponentów formularza,
   — włączaniem do formularza innych plików niezależnie od ich formatu;
2. definiowaniem interakcji formularza zarówno w obszarze danych zapisanych w formularzu (akcje wewnętrzne), jak i w kontaktach ze światem zewnętrznym (akcje zewnętrzne), które umożliwiają:
   — zbudowanie, podpisanie, szyfrowanie i wysłanie dokumentu do wskazanych odbiorców z wykorzystaniem usług serwera poczty elektronicznej (SMTP),
   — zbudowanie wiadomości na podstawie wskazanych danych i wysłanie jej z wykorzystaniem protokołu SOAP, a następnie odbiór i interpretacja odpowiedzi z serwera SOAP,
   — wymuszenie wprowadzenia obliczonych wartości domyślnych do wskazanych pól formularza,
   — wymuszenie walidacji (w tym pełna weryfikacja wskazanego podpisu) i wskazanie błędów w danych wprowadzonych do formularza,
   — wykonanie wskazanego podpisu z użyciem danych wymaganych do jego złożenia;
3. bezpieczeństwem, czyli możliwością odzwierciedlenia w formularzu reguł bezpieczeństwa kolejnych etapów i obszarów wspomaganej transakcji, a w szczególności możliwość włączenia do zawartości podpisu różnych części formularza oraz włączenie części formularza do zawartości wielu podpisów.

Omówiona powyżej warstwa logiki, rozszerzająca koncepcję formularza, jest bardzo ważnym jego elementem, gdyż stanowi ona podstawę do budowania formularzy, które precyzyjnie odzwierciedlają realizację wspomaganej transakcji. Ponadto dynamiczne kształtowanie atrybutów komponentów graficznych pozwala na sterowanie przebiegiem transakcji na podstawie zmieniającego się stanu pól formularza. Na rysunku 9.12 zaprezentowano główne role formularza elektronicznego realizowane w poszczególnych warstwach.

9.4. Oprogramowanie

Zaprojektowany formularz elektroniczny ma postać pliku, w którym zapisane są wszystkie dane i informacje potrzebne do jego funkcjonowania. Są one przechowywane w odpowiednich miejscach struktury formularza, która została omówiona w podrozdziale 9.2.



Rys. 9.12: Trzy warstwy dokumentu elektronicznego i ich rola (źródło: [208])

Plik formularza ma popularny format XML, który zapewnia łatwość przenoszenia danych pomiędzy systemami a użytkownikami i jest neutralny technologicznie, a dokumenty w tym formacie umożliwiają łatwe, zunifikowane przetwarzanie (wyszukiwanie, odczytanie, zapisywanie danych) z wykorzystaniem prostych w zastosowaniu bibliotek (*parserów*). Ponadto korzystanie z formatu XML umożliwia automatyczną weryfikację poprawności struktury przesyłanego pliku. Jest to zapewnione automatycznie poprzez:

— weryfikację poprawności struktury XML (*well-formed document*), która obejmuje sprawdzenie zgodności z ogólnymi zasadami struktury XML,
— weryfikację poprawności składniowej (*valid document*), czyli zgodności z definicją składni zapisanej w schemacie XSD (*XSD - XML Schema Definition*) lub DTD (*DTD - Data Type Definition*).

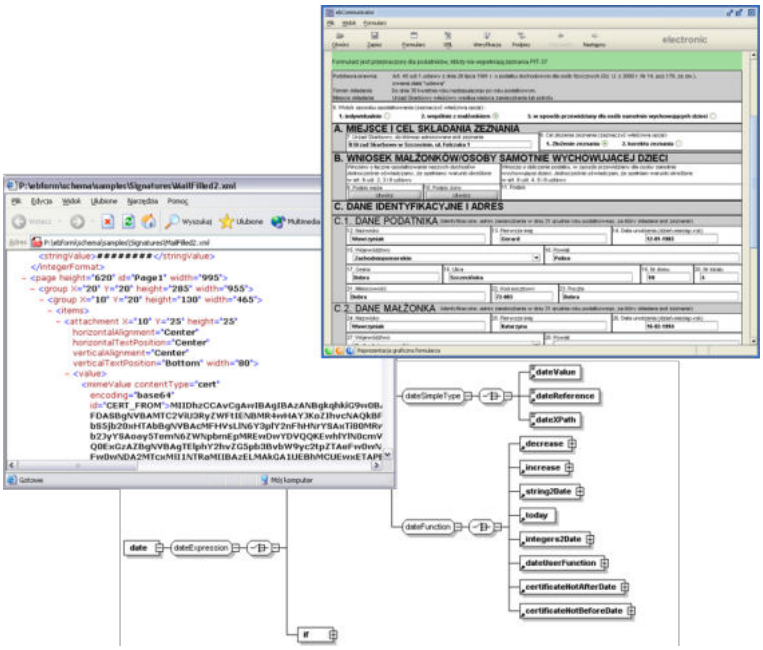
Szczegółowy opis struktury pliku XML przedstawiono w postaci schematów składni w Załączniku A, a relacje pomiędzy plikiem formularza, jego składnią i aplikacją użytkownika pokazano na rysunku 9.13.

Funkcje obsługi formularza oraz interakcja z użytkownikiem są realizowane poprzez aplikację użytkownika. Natomiast do wspomagania twórców oprogramowania i rozwiązań związanych z integracją formularzy elektronicznych z innymi systemami mogą służyć narzędzia rozwojowe, czyli biblioteka funkcji obsługi formularza i tzw. komponent serwerowy.

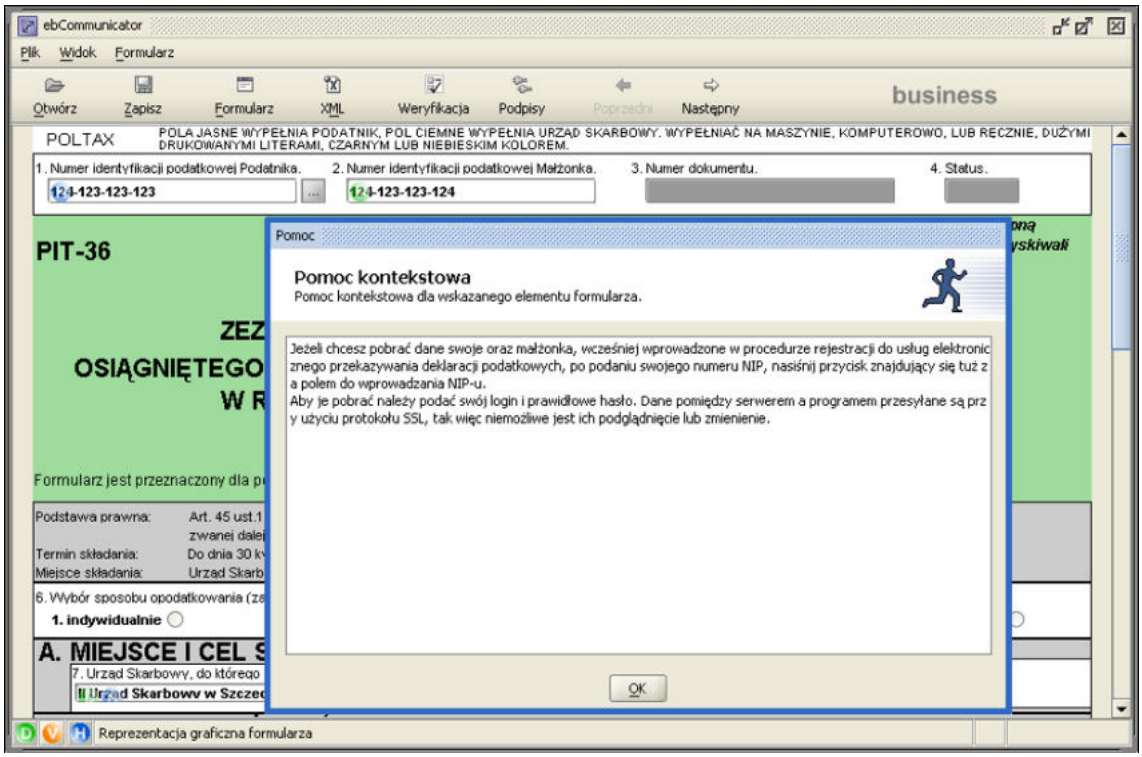
9.4.1. Aplikacja użytkownika

Zadaniem aplikacji użytkownika jest odczytanie i interpretacja wskazanego pliku formularza. Ponadto aplikacja ta umożliwia:

1. realizację interakcji w zakresie edycji danych (wprowadzanie, modyfikacja, usuwanie danych w polach formularza),
2. wyświetlanie bieżącej postaci formularza zgodnie z informacjami o sposobie prezentacji stosownie do aktualnego stanu danych w formularzu (kolory, napisy) zgodnie z jego logiką odzwierciedloną za pomocą wyrażeń,
3. interpretację i wykonywanie działań związanych z pomocą dla użytkownika (wyświetlanie tekstów podpowiedzi i pomocy, weryfikacja poprawności wprowadzonych danych, wprowadzanie wartości domyślnych). Przykład użycia i wizualizacji tekstu pomocy kontekstowej dla wybranego atrybutu pokazano na rysunku 9.14,



Rys. 9.13: Plik (XML), składnia (XSD) i aplikacja użytkownika formularza



Rys. 9.14: Przykład funkcji aplikacji użytkownika – wyświetlenie tekstu pomocy dla wybranego komponentu formularza

4. realizację działań (opisanych w podrozdziale 9.2.4) wykonywanych w wyniku naciśnięcia *Przycisku* (interakcji użytkownika), które są przedstawione w tabeli 9.4,
5. pełną obsługę wszystkich typów podpisów elektronicznych zdefiniowanych w formularzu, które opisano w podrozdziałach 9.2.5 i 9.4.2.

9.4.2. Obsługa podpisów elektronicznych

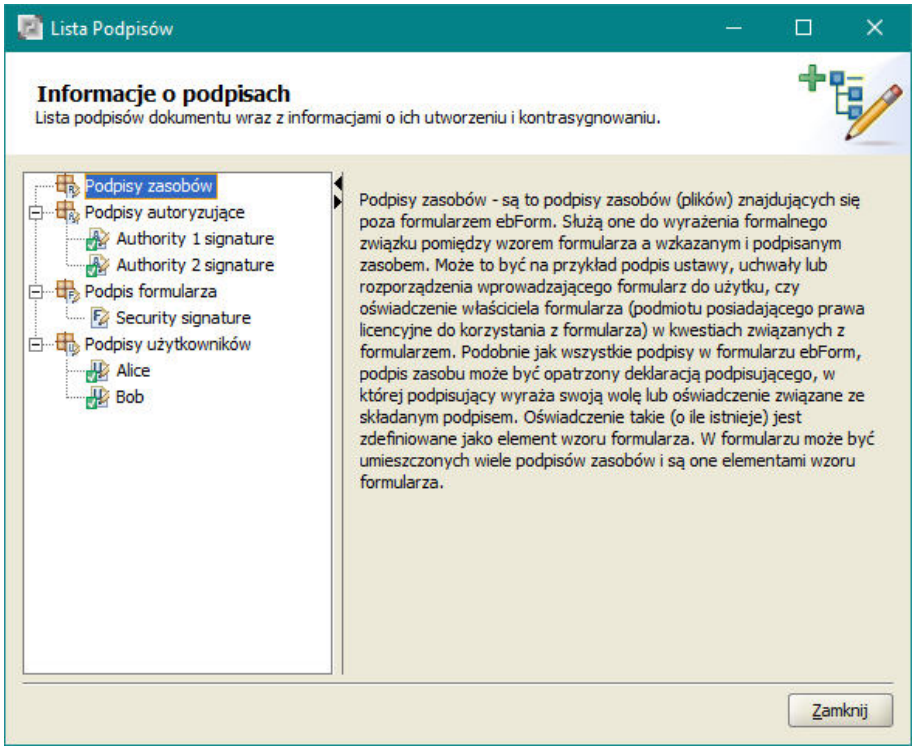
Jednym z podstawowych zadań wykonywanym przez aplikację użytkownika formularzy elektronicznych jest obsługa wszystkich typów podpisów elektronicznych zdefiniowanych w jego strukturze. Wykorzystywane w zaprojektowanym formularzu elektronicznym i obsługiwane przez aplikację użytkownika typy podpisów opisano w podrozdziale 9.2.5 i są one zgodne ze specyfikacjami omówionymi w podrozdziale 5.2.1, a ich przetwarzanie opisano w podrozdziale 5.2.3.

Obsługa podpisu elektronicznego wymaga korzystania z funkcji zaimplementowanych w aplikacji użytkownika, a w szczególności z:

— obsługi kluczy kryptograficznych i certyfikatów do tworzenia podpisów, czyli magazynów kluczy w postaci plików PKCS#12, (PFX) lub JKS (*Java Key Store*) albo w postaci kart kryptograficznych dostępnych z wykorzystaniem interfejsu PKCS#11,
— utrzymywania aktualnych certyfikatów wystawców i list CRL wykorzystywanych do pełnej weryfikacji podpisów.

Funkcje tworzenia, weryfikowania oraz wyświetlania informacji o podpisie mogą być uruchomione z użyciem przycisków zaprojektowanych i zaimplementowanych w formularzu w postaci wyrażeń akcji `(actionExpression)`. Operacje te mogą być również wykonane przez użycie funkcji aplikacji użytkownika, które są dostępne w *menu* lub *Belce narzędzi*. Zadania wykonywane przez aplikację w zakresie obsługi podpisów to:

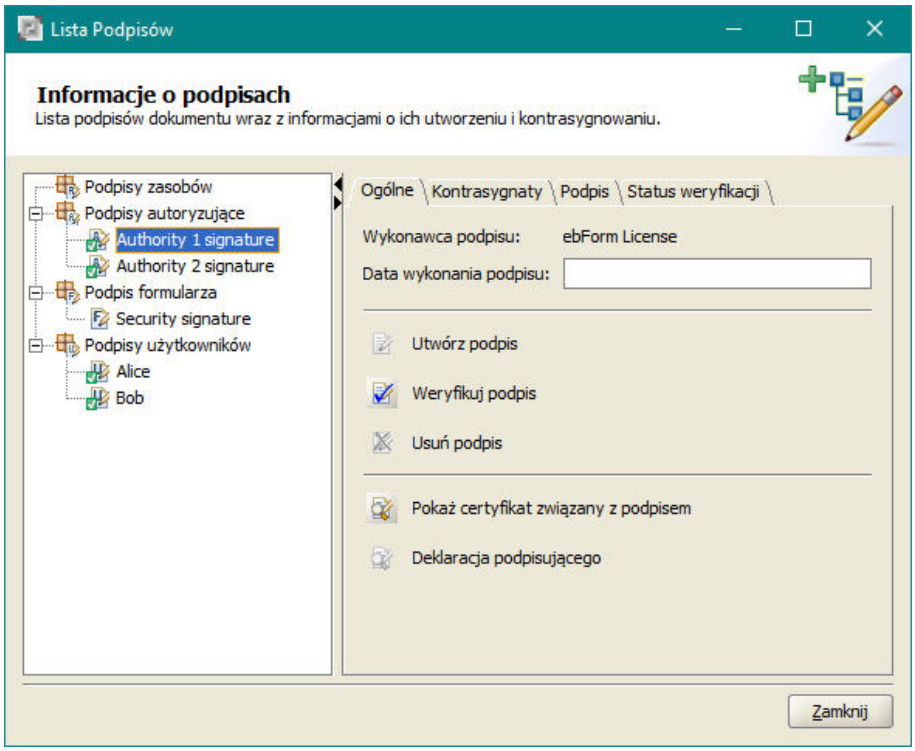
— prezentacja istniejącej w formularzu struktury podpisów wraz z opisami znaczenia poszczególnych ich typów oraz wskazanie podpisów istniejących (wykonanych) w formularzu (rysunek 9.15),



Rys. 9.15: Prezentacja struktury podpisów w formularzu

— obsługa wskazanego podpisu, czyli utworzenie podpisu i jego weryfikacja (o ile podpis jest wykonany), usunięcie podpisu (o ile istnieje i w logice formularza zaznaczono, że podpis może być usuwany), wyświetlanie szczegółowych informacji o certyfikacie zapi-

sanym w strukturze podpisu, a także wyświetlenie deklaracji podpisującego, czyli opisu znaczenia podpisu w transakcji. Ponadto w ramach obsługi podpisu aplikacja prezentuje nazwy podpisów kontrasygnujących i kontrasygnowanych przez podpis, wartość podpisu oraz informacje o statusie weryfikacji (rysunek 9.16).



Rys. 9.16: Obsługa wskazanego podpisu

Tworzenie i weryfikacja podpisów elektronicznych w zakresie integralności i poprawnej zawartości podpisów w formularzach są realizowane zgodnie ze schematami przedstawionymi w podrozdziale 5.2.3 z wykorzystaniem koncepcji autora *szablonu podpisu* (podrozdział 9.2.5), która została opisana w pracach [206, 208]. Zadaniem tego *szablonu* jest wyeliminowanie podatności referencji podpisu zaprezentowanej w podrozdziale 5.5.

W formularzu nie są przechowywane żadne parametry niezbędne do weryfikacji certyfikatu podpisu. Zadaniem aplikacji jest takie zarządzanie danymi do weryfikacji certyfikatów, aby możliwa była poprawna ich weryfikacja z użyciem aktualnych danych (listy certyfikatów unieważnionych i certyfikatów budujących poprawną ścieżkę certyfikacji). Z kolei zarządzanie tymi danymi jest realizowane przez automatyczne pobieranie plików i przechowywanie ich w lokalnym systemie plików. Ważność certyfikatu jest stwierdzana przy użyciu usług infrastruktury klucza publicznego.

Formularz elektroniczny i aplikacja użytkownika służą do wygenerowania podpisu elektronicznego zgodnie ze standardem XMLdSig niezależnie od konieczności interpretowania i wykonania warstwy logiki. Na etapie weryfikacji zadaniem aplikacji jest sprawdzenie ważności podpisu, a w tym również ważności certyfikatu. Zadanie to jest realizowane z wykorzystaniem standardowej infrastruktury klucza publicznego i do tej weryfikacji nie są potrzebne żadne informacje zapisane w logice formularza. Jak pokazano w podrozdziale 6.5.3 do weryfikacji ważności certyfikatu jako alternatywę dla PKI można wykorzystać rozwiązania oparte na łańcuchach bloków, których zastosowanie zostanie omówione w rozdziale 10.

9.4.3. Komponenty wspomagające

Oprócz aplikacji użytkownika, która zapewnia realizację transakcji przez użytkowników, konieczne jest zintegrowanie formularzy z otoczeniem, czyli włączenie do procesu realizacji transakcji środowiska technicznego rozumianego jako zbiór zautomatyzowanych usług i systemów dostępnych w usługach sieciowych na przykład w architekturze *SOA*. W celu usprawnienia integracji formularza elektronicznego z innymi rozwiązaniami wykonano bibliotekę funkcji, które pozwalają na przetwarzanie formularza z uwzględnieniem specyfiki jego składni.

Biblioteka funkcji

Biblioteka funkcji jest zbiorem funkcji obsługi formularza, których celem jest ułatwienie budowania nowych lub rozszerzanie istniejących rozwiązań informatycznych, które wspomagają integrację formularza z usługami i systemami informatycznymi. Do funkcji tych należą:

— funkcje ogólne oraz wspomagające przetwarzanie struktury XML: otworzenie i zapisanie pliku, wyszukiwanie elementu, kodowanie i dekodowanie `Base64` obiektów binarnych, wyszukiwanie węzłów XML na podstawie relacji przodek-potomek, odczytywanie i ustawianie wartości atrybutów oraz zawartości elementów,

— funkcje specyficzne dla przetwarzania formularza: odczytanie przestrzeni nazw, identyfikatorów formularza i jego elementów, odczytywanie i ustawianie wartości pól formularza wskazanych przez wartości ich identyfikatorów z uwzględnieniem ich typów, sprawdzanie dostępności elementów (`<isEditable>`), określanie statusu wskazanych pól – czy są podpisane (czy istnieje podpis, którego są zawartością),

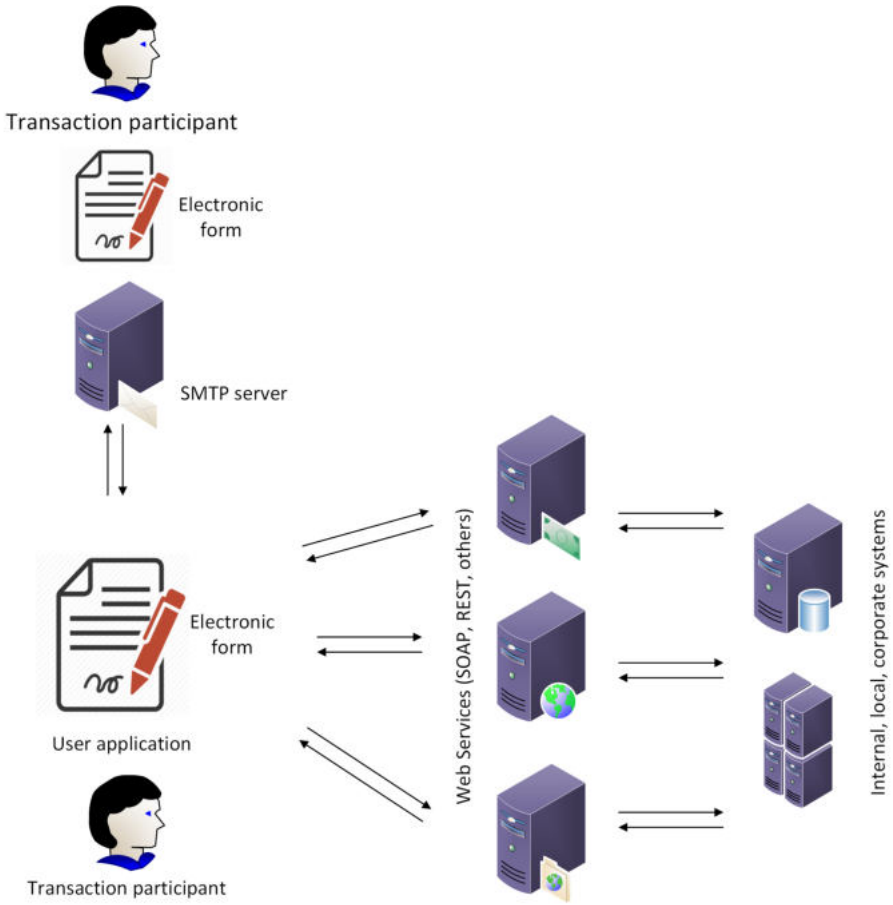
— funkcje obsługi podpisów zgodnie ze zdefiniowaną w formularzu logiką, które obejmują: zbudowanie listy identyfikatorów zdefiniowanych podpisów, określenie typu i nazwy wskazanego podpisu, sprawdzenie, czy wskazany podpis istnieje (jest wykonany), odczytanie obiektu podpisu formularza, sprawdzenie statusu (weryfikacja poprawności) wszystkich podpisów, obsługa magazynów kluczy, generowanie wskazanego podpisu z użyciem klucza zapisanego w magazynie kluczy (PKCS#11, PKCS#12, JKS) z uwzględnieniem logiki kontrasygnowania podpisów zdefiniowanej w formularzu, co omówiono w podrozdziale 9.2.5 i przedstawiono na schematach zaprezentowanych w podrozdziale A.4 Załącznika A.

Komponent serwerowy

Kolejnym komponentem, który ułatwił i przyspieszył budowę oraz integrację przetwarzania formularza z innymi systemami, jest komponent serwerowy. Jest to aplikacja serwerowa, uruchamiana jako aplikacja WAR (Web ARchive) na serwerze *Tomcat*. Miejsce tego komponentu w ogólnym, kontekstowym schemacie przetwarzania formularza pokazano na rysunku 9.17.

Koncepcja wykorzystania komponentu serwerowego polega na przetwarzaniu otrzymanego formularza zgodnie z logiką zapisaną w tzw. deskryptorze procesów. Na przetwarzanie formularza składają się:

1. odczytanie aktualnych danych (sekretów i lokalizacji) dostępu do magazynów kluczy (magazyn kluczy zaufanych, kwalifikowanych i innych) i wskazanie położenia plików list CRL,
2. zidentyfikowanie typu formularza na podstawie wskazanych w deskryptorze cech będących podstawą do uruchomienia przetwarzania specyficznego dla danego typu formularza,



Rys. 9.17: Formularz w środowisku – strona transakcji (człowiek), aplikacja użytkownika, usługa sieciowa (poczta) oraz serwery i systemy wewnętrzne

3. identyfikacja etapu realizacji, która jest dokonywana poprzez wskazanie stanu pól formularza odzwierciedlonych wyrażeniami udostępniającymi cechy i ich wartości,
4. uruchomianie realizacji procesu, na który składa się ciąg operacji takich jak: zbudowanie i wysłanie wiadomości pocztowej do listy odbiorców, wprowadzenie danych do wskazanych pól formularza, wykonanie lub weryfikacja wskazanych podpisów z użyciem wskazanych w punkcie pierwszym magazynów kluczy i sekretów, zapisanie formularza na dysku lub przekazanie do innych systemów.

Po wykonaniu opisanych powyżej procesów zmodyfikowany formularz jest zwracany do nadawcy w trybie synchronicznej odpowiedzi na żądanie w protokole SOAP.

9.5. Podsumowanie

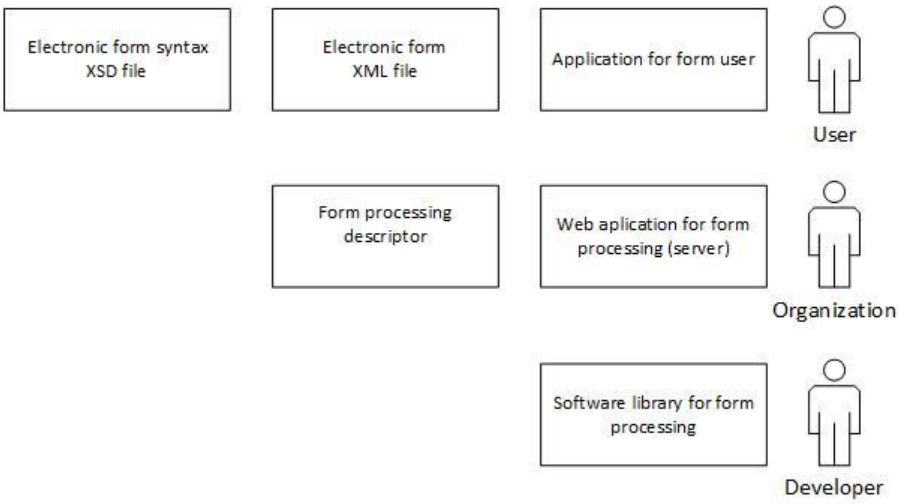
W rozdziale 9 zaprezentowano genezę, założenia, budowę oraz funkcjonowanie formularzy elektronicznych. Budując rozwiązanie formularzy elektronicznych stworzono statyczne i dynamiczne atrybuty komponentów graficznych, które odzwierciedlają stan realizowanej transakcji wykorzystując w tym celu tzw. wyrażenia określonych typów, które z kolei odwołują się do elementów struktury formularza oraz do zbioru wbudowanych funkcji różnych typów. Wyrażenia wiążące logicznie elementy komponentów i funkcji umożliwiły modelowanie dynamicznego zachowania (dostosowanie wartości atrybutów i uruchamia-

nie wskazanych działań) formularza stosownie do stanu realizowanych transakcji. Istotnym aspektem zaprojektowanych formularzy elektronicznych jest system podpisów elektronicznych, które zabezpieczają formularz w różnych obszarach takich jak: środowisko zewnętrzne, struktura logiki i prezentacji formularza, dane użytkowników.

W rozdziale podkreślono również rolę biblioteki funkcji, które pozwalają na łatwą budowę rozwiązań programistycznych operujących na obiektach formularzy elektronicznych.

Przedstawiono także zaprojektowaną aplikację serwerową, działającą w środowisku protokołu SOAP. Jej zadaniem jest odbiór żądania wysłanego przez aplikację użytkownika jako efekt wykonania logiki akcji zdefiniowanej w formularzu, a następnie wybór i wykonanie jednego ze zdefiniowanych procesów (wyrażonych w pliku deskryptora procesów w aplikacji serwerowej) dostosowanych do konkretnej transakcji i etapu jej realizacji. Proces przetwarzania kończy synchroniczny zwrot przetworzonego formularza do aplikacji użytkownika.

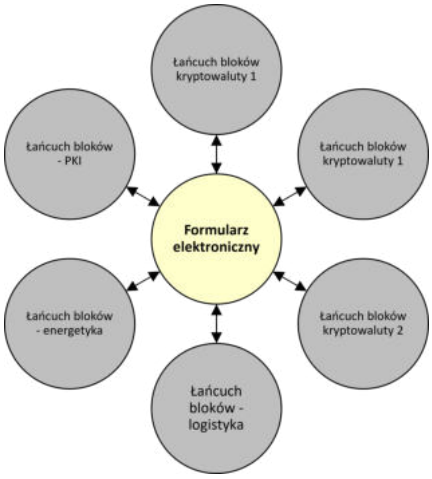
Rysunek 9.18 przedstawia poszczególne komponenty zaprojektowanego formularza elektronicznego takie jak: aplikacja użytkownika, aplikacja serwerowej i biblioteka obsługi formularzy.



Rys. 9.18: Składniki techniczne wspomagające formularze elektroniczne

Wszystkie rozważania zawarte w tym rozdziale potwierdzają, że zaprojektowany formularz elektroniczny wiernie odzwierciedla dokument tradycyjny, jest zrozumiały dla użytkownika i dostosowuje się do stanu realizowanej transakcji oraz umożliwia bezpieczne sterowanie jej przebiegiem.

10. Integracja formularzy elektronicznych z łańcuchami bloków



Rys. 10.1: Formularz elektroniczny zintegrowany z różnymi funkcjonalnie i technologicznie łańcuchami bloków

Formularze elektroniczne, jak wielokrotnie w pracy wspomniano, funkcjonują w zróżnicowanych środowiskach w wymiarze organizacyjno-prawnym oraz w wymiarze technologiczno-implementacyjnym. Ponieważ łańcuchy bloków są rejestrami rozproszonymi, to aby osiągnąć korzyści opisane w podrozdziale 6.3, trzeba je zintegrować z formularzami. Należy zauważyć, że brak jest standardowych, zunifikowanych mechanizmów komunikacji z łańcuchami bloków. Dlatego niemożliwe jest zaimplementowanie w logice formularza wbudowanej uniwersalnej funkcji lub „akcji" zapewniającej komunikację z różnymi łańcuchami bloków. Problem ten ilustruje rysunek 10.1, na którym jeden formularz korzysta z usług kilku różnych łańcuchów bloków.

W tym rozdziale zaprezentowano trzy sposoby integracji łańcuchów bloków z formularzami elektronicznymi, które rozwiązują problem braku uniwersalnego rozwiązania. Sposoby te to:

1. integracja w oparciu o usługi sieciowe,
2. integracja poprzez rozszerzenie wbudowanych funkcji formularza,
3. integracja poprzez wykorzystanie uniwersalnego rozszerzania funkcjonalności formularza.

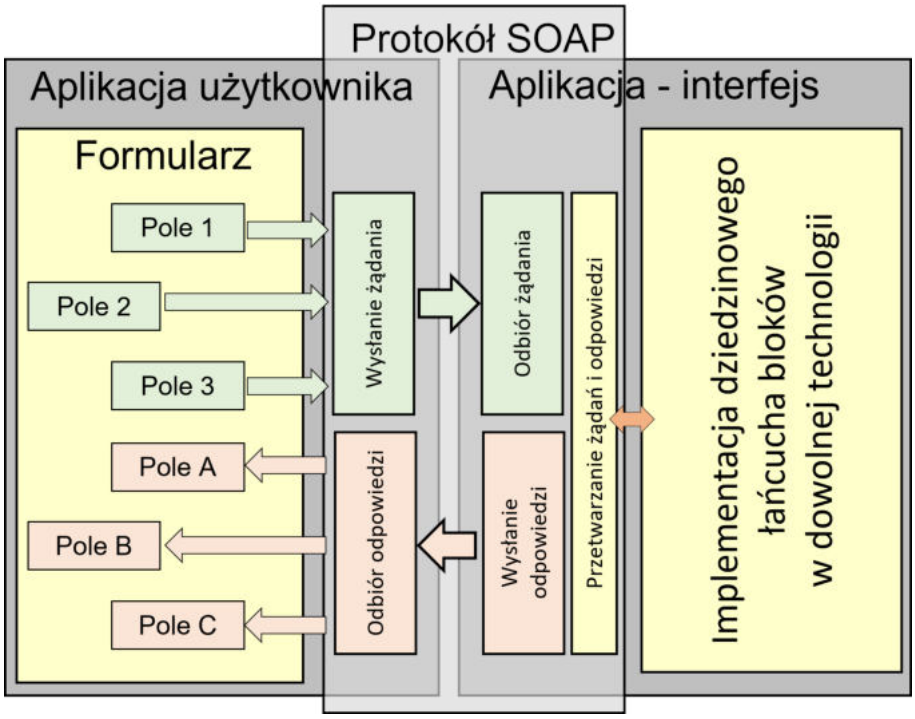
Sposoby te są neutralne z punktu widzenia obszarów zastosowań, stosowanych procedur realizowanych transakcji oraz różnorodności platform i rozwiązań technicznych.

10.1. Integracja w oparciu o usługi sieciowe

Integracja w oparciu o usługi sieciowe jest rozwiązaniem, w którym dostęp do obsługi łańcucha bloków jest realizowany poprzez usługi sieciowe. Komunikacja pomiędzy aplikacją użytkownika formularza a łańcuchem bloków odbywa się w następujących etapach (rysunek 10.2):

— budowanie żądania do usługi na podstawie danych w polach formularza zgodnie z jego logiką,
— wysłanie zbudowanego żądania do usługi,
— odebranie odpowiedzi wygenerowanej przez usługę łańcucha bloków,
— wizualizacja otrzymanych danych zgodnie z logiką formularza.

Z punktu widzenia funkcji wbudowanych do składni formularza i obsługiwanych przez aplikację użytkownika jest to rozwiązanie naturalne i omówiono je w podrozdziałach 9.2.4 i 11.2. Jednak z praktycznego punktu widzenia to rozwiązanie wymaga stworzenia oprogramowania interfejsowego, udostępniającego określone usługi w protokole SOAP. Rozwiązanie to brano pod uwagę dla szeroko dostępnych usług, dla których wymagany jest standardowy interfejs.



Rys. 10.2: Interakcja formularzy elektronicznych z łańcuchami bloków z wykorzystaniem protokołu SOAP

10.2. Integracja poprzez rozszerzenie wbudowanych funkcji formularza elektronicznego

Kolejnym sposobem integracji łańcuchów bloków z formularzem jest rozszerzenie jego funkcjonalności. W tym celu należy dokonać:

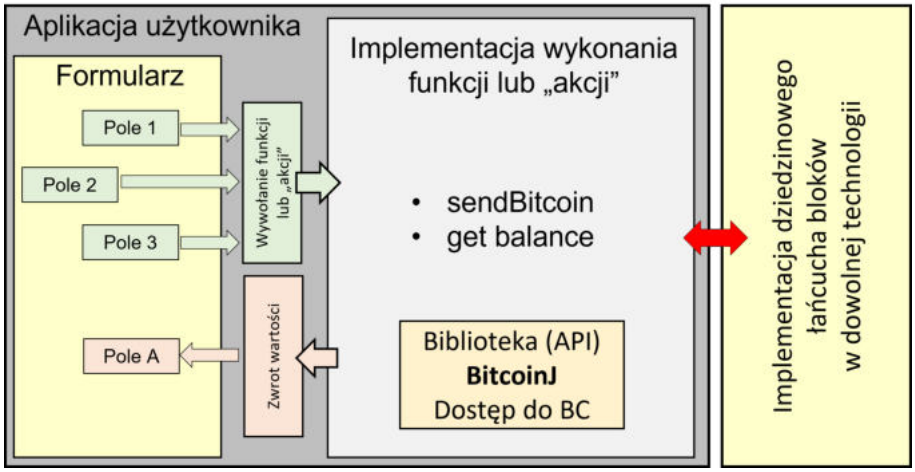
A. analizy natury wprowadzanego rozszerzenia, czyli protokołów i mechanizmów korzystania z łańcucha bloków,
B. rozszerzenia składni formularza zgodnie z wynikami analizy, czyli modyfikacji schematu XSD,
C. implementacji zmian w kodzie źródłowym w zakresie określonym w analizie oraz zgodnie ze zmodyfikowaną składnią formularza.

Z uwagi na różnorodność rozwiązań i zastosowań łańcuchów bloków, należy każdą interakcję z łańcuchem bloków zaimplementować modyfikując składnię i rozszerzając oprogramowanie formularza. Prowadzi to do rozrostu składni i oprogramowania, a także stwarza problemy związane z zarządzaniem wersjami oraz dystrybucją rozwiązań z zapewnieniem kompatybilności wstecznej.

Biorąc pod uwagę popularność i powszechność stosowania kryptowaluty *Bitcoin* poniżej przedstawiono rozwiązanie eksperymentalne polegające na rozszerzeniu funkcji formularza elektronicznego o funkcje interakcji z łańcuchem bloków *Bitcoin*.

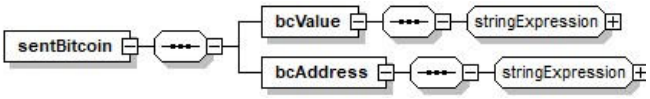
Rysunek 10.3 przedstawia implementację komunikacji formularza elektronicznego z kryptowalutą *Bitcoin*, w której uwzględniono:

— zbudowanie mapy argumentów wywołania funkcji lub „akcji",
— wywołanie procedury implementującej wykonanie funkcji lub „akcji" (*sendBitcoin*, *getBalance*), która przy realizacji zadania korzysta z bibliotek/API dostępu do łańcucha bloków (*BitcoinJ*),
— zwrócenie odpowiednich wartości, po wykonaniu procedury implementującej, do środowiska aplikacji użytkownika, gdzie następuje aktualizacja prezentacji danych w formularzu zgodnie ze zdefiniowaną logiką.



Rys. 10.3: Model logiczny implementacji komunikacji formularza elektronicznego z kryptowalutą *Bitcoin*

Włączenie interakcji z łańcuchem bloków wymaga uwzględnienia tego faktu w składni formularza (XSD). Na rysunku 10.4a pokazano składnię uruchamiania „akcji" `<sendBitcoin>` transferu wartości na wskazany adres. Wartości te są wskazane w elementach `<bcValue>` i `<bcAddress>`. Z kolei na rysunku 10.4b zaprezentowano składnię funkcji `<getBitcoinBalance>` udostepniającą aktualny stan transakcji na koncie wskazanym przez `<bitcoinAddress>`.



(a) Składnia definiowania wywołania „akcji" wykonania transferu

(b) Składnia wywołania funkcji otrzymania stanu konta o zadanym adresie

Rys. 10.4: Składnia funkcji i „akcji" budujących interakcję z kryptowalutą *Bitcoin*

Kolejnym, ostatnim etapem rozszerzania funkcji integrującej formularz elektroniczny z kruptowalutą *Bitcoin* jest implementacja transferu `<sendBitcoin>` pokazana na wy-

druku 10.1 i otrzymania salda `<getBitcoinBalance>` dla wskazanego adresu (klucza) pokazana na wydruku 10.2.

```
import java.io.File;

public class BitcoinTransferExample {

    public static void main(String[] args) {
    //  Wskazanie sieci testowej
        NetworkParameters networkParameters = TestNet3Params.get();

        Context context = new Context(networkParameters);

      System.out.println("A");

        // Initialize WalletAppKit
        WalletAppKit kit = new WalletAppKit(networkParameters, new File("."), "wallet") {
            @Override
            protected void onSetupCompleted() {
             // stworzenie i rozgloszenie transakcji
                sendBitcoin(this);
            }
        };

     // Uruchomienie portfela
        kit.startAsync();
        kit.awaitRunning();

     // Zatrzymanie portfela
        kit.stopAsync();
        kit.awaitTerminated();
    }

    private static void sendBitcoin(WalletAppKit kit) {
     // Stworzenie nowego klucza i adresu
        ECKey key = kit.wallet().freshReceiveKey();
        Address address = LegacyAddress.fromKey(kit.params(), key);

     // Informacja o otrzymanym adresie
        System.out.println("Wygenerowany adres: " + address.toString());

   // Wyslanie kwoty na podany adres
        SendRequest sendRequest = SendRequest.to(address, Coin.parseCoin("0.01"));

     // Wyslaniae transakcji
        try {
            kit.wallet().completeTx(sendRequest);
            kit.wallet().commitTx(sendRequest.tx);
            System.out.println("Transakcja zreaizowana! Skrot: " +
                        sendRequest.tx.getTxId());
        } catch (InsufficientMoneyException e) {
            System.out.println("Brak srodkow na realizacje transakcji.");
        }
    }
}
```

Wydruk 10.1: Kod źródłowy realizacji transakcji *Bitcoin*

```
package com.ebstream;
import org.bitcoinj.base.Coin;
import org.bitcoinj.core.*;
import org.bitcoinj.kits.WalletAppKit;
import org.bitcoinj.params.MainNetParams;
import org.bitcoinj.wallet.Wallet;

import java.io.File;

public class BitcoinBalanceExample {

    public static void main(String[] args) {
        // Inicjalizacja sieci (MainNet - tylko tutaj)
        NetworkParameters networkParameters = MainNetParams.get();
```

```
15
        Context context = Context.getOrCreate(networkParameters);
17
        File walletDir = new File(".");
19
        String walletPrefix = "walletappkit-example";
21
        // Tworzenie instancji portfela
23      WalletAppKit kit = new WalletAppKit(networkParameters, walletDir, walletPrefix);

25      // Inicjalizacja portfela
        kit.startAsync();
27      kit.awaitRunning();

29      // Pobieranie portfela
        Wallet wallet = kit.wallet();
31
        // Pobieranie saldo portfela
33      Coin balance = wallet.getBalance();

35      // Wydrukowanie salda
        System.out.println("Saldo portfela: " + balance.toFriendlyString());
37
        // Zatrzymanie portfela
39      kit.stopAsync();
        kit.awaitTerminated();
41      System.out.println("Koniec Programu");
    }
43 }
```

Wydruk 10.2: Kod źródłowy otrzymania salda transakcji *Bitcoin*

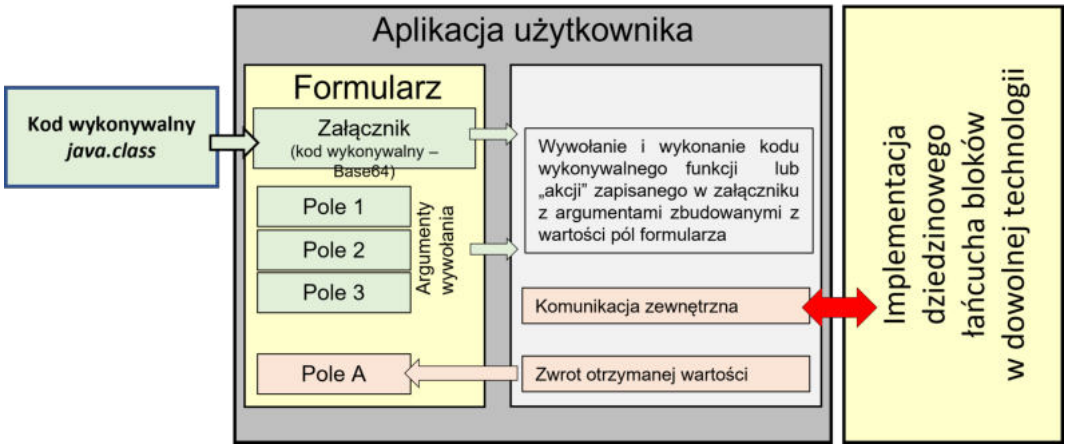
10.3. Integracja poprzez wykorzystanie uniwersalnego rozszerzania funkcjonalności formularza

Ostatnim sposobem integracji formularza z łańcuchami bloków jest opracowany przez autora mechanizm uruchamiania kodów wykonywalnych zapisanych jako dane w formularzu w komponencie *Załącznik* (`<Attachment>`) w ramach jego logiki. Mechanizm ten można wykorzystać do komunikacji z różnymi łańcuchami bloków oraz różnymi systemami PKI opartymi na łańcuchach. Korzystanie z tego mechanizmu polega na:

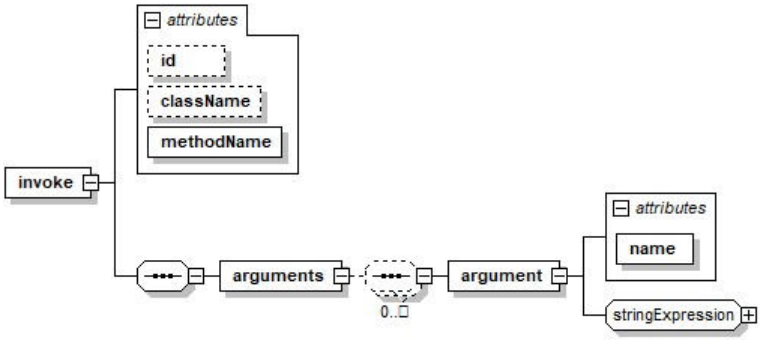
1. stworzeniu klasy w języku *java* i umieszczenie binariów skompilowanej i zakodowanej do postaci *Base64* klasy w załączniku formularza (komponent `<Attachment>` w strukturze formularza),
2. zdefiniowaniu w strukturze formularza „akcji” przyporządkowanej do komponentu `<Button>` ze zdefiniowaną mapą *nazwa – wartość* danych i nazwą metody klasy,
3. uruchomieniu tak zdefiniowanej „akcji” poprzez naciśnięcie stosownego przycisku, co powoduje dynamiczne załadowanie klasy wskazanej w poprzednim punkcie i wywołanie wskazanej metody z przekazaniem do niej mapy danych,
4. interpretacji wartości zwracanej przez wywołaną metodę w środowisku formularza – aplikacji użytkownika.

Na rysunku 10.5 zaprezentowano mechanizm uruchamiania kodów wykonywalnych zapisanych w komponencie graficznym *Załącznik* (`<Attachment>`).

