WYMAGANIA TECHNICZNE, JAKIE SPEŁNIAJĄ SYSTEMY INFORMACYJNE WSPÓŁPRACUJĄCE Z CENTRALNYM SYSTEMEM INFORMACJI RYNKU ENERGII

(Wstępny projekt zmian Załącznika nr 5.
do IRiESP-OIRE)

Nota prawna

Metryka dokumentu:

Nazwa dokumentu	WYMAGANIA TECHNICZNE, JAKIE SPEŁNIAJĄ SYSTEMY INFORMACYJNE WSPÓŁPRACUJĄCE Z CENTRALNYM SYSTEMEM INFORMACJI RYNKU ENERGII Standard techniczny wymiany informacji z wykorzystaniem protokołu AS4
Nazwa pliku	OIRE_2023-10-12_WymaganiaTechnicznePSE-AS4.docx
Wersja dokumentu	Z dnia 12 października 2023
Data opracowania	2023-10-12
Autor dokumentu	Projekt OIRE – CGI oraz PSE
Osoba weryfikująca	Projekt OIRE – Zespół IT (QC)
Zawartość dokumentu (krótki opis)	Wymagania techniczne dla systemów teleinformatycznych współpracujących z CSIRE wraz ze specyfikacją techniczną protokołu AS4.
Etap / Proces	Strumień 3: Budowa, testowanie i uruchomienie CSIRE/S3.4 Publikacja wymagań technicznych, w tym w zakresie oprogramowania, jakie muszą spełniać systemy informacyjne współpracujące z CSIRE.

Historia zmian dokumentu:

L.p.	Wersja	Opis zmiany	Data przekazania	Opracowujący zmianę	Firma
1.	Z dnia 31 maja 2023 r.	Publikacja na potrzeby wstępnych konsultacji projektu zmian	2023-05-31	Projekt OIRE – CGI oraz PSE	PSE S.A.
2.	Z dnia 12 października 2023 r.	Dodanie przykładu wywołań operacji. aktualizacja dokumentu po wstępnych konsultacjach	2023-10-12	Projekt OIRE – CGI oraz PSE	PSE S.A.
3.	Z dnia 12 października 2023 r.	Dodano wymagania dotyczące środowisk systemów współpracujących z CSIRE	2023-10-12	Projekt OIRE – CGI oraz PSE	PSE S.A.
4.	Z dnia 12 października 2023 r.	Dodanie konfiguracji kolejek wyjściowych.	2023-10-12	Projekt OIRE – CGI oraz PSE	PSE S.A.
5.	Z dnia 12 października 2023 r.	Dodanie rozdziału z WSDL	2023-10-12	Projekt OIRE – CGI oraz PSE	PSE S.A.
6.	Z dnia 12 października 2023 r.	Publikacja na potrzeby wstępnych konsultacji projektu zmian	2023-10-12	Projekt OIRE – CGI oraz PSE	PSE S.A.

SPIS TREŚCI

1. WYKAZ DEFINICJI I SKRÓTÓW

Niniejszy rozdział zawiera wykaz definicji pojęć oraz wykaz skrótów stosowanych w niniejszym dokumencie, a także spis dokumentów powiązanych z niniejszym dokumentem.

1.1. Wykaz definicji

Definicja	Objaśnienie
Centralny System Informacji Rynku Energii	System informacyjny służący do przetwarzania informacji rynku energii na potrzeby realizacji procesów rynku energii elektrycznej oraz wymiany informacji pomiędzy Użytkownikami systemu elektroenergetycznego.
Kod EIC	Kod służący do identyfikacji podmiotów na europejskim rynku energii. Kody nadawane są przez Centralne Biuro Kodów EIC (ENTSO-E) i przez Lokalne Biura Kodów EIC w poszczególnych krajach. W Polsce Lokalne Biura Kodów EIC prowadzone są przez Polskie Sieci Elektroenergetyczne S.A. (numer identyfikacyjny 19) oraz Operatora Gazociągów Przesyłowych GAZ-SYSTEM S.A. (numer identyfikacyjny 53)
Kontrahent	Użytkownik profesjonalny lub Użytkownik uprawniony będący stroną Umowy CSIRE, bądź podmiot ubiegający się o jej zawarcie.
Message Consumer	Aplikacja biznesowa lub usługa pośrednicząca odpowiadająca w procesie za odbiór komunikatu.
Message Producer	Aplikacja biznesowa lub usługa pośrednicząca odpowiadająca w procesie za przygotowanie komunikatu.
Message Service Handler	Usługa umożliwiająca wymianę wiadomości pomiędzy partnerami biznesowymi
Nadawca fizyczny	Podmiot udostępniający Kontrahentowi system informacyjny oraz zapewniający jego obsługę w celu realizacji przez Kontrahenta procesów rynku energii lub wymiany informacji rynku energii.
Operator informacji rynku energii	Podmiot odpowiedzialny za zarządzanie i administrowanie Centralnym systemem informacji rynku energii oraz przetwarzanie zgromadzonych w nim informacji na potrzeby realizacji procesów rynku energii.
Organizacja	Reprezentacja podmiotu rynku energii w systemie CSIRE
Portal Użytkownika profesjonalnego	Portal dedykowany dla Użytkowników profesjonalnych oraz Użytkowników uprawnionych. Umożliwia on realizację procesów rynku energii i wymianę informacji rynku energii za pośrednictwem CSIRE.
Protokół AS4 (Application Statement 4)	Standard opisujący bezpieczne i niezawodne przesyłanie komunikatów przez publiczną sieć Internet. Protokół ten bazuje na powszechnie znanych i sprawdzonych rozwiązaniach, takich jak protokoły HTTP, TLS, SOAP oraz usługach sieciowych (web service). Reprezentuje otwarty standard wymiany danych typu B2B opisany w specyfikacji OASIS ebMS 3.0.
Receiving MSH	Usługa pełniąca rolę punktu docelowego w wymianie wiadomości pomiędzy partnerami biznesowymi.
Sending MSH	Usługa pełniąca rolę punktu inicjującego wymianę wiadomości w imieniu partnera biznesowego inicjującego wymianę komunikatów.

Definicja	Objaśnienie
Użytkownik uprawniony	Podmiot realizujący wymianę informacji rynku energii za pośrednictwem CSIRE, niebędący Użytkownikiem profesjonalnym lub Użytkownik profesjonalny działający na podstawie upoważnienia Użytkownika KSE.
Użytkownik profesjonalny	Podmiot realizujący procesy rynku energii i wymianę informacji rynku energii za pośrednictwem CSIRE.
WS-Security	Standard OASIS określający mechanizm zabezpiecznia usług Web Service.

Tabela 1. Wykaz definicji

1.2. Lista skrótów

Skrót	Rozwinięcie
AS4	Protokół AS4 (Application Statement 4)
A2A	*Administration-to-Administration*
B2A	*Business-to-Administration*
B2B	*Business-to-Business*
CSIRE	Centralny System Informacji Rynku Energii
CSWI	Centralny System Wymiany Informacji
DNS	*Domain Name System*
ENTSOG	*European Network of Transmission System Operators for Gas*
FIFO	*First In First Out*
IRiESP – OIRE	Instrukcja Ruchu i Eksploatacji Sieci Przesyłowej część „Sposób funkcjonowania Centralnego systemu informacji rynku energii oraz współpracy Operatora systemu przesyłowego elektroenergetycznego, działającego jako Operator informacji rynku energii, z Użytkownikami systemu elektroenergetycznego i innymi podmiotami zobowiązanymi lub uprawionymi do korzystania z Centralnego systemu informacji rynku energii”
JSON	*JavaScript Object Notation*
MEP	*Message Exchange Patterns*
MPC	*Message Partition Channels*
MSH	*Message Service Handler*
OIRE	Operator informacji rynku energii
OSD	Operator systemu dystrybucyjnego
PTPiREE	Polskie Towarzystwo Przesyłu i Rozdziału Energii Elektrycznej
SE	Sprzedawca
SEu	Sprzedawca z urzędu
SEr	Sprzedawca rezerwowy
SOAP	*Simple Object Access Protocol*
SWI	Standardy Wymiany Informacji
TLS	*Transport Layer Security*
TSKB	Techniczne Standardy Komunikacji Biznesowej
XML	*Extensible Markup Language*

Skrót	Rozwinięcie
XSD	*XML Schema Definition*
WSS	*Web Services Security (WS-Security)*

Tabela 2. Lista skrótów

1.3. Dokumenty powiązane

Lp.	Nazwa dokumentu powiązanego	Wersja dokumentu	Używany skrót nazwy
1.	Instrukcja Ruchu i Eksploatacji Sieci Przesyłowej – Sposób funkcjonowania Centralnego systemu informacji rynku energii oraz współpracy Operatora systemu przesyłowego elektroenergetycznego, działającego jako Operator informacji rynku energii, z Użytkownikami systemu elektroenergetycznego i innymi podmiotami zobowiązanymi lub uprawionymi do korzystania z Centralnego systemu informacji rynku energii.	Z dnia 6 kwietnia 2023 r.	IRiESP-OIRE
2.	Techniczne standardy komunikacji biznesowej	Z dnia 4 kwietnia 2023 r.	TSKB

Tabela 3. Dokumenty powiązane

2. WSTĘP

Protokół AS4 [AS4-Profile] określa otwarty standard bezpiecznego oraz niezawodnego przesyłania komunikatów poprzez sieć Internet z wykorzystaniem usługi sieciowych. Wykorzystuje powszechnie znane rozwiązania takie jak SOAP, MIME oraz WS-Security. Zazwyczaj jest stosowany w modelach B2B, B2A oraz A2A.

Dzięki możliwości przesyłania różnych typów komunikatów takich jak pliki: binarne, XML lub JSON, zapewnia wysoki poziom elastyczności.

Powyższe cechy oraz istnienie zarówno komercyjnych jak i otwartych implementacji protokołu AS4 spowodowały, iż został on przyjęty przez Komisję Europejską do budowy komponentu eDelivery w ramach Digital Europe Programme.

Ponadto jest on wykorzystywany także przez podmioty skupione w ENTSOG w ramach rozwoju wewnątrzwspólnotowego rynku gazu.

AS4 został przyjęty przez PTPiREE jako standard wymiany komunikatów w projekcie budowy CSWI, a OIRE zaakceptował ten standard dla systemu CSIRE.

3. CEL

Niniejszy dokument opisuje wykorzystanie protokołu AS4 do wymiany danych z CSIRE. Przedstawione informacje będą służyć do przygotowania konfiguracji systemów informacyjnych Użytkowników profesjonalnych, Użytkowników uprawnionych oraz Nadawców fizycznych do współdziałania z OIRE w modelu B2B.

4. ZAKRES

4.1. Podmioty

Konfiguracja opisana w niniejszym standardzie dotyczy systemów informacyjnych Użytkowników profesjonalnych, Użytkowników uprawnionych oraz Nadawców fizycznych wymieniających dane z CSIRE. Kontrahenci korzystający z Nadawców fizycznych będą wykorzystywać ich kanały komunikacyjne oraz będą identyfikowani na podstawie zawartości komunikatów.

4.2. Kompozycja dokumentu

Standard techniczny wymiany informacji z wykorzystaniem protokołu AS4 [PSE-AS4] opisany w niniejszym materiale zawiera informacje o zmianach lub wybranych opcjach w stosunku do norm pochodzących z zewnętrznych dokumentów.

Bazuje on na “AS4 Profile of ebMS 3.0 Version 1.0 OASIS Standard” [AS4-Profile], który wykorzystuje między innymi standard “OASIS ebXML Messaging Services Version 3.0: Part 1, Core Features OASIS Standard” [ebMS3CORE]. Ponadto występują odwołania do dokumentów opracowanych w celu implementacji Protokołu AS4 w konkretnych zastosowaniach tj. „ENTSOG AS4 Profile 3.6” [EG-AS4-Profile] oraz “AS4 Interoperability Profile for Four-Corner Networks Version 1.0 Committee Specification 01” [BDX-AS4-v1.0].

4.3. Język

W wypadku części informacji pochodzących w zewnętrznych dokumentów, pozostawiono ich oryginalną wersję językową.

5. KOMUNIKACJA

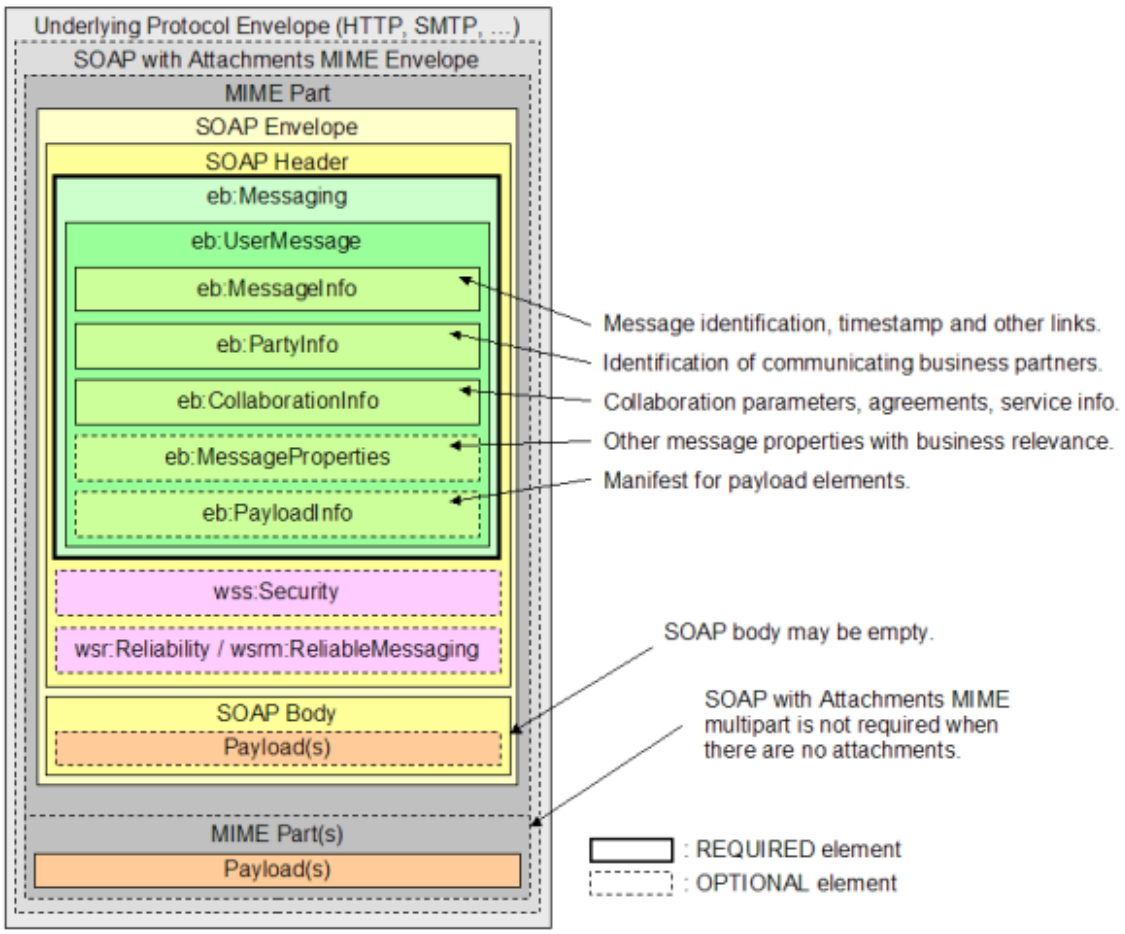
5.1. Struktura wiadomości

Standard wymiany komunikatów na potrzeby wymiany danych z CSIRE [PSE-AS4] bazuje na wymianie komunikatów biznesowych poprzez wiadomości AS4.

