
# PolishAPI

## Specyfikacja interfejsu na potrzeby usług świadczonych przez strony trzecie w oparciu o dostęp do rachunków płatniczych

*Dokument opracowany przez Grupę Projektową ds. PolishAPI*

9 lipca 2018
**Wersja 2.0**

**Licencja**

# Spis treści

1 Wstęp ..... 7
1.1 Kontekst ..... 7
1.2 Struktura dokumentu ..... 8
1.3 Misja Standardu PolishAPI ..... 8
1.4 Główne założenia ..... 9
1.4.1 Aktorzy w procesach definiowanych w standardzie PolishAPI ..... 9
1.4.2 Wymagania dot. aktorów w procesach definiowanych w standardzie PolishAPI ..... 10
1.4.3 Mechanizmy uwierzytelniania PSU ..... 11
1.4.4 Zarządzanie zgodami PSU na wykonywanie usług przez TPP ..... 13
1.4.5 Zastosowanie mechanizmu silnego uwierzytelnienia (SCA) ..... 21
1.4.6 Realizacja usług w zakresie Zgodności ..... 21
1.4.7 Realizacja usług w zakresie Premium ..... 21
1.5 Rozwój standardu PolishAPI ..... 21
2 Słownik pojęć użytych w dokumencie ..... 22
3 Definicja biznesowa usług z zakresu Zgodności ..... 24
3.1 Definicja biznesowa zakresu Zgodności dla usługi PIS ..... 24
3.1.1 Rodzaje transakcji w zakresie Zgodności ..... 24
3.1.2 Informacja o statusie transakcji ..... 24
3.1.3 Definicja rachunku płatniczego ..... 25
3.1.4 Lista pól wymaganych przez ASPSP w zakresie Zgodności ..... 25
3.1.5 Diagram zapytania w ramach usługi PIS w zakresie Zgodności ..... 28
3.1.6 Autoryzacja transakcji płatniczej zainicjowanej za pomocą usługi PIS ..... 28
3.2 Definicja biznesowa zakresu Zgodności dla usługi AIS ..... 29
3.2.1 Definicja rachunku płatniczego ..... 29
3.2.2 Częstotliwość zapytań w zakresie Zgodności ..... 29
3.2.3 Zakres informacji dot. historii rachunku płatniczego w zakresie Zgodności ..... 29
3.2.4 Lista pól udostępnianych przez ASPSP w zakresie Zgodności ..... 30
3.2.5 Diagramy zapytań w ramach usługi AIS w zakresie Zgodności ..... 32
3.3 Definicja biznesowa zakresu Zgodności dla usługi CAF ..... 33
3.3.1 Lista pól wymaganych przez ASPSP w zakresie Zgodności ..... 33
3.3.2 Diagram zapytania w ramach usługi CAF w zakresie Zgodności ..... 33
4 Przykładowe przypadki użycia ..... 34
4.1 Przypadek Użycia #1: inicjacja pojedynczej płatności przez PISP (PIS) ..... 34
4.1.1 Inicjacja pojedynczej płatności przez PISP z wykorzystaniem mechanizmu uwierzytelniania po stronie ASPSP ..... 34
4.1.2 Inicjacja pojedynczej płatności przez PISP z wykorzystaniem mechanizmu uwierzytelniania po stronie ASPSP z wyborem rachunku po stronie ASPSP ..... 34
4.1.3 Inicjacja pojedynczej płatności przez PISP z wykorzystaniem mechanizmu uwierzytelniania w zewnętrznym narzędziu autoryzacyjnym ..... 35
4.2 Przypadek Użycia #2: wyświetlenie informacji o rachunku płatniczym przez AISP (AIS) 35
4.3 Przypadek użycia #3: zapytanie o dostępność środków przez PIISP (CAF) ..... 36