Składnię definicji uruchamiania kodu wykonywalnego, zapisanego w komponencie graficznym *Załącznik* (`<Attachment>`), z przekazaniem do niego argumentów (wartości pól formularza), pokazano na rysunku 10.6



Rys. 10.5: Uruchomienie kodu wykonywalnego zapisanego w *Załączniku* (`<Attachment>`)



Rys. 10.6: Składnia definicji uruchamiania kodu wykonywalnego zapisanego w formularzu

Na wydruku 10.3 zaprezentowano kod źródłowy klasy *java*, której zadaniem jest zbudowanie instancji klasy przekazanej jako binaria (tablica bajtów). Binaria te są pobrane z komponentu *Załącznik* (`<Attachment>`) i są skompilowanym kodem wykonywalnym. Ponadto w klasie zaimplementowano uruchamianie metod wspomnianej klasy z przekazaniem do nich argumentów. W ten sposób użycie klasy i jej metod umożliwia uruchamianie kodów wykonywalnych, zapisanych jako dane w polu formularza, z przekazaniem argumentów zbudowanych zgodnie z logiką formularza (warstwa logiki).

```
import java.lang.reflect.InvocationTargetException;
import java.lang.reflect.Method;
import java.util.Map;

public class ByteClassLoader extends ClassLoader {

  public Class<?> clazz = null;
  public Object object = null;


  public ByteClassLoader(byte[] bytes) throws InstantiationException,
                        IllegalAccessException,
                        NoSuchMethodException,
                        SecurityException,
                        IllegalArgumentException,
                        InvocationTargetException
  {
    clazz = defineClass(null, bytes, 0, bytes.length);
    object = clazz.getDeclaredConstructor().newInstance();
  }

```

```
23    Object invoke(String methodName) throws NoSuchMethodException,
                        SecurityException,
25                      IllegalAccessException,
                        IllegalArgumentException,
27                      InvocationTargetException
      {
29      Method method = clazz.getMethod(methodName);
        Object retObj = method.invoke(object);
31      return retObj;
      }
33
35    Object invoke(String methodName, String arg) throws NoSuchMethodException,
                              SecurityException,
37                            IllegalAccessException,
                              IllegalArgumentException,
39                            InvocationTargetException
      {
41      Method method = clazz.getMethod(methodName, String.class);
        Object retObj = method.invoke(object, arg);
43      return retObj;
      }
45
      Object invoke(String methodName, Map<String, String> arg) throws NoSuchMethodException,
47                                     SecurityException,
                                       IllegalAccessException,
49                                     IllegalArgumentException,
                                       InvocationTargetException
51    {
        Method method = clazz.getMethod(methodName, Map.class);
53      Object retObj = method.invoke(object, arg);
        return retObj;
55    }

57    Object invoke(String methodName, Object argObj) throws NoSuchMethodException,
                              SecurityException,
59                            IllegalAccessException,
                              IllegalArgumentException,
61                            InvocationTargetException
      {
63      System.out.println("Invoke HERE ");
        Method method = clazz.getMethod(methodName, argObj.getClass());
65      Object retObj = method.invoke(object, argObj);
        return retObj;
67    }

69  }
```

Wydruk 10.3: Kod źródłowy uruchamiania wskazanych metod dynamicznie ładowanej klasy

Kolejny wydruk (10.4) jest kodem źródłowym programu, którego zadaniem jest sprawdzenie poprawności i wykonalności koncepcji uruchamiania dynamicznie ładowanej instancji klasy skompilowanej i zakodowanej do postaci tekstu (*Base64*). Tę klasę testową pokazano na wydruku 10.5.

```
1     package com.ebstream;

3     import java.io.FileNotFoundException;
      import java.lang.reflect.InvocationTargetException;
5     import java.util.Arrays;
      import java.util.Base64;
7     import java.util.HashMap;
      import java.util.Map;
9
      public class TestClassLoader {
11
        public static String readBase64File(String fileName) throws FileNotFoundException
```

```
13      {
          byte[] fileBytes = Tools.loadFileBytes(fileName);
15        String fileString = Base64.getEncoder().encodeToString(fileBytes);
          return fileString;
17      }

19

21      public static void main(String[] args) throws ClassNotFoundException,
                                InstantiationException,
23                              IllegalAccessException, NoSuchMethodException,
                                SecurityException, IllegalArgumentException,
25                              InvocationTargetException, FileNotFoundException {
          System.out.println("Testing ClassLoader - START");
27

29
//        ZAKODOWANA DO BASE64 SKOMPILOWANA KLASA JAVA
31        String class_1Base64 = "yv66vgAAADsAdwcAA \dots gEAGXBsL25hdmVzL3Rlc";

33        if(true)
          {
35          System.out.println("Class_1 test - START");

37          byte[] classBytes = Base64.getDecoder().decode(class_1Base64);
            ByteClassLoader classLoader = new ByteClassLoader(classBytes);
39
            Map<String, String> map = new HashMap<String, String>();
41            map.put("key1", "val1");
              map.put("key2", "val2");
43            map.put("key3", "val3");
              map.put("key4", "val4");
45
            Object o = classLoader.invoke("method_1");
47               o = classLoader.invoke("method_2");
                 o = classLoader.invoke("print", "STRING");
49               o = classLoader.invoke("doSth", map);

51          System.out.println("\nReturn Object:\n" + o.toString() + " Class :"  +
                                    o.getClass().getCanonicalName());
53
            System.out.println("Class_1 test - END\n");
55        }

57        System.out.println("Testing ClassLoader - END");
        }
59  }
```

Wydruk 10.4: Program testujący poprawność uruchamiania klasy zakodowanej do postaci *Base64* ze wskazanymi metodami

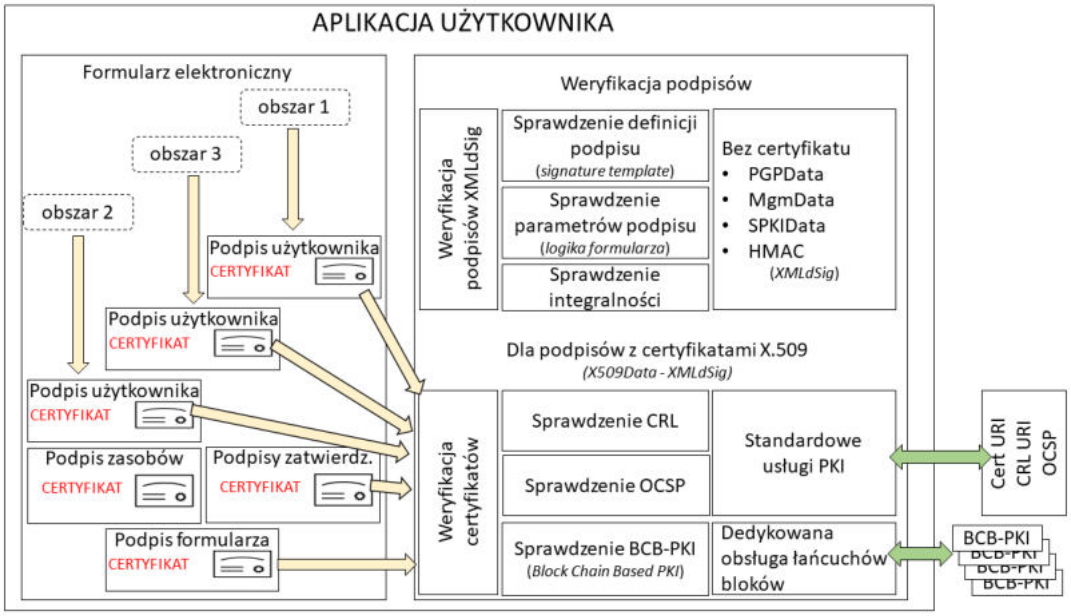
```
1   package pl.naves.test;

3   import java.util.HashMap;
    import java.util.Map.Entry;
5
    public class TestClass_2 {
7
      public String doMap(HashMap<String, String> args)
9     {
        String retStr = this.getClass().getCanonicalName() +
11                  ".doSth(Map<String, String> args):\\n";
        for (Entry<String, String> entry : args.entrySet())
13      {
          String k = entry.getKey();
15        String v = entry.getValue();
          retStr = retStr + "[Key: " + k + "]; Value: " + v + "\n";
17      }
        return retStr;
19    }

21    public static void main(String[] args)
```

```
    {
23    System.out.println("TestClass2 - Completed");
    }
25
  }
```

Wydruk 10.5: Klasa testowa

10.4. Wykorzystanie łańcuchów bloków do weryfikacji certyfikatów

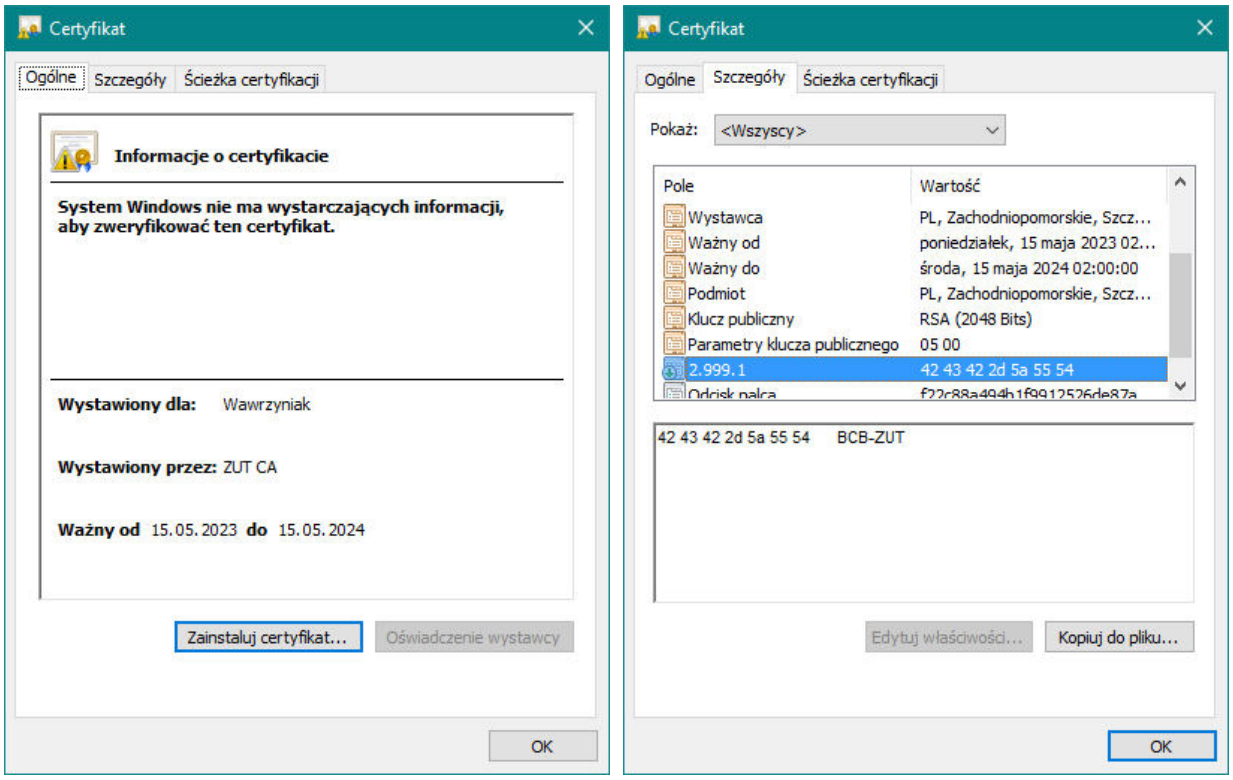
W podrozdziale 6.5.3 omówiono problematykę i podano wnioski dotyczące wykorzystania technologii łańcuchów bloków – technologii rejestrów rozproszonych DLT (*Didstributed Ledger Technology*) – w systemach infrastruktury klucza publicznego. Wymaga to stworzenia dedykowanych – dla poszczególnych rozwiązań – interfejsów dostępu do danych trzymanych w tych rejestrach (PKI opartych na łańcuchach bloków). Na rysunku 10.7 zaprezentowano zaprojektowany przez autora schemat weryfikacji ważności certyfikatów z zastosowaniem dedykowanych procedur obsługi dla różnych implementacji łańcuchów bloków (*Blockchain Based PKI* – BCB-PKI).



Rys. 10.7: Schemat weryfikacji ważności certyfikatów z wykorzystaniem różnych systemów BCB-PKI

Automatyczny wybór procedury weryfikacji przyporządkowanej do konkretnego BCB – PKI jest dokonywany na podstawie informacji zapisanej w rozszerzeniu certyfikatu X.509 (*CertificateExtension*). Na rysunku 10.8b pokazano przykład wygenerowanego certyfikatu, w którym umieszczono rozszerzenie *BCB-ZUT* przeznaczone do wskazania łańcucha bloków i któremu nadano przykładową wartość *identyfikatora obiektu – OID Object IDentifier*.

Opracowany i zaprezentowany powyżej sposób sprawdzania ważności certyfikatu jest tak samo skuteczny jak skorzystanie z klasycznego systemu PKI i gwarantuje taki sam poziom bezpieczeństwa certyfikatu, podpisu i realizowanej transakcji.



(a) Zakładka ogólna

(b) Szczegóły – oid „2.999.1” i wartość „BCB-ZUT”

Rys. 10.8: Certyfikat z rozszerzeniem wskazującym na łańcuch bloków

Nie zawsze istnieje konieczność użycia certyfikatów X.509 w celu zapewnienia niezaprzeczalności pochodzenia. W takich przypadkach nie korzysta się z infrastruktury PKI. Specyfikacja W3 XMLdSig, która stanowi podstawę zaprojektowanego w formularzu systemu podpisów, definiuje również podpisy bez użycia certyfikatów X.509. Takie podpisy posiadają te same cechy jak podpis z certyfikatem, ale zapewniają anonimowość. Zgodnie ze wspomnianą specyfikacją do podpisów tych należą (podrozdział 7.5 w [168]):

— podpisy oparte na technice PGP,
— podpisy zawierające informacje związane ze sposobem pozyskania klucza publicznego za pomocą tzw. *identyfikatora klucza publicznego podmiotu* (*SPKI - Subject Public Key Identifier*),
— podpisy, w których strony ustalają między sobą informacje o sposobie dystrybucji kluczy (MgmtData) (metoda niezalecana),
— podpisy wykorzystujące kody uwierzytelniające oparte na wartości funkcji skrótu *(HMAC - Hash Message Authentication Code)*.

Reasumując należy stwierdzić, że zaprojektowany system weryfikacji certyfikatów przez aplikację użytkownika formularza oprócz wykorzystania klasycznej infrastruktury PKI daje również możliwość korzystania z infrastruktury klucza publicznego opartej na łańcuchach bloków. Natomiast w przypadku konieczności zapewnienia anonimowości podpisu fromularz jest uwierzytelniany podpisem XMLdSig bez certyfikatu.

10.5. Podsumowanie

W rozdziale 10 przedstawiono trzy koncepcje, opracowane przez autora, integracji formularzy elektronicznych z łańcuchami bloków obejmujące również zaprojektowany przez autora sposób weryfikacji ważności certyfikatów z zastosowaniem dedykowanych procedur obsługi dla różnych implementacji łańcuchów bloków.

Pierwsza koncepcja polega na wykorzystaniu technologii usług sieciowych do udostępniania interakcji z łańcuchami bloków w postaci wymiany standardowych komunikatów protokołu SOAP.

W koncepcji drugiej zastosowano rozbudowywanie składni i funkcji aplikacji użytkownika formularza o dedykowane implementacje interakcji formularzy z określonymi łańcuchami bloków.

Koncepcja trzecia zawiera opracowany przez autora mechanizm uruchamiania kodów wykonywalnych zapisanych jako dane w formularzu w ramach jego logiki. Mechanizm ten umożliwia komunikację z różnymi łańcuchami bloków oraz z różnymi systemami PKI opartymi na łańcuchach bloków.

Powyższe koncepcje uwzględniają weryfikację ważności certyfikatów przez aplikację z wykorzystaniem dedykowanych procedur dla wskazanych systemów PKI opartych na łańcuchach bloków. Jednocześnie podkreślono, że dla podpisów anonimowych (bez certyfikatów) nie jest konieczne korzystanie z infrastruktury PKI.

Reasumując można stwierdzić, że zaprojektowany formularz elektroniczny i aplikacja użytkownika zapewniają możliwość wspomagania realizacji transakcji w sposób bezpieczny z wykorzystaniem różnych technik weryfikacji certyfikatu z użyciem zarówno scentralizowanych systemów PKI, jak i systemów rozproszonych opartych na łańcuchach bloków.

11. Transakcje wspomagane formularzem elektronicznym

Formularz elektroniczny jako narzędzie wspomagające realizację transakcji powinien spełniać następujące wymagania:
— logika formularza musi być adekwatna do sposobu realizacji transakcji,
— musi działać w rozproszonym, heterogenicznym organizacyjnie i technologicznie środowisku,
— musi zapewnić pełne bezpieczeństwo realizowanych transakcji.

W rozwiązaniach opisanych w rozdziale 9 formularz elektroniczny spełnia te wymagania poprzez:
— swoją strukturę wyrażającą trójwarstwową naturę formularza,
— element wykonawczy jakim jest aplikacja użytkownika,
— dodatkowe oprogramowanie wspomagające integrację formularza z usługami i systemami stron realizacji transakcji.

W kolejnych podrozdziałach zostanie przedstawiony sposób wykorzystania formularzy elektronicznych spełniających powyższe wymagania.

11.1. Cykl życia formularza elektronicznego

Implementacja formularza we wspomaganiu realizacji transakcji wymaga systematycznego podejścia i obejmuje następujące etapy:

A. **Analizę** procesu realizacji transakcji, na który składają się:
   a) działania,
   b) wymieniane dane,
   c) strony (ludzie i systemy) biorące udział w transakcji, odpowiedzialne za realizację wskazanych działań,
   d) wskazania zakresu wymiany danych pomiędzy stronami transakcji;

B. **Projekt** kolejnych kroków/działań realizowanej transakcji, który musi być zgodny z ustaleniami poczynionymi w etapie A. W ramach projektu należy:
   a) szczegółowo określić zakres danych, ich rodzaj oraz strony za nie odpowiedzialne,
   b) określić właściwości atrybutów statycznych i dynamicznych komponentów graficznych w taki sposób, aby odzwierciedlały one nie tylko znaczenie danych w transakcji, ale również aktualny stan realizacji transakcji,
   c) zdefiniować reguły poprawności danych dla pól formularza w relacji do wartości danych zapisanych w innych polach formularza,
   d) określenie sposób wymiany danych pomiędzy stronami i systemami,
   e) przyporządkować podpisy do danych (lub grup danych), które odzwierciedlają odpowiedzialność stron;

C. **Implementację**, która obejmuje wykonanie formularza oraz oprogramowania z nim związanego, a także prace integracyjne zgodnie z projektem opisanym w etapie B. Efektem działań wykonanych w tym etapie są:
   a) formularz w postaci pliku o strukturze XML zgodnej ze składnią opisaną w schemacie XSD,
   b) oprogramowanie (np. aplikacja serwerowa) i jego integracja z systemami wewnętrznymi organizacji;
D. **Wdrożenie** formularza do realizacji transakcji przy spełnieniu warunku koniecznego polegającego na wykonaniu wymaganych podpisów zabezpieczających formularz, czyli:
   a) opcjonalne podpisy zasobów zewnętrznych,
   b) opcjonalne podpisy właścicieli odpowiedzialnych za środowisko realizacji transakcji,
   c) obowiązkowy podpis formularza, który zabezpiecza cały formularz i kontrasygnuje wymienione wyżej podpisy o ile one istnieją.

W zaprezentowanym powyżej „cyklu życia" formularza uwzględniono jego integrację ze zautomatyzowanymi systemami i usługami stron biorących udział w transakcji. Możliwe jest jednak wykorzystanie formularzy w sposób ograniczony do korzystania jedynie z usług sieciowych poczty elektronicznej. W tym przypadku usługa może być wykorzystana jedynie do transferu dokumentów pomiędzy stronami, przy czym poczta elektroniczna może być zintegrowana z logiką formularza w sposób bardziej wyrafinowany, co opisano w podrozdziale 9.2.4 w przykładzie implementacji „akcji" budowania i wysyłania wiadomości poczty elektronicznej.

11.2. Sterowanie realizacją transakcji

Odpowiednie wykorzystanie warstwy logiki formularza poprzez zastosowanie:
— statycznych i dynamicznych atrybutów graficznych komponentów formularza omówionych w podrozdziałach 9.2.1 i 9.2.2,
— „działań" przedstawionych w podrozdziale 9.2.4,
— systemu podpisów omówionym w podrozdziale 9.2.5,
— wyrażeń zaprezentowanych w podrozdziale 9.2.6,
pozwala na zaprojektowanie i zaimplementowanie formularza z możliwością sterowania przebiegiem realizacji transakcji.

W praktyce polega to na takim zaprojektowaniu formularza, aby dane (pola i komponenty) odzwierciedlały poszczególne kroki realizowanej transakcji, przy czym wartości atrybutów tych komponentów należy zdefiniować w zależności od aktualnego stanu (etapu) realizowanej transakcji, określanego z wykorzystaniem wyrażeń odwołujących się do wartości zapisanych w polach formularza.

Praktyczna implementacja sterowania przebiegiem realizacji transakcji za pomocą formularza polega na wykorzystaniu kolorów, aktywacji/dezaktywacji oraz ukrywaniu/pokazywaniu fragmentów formularza (np. atrybut grupy komponentów) z jednoczesnym dostosowaniem treści wyświetlanych tekstów i tytułów komponentów (np. napis na przycisku) oraz wyboru odpowiedniej akcji za pomocą wyrażeń do aktualnego i przewidywanego stanu realizacji transakcji.

Przykładem formularza sterującego wykonaniem transakcji może być formularz przeznaczony do korespondencji pomiędzy komornikami skarbowymi a urzędem skarbowym. Transakcja ta polega na:

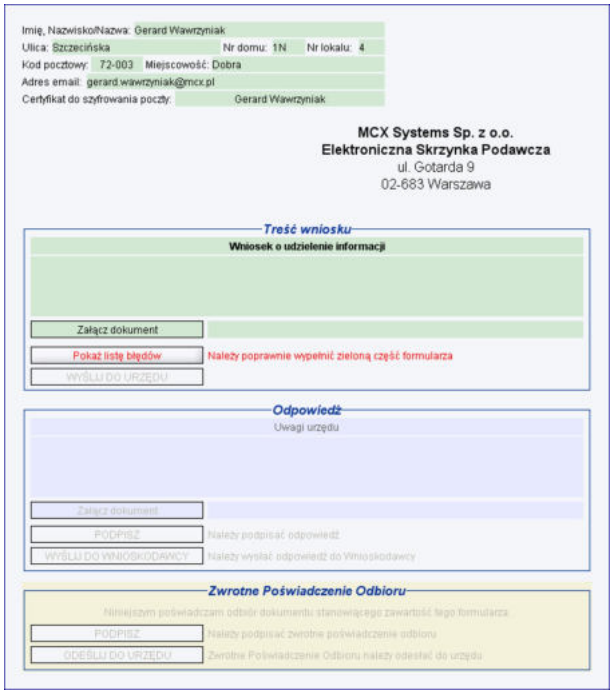
Krok 1. zbudowaniu, podpisaniu i wysłaniu zapytania komornika do urzędu skarbowego,
Krok 2. przygotowaniu, podpisaniu i wysłaniu odpowiedzi urzędu skarbowego do komornika,
Krok 3. podpisaniu i wysłaniu do urzędu Zwrotnego Potwierdzenia Odbioru.

Komunikacja od komornika do urzędu jest realizowana z użyciem usługi sieciowej zaimplementowanej z wykorzystaniem protokołu SOAP (Krok 1, Krok 3), a komunikacja od urzędu do komornika (Krok 2) odbywa się z wykorzystaniem szyfrowanej poczty elektronicznej. Wszystkie kroki transakcji są uwierzytelniane podpisem elektronicznym.

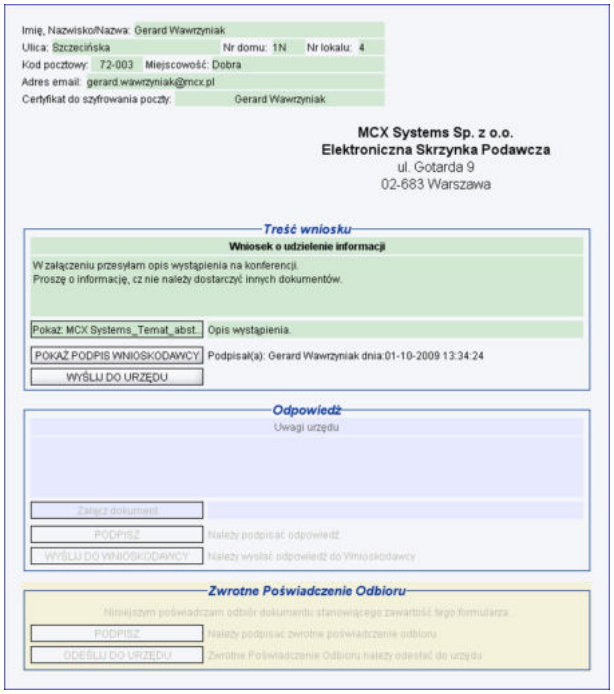
Szczegółowy przebieg powyższej transakcji jest następujący:

1. Komornik wprowadza do pól tekstowych formularza swoje dane adresowe, na które składają się: imię i nazwisko, adres pocztowy, adres e-mail oraz własny certyfikat X.509 do szyfrowania poczty, który wprowadza do pola „Załącznik". Następnie wprowadza treść zapytania do części formularza oznaczonej jako „Treść wniosku" oraz załącznik do komponentu „Załącznik", którego zawartością jest dowolny plik (np. PDF). Gdy wszystkie dane zostaną poprawnie wprowadzone, komunikat o błędzie jest ukrywany, a przycisk umożliwiający wyświetlenie błędów zmienia swój tytuł na „Podpisz". Po wykonaniu podpisu aktywny staje się przycisk wysłania. Ten krok pokazano na rysunku 11.1.
2. Po naciśnięciu przycisku „Wyślij do urzędu" aplikacja buduje wiadomość SOAP, której ciało („body") stanowi plik formularza, i wysyła tą wiadomość na adres serwera zapisanego w strukturze „akcji" wysyłania wiadomości SOAP. Aplikacja na serwerze urzędu po weryfikacji formularza (np. weryfikacja podpisów) wprowadza dane dotyczące odbioru dokumentu: nazwę podmiotu, datę i czas doręczenia, datę i czas otrzymania oraz wykonuje w formularzu podpis Urzędowego Potwierdzenia Odbioru. Zmiana stanu danych w formularzu powoduje wyświetlenie wprowadzonych danych oraz uwidacznia przycisk akcji wyświetlającej informacje o podpisie urzędu. Formularz z Urzędowym Potwierdzeniem Odbioru jest zwracany do aplikacji klienta urzędu jako synchroniczna odpowiedź wiadomości SOAP. Jednocześnie formularz jest wysyłany elektroniczną pocztą wewnętrzną do urzędnika odpowiedzialnego za udzielenie odpowiedzi (zgodnie z logiką zapisaną w deskryptorze procesów aplikacji serwerowej). Na tym kończy się proces dostarczania wniosku do urzędnika.
3. Urzędnik odpowiedzialny za przygotowanie odpowiedzi otrzymuje formularz, w którym aktywna jest, dotychczas nieaktywna, część formularza przeznaczona na wprowadzenie odpowiedzi. Po wprowadzeniu poprawnych danych odpowiedzi następuje zmiana przeznaczenia przycisku „Pokaż listę błędów" na „Podpisz", a po wykonaniu podpisu następuje aktywacja przycisku „ Wyślij do wnioskodawcy". Po naciśnięciu tego przycisku aplikacja buduje wiadomość poczty elektronicznej, po czym szyfruje ją z użyciem klucza publicznego odczytanego z certyfikatu zapisanego w „Załączniku" i wysyła ją do wnioskodawcy. Ten krok pokazano na rysunku 11.2.
4. Po wysłaniu odpowiedzi do wnioskodawcy następuje aktywacja tej części formularza, która odpowiada za obsługę Zwrotnego Poświadczenia Odbioru (ZPO). Wnioskodawca naciska aktywny przycisk „Podpis" w wyniku czego, po wykonaniu podpisu aktywowany jest przycisk „Odeślij do urzędu" i po wysłaniu ZPO do urzędu wyświetlana jest odpowiedź urzędu na pytanie wnioskodawcy, a w formularzu wyświetlana jest informacja o obiorze wiadomości z urzędu. Ten krok pokazano na rysunku 11.3.

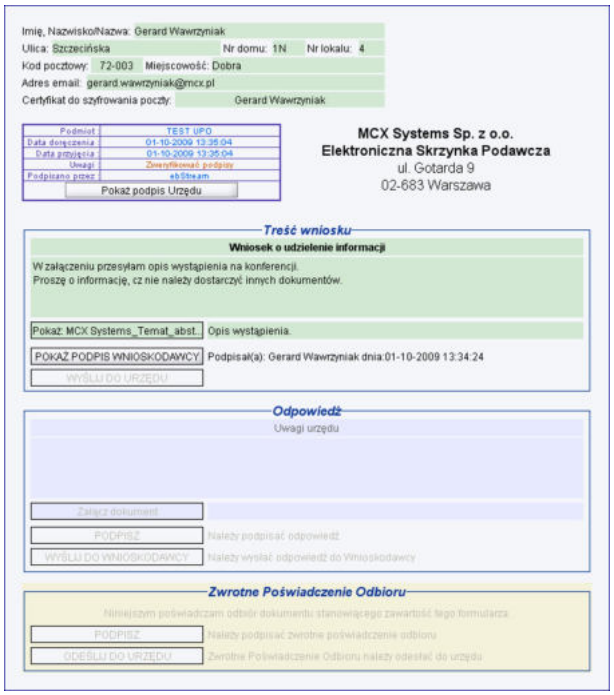
Po tak zrealizowanej transakcji obie strony posiadają udokumentowany i potwierdzony podpisami zapis jej przebiegu.



(a) Formularz gotowy do wypełnienia

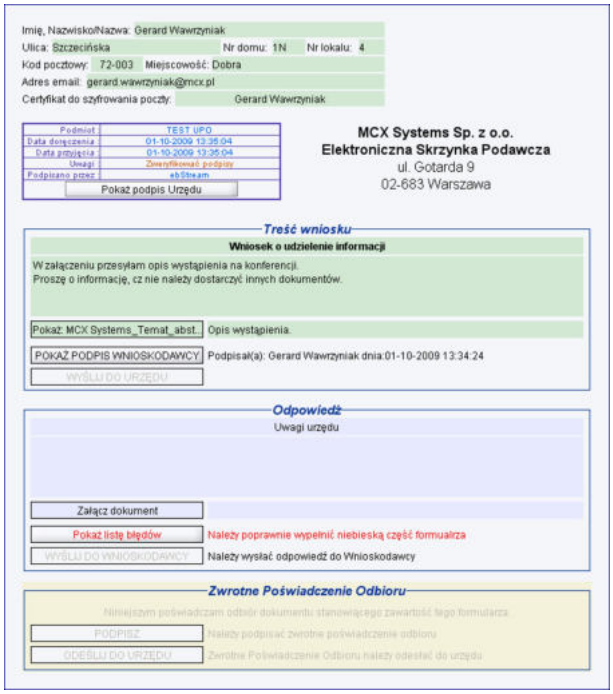


(b) Formularz wypełniony i podpisany, gotowy do wysłania

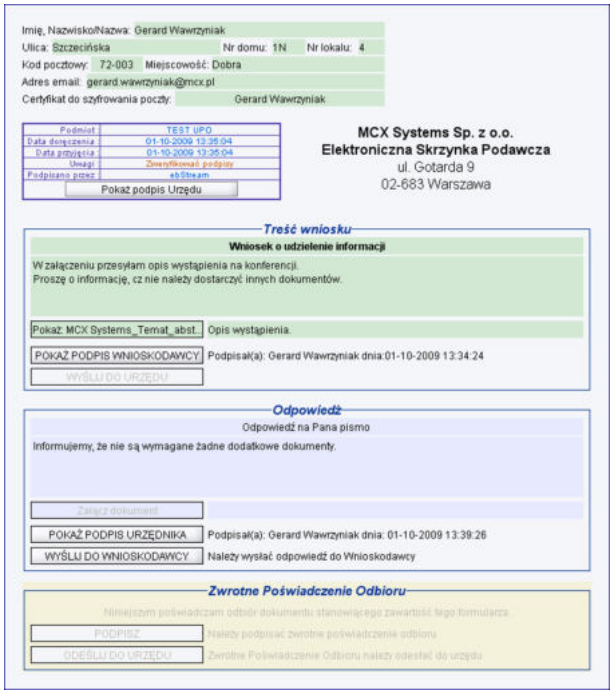


(c) Urzędowe Potwierdzenie Odbioru

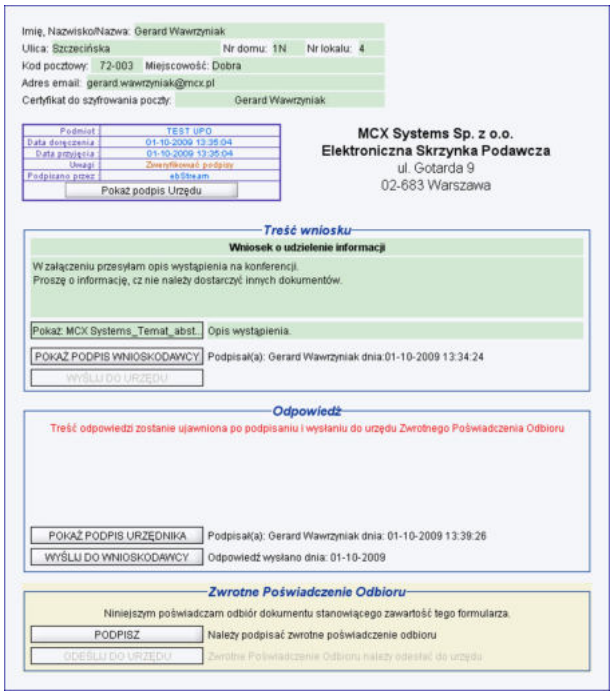
Rys. 11.1: Sterowanie transakcją: Komornik – przygotowanie, podpisanie i wysłanie wniosku do urzędu



(a) Formularz gotowy do wypełnienia

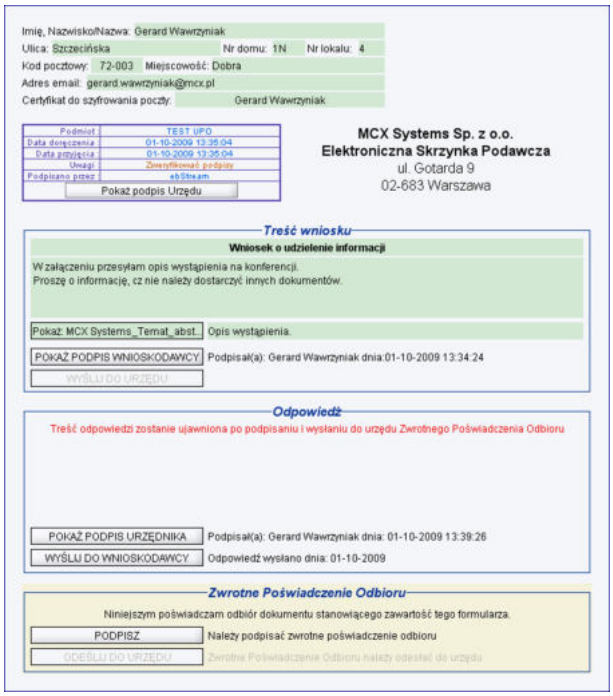


(b) Formularz wypełniony i podpisany, gotowy do wysłania

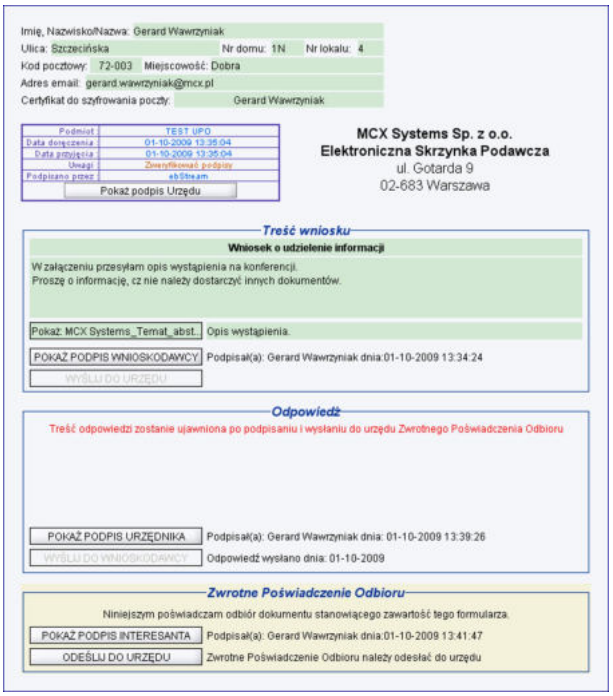


(c) Formularz wysłany - wyświetlony status wysłania

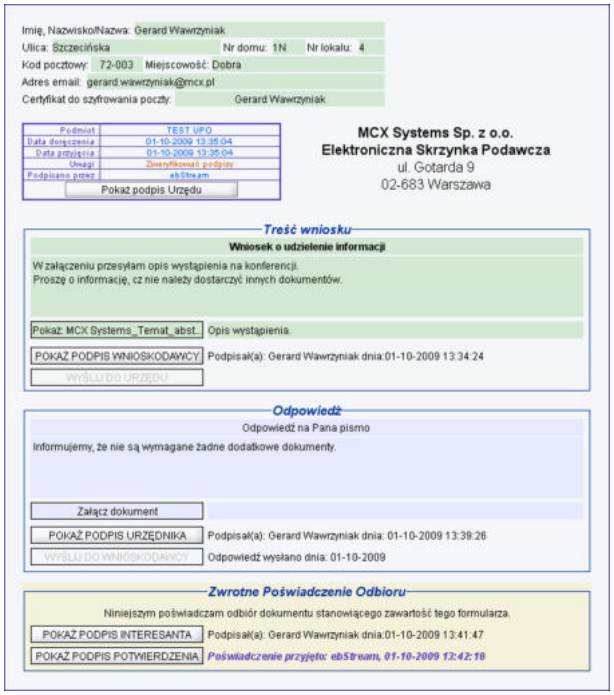
Rys. 11.2: Sterowanie transakcją: Urząd – przygotowanie, podpisanie i wysłanie odpowiedzi do komornika



(a) Odpowiedź przysłana do komornika



(b) Podpisane Zwrotne Poświadczenie Odbioru, gotowe do odesłania



(c) Formularz potwierdzony, gotowy do odczytu

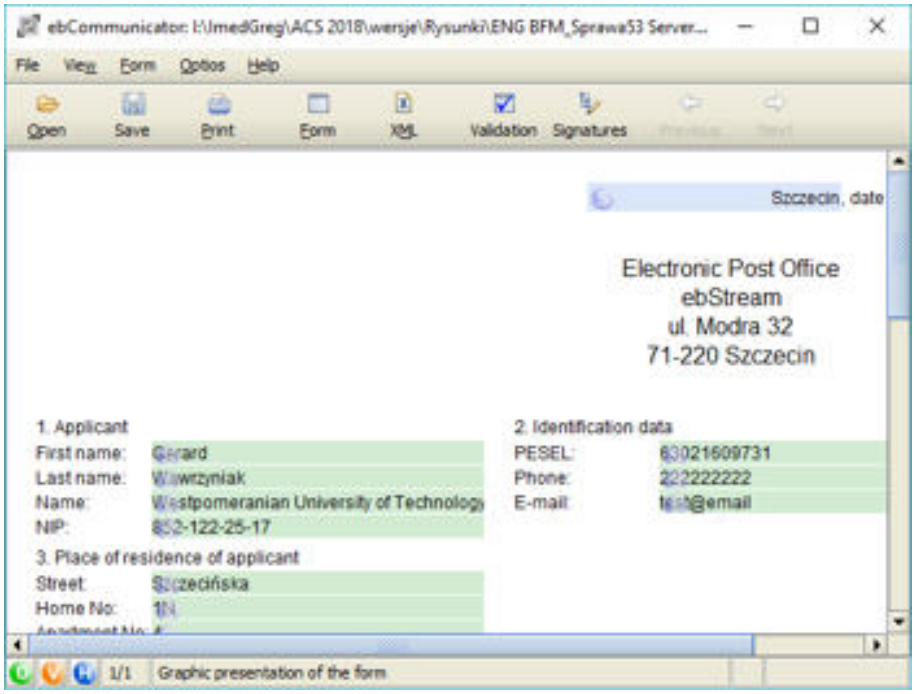
Rys. 11.3: Sterowanie transakcją: Potwierdzenie otrzymania odpowiedzi na wniosek

11.3. Bezpieczeństwo transakcji

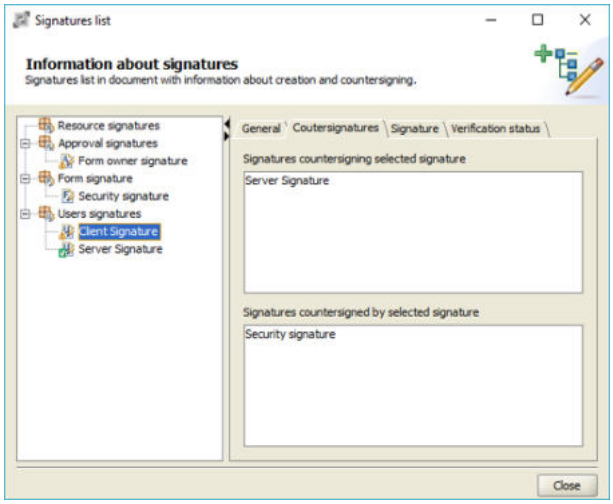
11.3.1. Podpisy w transakcji

Bezpieczeństwo formularza zapewnione za pomocą systemu podpisów elektronicznych zostało omówione w podrozdziale 9.2.5. Podpisy są integralną częścią struktury formularza i poprzez ich relacje z różnymi częściami formularza stanowią element jego logiki, która została opisana w podrozdziale 9.3.3. Dynamiczne odzwierciedlanie stanu podpisów w atrybutach komponentów graficznych i „działań" jest realizowane przez wyrażenia binarne (funkcje obsługi obiektów PKI). W ten sposób logika formularza pozwala na dostosowanie obsługi podpisów do specyfiki realizowanej transakcji. Pokazano to na rysunkach 11.1, 11.2 i 11.3 w podrozdziale 11.2 jako dostosowanie napisów, tytułów przycisków i wykonywanych działań po naciśnięciu przycisku. Natomiast na rysunkach 11.4 i 11.5 pokazano implementację dwuetapowej obsługi podpisów w transakcji. Pierwszy etap to podpisanie formularza (wniosku) przez Klienta, a drugi etap to wygenerowanie i podpisanie Urzędowego Potwierdzenia Odbioru (tego wniosku) przez serwer Elektronicznej Skrzynki Podawczej.

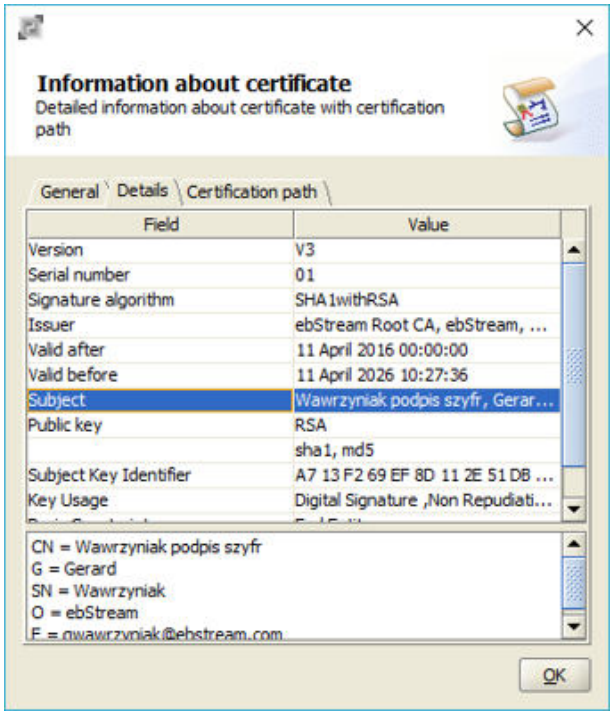
Z powyższego wynika, że w formularzu elektronicznym powstają kolejne podpisane zapisy realizowanej transakcji. Podpisy te stanowią niezaprzeczalny dowód realizacji poszczególnych kroków, który jest dostępny dla stron transakcji.



(a) Wypełniony i podpisany formularz

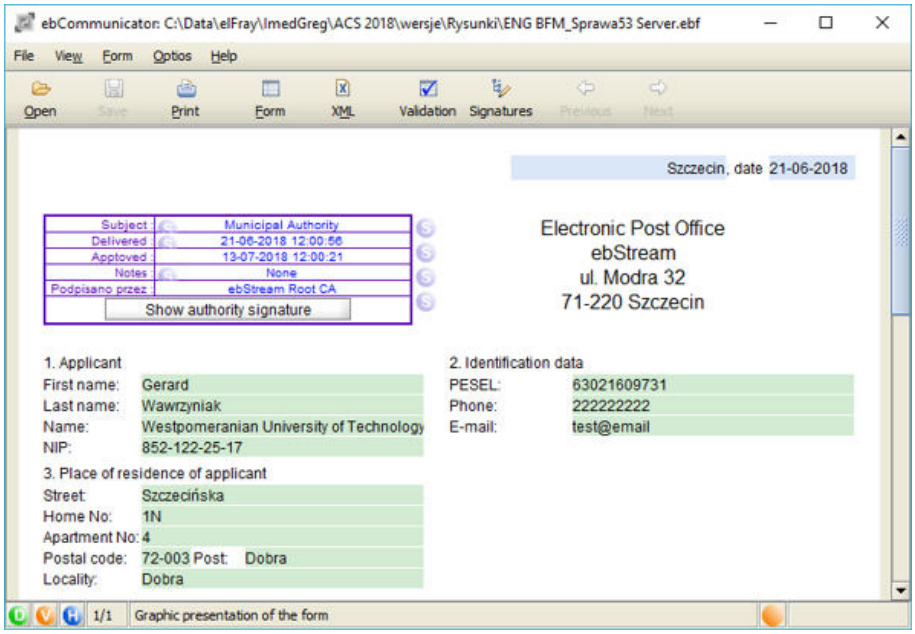


(b) Podpisy w formularzu – podpis klienta

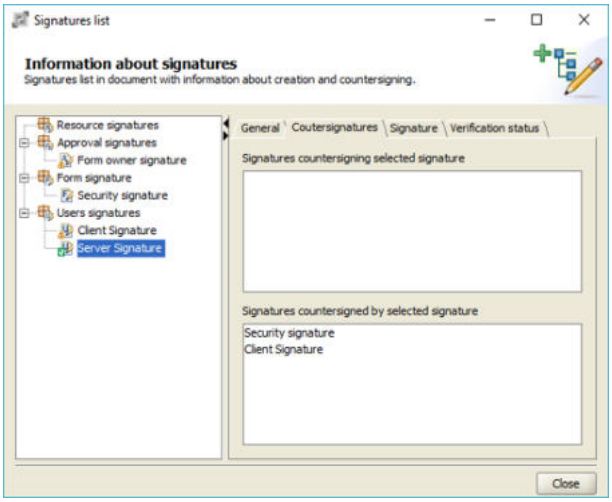


(c) Certyfikat podpisu klienta

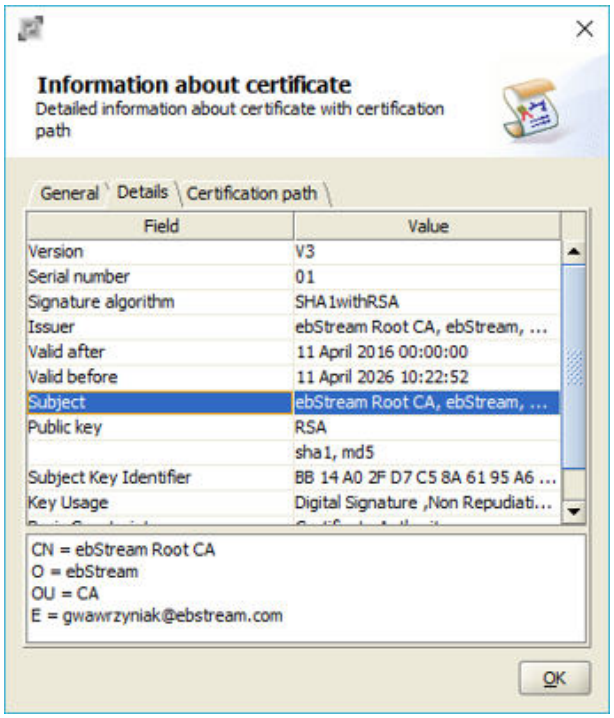
Rys. 11.4: Obsługa podpisów w formularzu – wysłanie formularza do ESP



(a) Formularz z aktywowanym i podpisanym komponentem UPO



(b) Podpisy w formularzu – podpis serwera (UPO)



(c) Certyfikat podpisu serwera

Rys. 11.5: Obsługa podpisów w formularzu – odbiór UPO

11.3.2. Zabezpieczenie protokołu SOAP

Z punku widzenia bezpieczeństwa protokołu SOAP istotne są następujące parametry:

1. `@authenticationMethod`, który wskazuje sposób uwierzytelnienia połączenia, może przyjąć jedną z trzech wartości:
   None – brak uwierzytelnienia,
   Basic – uwierzytelnienie jednostronne (serwera) realizowane przez aplikację użytkownika, a tożsamość serwera jest weryfikowana z użyciem certyfikatu serwera umieszczonego w komponencie `<Atachment>`, który jest zabezpieczony co najmniej podpisem formularza,
   Client-Cert – uwierzytelnienie dwustronne (klient, serwer) z wykorzystaniem certyfikatu serwera zapisanego w strukturze formularza (`<Attachment>`); nawiązanie połączenia jest realizowane przez aplikację użytkownika, która dostarcza danych do uwierzytelnienia,
2. `@verifyHost` – informacja, czy serwer ma być weryfikowany podczas nawiązywania połączenia,
3. `<hostCertificate>` - wyrażenie typu binarnego wskazującego na serwer usługi SOAP.

Wykorzystanie komponentu `<Attachment>` do przechowywania certyfikatów pozwala na użycie zaufanego certyfikatu (to znaczy będącego elementem podpisanej logiki formularza) w procedurze nawiązywania połączenia z serwerem. Ponadto użycie funkcji odczytywania danych certyfikatu i dynamicznych atrybutów napisów umożliwia wyświetlanie szczegółów certyfikatu w treści formularza.

Przykładem realizacji „działania” wysyłania wiadomości z wykorzystaniem protokołu SOAP jest struktura formularza pokazana na wydruku 11.1.

```
<soap-msg authenticationMethod="None" soapAction="" verifyHost="false">
  <hostCertificate>
    <mimeReference>ServerCertificate</mimeReference>
  </hostCertificate>
  <endPointURL>
    <stringReference>Office.ServiceAddress</stringReference>
  </endPointURL>
  <request>
    <body signed="never"/>
  </request>
  <response>
    <body signed="never">
      <ns1:ebFormResponse>
        <message/>
        <date/>
      </ns1:ebFormResponse>
    </body>
  </response>
  <confirmationText signed="never">
    <stringXPath>//ns1:ebFormResponse/message/node()</stringXPath>
  </confirmationText>
</soap-msg>
```

Wydruk 11.1: Implementacja obsługi wiadomości w protokole SOAP

Poszczególne parametry wywołania tego „działania” obejmują:

— wskazanie certyfikatu X.509 jako referencji do elementu identyfikowanego przez `<ServerCertificate>`, który służy do weryfikacji serwera usługi sieciowej,
— wskazanie adresu URL usługi jako referencji do elementu `Office.ServiceAddress`, która jest posadowiona na serwerze usług sieciowych,
— zawartość elementu żądania wiadomości SOAP (`<SOAP-Body>\<request>`), w tym przykładzie pusty element `<body>` oznacza, że zawartością żądania jest cały formularz,

— zawartość elementu odpowiedzi (`SOAP-Body\response`), w tym przykładzie element odpowiedzi z przestrzeni (`ns1:ebFormResponse`) jest modyfikowany przez serwer usług sieciowych i zwracany do aplikacji formularza, gdzie jest wyświetlony jako zawartość wskazywana przez element `<stringXPath>` w `<confirmationText>`.

Składnia formularza dla parametrów użycia protokołu SOAP została pokazana na rysunku A.12–21 w Załączniku A.

11.3.3. Bezpieczna poczta elektroniczna

Struktura definiowania wiadomości pocztowej przewiduje możliwość zdefiniowania listy adresatów ze wskazaniem dla każdego z nich certyfikatu z kluczem publicznym do szyfrowania wiadomości. Wykonanie „działania” `<mail>` polega na utworzeniu wiadomości pocztowej, następnie podpisaniu jej, zaszyfrowaniu i wysyłaniu do wszystkich odbiorców, dla których wskazano certyfikat z kluczem do szyfrowania. Wysyłana poczta może być podpisana niezależnie od podpisów w formularzu stanowiącym załącznik do tej poczty.

Na wydruku 11.2 przedstawiono przykład zdefiniowania wiadomości pocztowej w strukturze formularza, która zawiera:

— wskazanie odbiorcy (`test@domain.com`),
— informacje o statusie wysyłania poczt – wysyłanie poczty ma charakter asynchroniczny i może wyć wykonane wielokrotnie, dlatego należy przechowywać informacje o ostatniej próbie wysyłania oraz statusie wykonania tej operacji (`<isSent>`, `<sentDate>`, `<sentAddress>`),
— zawartość pola `<Subject>` – jest ona budowana w oparciu o definicję wykorzystującą wyrażenia typu `(stringExpression)`,
— zawartość pola `<Body>`, czyli treść wiadomości pocztowej, która jest budowana z wykorzystaniem definicji opartej na wyrażeniach typu String (`stringExpression`).

```
<mail>
<receipients>
  <receipient>
    <address>
      <stringValue>test@domain.com</stringValue>
    </address>
    <mailStatus>
      <isSent>
        <booleanValue signed="never">
          true
        </booleanValue>
      </isSent>
      <sentDate>
        <dateValue signed="never">
          2019-06-11
            </dateValue>
      </sentDate>
      <sentAddress>
        <stringValue signed="never"/>
      </sentAddress>
    </mailStatus>
  </receipient>
</receipients>
<subject>
  <stringValue>
    Mail subject Test it can be composed using String expressions
  </stringValue>
</subject>
<body>
  <stringValue>
    Mail body Test it can be composed using String expressions
  </stringValue>
</body>
<attachments>
```

```
  <attachment>
36    <name>
        <stringValue>AnyFile.TXT</stringValue>
38    </name>
  </attachment>
40 </attachments>
</mail>
```

Wydruk 11.2: Przykład zbudowania i wysyłania wiadomości mail (SMTP)

Aplikacja użytkownika interpretuje przedstawioną na wydruku 11.2 strukturę używając aktualnych wartości parametrów i dostarcza zbudowaną wiadomość SMTP do serwera wysyłającego pocztę.

W rozdziale 9.2.4 omówiono, a na rysunku 9.6 pokazano, formularz demonstracyjny budowania i wysyłania wiadomości e-mail przy użyciu wszystkich elementów poczty. Schemat składni „akcji” `<mail>` do obsługi wysłania wiadomości zaprezentowano na rysunku A.12–20 w Załączniku A.