Wiadomości AS4 powinny być budowane zgodnie z opisywanym przez OASIS standardem ebMS 3.0 [ebMS3CORE].

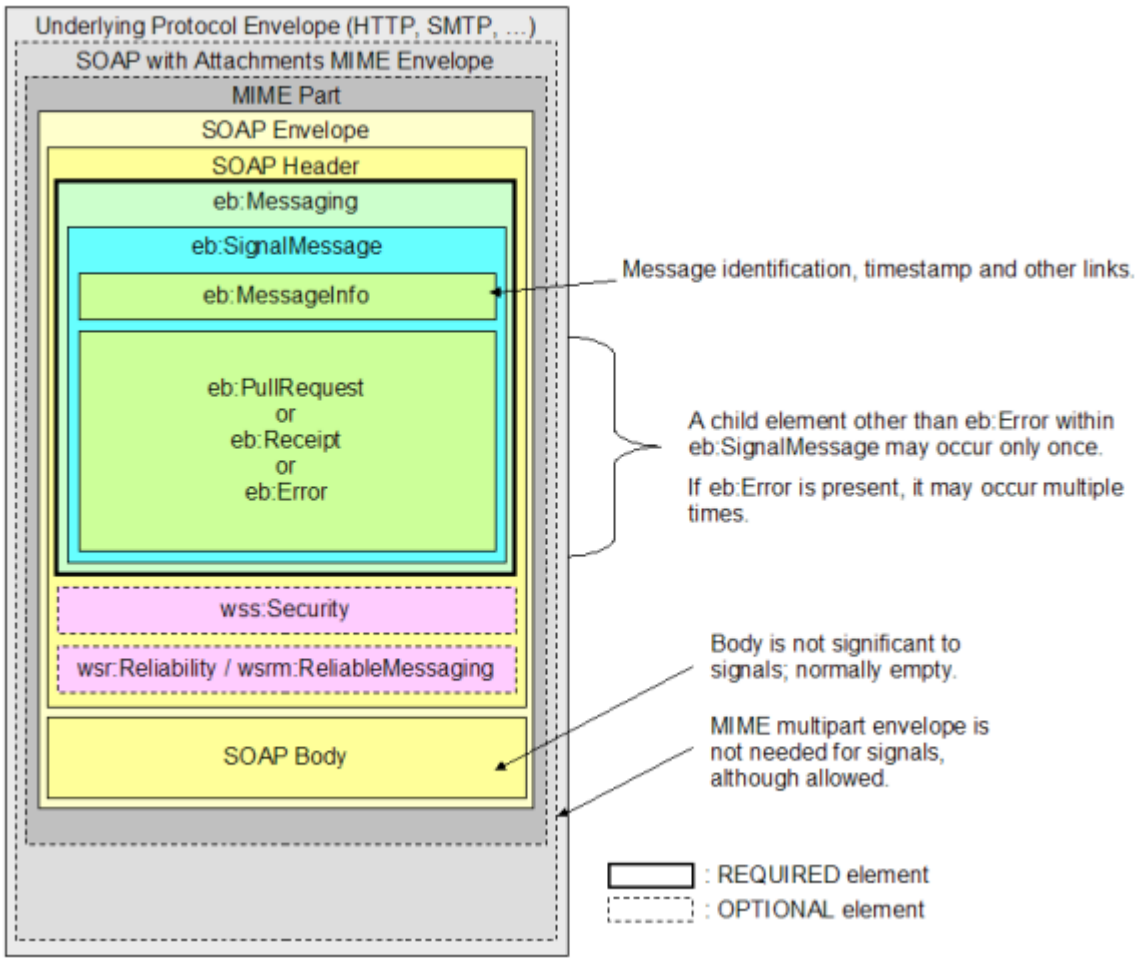
Struktura dwóch podstawowych wiadomości przekazywanych podczas transmisji pomiędzy MSH uczestniczącymi w wymianie danych, znajduje się na poniższych rysunkach.

Struktura wiadomości biznesowej



Rysunek 1 Struktura wiadomości (User Message Structure, [ebMS3CORE])

Struktura wiadomości sygnałowej



Rysunek 2 Struktura wiadomości sygnałowej (Signal Message Structure, [ebMS3CORE])

5.2. Podstawowe informacje dotyczące wymiany danych.

Implementacja protokołu AS4 zakłada centralną rolę CSIRE w komunikacji między stronami rynku i wymusza inicjacje komunikacji z systemów zewnętrznych zarówno dla wiadomości wysyłanych do systemu jak i wiadomości pobieranych z systemu CSIRE.

System CSIRE będzie zarówno producentem (*Message Producer*) jak i konsumentem (*Message Consumer*) wiadomości, przy czym sposób ich przekazania będzie różny zależnie od kierunku komunikacji.

System CSIRE w komunikacji z systemami zewnętrznymi będzie zawsze występował w roli Receiving MSH (czyli występować będzie w roli serwera usługi), zaś systemy zewnętrzne zawsze będą występować w roli Sending MSH (czyli będą występować w roli klientów usługi).

Oznacza to, iż wiadomości wysyłane do CSIRE będą przekazywane przez wywołanie AS4 pochodzące z systemów zewnętrznych wg. wzorca One-Way Push (opisany w 5.4.1), zaś wiadomości pochodzące z systemu CSIRE będą musiały być pobrane przez systemy zewnętrzne wg. wzorca Two-Way/Sync (opisany w 5.4.2)

Podstawowe założenia komunikacji z CSIRE:

- Wysyłanie wiadomości do systemu CSIRE odbywać się będzie poprzez wywołanie udostępnionej usługi (operacja SendMessage, patrz 5.4.4) odpowiadającej za przyjęcie i zarejestrowanie transakcji.
- Wiadomości wychodzące z CSIRE zostaną udostępnione do pobrania i to w gestii systemów zewnętrznych będzie pobranie ich z systemu CSIRE (za pomocą operacji PeekMessage patrz 5.4.5) i potwierdzenie ich poprawnego odebrania (za pomocą operacji DequeueMessage).
- Wywołanie operacji DequeueMessage zapewnia niezaprzeczalność dostarczenia wiadomości do systemu zewnętrznego (nie da się poprawnie wywołać operacji DequeueMessage bez poprawnego odczytania rezultatu operacji PeekMessage)

Dla systemów zewnętrznych komunikujących się z CSIRE oznacza to:

- Aktywna komunikacja z systemów zewnętrznych dla wiadomości wychodzących z CSIRE – konieczność cyklicznego odpytywania CSIRE poprzez wywołanie operacji PeekMessage.
- Systemy zewnętrzne zarządzają szybkością pobierania i przetwarzania wiadomości.
- Systemy zewnętrzne zarządzają kolejnością przetwarzania wiadomości (CSIRE wymusza pobranie w kolejności)
- WSDL opisujący WebService zawierający operacje SendMessage, PeekMessage oraz DequeueMessage znajduje się w rozdziale 10.

5.2.1. Założenia odnośnie przekazywanych wiadomości biznesowych.

- Wiadomości biznesowe przekazywane w elemencie paylaod wiadomości AS4 UserMessage (niezależnie czy payload jest cześcią wiadomości czy załącznikiem) powinny być poprawnymi komunikatami XML zgodnymi ze schematami XSD udostępnionymi niezeleżnie.
- Schematy XSD są zgodne ze specyfikacją XML Schema 1.0.
- W ramach pojedynczego wysłania lub odebrania wiadomości z/do CSIRE przekazana może być jedna wiadomość biznesowa zgodna z XSD.
- Grupowanie (paczkowanie) np. dla profili dobowych zostanie uwzględnione w ramach schematów XSD (czyli np. jedna wiadomość, zgodna z XSD, będzie zawierać wiele profili dobowych).
- Wiadomości biznesowe mogą być przekazywane do CSIRE jako payload będący częścią wiadomości AS4 lub jako załącznik. W przypadku użycia kompresji payload musi być przekazany jako załącznik.
- CSIRE będzie udostępniać wiadomości w payload będącym cześcią wiadomości AS4 z wyjątkiem sytuacji gdy włączone zostanie użycie kompresji - wtedy wiadomości będą przekazywane w załączniku.

5.3. Parametry przetwarzania wiadomości

Poniżej znajduje się lista parametrów określających tryb przetwarzania wiadomości (P-Mode) wykorzystywanych w niniejszej specyfikacji, wraz z informacją o charakterze danego parametru.

5.3.1. Parametry PMode dostępne do konfiguracji dla systemów zewnętrznych

Tabela 4 Parametry PMode dostępne do konfiguracji

PMode	**Wymagalność**	**Opis**	**Wartość**
PMode.ID	Obowiązkowy	Identyfikuje zestaw parametrów PMode.	Wygenerowany identyfikator UUID
PMode.Agreement	Obowiązkowy	Jest używany w połączeniu z PMode[1].BusinessInfo.Service i PMode[1].BusinessInfo.Action w celu jednoznacznej identyfikacji zestawu parametrów PMode na podstawie nagłówka komunikatu AS4 (atrybuty w CollaborationInfo ComplexElement).	Dowolny tekst
PMode.Initiator.Party	Obowiązkowy	Kwalifikuje stronę inicjującą MEP	Stała wartość: Identyfikator organizacji
PMode.Initiator.Role	Obowiązkowy	Producent wiadomości pełni rolę inicjatora, czyli rolę strony wysyłającej pierwszą wiadomość wzorca MEP.	Stała wartość: Rola organizacji na rynku
PMode.Responder.Party	Obowiązkowy	Kwalifikuje stronę odbierającą MEP	Stała wartość: Identyfikator organizacji dla roli OIRE
PMode.Responder.Role	Obowiązkowy	Rola odbiorcy wiadomości.	Stała wartość: Rola organizacji na rynku (OIRE)
PMode.MEP	Obowiązkowy	Wzorzec wymiany komunikatów (musi to być identyfikator URI), zob. także 5.4: One-Way MEP reguluje wymianę pojedynczej jednostki wiadomości użytkownika, niezwiązanej z innymi wiadomościami użytkownika: http://docs.oasis-open.org/ebxml-msg/ebms/v3.0/ns/core/200704/oneWay . Two-Way MEP zarządza wymianą dwóch jednostek wiadomości użytkownika w przeciwnych kierunkach: http://docs.oasis-open.org/ebxml-msg/ebms/v3.0/ns/core/200704/twoWay	Możliwe wartości: • One-Way/Push: http://docs.oasis-open.org/ebxml-msg/ebms/v3.0/ns/core/200704/oneWay • Two-Way/Sync: http://docs.oasis-open.org/ebxml-msg/ebms/v3.0/ns/core/200704/twoWay

PMode	**Wymagalność**	**Opis**	**Wartość**
PMode.MEPBinding	Obowiązkowy	Powiązanie kanału transportowego przypisane do MEP (push, pull, sync, push-and-push, push-and-pull, pull-and-push, pull-and-pull, …). CSIRE obsługuje tylko push i sync, musi być zgodny z PMode.MEP	Stała wartość w zależności od MEP: • One-Way/Push: http://docs.oasis-open.org/ebxml-msg/ebms/v3.0/ns/core/200704/push • Two-Way/Sync: http://docs.oasis-open.org/ebxml-msg/ebms/v3.0/ns/core/200704/sync
PMode[1].BusinessInfo.Service	Obowiązkowy	Nazwa usługi, do której ma zostać dostarczona wiadomość Użytkownika. Jest używany w połączeniu z PMode.Agreement i PMode[1].BusinessInfo.Action w celu jednoznacznej identyfikacji zestawu parametrów PMode na podstawie nagłówka komunikatu AS4. Jego zawartość musi być odwzorowana na element eb:Messaging/eb:UserMessage/eb:CollaborationInfo/eb:Service	Stała wartość: MarketMessaging
PMode[1].BusinessInfo.Action	Obowiązkowy	Nazwa akcji, którą ma wywołać UserMessage. Jest używana w połączeniu z PMode.Agreement i PMode[1].BusinessInfo.Service do jednoznacznej identyfikacji zestawu parametrów PMode na podstawie nagłówka komunikatu AS4. Jest jedną ze stałych wartości dla CSIRE. Jego zawartość powinna być odwzorowana na element eb:Messaging/eb:UserMessage/eb:CollaborationInfo/eb:Action	Możliwe wartości zależą od wzroca MEP: Possible values depend on MEP: One-Way/Push: • SendMessage • DequeueMessage Two-Way/Sync: • PeekMessage.request • PeekMessage.reply

PMode	Wymaganość	Opis	Wartość
PMode[1].PayloadService.CompressionType	Opcjonalny	Jeśli jest ustawiony, CSIRE zdekompresuje payload z żądania oraz skompresuje payload dla odpowiedzi zawierającej wiadomość biznesową. Dotyczy tylko payloadu w załączniku SOAP. W systemie CSIRE kompresja AS4 jest włączona domyślnie, przy czym uprawniony użytkownik Portalu Użytkownika profesjonalnego może wyłączyć kompresję AS4 dla określonej Organizacji (podmiotu rynkowego)	application/gzip
PMode[1].Security.X509.Sign	Obowiązkowy	Wartość logiczna wskazująca, czy wiadomości powinny być podpisywane.	Yes/No
PMode[1].Security.X509.Encryption.Encrypt	Obowiązkowy	Parametr wskazujący (jeśli jest prawdziwy), że MSH zaszyfruje: • Wszystkie części payloadu: Każda treść SOAP również zostanie zaszyfrowana. • Załączniki. MSH nie zaszyfruje nagłówka. Jeśli wymagana jest poufność danych w nagłówku, można to osiągnąć poprzez zabezpieczenie na poziomie transportu .	Yes/No

5.3.2. Pozostałe PMode (z wartością stałą bądź nieobsługiwane)

Tabela 5 Parametry PMode ze stałą wartością bądź nieobsługiwane

PMode	Opis	Wartość w CSIRE
PMode[1].Protocol.SOAPVersion	Wersja SOAP, która ma być używana (1.1 lub 1.2).	Stała wartość 1.2
PMode[1].Security.WSSVersion	Wartość reprezentuje wersję WS-Security, która ma być używana, i ma dwie możliwe wartości: 1.0 1.1	Stała wartość 1.1
PMode[1].Security.X509.Encryption.Certificate	Certyfikat publiczny do odszyfrowywania otrzymanych wiadomości.	Zarządzanie certyfikatami odbywa się z użyciem Portalu Użytkownika profesjonalnego.