5 Specyfikacja techniczna PolishAPI ..... 38
5.1 Założenia techniczne ..... 38
5.2 Nawiązanie sesji XS2A ..... 39
5.3 Definicja tokena dostępu ..... 40
5.4 Wzajemne uwierzytelnienie TPP i ASPSP ..... 40
5.5 Protokół komunikacyjny ..... 41
5.6 Schemat nazewnictwa zasobów ..... 41
5.7 Kanoniczny model danych ..... 41
5.8 Operacje ..... 44
5.9 Sortowanie ..... 44
5.10 Filtrowanie ..... 44
5.11 Stronicowanie ..... 44
5.12 Statusy odpowiedzi ..... 44
5.13 Nagłówki HTTP ..... 45
5.14 Format wiadomości ..... 46
5.15 Podstawowe formaty danych ..... 46
6 Bezpieczeństwo informacji ..... 47
6.1 Uwierzytelnienie TPP ..... 47
6.2 Autoryzacja TPP ..... 47
6.3 Autoryzacja PSU dla operacji wykonywanych przez TPP ..... 47
6.4 Bezpieczeństwo w przypadku aplikacji mobilnych ..... 47
6.5 Walidacja i zapewnienie integralności danych ..... 48
6.6 Kryptografia ..... 48
6.7 Ochrona przed nadużyciami API ..... 49
6.8 Logowanie informacji audytowych ..... 49
7 Opis techniczny procesu uwierzytelniania i autoryzacji ..... 50
7.1 Parametry scope oraz scope_details ..... 50
7.2 Mechanizm uwierzytelniania po stronie ASPSP ..... 51
7.2.1 Przekierowanie z TPP do ASPSP ..... 51
7.2.2 Uwierzytelnienie PSU i autoryzacja ..... 51
7.2.3 Zwrotne przekierowanie przeglądarki PSU do TPP ..... 51
7.2.4 Pobranie tokenu na podstawie Auhtorization Code ..... 52
7.2.5 Wycofanie zgody ..... 52
7.2.6 Stosowanie struktury scope_details ..... 52
7.3 Mechanizm uwierzytelniania w zewnętrznym narzędziu autoryzacyjnym (*decoupled*) ..... 53
7.4 Pobranie *access tokena* na podstawie *refresh tokena* ..... 55
7.5 Pobranie nowego *access tokena* na podstawie *exchange tokena* ..... 55
8 Opis techniczny usługi PIS ..... 57
8.1 Diagram aktywności w usłudze PIS ..... 57

8.2 Struktura zapytań interfejsu XS2A ..... 57
8.3 Struktura zapytań interfejsu wywołań zwrotnych - CallBack ..... 57
9 Opis techniczny usługi AIS ..... 59
9.1 Diagram aktywności w usłudze AIS ..... 59
9.2 Struktura zapytań interfejsu XS2A ..... 59
9.3 Struktura zapytań interfejsu wywołań zwrotnych - CallBack ..... 60
10 Opis techniczny usług CAF ..... 61
10.1 Diagram aktywności w usłudze CAF ..... 61
10.2 Struktura zapytania interfejsu XS2A(w tym opis pól i wymagalność) ..... 61
11 Utylizacja metod interfejsu XS2A oraz usług autoryzacyjnych – diagramy sekwencji ..... 62
11.1 Nawiązywanie sesji XS2A z uwierzytelnieniem PSU metodą *redirection* ..... 63
11.2 Nawiązywanie sesji XS2A z uwierzytelnieniem PSU metodą *decoupled* ..... 65
11.3 Nawiązywanie sesji XS2A z uwierzytelnieniem PSU metodą *refresh token* ..... 67
11.4 Nawiązywanie sesji XS2A z uwierzytelnieniem PSU metodą *exchange token* ..... 67
11.5 Wywołanie metod interfejsu XS2A z użyciem sesji ..... 69
11.6 Wywołanie metod interfejsu XS2A bez użycia sesji ..... 72
12 Kody błędów ..... 74
12.1 Kody błędów dla kodu odpowiedzi HTTP 403 ..... 75
13 Rekomendacje implementacji standardu ..... 76
13.1 Obsługa przekroczenia maksymalnego dozwolonego czasu (*timeout*) ..... 76
13.2 Weryfikacja TPP ..... 76
13.3 Serwer Autoryzacji ..... 76
13.4 Antyfraud ..... 76
14 Spis Załączników ..... 77