11.3.4. Podsumowanie

Z rozważań przedstawionych w podrozdziale 11.3 wynika, że realizacja transakcji z wykorzystaniem formularzy elektronicznych jest zabezpieczana zgodnie z logiką zapisaną w formularzu. Na mechanizmy zabezpieczające składają się:

1. system podpisów w formularzu (podrozdział 11.3.1), które zabezpieczają dokumenty: zewnętrzne, „wzór formularza” to znaczy całą jego strukturę bez wprowadzanych danych oraz dane użytkowników (stron transakcji),
2. uwierzytelnianie stron w protokole SOAP na trzy sposoby (brak uwierzytelnienia, uwierzytelnienie serwera, uwierzytelnienie dwustronne), które zostały omówione w podrozdziale 11.3.2,
3. zabezpieczenie generowanych przez formularz wiadomości pocztowych poprzez podpisywanie wiadomości i jej szyfrowanie do odbiorców zgodnie z opisem w podrozdziale 11.3.3.

Przedstawione mechanizmy zabezpieczeń zapewniają wysoki poziom bezpieczeństwa informacji towarzyszącej realizacji transakcji z wykorzystaniem formularza elektronicznego.

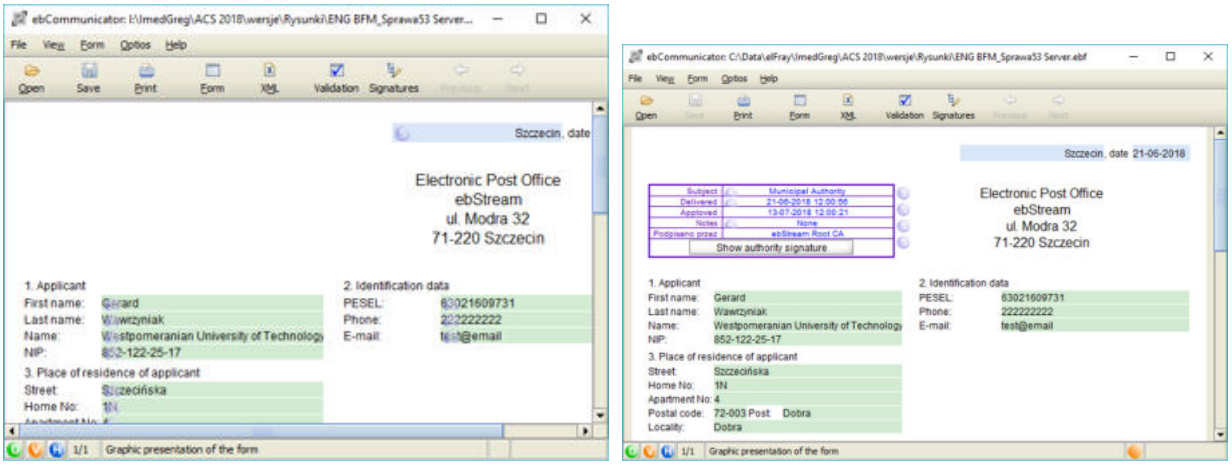
11.4. Rodzaje wspomaganych transakcji

Formularz elektroniczny może wspomagać różnorodne transakcje. Sposób wspomagania uzależniony jest od charakteru transakcji i dostępnego środowiska technicznego. W tym podrozdziale autor skoncentruje się na omówieniu przykładów sposobów zaimplementowania wspomagania realizacji transakcji prostych (jednoetapowych) oraz złożonych (wieloetapowych).

11.4.1. Transakcja prosta

Prosta transakcja jednoetapowa (jednokierunkowa) polega na skutecznym prawnie przesłaniu formularza od nadawcy do serwera adresata z otrzymaniem potwierdzenia odbioru. Zarówno treść formularza, jak i potwierdzenie odbioru, są podpisane podpisami użytkowników o zawartości odpowiednio – treść dokumentu, potwierdzenie odbioru. Formularz po wypełnieniu i podpisaniu przez nadawcę (rys. 11.6a) jest wysyłany do serwera

odbiorcy, na którym są wprowadzane dane związane z potwierdzeniem odbioru, a następnie są one elektronicznie podpisane. Formularz z podpisanym potwierdzeniem odbioru jest zwracany do nadawcy jako odpowiedź w protokole SOAP. Formularz z potwierdzeniem odbioru pokazano na rysunku 11.6b. Możliwa jest również realizacja transakcji z wykorzystaniem jedynie poczty elektronicznej jak pokazano na rysunku 11.7.

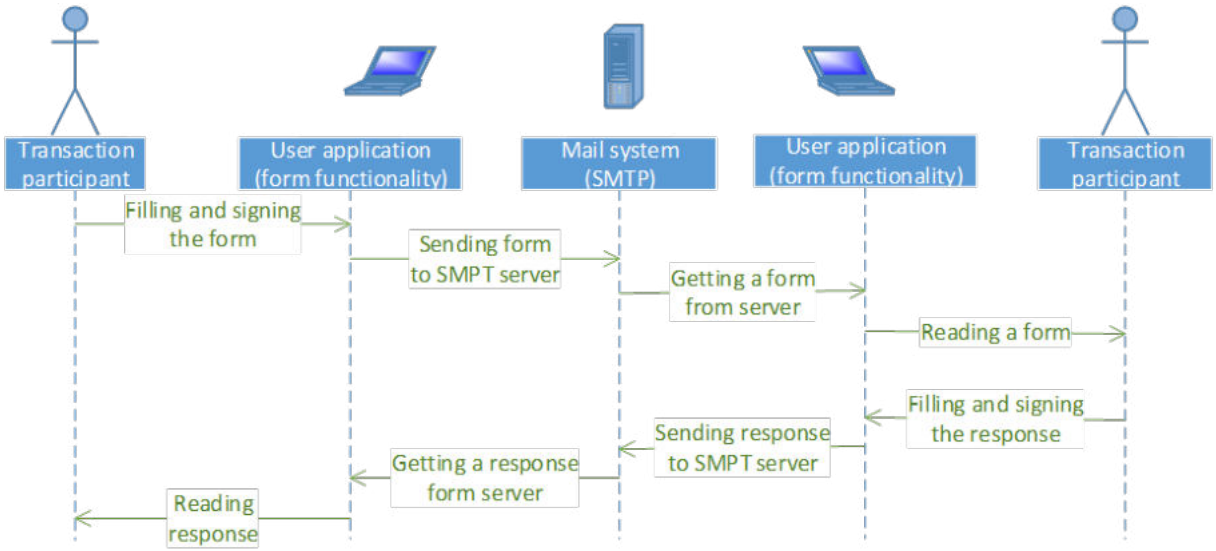


(a) Przykład formularza przed wysłaniem do Elektronicznej Skrzynki Podawczej

(b) Ten sam formularz po wysłaniu i otrzymaniu podpisanego potwierdzenia odbioru

Rys. 11.6: Transakcja prosta – wysłanie dokumentu i odbiór potwierdzenia odbioru

Z powyższych rozważań wynika, że do realizacji transakcji wystarczy formularz elektroniczny z aplikacją oraz dostęp do usług poczty elektronicznej.

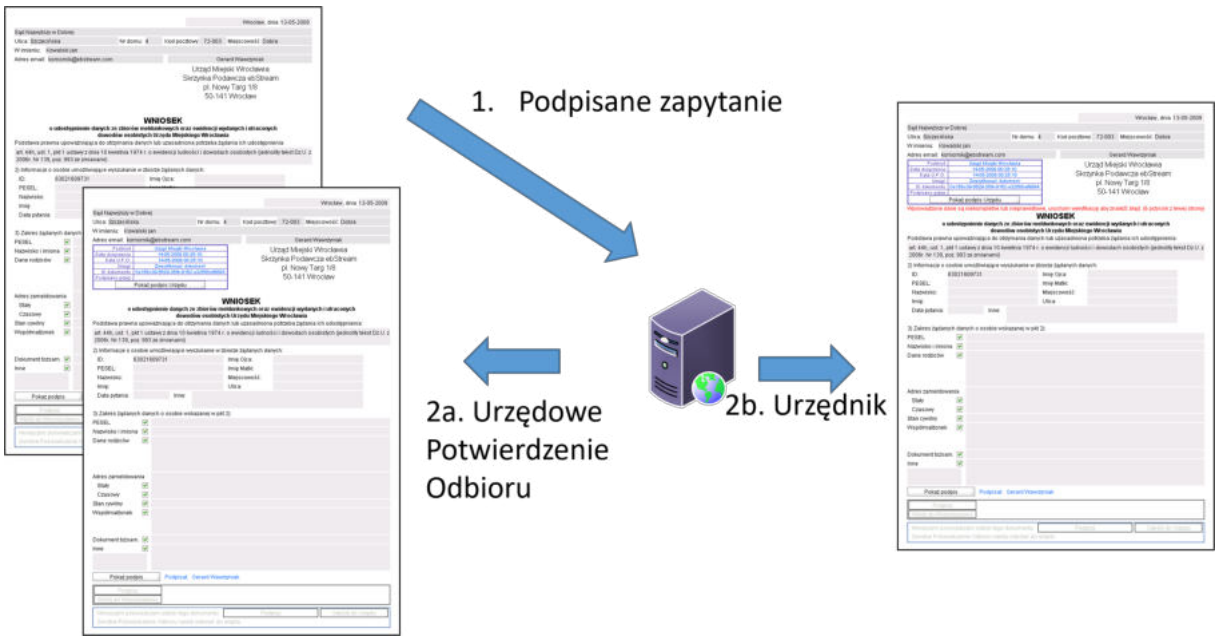


Rys. 11.7: Implementacja transakcji z wykorzystaniem poczty elektronicznej

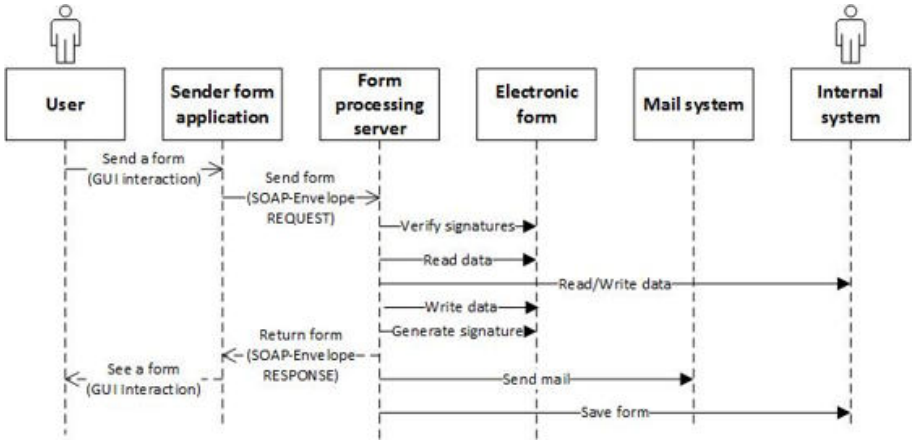
11.4.2. Transakcja złożona

Transakcja złożona składa się z więcej niż dwóch czynności (kroków). Przykładem takiej transakcji może być udzielenie odpowiedzi urzędu miasta sądowi na złożone zapy-

tanie z zachowaniem skuteczności prawnej poszczególnych kroków (etapów) transakcji. Na rysunku 11.8 pokazano czynności związane z przygotowaniem zapytania, które obejmuje wskazanie przedmiotu zapytania i podpisanie wniosku oraz wysłanie go do serwera Elektronicznej Skrzynki Podawczej (ESP). Po wygenerowaniu i podpisaniu danych Urzędowego Potwierdzenia Odbioru (UPO) przez aplikację serwera, formularz jest odsyłany jako odpowiedź SOAP do aplikacji użytkownika, który może odczytać dane UPO i sprawdzić jego podpis. Po wykonaniu tego etapu obie strony posiadają niezaprzeczalne dowody wysłania i odebrania wniosku.



(a) Schemat wykonywanych czynności



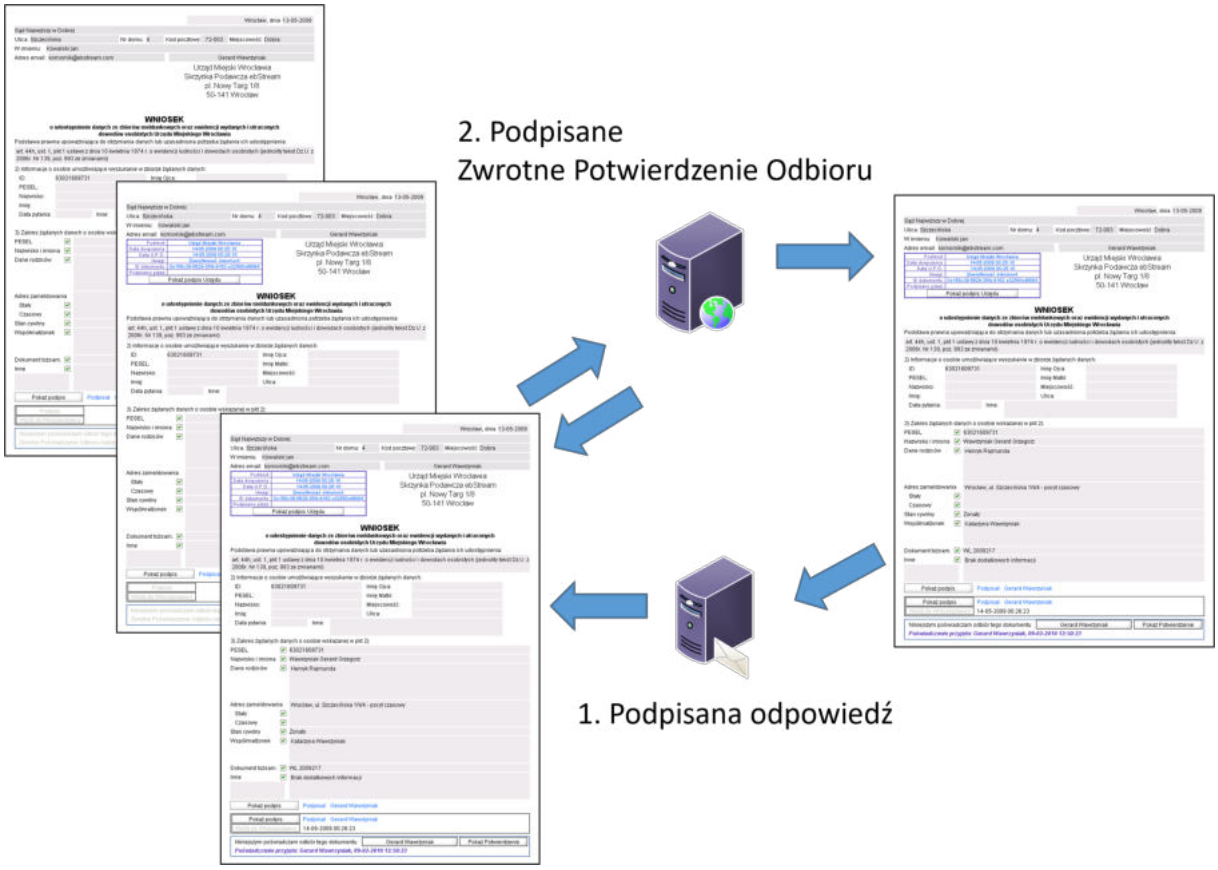
(b) Diagram sekwencji – wykorzystanie protokołów SOAP i SMTP

Rys. 11.8: Etap pierwszy – wysłanie zapytania (sąd – urząd)

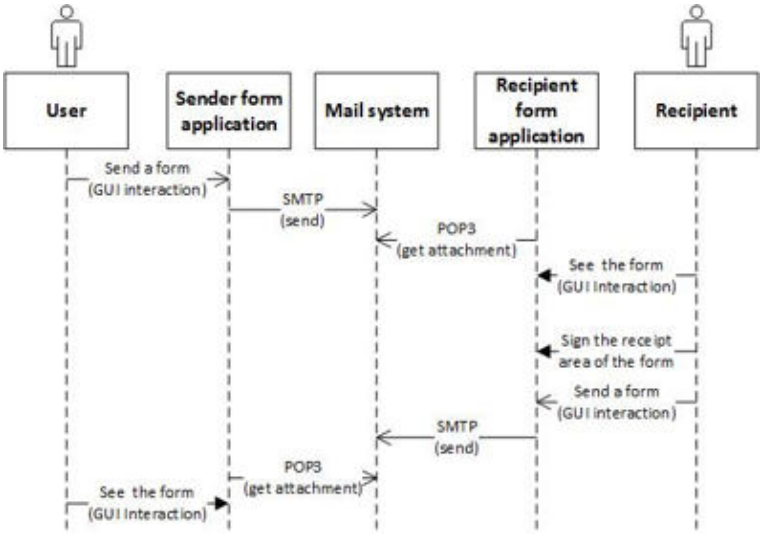
Po otrzymaniu wniosku urzędnik przygotowuje odpowiedź w zakresie zgodnym z wnioskiem, po czym ją podpisuje i wysyła do sądu poprzez naciśnięcie stosownego, aktywnego przycisku. Odpowiedź jest wysyłana za pośrednictwem serwera SMTP wskazanego w logice formularza. Odbiorca po otrzymaniu odpowiedzi podpisuje tzw. Zwrotne Poświadcze-

nie Odbioru i wysyła je do serwera Elektronicznej Skrzynki Podawczej, który przekazuje formularz do urzędnika obsługującego sprawę.

Na rysunku 11.9 pokazano przebieg wykonywanych czynności związanych z udzieleniem odpowiedzi.



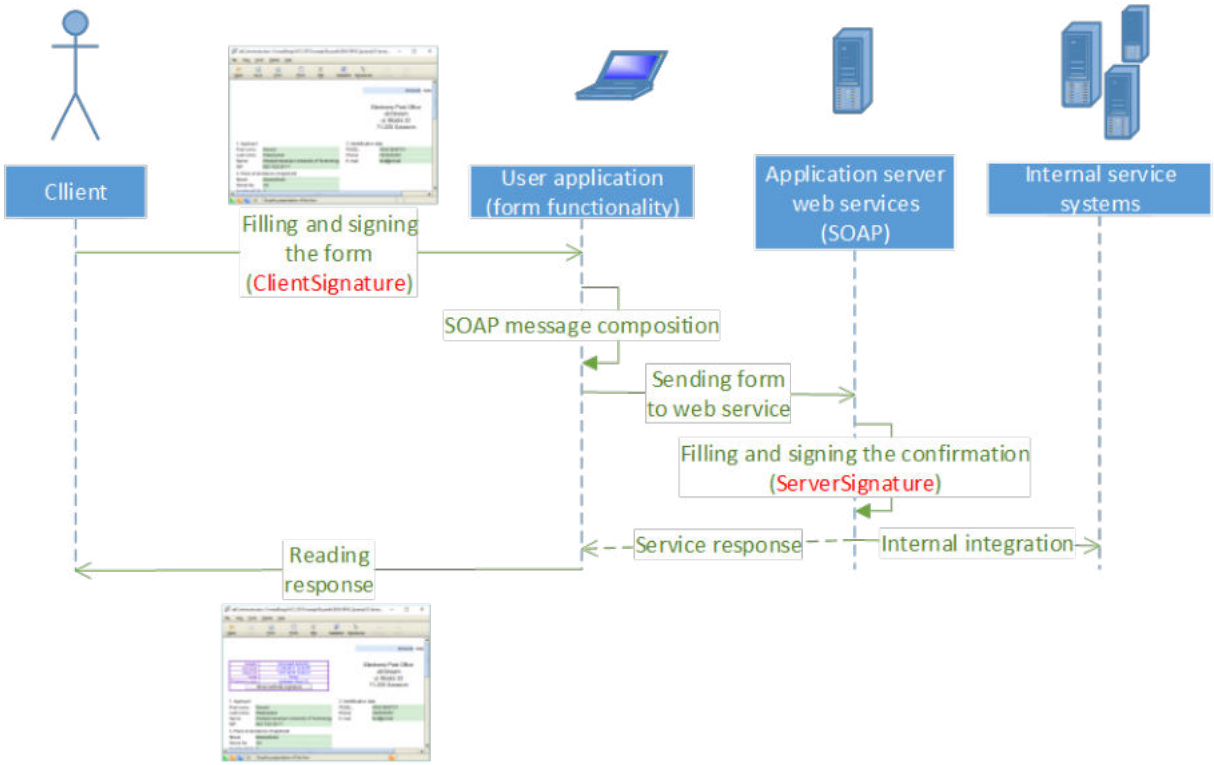
(a) Schemat wykonywanych czynności



(b) Diagram sekwencji – wykorzystanie SMTP

Rys. 11.9: Etap drugi – uzyskanie odpowiedzi na wniosek (urząd – sąd)

Z kolei na rysunku 11.10 przedstawiono przykład realizacji transakcji z wykorzystaniem protokołu SOAP.



Rys. 11.10: Realizacja transakcji z wykorzystaniem protokołu SOAP

11.4.3. Integracja formularza z systemami

Realizacja transakcji może wymagać integracji formularza z różnymi systemami. Poprawność takiej integracji w prezentowanej koncepcji formularza znalazła potwierdzenie w licznych implementacjach. Przykładem takich implementacji są następujące rozwiązania:

1. ElDok (system obiegu dokumentów dla administracji) – formularze wykorzystano w około 50 urzędach gmin, powiatów i miast we współpracy z ZETO Lublin, ZETO Jelenia Góra, ZETO Olsztyn, Maxux Net Communication,
2. system obiegu dokumentów w administracji – integracja formularzy we współpracy z ZETO Koszalin,
3. ICOR (system informatyczny Urzędu Miasta Szczecin) – wdrożono około 80 formularzy w systemie Multiplan Szczecin,
4. Lotus Notes – wdrożono około 30 formularzy w Urzędzie Miasta Wrocław i Sądzie Okręgowym we Wrocławiu,
5. faktury w Carrefour – generowanie i dystrybucja faktur, obsłużono tysiące dokumentów we współpracy z IMP Polska i InforSys,
6. faktoring dla banku w Warszawie – stworzono i wdrożono rozwiązanie, w którym podpisane faktury stanowiły podstawę rozliczeń faktoringowych,
7. korespondencja pomiędzy komornikami skarbowymi a urzędem skarbowym – formularze obsługi transakcji wdrożono w Drugim Urzędzie Skarbowym w Koszalinie,
8. Symfonia – integracja faktur elektronicznych z systemem we współpracy z IMP Polska i SageSymfonia,

9. obieg formularza zakupów w Uniwersytecie Warmińsko-Mazurskim oparty na SMTP,
10. ELO Digital Office – integracja formularzy z systemem obiegu dokumentów we współpracy z ELO Gorzów Wielkopolski,
11. SignForm – współpraca technologiczna.

W wymienionych wyżej wdrożeniach formularza elektronicznego nie były zaimplementowane następujące cechy i koncepcje:

— eliminacja podatności podpisów XML polegającej na możliwości manipulacji zawartości podpisów w dokumentach XML (podrozdział 5.5). Problem został rozwiązany przez rozszerzenie funkcji warstwy logiki i zastosowanie tzw. szablonu podpisu jako rozwiązanie ogólne dla dokumentów oraz jako rozwiązanie specjalizowane dla formularzy elektronicznych ([206, 208]),
— integracja logiki formularza z łańcuchami bloków w celu rozszerzenia możliwości w zakresie wspomagania realizacji różnych transakcji rozdział 10, podrozdział 11.5,
— możliwość uruchamiania kodów wykonywalnych (skompilowane klasy *java*) przechowywanych jako dane w polach-komponentach formularza (<attachment>) (*Załącznik*) (podrozdział 10.3). Daje to możliwość wykorzystania zewnętrznych kodów wykonywalnych, be konieczności modyfikacji składni formularza i aplikacji do jego użytkowania.

11.5. Łańcuchy bloków w transakcjach wspomaganych formularzami elektronicznymi

Możliwości i cechy łańcuchów bloków zostały opisane w rozdziale 6, natomiast w rozdziale 9 omówiono cechy i możliwości formularzy elektronicznych. Z kolei w rozdziale 10 zaprezentowano zaproponowane przez autora sposoby integracji formularzy z łańcuchami bloków. W niniejszym podrozdziale zaprezentowano autorski sposób wykorzystania łańcuchów bloków w formularzach elektronicznych wspomagających realizację transakcji.

Łańcuch bloków, jako technologia rejestrów rozproszonych (*DLT – Distributed Ledger Technology*), w realizacji transakcji może:

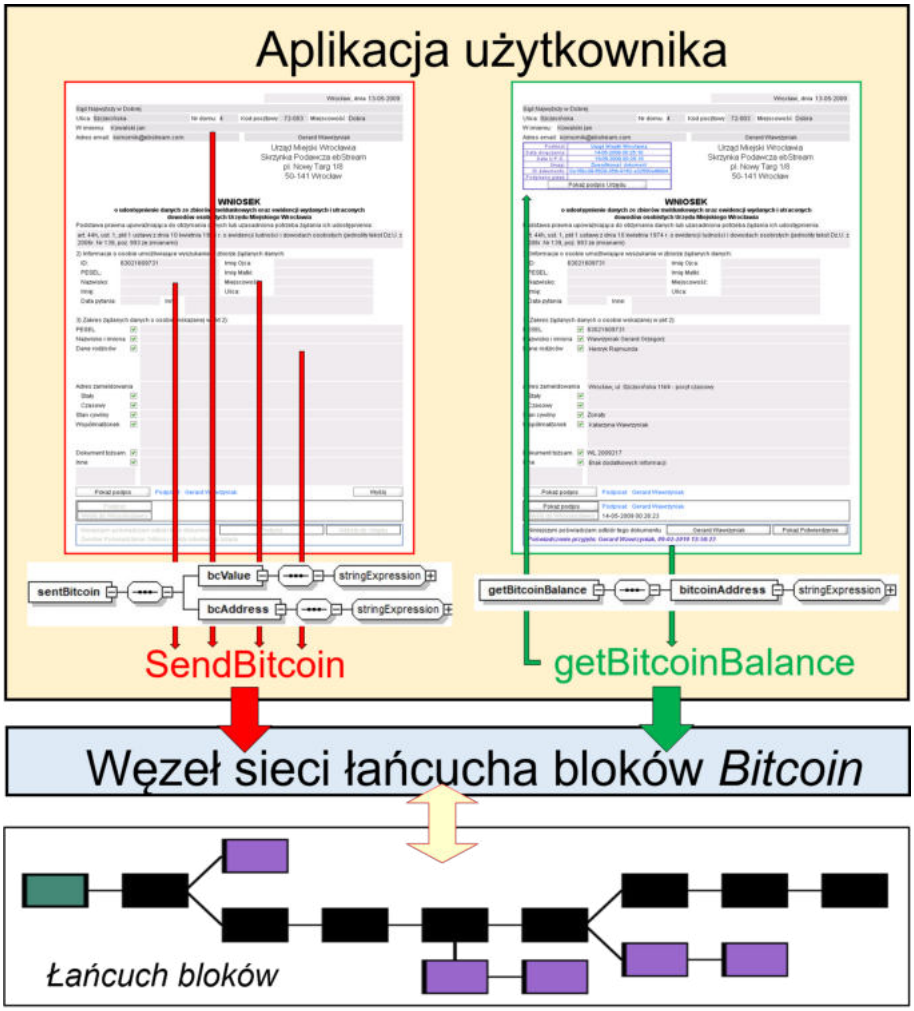
— być samodzielną operacją lub jej istotną częścią – przykładem są kryptowaluty (np. *Bitcoin*) i realizowane z ich użyciem transakcje (nie jest możliwe realizowanie tego typu transakcji bez korzystania z łańcuchów bloków),
— wspomagać realizację transakcji – przykładem są: koncepcja uwierzytelniania dokumentów cyfrowych lub śledzenie zapisów w systemie przewarzania transakcji, o których wspomniano w rozdziale 6 (zastosowanie łańcuchów bloków nie jest niezbędne, gdyż w tych rozwiązaniach technologia rejestrów rozproszonych może być zastąpiona systemami centralnych baz danych).

Jak wielokrotnie wspomniano w pracy zadaniem mechanizmów wbudowanych do formularza elektronicznego jest:

— udostępnianie aktualnych wartości wyrażeń odwołujących się do wskazywanych wartości pól formularza (funkcje),
— uruchamianie „akcji”, które wykonują określone działania nie zwracając wartości.

Uwzględniając powyższe mechanizmy implementacja współpracy formularza elektronicznego z łańcuchem bloków, a dokładniej mówiąc lokalnym węzłem sieci, polega na wskazaniu w formularzu komponentów graficznych prezentujących dane związane z realizowaną transakcją i powiązanie tych danych z argumentami wskazanych funkcji i „akcji” odpowiedzialnych za wykonanie zaprogramowanych w aplikacji użytkownika algorytmów współpracy. Taka współpraca jest naturalna z punktu widzenia logiki formularza elektronicznego

i węzła sieci łańcucha bloków. Na rysunku 11.11 pokazano przykładową implementację formularza sterującego przebiegiem transakcji z wykorzystaniem łańcucha bloków *Bitcoin* w trakcie jej realizacji.



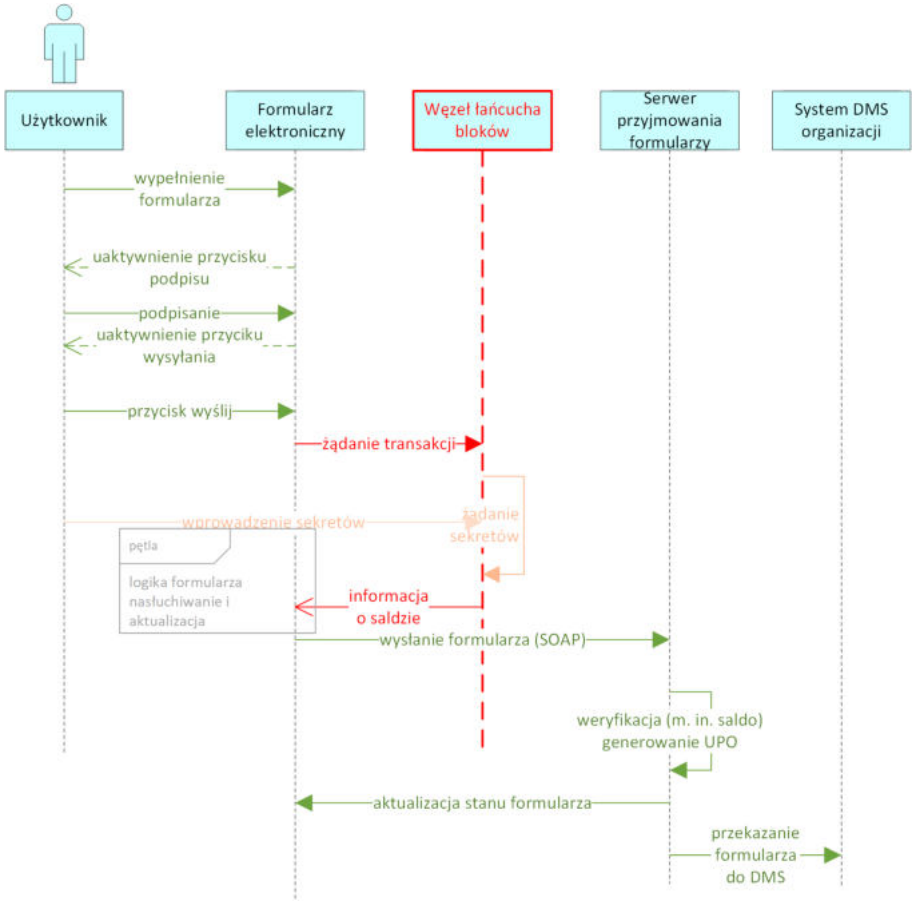
Rys. 11.11: Transakcja z udziałem łańcucha bloków *Bitcoin*

Z kolei na rysunku 11.12 przedstawiono diagram sekwencji działań dla tej transakcji, na którą składają się:

— wypełnienie i podpisanie formularza wniosku przez petenta,
— automatyczna (po podpisaniu) realizacja transakcji, sterowana przez logikę formularza, obejmująca:
  — zbudowanie żądania `<SendBitcoin>` z argumentami o wartościach zdefiniowanych wyrażeniami wiążącymi je z polami formularza,
  — wysłanie żądania realizacji transakcji do lokalnego węzła sieci łańcucha bloków (*Blockchain*).

Po zmianie stanu wartości funkcji aktualnego stanu salda, wskazanego adresem zapisanym w logice formularza `<GetBalance>`, następuje:
— sprawdzenie wysokości wniesionej opłaty i jej zgodności z wymaganiami,
— aktywowanie przycisku `<Button>`, którego naciśnięcie powoduje kontynuację realizacji transakcji,

— zbudowanie i wysłanie komunikatu SOAP, zawierającego podpisany i „opłacony” formularz, do serwera adresata,
— sprawdzenie poprawności formularza (wysokość wniesionej opłaty) oraz wpisanie i elektroniczne podpisanie przez serwer parametrów *Urzędowego Potwierdzenia Odbioru* (UPO),
— zwrócenie formularza z podpisanym UPO jako odpowiedzi `<Response>` na żądanie do aplikacji użytkownika,
— zaprezentowanie przez aplikację użytkownika zaktualizowanego formularza reprezentującego aktualny stan realizacji transakcji.



Rys. 11.12: Sekwencja działań w transakcji z płatnością *Bitcoin*

11.6. Interoperacyjność formularzy elektronicznych

Interoperacyjność jest jedną z najważniejszych właściwości usług publicznych, a w szczególności dokumentu elektronicznego, który jest elementem wykonawczym w sferze wymiany informacji. Rolę interoperacyjności w odniesieniu do formularza elektronicznego autor omówił w pracy [204]. Natomiast w pracy [207] wymienił obszary, w jakich należy zapewnić interoperacyjność dokumentu, czyli niezależność od obszarów zastosowania, od dojrzałości systemów informatycznych oraz od formatu dokumentu. Z kolei w artykule [205] autor przeanalizował interoperacyjność formularza w realizacji transakcji z uwzględnieniem następujących obszarów: format danych, aspekty bezpieczeństwa, wymiana komunikatów, usługi przetwarzania, interoperacyjność obiektów interakcja człowiek-maszyna, interoperacyność transgraniczna i kulturowa (np. wielojęzyczność). Zaprezentował w nim sposoby spełnienia wymagań interoperacyjności w tych obszarach. Dokonał również oceny poziomu dojrzałości interoperacyjnej, która wykazała, że formularz elektroniczny osiąga „poziom czwarty – dostosowany". Poziom ten oznacza, że organizacja korzystająca z tego rozwiązania jest w stanie dynamicznie zmieniać się i współdziałać w heterogenicznym pod względem technicznym, wielojęzykowym i wielokulturowym środowisku partnerów.

Podniesienie poziomu interoperacyjności umożliwia implementację komunikacji z rejestrami rozproszonymi w postaci łańcuchów bloków. Chociaż istnieje niewiele implementacji łańcuchów bloków o zasięgu globalnym, to mechanizmy formularza dają taką możliwość poprzez odpowiednią jego rozbudowę i funkcję umieszczania w formularzu (w komponencie *Załącznik* `<Attachment>`) kodów wykonywalnych (podrozdział 10.3).

11.7. Podsumowanie

W rozdziale 11 zaprezentowano rozważania związane z wykorzystaniem formularzy elektronicznych jako narzędzia wspomagającego realizację różnych transakcji. Zwrócono uwagę, że wymaga to starannego przygotowania procesu implementacji zgodnie z przedstawionym „cyklem życia" formularza (podrozdział 11.1) z uwzględnieniem jego roli w kontekście organizacyjno-prawnym, proceduralnym oraz środowiska technicznego.

W podrozdziale 11.2 pokazano sposób wykorzystania cech formularza do sterowania przebiegiem transakcji, a w podrozdziale 11.3 szczególną uwagę poświęcono bezpieczeństwu transakcji wspomaganych formularzami elektronicznymi. Ponieważ system podpisów w formularzu omówiono szczegółowo w podrozdziale 9.2.5, to w tej części pracy skoncentrowano się na bezpieczeństwie protokołu SOAP oraz podpisywanej i szyfrowanej poczcie elektronicznej (SMTP).

W podrozdziale 11.4 zademonstrowano realizację różnych transakcji z wykorzystaniem formularzy elektronicznych. Pokazano przykład prostej (jednoetapowej) transakcji, w której użytkownik wchodzi w interakcję z usługą z użyciem formularza elektronicznego. Przykład był punktem wyjścia do prezentacji transakcji złożonej z udziałem użytkownika i serwera. Transakcja ta wymaga zastosowania kilku protokołów komunikacyjnych (SOAP, SMTP). Natomiast w podrozdziale 11.4.3 zaprezentowano produkcyjne systemy, z którymi zintegrowano formularze elektroniczne.

W podrozdziale 11.5 zademonstrowano przykład formularza wspomagającego realizację transakcji z wykorzystaniem łańcucha bloków.

W ostatnim podrozdziale 11.6 omówiono wyniki opublikowanych prac autora, w których wykazano wysoki poziom interoperacyjności formularzy elektronicznych w różnych jej obszarach oraz wysoki poziom jego dojrzałości interoperacyjnej.

12. Wnioski – realizacja celu i weryfikacja hipotezy badawczej

Formularz elektroniczny na poziomie struktur, funkcji oraz jego roli w transakcji został zaprezentowany w rozdziałach 9 i 11. Formularz ten nie tylko wiernie odzwierciedla tradycyjny formularz papierowy, ale może funkcjonować w świecie wirtualnym, gdyż:
— jest łatwy do przetwarzania w zróżnicowanym technologicznie środowisku,
— może włączyć do realizacji transakcji wiele stron (ludzi i maszyn),
— odzwierciedla różne wymagania transakcji,
— transakcja może być sterowana logiką formularza,
— może być stosowany w sieciach o dowolnym stopniu rozproszenia,
— zapewnia bezpieczeństwo również w bezserwerowej infrastrukturze klucza publicznego,
— jest łatwy w realizacji prac wdrożeniowo-integracyjnych implementujących wspomaganie transakcji.

W tabeli 12.1 przedstawiono elementy struktury i funkcji formularza, które zapewniają spełnienie wymienionych wyżej postulatów dotyczących korzystania z formularza, a zarazem wykazano, że kształtują one możliwości realizacji transakcji o określonych cechach.

Tabela 12.1: Odzwierciedlenie konstrukcji formularza w cechach transakcji

Konstrukcja formularza (rozdział 9)	Cechy transakcji (rozdział 11)
Format dokumentu elektronicznego: — struktura XML (podrozdział 9.2) — zdefiniowana składnia (XSD) (Załącznik A)	Interoperacyjność w zakresie wymiany danych: — łatwa identyfikacja elementów z danymi — kontrola typów danych (podrozdział 9.2.3) — interoperacyjność danych (rozdział 4, podrozdział 11.6)

kontynuacja tabeli na następnej stronie

Tabela 12.1 – kontynuacja tabeli z poprzedniej strony

Konstrukcja formularza (rozdział 9)	Cechy transakcji (rozdział 11)
Trójwarstwowa natura formularza (podrozdział 9.3): — warstwa danych (podrozdział 9.3.1) — warstwa prezentacji (podrozdział 9.3.2) — warstwa logiki (podrozdział 9.3.3)	Sposoby korzystania z formularza w transakcji: — bezpieczne przechowywanie i przesyłanie danych — automatyczne odczytywanie i zapisywanie danych — wyświetlanie dokumentu – prezentacja znaczenia danych dla człowieka (włączenie człowieka do realizacji transakcji jako aktywnej strony) — definiowanie reguł transakcji w formularzu (np.: poprawność danych, podpisy, dynamiczny wygląd formularza) — przetwarzanie warstwy danych formularza przez komponent serwerowy bez udziału człowieka lub użycie dedykowanej biblioteki obsługi formularza (podrozdział 9.4.3) — obsługa formularza we wszystkich warstwach (obsługa danych, wizualizacja i obsługa logiki) z użyciem dedykowanej aplikacji użytkownika (podrozdział 9.4)
Koncepcja wyrażeń w formularzu: — koncepcja wyrażeń różnych typów (podrozdział 9.2.6) — dynamiczne i statyczne atrybuty komponentów graficznych (podrozdziały 9.2.1, 9.2.2)	Sterowanie realizacją transakcji (podrozdział 11.2): — definiowanie stanów realizacji transakcji, które odzwierciedlają aktualny stan danych w polach formularza — dostosowanie sposobu wyświetlania komponentów graficznych do aktualnego stanu realizacji transakcji (np.: wyświetlanie lub chowanie komponentów, zmiana kolorów, dostępności i edytowalności pól) — realizacja wieloetapowych transakcji z udziałem wielu stron (maszyn i ludzi) (podrozdział 11.4) — aktywny udział ludzi w realizowanych transakcjach (podrozdział 11.6)

kontynuacja tabeli na następnej stronie

Tabela 12.1 – kontynuacja tabeli z poprzedniej strony

Konstrukcja formularza (rozdział 9)	Cechy transakcji (rozdział 11)
„Działania/akcje” definiowane za pomocą dedykowamych wyrażeń (podrozdział 9.2.4) i/lub uruchamiane w wyniku decyzji użytkownika (strony transakcji). „Działania” te mogą dotyczyć: — operacji wewnętrznych związanych z wykonywaniem procedur obsługi elementów formularza — operacji zewnętrznych związanych z realizacją wymiany danych z otoczeniem – protokoły SOAP i SMTP	Wykonywanie logiki wyrażeń dla działań definiowanych w formularzu w efekcie decyzji użytkownika co umożliwia: — sterowanie realizacją transakcji (podrozdział 11.2) — wykorzystanie różnych usług w ramach realizacji jednej transakcji (podrozdział 11.2) — integrację formularza z systemami (podrozdział 11.4.3) z wykorzystaniem różnych protokołów — interoperacyjność systemów, usług i obiektów (rozdział 4, podrozdział 11.6) — realizację usług w środowisku o dowolnym stopniu centralizacji, a w szczególności bez udziału serwerów usługowych – na przykład tylko z wykorzystaniem poczty do wymiany dokumentów albo zwykłej wymiany plików (podrozdział 9.2.4, rysunek 9.6, podrozdział 11.3.3)
Bezpieczeństwo formularza (podrozdział 5) zapewnione przez: — stosowanie standardowego przetwarzania podpisów (rozdział 5.2.3), — wbudowane typy podpisów formularza (podrozdział 5.3, 9.2.5) jako element logiki formularza (podrozdział 9.3.3), — eliminację podatności referencji podpisu XML (podrozdział 5.5)	Bezpieczeństwo transakcji obejmujące: — interoperacyjność procedur przetwarzania podpisów (podrozdział 5.2.3) — interoperacyjność tożsamości elektronicznej (podrozdział 11.6) — przyporządkowanie fragmentów formularza odzwierciedlających kroki transakcji do stron za nie odpowiedzialnych (podrozdziały 11.2, 11.3.1) — eliminację podatności referencji (podrozdział 9.2.5) — zabezpieczanie podpisami użytkowników fragmentów formularza odzwierciedlających etapy realizacji transakcji (podrozdział 11.3.1)
Bezpieczeństwo protokołów jako elementów wyrażeń „akcji” (podrozdział 9.2.4) zapewnione przez: — wymianę wiadomości za pomocą poczty elektronicznej – protokół SMTP — wymiana komunikatów w usługach sieciowych – protokół SOAP	Bezpieczeństwo transakcji realizowanych w środowisku technicznym zapewnione przez: — standardowe mechanizmy zabezpieczeń w protokole SOAP z uwzględnieniem uwierzytelnienia jednostronnego, dwustronnego oraz bez uwierzytelnienia (podrozdział 11.3.2) — mechanizmy zabezpieczeń wymienianych wiadomości poczty elektronicznej – protokół SMTP (podrozdział 11.3.3)

Konstrukcja formularza i cechy transakcji, które zostały przeanalizowane w tabeli 12.1, pozwaliły na:

— zrealizoanie celu badawczego, czyli *opracowanie modelu transakcji rozproszonych sterowanych formularzami elektronicznymi* (podrozdział 8.1),

— pozytywne zweryfikowanie hipotezy badawczej, zgodnie z którą *zastosowanie formularza elektronicznego z własną logiką umożliwi realizację transakcji rozproszonych w heterogenicznym środowisku bez używania infrastruktury serwerowej i udziału scentralizowanej infrastruktury PKI* (podrozdział 8.2).

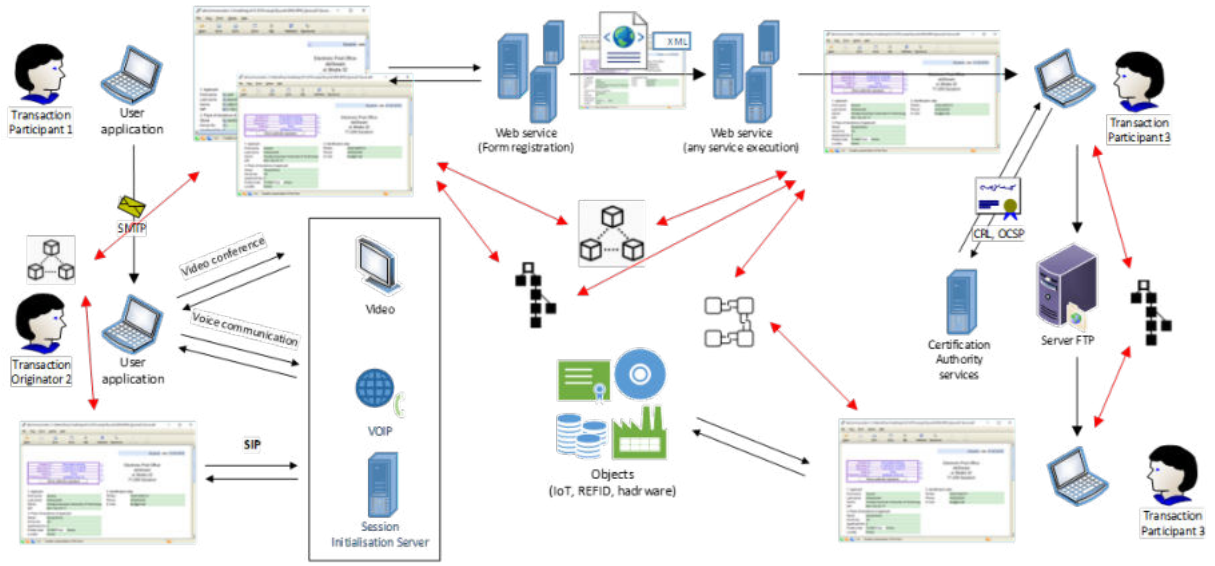
12.1. Model transakcji rozproszonych

Nie istnieje jeden określony typ transakcji, której realizacja może być wspomagana przez formularz elektroniczny. Przyczyną tego jest zróżnicowanie środowiska, w którym funkcjonują transakcje, pod względem technicznym, organizacyjno-prawnym, proceduralnym, a także stopnia centralizacji (rozproszenia) systemów i usług. Dlatego w pracy zaproponowano, aby model transakcji zdefiniować jako zbiór zasad, cech i możliwości związanych ze stosowaniem oraz implementacją formularza elektronicznego. Zbiór ten określono na podstawie:
— właściwości i funkcji formularza elektronicznego (rozdział 9),
— sposobów jego integracji z łańcuchami bloków (rozdział 10),
— możliwości formularza w zakresie wspomagania realizacji różnych transakcji (rozdział 11).

Model transakcji oraz wskazówki wdrażania opisane w „cyklu życia" formularza (podrozdział 11.1) stanowią podstawę do implementacji i wdrażania formularzy jako mechanizmu wymiany informacji w realizowanych transakcjach. W związku z tym przyjęto, że **model transakcji rozproszonych sterowanych formularzami elektronicznymi składa się z następujących postulatów**, które opracowano na podstawie tabeli 12.1:

I. transakcja może być sterowana logiką zapisaną w formularzu elektronicznym (podrozdział 11.2),
II. w realizacji transakcji w obszarze wymiany dokumentów biorą udział ludzie (strony transakcji) oraz systemy informatyczne i usługi sieciowe (podrozdział 11.6),
III. system wspomagania realizacji transakcji może być zdecentralizowany w dowolnym stopniu – od transakcji realizowanych przez osoby biorące udział w transakcji z wykorzystaniem SMTP, poprzez bardziej zaawansowane użycie poczty i usług sieciowych, po zaawansowane mechanizmy obsługi protokołu SOAP (podrozdział 11.4),
IV. formularz z aplikacją użytkownika może być zintegrowany z różnymi systemami w heterogenicznym technicznie środowisku (podrozdział 9.2.4, rozdział 11),
V. formularz może korzystać ze zróżnicowanych implementacji łańcuchów bloków (rozdział 10, podrozdział 11.5),
VI. formularz elektroniczny zapewnienia wysoki poziomu bezpieczeństwa poprzez użycie podpisów elektronicznych w zakresie struktury formularza (podrozdział 9.2.5) oraz dla mechanizmów realizacji transakcji (podrozdział 11.3),
VII. wspomaganie realizacji transakcji z użyciem formularzy elektronicznych jest implementowane zgodnie z „cyklem życia" formularza (podrozdział 11.1).

Na rysunku 12.1 przedstawiono przykład transakcji odzwierciedlającej model transakcji rozproszonej z wykorzystaniem formularza elektronicznego jako narzędzia wspomagającego jej realizację.



Rys. 12.1: Model transakcji rozproszonej w środowisku heterogenicznym

12.2. Weryfikacja hipotezy badawczej

Przyjęta w pracy hipoteza badawcza (podrozdział 8.2) zakłada, że **skuteczne wspomaganie realizacji transakcji:**

a. rozproszonych (skrajnie zdecentralizowanych),
b. bez używania infrastruktury serwerowej,
c. w heterogenicznym środowisku (zróżnicowanym technicznie, organizacyjnie, kulturowo),
d. bez udziału scentralizowanej infrastruktury PKI,
e. w integracji z łańcuchami bloków,

może być zrealizowane przez użycie **formularza elektronicznego z własną logiką wbudowaną w jego strukturę.**

Hipoteza ta została pozytywnie zweryfikowana poprzez wykazanie, że formularz elektroniczny i wspomagana przez niego transakcja o cechach przeanalizowanych w tabeli 12.1, umożliwiają:

Sterowanie przebiegiem transakcji z wykorzystaniem warstwy logiki;

Realizację rozproszonych transakcji bez infrastruktury serwerowej poprzez wymianę formularzy (bezpiecznych dokumentów) pomiędzy stronami jedynie w oparciu usługę poczty elektronicznej oraz przekazywanie plików formularzy w dowolny inny sposób;

Realizację transakcji w heterogenicznym środowisku dzięki możliwości użycia różnych protokołów komunikacyjnych (np.: SMTP, SOAP, łańcuchy bloków) definiowanej w warstwie logiki formularza, a w szczególności koncepcji wyrażeń „akcji" - „działań";

Weryfikację podpisów bez infrastruktury PKI poprzez wykorzystanie zde-

centralizowanych usług PKI opartych na łańcuchach bloków przechowujących informacje o zmianach statusów certyfikatów;

Zapewnienie bezpieczeństwa transakcji poprzez wykorzystanie systemu podpisów w formularzu oraz mechanizmów bezpieczeństwa wymiany komunikatów (SOAP, SMTP).