PMode	Opis	Wartość w CSIRE
PMode[1].Security.X509.Signature.Certificate	Certyfikat publiczny do weryfikacji otrzymanych podpisanych wiadomości.	Zarządzanie certyfikatami odbywa się z użyciem Portalu Użytkownika profesjonalnego.
PMode[1].Security.X509.Signature.HashFunction	Algorytm używany do obliczania skrótu podpisywanej wiadomości. Definicje tych wartości znajdują się w specyfikacji [**Błąd! Nie można odnaleźć źródła odwołania.**].	http://www.w3.org/2001/04/xmlence#sha256
PMode[1].Security.X509.Signature.Algorithm	Identyfikuje algorytm obliczania wartości podpisu cyfrowego.	http://www.w3.org/2001/04/xmldsig-more#rsa-sha256
PMode[1].Security.X509.Encryption.Algorithm	Algorytm szyfrowania, który ma być używany.	Patrz 6.3.2
PMode[1].Security.X509.Encryption.MinimumStrength	Wartość całkowita określająca efektywną siłę, którą algorytm szyfrowania musi zapewnić w postaci efektywnych lub losowych bitów. Wartość jest mniejsza niż długość klucza w bitach, gdy w kluczu używane są bity kontrolne. Np. 8 bitów kontrolnych 64-bitowego klucza DES nie zostanie uwzględnionych w zliczaniu. Ustawienie MinimumStrength na 56 jest wymagane, aby mieć minimalną siłę równą tej dostarczanej przez DES.	Stała wartość 128
PMode[1].ErrorHandling.Report.AsResponse	Ten parametr typu boolean wskazuje, czy (jeśli „prawda”) błędy wygenerowane w wyniku odebrania błędnej wiadomości są przesyłane przez tylny kanał bazowego protokołu powiązanego z błędną wiadomością, czy nie.	Zawsze prawda.
PMode[1].ReceptionAwareness.Retry	Parametr logiczny wskazujący (jeśli to prawda), że kroki podjęte w celu zapewnienia odbioru wiadomości zostaną powtórzone, jeśli to konieczne.	Zawsze prawda.
PMode.Initiator.Authorization.username	Opisuje informacje autoryzacyjne dla komunikatów wysyłanych przez inicjatora, które mają być przetwarzane po stronie odbiorcy.	Nieużywany. CSIRE nie oczekuje, że otrzyma nazwę użytkownika/hasło przez kanał AS4.
PMode.Initiator.Authorization.password		
PMode.Responder.Authorization.username	Opisuje informacje autoryzacyjne dla wiadomości wysyłanych przez	Nieużywany. CSIRE nie przewiduje wysyłania nazwy użytkownika/hasła kanałem AS4.

PMode	Opis	Wartość w CSIRE
PMode.Responder.Authorization.password	respondenta, które mają być przetwarzane po stronie inicjatora.	
PMode[1].Protocol.Address	Reprezentuje adres (adres URL punktu końcowego) odbiornika MSH (lub strony odbiorcy), do którego mają być wysyłane komunikaty.	Nieużywany. Organizacje zawsze inicjują komunikację z CSIRE, dlatego konfiguracja adresu URL, na który organizacje mają otrzymywać wiadomości, nie jest wymagana.
PMode[1].BusinessInfo.PayloadProfile.maxSize	Ten parametr pozwala na określenie maksymalnego rozmiaru w kilobajtach dla całego payloadu, czyli dla sumy wszystkich części ładunku.	Nieużywany. Dla wszystkich wiadomości wymienianych z CSIRE stosowana jest stała wartość maksymalna wynosząca 100 MB.
PMode[1].BusinessInfo.Properties[]	Wartością tego parametru jest lista właściwości. Właściwość to struktura danych składająca się z czterech wartości: nazwy właściwości, której można użyć jako identyfikator właściwości (np. wymagana właściwość o nazwie „messagetype” może być zapisana jako: Właściwości[typ wiadomości].required="true"); opis właściwości; typ danych właściwości; i Wartość logiczna wskazująca, czy właściwość jest oczekiwana, czy opcjonalna w komunikacie użytkownika. Ten parametr steruje zawartością elementu eb:Messaging/eb:UserMessage/eb:MessageProperties.	Nieużywany.
PMode[1].BusinessInfo.PayloadProfile[]	Ten parametr pozwala na określenie ograniczenia lub profilu dla payloadu.	Nieużywany.
PMode[1].ErrorHandling.Report.ProcessErrorNotifyConsumer	Parametr logiczny wskazujący (jeśli true), że konsument (aplikacja/strona) komunikatu użytkownika pasującego do PMode powinien zostać powiadomiony, gdy wystąpi błąd w odbierającym MSH.	Nieużywany.

PMode	Opis	Wartość w CSIRE
PMode[1].ErrorHandling.Report.DeliveryFailuresNotifyProducer	Parametr typu boolean wskazujący (jeśli true), że podczas przetwarzania komunikatu użytkownika do wysłania producent (aplikacja/strona) komunikatu użytkownika pasującego do PMode powinien zostać powiadomiony, gdy wystąpi błąd w wysyłającym MSH.	Nieużywany.
PMode[1].ErrorHandling.Report.MissingReceiptNotifyProducer	Parametr typu boolean wskazujący (jeśli jest prawdziwy), że błąd EBMS:0301 MissingReceipt musi zostać zwrócony przez wysyłający MSH do odbierającego MSH w przypadku, gdy nie zostanie zwrócony żaden AS4 Receipt.	Nieużywany
PMode[1].ErrorHandling.Report.ProcessErrorNotifyProducer	CSIRE zawsze zwraca wszelkie błędy, które wystąpiły podczas przetwarzania UserMessages, ponieważ jest to kluczowe dla rynków centralnych, wszystkie organizacje muszą wiedzieć, kiedy ich transakcja biznesowa nie została pomyślnie przetworzona i podjąć odpowiednie działania.	Nieużywany.
PMode[1].ErrorHandling.Report.ReceiverErrorsTo	Adres lub rozdzielona przecinkami lista adresów, na które mają być wysyłane błędy ebMS wygenerowane przez MSH, który odbiera błędny komunikat. np. Może to być adres MSH wysyłającego błędną wiadomość.	Nieużywany.
PMode[1].ErrorHandling.Report.SenderErrorsTo	Adres — lub rozdzielona przecinkami lista adresów — na który mają zostać wysłane błędy wygenerowane przez MSH, który próbował wysłać błędny komunikat.	Nieużywany.
PMode[1].Protocol.Address	Adres URL punktu końcowego odbiornika MSH (lub strony odbiorcy), do którego mają być wysyłane komunikaty w części PMode.	Nieużywany.
PMode[1].ReceptionAwareness	Parametr logiczny wskazujący (jeśli prawda), że należy podjąć kroki w celu zapewnienia odbioru wiadomości.	Nieużywany.

PMode	Opis	Wartość w CSIRE
PMode[1].ReceptionAwareness.Retry.Parameters	Parametr określający wymagania dotyczące ponownych prób wywołania.	Nieużywany.
PMode[1].ReceptionAwareness.DuplicateDetection	Wykrywanie zduplikowanych wiadomości jest zawsze włączone w CSIRE i nie można nim sterować za pomocą parametrów.	Nieużywany.
PMode[1].ReceptionAwareness.DuplicateDetection.Parameters	Wykrywanie zduplikowanych wiadomości jest zawsze włączone w CSIRE i nie można nim sterować za pomocą parametrów.	Nieużywany.
PMode[1].Security.PModeAuthorize	Parametr logiczny wskazujący (jeśli true), że komunikat w MEP musi zostać autoryzowany do przetwarzania w trybie PMode. Jeśli parametr ma wartość true, oznacza to, że w tym celu należy użyć następujących elementów: PMode.Responder.Authorization.{ username/password}, jeśli wiadomość jest wysyłana przez Respondera . PMode.Initiator.Authorization, jeśli wiadomość jest wysyłana przez Initiator . np. po ustawieniu na true dla komunikatu PushRequest wysłanego przez inicjatora, push będzie autoryzowany tylko przez MPC wskazany przez ten sygnał Push , jeśli: MPC jest taki sam , jak określono w nodze PMode dla przesyłanej wiadomości; I sygnał zawiera ważne dane uwierzytelniające (tj. nazwę użytkownika/hasło).	Nieużywany.
PMode[1].Security.SendReceipt	Parametr logiczny wskazujący (jeśli true), że podpisana wiadomość Receipt zawierająca skrót wiadomości musi zostać odesłany.	Nieużywany.

PMode	Opis	Wartość w CSIRE
PMode[1].Security.SendReceipt.NonRepudiation	Parametr logiczny wskazujący (jeśli true), że wymagana jest niezaprzeczalność odbioru . W przeciwnym razie (jeśli false) wymagana jest tylko świadomość odbioru. Niezaprzeczalność uniemożliwia odbiorcy zaprzeczenie odbioru wiadomości. Potwierdzenia niezaprzeczalności muszą być wysyłane synchronicznie dla każdego typu wiadomości.	Nieużywany.
PMode[1].Security.SendReceipt.ReplyPattern	Wskazuje, czy ma zostać wysłany sygnał odbioru: jako wywołanie zwrotne na oddzielnym połączeniu. (wartość " wywołanie zwrotne "); Lub synchronicznie w odpowiedzi HTTP lub kanale zwrotnym (wartość „ response ”). Jeśli nie ma go w PMode, można użyć dowolnego wzorca.	Nieużywany.
PMode[1].Security.UsernameToken.username	Nazwa użytkownika do uwzględnienia w tokenie nazwy użytkownika WSS .	Nieużywany.
PMode[1].Security.UsernameToken.password	Hasło do użycia wewnątrz tokena nazwy użytkownika WSS.	Nieużywany.
PMode[1].Security.UsernameToken.Digest	Wskazuje, czy skrót hasła zostanie uwzględniony w elemencie WSS UsernameToken.	Nieużywany.
PMode[1].Security.UsernameToken.Nonce	Wskazuje, czy element WSS UsernameToken będzie zawierał element Nonce. Nonce => liczba lub ciąg bitów używany tylko raz w inżynierii bezpieczeństwa.	Nieużywany.
PMode[1].Security.UsernameToken.Created	Wskazuje, czy element WSS UsernameToken będzie miał utworzony element sygnatury czasowej.	Nieużywany.

5.4. Wzorce wymiany komunikatów AS4 (MEP)

W ramach rozwiązania stosowanego na potrzeby CSIRE, wykorzystywane będą dwa, spośród czterech dostępnych w ramach Protokołu AS4, wzorców wymiany wiadomości.

Każda interakcja pomiędzy stronami wymieniającymi komunikaty (OIRE, Użytkownicy profesjonalni, Użytkownicy uprawnieni), będzie wymagała zastosowania odpowiedniego wzorca (MEP).

Poniżej przedstawione zostaną poszczególne wzorce wymiany wiadomości.

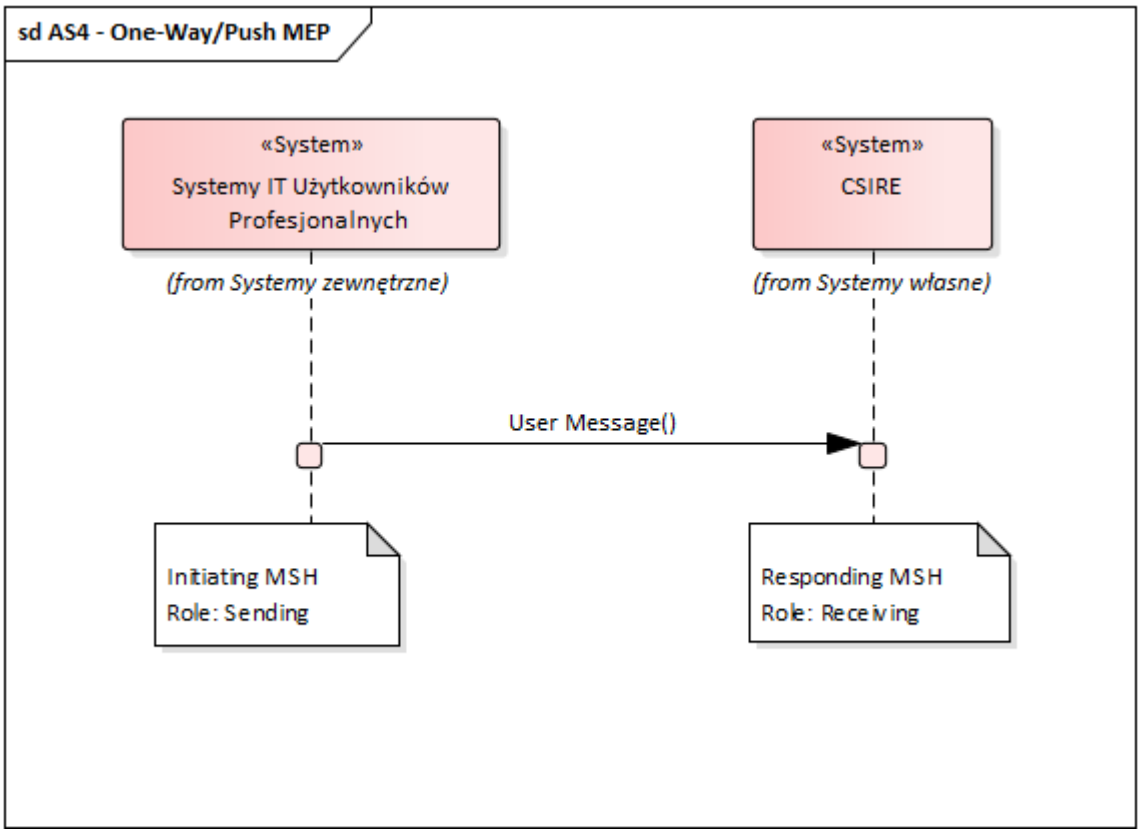
5.4.1. One-Way/Push MEP

Opis wzorca komunikacji można przedstawić za pomocą sekwencji następujących po sobie zdarzeń.