# Spis ilustracji

Ilustracja 1: Ogólny schemat komunikacji w Standardzie PolishAPI ..... 9
Ilustracja 2: Ogólny schemat zależności pomiędzy aktorami w Standardzie PolishAPI ..... 10
Ilustracja 3: Uwierzytelnianie w zewnętrznym narzędziu autoryzacyjnym ..... 12
Ilustracja 4: PIS. Udzielenie zgody (uwierzytelnianie po stronie ASPSP) ..... 14
Ilustracja 5: PIS. Udzielenie zgody (uwierzytelnianie po stronie ASPSP), wybór rachunku po stronie ASPSP ..... 15
Ilustracja 6: PIS. Udzielenie zgody (uwierzytelnianie w zewnętrznym narzędziu autoryzacyjnym) 16
Ilustracja 7: AIS. Udzielenie zgody z ręcznym wprowadzeniem numeru rachunku (uwierzytelnienie po stronie ASPSP) ..... 17
Ilustracja 8: AIS. Udzielenie zgody z ręcznym wprowadzeniem numeru rachunku (uwierzytelnianie w zewnętrznym narzędziu autoryzacyjnym) ..... 18
Ilustracja 9: AIS. Udzielenie zgody z pobraniem listy rachunków (uwierzytelnianie po stronie ASPSP) 19
Ilustracja 10: AIS. Udzielenie zgody z pobraniem listy rachunków (uwierzytelnianie w zewnętrznym narzędziu autoryzacyjnym) ..... 20
Ilustracja 11: AIS. Cofnięcie zgody ..... 20
Ilustracja 12: Diagram statusów płatności ..... 25
Ilustracja 13: PIS / Udzielenie zgody i realizacja inicjacji płatności oraz pobranie statusu płatności 34
Ilustracja 14: PIS / Udzielenie zgody i realizacja inicjacji płatności (wybór rachunku po stronie ASPSP) oraz pobranie statusu płatności ..... 35
Ilustracja 15: PIS / Udzielenie zgody i realizacja inicjacji płatności (uwierzytelnienie w zewnętrznym narzędziu autoryzacyjnym) oraz pobranie statusu płatności ..... 35
Ilustracja 16: AIS / Wyświetlenie informacji o rachunku płatniczym przez AISP ..... 36
Ilustracja 17: Ilustracja 16: CAF / Zapytanie o dostępność środków ..... 37
Ilustracja 18: Wysokopoziomowy diagram nawiązywania sesji XS2A ..... 39
Ilustracja 19: Mechanizm uwierzytelniania po stronie ASPSP ..... 51
Ilustracja 20: Wysokopoziomowy diagram aktywności w usłudze PIS ..... 57
Ilustracja 21: Wysokopoziomowy diagram aktywności w usłudze AIS ..... 59
Ilustracja 22: Wysokopoziomowy diagram aktywności w usłudze CAF ..... 61

# 1 Wstęp

## 1.1 Kontekst

Wdrożenie przez Unię Europejską nowej dyrektywy w sprawie usług płatniczych w ramach rynku wewnętrznego (PSD2) wprowadza możliwość oferowania nowych produktów i usług związanych nie tylko z rynkiem usług płatniczych, ale także szerzej rozumianym rynkiem usług finansowych. Zarówno podmioty będące obecne na tym rynku, takie jak banki, Spółdzielcze Kasy Oszczędnościowo-Kredytowe czy oddziały zagranicznych instytucji kredytowych, ale także nowe rodzaje podmiotów (dostawcy będący stroną trzecią, Third Party Providers, TPP) będą mogły wykorzystać możliwość oferowania nowych usług budowanych w oparciu o PSD2, akty wykonawcze (w tym Regulacyjne Standardy Techniczne – RTS) i akty prawa krajowego. Nowymi kategoriami usług są:

a) ***Account Information Service*** **(AIS)** – usługa dostępu do informacji o rachunku, zdefiniowana w art. 67 PSD2
b) ***Payment Initiation Service*** **(PIS)** – usługa inicjowania transakcji płatniczej, zdefiniowana w art. 66 PSD2
c) ***Confirmation of the Availability of Funds*** **(CAF)** – usługa potwierdzania dostępności na rachunku płatniczym płatnika kwoty niezbędnej do wykonania transakcji płatniczej, zdefiniowana w art. 65 PSD2

Umożliwienie realizacji powyższych usług przez podmioty do tego uprawnione wymaga przygotowania przez dostawców prowadzących rachunek (ASPSP) dedykowanych interfejsów umożliwiających dostęp do rachunków płatniczych (interfejs XS2A) przez uprawione do tego strony trzecie (TPP), opartych o otwarte API.