Należy podkreślić, że koncepcja formularza elektronicznego zaproponowana przez autora umożliwiła pierwsze w Polsce praktyczne integracje formularzy elektronicznych z produkcyjnymi systemami obiegu dokumentów (podrozdział 11.4.3).

Stanowi to dowód na poprawność koncepcji, a także przyjętych i zaimplementowanych rozwiązań. Na rysunku 12.2 pokazano lokalizacje systemów (urzędy administracji państwowej i samorządowe), z którymi zintegrowano zaprojektowane formularze elektroniczne.

Rys. 12.2: Lokalizacje, w których zintegrowano zaprojektowane formularze elektroniczne z systemami obiegu dokumentów

Warto również nadmienić, że propozycja formularzy elektronicznych z aplikacją do korzystania z nich znalazła uznanie kapituły nagrody EOIF (Elektroniczny Obieg Informacji w Firmie), która przyznała I Nagrodę w Konkursie Złoty Bit za „Najlepszy aktualnie dostępny na terenie Polski produkt software'owy wspomagający systemy obiegu informacji (dokumentów)".

Reasumując należy stwierdzić, że w pracy udowodniono hipotezę o możliwości realizacji transakcji rozproszonych bez użycia infrastruktury serwerowej i scentralizowanej infrastruktury PKI z wykorzystaniem formularzy elektronicznych z własną logiką. Zrealizowano również cel pracy prezentując model transakcji rozproszonych sterowanych formularzami elektronicznymi.

13. Kierunki przyszłych badań

Podczas realizacji pracy, w tym analizy literatury oraz prac implementujących wykorzystanie łańcuchów bloków w formularzu elektronicznym, zidentyfikowano problemy, które mogą stać się przesłanką do rozszerzenia prac badawczych i inżynierskich w tym zakresie. Problemy te dotyczą przede wszystkim:
— relacji pomiędzy formularzami a łańcuchami bloków,
— koncepcji formularzy elektronicznych z punktu widzenia inteligentnych kontraktów,
— zastosowania koncepcji łańcuchów bloków w zabezpieczaniu i uwierzytelnianiu dokumentów, a szczególnie formularzy elektronicznych w realizowanych transakcjach.

13.1. Łańcuchy bloków a formularz elektroniczny

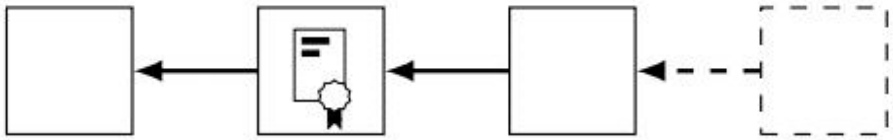
W podrozdziale 6.4 przedstawiono ogólną taksonomię zastosowań łańcuchów bloków oraz kilka przykładów ich zastosowań. Prezentując te zagadnienia przyjęto zasadę eksponowania tych cech łańcuchów bloków, które są uniwersalne i niezależne od branż, w których są stosowane.

Badania przeprowadzone w trakcie realizacji pracy zweryfikowały to podejście, gdyż zauważono szerokie spektrum zastosowań łańcuchów bloków oraz różnorodność technologii, które je implementują. Należy zauważyć, że zarówno technologia łańcuchów bloków, jak i ich zastosowania, znajdują się na wczesnym etapie rozwoju. Konsekwencją tego jest brak zunifikowanych, spójnych, standardowych mechanizmów wymiany komunikatów i interakcji z różnymi implementacjami łańcuchów bloków. Konieczne jest zatem budowanie dedykowanych mechanizmów integrujących łańcuchy bloków z realizacją konkretnych usług – transakcji. Omówiona w pracy koncepcja formularzy elektronicznych zakłada jak najdalej idącą interoperacyjność opartą na uniwersalnych, standardowych rozwiązaniach i mechanizmach. Jak podkreślono w rozdziale 10 brak standardowych, zunifikowanych mechanizmów komunikacji z łańcuchami bloków spowodował konieczność zaprojektowania sposobów integracji opisanych w podrozdziałach 10.1, 10.2, 10.3.

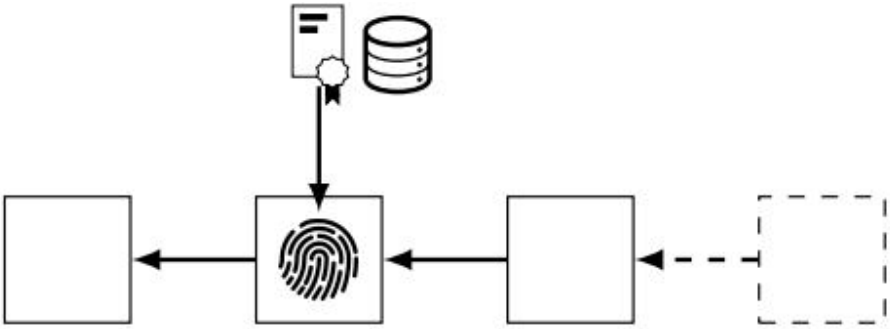
13.2. Formularz elektroniczny jako blok łańcucha

Oprócz konieczności unifikacji procedur interakcji z łańcuchami bloków, co zasygnalizowano w podrozdziale 13.1, kolejnym aspektem praktycznego wykorzystania koncepcji łańcuchów bloków w formularzu elektronicznym jest określenie relacji pomiędzy nimi. W pracy [30], powołując się na [143, 215, 98, 29], zauważono, że dane mogą być przechowywane w łańcuchach bloków w następujący sposób:
— w całości (rys, 13.1a),
— tylko wartość funkcji skrótu (rys. 13.1b), wtedy dane są przechowywane poza łańcuchem (np. prywatny serwer lub rozproszona tablica haseł (DHT) – rysunek 13.1c).

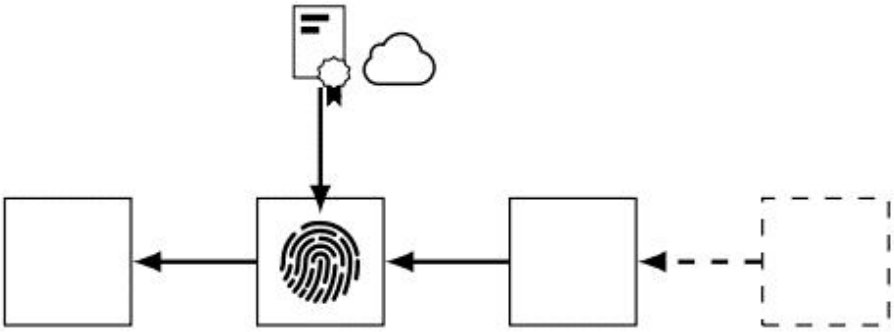
Zgodnie z rysunkiem 13.1 relacje pomiędzy danymi a łańcuchem bloków mogą przyjmować jedną z trzech postaci:



(a) Pełne dane (ikona certyfikatu) – przechowywane w łańcuchu (**On-chain**)



(b) Tylko wartości funkcji skrótów (ikona odcisku palca) – przechowywane w łańcuchu, a właściwe dane są przechowywane na prywatnym serwerze (ikona bazy danych) (**Off-chain**)



(c) Surowe dane – z wykorzystaniem DHT (ikona chmury) (**DHT**)

Rys. 13.1: Trzy sposoby przechowywania danych aplikacji opartych na łańcuchach bloków (źródło: [30])

On-chain – dane istotne dla konkretnego zastosowanie (transakcji) są przechowywane bezpośrednio na łańcuchu bloków, tj. w ramach bloków i/lub transakcji. Pełne przechowywanie w łańcuchu oznacza, że wszystkie dane są przechowywane na blockchainie. Niektóre implementacje przechowują tylko hash, a rzeczywiste dane są przechowywane w innym miejscu.

Off-chain – dane nie są przechowywane w łańcuchu bloków, lecz mogą być przechowywane publicznie lub prywatnie. Publiczne przechowywanie jest wtedy, gdy każdy ma dostęp do danych bez ograniczeń, na przykład poprzez publicznie dostępny serwer internetowy. Natomiast prywatne przechowywanie umożliwia jedynie ograniczony dostęp, a dane są przechowywane pod kontrolą ograniczonej liczby podmiotów.

DHT – jest to szczególny przykład przechowywania danych poza łańcuchem. Przechowywane dane są rozdzielone pomiędzy wielu uczestników. Wartość skrótu może być użyta do uzyskania dostępu do danych. Jeśli używana jest kryptograficznie bezpieczna funkcja skrótu (w DHT) do adresowania przechowywanych danych, to wówczas dane mogą być znakowane czasem, zabezpieczone pod względem integralności i odporności

na manipulacje poprzez zapieczętowanie (podpisanie) wartości skrótów w łańcuchu bloków.

Uogólniając powyższe spostrzeżenia można stwierdzić, że istnieją dwa typy relacji pomiędzy danymi a łańcuchem bloków. Pierwszy to taki, w którym dane są przechowywane w strukturze bloku przeznaczonym na przechowywanie danych lub jako dane transakcji trzymanych w blokach, jeżeli zawartość danych w bloku stanowią transakcje. Drugi, to przechowywanie danych właściwych w zewnętrznych magazynach (np. w klasycznej bazie danych lub magazynie implementowanym w chmurze). Zaproponowana w pracy struktura łańcucha bloków przewiduje, że struktura bloku na wyższym poziomie abstrakcji obejmującym dane w powiązanych blokach, jest zawsze taka sama, a zawartość danych jest dostosowana, dedykowana do konkretnych zadań.

Podobnie jak wyżej, zgodnie z rysunkiem 13.2, formularze elektroniczne mogą:

— przechowywać w swojej strukturze referencje do danych właściwych trzymanych w łańcuchu bloków, a logika formularza powinna definiować dostęp do tych danych,
— stanowić dane, które są częścią składową w blokach łańcucha.

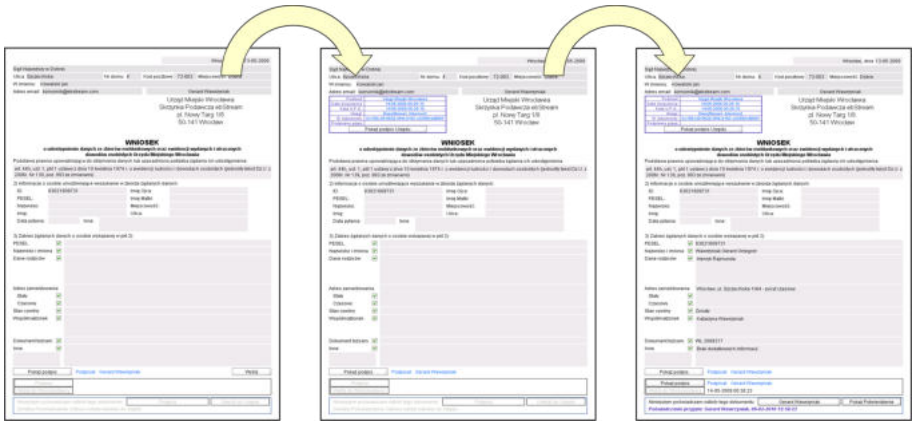
W obu przypadkach każda interakcja pomiędzy formularzem a łańcuchem bloków wymaga zaimplementowania dedykowanego dla łańcucha bloków oprogramowania umożliwiającego tę interakcję.

Innym podejściem do relacji pomiędzy formularzem elektronicznym, danymi a łańcuchem bloków jest budowanie łańcucha bloków, w którym blokiem jest formularz elektroniczny, w którego strukturze istnieją kryptograficzne relacje pomiędzy innymi formularzami (rysunek 13.2). W ten sposób powstałby łańcuch bloków, w którym blokami są dokumenty elektroniczne powstałe w wyniku zrealizowanych transakcji (podrozdział 3.6, rozdział 11) powiązane podpisami, a danymi byłyby wartości zapisane w polach formularza. Takie podejście wymagałoby zbudowania dodatkowego typu podpisu (np. podpis odłączony XML) lub innego mechanizmu kryptograficznego (np. wartości funkcji skrótów) stanowiącego bezpieczną niezmienialną relację pomiędzy formularzami.

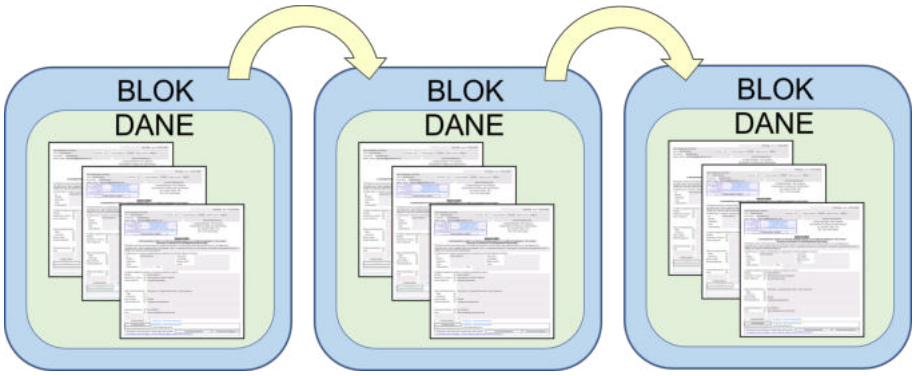
13.3. Formularz elektroniczny a inteligentny kontrakt

Inteligentny kontrakt jest programem komputerowym, który automatycznie wykonuje działania w momencie spełnienia określonych warunków. Logika kodu inteligentnego kontraktu jest trzymana w łańcuchu bloków i jest wykonywana bez udziału człowieka, a przebieg i wynik realizacji transakcji reprezentowanej przez inteligentny kontrakt jest zapisywany w łańcuchu bloków. Przedstawione pojęcie jest zgodne z określeniem zaproponowanym przez Nick'a Szabo w 1994 roku ([186]), a współczesne definicje są często opisywane w kontekście popularnej technologii „Ethereum".

Łatwo zauważyć, że formularz elektroniczny omówiony w podrozdziale 3.6 i rozdziale 11, może być użyty do sterowania przebiegiem realizacji transakcji i jednocześnie może uruchamiać wykonanie określonych działań na podstawie aktualnego stanu danych w formularzu, które reprezentują aktualne stany realizacji transakcji. W stosunku do zaprezentowanego pojęcia inteligentnego kontraktu formularz elektroniczny umożliwia włączenie ludzi i maszyn do realizowanego procesu. Trudno sobie wyobrazić realizację transakcji bez udziału ludzi, a inteligentne kontrakty, realizowane zgodnie z opisanym pojęciem, wymagają zastosowania dedykowanej aplikacji do interakcji z inteligentnym kontraktem. Natomiast formularz elektroniczny nie wymaga tego elementu, gdyż jego trójwarstwowa natura zapewnia gromadzenie i przechowywanie danych, ich prezentację w kontekście zna-



(a) Łańcuch bloków – formularzy



(b) Łańcuch bloków z transakcjami – formularze są transakcjami

Rys. 13.2: Formularz jako blok w łańcuchu (dane w polach formularza) i formularze jako transakcje w danych bloku

czeniowym, a także uruchamianie działań stosownych do stanu i wymagań realizowanych transakcji.

Powyższe rozważania skłaniają do postawienia pytania czy formularz elektroniczny jest inteligentnym kontraktem? Wydaje się, że jest on zgodny z przedstawionym opisem inteligentnego kontraktu, jednak należałoby dokładnie przeanalizować zarówno pojęcie i cechy inteligentnego kontraktu, jak też rozszerzyć cechy formularza elektronicznego i redefiniować jego znaczenie tak, aby był on w pełni zgodny z precyzyjną definicją inteligentnego kontraktu.

13.4. Potencjał formularzy elektronicznych

Jak wielokrotnie wspomniano i wykazano w pracy najważniejszymi cechami formularzy elektronicznych, decydującymi o ich przydatności we wspomaganiu realizacji transakcji, są:

— interoperacyjność, która ma zapewnić łatwe i zrozumiałe interpretowanie treści informacji w zakresie jej tworzenia, przesyłania, przetwarzania i przechowywania w zautomatyzowanych systemach informatycznych, a także przez ludzi,
— bezpieczeństwo rozumiane jako zapewnienie autentyczności, integralności, niezaprze-

czalności pochodzenia, co prowadzi do bezpieczeństwa rozumianego jako skuteczność prawna informacji odzwierciedlającej przebieg i skutki realizowanej transakcji.

Zaprezentowane cechy formularzy elektronicznych pozwalają na wspomaganie realizacji dowolnych transakcji od drobnych, realizowanych pomiędzy prywatnymi osobami, po zaawansowane korporacyjne z udziałem ludzi i integracją z dedykowanymi systemami dziedzinowymi oraz usługami sieciowymi.

Przeanalizowana w pracy idea formularzy elektronicznych, a zwłaszcza koncepcja trzech warstw dokumentu z warstwą logiki i wyrażeń, stwarza możliwość rozszerzania funkcjonalności formularzy poprzez rozbudowę składni struktury formularza oraz odpowiednie implementacje tych rozszerzeń w postaci kolejnych funkcji i „działań" w aplikacji użytkownika. Dzięki temu możliwe jest:

— osiąganie najwyższych poziomów dojrzałości realizacji usług elektronicznych,
— integrowanie formularzy z różnymi platformami komunikacyjnymi,
— integrowanie formularzu z nowo powstającymi technologiami.

Poziom bezpieczeństwa poszczególnych etapów realizacji transakcji, jak i zabezpieczenia środków wymiany komunikatów i wiadomości, zostały podniesione przez eliminację podatności polegającej na możliwości zmanipulowania zawartości podpisów poprzez niezabezpieczone referencje podpisów XML (zastosowanie tzw. *szablonu podpisu* [206, 208]).

Omawiany formularz elektroniczny posiada duży potencjał rozwojowy, co zostanie przedstawione w kolejnych podrozdziałach.

13.4.1. Wysoki poziom dojrzałości i potencjał interoperacyjności

W podrozdziale 3.4 omówiono różne technologie stosowane we wspomaganiu wymiany informacji w realizacji transakcji wraz z cechami tych sposobów z punktu widzenia ich przydatności. Na tej podstawie wykazano, że formularz elektroniczny spełnia wszystkie oczekiwania i z powodzeniem może zastąpić dotychczasowe technologie.

Zastosowanie formularza w realizacji usług elektronicznych pozwala na osiągnięcie najwyższych poziomów – trzeciego lub czwartego (podrozdział 3.3) – czyli *możliwe jest pełne elektroniczne załatwienie sprawy łącznie z podejmowaniem decyzji, informowaniem zainteresowanych stron, potwierdzaniem, dostawą i płatnością.* Dodatkowo należy stwierdzić, że można to zrealizować w dowolnej topologii sieci z możliwością integracji z różnymi systemami i usługami informatycznymi, w sposób sterowany logiką formularza elektronicznego (co wykazano w treści pracy).

Omówione w pracy formularze elektroniczne umożliwiają realizację transakcji co najmniej odzwierciedlając tradycyjne transakcje oparte na dokumentach papierowych. Ponadto dają o wiele więcej możliwości i usprawnień (wartości wyliczane, wartości domyślne, teksty podpowiedzi i pomocy, weryfikacja poprawności danych, integracja z usługami i systemami informatycznymi) z jednoczesnym zachowaniem pełnego bezpieczeństwa.

Reasumując należy zauważyć, że formularze elektroniczne, oprócz dojrzałości usług elektronicznych, zapewniają:

— szeroki zakres funkcjonalności, dzięki któremu spełnione są zróżnicowane wymagania pozwalające na budowanie kompleksowych rozwiązań – usług,
— budowę ogólnie dostępnych usług, zintegrowanych z różnymi platformami,
— stabilność i niezawodność realizowanych usług poprzez prostotę zastosowanych rozwiązań,
— bezpieczeństwo,
— interfejs dostępu do usług elektronicznych poprzez istnienie warstwy prezentacji dynamicznie dostosowującej się do stanu realizacji transakcji,

— łatwą adaptację do zmieniającego się środowiska (potrzeb i oczekiwań), a także powstających nowych technologii.

Przedstawione powyżej właściwości formularzy elektronicznych powinny stanowić podstawę do akceptacji tego typu rozwiązań w budowie nowoczesnych elektronicznych usług publicznych.

13.4.2. Integracja z różnymi technologiami komunikacyjnymi

Realizacji transakcji towarzyszy nie tylko wymiana bezpiecznej informacji dokumentującej etapy transakcji i decyzje stron, ale również konieczność zapewnienia swobodnego komunikowania się stron transakcji. Oprócz możliwości sterowania przebiegiem realizacji transakcji oraz uruchamianiem „działań" odpowiednio do stanu tej transakcji, zawartość logiki formularza można rozszerzyć o parametry niezbędne do skontaktowania się z osobami zainteresowanymi transakcją. Taką możliwość dają usługi *Voice over IP (VoIP)* lub usługi komunikacji multimedialnej (wideokonferencje, czaty głosowe, transmisje strumieniowe). Protokołem komunikacyjnym, który jest stosowany do inicjacji, zarządzania i zakończenia sesji komunikacyjnych w sieci IP, jest *Session Initialisation Protocol (SIP)*. Korzysta on z adresów IP do identyfikacji i lokalizacji uczestników. Definiuje on również format wymienianych w protokole komunikatów – żądań, odpowiedzi i powiadomień. W fazie realizacji funkcje protokołu obejmują:
— inicjowanie i zakończenie połączenia,
— ustalenie parametrów sesji (kodeki audio, video, formaty mediów, parametry transmisji danych itp.),
— przekierowanie i wyszukiwanie użytkowników w sieci IP,
— zarządzanie stanem sesji (dostępny, niedostępny, zajęty).
Dane niezbędne do obsługi połączeń pomiędzy stronami transakcji mogą być umieszczone w warstwie danych i/lub w warstwie logiki formularza, a użycie odpowiedniego przycisku może uruchomić obsługę komunikacji realizowaną przez aplikację użytkownika z wykorzystaniem protokołu SIP z parametrami „dostarczonymi" przez warstwę logiki formularza. Zakres możliwych połączeń/parametrów SIP, dzięki wyrażeniom, odzwierciedlałby stan realizacji transakcji. W ten sposób realizacji transakcji, oprócz wymiany bezpiecznych prawnie danych, towarzyszyć może komunikacja video, audio lub inna z możliwością informowania czy osoba związana z transakcją na określonym etapem jest dostępna.

Słownik pojęć

Def 1. (Anonimowość) *Ukrycie tożsamości podmiotu uczestniczącego w danym procesie.*

Def 2. (Atestacja) *Środki zapewniające aktualność autoryzacji (patrz def. 3) do użycia lub wykorzystania informacji lub zasobów.*
Inaczej „oświadczenie zgodności", wyrażające zapewnienie, że zostały spełnione wyspecyfikowane „wymagania". Takie wymaganie samo w sobie nie daje gwarancji kontraktowych lub innych gwarancji prawnych.
Działania atestacyjne wykonywane przez stronę pierwszą i trzecią są wyróżnione za pomocą terminów „deklaracja", „certyfikacja" (patrz def. 4) i „akredytacja". Dla atestacji wykonywanej przez stronę drugą nie ma specjalnego terminu.

Def 3. (Autoryzacja) *Nadanie formalnego uprawnienia (innemu podmiotowi) do wykonania lub wykonywania jakiegoś zadania lub przyjęcia jakiejś roli.*

Def 4. (Certyfikacja) *Formalna aprobata dla informacji wyrażona przez zaufany podmiot.*

Def 5. (Dokument) *Każdy przedmiot, który może być wykorzystany jako przedmiot prawa potwierdzając określony stosunek prawny, okoliczności lub fakty jakiejś sprawy.*

Def 6. (Dokument binarny) *Patrz def. 8.*

Def 7. (Dokument cyfrowy) *Patrz def. 8.*

Def 8. (Dokument elektroniczny) *Dokument (patrz def. 5) w postaci pliku (patrz def. 18) tekstowego, graficznego, muzycznego, filmowego lub mieszanego będącego wynikiem pracy z danym programem komputerowym, dający się zapisać, a następnie odczytać. Posiada cechy dokumentu, czyli potwierdza prawdziwość zaistnienia danego wydarzenia, okoliczności, zjawiska oraz cechy pliku. Może być więc dowodem w procedurach prawnych.*

Def 9. (Dokument elektroniczny – ustawa) *Dokument elektroniczny – stanowiący odrębną całość znaczeniową zbiór danych uporządkowanych w określonej strukturze wewnętrznej i zapisany na informatycznym nośniku danych (patrz def. 12) (Ustawa z dnia 17 lutego 2005 r. o informatyzacji działalności podmiotów realizujących zadania publiczne).*

Def 10. (Formularz) *Druk urzędowy z miejscami wolnymi, w które wpisuje się wymagane i właściwe dla danego typu pisma informacje. Czyli jest to wzór dokumentu z rubrykami do wypełnienia. Aby formularz mógł być dowodem - dokumentem (patrz def. 5), musi spełniać określone wymagania formalne i merytoryczne.*

Def 11. (Identyfikacja) *Pojęcie tożsame z pojęciem uwierzytelnienie (patrz def. 29).*

Def 12. (Informatyczny nośnik danych) *Materiał lub urządzenie służące do zapisywania, przechowywania i odczytywania danych w postaci cyfrowej (Ustawa z dnia 17 lutego 2005 r. o informatyzacji działalności podmiotów realizujących zadania publiczne).*

Def 13. (Integralność) *Jest to zapewnienie, że informacja nie uległa modyfikacjom za pomocą nieuprawnionych, lub nieautoryzowanych środków.*

Def 14. (Interoperacyjność) *Jest to zdolność osób i usług zaangażowanych w transakcje do wymiany i wykorzystania informacji w celu osiągnięcia określonego celu, obejmuje również zdolność do transgranicznej wymiany danych.*

Def 15. (Kontrola dostępu) *Zapewnienie dostępu do zasobów jedynie uprawnionym użytkownikom.*

Def 16. (Niezaprzeczalność) *Ochrona przed wyparciem się zobowiązań lub działań, które miały miejsce.*

Def 17. (Oznaczenie czasu) *Zarejestrowanie czasu wytworzenia lub istnienia informacji dokonane przez podmiot.*

Def 18. (Plik) *Uporządkowany zbiór danych zapisanych w pamięci trwałej (dysk twardy, nośnik) widziany przez użytkownika systemu operacyjnego jako jednostkowy, niepodzielny, zamknięty całościowo, mający odrębną nazwę, określoną strukturę i atrybuty.*

Def 19. (Plik danych) *Patrz def. 18.*

Def 20. (Plik komputerowy) *Patrz def. 18.*

Def 21. (Podpis) *Środki służące związaniu informacji z podmiotem.*

Def 22. (Pokwitowanie) *Potwierdzenie otrzymania wiadomości.*

Def 23. (Poufność) *Pojęcie tożsame z prywatnością, rozumiane jako ochrona przed ujawnieniem informacji osobom innym niż uprawnione.*

Def 24. (Prywatność) *Patrz def. 23.*

Def 25. (System pisma) *Sposób zapisu języka przy pomocy znaków graficznych. Podstawowe systemy pisma to pismo piktograficzne, ideograficzne, pisma fonetyczne, złożone.*

Def 26. (Świadectwo) *Zweryfikowanie faktu wytworzenia lub istnienia informacji dokonane przez podmiot inny niż twórca.*

Def 27. (Tożsamość) *Jest to zbiór cech podmiotu, który w sposób unikalny i jednoznaczny pozwala wyróżnić ten podmiot z całej populacji podmiotów, do której on należy.*

Def 28. (Transakcja) *jest ustaloną formalnie sekwencją działań wynikającą z ustaleń umownych lub innych relacji wiążących jedną lub większą liczbę stron. Działania te mają na celu osiągnięcie zamierzonych celów (gospodarczych, handlowych, społecznych, prawnych). Transakcje są związane z dowolną działalnością, a przedmiotem transakcji mogą być zarówno dobra materialne, jak i niematerialne.*

Def 29. (Uwierzytelnienie) *Jest to proces rozpoznania tożsamości. Jest to mechanizm kojarzenia przychodzącego żądania z zestawem poświadczeń identyfikujących tożsamość (patrz def. 27). Podane dane uwierzytelniające (dane tożsamości def. 27) są porównywane z danymi znajdującymi się w zasobach (pliku, bazie danych) autoryzowanego użytkownika w lokalnym systemie operacyjnym lub w serwerze uwierzytelniającym.*

Def 30. (Unieważnienie) *Ochrona przed wyparciem się zobowiązań lub działań, które miały miejsce.*

Def 31. (Uwierzytelnienie wiadomości) *Potwierdzenie źródła wiadomości, znane również jako uwierzytelnienie źródła danych.*

Def 32. (Własność) *Środki zapewniające przekazanie podmiotowi prawa do użycia lub przekazania zasobu innym podmiotowi.*

Def 33. (Zatwierdzenie) *Potwierdzenie dostarczenia usług.*

Spis rysunków

2.1 Obszar „Żyznego Półksiężyca” 8
2.2 Tokeny sumeryjskie 9
2.3 Bulla sumeryjska 10
2.4 Pismo sumeryjskie 10
2.5 Ewolucja pisma sumeryjskiego 10
2.6 Ewolucja pisma sumeryjskiego 11
2.7 Papirus z czasów Ptolemeuszów 11
2.8 Alfabet fenicki 12
2.9 Pismo „Kipu” 12
2.10 Rabosz dwudzielny 13
2.11 Wampum traktatowy 13
2.12 Starożytne pieczęcie 16
2.13 Cylindryczna pieczęć z Mezopotamii 16
2.14 Bulla papieska 17
2.15 Indyjski dokument prawny z 1952 roku 17

3.1 Poziomy dojrzałości usług elektronicznych 21
3.2 Dokument vs. dokument elektroniczny 26

5.1 Struktura podpisu XMLdSig 35
5.2 Struktura zawartości elementu `<Reference>` podpisu XMLdSig 36
5.3 Podpisy odłączony, otoczony i otaczający 37
5.4 Schematy przetwarzania podpisu XMLdSig 38

6.1 Stuart Haber i Scott Stornetta - twórcy koncepcji łańcucha bloków . . 45
6.2 Pozycja technologii łańcuchów bloków 47
6.3 Wzrost przychodów z technologii łańcuchów bloków 48
6.4 Topologia systemów 49
6.5 Schemat łańcucha bloków 51
6.6 Łańcuch bloków oraz struktura bloków 51
6.7 Wprowadzanie transakcji do łańcucha bloków 53
6.8 Topologia sieci publicznego łańcucha bloków 54
6.9 Topologia sieci prywatnego łańcucha bloków 54
6.10 Topologia sieci konsorcjalnego łańcucha bloków 55
6.11 Topologie łańcuchów bloków - publiczny, hybrydowy, *permissioned*, prywatny 56
6.12 Zastosowania łańcuchów bloków według ich typów 57
6.13 Taksonomia zastosowań łańcuchów bloków 62
6.14 Obszary zastosowań łańcuchów bloków 63
6.15 System przetwarzania transakcji TPS 64
6.16 Platforma uwierzytelniania danych medycznych 65

6.17 Ścieżka (łańcuch) zaufania 66
6.18 Architektura modelu PKI opartego na logach 67
6.19 Klasyczna infrastreuktura PKI vs. PKI z siecią Web of Trust 68
6.20 „Standardowy” i hybrydowy” certyfikat X.509 70

9.1 Główne elementy struktury formularza 79
9.2 Komponenty graficzne – 1 80
9.3 Komponenty graficzne – 2 81
9.4 Komponenty graficzne – 2 85
9.5 Weryfikacja poprawności danych 86
9.6 Budowanie wiadomości pocztowej 86
9.7 Podpis w formularzu (1) 90
9.8 Podpis w formularzu (2) 90
9.9 Obsługa podpisów w aplikacji użytkownika 92
9.10 Obsługa podpisów w aplikacji użytkownika 92
9.11 Koncepcja struktury wyrażeń 93
9.12 Trzy warstwy dokumentu elektronicznego i ich rola (źródło: [208]) . . . 96
9.13 Plik (XML), składnia(XSD) i aplikacja użytkownika formularza 97
9.14 Funkcja aplikacji – wyświetlenie tekstu pomocy 97
9.15 Prezentacja struktury podpisów w formularzu 98
9.16 Obsługa wskazanego podpisu 99
9.17 Formularz w środowisku – strona transakcji (człowiek), aplikacja użytkownika, usługa sieciowa (poczta) oraz serwery i systemy wewnętrzne 101
9.18 Składniki techniczne wspomagające formularze elektroniczne 102

10.1 Formularz elektroniczny zintegraowany z łańcuchami bloków 103
10.2 Interakcja formularzy elektronicznych z łańcuchami bloków z wykorzystaniem protokołu SOAP 104
10.3 Model logiczny implementacji komunikacji formularza elektronicznego z kryptowalutą *Bitcoin* 105
10.4 Składnia funkcji i „akcji” budujących interakcję z kryptowalutą *Bitcoin* 105
10.5 Uruchomienie kodu wykonywalnego zapisanego w *Załączniku* (`<Attachment>`) 108
10.6 Składnia definicji uruchamiania kodu wykonywalnego zapisanego w formularzu 108
10.7 Schemat weryfikacji ważności certyfikatów z wykorzystaniem różnych systemów BCB-PKI 111
10.8 Certyfikat z rozszerzeniem wskazującym na łańcuch bloków 112

11.1 Sterowanie transakcją: Komornik – przygotowanie, podpisanie i wysłanie wniosku do urzędu 117
11.2 Sterowanie transakcją: Urząd – przygotowanie, podpisanie i wysłanie odpowiedzi do komornika 118
11.3 Sterowanie transakcją: Potwierdzenie otrzymania odpowiedzi na wniosek 119
11.4 Obsługa podpisów w formularzu – wysłanie formularza do ESP 121
11.5 Obsługa podpisów w formularzu – odbiór UPO 122
11.6 Transakcja prosta – wysłanie dokumentu i odbiór potwierdzenia odbioru 126
11.7 Implementacja transakcji z wykorzystaniem poczty elektronicznej . . . 126

11.8 Etap pierwszy – wysłanie zapytania (sąd – urząd) 127
11.9 Etap drugi – uzyskanie odpowiedzi na wniosek (urząd – sąd) 128
11.10 Realizacja transakcji z wykorzystaniem protokołu SOAP 129
11.11 Transakcja z udziałem łańcucha bloków *Bitcoin* 131
11.12 Sekwencja działań w transakcji z płatnością *Bitcoin* 132

12.1 Model transakcji rozproszonej w środowisku heterogenicznym 138
12.2 Lokalizacje, w których zintegrowano zaprojektowane formularze elektroniczne z systemami obiegu dokumentów 139

13.1 Trzy sposoby przechowywania danych aplikacji opartych na łańcuchach bloków 141
13.2 Formularz jako blok w łańcuchu (dane w polach formularza) i formularze jako transakcje w danych bloku 143

A.1–1 Ogólna budowa formularza 172
A.1–2 Struktura strony i elementu grupującego w formularzu oraz komponenty graficzne 173
A.1–3 Własności strony i grupy 173
A.1–4 Wartości domyślne i metadane w formularzu 174
A.2–1 Składnia komponentu *Przycisk* `<button>` 175
A.2–2 Składnia komponentu *Przycisk wyboru* `<checkBox>` 175
A.2–3 Składnia komponentu *Lista rozwijalna* `<dropDownList>` 176
A.2–4 Składnia komponentu *Obszar tekstu* `<textArea>` 176
A.2–5 Składnia komponentu *Tekst* `<text>` 177
A.2–6 Składnia komponentu *Pole danych* `<textField>` 178
A.2–7 Składnia komponentu *Grupa przycisków opcji* `<groupRadioButton>` . . 179
A.2–8 Składnia komponentu *Obraz* `<image>` 180
A.2–9 Składnia komponentu *Załącznik* `<attachment>` 181
A.2–10 Składnia komponentu *Linia* `<line>` 181
A.2–11 Składnia komponentu *Prostokąt* `<rectangle>` 182
A.3–1 Tło komponentu `<background>` 183
A.3–2 Obramowanie komponentu `<border>` 183
A.3–3 Tytuł komponentu `<caption>` 183
A.3–4 Formatowanie daty `<dateFormat>` 183
A.3–5 Format wartości dziesiętnej `<decimalFormat>` 183
A.3–6 Kolor nieaktywnego tekstu `<disabledText>` 183
A.3–7 Edytowalność komponentu `<editable>` 183
A.3–8 Dostępność komponentu `<enabled>` 184
A.3–9 Definiowanie czcionki `<font>` 184
A.3–10 Kolor czcionki komponentu `<foreground>` 184
A.3–11 Tekst pomocy `<helpText>` 184
A.3–12 Ikonka `<icon>` 184
A.3–13 Formatowanie liczby całkowitej `<integerFormat>` 184
A.3–14 Ustawienia międzynarodowe `<international>` 184
A.3–15 Język formularza `<languge>` 184
A.3–16 Kolor tekstu wybranego `<selectetText>` 184
A.3–17 Kolor wskazanego komponentu `<selection>` 185
A.3–18 Sposób budowania zawartości podpisu `<signatureFilter>` 185

A.3–19 Wymagania odnośnie podpisu `<signatureQualification>` 185
A.3–20 Podpisywanie komponentów `<signedBy>` 185
A.3–21 Tekst podpowiedzi `<tipText>` 185
A.3–22 „Widzialność” komponentu `<visible>` 185
A.3–23 Reguły poprawności komponentu `<validity>` 186
A.4–1 Struktura formularza – typy podpisów 187
A.4–2 Podpisy zasobów `<resources>` 187
A.4–3 Podpisy zatwierdzające `<approvedBy>` 188
A.4–4 Podpis formularza `<formSignature>` 188
A.4–5 Podpisy użytkowników `<signature>` 189
A.4–6 Struktura podpisu XML zgodnego z rekomendacją W3C 189
A.5–1 Koncepcja sposobu definiowania składni wyrażenia dowolnego typu (pojęcie abstrakcyjne) `<ExpressionOfType>` 190
A.6–1 Składnia XML dla definiowania wyrażeń typu logicznego `(booleanExpression)` 191
A.6–2 Operatory logiczne 191
A.6–3 Typy porównań 192
A.6–4 Operatory porównywania dla różnych typów porównywanych wartości . 192
A.6–5 Funkcje logiczne obsługi certyfikatów 193
A.6–6 Funkcje logiczne obsługi podpisów 194
A.6–7 Funkcje logiczne obsługi czcionki 194
A.6–8 Inne funkcje logiczne 195
A.7–1 Składnia XML dla definiowania wyrażeń typu tekstowego `<stringExpression>` 196
A.7–2 Funkcje tekstu dla certyfikatów X.509 196
A.7–3 Funkcje obsługi listy certyfikatów unieważnionych 197
A.7–4 Funkcje konwersji do tekstu 197
A.7–5 Dostęp do metadanych Dublin Core 197
A.7–6 Standardowe funkcje obsługi tekstu 198
A.7–7 Standardowe funkcje dostępu do danych o podpisie 198
A.7–8 Inne użyteczne funkcje 199
A.7–9 Globalnie unikalny identyfikator uniwersalny 199
A.7–10 STRING RPC *nieaktualne* 200
A.8–1 Składnia XML dla definiowania wyrażeń typu dziesiętnego `(decimalExpression)` 201
A.8–2 Funkcje dziesiętne i numeryczne 202
A.9–1 Składnia XML dla definiowania wyrażeń typu całkowitego `(integerExpression)` 203
A.9–2 Wyrażenia dla wartości numerycznych (połączenie wartości dziesiętnych i całkowitych z częściowym włączeniem *funkcji całkowitych*) 203
A.9–3 Funkcje całkowite – ogólne 204
A.9–4 Funkcje całkowite – RGB 204
A.9–5 Funkcje całkowite dla daty 204
A.9–6 Inne funkcje całkowite 204
A.10–1 Składnia XML dla definiowania wyrażeń daty `<dateExpression>` . . . 205
A.10–2 Funkcje obsługi daty w certyfikatach i CRL 205
A.10–3 Inne funkcje daty 206
A.11–1 Składnia XML dla definiowania binariów `<mimeExpression>` 207

A.11–2 Kody jedno- i dwuwymiarowe 207
A.11–3 Funkcje binarne dla elementów PKI 208
A.11–4 Obsługa danych biometrycznych w podpisie 208
A.11–5 Konwersje do binariów 208
A.11–6 Konstruowanie listy argumentów w protokole SOAP-RPC (Remote Procedure Call) 209
A.12–1 Składnia XML dla definiowania wyrażeń uruchamiania działań `<actionExpression>` 210
A.12–2 Działanie „puste” 210
A.12–3 Wywołanie komendy Systemu Operacyjnego 210
A.12–4 Wyświetlenie strony o wskazanym numerze 210
A.12–5 Zapisanie formularza na dysku 210
A.12–6 Otwieranie pliku 211
A.12–7 Otwieranie formularza 211
A.12–8 Otwieranie obiektu binarnego 211
A.12–9 Wykonanie podpisu 211
A.12–10 Usuwanie podpisu 211
A.12–11 Wyświetlenie danych o podpisie 211
A.12–12 Wyświetlenie danych o certyfikacie 211
A.12–13 Wyświetlenie deklaracji podpisu 212
A.12–14 Wykonanie „stamp” 212
A.12–15 Aktualizacja wartości domyślnych 212
A.12–16 Walidacja komponentów 212
A.12–17 Wybór *akcji* i uruchomienie działania 212
A.12–18 Generowanie certyfikatu X.509 213
A.12–19 Genrowanie listy certyfikatów unieważnionych *CRL* 214
A.12–20 Składnia działania wiadomości e’mail 215
A.12–21 Konstruowanie wymiany wiadomości w protokole SOAP 216
A.13–1 Składnia XML dla definiowania koloru `(colorExpression)` 217
A.13–2 Kolory – konwersja 217
A.14–1 Składnia XML dla definiowania obramowania `(borderExpression)` . . 218
A.14–2 Obramowania 218
A.15–1 Składnia XML dla definiowania czcionki `<fontExpression>` 219
A.15–2 Wyrażenie służące do wskazania komponentu `<itemExpression>` . . . 220
A.16–1 Wyrażenie wskazujące na informacje o kluczu `(keyInfoExpressison)` . 221
A.16–2 Wyrażenie do obsługi podpisów XMLdSig `(xmlDSigExpression)` . . . 221
A.17–1 Wyrażenie do obsługi fragmentów dokumentów XML `(xmlDocumentExpression)` 222

Spis tabel

3.1 Technologie wymiany informacji a cechy dokumentu elektronicznego 23

6.1 Hierarchia certyfikatów „hybrydowych” w łańcuchu bloków 70
6.2 Porównanie wybranych cech systemów PKI opartych na łańcuchach bloków 70

9.1 Statyczne atrybuty komponentów 81
9.2 Dynamiczne atrybuty komponentów 82
9.3 Typy danych przechowywanych w komponentach 83
9.4 Zestawienie „akcji” formularza ze wskazaniem schemtów struktur 84
9.5 Atrybuty podpisów 87
9.6 Referencje do podpisów w komponentach 89

12.1 Odzwierciedlenie konstrukcji formularza w cechach transakcji 134

Spis wydruków

9.1 Przykład elementu `<Reference>` w strukturze podpisu XMLdSig z elementem XPath wyboru fragmentów do podpisu 91
10.1 Kod źródłowy realizacji transakcji *Bitcoin* 106
10.2 Kod źródłowy otrzymania salda transakcji *Bitcoin* 106
10.3 Kod źródłowy uruchamiania wskazanych metod dynamicznie ładowanej klasy 108
10.4 Program testujący poprawność uruchamiania klasy zakodowanej do postaci *Base64* ze wskazanymi metodami 109
10.5 Klasa testowa 110
11.1 Implementacja obsługi wiadomości w protokole SOAP 123
11.2 Przykład zbudowania i wysyłania wiadomości mail (SMTP) 124

Literatura

[1] URL: www.businessdictionary.com/definition/transaction.html (visited on 03/03/2017).

[2] URL: https://www.microsoft.com (visited on 04/22/2022).

[3] URL: mfiles.pl/pl/index.php/Transaction (visited on 03/03/2017).

[4] Limited 2021 Monax Industries, ed. *Monax*. URL: https://monax.io/ (visited on 01/22/2022).

[5] Binance Academy, ed. *History of Blockchain*. Dec. 6, 2018. URL: https://academy.binance.com/en/articles/history-of-blockchain.

[6] Abu Shohel Ahmed and Tuomas Aura. "Turning Trust Around: Smart Contract-Assisted Public Key Infrastructure". In: *2018 17th IEEE International Conference On Trust, Security And Privacy In Computing And Communications/ 12th IEEE International Conference On Big Data Science And Engineering (TrustCom/BigDataSE)*. IEEE, Aug. 2018. DOI: 10.1109/trustcom/bigdatase.2018.00026.

[7] Muneeb Ali et al. "Blockstack: A global naming and storage system secured by blockchains". In: *2016 USENIX annual technical conference (USENIX ATC 16)*. 2016, pp. 181–194.

[8] Hussein Alnabulsi, Rafiqul Islam, and Majharul Talukder. "GMSA: Gathering Multiple Signatures Approach to Defend Against Code Injection Attacks". In: *IEEE Access* 6 (2018), pp. 77829–77840. DOI: 10.1109/access.2018.2884201.

[9] Abdulaziz Alshammari et al. "Security Threats and Challenges in Cloud Computing". In: *2017 IEEE 4th International Conference on Cyber Security and Cloud Computing (CSCloud)*. IEEE, June 2017. DOI: 10.1109/cscloud.2017.59.

[10] Rainer Alt and Stefan Zbornik. "Integration of Electronic Services in the Execution of Business Transactions". In: *Towards the Knowledge Society*. Springer US, 2003, pp. 717–732. DOI: 10.1007/978-0-387-35617-4_46.

[11] Stanisław Ambroszkiewicz et al. "The SOA Paradigm and e-Service Architecture Reconsidered from the e-Business Perspective". In: *Current Trends in Web Engineering*. Springer Berlin Heidelberg, 2010, pp. 256–265. DOI: 10.1007/978-3-642-16985-4_23.

[12] Mohammad Javad Amiri, Divyakant Agrawal, and Amr El Abbadi. "CAPER: A Cross-Application Permissioned Blockchain". In: *Proc. VLDB Endow.* 12.11 (July 2019), pp. 1385–1398. ISSN: 2150-8097. DOI: 10.14778/3342263.3342275. URL: https://doi.org/10.14778/3342263.3342275.

[13] Hiroaki Anada et al. "Identity-Embedding Method for Decentralized Public-Key Infrastructure". In: *Trusted Systems*. Ed. by Moti Yung, Liehuang Zhu, and Yanjiang Yang. Cham: Springer International Publishing, 2015, pp. 1–14. ISBN: 978-3-319-27998-5.

[14] Claudio A. Ardagna et al. "XML Security". In: *Security, Privacy, and Trust in Modern Data Management*. Springer Berlin Heidelberg, 2007, pp. 71–86. DOI: 10.1007/978-3-540-69861-6_6.

[15] L Axon and M Goldsmith. "PB-PKI: a privacy-aware blockchain-based PKI". In: vol. 6. SCITEPRESS, 2016.

[16] Louise Axon. "Privacy-awareness in blockchain-based PKI". In: *Cdt technical paper series* 21 (2015), p. 15.

[17] A. A. Abd El-Aziz and A. Kannan. "A comprehensive presentation to XML signature and encryption". In: *2013 International Conference on Recent Trends in Information Technology (ICRTIT)*, pp. 643–651. DOI: 10.1109/ICRTIT.2013.6844276.

[18] James Backhouse. "Interoperability of identity and identity management systems". In: *Datenschutz und Datensicherheit-DuD* 30.9 (2006), pp. 568–570.

[19] Mustafa Al-Bassam. "SCPKI: A Smart Contract-based PKI and Identity System". In: *Proceedings of the ACM Workshop on Blockchain, Cryptocurrencies and Contracts*. ACM, Apr. 2017. DOI: 10.1145/3055518.3055530.

[20] Maria Bengtsson and Sören Kock. "" Coopetition" in business Networks—to cooperate and compete simultaneously". In: *Industrial marketing management* 29.5 (2000), pp. 411–426.

[21] Philip Bernstein and Eric Newcomer. *Principles of transaction processing*. Morgan Kaufmann, 2009.

[22] bitcoin.org, ed. *Bitcoin*. URL: https://bitcoin.org/pl/ (visited on 01/22/2022).

[23] Marius Bjerkenes and Moutaz Haddara. "Blockchain Technology Solutions for Supply Chains". In: *Advances in Intelligent Systems and Computing*. Springer International Publishing, Oct. 2019, pp. 909–918. DOI: 10.1007/978-3-030-32520-6_65.

[24] Sharon Boeyen et al. *Internet X.509 Public Key Infrastructure Certificate and Certificate Revocation List (CRL) Profile*. RFC 5280. May 2008. DOI: 10.17487/RFC5280. URL: https://www.rfc-editor.org/info/rfc5280.

[25] Joppe W Bos et al. "Elliptic curve cryptography in practice". In: *International Conference on Financial Cryptography and Data Security*. Springer. 2014, pp. 157–175.

[26] Tim Bray et al. *Extensible Markup Language (XML) 1.0 (Fifth Edition)*. W3C Recommendation. https://www.w3.org/TR/2008/REC-xml-20081126/. W3C, Nov. 2008.

[27] Tim Bray et al. *Namespaces in XML 1.0 (Third Edition)*. W3C Recommendation. https://www.w3.org/TR/2009/REC-xml-names-20091208/. W3C, Dec. 2009.

[28] Suzanne Briet. *Qu'est-ce que la documentation?* Ed. by Paris: Documentaires Industrielles et Techniques. 1951.

[29] Clemens Brunner, Fabian Knirsch, and Dominik Engel. "SPROOF: A Platform for Issuing and Verifying Documents in a Public Blockchain". In: *Proceedings of the 5th International Conference on Information Systems Security and Privacy*. SCITEPRESS - Science and Technology Publications, 2019. DOI: 10.5220/0007245600150025.

[30] Clemens Brunner et al. "A Comparison of Blockchain-based PKI Implementations". In: *Proceedings of the 6th International Conference on Information Systems Security and Privacy*. SCITEPRESS - Science and Technology Publications, 2020. DOI: 10.5220/0008914503330340.

[31] Francesco Buccafurriand, Lidia Fotia, and Gianluca Lax. *Implementing Advanced Electronic Signature by Public Digital Identity System (SPID)*. EGOVIS 2016, LNCS 9831. Springer International Publishing Switzerland 2016, 2016.

[32] Joseph M. Bugajski, Robert L. Grossman, and Steve Vejcik. "A Service Oriented Architecture Supporting Data Interoperability for Payments Card Processing Sys-

tems". In: *Service-Oriented Computing – ICSOC 2007*. Springer Berlin Heidelberg, 2006, pp. 591–600. DOI: 10.1007/11948148_57.

[33] Laurence Bull and David M. Squire. "XML Signature Extensibility Using Custom Transforms". In: *Web Information Systems – WISE 2004*. Springer Berlin Heidelberg, 2004, pp. 102–112. DOI: 10.1007/978-3-540-30480-7_12.