1. partner inicjujący (system zewnętrzny), wykorzystując Message Handler (*Initiating MSH*), wysyła wiadomość do partnera odbierającego (*Receiving MSH*).

2. w rekacji na przesłaną wiadomość, w sposób synchroniczny otrzymuje jedynie status odpowiedzi HTTP (202) oznaczający przyjęcie wiadomości do dalszego procesowania.

Wzorzec ten obrazuje następujący diagram:



Rysunek 3 One-Way/Push MEP

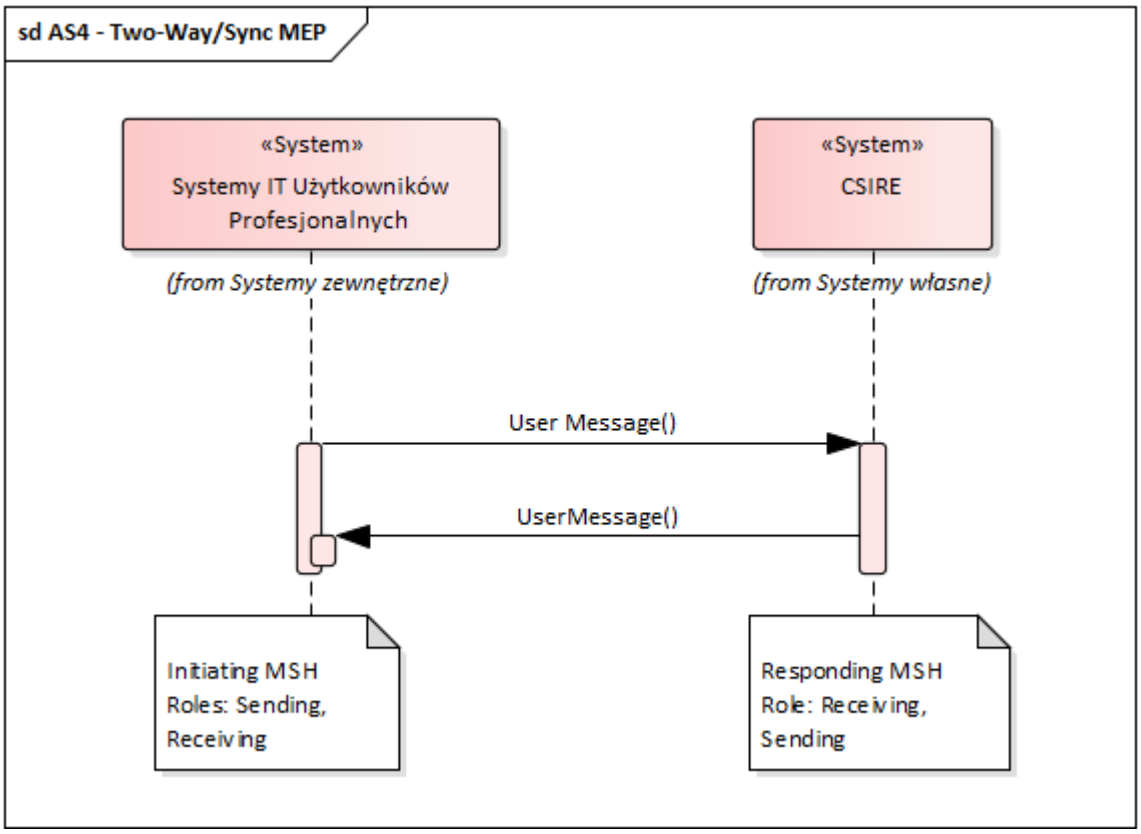
5.4.2. Two-Way/Sync MEP

Opis wzorca komunikacji można przedstawić za pomocą sekwencji następujących po sobie zdarzeń.

1. partner inicjujący (system zewnętrzny), wykorzystując Message Handler (*Initiating MSH*), wysyła wiadomość do partnera odbierającego (*Receiving MSH*)

2. odpytywany Message Handler (CSIRE) zwraca do partnera inicjującego synchronicznie odpowiedź na zadane żądanie.

Wzorzec ten obrazuje następujący diagram:



Rysunek 4 Two-Way/Sync MEP

5.4.3. Wzorce komunikacji systemu CSIRE

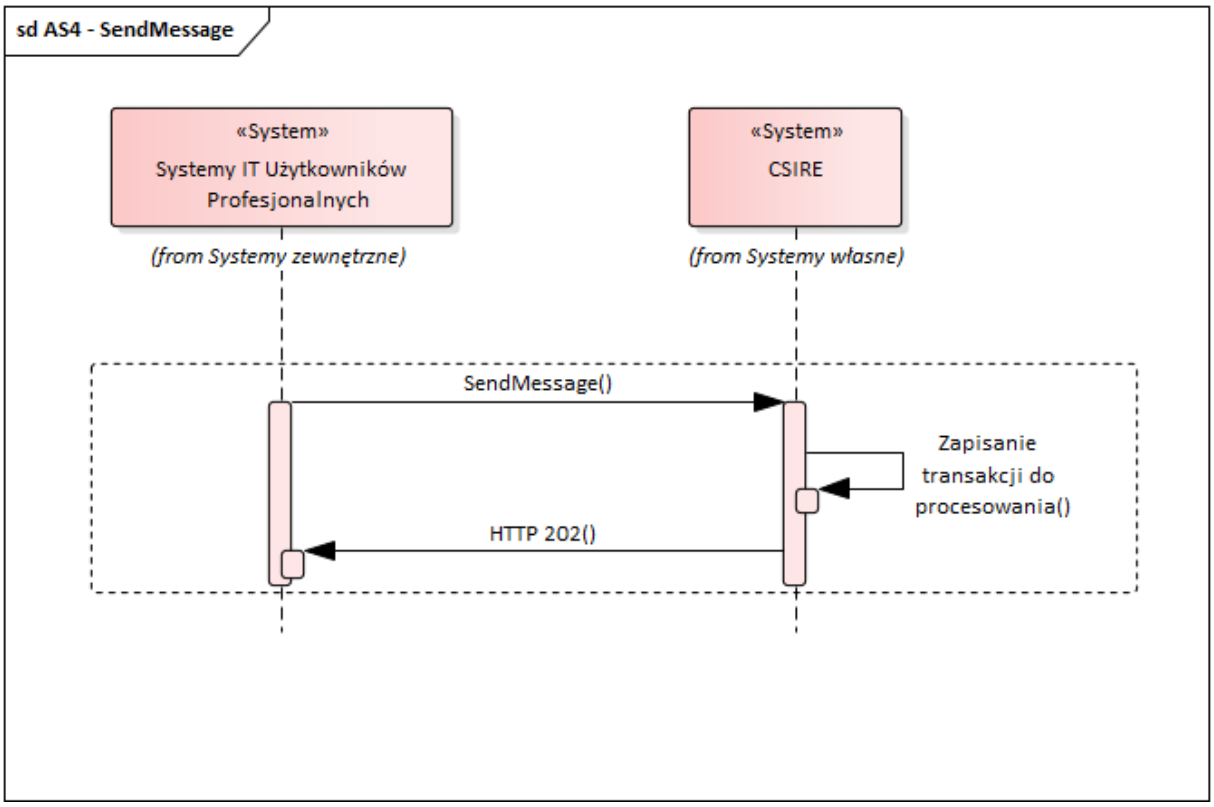
Poniżej przedstawiono sposób komunikacji z systemem CSIRE przy wykorzystaniu mechanizmów AS4.

Dla poniżej przedstawionych operacji opisane są jedynie techniczne kody błędów tzn. takie które wynikają wprost z implementacji warstwy transportowej lub warstwy AS4. Dokument nie opisuje biznesowych kodów błędów pochodzących z TSKB – wiadomości zawierające takie kody biznesowe będą pobierane z użyciem operacji PeekMessage opisanej poniżej (analogicznie jak wszystkie inne wiadomości opisane w TSKB).

5.4.4. Wysłanie wiadomości do CSIRE

Aby wysłać wiadomość do CSIRE system zewnętrzny musi wywołać operację SendMessage, która będzie zrealizowana wg. wzorca One-Way Push.

W scenariuszu tym system zewnętrzny wysyła do CSIRE wiadomość i w sposób synchroniczny otrzymuje jedynie status odpowiedzi (HTTP 202) potwierdzający przyjęcie wiadomości do procesowania.



Rysunek 5 Operacja SendMessage

5.4.4.1. Operacja SendMessage

- Jako wywołanie jest przesyłana wiadomość UserMessage (AS4) zawierająca payload zgodny z XSD (patrz 5.4.4.2)
- W przypadku przyjęcia wiadomości do procesowania zwracany jest kod HTTP 202 a wiadomość zapisywana jest w systemie do dalszego procesowania.
  Notyfikacje dotyczące przetwarzania (zgodne z specyfikacją wiadomości opisaną w TSKB) zostaną wygenerowane przez CSIRE i będą mogły być pobrane z użyciem operacji PeekMessage opisanej kolejnych rozdziałach.
- W przypadku błędu przyjęcia wiadomości do procesowania zwracany jest komunikat zgodny z opisem w punktach 5.4.6 oraz 5.4.7

5.4.4.2. Struktura wiadomości dla SendMessage

Struktura wiadomości UserMessage (AS4) przekazywanej w ramach operacji SendMessage

Element	Kardynalność	Typ	Opis
SendMessageRequest	1..1	Complex Element	Główny element reprezentujący wywołanie SendMessage
MessageContainer	1..1	Complex Element	Element zawierający wiadomość przekazywaną w ramach operacji SendMessage
Payload	1..1	Complex Element	Zawiera wiadomość XML zgodną z schematem XSD opracowanym są na podstawie opisu

			komunikatów z TSKB i zgodnym ze specyfikacją XML Schema 1.0.

5.4.4.2.1. Przykład wywołania SendMessage

```
<soapenv:Envelope xmlns:soapenv="http://www.w3.org/2003/05/soap-envelope"
xmlns:urn="urn:cms:b2b:v01" xmlns:urn1="urn:cms:b2b:message:v01:v1">
  <soapenv:Header>
    <eb:Messaging xmlns:eb="http://docs.oasis-open.org/ebxml-msg/ebms/v3.0/ns/core/200704/"
soapenv:mustUnderstand="1">
      <eb:UserMessage>
        <eb:MessageInfo>
          <eb:Timestamp>2023-08-03T07:36:20.656Z</eb:Timestamp>
          <eb:MessageId>d7c3eccf-0781-4789-a456-035b39e8bb20</eb:MessageId>
        </eb:MessageInfo>
        <eb:PartyInfo>
          <eb:From>
            <eb:PartyId>ExampleParty1</eb:PartyId>
            <eb:Role>ExampleParty1Role</eb:Role>
          </eb:From>
          <eb:To>
            <eb:PartyId>ExampleParty2</eb:PartyId>
            <eb:Role>ExampleParty2Role</eb:Role>
          </eb:To>
        </eb:PartyInfo>
        <eb:CollaborationInfo>
          <eb:AgreementRef>SendMessageAgreementExample</eb:AgreementRef>
          <eb:Service>MarketMessaging</eb:Service>
          <eb:Action>SendMessage</eb:Action>
          <eb:ConversationId>2a81ffbd-0d3d-4cbd-8601-d916e0ed2fe2</eb:ConversationId>
        </eb:CollaborationInfo>
      </eb:UserMessage>
    </eb:Messaging>
  </soapenv:Header>
  <soapenv:Body>
    <urn:SendMessageRequest>
      <urn:MessageContainer>
        <urn:Payload>
          ...
        </urn:Payload>
      </urn:MessageContainer>
    </urn:SendMessageRequest>
  </soapenv:Body>
</soapenv:Envelope>
```

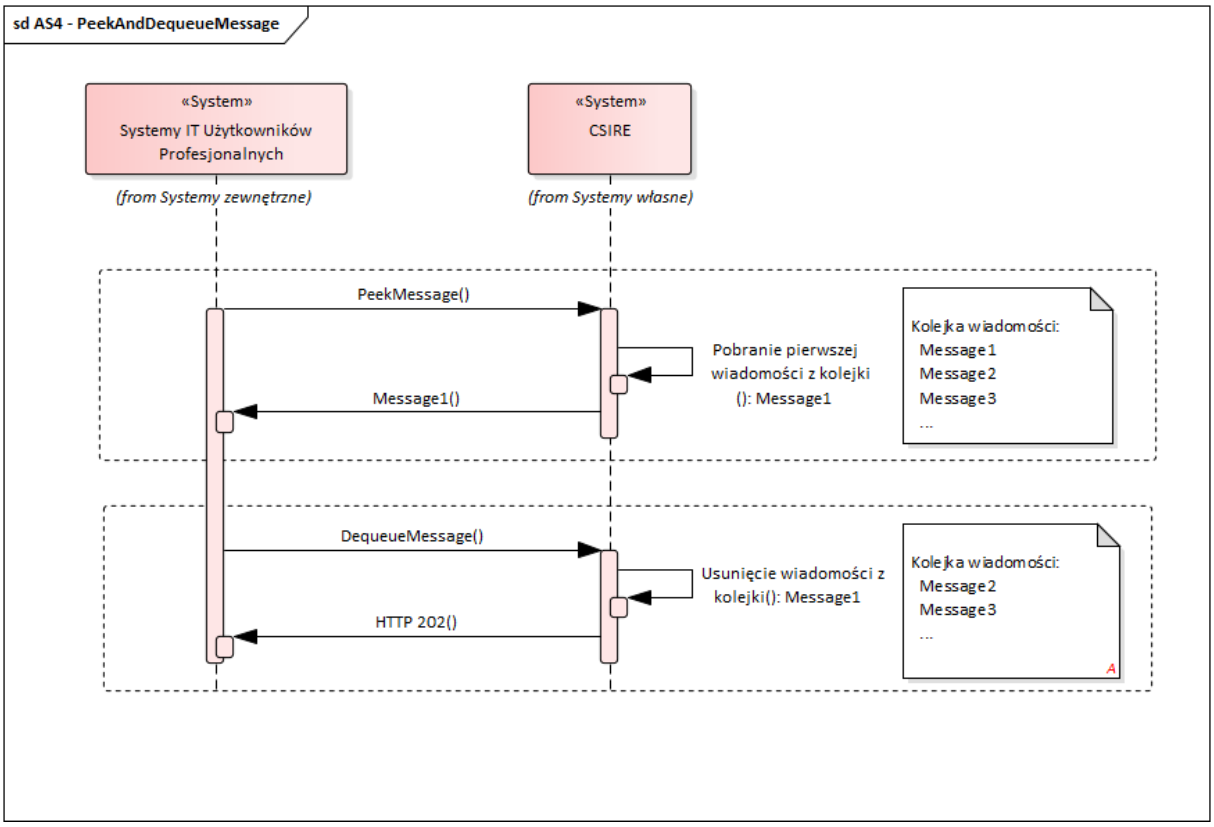
5.4.4.2.1. Przykład odpowiedzi w przypadku błędu EBMS:0001

```
<soapenv:Envelope xmlns:soapenv="http://schemas.xmlsoap.org/soap/envelope/"
                  xmlns:eb="http://docs.oasis-open.org/ebxml-msg/ebms/v3.0/ns/core/200704/">
  <soapenv:Header>
    <eb:Messaging soapenv:mustUnderstand="1">
      <eb:SignalMessage>
        <eb:MessageInfo>
          <eb:Timestamp>2023-08-03T07:21:17.993Z</eb:Timestamp>
          <eb:MessageId>d7c3eccf-0781-4789-a456-035b39e8bb20</eb:MessageId>
        </eb:MessageInfo>
        <eb:Error origin="ebMS"
                  category="Content"
                  errorCode="EBMS:0001"
                  severity="failure"
                  refToMessageInError="d7c3eccf-0781-4789-a456-375b39e8bccf">
          <eb:Description>Value not recognized</eb:Description>
        </eb:Error>
      </eb:SignalMessage>
    </eb:Messaging>
  </soapenv:Header>
  <soapenv:Body/>
</soapenv:Envelope>
```