Banki oraz inne podmioty współpracujące w ramach Związku Banków Polskich podjęły decyzję o stworzeniu wspólnego, uniwersalnego standardu API, wykorzystującego dotychczasowe osiągnięcia polskiego sektora bankowego i płatniczego, najlepsze praktyki i doświadczenia, w tym z zagranicznych standardów API, oraz istniejące już interfejsy w ramach infrastruktury międzybankowej. Standard ten będzie mógł być wdrożony przez banki oraz inne ASPSP, zgodne z niezależnie podjętymi decyzjami biznesowymi. W ramach prac grup biznesowej, IT i bezpieczeństwa oraz prawnej, powstały założenia, a następnie opis standardu, przedstawiony w niniejszym dokumencie.

Jako podstawę do niniejszej wersji standardu przyjęto Rozporządzenie Delegowane w odniesieniu do regulacyjnych standardów technicznych dotyczących silnego uwierzytelniania klienta i wspólnych i bezpiecznych otwartych standardów komunikacji (RTS), opublikowane w Dzienniku Urzędowym Unii Europejskiej w dniu 13 marca 2018 roku (https://eur-lex.europa.eu/legal-content/PL/TXT/?uri=uriserv:OJ.L_.2018.069.01.0023.01.POL&toc=OJ:L:2018:069:TOC).

W tworzeniu niniejszego standardu brały udział następujące podmioty (wymienione w kolejności alfabetycznej):

1) Allegro Group
2) Biuro Informacji Kredytowej S.A.
3) Billbird S.A.
4) Blue Media S.A.
5) Diners Club Polska
6) Krajowa Izba Rozliczeniowa S.A.
7) Kontomierz.pl Sp. z o.o.
8) Krajowa Kasa Oszczędnościowo – Kredytowa
9) Krajowy Związek Banków Spółdzielczych
10) PayU S.A.
11) Polska Izba Informatyki i Telekomunikacji

12) Polska Izba Ubezpieczeń
13) Polski Standard Płatności Sp. z o.o.
14) Polska Organizacja Niebankowych Instytucji Płatności
15) Skycash Poland S.A.
16) Spółdzielcza Kasa Oszczędnościowo – Kredytowa im. F. Stefczyka
17) Związek Banków Polskich wraz ze stowarzyszonymi bankami[1]

Projekt specyfikacji był poddany konsultacjom publicznym (w dn. 17-31 stycznia 2018), w wyniku których 21 podmiotów polskich i zagranicznych przekazało ok. 300 uwag i komentarzy, częściowo uwzględnionych w niniejszym dokumencie.

## 1.2 Struktura dokumentu

Dokument składa się z dwóch zasadniczych części oraz załączników:

a) Części dotyczącej charakterystyki biznesowej Standardu PolishAPI (rozdziały 1 – 4)
b) Części dotyczącej rozwiązań technologicznych przyjętych w standardzie PolishAPI (rozdziały 5 – 13)
c) Załączników, których spis znajduje się w rozdziale 14

## 1.3 Misja Standardu PolishAPI

Głównym celem dokumentu jest zdefiniowanie interfejsów dla usług opisanych w PSD2 oraz powiązanych aktach prawnych, w zakresie interakcji pomiędzy ASPSP a TPP podczas realizacji usług AIS, PIS oraz CAF. Wymóg pojawienia się otwartych API daje również szansę, aby w ramach jednego standardu ASPSP oraz TPP miały możliwość zaoferowania nie tylko usług wymaganych przepisami prawa, ale także dodatkowych usług, które swoim zakresem wykraczają poza ramy wyznaczone przez prawodawcę. Możemy zatem wyróżnić następujące zakresy usług w ramach standardu PolishAPI:

a) **Zakres Zgodności** w ramach usług AIS, PIS i CAF - usługi wymagane przez PSD2,
b) **Zakres Premium** w ramach usług AIS, PIS i CAF - usługi wykraczające poza wymogi PSD2, poza zakresem niniejszego dokumentu.

Każdy ASPSP i TPP może skorzystać ze standardu PolishAPI jak z otwartego standardu. Korzystanie ze standardu nie jest obowiązkowe. Każdy z podmiotów działających na rynku w oparciu o dyrektywę PSD2, może stosować dowolne rozwiązanie, zgodne z PSD2 oraz powiązanymi aktami prawnymi.