[34] Benedikt Bunz et al. "FlyClient: Super-Light Clients for Cryptocurrencies". In: *2020 IEEE Symposium on Security and Privacy (SP)*. IEEE, May 2020. DOI: 10.1109/sp40000.2020.00049.

[35] Christian Cachin et al. "Architecture of the hyperledger blockchain fabric". In: *Workshop on distributed cryptocurrencies and consensus ledgers*. Vol. 310. 4. Chicago, IL. 2016, pp. 1–4.

[36] CapgeminiIDC et al. *Digitizing Public Services in Europe:Putting ambition into action, 9th Benchmark Measurement*. European Commission, Directorate General for Information Society and Media, 2010. URL: https://www.capgemini.com/wp-content/uploads/2017/07/Digitizing_Public_Services_in_Europe__Putting_Ambition_into_Action.pdf.

[37] G. Caronni. "Walking the Web of trust". In: *Proceedings IEEE 9th International Workshops on Enabling Technologies: Infrastructure for Collaborative Enterprises (WET ICE 2000)*. IEEE Comput. Soc. DOI: 10.1109/enabl.2000.883720.

[38] Miguel Castro, Barbara Liskov, et al. "Practical byzantine fault tolerance". In: *OsDI*. Vol. 99. 1999. 1999, pp. 173–186.

[39] Shuchih E. Chang and Yichian Chen. "When Blockchain Meets Supply Chain: A Systematic Literature Review on Current Development and Potential Applications". In: *IEEE Access* 8 (2020), pp. 62478–62494. DOI: 10.1109/access.2020.2983601. URL: https://ieeexplore.ieee.org/abstract/document/9047881 (visited on 01/23/2022).

[40] Jing Chen et al. "CertChain: Public and Efficient Certificate Audit Based on Blockchain for TLS Connections." In: *INFOCOM*. IEEE, 2018, pp. 2060–2068. ISBN: 978-1-5386-4128-6. URL: http://dblp.uni-trier.de/db/conf/infocom/infocom2018.html#ChenYY0JD18.

[41] Ying Chen, Wenping Guo, and Xiaoming Zhao. "Study of XML digital signature for resource document fragment". In: *The 2nd International Conference on Information Science and Engineering*. IEEE, Dec. 2010. DOI: 10.1109/icise.2010.5690206.

[42] Christian Chiarcos. "Interoperability of Corpora and Annotations". In: *Linked Data in Linguistics: Representing and Connecting Language Data and Language Metadata*. Ed. by Christian Chiarcos, Sebastian Nordhoff, and Sebastian Hellmann. Berlin, Heidelberg: Springer Berlin Heidelberg, 2012, pp. 161–179. ISBN: 978-3-642-28249-2. DOI: 10.1007/978-3-642-28249-2_16. URL: https://doi.org/10.1007/978-3-642-28249-2_16.

[43] Corda, ed. *Corda*. URL: https://www.corda.net/ (visited on 01/22/2022).

[44] Transaction Processing Performance Council. "TPC benchmark C (standard specification, revision 5.11), 2010". In: *URL: http://www. tpc. org/tpcc* (2010), p. 100.

[45] *Council Directive 91/250/EC, 14.5.1991 on the legal protection of computer programmes. Official Journal of the European Communities. No L 122, 17.05.91.* 1991.

[46] Counterparty.io, ed. *Counterparty extends Bitcoinin new and powerful ways*. URL: https://counterparty.io/ (visited on 01/22/2022).

[47] Juan Carlos Cruellas et al. *XML Advanced Electronic Signatures (XAdES)*. Tech. rep. W3C, Feb. 20, 2003.

[48] Fangfang Dai et al. "From Bitcoin to cybersecurity: A comparative study of blockchain application and security issues". In: *2017 4th International Conference on Systems and Informatics (ICSAI)*. IEEE, Nov. 2017. DOI: 10.1109/icsai.2017.8248427.

[49] Wei Dai. "B-Money, A Scheme for a Group of Untraceable Digital Pseudonyms to Pay Each Other With Money and to Enforce Contracts Amongst Themselves Without Outside Help". In: *Weidai. com* (1998).

[50] Wei Dai. *b-money, a scheme for a group of untraceable digital pseudonyms to pay each other with money and to enforce contracts amongst themselves without outside help*. Ed. by Wej Dai. 2006.

[51] Portal Bazy Danych, ed. *7.2. Transakcje*. May 7, 2022. URL: https://mst.mimuw.edu.pl/lecture.php?lecture=bad&part=Ch7.

[52] Christian Decker and Roger Wattenhofer. "Information propagation in the Bitcoin network". In: *IEEE P2P 2013 Proceedings*. IEEE, Sept. 2013. DOI: 10.1109/p2p.2013.6688704.

[53] Pramod Dhakal. "The law of rule: centralized, decentralized and distributed systems". In: *Report for CFFN, NRN-Canada, NRNA* (2009).

[54] *Document management - Portable document format - Part1: PDF 1.7. - PDF 32000-1:2008*. Tech. rep. Adobe Systems Incorporated, 2008.

[55] *Document management - Portable document format - Part1: PDF 1.7. - PDF 32000-1:2008*. Tech. rep. International Organization for Standardization, 2008.

[56] *Electronic Signatures an Infractructures (ESI); CMS Advanced Electronic Signatures (CAdES)*. Tech. rep. [31]ETSI TS 101 733 v2.2.1 (2013-04). ETSI.

[57] ethereum.org, ed. *Ethereum*. Ethererum.Org. URL: https://ethereum.org/en/ (visited on 01/22/2022).

[58] *ETSI TS 102 778-1 (778-2, 778-3, 778-4, 778-5), Electronic Signatures and Infrastructures (ESI). PDF Advanced Electronic Signature Profiles (5 parts)*. Tech. rep. ETSI.

[59] Commission of the European Communities. *The Role of eGovernment for Europe's Future, Brussels*. Communication from the Commission to the Council, the European Parliament, the European Economic and Social Committee and the Committee of the Regions. Chap. COM(2003) 567 final.

[60] *European Public Administration Network, eGovernment Working Group: Key Principles of an Interoperability Architecture. Brussels. Available at http://www. epractice.eu/document/2963. Accessed 7 Jun 2019.*

[61] Tooba Faisal, Nicolas Courtois, and Antoaneta Serguieva. "The Evolution of Embedding Metadata in Blockchain Transactions". In: *2018 International Joint Conference on Neural Networks (IJCNN)*. IEEE, July 2018. DOI: 10.1109/ijcnn.2018.8489377. URL: https://ieeexplore.ieee.org/abstract/document/8489377.

[62] Hal Finney. *Rüttelplatten Baublog – Rütteln, Verdichten, Befestigen, Rüttelplatten Baublog*. Aug. 15, 2004.

[63] Energy Web Foundation, ed. *Energy Web*. URL: http://energyweb.org (visited on 01/22/2022).

[64] Hyperledger Foundation, ed. *Hyperledger Fabric*. URL: https://www.hyperledger.org/use/fabric (visited on 01/22/2022).

[65] Conner Fromknecht, Dragos Velicanu, and Sophia Yakoubov. "Certcoin: A namecoin based decentralized authentication system". In: *Massachusetts Inst. Technol., Cambridge, MA, USA, Tech. Rep* 6 (2014), pp. 46–56.

[66] Julian Gantner, Andreas Geyer-Schulz, and Anke Thede. "A single sign-on protocol for distributed web applications based on standard internet mechanisms". In: *e-Business and Telecommunication Networks*. Springer, 2006, pp. 176–183. URL: https://link.springer.com/content/pdf/10.1007/1-4020-4761-4_14.pdf.

[67] Gartner, ed. *5 Trends Emerge in the Gartner Hype Cycle for Emerging Technologies, 2018*. Nov. 5, 2018. URL: https://www.gartner.com/smarterwithgartner/5-trends-emerge-in-gartner-hype-cycle-for-emerging-technologies-2018 (visited on 01/22/2022).

[68] Jaideep Gera et al. "Blockchain Technology for Fraudulent Practices in Insurance Claim Process". In: *2020 5th International Conference on Communication and Electronics Systems (ICCES)*. IEEE, June 2020, pp. 1068–1075. DOI: 10.1109/icces48766.2020.9138012.

[69] IGI Global, ed. *What is Centralized System*. URL: http://www.igi-global.com/dictionary/centralised-system/3590 (visited on 04/22/2022).

[70] Soner Gökten and Burak Özdoğan. "The Doors Are Opening for the New Pedigree: A Futuristic View for the Effects of Blockchain Technology on Accounting Applications". In: *Contributions to Management Science*. Springer International Publishing, Nov. 2019, pp. 425–438. DOI: 10.1007/978-3-030-29739-8_20. URL: https://link.springer.com/chapter/10.1007/978-3-030-29739-8_20 (visited on 01/23/2022).

[71] Peter Gonczol et al. "Blockchain Implementations and Use Cases for Supply Chains -A Survey". In: *IEEE Access* 8 (2020), pp. 11856–11871. DOI: 10.1109/access.2020.2964880. URL: https://ieeexplore.ieee.org/abstract/document/8952728 (visited on 01/23/2022).

[72] Portal Szybka Gotówka, ed. *Transakcja*. URL: https://szybkagotowka.pl/Wiki/transakcja (visited on 05/07/2022).

[73] Portal gov.pl. *Załatw sprawy urzędowe przez internet na ePUAP*. URL: https://www.gov.pl/web/gov/zalatwiaj-sprawy-urzedowe-przez-internet-na-epuap (visited on 07/07/2022).

[74] Jim Gray. "The Transaction Concept: Virtues and Limitations". In: *Readings in Database Systems*. San Francisco, CA, USA: Morgan Kaufmann Publishers Inc., 1988, pp. 140–150. ISBN: 0934613656.

[75] Jim Gray and Leslie Lamport. "Consensus on transaction commit". In: *ACM Transactions on Database Systems (TODS)* 31.1 (2006), pp. 133–160.

[76] Gideon Greenspan. *MultiChain Private Blockchain — White Paper*. Coin Sciences Ltd, June 12, 2015. URL: https://www.multichain.com/download/MultiChain-White-Paper.pdf.

[77] Q. Gu and P. Lago. "Exploring service-oriented system engineering challenges: a systematic literature review". English. In: *Service Oriented Computing and Applications* (2009). ISSN: 1863-2386. DOI: 10.1007/s11761-009-0046-7. URL: https://research.vu.nl/en/publications/exploring-service-oriented-system-engineering-challenges-a-system (visited on 04/22/2022).

[78] Qing Gu and Patricia Lago. "Exploring service-oriented system engineering challenges: a systematic literature review". In: *Service Oriented Computing and Applications* 3.3 (2009), pp. 171–188.

[79] Abhinav Nath Gupta and P. Santhi Thilagam. "Detection of XML Signature Wrapping Attack Using Node Counting". In: *Proceedings of the 3rd International Symposium on Big Data and Cloud Computing Challenges (ISBCC – 16')*. Springer International Publishing, 2016, pp. 57–63. DOI: 10.1007/978-3-319-30348-2_5.

[80] Suyash Gupta and Mohammad Sadoghi. "Blockchain Transaction Processing". In: *Encyclopedia of Big Data Technologies*. Ed. by Sherif Sakr and Albert Zomaya. Cham: Springer International Publishing, 2018, pp. 1–11. ISBN: 978-3-319-63962-8. DOI: 10.1007/978-3-319-63962-8_333-1. URL: https://doi.org/10.1007/978-3-319-63962-8_333-1.

[81] Stuart Haber and W Scott Stornetta. "How to time-stamp a digital document". In: *Conference on the Theory and Application of Cryptography*. Springer. 1990, pp. 437–455. URL: https://link.springer.com/content/pdf/10.1007/3-540-38424-3_32.pdf.

[82] Mohamed Tahar Hammi, Patrick Bellot, and Ahmed Serhrouchni. "BCTrust: A decentralized authentication blockchain-based mechanism". In: *2018 IEEE Wireless Communications and Networking Conference (WCNC)*. IEEE, Apr. 2018. DOI: 10.1109/wcnc.2018.8376948.

[83] Enze He and Qiaoyan Wen. "A Single Sign-On Scheme for Cross Domain Web Applications Based on SOA". In: *Internet of Things*. Springer Berlin Heidelberg, 2012, pp. 581–589. DOI: 10.1007/978-3-642-32427-7_83.

[84] Shidokht Hejazi-Sepehr, Ross Kitsis, and Ali Sharif. "Transwarp Conduit: Interoperable Blockchain Application Framework". In: *arXiv preprint arXiv:1906.03256* (2019). URL: https://arxiv.org/abs/1906.03256 (visited on 01/23/2022).

[85] Maurice Herlihy and Mark Moir. *Enhancing Accountability and Trust in Distributed Ledgers*. 2016. DOI: 10.48550/ARXIV.1606.07490.

[86] Hill. *Taxonomy of Attacks against XML Digital Signature and Encryption Command Injection in XML Signatures and Encryption*. Ed. by iSEC Partners 2007. Dec. 19, 2019.

[87] Russ Housley. *Cryptographic Message Syntax (CMS)*. RFC 5652. Sept. 2009. DOI: 10.17487/RFC5652. URL: https://www.rfc-editor.org/info/rfc5652.

[88] *IEEE Standard Computer Dictionary: A Compilation of IEEE Standard Computer Glossaries, IEEE Standard 610-1990, New York, NY, USA, 1990*. 1990.

[89] Investopedia, ed. *Transaction*. May 7, 2022. URL: https://www.investopedia.com/terms/t/transaction.asp.

[90] *ISO/IEC 2382–1:1993 Information Technology – Vocabulary – Part 1: Fundamental Terms, International Organization for Standardization*. 1993.

[91] Stefan Jablonski et al. *Guide to Web Application and Platform Architectures*. Springer Berlin Heidelberg, 2004. DOI: 10.1007/978-3-662-07631-6. URL: https://link.springer.com/content/pdf/bfm:978-3-662-07631-6/1.pdf.

[92] Markus Jakobsson and Ari Juels. "Proofs of work and bread pudding protocols". In: *Secure information networks*. Springer, 1999, pp. 258–272.

[93] Praveen Jayachandran. "The difference between public and private blockchain". In: *Blockchain Unleashed: IBM Blockchain Blog* 2017 (2017).

[94] Meiko Jensen et al. "On the effectiveness of XML Schema validation for countering XML Signature Wrapping attacks". In: *2011 1st International Workshop on Securing Services on the Cloud (IWSSC)*. IEEE, Sept. 2011. DOI: 10.1109/iwsscloud.2011.6049019.

[95] Jongil Jeong et al. "Integration of Single Sign-On and Role-Based Access Control Profiles for Grid Computing". In: *Frontiers of WWW Research and Development - APWeb 2006*. Springer Berlin Heidelberg, 2006, pp. 880–885. DOI: 10.1007/11610113_89.

[96] Wenbo Jiang et al. "PTAS: Privacy-preserving Thin-client Authentication Scheme in blockchain-based PKI". In: *Future Generation Computer Systems* 96 (July 2019), pp. 185–195. DOI: 10.1016/j.future.2019.01.026.

[97] Burt Kaliski. *PKCS #7: Cryptographic Message Syntax Version 1.5*. RFC 2315. Mar. 1998. DOI: 10.17487/RFC2315. URL: https://www.rfc-editor.org/info/rfc2315.

[98] Enis Karaarslan and Eylul Adiguzel. "Blockchain Based DNS and PKI Solutions". In: *IEEE Communications Standards Magazine* 2.3 (Sept. 2018). DOI: 10.1109/mcomstd.2018.1800023.

[99] Gregor Karlinger. "XML Electronic Signatures Application according to the international standard XML Signature Syntax and Processing". In: (). DOI: 10.1007/978-0-387-35413-2_36.

[100] Anish Karmarkar et al. *SOAP Version 1.2 Specification Assertions and Test Collection (Second Edition)*. W3C Recommendation. https://www.w3.org/TR/2007/REC-soap12 -testcollection-20070427/. W3C, Apr. 2007.

[101] Stephan Kassel et al. "An interoperability architecture for networked service delivery". In: *IFIP-International Workshop on Enterprise Interoperability*. Springer. 2009, pp. 125–132.

[102] Jonathan Katz and Yehuda Lindell. *Introduction to modern cryptography*. CRC press, 2020. URL: https://books.google.pl/books?hl=pl&lr=&id-%20=RsoOEAAAQBAJ&oi=fnd&pg=PT12&dq=Katz%20J,%20Lindell%20Y%20(2007)%20Introduction%20to%20modern%20cryptography.%20Chapman%20&%20Hall/CRC,%20Boca%20Raton&ots=vJak22zFMs&sig=Z4mp2yFWS0ab3nFeZEQcDqosHfA&redir_esc=y#v=onepage&q&f=false.

[103] Sunny King and Scott Nadal. "Ppcoin: Peer-to-peer crypto-currency with proof-of-stake". In: *self-published paper, August* 19.1 (2012).

[104] Dr. John C. Klensin. *Simple Mail Transfer Protocol*. RFC 2821. Apr. 1, 2001. DOI: 10.17487/RFC2821. URL: https://www.rfc-editor.org/info/rfc2821.

[105] Tomas Knap and Irena Mlynkova. "Towards more secure web services - Exploiting and analysing XML signature security issues". In: *2009 Third International Conference on Research Challenges in Information Science*. IEEE, Apr. 2009. DOI: 10.1109/rcis.2009.5089267.

[106] Stefan Konst. *Sichere Log-Dateien auf Grundlage kryptographisch verketteter Einträge*. de. Universitätsbibliothek Braunschweig, 2000. DOI: 10.24355/DBBS.084-201707191306.

[107] Hadi Razzaghi Kouchaksaraei and Alexander G. Chefranov. "Countering Wrapping Attack on XML Signature in SOAP Message for Cloud Computing". In: (2013). DOI: 10.48550/ARXIV.1310.0441.

[108] Dr. Hugo Krawczyk, Mihir Bellare, and Ran Canetti. *HMAC: Keyed-Hashing for Message Authentication*. RFC 2104. Feb. 1997. DOI: 10.17487/RFC2104. URL: https://www.rfc-editor.org/info/rfc2104.

[109] Wolfgang Kubbilun et al. "Trustworthy Verification and Visualisation of Multiple XML-Signatures". In: *Communications and Multimedia Security*. Springer Berlin Heidelberg, 2005, pp. 311–320. DOI: 10.1007/11552055_41.

[110] Murat Yasin Kubilay, Mehmet Sabir Kiraz, and Hacı Ali Mantar. "CertLedger: A new PKI model with Certificate Transparency based on blockchain". In: *Computers & Security* 85 (2019), pp. 333–352.

[111] Jitendra Kumar et al. "XML wrapping attack mitigation using positional token". In: *2017 International Conference on Public Key Infrastructure and its Applications (PKIA)*. IEEE, Nov. 2017. DOI: 10.1109/pkia.2017.8278958.

[112] Olga Labazova, Tobias Dehling, and Ali Sunyaev. "From hype to reality: A taxonomy of blockchain applications". In: *Proceedings of the 52nd Hawaii International Conference on System Sciences (HICSS 2019)*. 2019. URL: https://papers.ssrn.com/sol3/papers.cfm?abstract_id=3250648.

[113] RSK Labs, ed. *RSK*. URL: https://www.rsk.co/ (visited on 01/23/2022).

[114] L Lamport. *The Part-Time Parliament. ACM Transactions on Computer Systems 16 (2) 133 169*. 1998.

[115] Dominic Letz. "Blockquick: Super-light client protocol for blockchain validation on constrained devices". In: *Cryptology ePrint Archive* (2019).

[116] Shuwei Li and Fucheng You. "Research on XML Digital Signature Application in the Log Protection of Typesetting on Web". In: *2012 International Conference on Control Engineering and Communication Technology*. IEEE, Dec. 2012. DOI: 10.1109/iccect.2012.15.

[117] Xueping Liang et al. "ProvChain: A Blockchain-Based Data Provenance Architecture in Cloud Environment with Enhanced Privacy and Availability". In: *2017 17th IEEE/ACM International Symposium on Cluster, Cloud and Grid Computing (CCGRID)*. IEEE, May 2017. DOI: 10.1109/ccgrid.2017.8.

[118] *Libre Office The Document Foundation*. URL: https://pl.libreoffice.org/ (visited on 05/24/2022).

[119] Iuon-Chang Lin and Tzu-Chun Liao. "A survey of blockchain security issues and challenges." In: *International Journal of Network Security* 19.5 (2017), pp. 653–659. DOI: 10.6633/IJNS.201709.19(5).01. URL: https://scholar.google.pl/scholar?q=A%20survey%20of%20blockchain%20security%20issues%20and%20challenges.%20International%20Journal%20of%20Network%20Security.&hl=pl&as_sdt=0&as_vis=1&oi=scholart.

[120] Litecoin, ed. *Litecoin*. URL: https://litecoin.com (visited on 01/22/2022).

[121] Kevin Liu and David Booth. *Web Services Description Language (WSDL) Version 2.0 Part 0: Primer*. W3C Recommendation. https://www.w3.org/TR/2007/REC-wsdl20-primer-20070626/. W3C, June 2007.

[122] Miao Liu and Ya Li. "An Implementation of a XML Documents Secure Circulation System". In: *Journal of Physics: Conference Series*. Vol. 1060. 1. IOP Publishing. 2018, p. 012004.

[123] Axon LM and Goldsmith Michael. "PB-PKI: A privacy-aware blockchain-based PKI". In: (2016).

[124] Xiaoying Ma et al. "Analysis of Blockchain Technology and its Application in the Field of Radio Monitoring". In: *2021 International Conference on Computer, Blockchain and Financial Development (CBFD)*. IEEE, Apr. 2021. DOI: 10.1109/cbfd52659.2021.00097. URL: https://ieeexplore.ieee.org/document/9759140.

[125] Zakaria Maamar and Bernard Moulin. "Interoperability of distributed and heterogeneous systems based on software agent-oriented frameworks". In: *International Workshop on Cooperative Information Agents*. Springer. 1997, pp. 248–259.

[126] Christian Mainka et al. "Making XML Signatures Immune to XML Signature Wrapping Attacks". In: *Communications in Computer and Information Science*. Springer International Publishing, 2013, pp. 151–167. DOI: 10.1007/978-3-319-04519-1_10.

[127] Murray Maloney et al. *W3C XML Schema Definition Language (XSD) 1.1 Part 1: Structures*. W3C Recommendation. W3C, Apr. 2012.

[128] Stephanos Matsumoto and Raphael M Reischuk. "IKP: Turning a PKI around with decentralized automated incentives". In: *2017 IEEE Symposium on Security and Privacy (SP)*. IEEE. 2017, pp. 410–426.

[129] Stephanos Matsumoto and Raphael M. Reischuk. *IKP: Turning a PKI Around with Blockchains*. Cryptology ePrint Archive, Paper 2016/1018. https://eprint.iacr.org/2016/1018. 2016. URL: https://eprint.iacr.org/2016/1018.

[130] Stephanos Matsumoto, Pawel Szalachowski, and Adrian Perrig. "Deployment challenges in log-based PKI enhancements". In: *Proceedings of the Eighth European Workshop on System Security*. ACM, Apr. 2015. DOI: 10.1145/2751323.2751324.

[131] Juri Mattila. *The blockchain phenomenon–the disruptive potential of distributed consensus architectures*. Tech. rep. ETLA working papers, 2016.

[132] Michael McIntosh and Paula Austel. "XML signature elent wrapping attacks and countermeasures". In: *Proceedings of the 2005 workshop on Secure web services - SWS '05*. ACM Press, 2005. DOI: 10.1145/1103022.1103026.

[133] Robert Meersman, Zahir Tari, and Angelo Corsaro. *On the Move to Meaningful Internet Systems 2004: OTM 2004 Workshops: OTM Confederated International Workshops and Posters, GADA, JTRES, MIOS, WORM, WOSE, PhDS, and INTEROP 2004, Agia Napa, Cyprus, October 25-29, 2004. Proceedings*. Vol. 3292. Springer Science & Business Media, 2004.

[134] Noah Mendelsohn et al. *SOAP Version 1.2 Part 1: Messaging Framework (Second Edition)*. W3C Recommendation. https://www.w3.org/TR/2007/REC-soap12 - part1-20070427/. W3C, Apr. 2007.

[135] Alfred J. Menezes, Paul C. Van Oorschot, and Scott A. Vanstone. *Kryptografia stosowana*. Wydawnictwa Naukowo-Techniczne, 2005. ISBN: 8320429927.

[136] Merriam-Webster, ed. *Transactiob*. May 7, 2022. URL: https://www.merriam-webster.com/dictionary/transaction.

[137] Charles Miller. *Stuart Haber and Scott Stornetta: How our timestamping mechanism was used in Bitcoin*. Ed. by https://coingeek.com/. Nov. 28, 2021. URL: https://coingeek.com/stuart-haber-and-scott-stornetta-how-our-timestamping-mechanism-was-used-in-bitcoin-video/ (visited on 01/21/2022).

[138] Mahdi H. Miraz and Maaruf Ali. "Applications of Blockchain Technology beyond Cryptocurrency". In: *Annals of Emerging Technologies in Computing* 2.1 (Jan. 2018), pp. 1–6. DOI: 10.33166/aetic.2018.01.001. URL: https://arxiv.org/abs/1801.03528 (visited on 01/23/2022).

[139] Nilo Mitra and Yves Lafon. *SOAP Version 1.2 Part 0: Primer (Second Edition)*. Tech. rep. https://www.w3.org/TR/2007/REC-soap12-part0-20070427/. W3C, Apr. 2007.

[140] Monero, ed. *Monero*. URL: https://www.getmonero.org/ (visited on 01/22/2022).

[141] Jean-Jacques Moreau et al. *SOAP Version 1.2 Part 2: Adjuncts (Second Edition)*. W3C Recommendation. W3C, Apr. 2007.

[142] Gayathri Nagasubramanian et al. "Securing e-health records using keyless signature infrastructure blockchain technology in the cloud". In: *Neural Computing and Applications* 32.3 (Nov. 2018), pp. 639–647. DOI: 10.1007/s00521-018-3915-1.

[143] Satoshi Nakamoto. "Bitcoin: A peer-to-peer electronic cash system". In: *Decentralized Business Review* (2008), p. 21260. URL: https://scholar.google.com/scholar?hl=pl&as_sdt=0,5&q=satoshi%20nakamoto&btnG=&oq=satoshi.

[144] Sasa Nesic. "Semantic Document Model to Enhance Data and Knowledge Interoperability". In: *Web 2.0 and Semantic Web*. Springer US, 2009, pp. 135–160. DOI: 10.1007/978-1-4419-1219-0_6.

[145] Kim Nguyen. "Certification of eIDAS trust services and new global transparency trends". In: *Datenschutz und Datensicherheit-DuD* 42.7 (2018), pp. 424–428.

[146] M. Niranjanamurthy, B. N. Nithya, and S. Jagannatha. "Analysis of Blockchain technology: pros, cons and SWOT". In: *Cluster Computing* 22.S6 (Mar. 2018), pp. 14743–14757. DOI: 10.1007/s10586-018-2387-5. URL: https://link.springer.com/article/10.1007/s10586-018-2387-5.

[147] Daniel E O'Leary. "Configuring blockchain architectures for transaction information in blockchain consortiums: The case of accounting and supply chain systems". In: *Intelligent Systems in Accounting, Finance and Management* 24.4 (2017), pp. 138–147. URL: https://onlinelibrary.wiley.com/doi/full/10.1002/isaf.1417.

[148] Amrutanshu Panigrahi, Ajit Kumar Nayak, and Rourab Paul. *Smart Contract Assisted Blockchain based PKI System*. 2022. DOI: 10.48550/ARXIV.2207.09127.

[149] Massimo Paolucci and Berrand Souville. "Data interoperability in the future of middleware". In: *Journal of Internet Services and Applications* 3.1 (2012), pp. 127–131.

[150] Mike P. Papazoglou and Willem-Jan van den Heuvel. "Service oriented architectures: approaches, technologies and research issues". In: *The VLDB Journal* 16.3 (Mar. 2007), pp. 389–415. DOI: 10.1007/s00778-007-0044-3.

[151] Parity, ed. *Parity Technologies (2018) Parity ethereum blockchain*. URL: https://www.parity.io/ (visited on 11/18/2022).

[152] Victor Pastena. "Some evidence on the SEC's system of continuous disclosure". In: *Accounting Review* (1979), pp. 776–783. URL: https://www.jstor.org/stable/245632 (visited on 11/14/2022).

[153] Sumathi Pawar and Niranjan N Chiplunkar. "Open source apis for processing the XML result of web services". In: *2017 International Conference on Advances in Computing, Communications and Informatics (ICACCI)*. IEEE, Sept. 2017. DOI: 10.1109/icacci.2017.8126114.

[154] M. E. Peck. "The cryptoanarchists' answer to cash". In: *IEEE Spectrum* 49.6 (June 2012), pp. 50–56. DOI: 10.1109/mspec.2012.6203968.

[155] R. Perlman. "An overview of PKI trust models". In: *IEEE Network* 13.6 (1999), pp. 38–43. DOI: 10.1109/65.806987.

[156] Radoslaw Petermann. *Electronic Document*. Ed. by ArchNet. Łódź, 1988.

[157] Wim Peters. "Establishing interoperability between linguistic and terminological ontologies". In: *New Trends of Research in Ontologies and Lexical Resources*. Springer, 2013, pp. 27–42.

[158] Massimo Di Pierro. "What Is the Blockchain?" In: *Computing in Science & Engineering* 19.5 (2017), pp. 92–95. DOI: 10.1109/mcse.2017.3421554.

[159] Marc Pilkington. *Blockchain technology: principles and applications*. DOI: 10.4337/9781784717766.00019. URL: https://www.elgaronline.com/view/edcoll/9781784717759/9781784717759.00019.xml.

[160] J Ronald Prins and Business Unit Cybercrime. "DigiNotar certificate authority breach "operation black tulip"". In: *Fox-IT, November* (2011), p. 18.

[161] Joannah Ravi and Balamurugan Balusamy. "Provision of XML Security in E-Commerce Applications with XML Digital Signatures Using Virtual Smart Card". In: *Proceedings of First International Conference on Information and Communication Technology for Intelligent Systems: Volume 2*. Springer International Publishing, 2016, pp. 411–421. DOI: 10.1007/978-3-319-30927-9_40.

[162] Abdul Razaque, Ankit Trivedi, and Mansi Joshi. "Novel Signature Verification Process". In: *2016 IEEE Long Island Systems, Applications and Technology Conference (LISAT)*. Farmingdale, NY, USA, Apr. 20, 2016.

[163] *Regulation (EU) No 910/2014 of the European Parliament and of the Council of 23 July 2014 on electronic identification and trust services for electronic transaction in the internal market and repealing Directive 1999/93/EC*.

[164] Zabihollah Rezaee, William Ford, and Rick Elam. "Real-time accounting systems". In: *Internal Auditor* 57.2 (2000), pp. 62–62.

[165] Reza Rezaei, Thiam Kian Chiew, and Sai Peck Lee. "A review on E-business Interoperability Frameworks". In: *Journal of Systems and Software* 93 (2014), pp. 199–216. ISSN: 0164-1212. DOI: https://doi.org/10.1016/j.jss.2014.02.004. URL: https://www.sciencedirect.com/science/article/pii/S016412121400051X.

[166] Robert Richards. "XML Security". In: *Pro PHP XML and Web Services*. Apress, 2006, pp. 441–489. DOI: 10.1007/978-1-4302-0139-7_12.

[167] ripple.com. *Ripple*. URL: https://ripple.com/ (visited on 01/22/2022).

[168] Thomas Roessler et al. *XML Signature Syntax and Processing Version 1.1*. W3C Recommendation. https://www.w3.org/TR/2013/REC-xmldsig-core1-20130411/. W3C, Apr. 2013.

[169] Jelurida Swiss SA, ed. *What is Nxt?* URL: https://www.jelurida.com/nxt/what-is-nxt (visited on 01/22/2022).

[170] Sergio Sanchez Garcia and Ana Gomez Oliva. "Solving identity management and interoperability problems at pan-European level". In: *OTM Confederated International Conferences" On the Move to Meaningful Internet Systems"*. Springer. 2009, pp. 805–809.

[171] Arturo J Sanchez-Ruiz et al. "The Data Interoperability Problem as an Exemplary Case Study in the Development of Software Collaboration Environments". In: *International Conference on Collaborative Computing: Networking, Applications and Worksharing*. Springer. 2008, pp. 379–389.

[172] Rui António dos Santos Cruz et al. *Peer-to-Peer Streaming Tracker Protocol (PPSTP)*. RFC 7846. May 2016. DOI: 10.17487/RFC7846. URL: https://www.rfc-editor.org/info/rfc7846.

[173] RA. K. Saravanaguru et al. *Securing Web Services Using XML Signature and XML Encryption*. 2013. DOI: 10.48550/ARXIV.1303.0910.

[174] Eder J. Scheid et al. "Bifröst: a Modular Blockchain Interoperability API". In: *2019 IEEE 44th Conference on Local Computer Networks (LCN)*. IEEE, Oct. 2019, pp. 332–339. DOI: 10.1109/lcn44214.2019.8990860. URL: https://ieeexplore.ieee.org/abstract/document/8990860.

[175] David Schwartz, Noah Youngs, Arthur Britto, et al. "The Ripple Protocol Consensus Algorithm.(2014)". In: (2014).

[176] Alex Selkirk. "XML and security". In: *BT Technology Journal* 19.3 (2001), pp. 23–34.

[177] Kwangsoo Seol et al. "Privacy-Preserving Attribute-Based Access Control Model for XML-Based Electronic Health Record System". In: *IEEE Access* 6 (2018), pp. 9114–9128. DOI: 10.1109/access.2018.2800288.

[178] Balazs Simon et al. "Evaluation of ws- standards based interoperability of soa products for the hungarian e-government infrastructure". In: *E-Government, E-Services and Global Processes*. Springer, 2010, pp. 19–31.

[179] Ankush Singla and Elisa Bertino. "Blockchain-Based PKI Solutions for IoT". In: *2018 IEEE 4th International Conference on Collaboration and Internet Computing (CIC)*. IEEE, Oct. 2018. DOI: 10.1109/cic.2018.00-45.

[180] Josh Spiegel, Jonathan Robie, and Michael Dyck. *XML Path Language (XPath) 3.1*. W3C Recommendation. W3C, Mar. 2017.

[181] International Organization for Standardization. *Document management — Portable document format — Part 1: PDF 1.7*. International Organization for Standardization, 2008. URL: https://www.iso.org/standard/51502.html.

[182] *Stuart Haber and Scott Stornetta: How our timestamping mechanism was used in Bitcoin*. en-US. Oct. 2021. URL: https://coingeek.com/stuart-haber-and-scott-stornetta-how-our-timestamping-mechanism-was-used-in-bitcoin-video/ (visited on 06/02/2022).

[183] Nick Szabo. *Bit Gold*. Ed. by Satoshi Nakamoto Institute. Dec. 29, 2005. URL: https://nakamotoinstitute.org/bit-gold/.

[184] Nick Szabo. *Bit gold*. Ed. by Unenumerated. URL: https://unenumerated.blogspot.com/2005/12/bit-gold.html (visited on 08/01/2022).

[185] Nick Szabo. "Secure property titles with owner authority". In: *Online at http://szabo.best. vwh. net/securetitle. html* (1998). (Visited on 11/02/2022).

[186] Nick Szabo. "Smart Contracts : Building Blocks for Digital Markets". In: 2018.

[187] Maurizio Talamo et al. "A Blockchain based PKI Validation System based on Rare Events Management". In: *Future Internet* 12.2 (Feb. 2020), p. 40. DOI: 10.3390/fi12020040.

[188] B3 Technologies, ed. *B3 Technologies*. URL: https://bi3technologies.com/ (visited on 01/22/2022).

[189] Henry Thompson et al. *W3C XML Schema Definition Language (XSD) 1.1 Part 2: Datatypes*. W3C Recommendation. W3C, Apr. 2012.

[190] Mohsen Toorani and Christian Gehrmann. "A decentralized dynamic pki based on blockchain". In: *Proceedings Of the 36th Annual ACM Symposium On Applied Computing*. 2021, pp. 1646–1655.

[191] *TRansaction*. URL: https://en.wiktionary.org%20/wiki/transact%20#English (visited on 04/22/2022).

[192] *Transaction*. URL: https://www.etymonline.com/word/transaction (visited on 04/22/2022).

[193] *Transakcja*. URL: https://mfiles.pl/pl/index.php/Transakcja (visited on 04/22/2022).

[194] Florian Tschorsch and Bjorn Scheuermann. "Bitcoin and Beyond: A Technical Survey on Decentralized Digital Currencies". In: *IEEE Communications Surveys & Tutorials* 18.3 (2016), pp. 2084–2123. DOI: 10.1109/comst.2016.2535718.

[195] Richard Twesige. "Bitcoin A simple explanation of Bitcoin and Block Chain technology JANUARY 2015 RICHARD LEE TWESIGE". en. In: (2015). DOI: 10.13140/2.1.1385.2486.

[196] *Ustawa z dnia 15 października w sprawie ogłoszenia jednolitego tekstu ustawy o informatyzacji działalności realizujących zadania publiczne.*

[197] Antonio Vallecillo, Juan Hernandez, and Jose M Troya. "New issues in object interoperability". In: *European Conference on Object-Oriented Programming*. Springer. 2000, pp. 256–269.

[198] Hans van der Veer and Anthony Wiles. *Achieving technical interoperability– the ETSI approach. ETSI white paper No. 3.* European Telecommunications Standards Institute: 2006.

[199] Rong Wang et al. "A Privacy-Aware PKI System Based on Permissioned Blockchains". In: *2018 IEEE 9th International Conference on Software Engineering and Service Science (ICSESS)*. IEEE, Nov. 2018. DOI: 10.1109/icsess.2018.8663738.

[200] Ting Wang et al. "A survey on the history of transaction management: from flat to grid transactions". In: *Distributed and Parallel Databases* 23.3 (Apr. 2008), pp. 235–270. DOI: 10.1007/s10619-008-7028-1.

[201] Wentong Wang, Ning Hu, and Xin Liu. "BlockCAM: A Blockchain-Based Cross-Domain Authentication Model". In: *2018 IEEE Third International Conference on Data Science in Cyberspace (DSC)*. IEEE, June 2018. DOI: 10.1109/dsc.2018.00143.

[202] Yanjiong Wang, Qiaoyan Wen, and Hua Zhang. "A Single Sign-On Scheme for Cross Domain Web Applications Using Identity-Based Cryptography". In: *2010 Second International Conference on Networks Security, Wireless Communications and Trusted Computing*. IEEE, 2010. DOI: 10.1109/nswctc.2010.120.

[203] Yunsen Wang and Alexander Kogan. "Designing confidentiality-preserving Blockchain-based transaction processing systems". In: *International Journal of Accounting Information Systems* 30 (Sept. 2018), pp. 1–18. DOI: 10.1016/j.accinf.2018.06.001.

[204] Gerard Wawrzyniak and Imed El Fray. "An Electronic Document for Distributed Electronic Services". In: *Computer Information Systems and Industrial Management*. Ed. by Khalid Saeed and Władysław Homenda. Cham: Springer International Publishing, 2016, pp. 617–630. ISBN: 978-3-319-45378-1. DOI: 10.1007/978-3-319-45378-1_54.

[205] Gerard Wawrzyniak and Imed El Fray. "Electronic Document Interoperability in Transactions Executions". In: *Advances in Soft and Hard Computing*. Ed. by Jerzy Pejaś et al. Cham: Springer International Publishing, 2018, pp. 358–372. ISBN: 978-3-030-03314-9. DOI: 10.1007/978-3-030-03314-9_31.

[206] Gerard Wawrzyniak and Imed El Fray. "New XML Signature Scheme That is Resistant to Some Attacks". In: *IEEE Access* 8 (2020), pp. 35815–35831. ISSN: 2169-3536. DOI: 10.1109/ACCESS.2020.2975034.

[207] Gerard Wawrzyniak and Imed El Fray. "The Implementation of Electronic Document in Transaction Execution". In: *Computer Information Systems and Industrial Management*. Ed. by Khalid Saeed, Władysław Homenda, and Rituparna Chaki. Cham: Springer International Publishing, 2017, pp. 697–708. ISBN: 978-3-319-59105-6. DOI: 10.1007/978-3-319-59105-6_60.

[208] Gerard Wawrzyniak et al. "Concept of an electronic form based on XML that ensures a high level of interoperability and security". In: *Pacific Asia Conference*

on Information Systems 12th to 14th July. PACIS2021 (July 12, 2021). Dubai, 2021. URL: https://aisel.aisnet.org/pacis2021/68.

[209] Ahmad Samer Wazan et al. "PKI interoperability: still an issue? A solution in the X. 509 realm". In: *Information Assurance and Security Education and Training.* Springer, 2013, pp. 68–82.

[210] Webster, ed. *Transaction.* URL: https://www.merriam-webster.com/dictionary/transacted (visited on 04/22/2022).

[211] Sanjiva Weerawarana et al. *Web Services Description Language (WSDL) Version 2.0 Part 1: Core Language.* W3C Recommendation. W3C, June 2007.

[212] ROLF T. WIGAND. "Electronic Commerce: Definition, Theory, and Context". In: *The Information Society* 13.1 (Mar. 1997), pp. 1–16. DOI: 10.1080/0197224971-29241.

[213] Wikipedia, ed. *Transakcja.* URL: https://pl.wikipedia.org/wiki/Transakcja_(finanse) (visited on 05/07/2022).

[214] Wikipedia, ed. *Transakcja (informatyka).* May 7, 2022. URL: https://pl.wikipedia.org/wiki/Transakcja_(informatyka).

[215] Gavin Wood et al. "Ethereum: A secure decentralised generalised transaction ledger". In: *Ethereum project yellow paper* 151.2014 (2014), pp. 1–32.

[216] Li Da Xu and Wattana Viriyasitavat. "Application of Blockchain in Collaborative Internet-of-Things Services". In: *IEEE Transactions on Computational Social Systems* 6.6 (2019), pp. 1295–1305. DOI: 10.1109/TCSS.2019.2913165. URL: https://ieeexplore.ieee.org/stamp/stamp.jsp?tp=&arnumber=8718260.

[217] Alexander Yakubov, Wazen Shbair, and Radu State. "BlockPGP: A Blockchain-Based Framework for PGP Key Servers". In: *2018 Sixth International Symposium on Computing and Networking Workshops (CANDARW).* IEEE, Nov. 2018. DOI: 10.1109/candarw.2018.00065.

[218] Alexander Yakubov et al. "A blockchain-based PKI management framework". In: *The First IEEE/IFIP International Workshop on Managing and Managed by Blockchain (Man2Block) colocated with IEEE/IFIP NOMS 2018, Tapei, Tawain 23-27 April 2018.* 2018.

[219] D Yermack. *Corporate Governance and Blockchains. Rev Financ 21, 7–31.* 2017.

[220] Jiangshan Yu and Mark Ryan. "Evaluating Web PKIs". In: *Software Architecture for Big Data and the Cloud.* Elsevier, 2017, pp. 105–126. DOI: 10.1016/b978-0-12-805467-3.00007-7.

[221] Internetowa Encykloepdia Zarządzania, ed. *Transakcja.* URL: https://mfiles.pl/pl/index.php/Transakcja (visited on 05/07/2022).

[222] Zcash, ed. *Zcash.* URL: https://z.cash/ (visited on 01/22/2022).

[223] Rui Zhang, Rui Xue, and Ling Liu. "Security and privacy on blockchain". In: *ACM Computing Surveys (CSUR)* 52.3 (2019), pp. 1–34. DOI: https://doi.org/10.1145/3316481. URL: https://dl.acm.org/doi/abs/10.1145/3316481 (visited on 01/23/2022).

[224] Tao Zhang et al. "A Modeling Approach for Service-Oriented Architecture". In: *Research and Practical Issues of Enterprise Information Systems.* Ed. by A. Min Tjoa, Li Xu, and Sohail S. Chaudhry. Boston, MA: Springer US, 2006, pp. 63–72. ISBN: 978-0-387-34456-0.

[225] Jun Zou, Yan Wang, and Mehmet A. Orgun. "A Dispute Arbitration Protocol Based on a Peer-to-Peer Service Contract Management Scheme". In: *2016 IEEE*

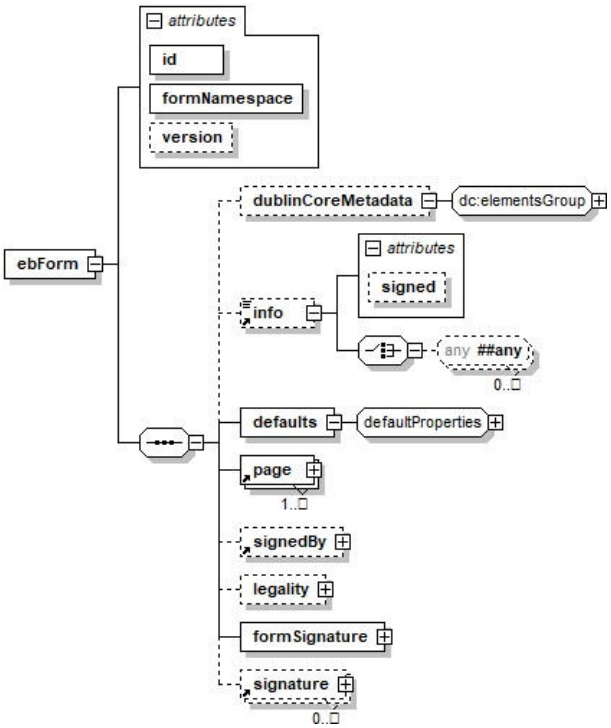
International Conference on Web Services (ICWS). IEEE, June 2016. DOI: 10.1109/icws.2016.15.

[226] Meiyun Zuo and Bei Wu. "SOA Oriented Web Services Operational Mechanism". In: *IFIP — The International Federation for Information Processing*. Springer US, pp. 753–762. DOI: 10.1007/978-0-387-75902-9_85.

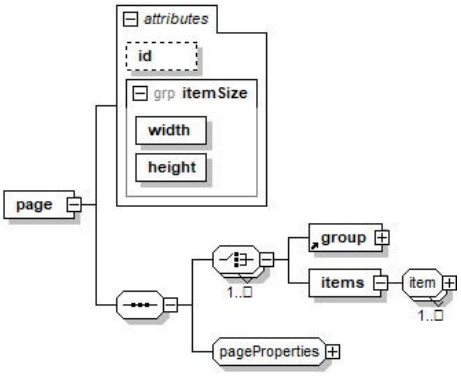
A. Składnia formularza elektronicznego

Załącznik A zawiera zbiór schematów prezentujących składnię pliku XSD notacji zastosowaniej przez oprogramowanie XML-Spy (Altova).