5.4.5. Pobranie wiadomości z CSIRE

W ceu zapewnienia niezaprzeczalności odebrania pobranie wiadomości z CSIRE zostało podzielone na dwie techniczne operacje:

- PeekMessage – zrealizowaną wg. wzorca Two-Way Sync
- DequeueMessage - zrealizowaną wg. wzorca One-Way Push

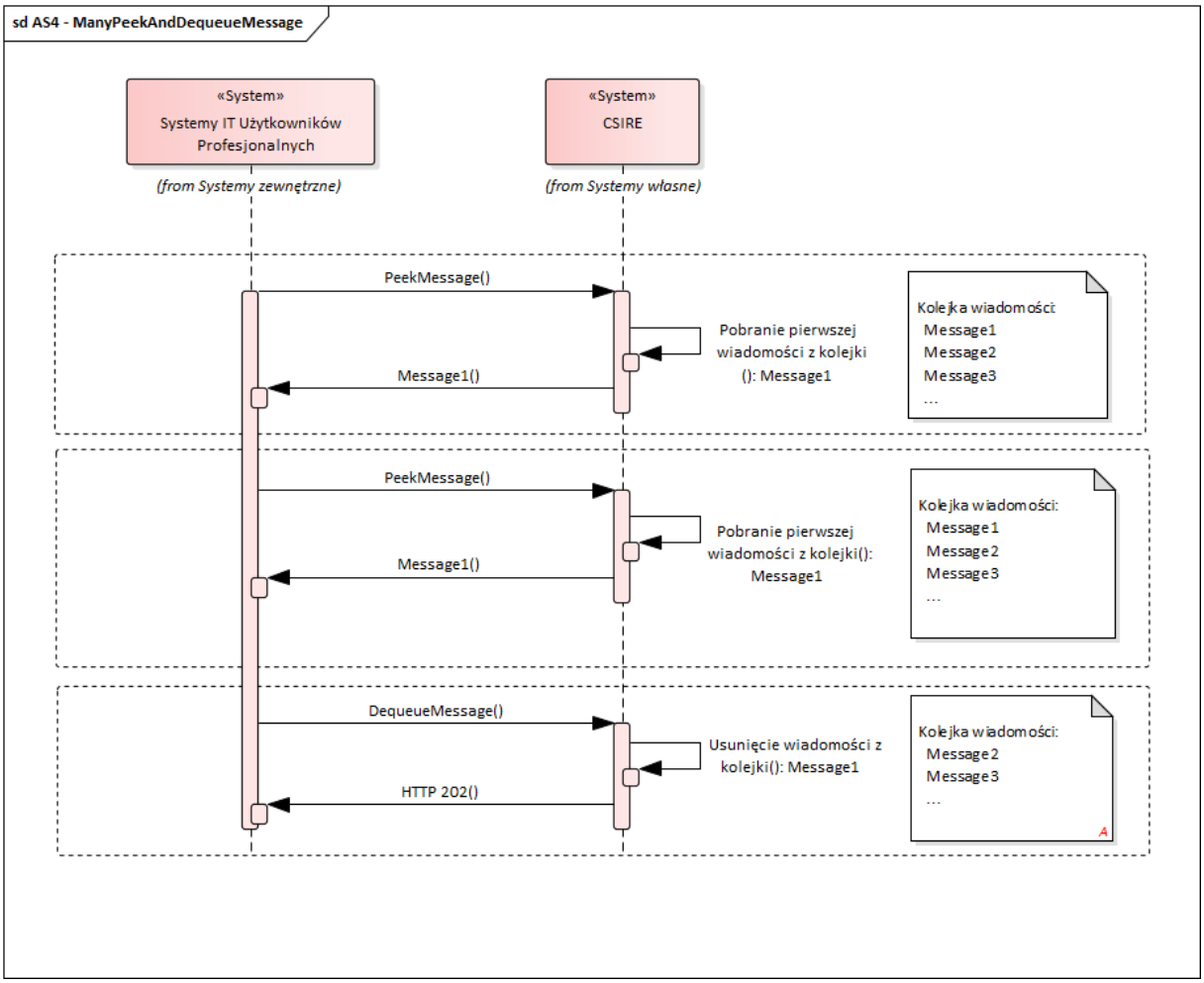


Rysunek 6 Operacje PeekMessage i DequeueMessage – prawidłowa sekwencja wywołań

Operacja PeekMessage służy do pobrania wiadomości z „kolejki" przez system zewnętrzny. Operacja ta zwraca pierwszą wiadomość w logicznej kolejce (zgodnie z FIFO), która nie została jeszcze usunięta. Należy pamiętać, że PeekMessage zwraca komunikat, który może zostać przetworzony przez wywołującego PeekMessage, bez uprzedniego usunięcia tej wiadomości z kolejki (z użyciem operacji DequeueMessage opisanej niżej).

Obowiązkiem Uczestnika Rynku jest regularne przeglądanie, przetwarzanie i usuwanie komunikatów z kolejki. CSIRE będzie kontynuował przetwarzanie i przygotowywanie kolejnych komunikatów niezależnie od odbierania ich przez Uczestnika Rynku. Wiadomości są dostarczane w kolejności, w jakiej CSIRE je utworzył.

Wielokrotne wywołanie operacji PeekMessage bez wywołania operacji DequeueMessage spowoduje zwrócenie tej samej wiadomości (patrz rysunek 7).



Rysunek 7 Pierwsze wywołanie PeekMessage bez DequeueMessage – nieprawidłowa sekwencja wywołań (jeśli nie chcemy ponownie pobrać tej samej wiadomości)

Do potwierdzenia poprawności pobrania wiadomości służy operacja DequeueMessage – po jej wykonaniu wiadomość jest usuwana z kolejki i system zewnętrzny będzie mógł przejść do pobierania następnej wiadomości.

Systemy zewnętrzne powinny cyklicznie odpytywać CSIRE (poprzez wywołanie operacji PeekMessage) odnośnie oczekujących wiadomości, w szczególności:

- W przypadku pobrania wiadomości z użyciem PeekMessage i technicznego potwierdzenia z użyciem DequeueMessage kolejne wywołanie PeekMessage powinno nastąpić niezwłocznie po wywołaniu DequeueMessage.
- W przypadku wywołania PeekMessage, dla którego CSIRE nie zwróciło wiadomości kolejne wywołaie PeekMessage powinno nastąpić po 15 sekundach.

5.4.5.1. Kolejki wyjściowe z CSIRE

- Operacja PeekMessage (opisana w 5.4.5.2) umożliwia podanie nazwy kolejki (w elemencie MessageDomain) z której chcemy pobrać wiadomość.
- Jeśli w wywołaniu operacji PeekMessage podamy wiele nazw kolejek (wiele elementów MessageDomain) system CSIRE zwróci jedną, najstarszą wiadomość z kolejek przekazanych w wywołaniu.

- Jeśli w wywołaniu operacji PeekMessage nie podamy nazwy kolejki system CSIRE zwróci jedną, najstarszą wiadomość z ze wszystkich kolejek.
- Zdefiniowanie wielu kolejek wyjściowych umożliwa systemom zewnętrznych równoległe pobieranie z nich wiadomości.

Nazwa kolejki	Przeznaczenie
AGREEMENTS	Wiadomości z grupy 1 procesów SWI
MPUPDATES	Wiadomości z grupy 2 procesów SWI
MPNOTIFICATIONS	Wiadomości z grupy 3 procesów SWI
MPREQUESTS	Wiadomości z grupy 4 procesów SWI
BRPCHANGE	Wiadomości z grupy 5 procesów SWI
DATALOAD	Wiadomości z grupy 6 procesów SWI bez profili dobowych (proces 6.1)
DAILYPROFILES	Wiadomości dotyczące zawierające profili dobowych (procesy 6.1, 7.1)
DATASHARE	Wiadomości z grupy 7 procesów SWI bez profili dobowych (proces 7.1)
CONNECTIONUPDATES	Wiadomości z grupy 8 procesów SWI
PARTIESINFOEXCHANGE	Wiadomości z grupy 9 procesów SWI
FACILITIESUPDATES	Wiadomości z grupy 10 procesów SWI

Tabela 6 Nazwy kolejek wyjściowych CSIRE

5.4.5.2. Operacja PeekMessage

- Zrealizowana wg. wzorca Two-Way Sync
- Wywołanie odpowiada wiadomości typu UserMessage (AS4) zawierającej payload zgodny z XSD (patrz 5.4.5.2)
- System zewnętrzny może w ramach wiadomości UserMessage wysłać informacje z jakiej kolejki systemu CSIRE chce pobrać wiadomość (element Message Domain).
- Poprawne wywołanie skutkuje zwróceniem wiadomości typu UserMessage (AS4) zawierającej payload zgodny z XSD (patrz 5.4.5.2).
- Niepoprawne wywołanie skutkuje błędem zgodnym z punktami 5.4.6 oraz 5.4.7.

5.4.5.3. Struktura wiadomości dla PeekMessage

Struktura wiadomości UserMessage (AS4) przekazywanej do systemu CSIRE jako wywołanie:

Element	Kardynalność	Typ	Opis
PeekMessageRequest	1..1	Complex Element	Główny element reprezentujący wywołanie PeekMessage
MessageDomains	0..1	Complex Element	Opcjonalny element zawierający liste kolejek z jakich należy pobrać

			wiadomość
MessageDomain	1..n	xs:string max=100	Element wskazujący z jakich kolejek z systemu CSIRE operacja PeekMessage ma pobrać pierwszą wiadomość

Struktura wiadomości UserMessage (AS4) przekazywanej z CSIRE jako odpowiedź na wywołanie:

Element	Kardynalność	Typ	Opis
PeekMessageResponse	1..1	Complex Element	Główny element reprezentujący odpowiedź na wywołanie PeekMessage
MessageContainer	0..1	Complex Element	Tylko dla komunikatów umieszczonych w kolejce
DocumentReferenceNumber	1..1	xs:string max=36	Identyfikator DocumentReferenceNumber (i.e. UUID) wygenerowany przez CMS w celu zidentyfikowania transferu danych komunikatu, który powinien zostać wykorzystany do późniejszego Dequeue tego komunikatu.
Payload	1..1	Complex Element	Zawiera wiadomość XML zgodną z schematem XSD opracowanym są na podstawie opisu komunikatów z TSKB i zgodnym ze specyfikacją XML Schema 1.0.

5.4.5.3.1. Przykład wywołania PeekMessage

```
<soapenv:Envelope xmlns:soapenv="http://www.w3.org/2003/05/soap-envelope"
xmlns:urn="urn:cms:b2b:v01" xmlns:urn1="urn:cms:b2b:message:v01:v1">
  <soapenv:Header>
    <eb:Messaging xmlns:eb="http://docs.oasis-open.org/ebxml-msg/ebms/v3.0/ns/core/200704/"
soapenv:mustUnderstand="1">
      <eb:UserMessage>
        <eb:MessageInfo>
          <eb:Timestamp>2023-08-03T07:36:20.656Z</eb:Timestamp>
          <eb:MessageId>d7c3eccf-0781-4789-a456-035b39e8bb20</eb:MessageId>
        </eb:MessageInfo>
        <eb:PartyInfo>
          <eb:From>
            <eb:PartyId>ExampleParty1</eb:PartyId>
            <eb:Role>ExampleParty1Role</eb:Role>
          </eb:From>
          <eb:To>
            <eb:PartyId>ExampleParty2</eb:PartyId>
            <eb:Role>ExampleParty2Role</eb:Role>
          </eb:To>
        </eb:PartyInfo>
        <eb:CollaborationInfo>
          <eb:AgreementRef>PeekMessageAgreementExample</eb:AgreementRef>
          <eb:Service>MarketMessaging</eb:Service>
          <eb:Action>PeekMessage.request</eb:Action>
          <eb:ConversationId>2a81ffbd-0d3d-4cbd-8601-d916e0ed2fe2</eb:ConversationId>
        </eb:CollaborationInfo>
      </eb:UserMessage>
    </eb:Messaging>
```

```
  </soapenv:Header>
  <soapenv:Body>
    <urn:PeekMessageRequest>
         <urn:MessageDomains>
            <urn:MessageDomain>DATALOAD</urn:MessageDomain>
         </urn:MessageDomains>
      </urn:PeekMessageRequest>
  </soapenv:Body>
</soapenv:Envelope>
```

5.4.5.3.1. Przykład odpowiedzi PeekMessage

```
<soapenv:Envelope xmlns:soapenv="http://www.w3.org/2003/05/soap-envelope"
xmlns:urn="urn:cms:b2b:v01" xmlns:urn1="urn:cms:b2b:message:v01:v1">
  <soapenv:Header>
    <eb:Messaging xmlns:eb="http://docs.oasis-open.org/ebxml-msg/ebms/v3.0/ns/core/200704/"
soapenv:mustUnderstand="true">
      <eb:UserMessage>
        <eb:MessageInfo>
          <eb:Timestamp>2023-08-03T07:36:21.641Z</eb:Timestamp>
          <eb:MessageId>d7c3eccf-0781-4789-a456-375b39e8bccf</eb:MessageId>
        </eb:MessageInfo>
        <eb:PartyInfo>
          <eb:From>
            <eb:PartyId>ExampleParty2</eb:PartyId>
            <eb:Role>ExampleParty2Role</eb:Role>
          </eb:From>
          <eb:To>
            <eb:PartyId>ExampleParty1</eb:PartyId>
            <eb:Role>ExampleParty1Role</eb:Role>
          </eb:To>
        </eb:PartyInfo>
        <eb:CollaborationInfo>
          <eb:AgreementRef>PeekMessageAgreementExample</eb:AgreementRef>
          <eb:Service>MarketMessaging</eb:Service>
          <eb:Action>PeekMessage.reply</eb:Action>
          <eb:ConversationId>2a81ffbd-0d3d-4cbd-8601-d916e0ed2fe2</eb:ConversationId>
        </eb:CollaborationInfo>
      </eb:UserMessage>
    </eb:Messaging>
  </soapenv:Header>
  <soapenv:Body>
    <urn:PeekMessageResponse>
      <urn:MessageContainer>
        <urn:DocumentReferenceNumber>cc3ae4a7-e93f-406a-99c8-
4bbc66ab5140</urn:DocumentReferenceNumber>
        <urn:Payload>
                  ...
        </urn:Payload>
      </urn:MessageContainer>
    </urn:PeekMessageResponse>
  </soapenv:Body>
</soapenv:Envelope>
```