Interakcje występujące pomiędzy TPP a PSU oraz ASPSP oraz PSU, a także zagadnienia związane z procesami wpisów do rejestru krajowego TPP, udzielania zezwolenia na działalność TPP w zakresie usług związanych z PSD2 przez organy administracji publicznej nie mieszczą się w zakresie niniejszego dokumentu.

Część zagadnień, pozostających w zakresie specyfikacji standardu, będzie do niego systematycznie dodawana w miarę prowadzenia prac projektowych i uzgodnieniowych (w tym konsultacji publicznych). Powyższe zastrzeżenie odnosi się m.in. do zagadnień związanych ze specyficznymi funkcjonalnościami dla rachunków korporacyjnych.

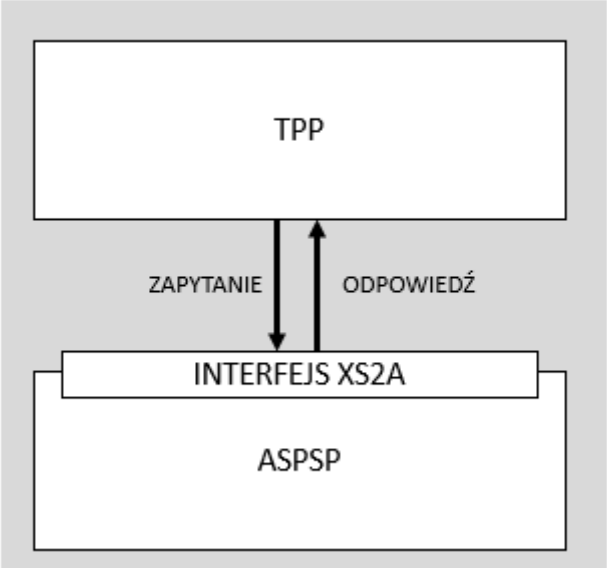
[1] Alior Bank S.A., Bank BGŻ BNP Paribas S.A., Bank Handlowy w Warszawie S.A., Bank Millennium S.A., Bank Pekao S.A., Bank Pocztowy S.A., Bank Polskiej Spółdzielczości S.A., Bank Zachodni WBK S.A., Credit Agricole Bank Polska S.A., Deutsche Bank Polska S.A., DNB Bank Polska S.A., Eurobank S.A., Getin Noble Bank S.A., Idea Bank S.A., ING Bank Śląski S.A., mBank S.A., Nest Bank S.A., PKO Bank Polski S.A., Raiffeisen Bank Polska S.A., SGB-Bank S.A.

## 1.4 Główne założenia

### 1.4.1 Aktorzy w procesach definiowanych w standardzie PolishAPI

Standard definiuje wyłącznie trzy kategorie aktorów, którzy mogą wziąć udział w procesach definiowanych w standardzie PolishAPI:

a) ***Payment Service User*** **(PSU)** – Użytkownik rachunku płatniczego, którego dotyczy dana transakcja płatnicza
b) ***Account Servicing Payment Service Provider*** **(ASPSP)** – Dostawca prowadzący rachunek płatniczy i udostępniający interfejs XS2A dla TPP
c) ***Third Party Provider*** **(TPP)** – Podmiot korzystający z interfejsu XS2A na podstawie i w ramach zgód wyrażonych przez PSU. ASPSP może występować również jako TPP i korzystać z interfejsów wystawionych przez inne ASPSP