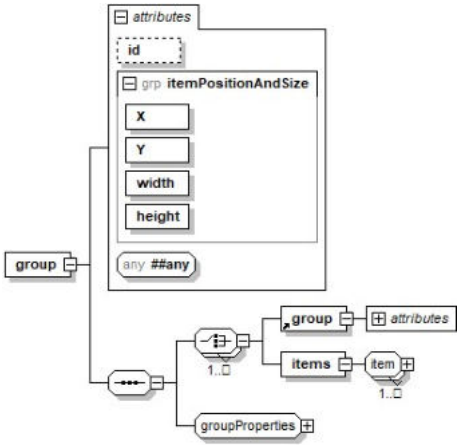
A.1. Ogólna struktura formularza



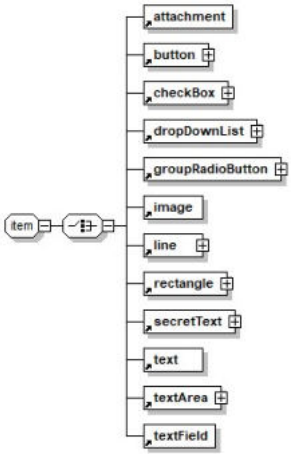
Rys. A.1–1: Ogólna budowa formularza



(a) Strona formularza

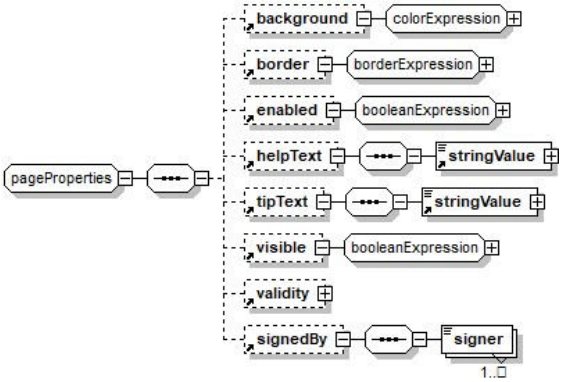


(b) Grupa

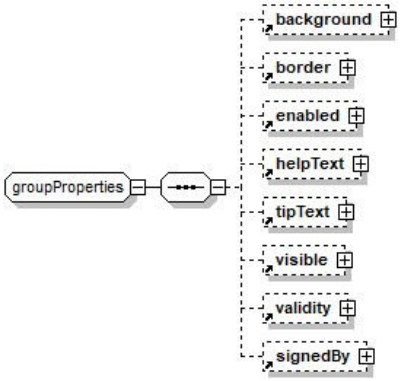


(c) Komponenty graficzne

Rys. A.1–2: Struktura strony i elementu grupującego w formularzu oraz komponenty graficzne

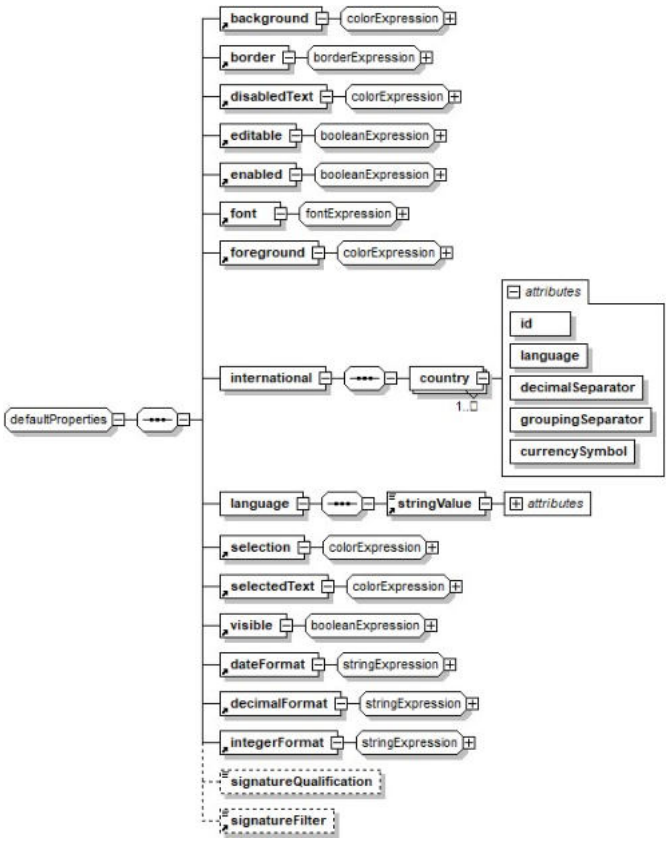


(a) Własności strony

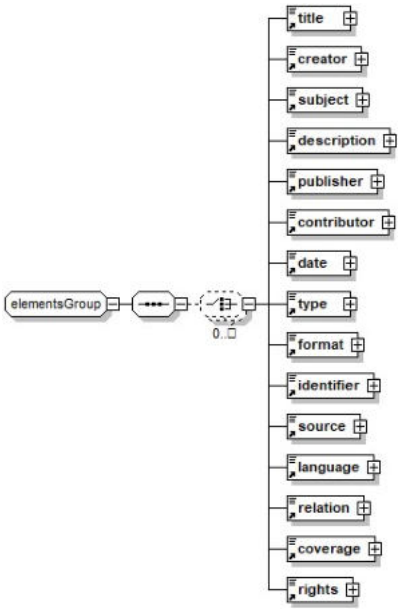


(b) Komponenty graficzne

Rys. A.1–3: Własności strony i grupy



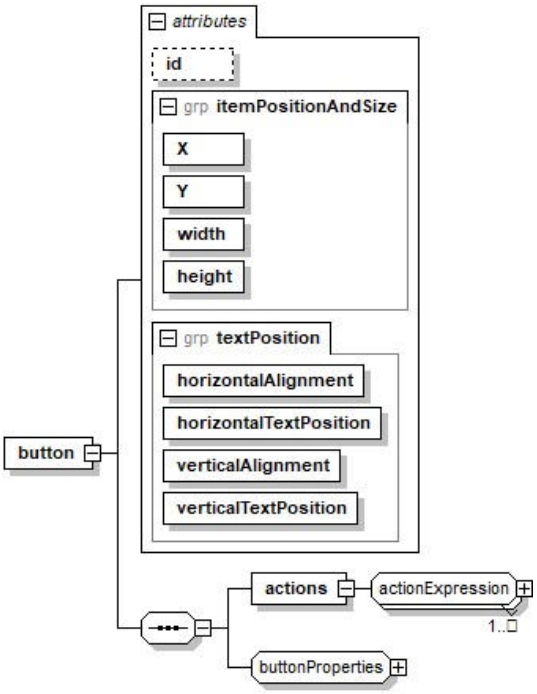
(a) Wartości domyślne



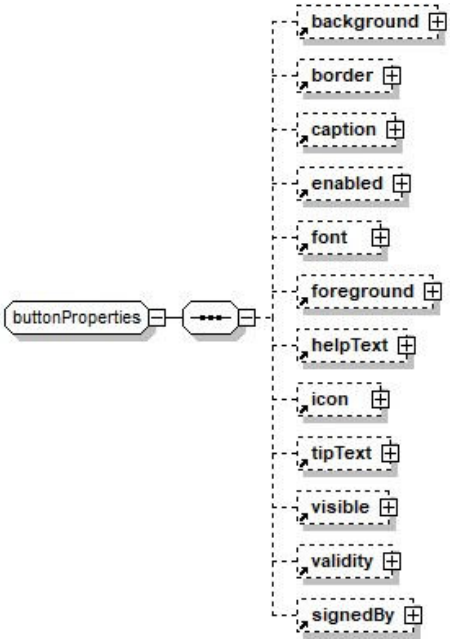
(b) Metadane Dublin Core

Rys. A.1–4: Wartości domyślne i metadane w formularzu

A.2. Graficzne komponenty formularza

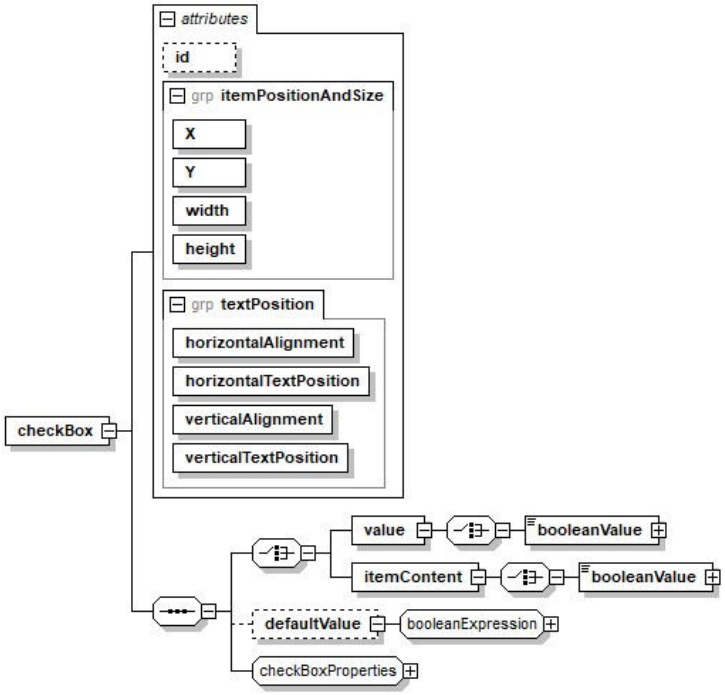


(a) Składnia definicji komponentu

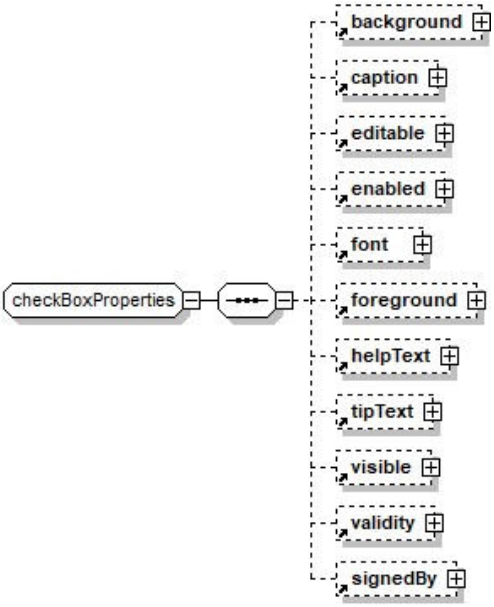


(b) Cechy komponentu

Rys. A.2–1: Składnia komponentu *Przycisk* `<button>`

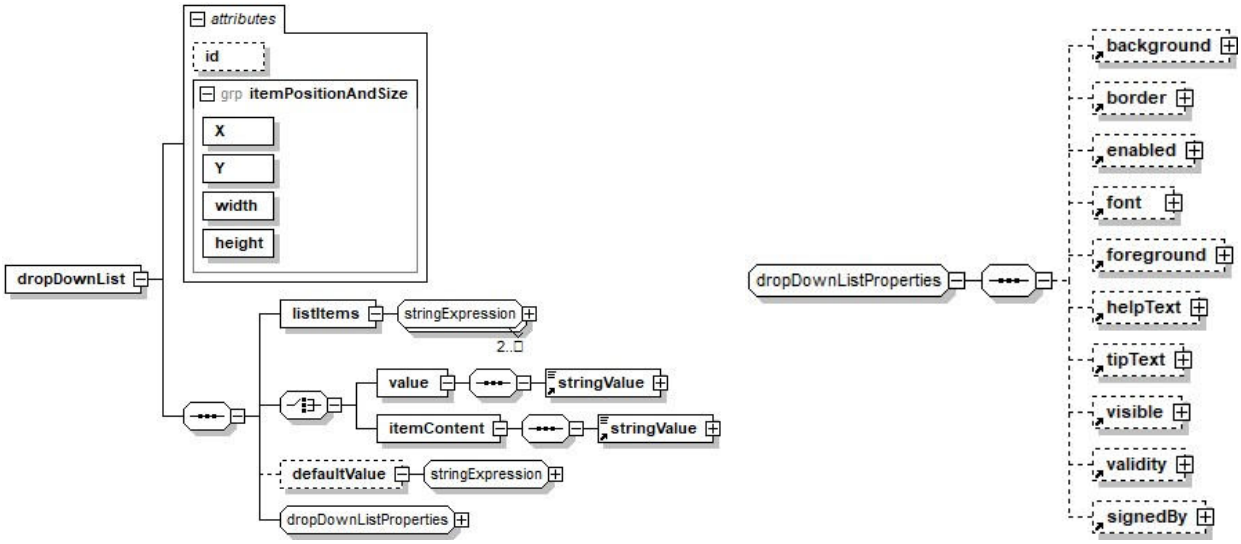


(a) Składnia definicji komponentu



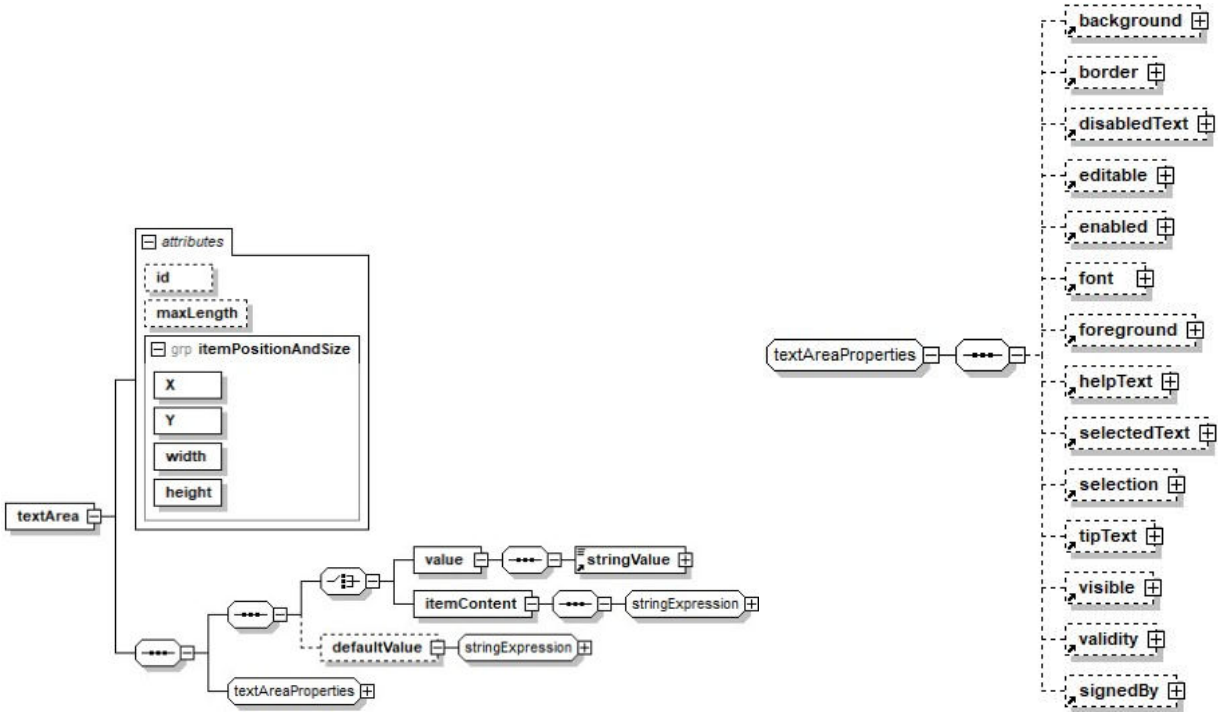
(b) Cechy komponentu

Rys. A.2–2: Składnia komponentu *Przycisk wyboru* `<checkBox>`



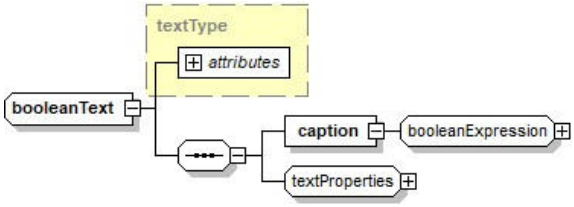
(a) Składnia definicji komponentu (b) Cechy komponentu

Rys. A.2–3: Składnia komponentu *Lista rozwijalna* `<dropDownList>`

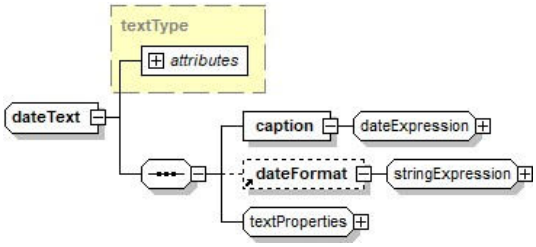


(a) Składnia definicji komponentu (b) Cechy komponentu

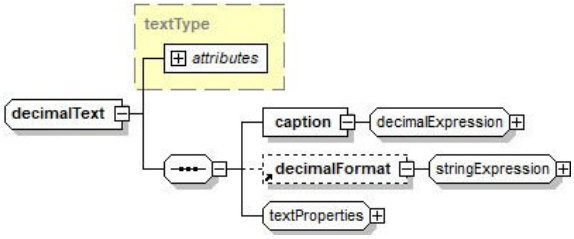
Rys. A.2–4: Składnia komponentu *Obszar tekstu* `<textArea>`



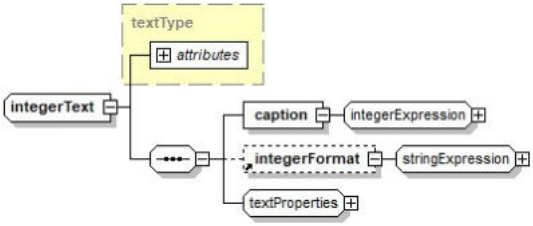
(a) Tekst logiczny



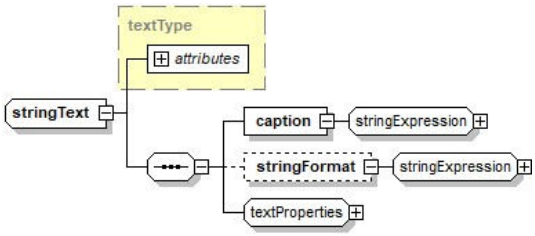
(b) Tekst daty



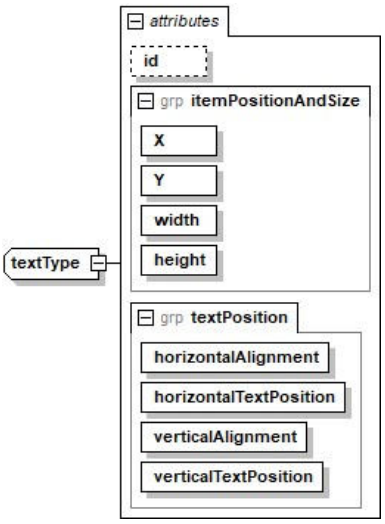
(c) Tekst liczby dziesiętnej



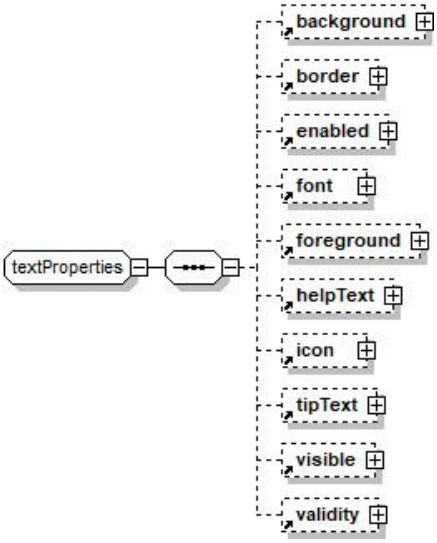
(d) Tekst liczby całkowitej



(e) Tekst typu napis

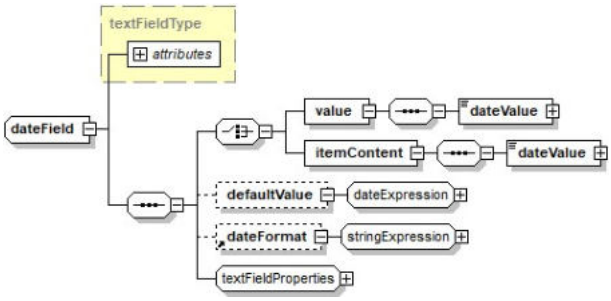


(f) Typ atrybutów tekstu

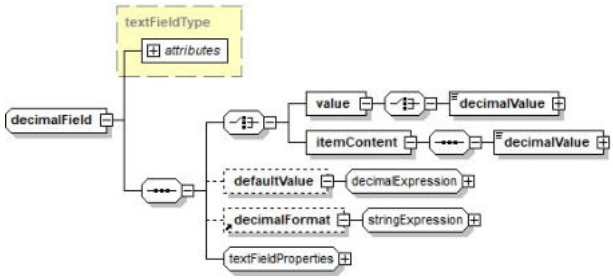


(g) Cechy tekstu

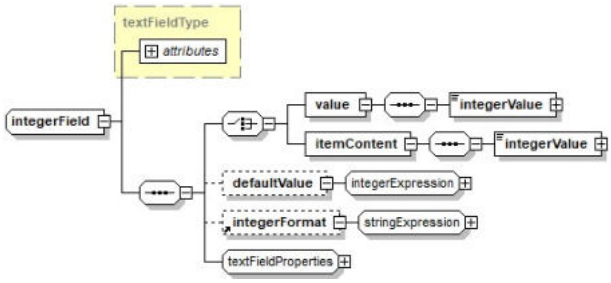
Rys. A.2–5: Składnia komponentu *Tekst* `<text>`



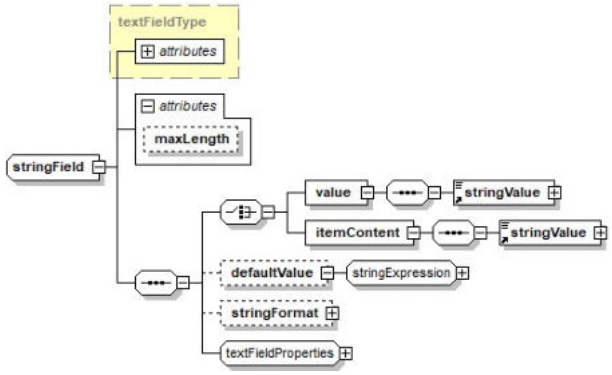
(a) Pole danych - data



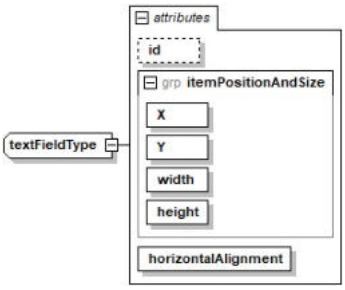
(b) Pole danych - wartość dziesiętna



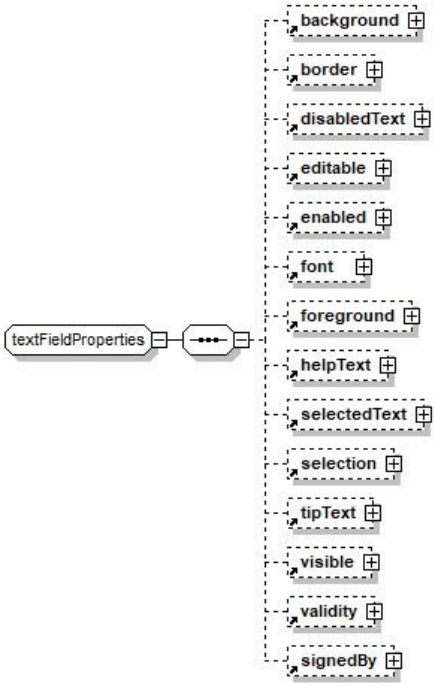
(c) Pole danych - wartość całkowita



(d) Pole danych - wartość tekstowa

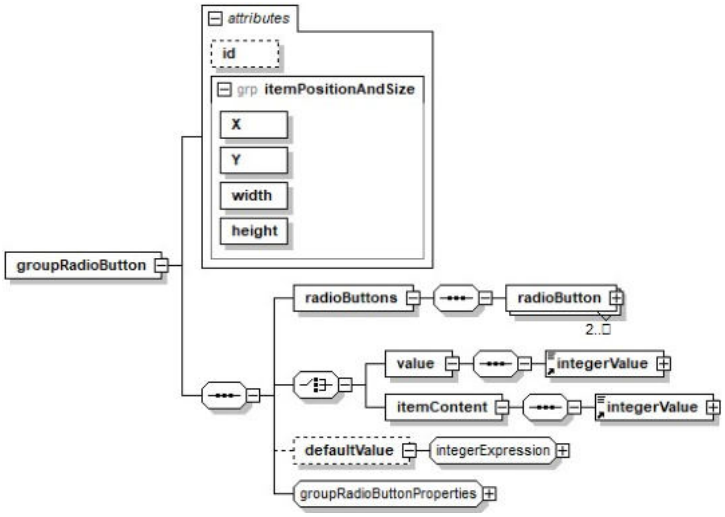


(e) Typ atrybutów pola danych

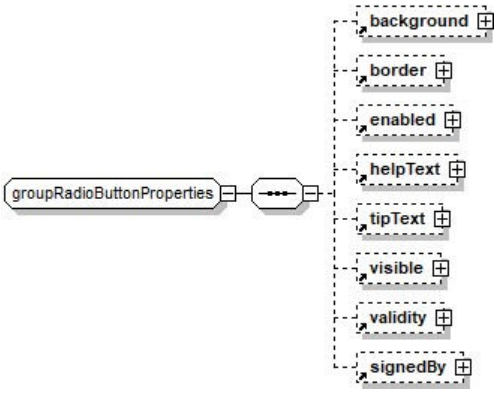


(f) Cechy pola danych

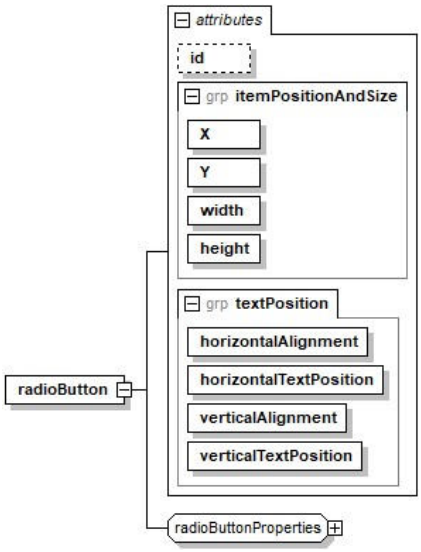
Rys. A.2–6: Składnia komponentu *Pole danych* `<textField>`



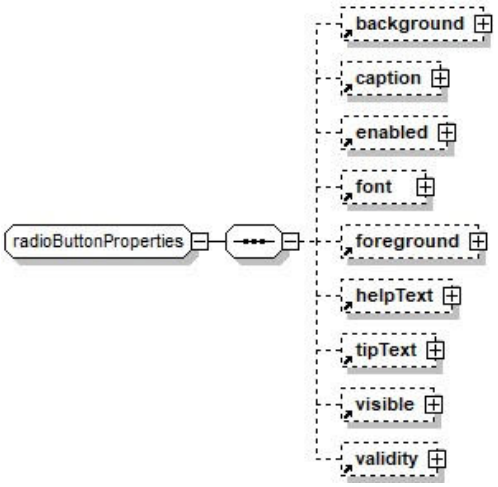
(a) Składnia definicji komponentu



(b) Cechy komponentu

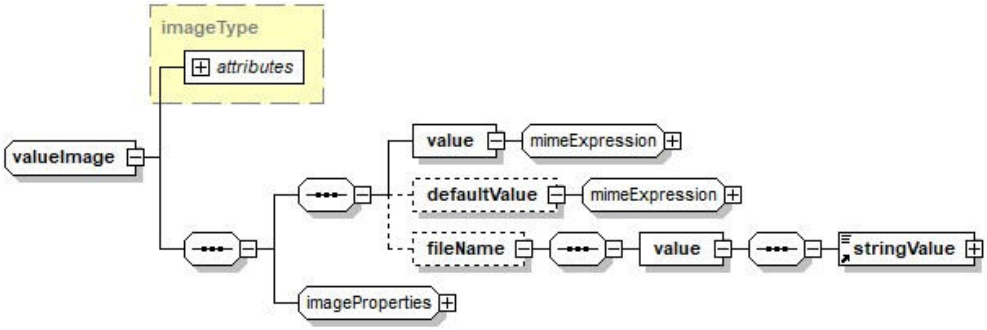


(c) Składnia definicji komponentu

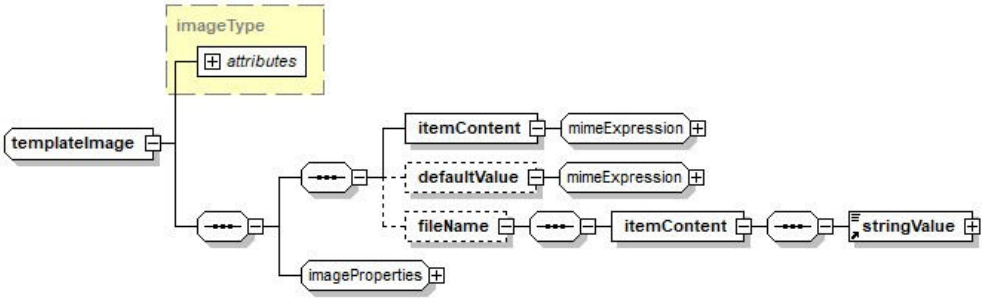


(d) Cechy komponentu

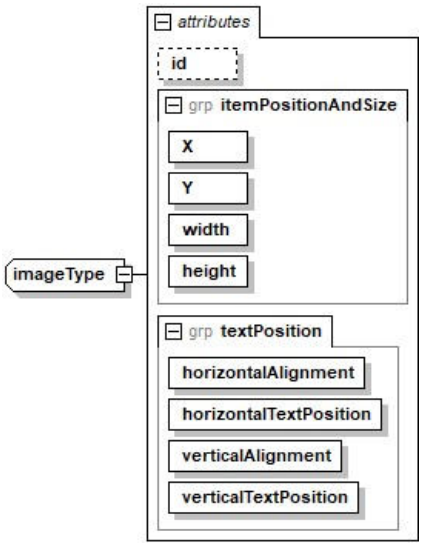
Rys. A.2–7: Składnia komponentu *Grupa przycisków opcji* `<groupRadioButton>`



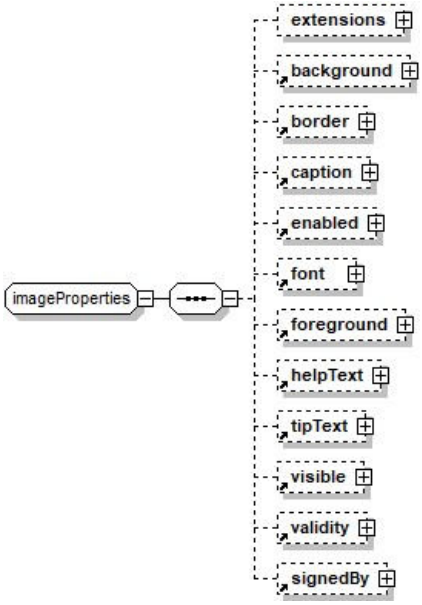
(a) Obraz - wartość



(b) Obraz - szablon

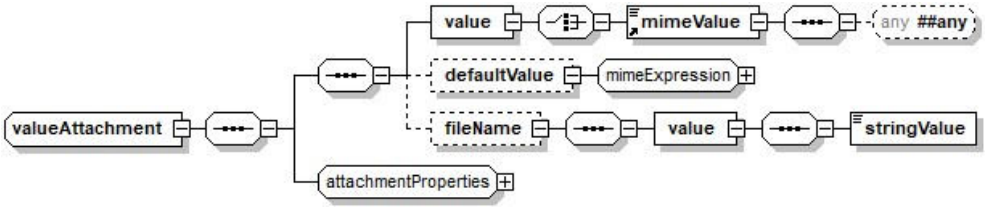


(c) Stałe atrybuty komponentu

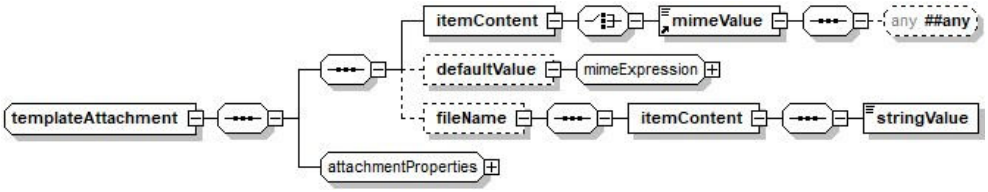


(d) Cechy komponentu

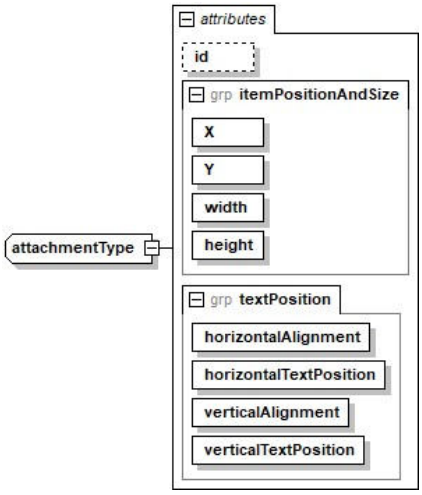
Rys. A.2–8: Składnia komponentu *Obraz* `<image>`



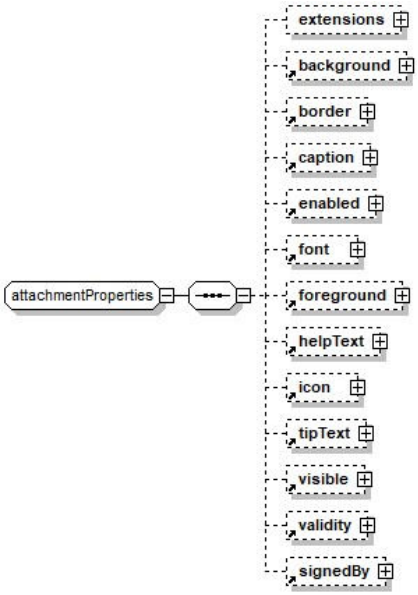
(a) Załącznik - wartość



(b) Załącznik - szablon

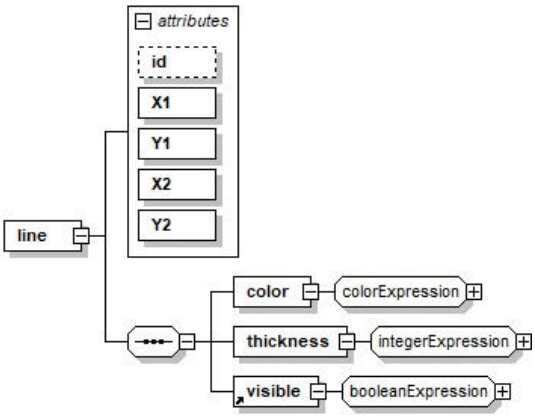


(c) Stałe atrybuty komponentu



(d) Cechy komponentu

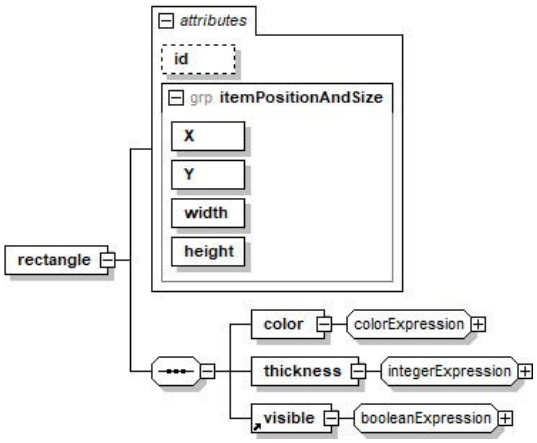
Rys. A.2–9: Składnia komponentu *Załącznik* `<attachment>`



Komponent graficzny *linia*, którego położenie wyznaczają współrzędne $X1, Y1, X2, Y2$, kolor komponentu jest określony przez wyrażenia `(colorExression)` w elemencie `<color>`, `(integerExpression)` oraz widzialność `(booleanExpression)`

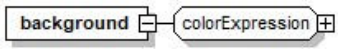
Rys. A.2–10: Składnia komponentu *Linia* `<line>`

Komponent graficzny *prostokąt*, którego położenie wyznaczają współrzędne $X1, Y1, X2, Y2$, kolor komponentu jest określony przez wyrażenia `(colorExression)` w elemencie `<color>`, `(integerExpression)` oraz widzialność `(booleanExpression)`



Rys. A.2–11: Składnia komponentu *Prostokąt* `<rectangle>`

A.3. Dynamiczne atrybuty komponentów formularza

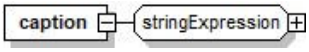


Definiowanie tła komponentu poprzez wskazanie wyrażenia (`colorExpression`)

Rys. A.3–1: Tło komponentu `<background>`

Definiowanie obramowania komponentu poprzez wskazanie wyrażenia (`borderExpression`)

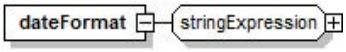
Rys. A.3–2: Obramowanie komponentu `<border>`



Definiowanie tytułu komponentu poprzez wskazanie wyrażenia (`stringExpression`)

Rys. A.3–3: Tytuł komponentu `<caption>`

Definiowanie formatowania daty wskazanej w wyrażeniu (`dateExpression`)

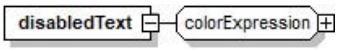


Rys. A.3–4: Formatowanie daty `<dateFormat>`

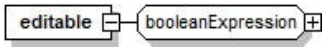
Definiowanie formatowania prezentacji wartości dziesiętnej poprzez wskazanie wyrażenia (`stringExpression`)

Rys. A.3–5: Format wartości dziesiętnej `<decimalFormat>`

Definiowanie koloru tekstu komponentu nieaktywanego (*disabled*), kolor wskazywany jest przez wyrażenie (`colorExpression`)



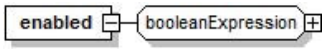
Rys. A.3–6: Kolor nieaktywnego tekstu `<disabledText>`



Definiowanie możliwości edycji wartości przechowywanej w komponencie poprzez wskazanie wyrażenia logicznego (`booleanExpression`)

Rys. A.3–7: Edytowalność komponentu `<editable>`

Definiowanie dostępności komponentu poprzez wskazanie wyrażenia logicznego (`booleanExpression`)

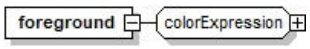


Rys. A.3–8: Dostępność komponentu `<enabled>`

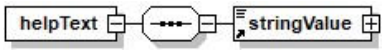
Definiowanie czcionki komponentu poprzez wskazanie wyrażenia (`fontExpression`)

Rys. A.3–9: Definiowanie czcionki `<font>`

Definiowanie koloru czcionki komponentu poprzez wskazanie wyrażenia (`colorExpression`)



Rys. A.3–10: Kolor czcionki komponentu `<foreground>`

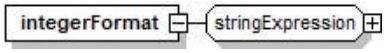


Definiowanie tektu pomocy poprzez wskazanie wyrażenia (`stringExpression`)

Rys. A.3–11: Tekst pomocy `<helpText>`

Definiowanie ikonki komponentu poprzez wskazanie wyrażenia `<mimeExpression>`

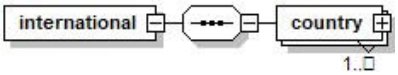
Rys. A.3–12: Ikonka `<icon>`



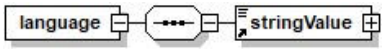
Definiowanie sposobu formatowania i wyświetlania wartości całkowitej poprzez wskazanie wyrażenia (`stringExpression`)

Rys. A.3–13: Formatowanie liczby całkowitej `<integerFormat>`

Definowanie ustwień międzynarodowych, takich jak wskazanie języków formularza zdefiniowane w elementach `<country>` zawierających dane „lokalizacyjne”: kod języka, separator dziesiętny, separator tysięcy, symbol waluty



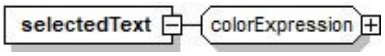
Rys. A.3–14: Ustawienia międzynarodowe `<international>`



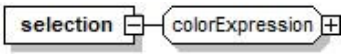
Definiowanie języka formularza wskazanego w elemencie `<stringValue>`

Rys. A.3–15: Język formularza `<languge>`

Definiowanie koloru wybranego (wskazanego) tekstu poprzez wskazanie wyrażenia (`colorExpression`)



Rys. A.3–16: Kolor tekstu wybranego `<selectetText>`

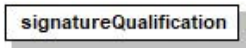


Definiowanie koloru wybranego komponentu poprzez wskazanie wyrażenia (`colorExpression`)

Rys. A.3–17: Kolor wskazanego komponentu `<selection>`

Definiowanie sposobu wybierania elementów XML do podpisu (XPath – standardowy lub ebForm – natywny dla formularza, niestandardowy, ale bardzo szybki)

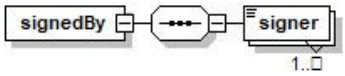
Rys. A.3–18: Sposób budowania zawartości podpisu `<signatureFilter>`



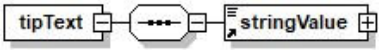
Definiowanie wymagalności odnośnie podpisu kwalifikowanego – podpis kwalifikowany obowiązkowy, pożądany lub brak wymagań.

Rys. A.3–19: Wymagania odnośnie podpisu `<signatureQualification>`

Definiowanie listy podpisów podpisujących komponent wskazywanych przez elementy `<signer>`



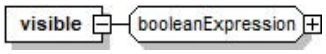
Rys. A.3–20: Podpisywanie komponentów `<signedBy>`



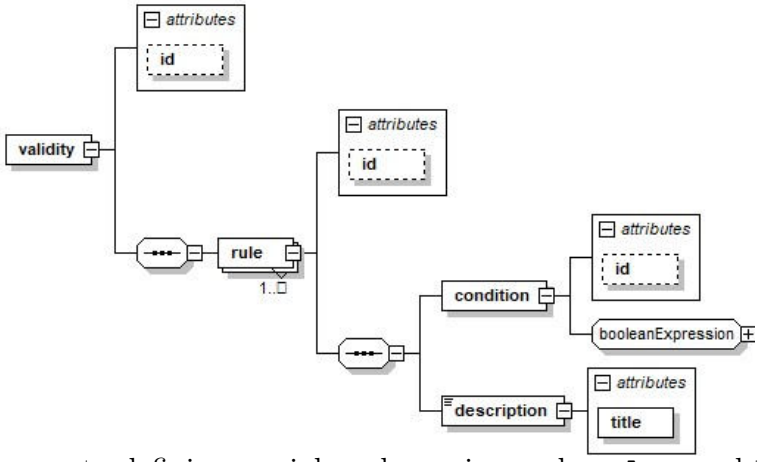
Definiowanie tekstu podpowiedzi tzw. „tip text”

Rys. A.3–21: Tekst podpowiedzi `<tipText>`

Definiowanie „widzialności” komponentu (czy komponent ma być wyświetlony, czy nie) poprzez wskazanie wyrażenia (`boolenExpression`)



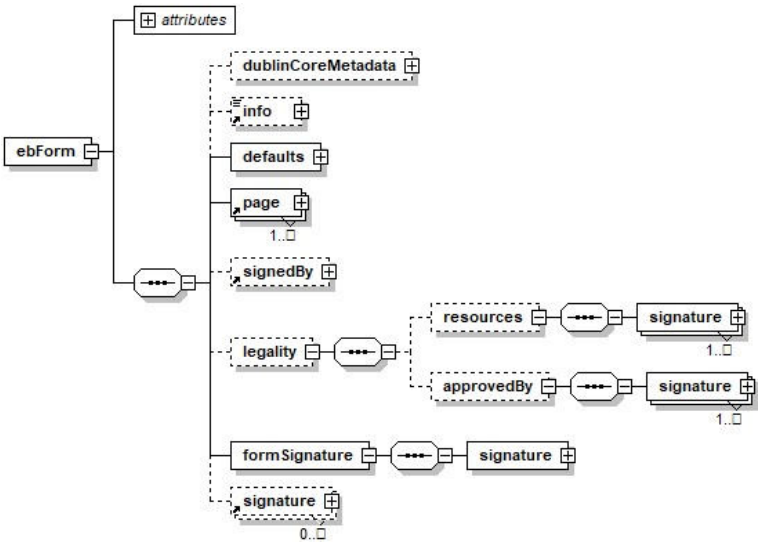
Rys. A.3–22: „Widzialność” komponentu `<visible>`



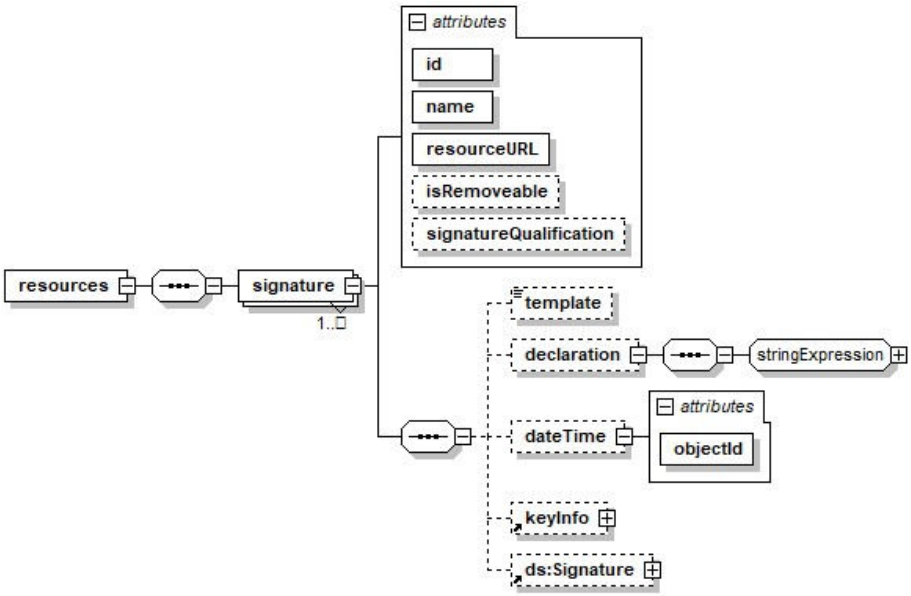
Poprawność komponentu definiowana jako sekwencja reguł `<rule>`, na które składają się: warunek `<condition>` zawierający wyrażenie logiczne oraz opis błędu

Rys. A.3–23: Reguły poprawności komponentu `<validity>`

A.4. Sytem podpisów

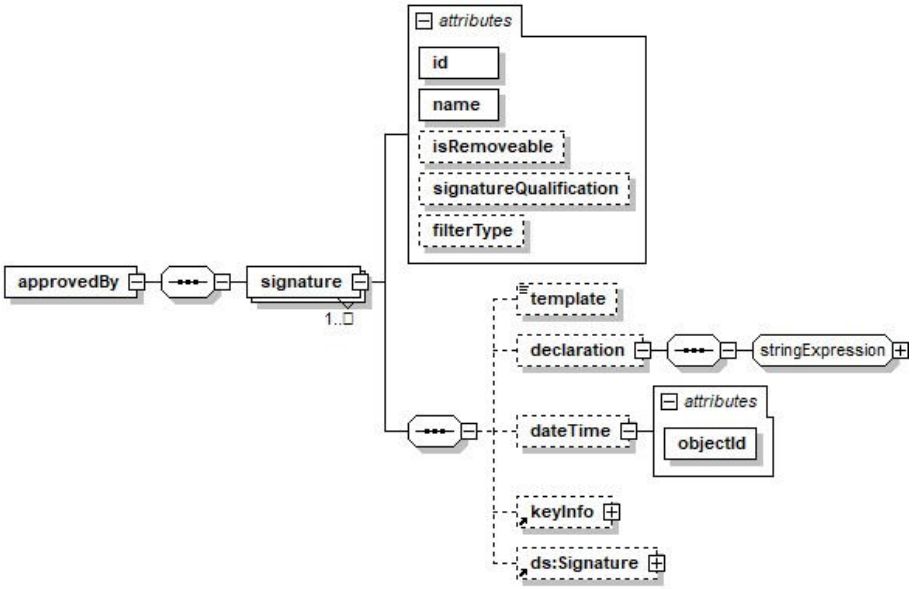


Rys. A.4–1: Struktura formularza – typy podpisów

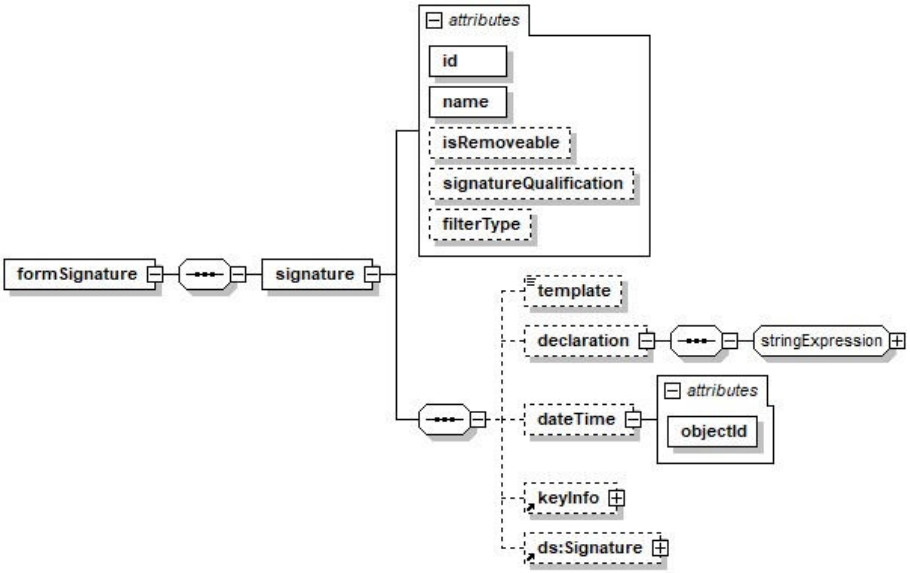


Rys. A.4–2: Typ podpisu – podpisy zasobów `<resources>` (Struktura keyInfo – rys. A.16–1b, A.12–20b)

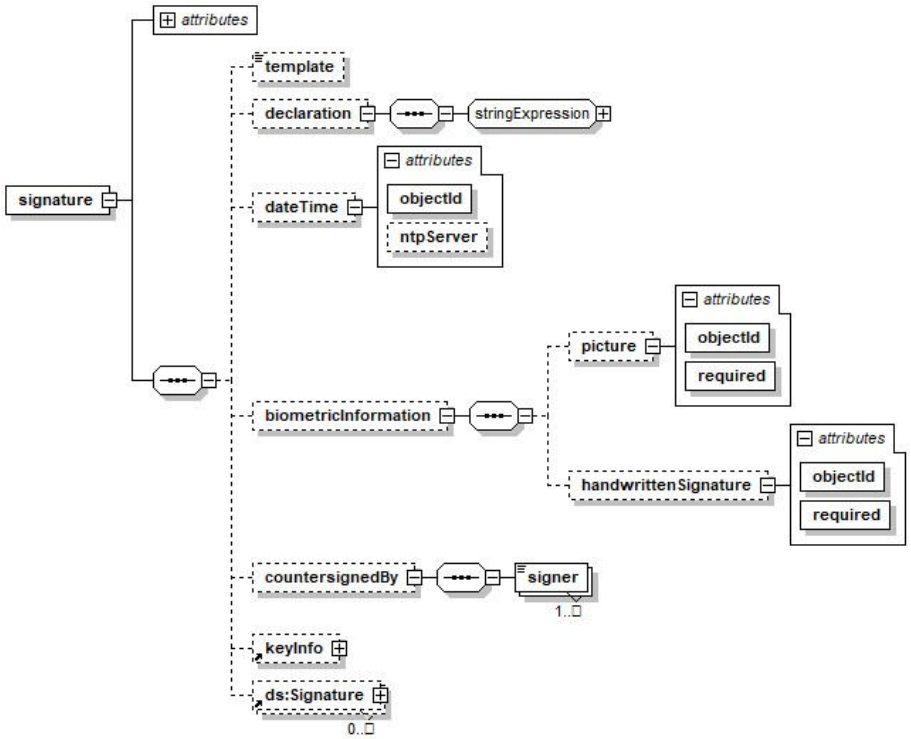
.



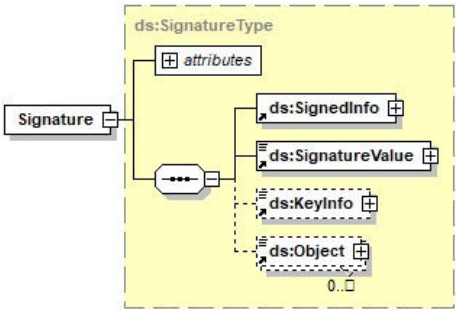
Rys. A.4–3: Typ podpisu – podpisy zatwierdzające `<approvedBy>` (Struktura keyInfo – rys. A.16–1b, A.12–20b)



Rys. A.4–4: Typ podpisu – podpis formularza `<formSignature>` (Struktura keyInfo – rys. A.16–1b, A.12–20b)



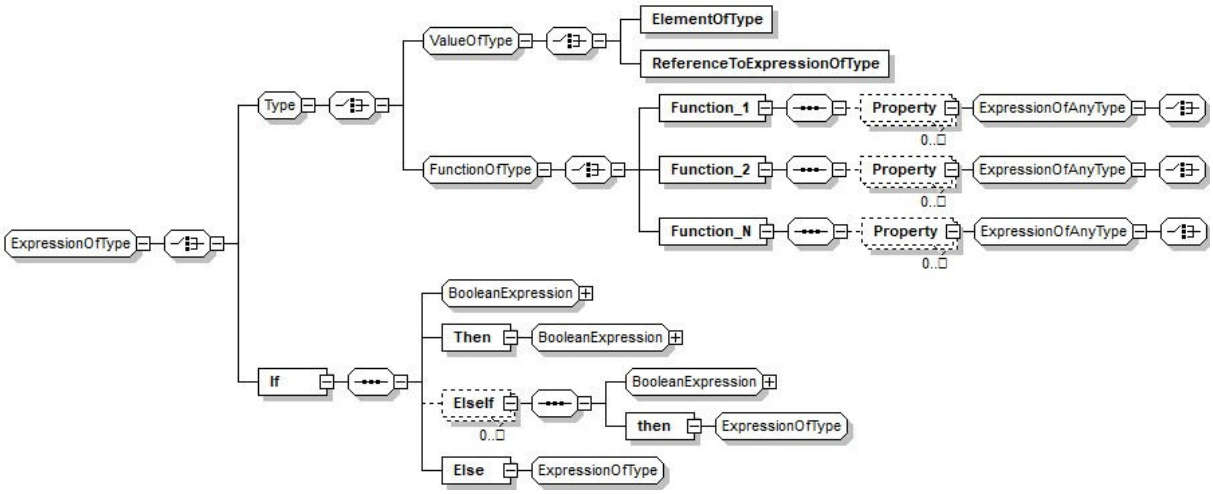
Rys. A.4–5: Typ podpisu – podpisy użytkowników `<signature>` (Struktura keyInfo – rys. A.16–1b, A.12–20b)



Rys. A.4–6: Struktura „standardowego” podpisu XML zgodnego z rekomendacją W3C

A.5. Wyrażenia

Definicja logiki formularza (trzeci wymiar, trzecia warstwa) jest nieodłącznym aspektem formularza i na niej oparta jest jego składnia. Podstawą definiowania logiki w tej warstwie są wyrażenia zapisane w strukturze formularza i interpretowane (wykonywane) przez aplikacje używające formularza. Na rysunku A.5–1 przedstawiono koncepcję definiowania składni dowolnych wyrażeń jako pojęcie abstrakcyjne. Tą abstrakcyjną składnię (strukturę) wykorzystano jako podstawę do zdefiniowania składni wyrażeń konkretnych funkcji określonego typu i działań (akcji). Abstrakcyjne wyrażenie dowolnego typu jest określone w sposób zaprezentowany na rysunku A.5–1.