5.4.5.3.2. Przykład odpowiedzi PeekMessage, gdy brak wiadomości w kolejce (EBMS:0006).

```
<env:Envelope xmlns:env="http://www.w3.org/2003/05/soap-envelope">
  <env:Header>
    <ns2:Messaging xmlns:ns2="http://docs.oasis-open.org/ebxml-msg/ebms/v3.0/ns/core/200704/"
                   xmlns:ns3="http://schemas.xmlsoap.org/soap/envelope/"
                   env:mustUnderstand="true">
      <ns2:SignalMessage>
        <ns2:MessageInfo>
          <ns2:Timestamp>2023-08-03T07:21:17.993Z</ns2:Timestamp>
          <ns2:MessageId>7d3e50b4-f372-4c48-865b-8193f3dd674c</ns2:MessageId>
          <ns2:RefToMessageId>10891C6e-8d0c-4701-9a1d-c84fd39d4832</ns2:RefToMessageId>
        </ns2:MessageInfo>
        <ns2:Error category="Communication"
                   errorCode="EBMS:0006"
                   origin="ebMS"
                   refToMessageInError="10891C6e-8d0c-4701-9a1d-c84fd39d4832"
```

```
                    severity="warning"
                    shortDescription="EmptyMessagePartitionChannel">
          <ns2:Description xml:lang="En">The Message queue is empty</ns2:Description>
          <ns2:ErrorDetail>The Message queue is empty</ns2:ErrorDetail>
        </ns2:Error>
      </ns2:SignalMessage>
    </ns2:Messaging>
  </env:Header>
  <env:Body/>
</env:Envelope>
```

5.4.5.4. Operacja DequeueMessage

- Zrealizowaną jako wzorzec One-Way Push
- Wywołanie odpowiada wiadomości typu UserMessage (AS4) zawierającej payload zgodny z XSD (patrz 5.4.5.4).
- Poprawne wywołanie skutkuje zwróceniem kodu HTTP 202.
- W przypadku błędu zwracany jest zwracany jest komunikat zgodny z opisem w punktach 5.4.6 oraz 5.4.7.

5.4.5.5. Struktura wiadomości dla DequeueMessage

Struktura wiadomości UserMessage (AS4) przekazywanej do systemu CSIRE jako wywołanie:

Element	Kardynalność	Typ	Opis
DequeueMessageRequest	1..1	Complex Element	Główny element reprezentujący wywołanie DequeueMessage
DocumentReferenceNumber	1..1	xs:string max=36	UUID - DocumentReferenceNumber w komunikacie z poprzednio podglądniętego komunikatu (patrz PeekMessage).

5.4.5.5.1. Przykład wywołania DequeueMessage

```
<soapenv:Envelope xmlns:soapenv="http://www.w3.org/2003/05/soap-envelope"
xmlns:urn="urn:cms:b2b:v01" xmlns:urn1="urn:cms:b2b:message:v01:v1">
  <soapenv:Header>
    <eb:Messaging xmlns:eb="http://docs.oasis-open.org/ebxml-msg/ebms/v3.0/ns/core/200704/"
soapenv:mustUnderstand="1">
      <eb:UserMessage>
        <eb:MessageInfo>
          <eb:Timestamp>2023-08-03T07:36:20.656Z</eb:Timestamp>
          <eb:MessageId>d7c3eccf-0781-4789-a456-035b39e8bb20</eb:MessageId>
        </eb:MessageInfo>
        <eb:PartyInfo>
          <eb:From>
            <eb:PartyId>ExampleParty1</eb:PartyId>
            <eb:Role>ExampleParty1Role</eb:Role>
          </eb:From>
          <eb:To>
            <eb:PartyId>ExampleParty2</eb:PartyId>
            <eb:Role>ExampleParty2Role</eb:Role>
          </eb:To>
        </eb:PartyInfo>
        <eb:CollaborationInfo>
          <eb:AgreementRef>DequeueMessageAgreementExample</eb:AgreementRef>
          <eb:Service>MarketMessaging</eb:Service>
          <eb:Action>DequeueMessage</eb:Action>
          <eb:ConversationId>2a81ffbd-0d3d-4cbd-8601-d916e0ed2fe2</eb:ConversationId>
        </eb:CollaborationInfo>
```

```
      </eb:UserMessage>
    </eb:Messaging>
  </soapenv:Header>
  <soapenv:Body>
    <urn:DequeueMessageRequest>
       <urn:DocumentReferenceNumber>cc3ae4a7-e93f-406a-99c8-
4bbc66ab5140</urn:DocumentReferenceNumber>
    </urn:DequeueMessageRequest>
  </soapenv:Body>
</soapenv:Envelope>
```

5.4.6. Techniczne kody błędów na poziomie warstwy transportowej

HTTP status	Kategoria	Znaczenie	Sugerowany sposób obsługi
500	Server	Błąd wewnętrzny systemu CSIRE	Ponowienie wywołania w późniejszym terminie. Kontakt z operatorem systemu w przypadku, gdyby problem nie ustąpił.
404	Client	Nieznana operacja	Sprawdzenie i poprawienie nazwy operacji przed ponowieniem wysyłki
408	Client	Timeout	Ponowienie wywołania w późniejszym terminie. Kontakt z operatorem systemu w przypadku, gdyby problem nie ustąpił.
401	Bezpieczeństwo	Odmowa dostępu	Odmowa dostępu — uwierzytelnianie użytkownika nie powiodło się lub nie zostało dostarczone w celu potwierdzenia tożsamości.
413	Client	Zbyt duża wiadomość	Proszę zweryfikować powod zbyt dużego rozmiaru wiadomości (np. zbyt wiele profili dobowych w ramach jednej wiadomości). Wiadomość powinna zostać podzielona na mniejsze części które powinny zostać wysłane ponownie.

Tabela 7 Techniczne kody błędów

5.4.7. Techniczne kody błędów AS4

Kanał AS4 zawsze zwraca błędy jako ebMS SignalMessages.

494

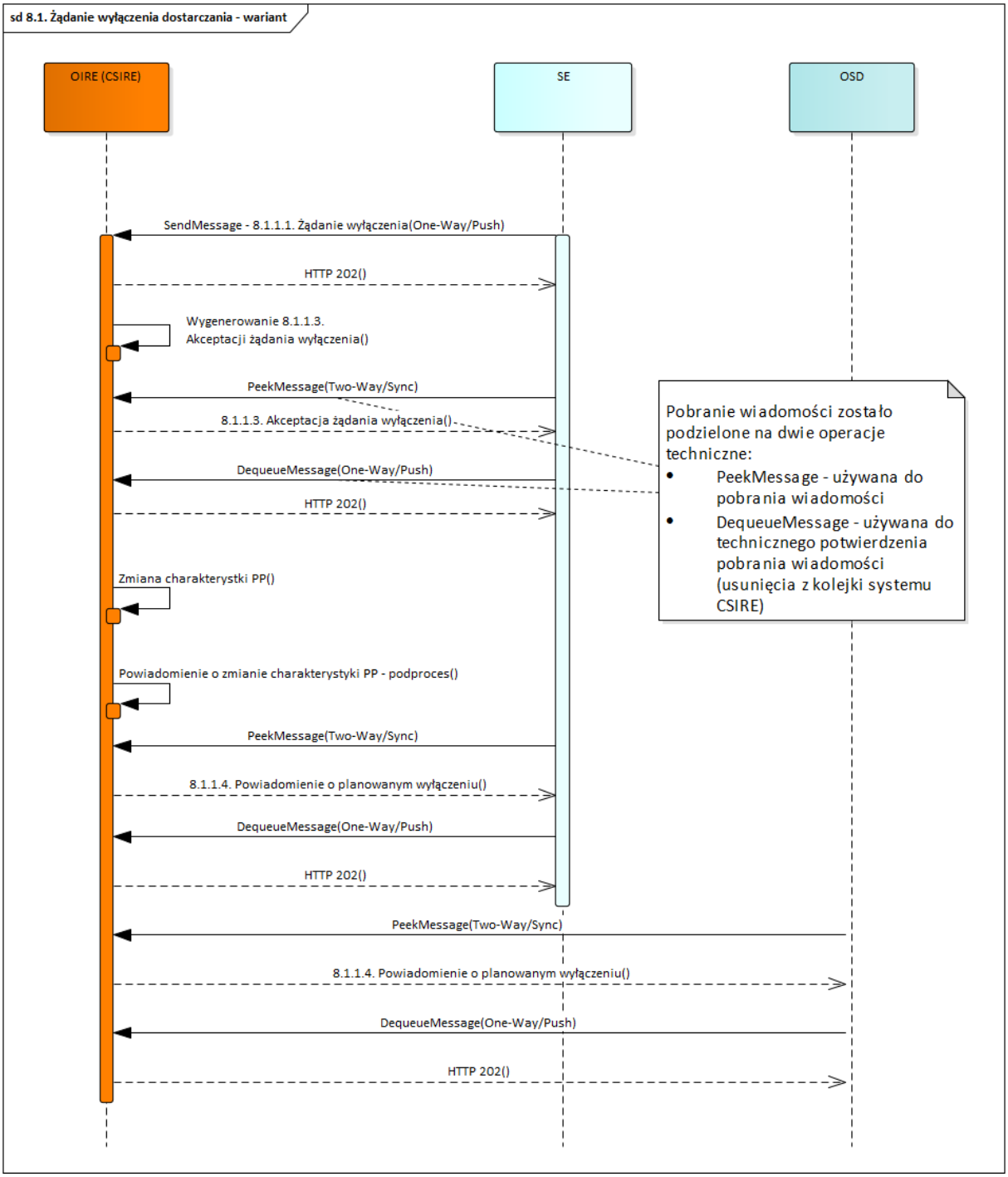
Kod błędu	Opis	Ważność	Znaczenie	Sugerowany sposób obsługi
EBMS:0001	Wartość nierozpoznana	Błąd	Dokument komunikatu jest dobrze sformułowany, a schemat prawidłowy, niemniej jednak jakiś element/atrybut zawiera wartość, której nie można rozpoznać i dlatego MSH nie może go użyć.	Popraw wiadomość i wyślij ponownie.
EBMS:0002	Funkcja nieobsługiwana	Ostrzeżenie	Chociaż dokument komunikatu jest prawidłowo sformułowany, a schemat prawidłowy, niektórych wartości elementu/atrybutu nie można przetworzyć zgodnie z oczekiwaniami, ponieważ powiązana funkcja nie jest obsługiwana przez MSH.	Usuń nieobsługiwane funkcje z wiadomości i wyślij poprawioną wiadomość.
EBMS:0003	Wartości niespójne	Błąd	Chociaż dokument komunikatu jest dobrze sformułowany, a schemat prawidłowy, wartość niektórych elementów/atrybutów jest niespójna albo z treścią innego elementu/atrybutu, albo z trybem przetwarzania MSH, albo z wymaganiami normatywnymi specyfikacji ebMS.	Sprawdź, czy poprawić komunikat lub zmienić konfigurację PMode. Po poprawieniu wyślij ponownie wiadomość.
EBMS:0004	Inny	Błąd		Sprawdź element ErrorDetail w Error, aby dowiedzieć się, co poszło nie tak. W przypadku, gdy payload nie jest prawidłowo sformułowany/schemat jest nieprawidłowy, payload musi zostać poprawiony przed próbą ponownego wysłania.

Kod błędu	Opis	Ważność	Znaczenie	Sugerowany sposób obsługi
EBMS:0005	Błąd połączenia	Błąd	MSH doświadcza tymczasowej lub trwałej awarii podczas próby otwarcia połączenia transportowego ze zdalnym MSH.	Odczekaj co najmniej 5 minut przed ponowną próbą. Spróbuj ponownie maksymalnie 3 razy, zanim skontaktujesz się z działem pomocy technicznej w celu uzyskania pomocy.
EBMS:0006	Pusty kanał partycji wiadomości	Ostrzeżenie	W kolejce wiadomości nie ma dostępnych wiadomości.	Ponów wywołanie po określonym czasie.
EBMS:0007	Niepoprawna wartość MIME	Błąd	Użycie MIME nie jest zgodne z wymaganym użyciem w tej specyfikacji.	Popraw załącznik i wyślij ponownie.
EBMS:0008	Funkcja nieobsługiwana	Błąd	Chociaż dokument komunikatu jest dobrze sformułowany, a schemat prawidłowy, obecność lub brak niektórych elementów/atrybutów nie jest zgodna z możliwościami MSH w odniesieniu do obsługiwanych funkcji.	Popraw wiadomość i wyślij ponownie.
EBMS:0009	Nieprawidłowy nagłówek	Błąd	Nagłówek ebMS jest albo źle sformułowany jako dokument XML, albo nie jest zgodny z regułami pakowania ebMS.	Popraw wiadomość i wyślij ponownie.
EBMS:0010	Niezgodność trybu przetwarzania	Błąd	Nagłówek ebMS lub inny nagłówek (np. niezawodność, bezpieczeństwo) oczekiwany przez MSH nie jest zgodny z oczekiwaną treścią na podstawie powiązanego trybu PMode.	Sprawdź, czy poprawić komunikat lub zmienić konfigurację PMode. Po poprawieniu wyślij ponownie wiadomość.
EBMS:0011	Błąd zewnętrznego payload	Błąd	MSH nie jest w stanie rozpoznać odniesienia do zewnętrznego payloadu (tj. części, która nie jest zawarta w komunikacie ebMS, identyfikowanym przez identyfikator URI PartInfo/href).	Popraw załącznik lub nagłówek SOAP w wiadomości i wyślij ponownie.

Kod błędu	Opis	Ważność	Znaczenie	Sugerowany sposób obsługi
EBMS:0101	Nieudane uwierzytelnianie	Błąd	Podpis w nagłówku Security przeznaczony dla aktora SOAP „ebms” nie mógł zostać zweryfikowany przez moduł Security.	Sprawdź, czy publiczny certyfikat skonfigurowany w CSIRE jest nadal poprawny. Jeśli nie, popraw certyfikat publiczny.
EBMS:0102	Nieudane odszyfrowywanie	Błąd	Zaszyfrowane dane odnoszące się do nagłówka Security przeznaczonego dla aktora SOAP „ebms” nie mogły zostać odszyfrowane przez moduł zabezpieczeń.	Sprawdź, czy wiadomość jest zaszyfrowana popranych kluczem.
EBMS:0103	Niezgodność z polityką bezpieczeństwa	Błąd	Metody zabezpieczeń, parametry, zakres lub inne wymagania lub umowy na poziomie polityki bezpieczeństwa nie zostały spełnione.	Popraw wiadomość i wyślij ponownie.

Tabela 8 Techniczne kody błędów AS4

5.4.8. Przykład realizacji początkowych kroków procesu SWI z mapowaniem na wywołania interfesju CSIRE



Rysunek 8 Diagram sekwencji wywołań systemu CSIRE dla początkowych kroków procesu 8.1. Żądanie wyłączenia dostarczania" dla "poprawnego" przebiegu.

Na powyższym diagramie przedstawiono sekwencję wywołań dla pierwszych kroków procesu „8.1. Żądanie wyłączenia dostarczania" z SWI w wersji 5.3 przy założeniu rozpoczęcia procesu przez SE/SEu i poprawnej komunikacji z systemem CSIRE (brak błędów technicznych i biznesowych).

- Pierwsze wywołanie rozpoczynające proces to wywołanie operacji SendMessage przez SE. Jako payload wiadomości przekazywany jest komunikat „8.1.1.1. Żądanie wyłączenia” zgodny z TSKB. Odpowiedź HTTP 202 oznacza przyjęcie wiadomości do procesowania.
- Po odebraniu wiadomości system CSIRE w ramach procesu 8.1 wygeneruje wiadomość „8.1.1.3. Akceptacja żądania wyłączenia” zgodną z TSKB. Ta wiadomość będzie czekać na pobranie przez SE, który uprzednio wywołał operację SendMessage.
- SE z użyciem operacji PeekMessage pobiera wiadomość „8.1.1.3. Akceptacja żądania wyłączenia” a następnie potwierdza odebranie wywołując operacje DequeueMessage (odpowiedź HTTP 202 oznacza poprawne zdjęcie wiadomości z kolejki)
- System CSIRE po zmianie charakterystyki PP wygeneruje wiadomości „8.1.1.4. Powiadomienie o planowanym wyłączeniu” zgodne z TSKB do SE oraz odpowiedniego OSD.
- Zarówno SEr/SEu jak i OSD pobiorą wiadomość „8.1.1.4. Powiadomienie o planowanym wyłączeniu” z użyciem operacji PeekMessage oraz potwierdzą odebranie z użyciem operacji DequeueMessage.

6. BEZPIECZEŃSTWO

Rozdział ten opisuje zagadnienia konfiguracji zabezpieczeń dla wykorzystania Profilu AS4 zdefiniowanego w dokumencie „ENTSOG AS4 Profile 3.6” [EG-AS4-Profile], w sposób zgodny z wymaganiami określonymi dla ENTSOG AS4 ebHandler oraz uwzględniający bieżące rekomendacje obowiązujące w PSE w zakresie stosowania zabezpieczeń kryptograficznych. Wymienione niżej wymagania konfiguracji zabezpieczeń stanowią aktualizację treści sekcji 2.3.4 „Security” dokumentu „ENTSOG AS4 Profile 3.6” [EG-AS4-Profile].

6.1. Zabezpieczenie komunikacji w warstwie sieci

Dla zabezpieczenia komunikacji sieciowej pomiędzy partnerami zastosowanie mają zasady zawarte w rozdziale 2.3.4.1 „Network Layer Security” dokumentu „ENTSOG AS4 Profile 3.6” [EG-AS4-Profile].

Dodatkowo, statyczne adresy (lub statyczne zakresy adresów) ustalone i zakomunikowane zgodnie z tymi zasadami powinny być użyte do ograniczenia swobody przepływów wiadomości przychodzących lub wychodzących, za pomocą urządzeń brzegowych sieci typu „firewall” lub urządzeń terminujących połączenia TLS, tylko z zarejestrowanymi uprzednio partnerami.

6.2. Zabezpieczenie komunikacji w warstwie transportowej

W celu zapewnienia poufności przesyłanych informacji w warstwie transportowej, spełnione muszą być warunki opisane w rozdziale 2.3.4.2 „Transport Layer Security” dokumentu „ENTSOG AS4 Profile 3.6” [EG-AS4-Profile]. Zastosowanie mają zatem parametry opisane w rozdziale 2.2.6.1 „Transport Layer Security” tego dokumentu, z dodatkowymi zastrzeżeniami wymienionymi poniżej:

1. Wymagane jest użycie protokołu TLS w wersji 1.2 lub 1.3 (rekomendowana). Obsługa protokołów SSL 2.x, 3.x oraz TLS w wersjach 1.0, 1.1, 1.2 musi być wyłączona.
2. W przypadku użycia TLS w wersji 1.3 strony komunikacji muszą wspierać obsługę zestawów algorytmów kryptograficznych TLS_AES_128_GCM_SHA256, TLS_AES_256_GCM_SHA384, TLS_CHACHA20_POLY1305_SHA256.
3. W przypadku użycia TLS w wersji 1.2 strony komunikacji muszą wspierać obsługę zestawów algorytmów kryptograficznych ECDHE-ECDSA-AES128-GCM-SHA256, ECDHE-RSA-AES128-GCM-SHA256, ECDHE-ECDSA-AES256-GCM-SHA384, ECDHE-RSA-AES256-GCM-SHA384, ECDHE-ECDSA-CHACHA20-POLY1305, ECDHE-RSA-CHACHA20-POLY1305, DHE-RSA-AES128-GCM-SHA256, DHE-RSA-AES256-GCM-SHA384, DHE-RSA-CHACHA20-POLY1305
4. Obsługa zestawów algorytmów kryptograficznych innych, niż wymienione powyżej musi być wyłączona.
5. Obustronne uwierzytelnianie TLS musi być stosowane. W tym celu dopuszcza się wykorzystanie odpowiednich certyfikatów wydanych dla nazw DNS urządzeń występujących w podwójnej roli serwera i klienta TLS.
6. Certyfikaty wykorzystywane przez odrębne komponenty infrastruktury zapewniające obsługę komunikacji TLS muszą spełniać wszystkie warunki określone w punkcie 6.4 „Certyfikaty oraz Infrastruktura Klucza Publicznego (PKI)”.

6.3. Zabezpieczenie komunikacji w warstwie komunikatu

Lista wspieranych algorytmów podpisywania i szyfrowania wiadomości przedstawiona w poniższych rozdziałach zostanie rozszerzona w ramach prac projektowych i opisana w kolejnej aktualizacji niniejszego dokumentu.

6.3.1. Podpisywanie wiadomości

CSIRE umożliwia podpisywanie wiadomości zarówno w przychodzących (żądanie) jak i wychodzących (odpowiedź/powiadomienie) wiadomościach. Podpis konfigurowany jest za pomocą parametru PMode PMode[1].Security.X509.Sign (patrz także 5.3.1).

CSIRE wspiera następujące standardy i specyfikace w odniesieniu do WS-Security i podpisów XML:

- BasicSecurityProfile-v1.1
- XML-DSIG-V1.0 (prefiks DS)
- WSS-SOAP-Message-Security-V1.1.1 (prefiks WSSE)
- WSS-WSU-V1.0 (prefiks WSU)

6.3.2. Szyfrowanie wiadomości

CSIRE umożliwia szyfrowanie wiadomości XML zarówno w przychodzących (żądanie) jak i wychodzących (odpowiedź/powiadomienie) wiadomościach, przy czym można skonfigurować dla każdego kierunku, czy szyfrowanie XML powinno być zapewnione w wiadomościach czy nie:

Wiadomości wejściowe:

- brak konfiguracji dla szyfrowania dla wiadomości wejściowych .
- CSIRE sprawdza wiadomość, jeśli jakikolwiek element zawiera znacznik EncryptedData i wtedy odszyfrowuje wiadomość.

Wiadomości wyjściowe:

- CSIRE używa parametru PMode PMode[1].Security.X509.Encryption.Encrypt (patrz sekcja 5.3.1) do kontrolowania, czy wiadomości wychodzące mają być szyfrowane przy użyciu publicznego certyfikatu przechowywanego dla organizacji.

Parametry i opcje używane do szyfrowania wiadomości:

- Typ identyfikatora klucza: Metoda, za pomocą której certyfikat jest identyfikowany po stronie odbiorcy.

  CSIRE stosuje następujący typ: Binary security token

Binary security token direct reference: Certyfikat podpisujący jest konwertowany na BinarySecurityToken i wstawiany do nagłówka bezpieczeństwa. Odniesienie do binarnego tokenu bezpieczeństwa jest również wstawiane do wsse:SecurityReferenceToken. Oznacza to, że cały certyfikat podpisu jest przekazywany do odbiorcy.

- Algorytm szyfrowania klucza: Algorytm używany do transportu wiadomości klucz symetryczny. Wybór dostępny na liście jest kontrolowany przez WS-Security Framework.

Algorytmy szyfrowania klucza używane w CSIRE :

- http://www.w3.org/2001/04/xmlenc#rsa-1_5
- http://www.w3.org/2001/04/xmlenc#rsa-oaep-mgf1p

- Algorytm szyfrowania: Algorytm stosowany do szyfrowania payload przy użyciu klucza symetrycznego wiadomości.

CSIRE używa poniższego algorytmu:

- AES128 w CBC: http://www.w3.org/2001/04/xmlenc#aes128-cbc

W przyszłości planowana jest implementacja AES-GCM:

- http://www.w3.org/2009/xmlenc11#aes128-gcm

6.4. Certyfikaty oraz Infrastruktura Klucza Publicznego (PKI)

Dla certyfikatów cyfrowych wykorzystywanych do zabezpieczenia komunikacji w warstwie komunikatu oraz certyfikatów cyfrowych wykorzystywanych do zabezpieczenia komunikacji w warstwie transportowej, stosuje się zasady opisane w rozdziale 2.3.4.4 „Certificates and Public Key Infrastructure” dokumentu „ENTSOG AS4 Profile 3.6” [EG-AS4-Profile], z zastrzeżeniem poniższych wyjątków i dodatkowych warunków:

1. Wybór Urzędu Certyfikacji PKI wydającego certyfikaty nie podlega przeglądowi przez ENTSOG.
2. Certyfikaty przeznaczone do wykorzystania produkcyjnego muszą być wydane przez powszechnie zaufane Centrum Certyfikacji PKI, spełniające warunki dla kwalifikowanych podmiotów świadczących usługi zaufania, zgodnie z przepisami rozporządzenia eIDAS i zarejestrowane na liście zaufania opublikowanej w witrynie „EU Trust Services Dashboard” Komisji Europejskiej, lub posiadające pieczęć AICPA/CICA WebTrust.
3. Nie dopuszcza się stosowania tych samych certyfikatów w środowiskach produkcyjnych i środowiskach testowych, za wyjątkiem certyfikatów uwierzytelniania serwera TLS, wydanych dla wielu domen DNS lub dla domen z „dziką kartą”.
4. Informacje o statusie odwołania wykorzystywanych certyfikatów, muszą być udostępniane w sposób niezawodny pod dostępnym dla stron uczestniczących w komunikacji adresem wskazanych w atrybutach CDP (CRL Distribution Point) lub AIA OCSP certyfikatu pod rygorem odrzucenia weryfikowanych tymi certyfikatami połączeń lub wiadomości.

6.5. Wymiana Certyfikatu

Procedura manualna – użytkownik reprezentujący administratora biznesowego dla uczestnika rynku będzie mógł samodzielnie skonfigurować certyfikat z użyciem Portalu Użytkownika profesjonalnego.

7. KOMPRESJA

Payload komunikatów AS4 wysyłany w ramach SendMessage, będzie mógł być skompresowany, aby umożliwić wydajne przesyłanie danych. Analogicznie dane odbierane przez system zewnętrzny z użyciem PeekMessage również będą mogły być skompresowane.

Stosowanie kompresji musi być zgodne z opisem profilu AS4 (patrz sekcja 3.1 w “AS4 Profile of ebMS 3.0 Version 1.0 OASIS Standard” [AS4-Profile]).

Kompresować można tylko payload podany jako załącznik SOAP, kompresja wiadomości przekazana w ramach treści wiadomości SOAP jest niedozwolona. Skompresowany załącznik SOAP musi być zgodny ze specyfikacją protokołu SOAP z załącznikami „SOAP Messages with Attachments” [SOAPATTACH].

Wpieranym algorytmem kompresji jest GZIP („GZIP file format specification version 4.3” [RFC1952]) – dane muszą być skompresowane przed dodaniem jako załącznik SOAP, zaś typ skompresowanego załącznika musi być ustawiony jako „application/gzip”.

8. REKOMENDACJE DOTYCZĄCE IMPLEMENTACJI ROZWIĄZANIA

8.1. Wprowadzenie

Wiele z parametrów przetwarzania (P-Mode'ów) definiuje w sposób jednoznaczny techniczne ustawienia i wymagania dotyczące implementacji, niemniej jendak istnieją parametry które wymagają konfiguracji i muszą być zaimplementowane zgodnie z wytycznymi i wskazówkami biznesowymi opisanymi poniżej.

8.2. Identyfikacja stron

Jednym z podstawowych warunków poprawnej wymiany komunikatów pomiędzy stronami, w ramach opisanego w tym dokumencie profilu, jest możliwość jednoznacznej identyfikacji podmiotów uczestniczących w komunikacji. Wobec powyższego, obligatoryjnym warunkiem do zapewnienia poprawnej komunikacji jest stosowanie przez strony kodów EIC jako identyfikatorów stron komunikacji.

Kod EIC musi być używany w dwóch parametrach trybów przetwarzania komunikatów. Mowa tutaj o wartościach dla PMode.Initiator.Party, oraz PMode.Responder.Party.

Identyfikatory EIC stron komunikacji AS4 pozwalają na jednoznaczną identyfikację partnera komunikacyjnego.

Partnerem komunikacyjnym może być zarówno podmiot biorący bezpośrednio udział w wymianie komunikatów biznesowych, jak i podmiot zewnętrzny, świadczący usługi komunikacyjne B2B na rzecz innych podmiotów (Nadawca fizyczny).