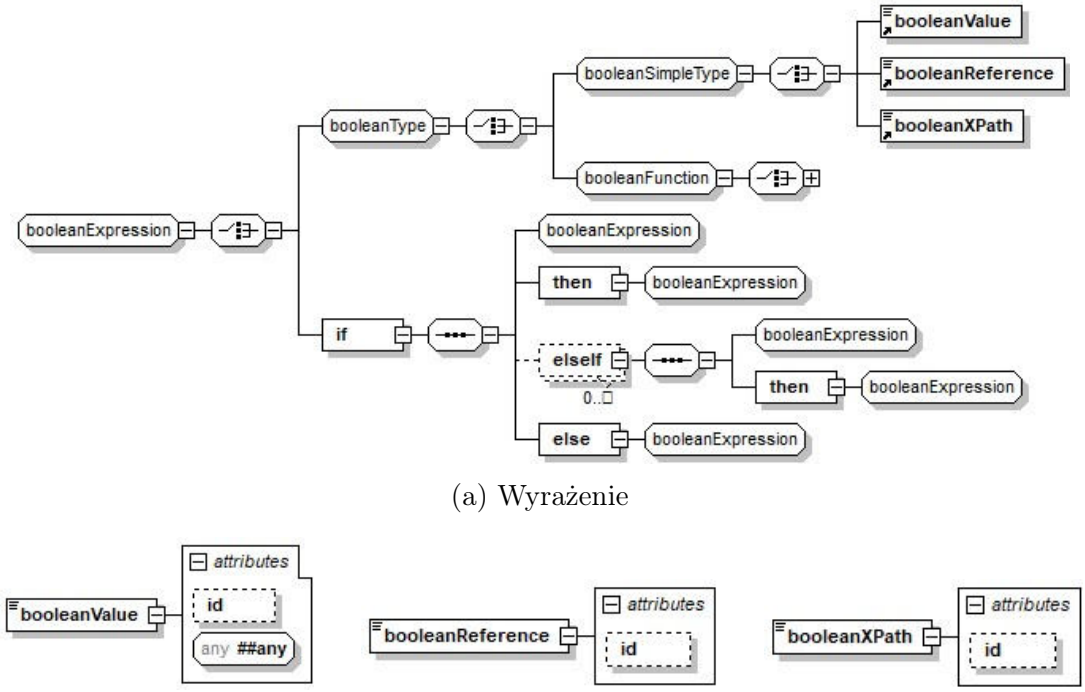
Rys. A.5–1: Koncepcja sposobu definiowania składni wyrażenia dowolnego typu (pojęcie abstrakcyjne) `<ExpressionOfType>`

Wyrażenie (dowolnego typu) definiuje sposób określenia wartości poprzez:

1. element `<ElementOfType>` wskazany wprost jako wartość (tego typu) zapisana w elemencie (XMLElement, wartość elementu dokumentu/formularza np. pole formularza),
2. element `<ReferenceToElementOfType>` będący referencją do wyrażenia tego typu,
3. element `<FuncyionOfType>`, czyli wartość wskazanej predefiniowanej funkcji obliczającej (zwracającej) wartość tego typu. Argumentami funkcji mogą być wyrażenia dowolnego typu,
4. wynik operacji logicznej wskazanej strukturą `<if>` zwracanej na podstawie aktualnej wartości wyrażenia logicznego `(BooleanExpression)`.

Wyrażenia są interpretowane i wykonywane przez aplikację w sposób dynamiczny, stosownie do aktualnego stanu danych/pól formularza.

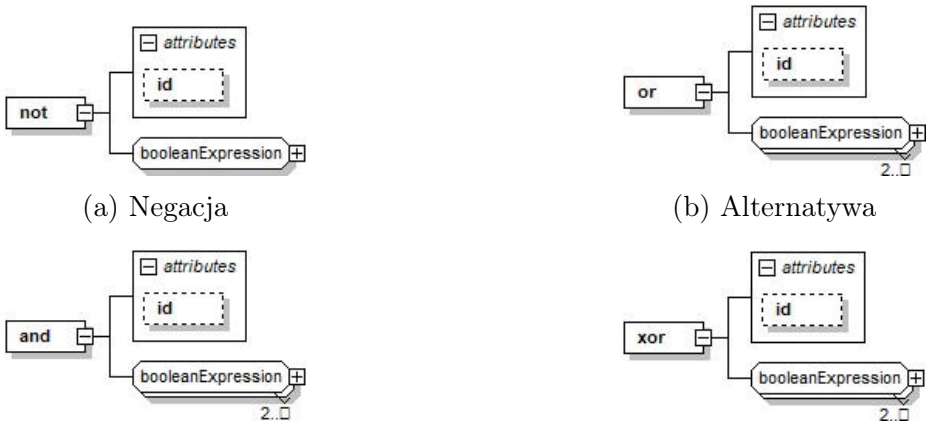
A.6. Wyrażenia logiczne (`booleanExpression`)



(a) Wyrażenie

(b) Wartość (c) Referencja (d) Wyrażenie XPath

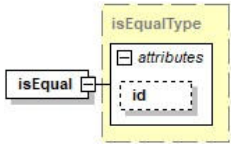
Rys. A.6–1: Składnia XML dla definiowania wyrażeń typu logicznego (`booleanExpression`)



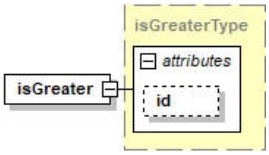
(a) Negacja (b) Alternatywa

(c) Koniunkcja (d) Alternatywa wykluczająca

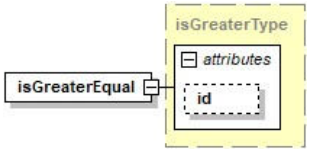
Rys. A.6–2: Operatory logiczne



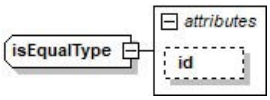
(a) Równość



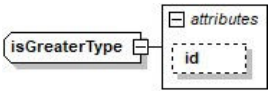
(b) Większość



(c) Relacja „większy równy”

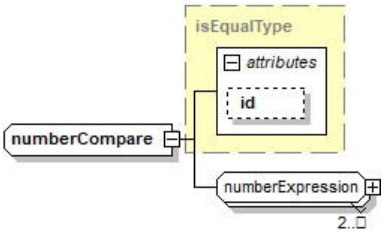


(d) Typ logiczny - „jest równy”

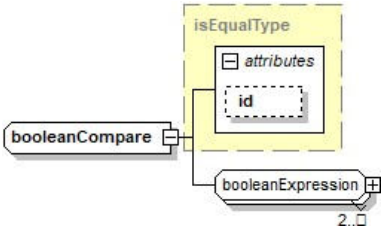


(e) Typ logiczny - „jest większy”

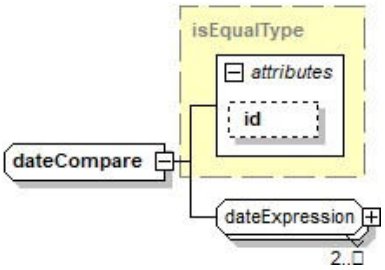
Rys. A.6–3: Typy porównań



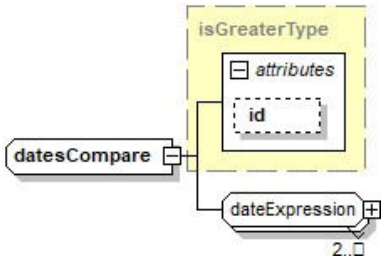
(a) Wartości numeryczne (dziesiętne i całkowite)



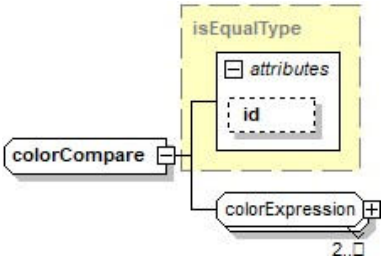
(b) Wartości logiczne



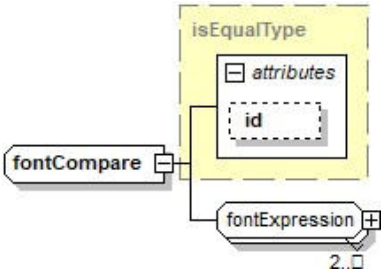
(c) Daty



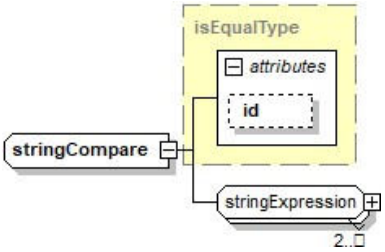
(d) Daty



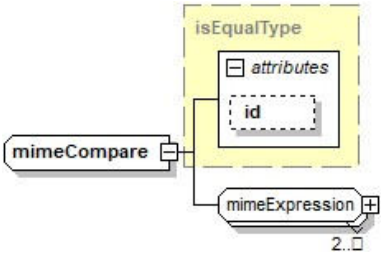
(e) Kolory



(f) Czcionka

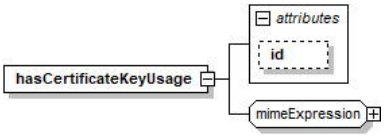


(g) Teksty

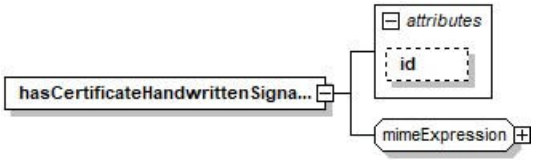


(h) Binaria

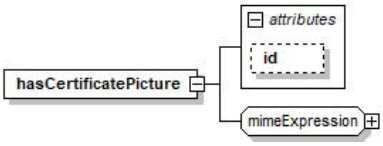
Rys. A.6–4: Operatory porównywania dla różnych typów porównywanych wartości



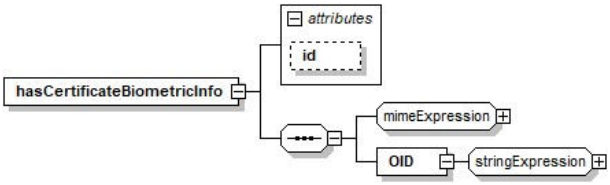
(a) Czy w certyfikacie podano użycie klucza



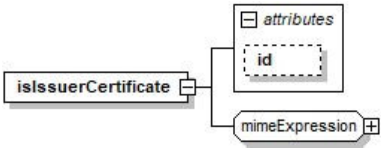
(b) Czy w certyfikacie podano dane o podpisie odręcznym



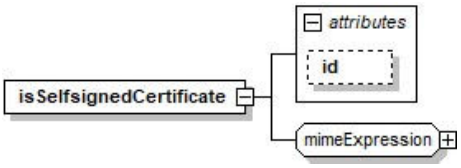
(c) Czy w certyfikacie podano dane o zdjęciu



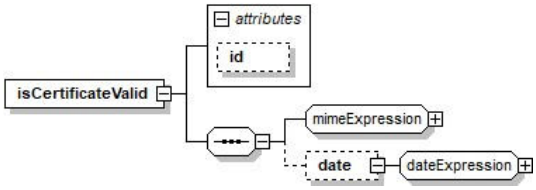
(d) Czy w certyfikacie podano dane biometryczne



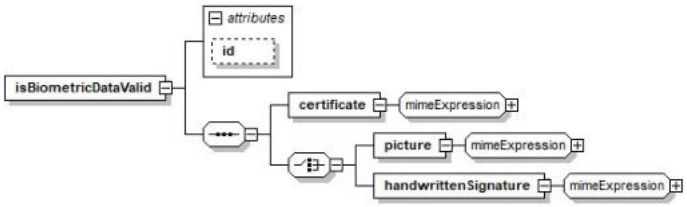
(e) Czy certyfikat jest certyfikatem wystawcy



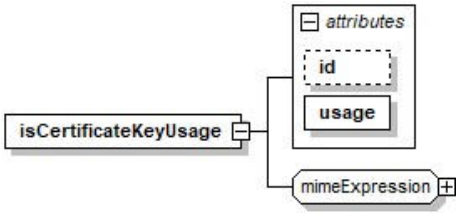
(f) Czy certyfikat jest autocertyfikatem



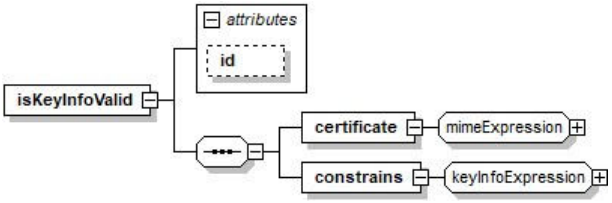
(g) Czy certyfikat jest ważny



(h) Czy dane biometryczne są poprawne

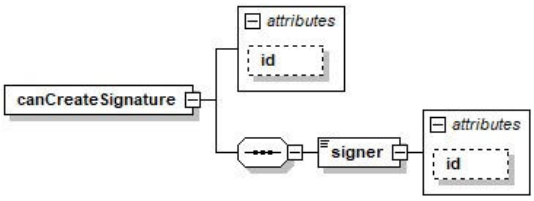


(i) Sprawdzenie zastosowania klucza

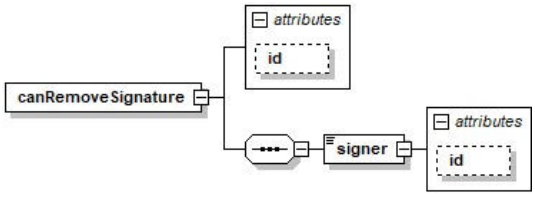


(j) Sprawdzenie poprawności danych o kluczu

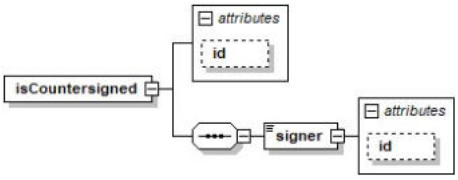
Rys. A.6–5: Funkcje logiczne obsługi certyfikatów



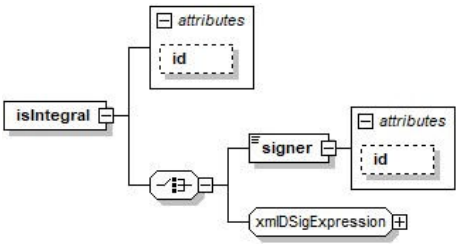
(a) Czy można zbudować podpis



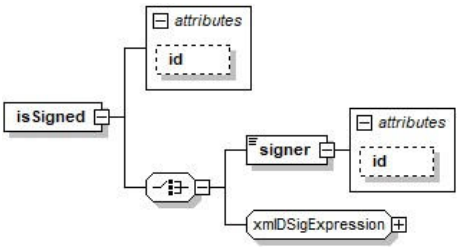
(b) Czy można zbudować podpis



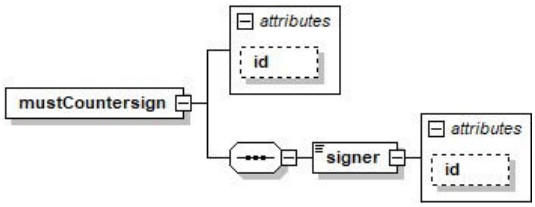
(c) Czy istnieje podpis kontrasygnujący



(d) Czy podpis jest integralny

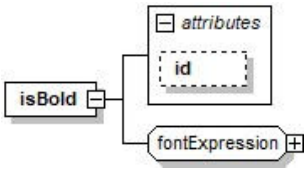


(e) Czy podpis jest wykonany

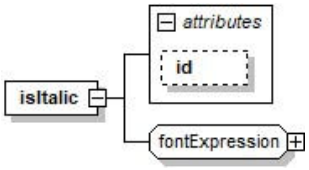


(f) Czy należy wykonać podpis kontrasygnujący

Rys. A.6–6: Funkcje logiczne obsługi podpisów

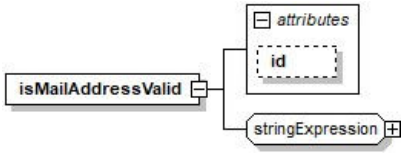


(a) Czy czcionka jest wytłuszczona

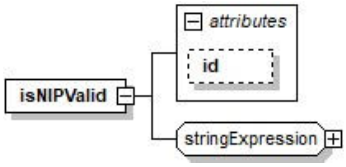


(b) Czy czcionka jest pochyła

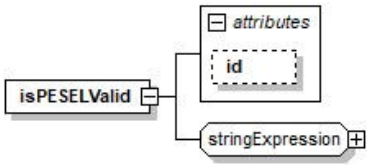
Rys. A.6–7: Funkcje logiczne obsługi czcionki



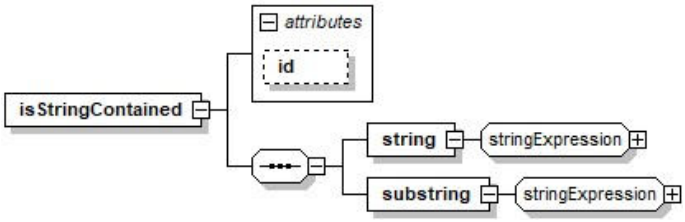
(a) Poprawność adresu e-mail



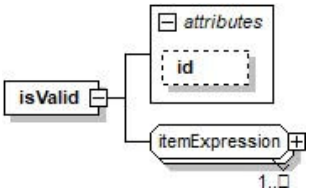
(b) Poprawność NIP



(c) Poprawność PESEL



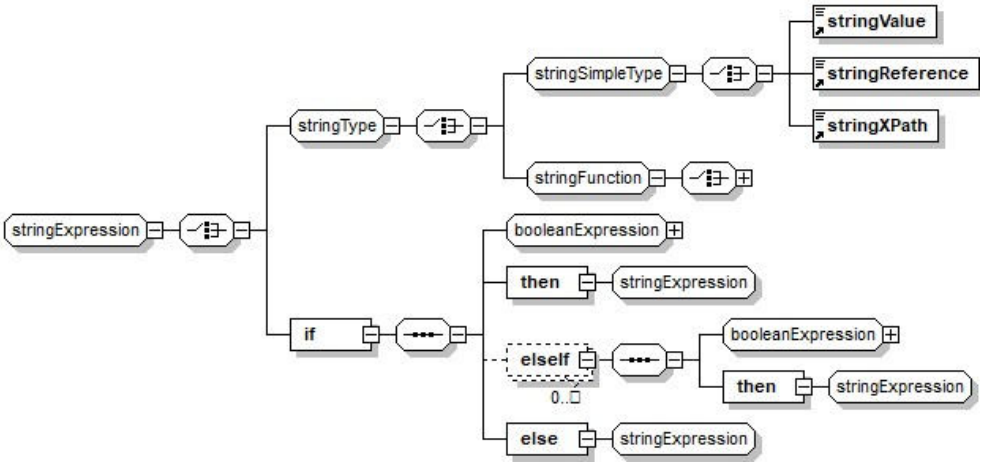
(d) Sprawdzenie zawierania tekstu



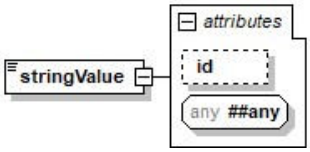
(e) Weryfikacja poprawności reguł w komponentach wskazanych przez `<itemExpression>`

Rys. A.6–8: Inne funkcje logiczne

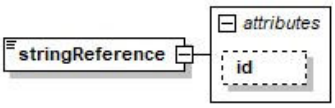
A.7. Wyrażenia napisów (stringExpression)



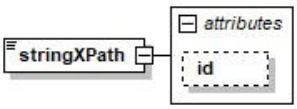
(a) Wyrażenie tekstu



(b) Wartość

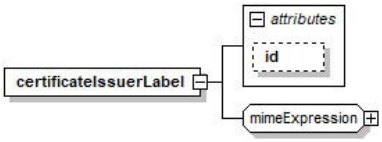


(c) Referencja

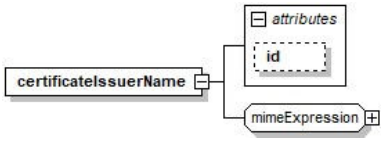


(d) Wyrażenie XPath

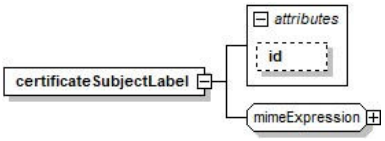
Rys. A.7–1: Składnia XML dla definiowania wyrażeń typu tekstowego `<stringExpression>`



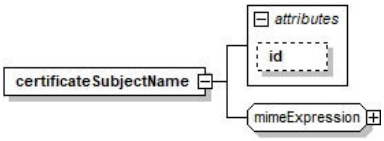
(a) Etykieta wystawcy certyfikatu



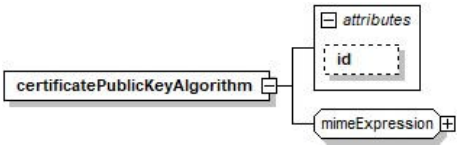
(b) Nazwa wystawcy ceryfikatu



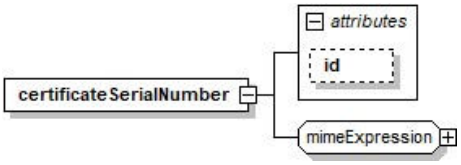
(c) Etykieta podmiotu



(d) Nazwa podmiotu

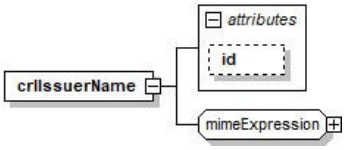


(e) Algorytm klucza pulicznego

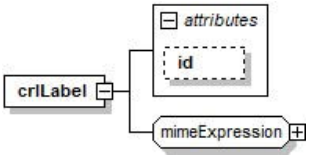


(f) Numer seryjny certyfikatu

Rys. A.7–2: Funkcje tekstu dla certyfikatów X.509

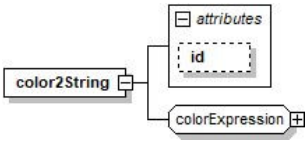


(a) Nazwa wystawcy

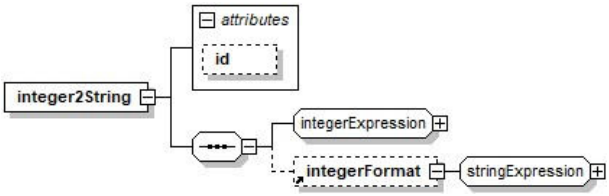


(b) Etykieta wystawcy

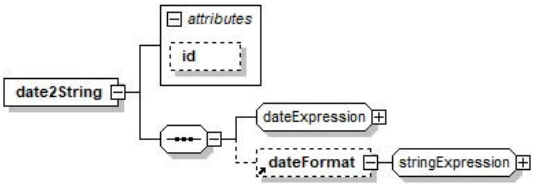
Rys. A.7–3: Funkcje obsługi listy certyfikatów unieważnionych



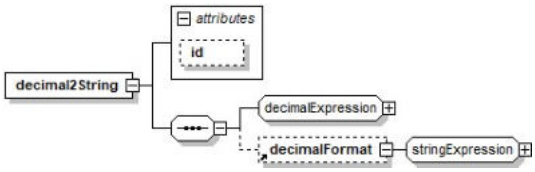
(a) Konwersja koloru



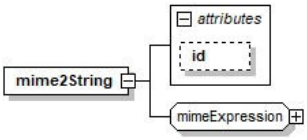
(b) Napis wartości całkowitej



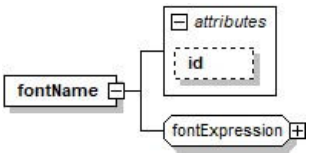
(c) Data



(d) Tekst z liczby dziesiętnej

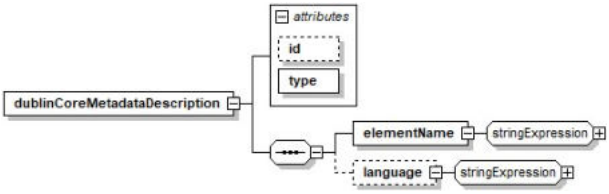


(e) Dekodowanie *Base64*

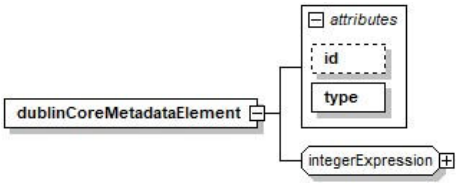


(f) Nazwa czcionki

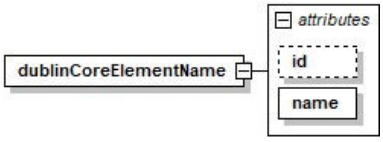
Rys. A.7–4: Funkcje konwersji do tekstu



(a) Opis elementu medtadanych Dublin Core

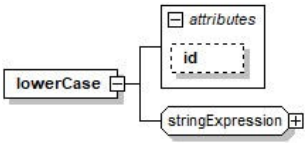


(b) Element metadanych Dublin Core

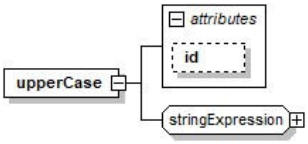


(c) Data

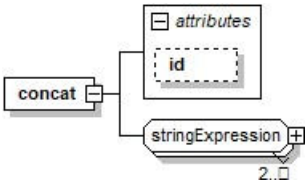
Rys. A.7–5: Dostęp do metadanych Dublin Core



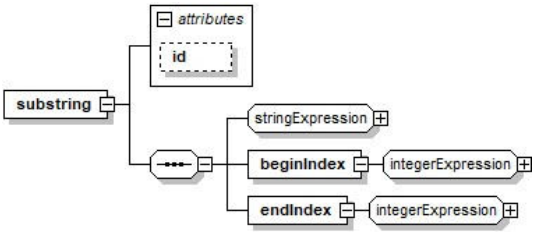
(a) Konwersja do małych liter



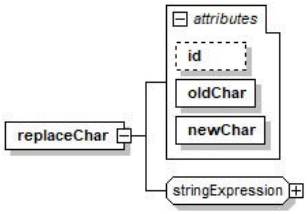
(b) Zamiana tekstu na wielkie znaki



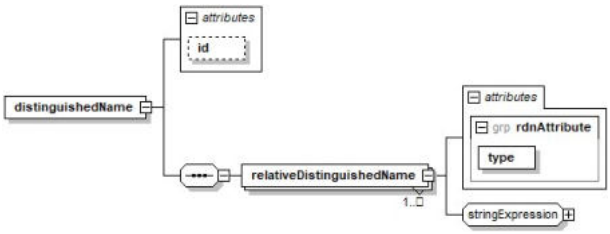
(c) Konkatenacja



(d) Fragment tekstu

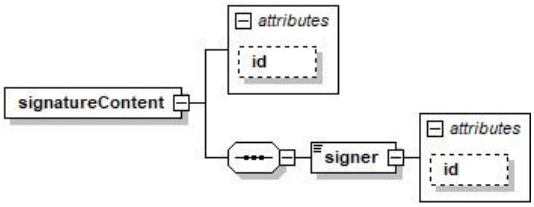


(e) Zmiana znaku w tekście

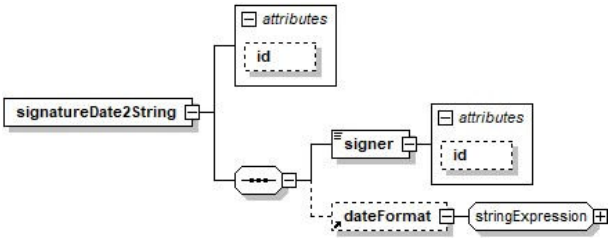


(f) Nazwa wyróżniona (konkatenacja nazw względnie wyróżnionych)

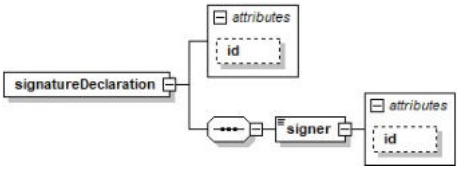
Rys. A.7–6: Standardowe funkcje obsługi tekstu



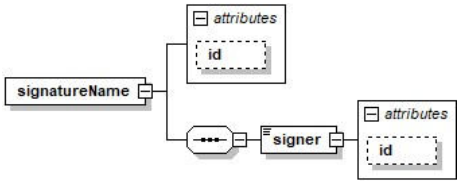
(a) Zawartość podpisu



(b) Tekst daty podpisu

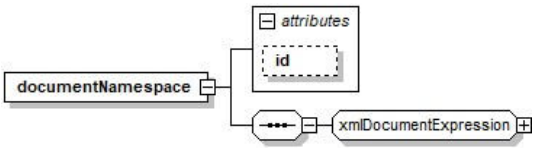


(c) Treść deklaracji podpisu

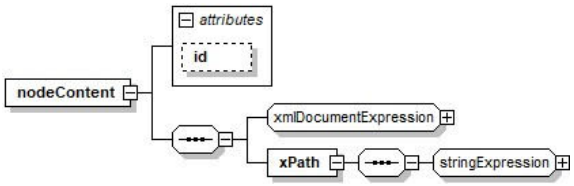


(d) Nazwa podpisu

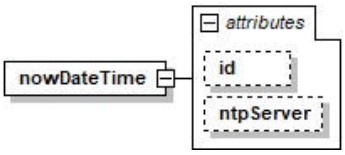
Rys. A.7–7: Standardowe funkcje dostępu do danych o podpisie



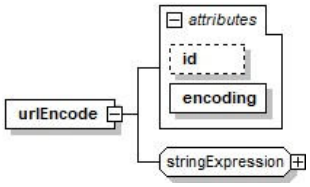
(a) Przestrzeń nazw dokumentu



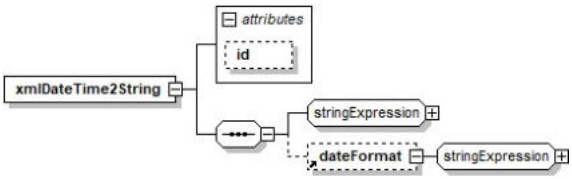
(b) Zawartość węzła XML



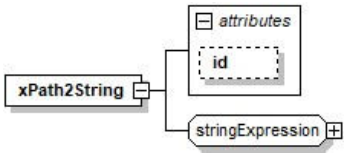
(c) Bieżący czas z serwera NTP



(d) URL encode

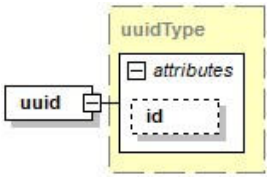


(e) Tekst daty z datetime XML

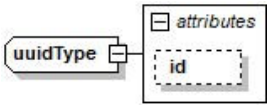


(f) Napis wskazany przez XPath

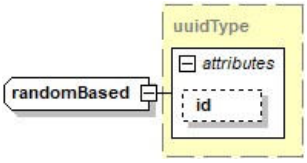
Rys. A.7–8: Inne użyteczne funkcje



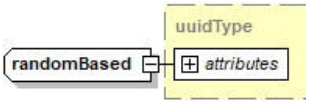
(a) Wskazanie UUID



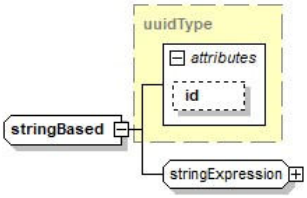
(b) Typ UUID „dziedziczenie”



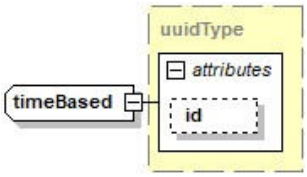
(c) Na podstawie binariów



(d) Losowe

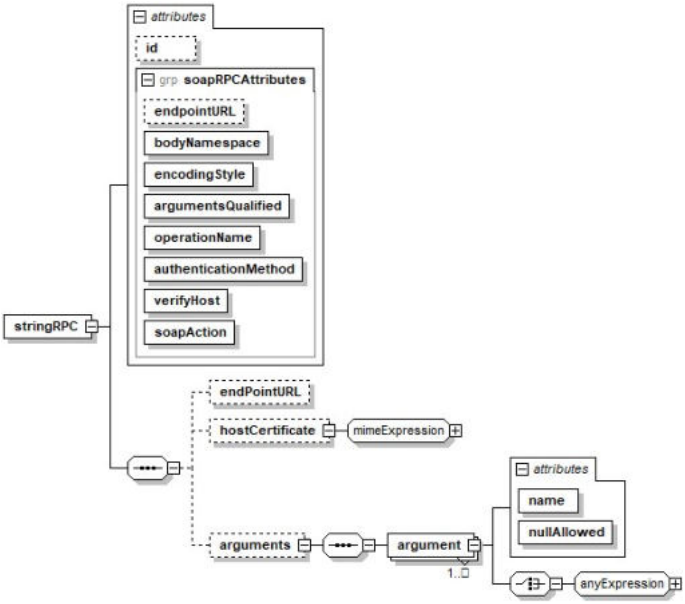


(e) Na podstawie tekstu



(f) Na podstawie czasu

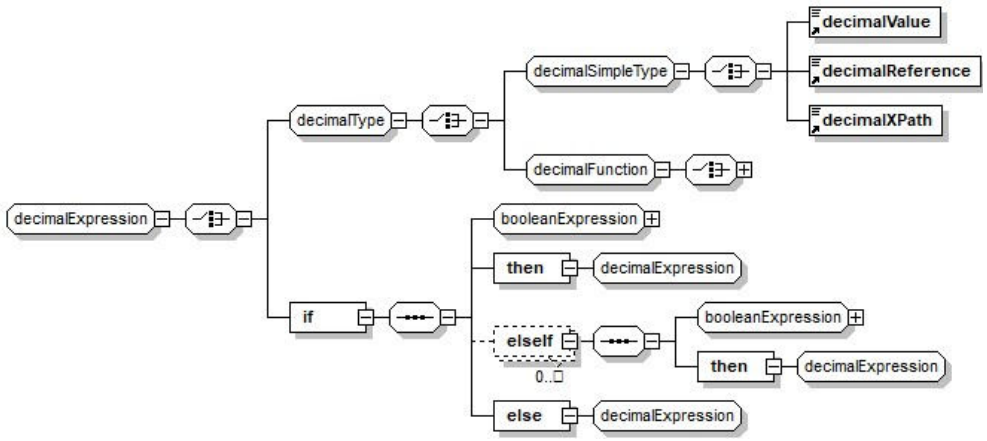
Rys. A.7–9: Globalnie unikalny identyfikator uniwersalny



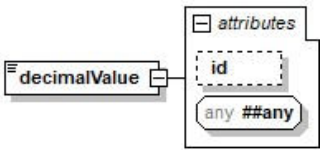
Funkcja zdeprecjonowana

Rys. A.7–10: STRING RPC *nieaktualne*

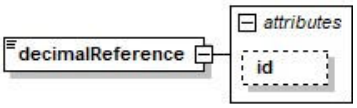
A.8. Wyrażenia dziesiętne (decimalExpression)



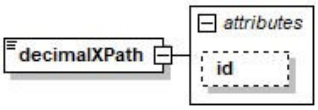
(a) Wyrażenie wartości dziesiętnej



(b) Wartość

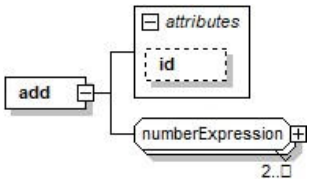


(c) Referencja

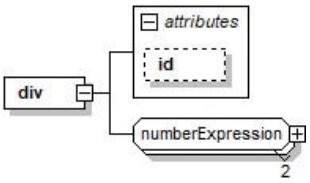


(d) Wyrażenie XPath

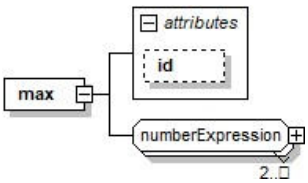
Rys. A.8–1: Składnia XML dla definiowania wyrażeń typu dziesiętnego (decimalExpression)



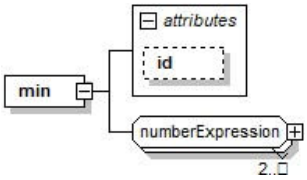
(a) Suma



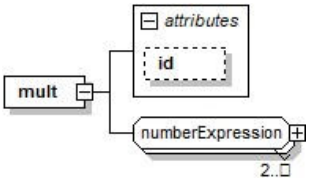
(b) Iloraz



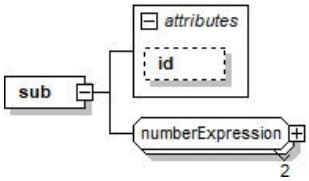
(c) Wartość maksymalna



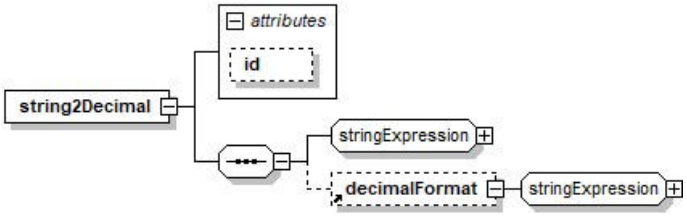
(d) Wartość minimalna



(e) Iloczyn



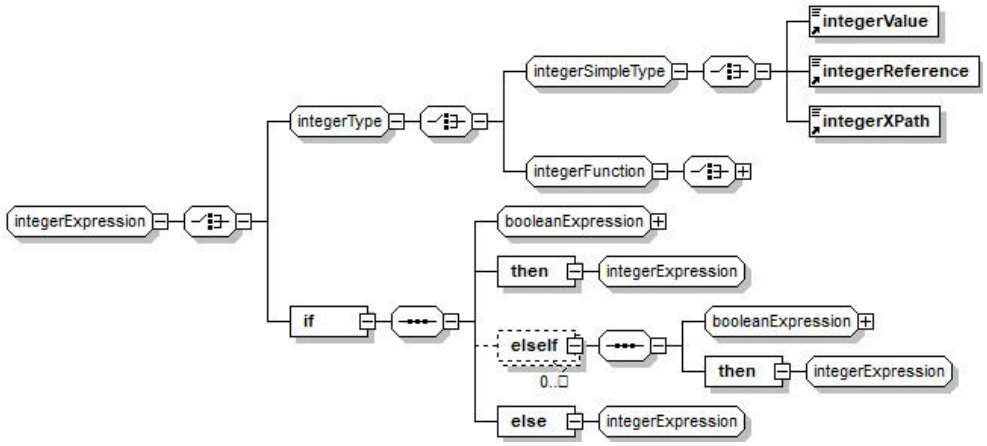
(f) Różnica



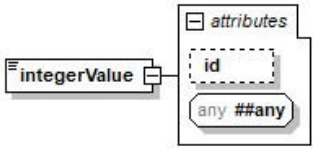
(g) Konwersja napisu do liczby dziesiętnej

Rys. A.8–2: Funkcje dziesiętne i numeryczne

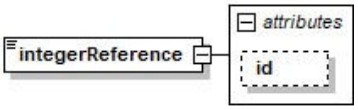
A.9. Wyrażenia całkowite (`integerExpression`) i „liczbowe” (`numberExpression`)



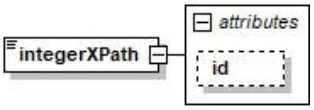
(a) Wyrażenie wartości całkowitej



(b) Wartość

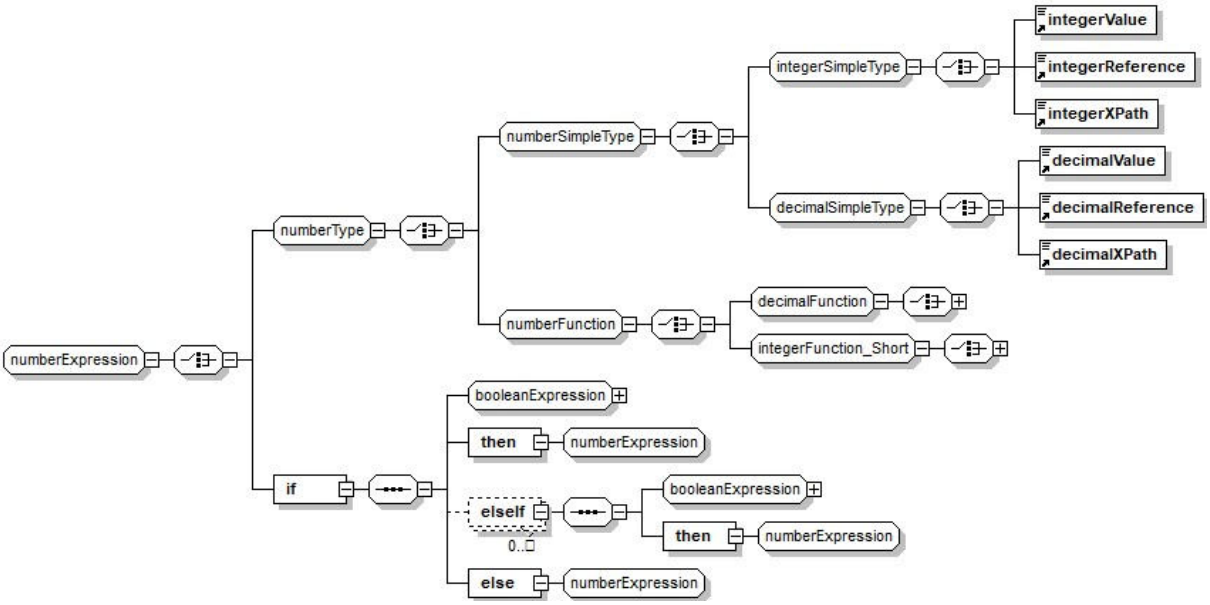


(c) Referencja

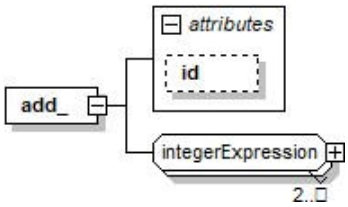


(d) Wyrażenie XPath

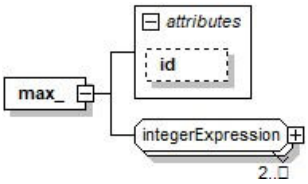
Rys. A.9–1: Składnia XML dla definiowania wyrażeń typu całkowitego (`integerExpression`)



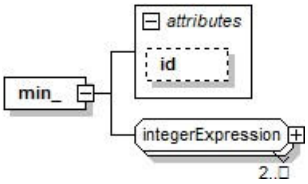
Rys. A.9–2: Wyrażenia dla wartości numerycznych (połączenie wartości dziesiętnych i całkowitych z częściowym włączeniem *funkcji całkowitych*)



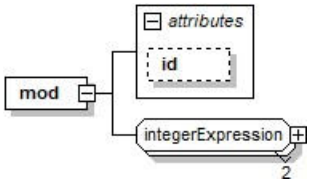
(a) Suma



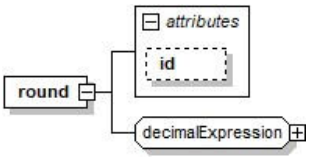
(b) Wartość maksymalna



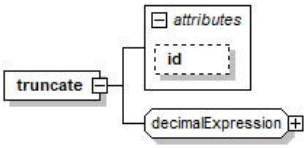
(c) Wartość minimalna



(d) Modulo

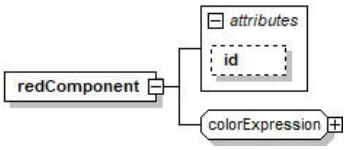


(e) Zaokrąglenie

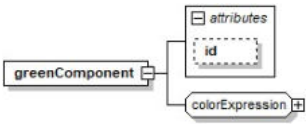


(f) Obcięcie

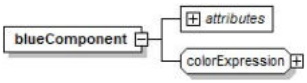
Rys. A.9–3: Funkcje całkowite – ogólne



(a) Czerwony

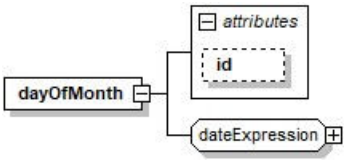


(b) Zielony

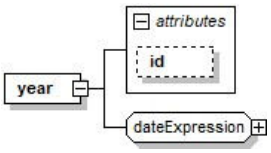


(c) Niebieski

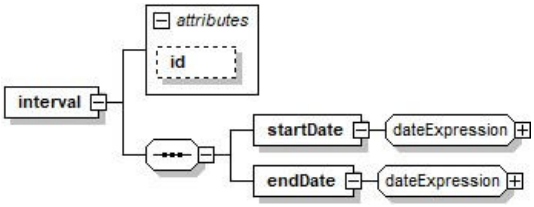
Rys. A.9–4: Funkcje całkowite – RGB



(a) Dzień miesiąca

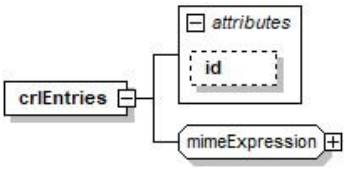


(b) Rok

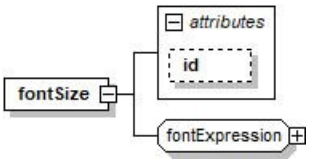


(c) Interwał między datami

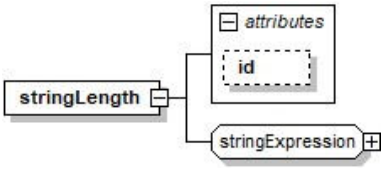
Rys. A.9–5: Funkcje całkowite dla daty



(a) Liczba wpisów na CRL



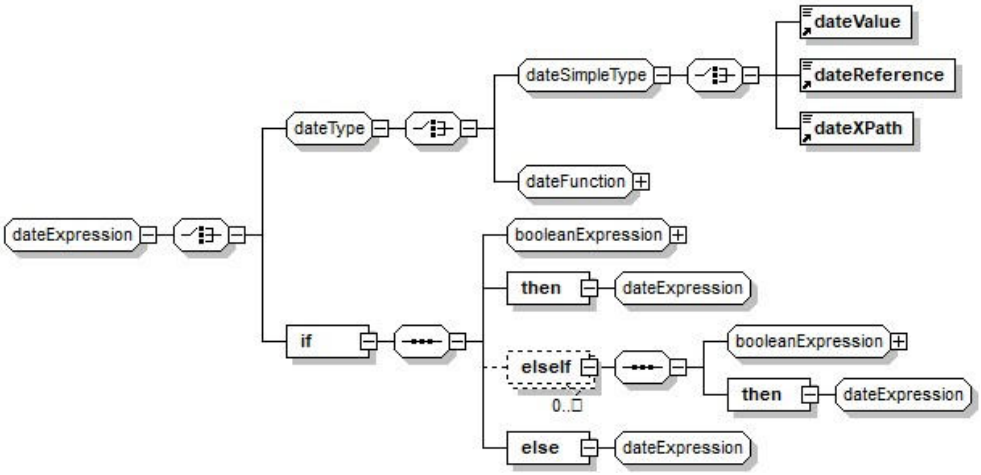
(b) Wielkość czcionki



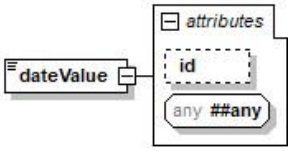
(c) Długość tekstu

Rys. A.9–6: Inne funkcje całkowite

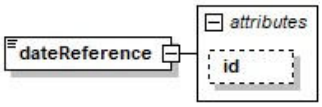
A.10. Wyrażenia daty (dateExpression)



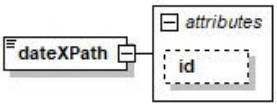
(a) Wyrażenie wskazania daty



(b) Wartość daty

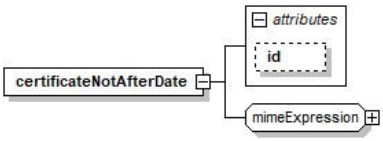


(c) Referencja

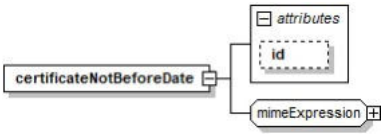


(d) Wyrażenie XPath

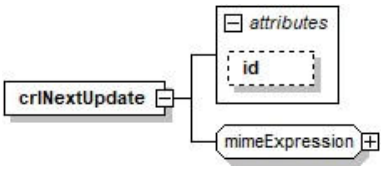
Rys. A.10–1: Składnia XML dla definiowania wyrażeń daty `<dateExpression>`



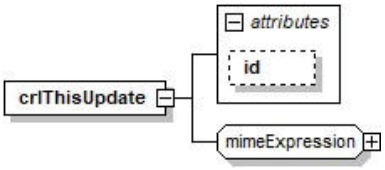
(a) Data po utracie ważności



(b) Data przed uzyskaniem ważności

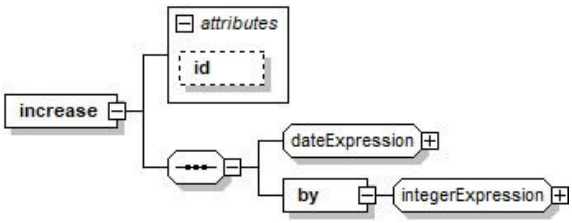


(c) Data następnego wydania CRL

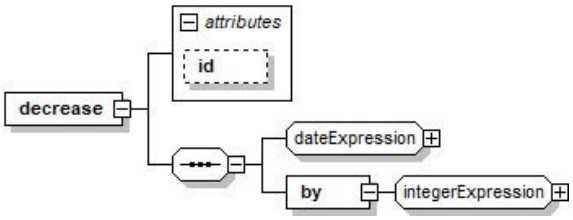


(d) Data wydania CRL

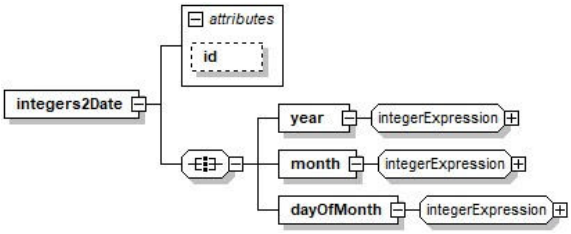
Rys. A.10–2: Funkcje obsługi daty w certyfikatach i CRL