W przypadku podmiotu biorącego bezpośrednio udział w wymianie komunikatów, wykorzystywany kod EIC będzie kodem partnera biznesowego.

Zaś w przypadku, gdy będziemy mieli do czynienia z podmiotem zewnętrznym, świadczącym usługi komunikacyjne w imieniu partnera biznesowego, wykorzystywany będzie kod EIC podmiotu zewnętrznego.

Poza kodem EIC przekazywanym w konfiguracji AS4 PMode oraz nagłówkami komunikatów AS4, do identyfikacji stron wymagane są dodatkowe kroki:

- Tożsamość systemu musi zostać utworzona w CSIRE dla każdej Organizacji.
- Tożsamość systemu wymaga rejestracji certyfikatu klienta, który należy również dostarczyć przy każdym żądaniu do CSIRE (wzajemny TLS), patrz także sekcja 6.4.
- Dla każdej Organizacji należy utworzyć w systemie Użytkownika Organizacji z unikalną nazwą użytkownika.
- Aby korzystać z kanału CSIRE AS4, Użytkownik Organizacji musi posiadać uprawnienia do Funkcji Systemu SendMessage, PeekMessage i DequeueMessage (patrz także punkt 5.4).

8.3. Dostarczenie wiadomości, powtórzenia, obsługa niedostępności

Systemy zewnętrzne komunikujące się z CSIRE powinny zapewnić, by żadna wiadomość nie została niedostarczona. W przypadku wystąpienia problemu komunikacyjnego podczas pierwszej próby, należy wymusić po stronie wysyłającego implementację ponownej wysyłki wiadomości.

Jednocześnie należy dopilnować, by żaden system zewnętrzny nie wygenerował zbyt dużego ruchu sieciowego, poprzez nieustanne podejmowane próby ponownego wysłania wiadomości,

która nie może być z powodów technicznych dostarczona (patrz kody błędów opisane w 5.4.6 i 5.4.7).

Rekomenduje się, by parametr dotyczący maksymalnej ilości powtórzeń (ang. *max retries*) był ustawiony na wartość nie mniejszą niż 2 i nie większą niż 5.

Jednocześnie okres, po którym podjęta zostanie kolejna próba dostarczenia wiadomości (ang. *retry period*), nie powinien być mniejszy niż 5000 millisekund.

Dodatkowym zaleceniem dla systemów zewnętrznych jest zwiększanie tego okresu po każdej ponowionej próbie.

W wypadku problemów w komunikacji, których nie można obsłużyć za pomocą powyżej opisanych mechanizmów, wykorzystywane są metody opisane w IRiESP-OIRE.

Systemy zewnętrzne powinny mieć możliwość kolejkowania wiadomości, których nie udało się dostarczyć do CSIRE (np. z powodu niedostępności) tak by możliwe było ponowne ich wysłanie po ustąpieniu niedostępności.

Kolejkowanie wiadomości powinno być zrealizowane w taki sposób aby zapewnić persystencje wiadomości, odporność na awarie (wyłączenie) oraz możliwość ponowienia zgodnie z oryginalną kolejnością.

System informacyjny podmiotu zewnętrznego powinien posiadać funkcjonalność ręcznego (tj. inicjowanego przez jego użytkownika) oraz automatycznego (tj. realizowanego wg. zdefiniowanych reguł) wznowienia wysyłania komunikatów po przywróceniu komunikacji z CSIRE.

8.4. Wymagania odnośnie środowisk systemów współpracujących z CSIRE

Każdy podmiot, który zamierza korzystać z systemu informacyjnego współdziałającego z CSIRE, musi dysponować środowiskiem produkcyjnym.

Oraz środowiskami nieprodukcyjnymi:

- certyfikacyjnym,
- pilotażowym.

Muszą być one oddzielone od środowiska produkcyjnego. Służą przetestowaniu współpracy systemów oraz zapewnienia kompatybilności.

Środowisko nieprodukcyjne powinno odzwierciedlać środowisko produkcyjne w zakresie architektury oraz wersji komponentów.

W środowisku nieprodukcyjnym powinny obowiązywać identyczne zasady zarządzania dostępem jak w środowisku produkcyjnym.

OIRE przewiduje weryfikację i przyłączenie do CSIRE co najwyżej jednego środowiska certyfikacyjnego, jednego środowiska pilotażowego oraz jednego środowiska produkcyjnego dla każdego Kontrahenta.

Środowisko certyfikacyjne musi być przygotowane do korzystania ze sztucznie wygenerowanych danych certyfikacyjnych (testowych).

Środowisko pilotażowe musi być przygotowane do korzytsania z danych sztucznie wygenerowanych (testowych), zanonimizowanych danych odpowiadających danym produkcyjnym lub danych produkcyjnych.

9. REKOMENDACJE W ZAKRESIE CERTYFIKACJI AS4

W celu ograniczenia ryzyk związanych z integracją systemów Użytkowników profesjonalnych oraz Użytkowników uprawnionych z systemem CSIRE, rekomendujemy wykorzystanie implementacji AS4, które przeszły testy interoperacyjności wykonywane m. in. przez Drummond Group.

Aktualna lista zweryfikowanych rozwiązań znajduje się w: https://www.drummondgroup.com/certified-products-2/b2b-interoperability/#appst

10. WEBSERVICE AS4 - WSDL

```
<?xml version="1.0" encoding="UTF-8"?>
<wsdl:definitions
  xmlns:ns1="urn:cms:b2b:v01"
  xmlns:tns="urn:cms:b2b:service:v01"
  xmlns:wsdl="http://schemas.xmlsoap.org/wsdl/"
  xmlns:soap="http://schemas.xmlsoap.org/wsdl/soap12/"
  targetNamespace="urn:cms:b2b:service:v01">
  <wsdl:types>
    <xs:schema xmlns:xs="http://www.w3.org/2001/XMLSchema" xmlns:b2b="urn:cms:b2b:v01"
      targetNamespace="urn:cms:b2b:v01" elementFormDefault="qualified"
      attributeFormDefault="unqualified" version="2.0.0.1">
      <xs:complexType name="DequeueMessageRequest_Type">
        <xs:sequence>
          <xs:element name="DocumentReferenceNumber" type="b2b:DocumentReferenceNumber_Type"
/>
        </xs:sequence>
      </xs:complexType>
      <xs:complexType name="DequeueMessageResponse_Type" />
      <xs:complexType name="MessageContainer_Type">
        <xs:sequence>
          <xs:element name="Payload" type="b2b:Payload_Type" />
        </xs:sequence>
      </xs:complexType>
      <xs:complexType name="ResponseMessageContainer_Type">
        <xs:sequence>
          <xs:element name="DocumentReferenceNumber" type="b2b:DocumentReferenceNumber_Type"
            minOccurs="0" />
          <xs:element name="Payload" type="b2b:ResponsePayload_Type" />
        </xs:sequence>
      </xs:complexType>
      <xs:complexType name="Payload_Type">
        <xs:sequence>
          <xs:any processContents="skip" namespace="##any" />
        </xs:sequence>
      </xs:complexType>
      <xs:complexType name="ResponsePayload_Type">
        <xs:sequence>
          <xs:any processContents="skip" namespace="##any" />
        </xs:sequence>
      </xs:complexType>
      <xs:complexType name="MessageDomains_Type">
        <xs:sequence>
          <xs:element name="MessageDomain" type="b2b:MessageDomain_Type" minOccurs="1"
            maxOccurs="unbounded" />
        </xs:sequence>
      </xs:complexType>
      <xs:complexType name="PeekMessageRequest_Type">
        <xs:sequence>
          <xs:element name="MessageDomains" type="b2b:MessageDomains_Type" minOccurs="0" />
        </xs:sequence>
      </xs:complexType>
      <xs:complexType name="PeekMessageResponse_Type">
        <xs:sequence>
          <xs:element name="MessageContainer" type="b2b:ResponseMessageContainer_Type"
minOccurs="0" />
        </xs:sequence>
      </xs:complexType>
      <xs:complexType name="SendMessageRequest_Type">
        <xs:sequence>
          <xs:element name="MessageContainer" type="b2b:MessageContainer_Type" />
        </xs:sequence>
      </xs:complexType>
      <xs:complexType name="SendMessageResponse_Type">
        <xs:sequence>
          <xs:element name="DocumentReferenceNumber" type="b2b:DocumentReferenceNumber_Type"
/>
        </xs:sequence>
      </xs:complexType>
      <xs:element name="DequeueMessageRequest" type="b2b:DequeueMessageRequest_Type" />
      <xs:element name="DequeueMessageResponse" type="b2b:DequeueMessageResponse_Type" />
      <xs:element name="PeekMessageRequest" type="b2b:PeekMessageRequest_Type" />
      <xs:element name="PeekMessageResponse" type="b2b:PeekMessageResponse_Type" />
      <xs:element name="SendMessageRequest" type="b2b:SendMessageRequest_Type" />
      <xs:element name="SendMessageResponse" type="b2b:SendMessageResponse_Type" />
```

```
      <xs:simpleType name="DocumentReferenceNumber_Type">
        <xs:restriction base="xs:string">
          <xs:maxLength value="36" />
        </xs:restriction>
      </xs:simpleType>
      <xs:simpleType name="MessageDomain_Type">
        <xs:restriction base="xs:string">
        </xs:restriction>
      </xs:simpleType>
    </xs:schema>
  </wsdl:types>
  <wsdl:message name="SendMessageRequest">
    <wsdl:part name="parameters" element="ns1:SendMessageRequest" />
  </wsdl:message>
  <wsdl:message name="SendMessageResponse">
    <wsdl:part name="parameters" element="ns1:SendMessageResponse" />
  </wsdl:message>
  <wsdl:message name="PeekMessageRequest">
    <wsdl:part name="parameters" element="ns1:PeekMessageRequest" />
  </wsdl:message>
  <wsdl:message name="PeekMessageResponse">
    <wsdl:part name="parameters" element="ns1:PeekMessageResponse" />
  </wsdl:message>
  <wsdl:message name="DequeueMessageRequest">
    <wsdl:part name="parameters" element="ns1:DequeueMessageRequest" />
  </wsdl:message>
  <wsdl:message name="DequeueMessageResponse">
    <wsdl:part name="parameters" element="ns1:DequeueMessageResponse" />
  </wsdl:message>
  <wsdl:portType name="marketMessagingB2BInboundServiceV01PortType">
    <wsdl:operation name="sendMessage">
      <wsdl:input message="tns:SendMessageRequest" />
      <wsdl:output message="tns:SendMessageResponse" />
    </wsdl:operation>
    <wsdl:operation name="peekMessage">
      <wsdl:input message="tns:PeekMessageRequest" />
      <wsdl:output message="tns:PeekMessageResponse" />
    </wsdl:operation>
    <wsdl:operation name="dequeueMessage">
      <wsdl:input message="tns:DequeueMessageRequest" />
      <wsdl:output message="tns:DequeueMessageResponse" />
    </wsdl:operation>
  </wsdl:portType>
  <wsdl:binding name="marketMessagingB2BInboundServiceV01HTTPEndpointBinding"
    type="tns:marketMessagingB2BInboundServiceV01PortType">
    <soap:binding style="document" transport="http://schemas.xmlsoap.org/soap/http" />
    <wsdl:operation name="sendMessage">
      <soap:operation soapAction="sendMessage" style="document" />
      <wsdl:input>
        <soap:body parts="parameters" use="literal" />
      </wsdl:input>
      <wsdl:output>
        <soap:body parts="parameters" use="literal" />
      </wsdl:output>
    </wsdl:operation>
    <wsdl:operation name="peekMessage">
      <soap:operation soapAction="peekMessage" style="document" />
      <wsdl:input>
        <soap:body parts="parameters" use="literal" />
      </wsdl:input>
      <wsdl:output>
        <soap:body parts="parameters" use="literal" />
      </wsdl:output>
    </wsdl:operation>
    <wsdl:operation name="dequeueMessage">
      <soap:operation soapAction="dequeueMessage" style="document" />
      <wsdl:input>
        <soap:body parts="parameters" use="literal" />
      </wsdl:input>
      <wsdl:output>
        <soap:body parts="parameters" use="literal" />
      </wsdl:output>
    </wsdl:operation>
  </wsdl:binding>
  <wsdl:service name="marketMessagingB2BInboundServiceV01">
    <wsdl:port name="marketMessagingB2BInboundServiceV01HTTPEndpoint"
      binding="tns:marketMessagingB2BInboundServiceV01HTTPEndpointBinding">
      <soap:address
```

```
            location="https://localhost:1234/soap/PSE?organisationUser=MyOrganisationB2BUser" />
        </wsdl:port>
    </wsdl:service>
</wsdl:definitions>
```

11. SPIS TABEL I RYSUNKÓW

12. ODNIESIENIA

Nazwa	Źródło
[AS4-Profile]	AS4 Profile of ebMS 3.0 Version 1.0 OASIS Standard 23 January 2013 http://docs.oasis-open.org/ebxml-msg/ebms/v3.0/profiles/AS4-profile/v1.0/os/AS4-profile-v1.0-os.html
[ebMS3CORE]	OASIS ebXML Messaging Services Version 3.0: Part 1, Core Features OASIS Standard 1 October 2007 http://docs.oasis-open.org/ebxml-msg/ebms/v3.0/core/ebms_core-3.0-spec.html
[BDX-AS4-v1.0]	AS4 Interoperability Profile for Four-Corner Networks Version 1.0 Committee Specification 01 12 November 2021 https://docs.oasis-open.org/bdxr/bdx-as4/v1.0/cs01/bdx-as4-v1.0-cs01.html
[EG-AS4-Profile]	ENTSOG AS4 Profile Version 3.6 – 2018-03-27 https://www.entsog.eu/sites/default/files/2019-05/INT0488-161115%20AS4%20Usage%20Profile_Rev_3.6_clean_final.pdf
[SOAP12]	SOAP Version 1.2 Part 1: Messaging Framework (Second Edition) W3C Recommendation 27 April 2007 https://www.w3.org/TR/soap12/
[SOAPATTACH]	SOAP Messages with Attachments: W3C Note 11 December 2000 https://www.w3.org/TR/SOAP-attachments/
[XMLDSIG]	XML-Signature Syntax and Processing (Second Edition). W3C Recommendation. 10 June 2008. http://www.w3.org/TR/xmldsig-core/
[WSS10]	Web Services Security: SOAP Message Security 1.0, 2004 http://docs.oasis-open.org/wss/2004/01/oasis-200401-wss-soap-message-security-1.0.pdf
[WSS11]	Web Services Security: SOAP Message Security 1.1. OASIS Standard incorporating Approved Errata. 1 November 2006 http://docs.oasis-open.org/wss/v1.1/wss-v1.1-spec-errata-os-SOAPMessageSecurity.pdf

Tabela 9 Odniesienia