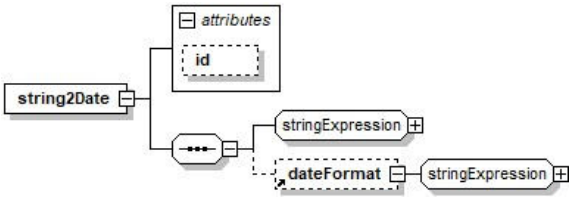
(a) Inkrementacja daty



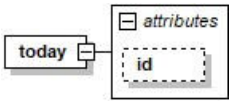
(b) Dekrementacja daty



(c) Konwercsja wartości całkowitych do daty



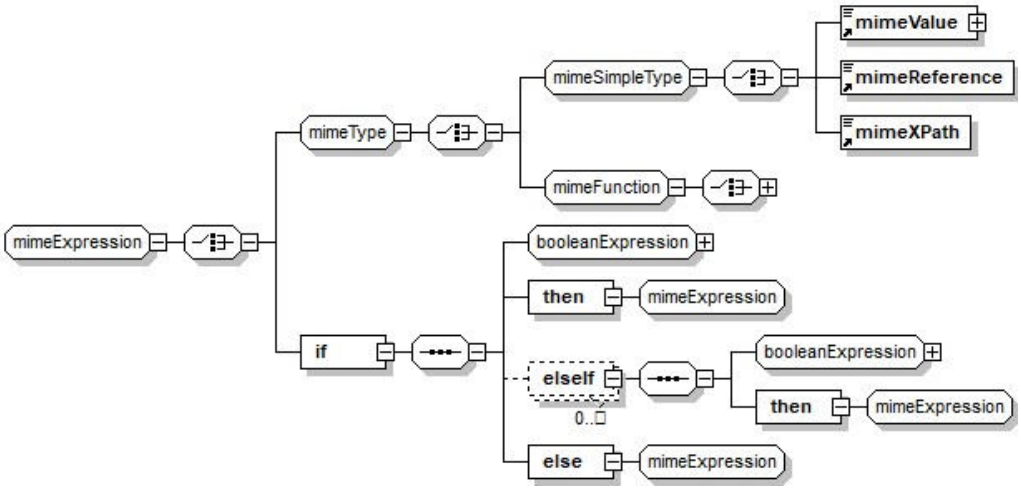
(d) Konwersja tekstu do daty



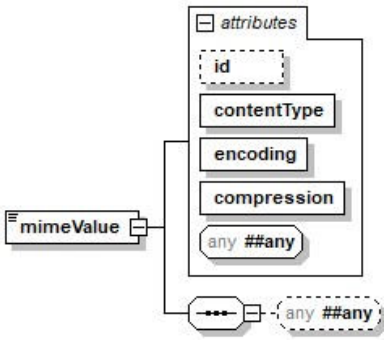
(e) Bieżąca data

Rys. A.10–3: Inne funkcje daty

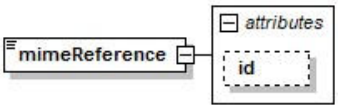
A.11. Wyrażenia binarne (`mimeExpression`)



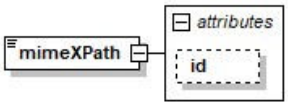
(a) Wyrażenie binarne



(b) Wartość binarna

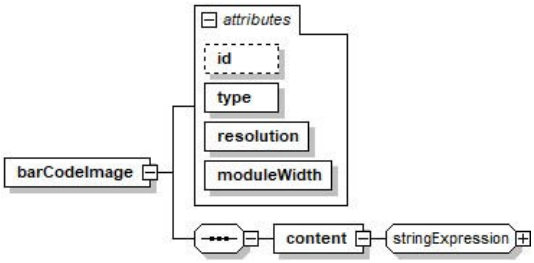


(c) Referencja

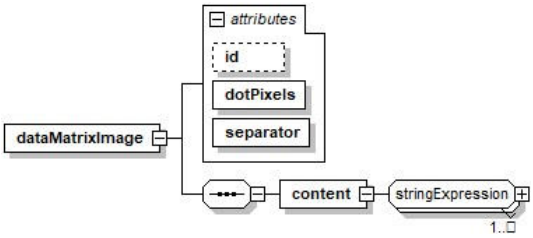


(d) Wyrażenie XPath

Rys. A.11–1: Składnia XML dla definiowania binariów `<mimeExpression>`

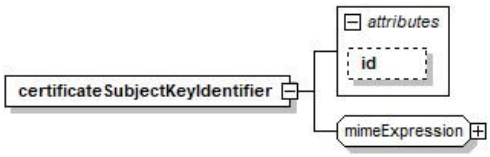


(a) Kod kreskowy

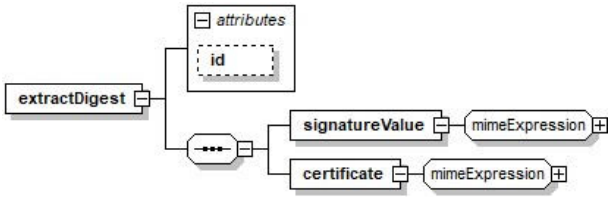


(b) Kod dwuwymiarowy

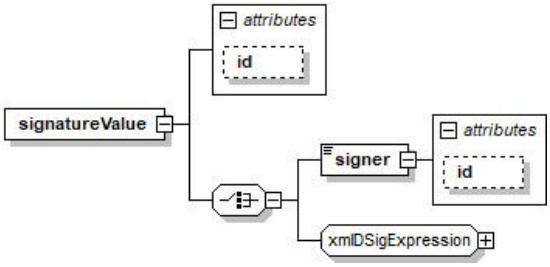
Rys. A.11–2: Kody jedno- i dwuwymiarowe



(a) Identyfikator klucza podmiotu

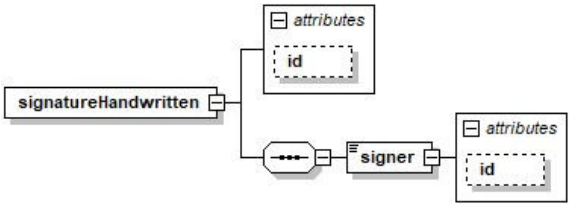


(b) Odczytanie skrótu

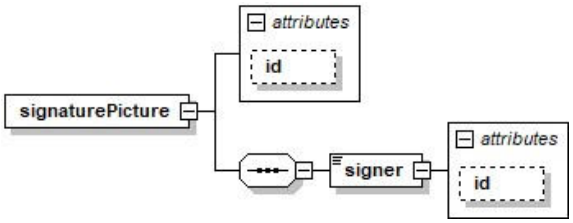


(c) Wartość podpisu

Rys. A.11–3: Funkcje binarne dla elementów PKI

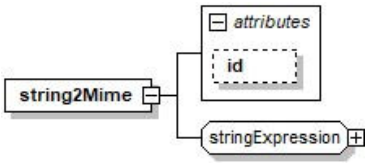


(a) Odczyt danych podpisu odręcznego

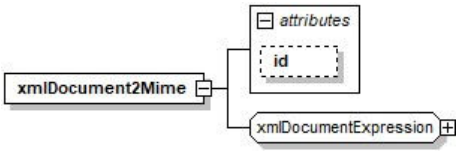


(b) Odczyt danych o obrazie

Rys. A.11–4: Obsługa danych biometrycznych w podpisie

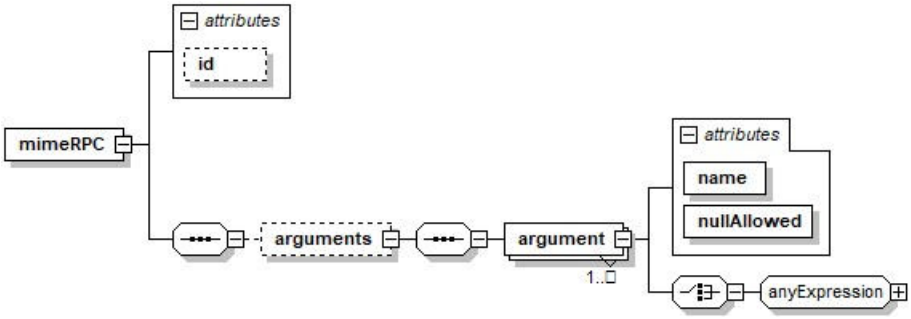


(a) Dekodowanie *Base64*



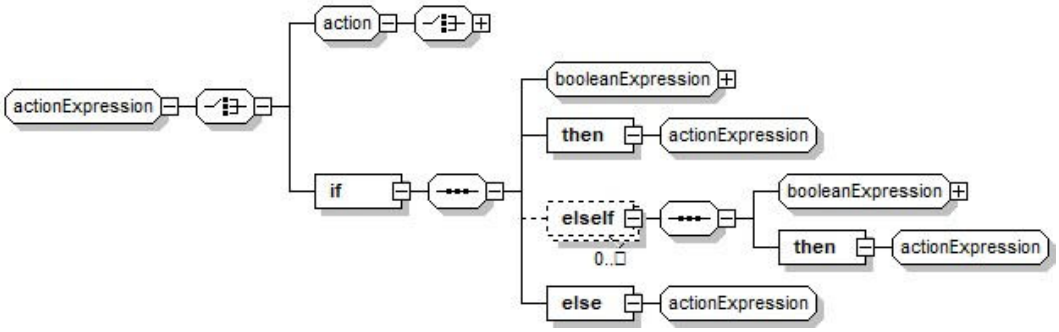
(b) Konwersja dokumentu XML do postaci binarnej (*Base64*)

Rys. A.11–5: Konwersje do binariów

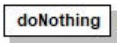


Rys. A.11–6: Konstruowanie listy argumentów w protokole SOAP-RPC (Remote Procedure Call)

A.12. Wyrażenia działań - akcji (`actionExpression`)



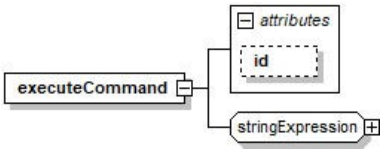
Rys. A.12–1: Składnia XML dla definiowania wyrażeń uruchamiania działań `<actionExpression>`



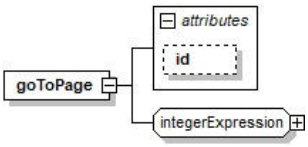
Akcja – instrukcja pusta, czyli żadne działanie nie jest wykonywane. Wprowadzone w celu uniknięcia problemów składniowych `<actionExpression>`

Rys. A.12–2: Działanie „puste”

Działnanie polega na „wywołaniu” przez aplikaję użytkownika komendy systemu operacyjnego zdefiniowanego przez wyrażenie tekstowe `<stringExpression>`



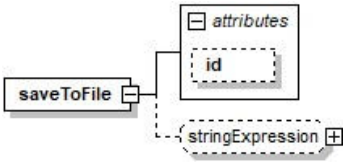
Rys. A.12–3: Wywołanie komendy Systemu Operacyjnego



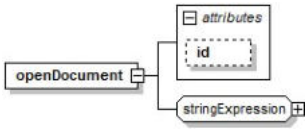
Aplikacja użytkownika wyświetla stronę formularza wskazaną przez wyrażenie całkowite `<integerExpression>`

Rys. A.12–4: Wyświetlenie strony o wskazanym numerze

Aplikacja użytkownika zapisuje formularz w lokalizacji i pod nazwą wskazaną przez wyrażenie tekstowe `<stringExpression>`



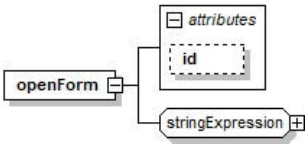
Rys. A.12–5: Zapisanie formularza na dysku



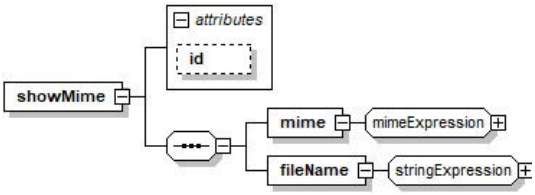
Aplikacja użytkownika otwiera w środowisku systemu operacyjnego plik wskazany przez wyrażenie tekstowe `<stringExpression>`

Rys. A.12–6: Otwieranie pliku

Aplikacja otwiera nowy formularz (zamykając bieżący) wskazany przez wyrażenie tekstowe `<stringExpression>`



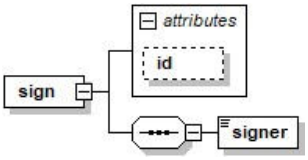
Rys. A.12–7: Otwieranie formularza



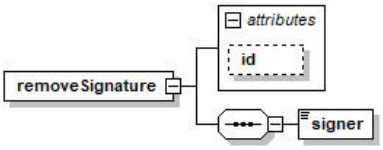
Aplikacja otwiera aplikację skojarzoną z nazwą pliku wskazaną przez wyrażenie tekstowe `<stringExpression>` w elemencie `<fileName>` z użyciem danych zdefiniowanych w wyrażeniu binarnym `(mimeExpression)` w elemencie `<mime>`

Rys. A.12–8: Otwieranie obiektu binarnego

Aplikacja generuje i umieszcza w formularzu podpis o identyfikatorze wskazanym w elemencie `<signer>`. Warunkiem wykonania operacji jest istnienie w formularzu wszystkich podpisów kontrasygnowanych



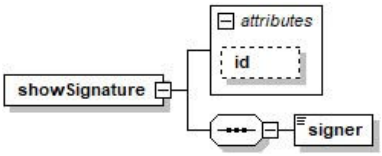
Rys. A.12–9: Wykonanie podpisu



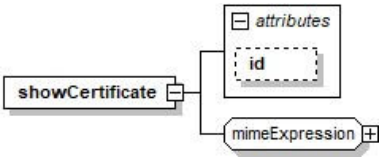
Aplikacja usuwa z formularza podpis o identyfikatorze wskazanym w elemencie `<signer>`. Warunkiem wykonania operacji jest brak istnienia wszystkich podpisów kontrasygnujących

Rys. A.12–10: Usuwanie podpisu

Aplikacja użytkownika wyświetla okienko ze szczegółowymi informacjami o podpisie wskazanym w elemencie `<signer>`

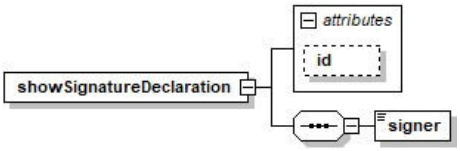


Rys. A.12–11: Wyświetlenie danych o podpisie



Aplikacja użytkownika wyświetla okienko ze szczegółowymi informacjami o certyfikacie w podpisie wskazanym w elemencie `<signer>`

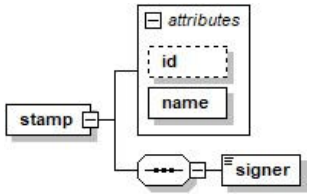
Rys. A.12–12: Wyświetlenie danych o certyfikacie



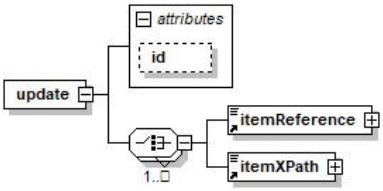
Aplikacja wyświetla „deklarację podpisującego” podpisu wskazanego w elemencie `<signer>`

Rys. A.12–13: Wyświetlenie deklaracji podpisu

Aplikacja oblicza wartość GUUI (globalnie uniwersalnego identyfikatora) dla podpisu wskazanego w elemencie `<signer>`



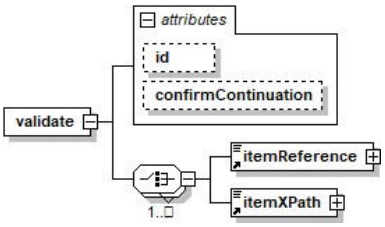
Rys. A.12–14: Wykonanie „stamp”



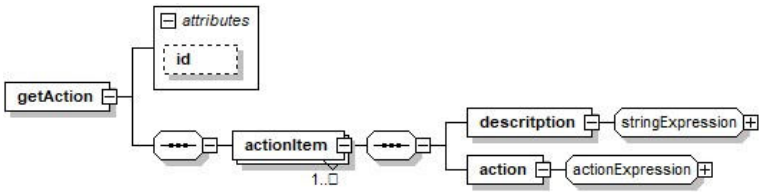
Aplikacja wykonuje wprowadzenie wartości domyślnych w elementach wskazanych przez sekwencję elementów `<itemReference>` albo sekwencję elementów `<itemXPath>`

Rys. A.12–15: Aktualizacja wartości domyślnych

Aplikacja wykonuje weryfikację reguł poprawności w elementach `<validity>` wskazanych przez sekwencję elementów `<itemReference>` albo sekwencję elementów `<itemXPath>` oraz wyświetla informację o wyniku weryfikacji – lista błędów z opisami (patrz element `<validity>`)

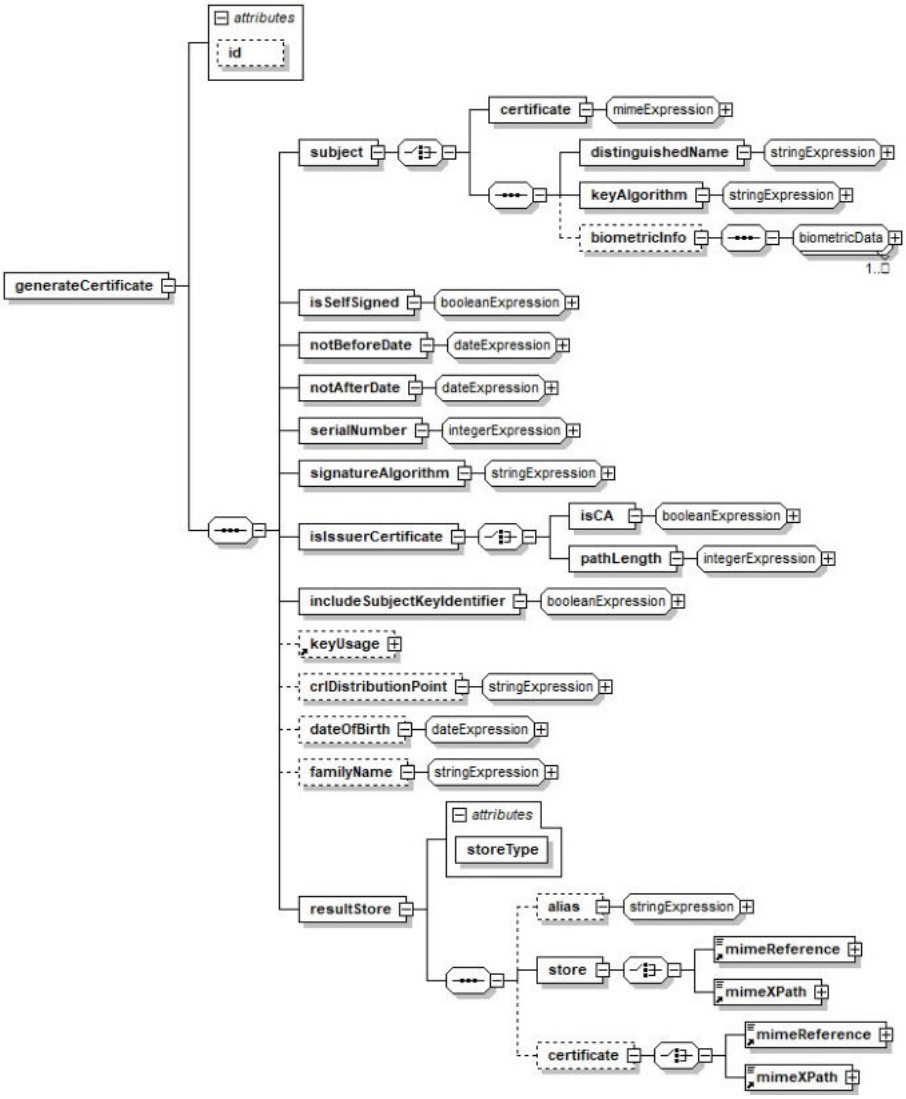


Rys. A.12–16: Walidacja komponentów

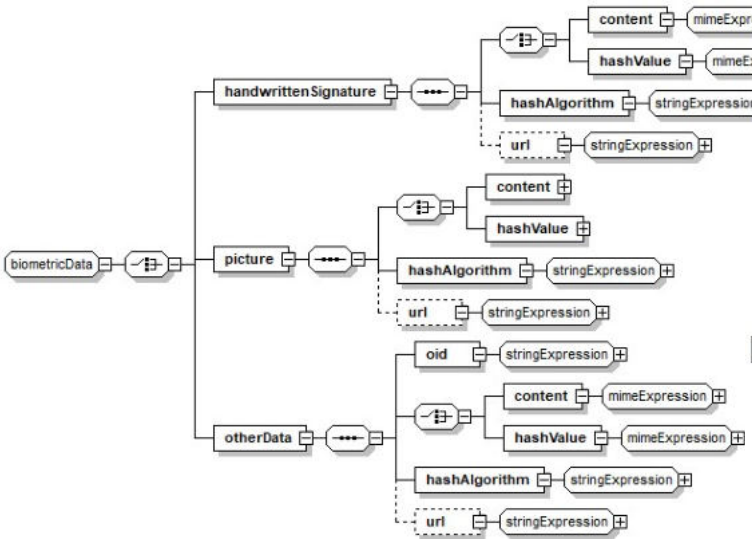


Aplikacja buduje i wyświetla listę z opisów zdefiniowanych w `(stringExpression)` w elementach `<description>` sekwencji `<actionItem>`, a następnie wykonuje akcję wybraną przez użytkownika wskazaną w `(actionExpression)` w elemencie `<action>`

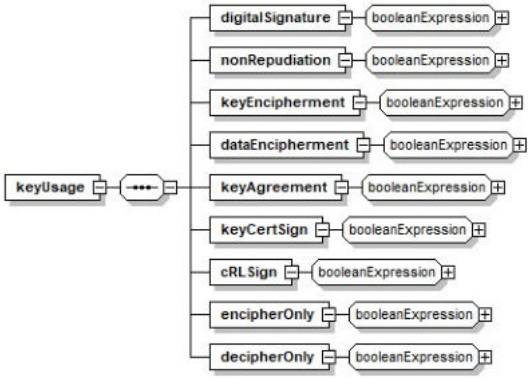
Rys. A.12–17: Wybór *akcji* i uruchomienie działania



(a) Dane do generowania certyfikatu

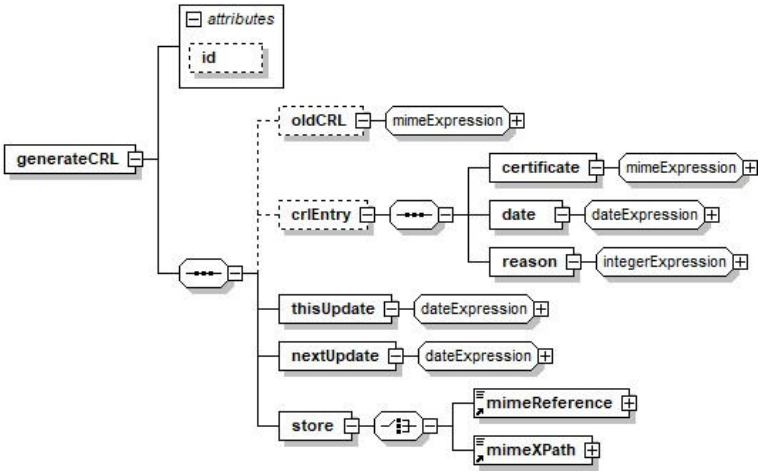


(b) Dane biometryczne

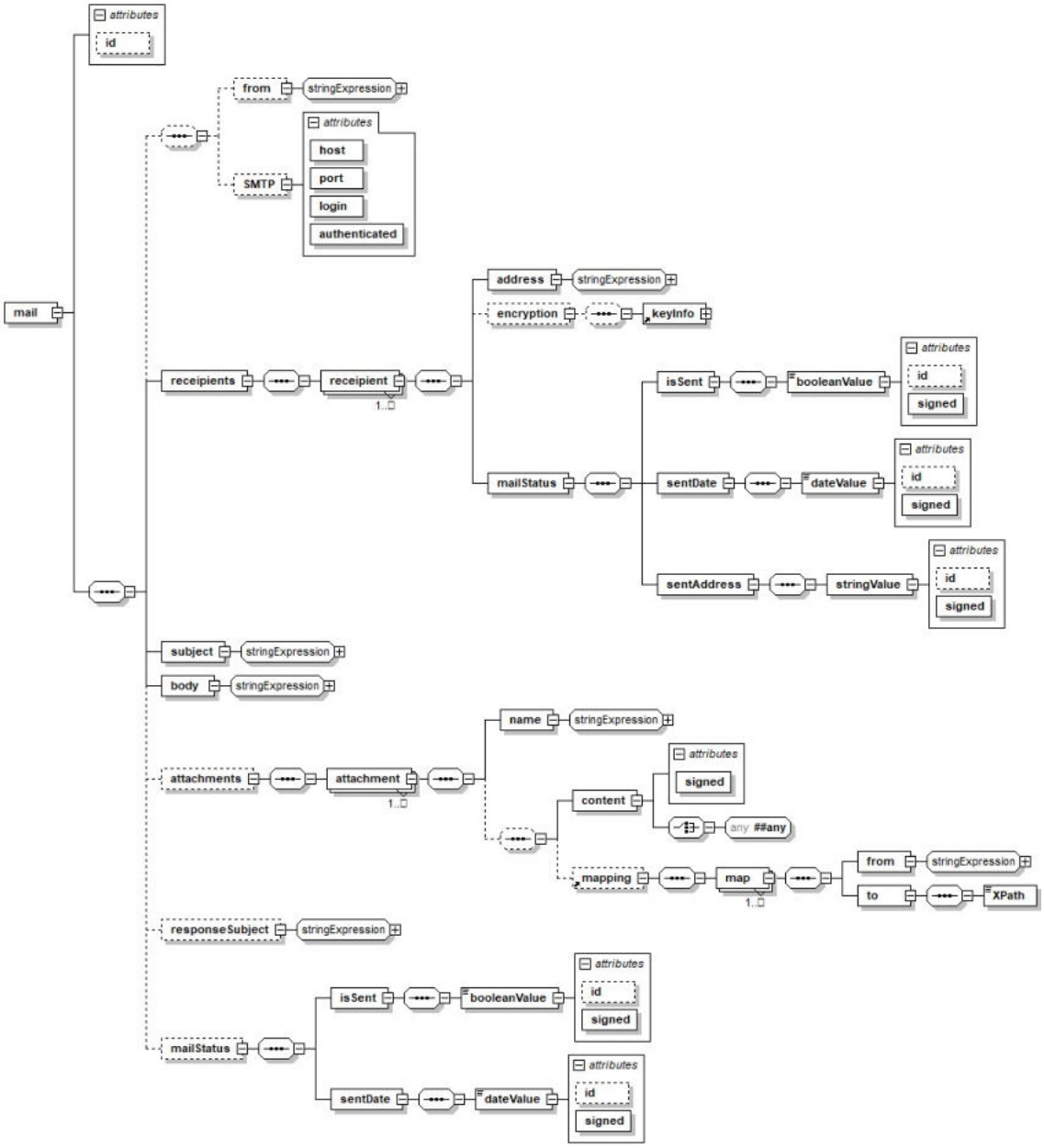


(c) Użycie klucza

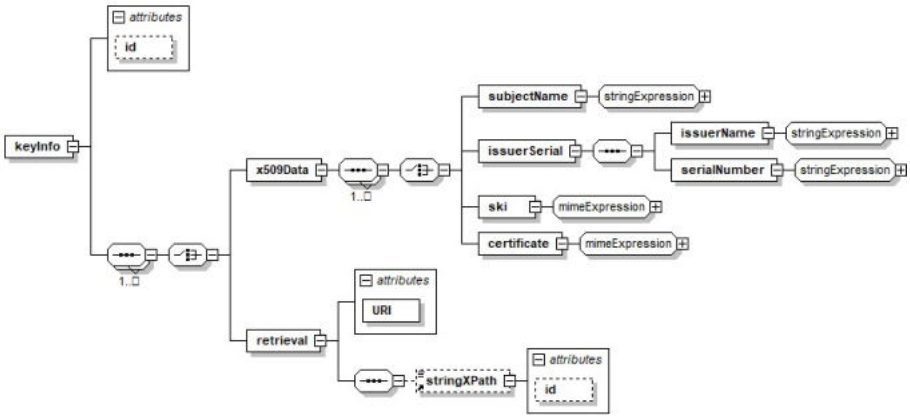
Rys. A.12–18: Generowanie certyfikatu X.509



Rys. A.12–19: Genrowanie listy certyfikatów unieważnionych *CRL*

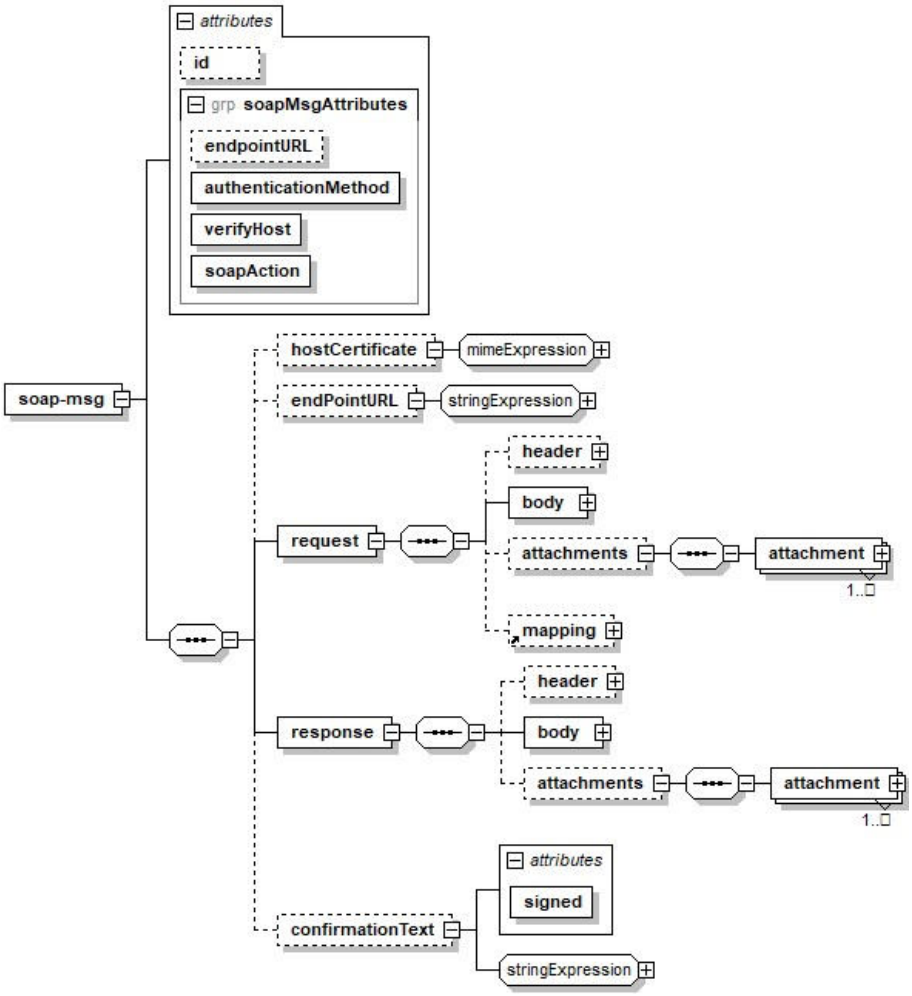


(a) Definiowanie wiadomości

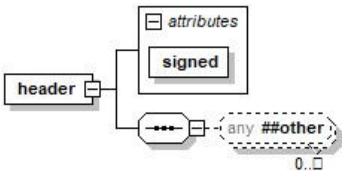


(b) Definiowanie sposobu pozyskania klucza publicznego

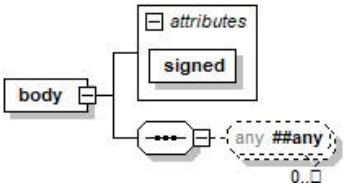
Rys. A.12–20: Składnia działania wiadomości e'mail



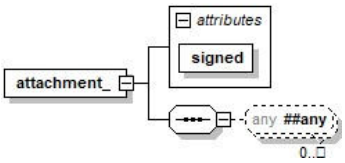
(a) Definiowanie wiadomości SOAP



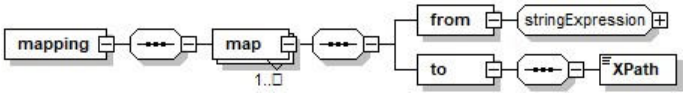
(b) SOAP header



(c) SOAP body



(d) SOAP attachment

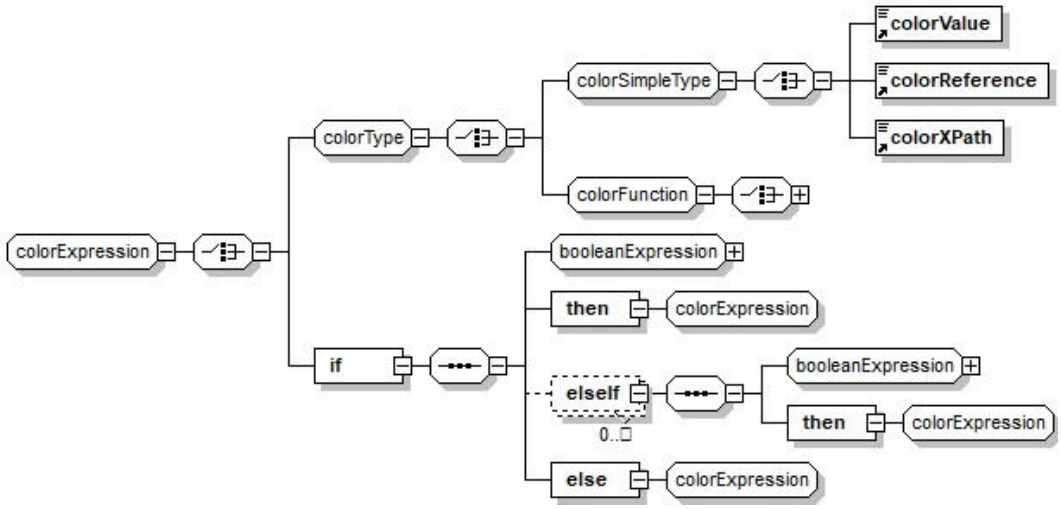


(e) SOAP-mapping

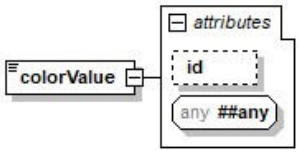
Rys. A.12–21: Konstruowanie wymiany wiadomości w protokole SOAP

A.13. Wyrażenia dla obsługi kolorów (colorExpression)

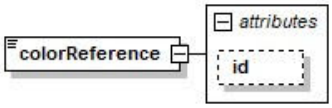
Wyrażenia udostępniające dane o kolorze (colorExpression) umożliwiają definiowanie wartości atrybutów komponentów w sposób dynamiczny w zależności od potrzeb implementowanej transakcji i stanu danych (pól) formularza.



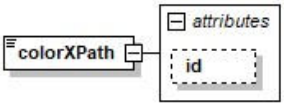
(a) Wyrażenie



(b) Wartość

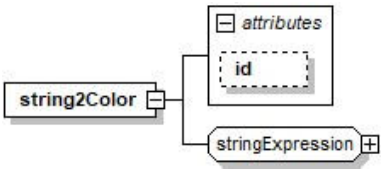


(c) Referencja

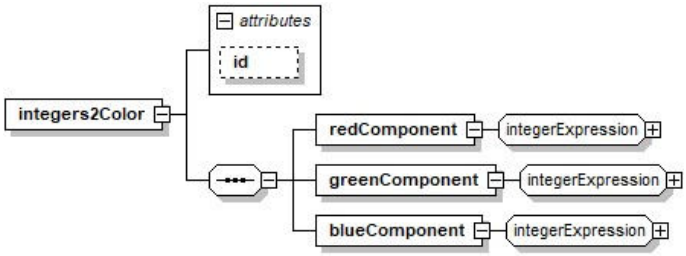


(d) Wyrażenie XPath

Rys. A.13–1: Składnia XML dla definiowania koloru (colorExpression)



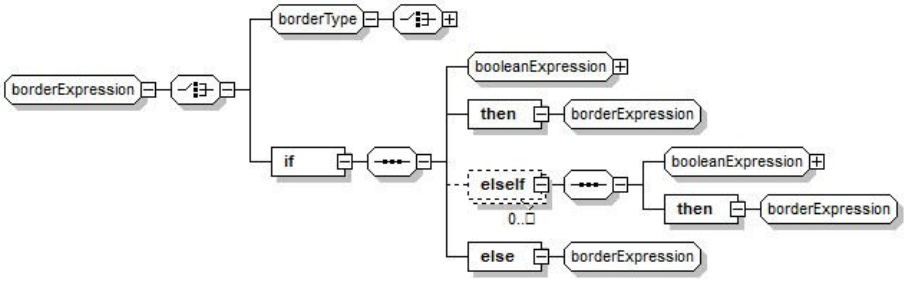
(a) Konwersja tekstu



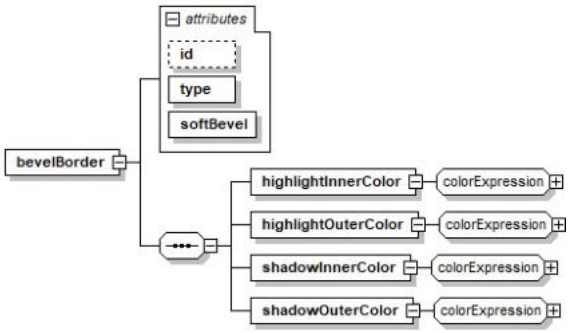
(b) Konwersja liczb całkowitych

Rys. A.13–2: Kolory – konwersja

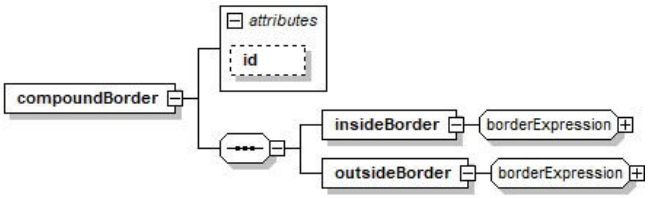
A.14. Wyrażenia dla obsługi obramowań (borderExpression)



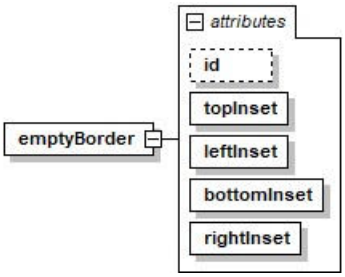
Rys. A.14–1: Składnia XML dla definiowania obramowania (borderExpression)



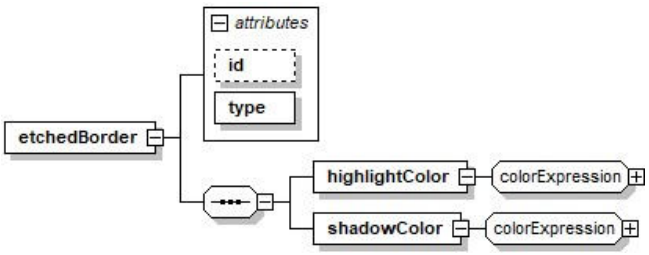
(a) Obramowanie <bevelBorder>



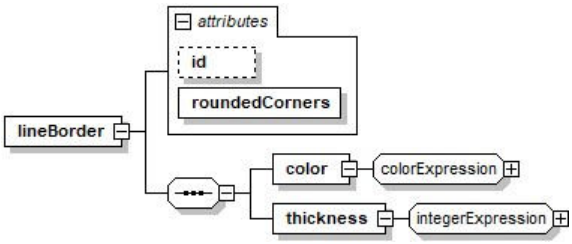
(b) Obramowanie <compoundBorder>



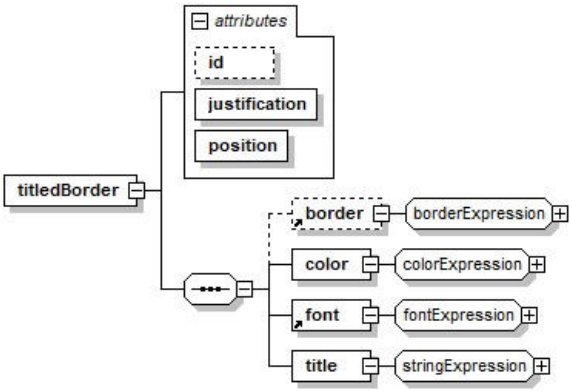
(c) Obramowanie puste <emptyBorder>



(d) Obramowanie <etchedBorder>



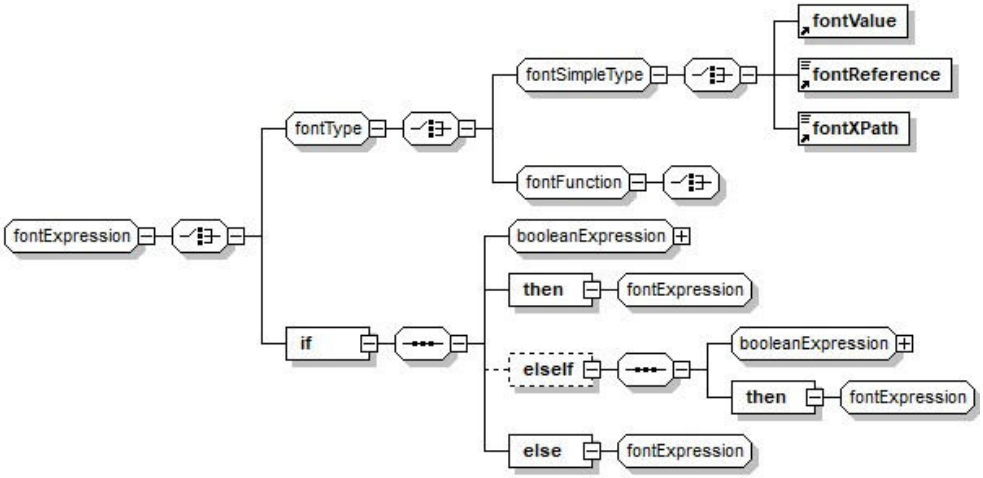
(e) Obramowanie linią <lineBorder>



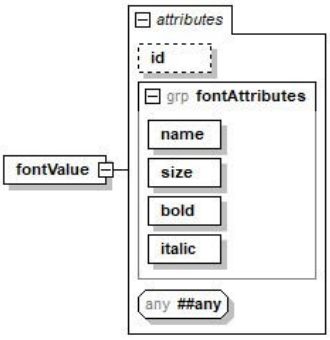
(f) Obramowanie z tytułem <titledBorder>

Rys. A.14–2: Obramowania

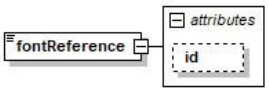
A.15. Wyrażenia dla obsługi czcionki `<fontExpression>` i komponentów `<itemExpression>`



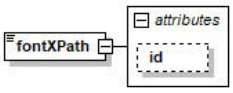
(a) Wyrażenie



(b) Wartość

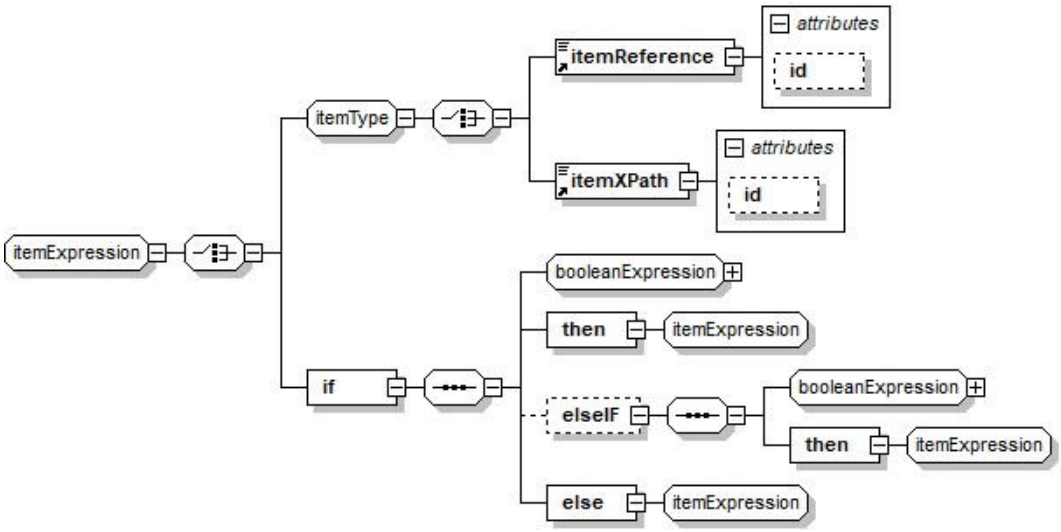


(c) Referencja



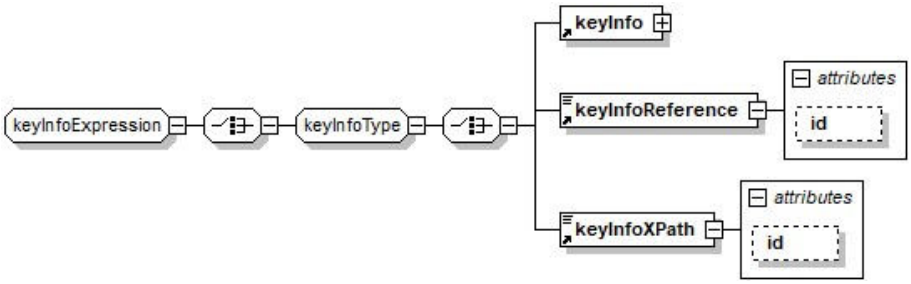
(d) Wyrażenie XPath

Rys. A.15–1: Składnia XML dla definiowania czcionki `<fontExpression>`

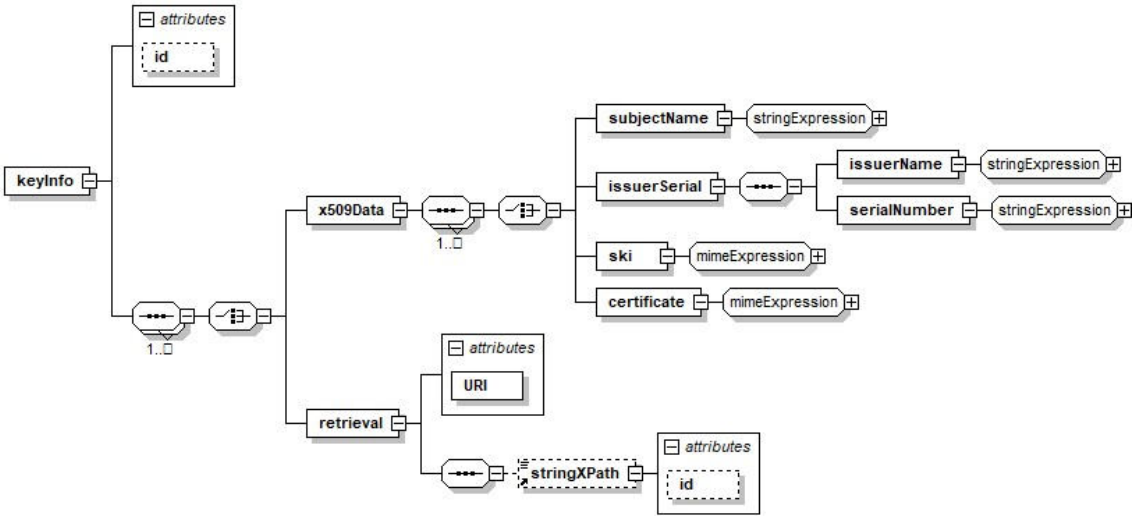


Rys. A.15–2: Wyrażenie służące do wskazania komponentu `<itemExpression>`

A.16. Wyrażenia dla dostępu do klucza (keyInfoExpression) i podpisu XML (xmlDSigExpression)

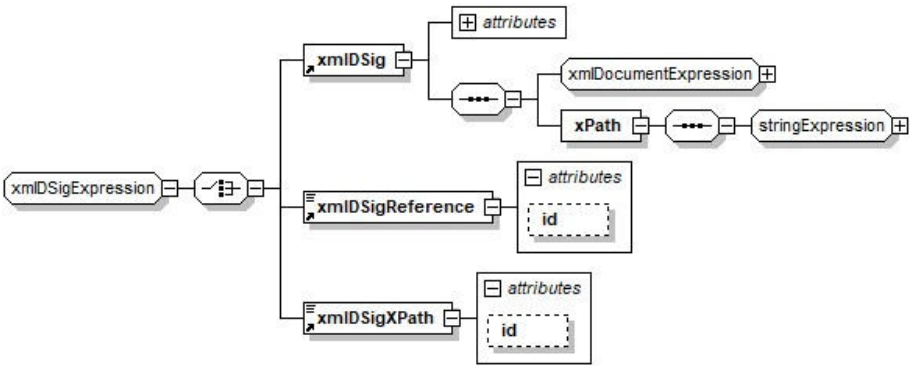


(a) Wyrażenie



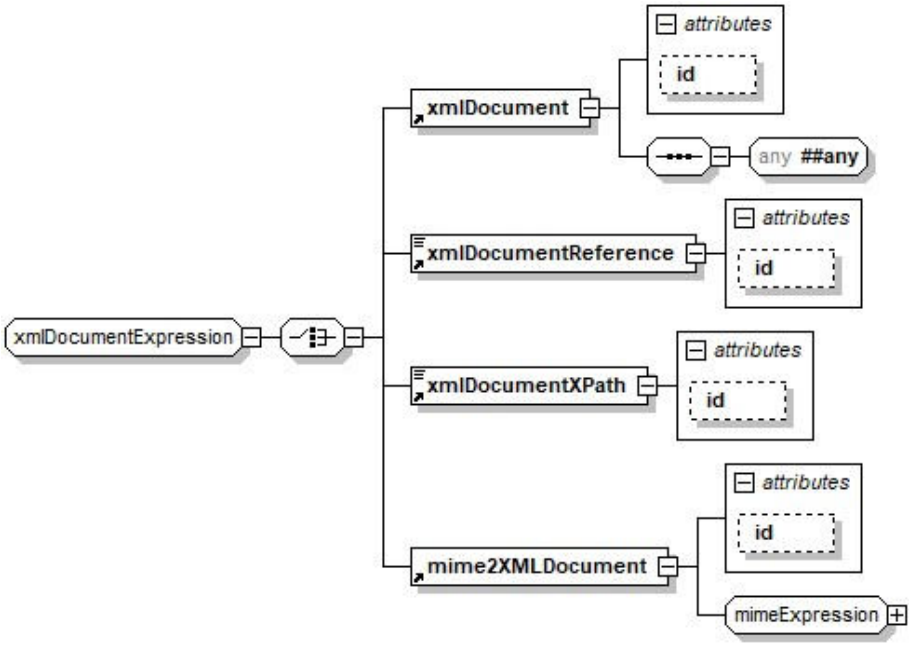
(b) Definiowanie sposobu pozyskania klucza publicznego

Rys. A.16–1: Wyrażenie wskazujące na informacje o kluczu (keyInfoExpressison)



Rys. A.16–2: Wyrażenie do obsługi podpisów XMLdSig (xmlDSigExpression)

A.17. Wyrażenia dla dostępu do dokumentu XML *(xmlDocumentExpression)*



Rys. A.17–1: Wyrażenie do obsługi fragmentów dokumentów XML (`xmlDocumentExpression`)