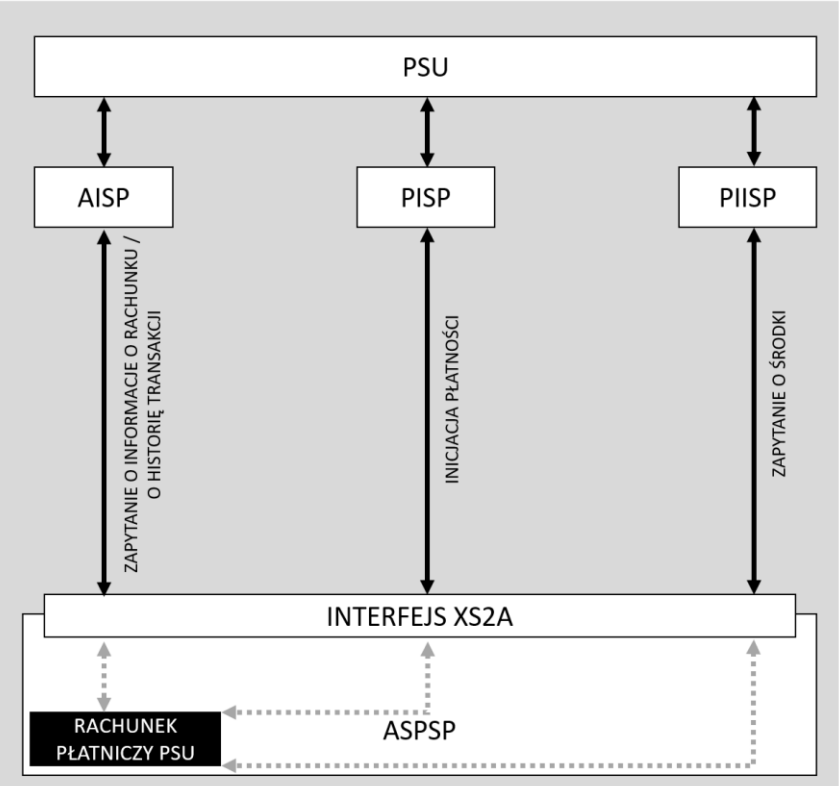

***Ilustracja 1: Ogólny schemat komunikacji w Standardzie PolishAPI***

Standard definiuje trzy role, w których mogą występować aktorzy biorący udział w procesach zdefiniowanych w ramach standardu PolishAPI. Poniższa kategoryzacja nie ogranicza podmiotów występujących w roli TPP do ubiegania się o wpis do rejestru krajowego w więcej niż jednej roli, a ma na celu jedynie zdefiniowanie ról poszczególnych aktorów w opisie komunikacji w ramach standardu PolishAPI.

a) ***Account Information Service Provider*** **(AISP)** - Dostawca Świadczący Usługę Dostępu do Informacji o Rachunku – TPP używające interfejsu XS2A w celu dostępu do informacji o rachunku płatniczym PSU.
b) ***Payment Initiation Service Provider*** **(PISP)** – Dostawca Świadczący Usługę Inicjowania Transakcji Płatniczej – TPP używające interfejsu XS2A w celu inicjacji transakcji płatniczej w ciężar rachunku PSU.
c) ***Payment Instrument Issuer Service Provider*** **(PIISP**) – Dostawca Wydający Instrumenty Płatnicze Oparte na Karcie – TPP używające interfejsu XS2A w celu potwierdzania dostępności na rachunku płatniczym PSU kwoty niezbędnej do wykonania transakcji płatniczej realizowanej w oparciu o instrument wydany przez PIISP.

Aktorzy mogą występować w następujących rolach:

| Aktor / Rola | PSU | ASPSP | TPP |
|---|---|---|---|
| **AISP** | NIE | TAK | TAK |
| **PISP** | NIE | TAK | TAK |
| **PIISP** | NIE | TAK | TAK |




***Ilustracja 2: Ogólny schemat zależności pomiędzy aktorami w Standardzie PolishAPI***

### 1.4.2 Wymagania dot. aktorów w procesach definiowanych w standardzie PolishAPI

a) ASPSP musi wdrożyć interfejs XS2A zgodny ze standardem PolishAPI. ASPSP może wdrożyć również inne standardy interfejsów XS2A, nie są one jednak objęte zakresem niniejszego dokumentu
b) Interfejsy wdrożone przez ASPSP muszą być zgodne z PSD2, Ustawą o Usługach Płatniczych oraz aktami powiązanymi, w szczególności z RTS-ami
c) TPP musi być zarejestrowane w przynajmniej jednym rejestrze w kraju członkowskim Unii Europejskiej w roli, w której chce występować podczas realizacji komunikacji opartej na standardzie PolishAPI
d) TPP oraz ASPSP muszą posiadać ważny certyfikat, służący do wzajemnej identyfikacji w interfejsie XS2A, otrzymany od kwalifikowanego dostawcy usług zaufania, spełniającego wymogi regulacyjne w obszarze usługa zaufania oraz identyfikacji elektronicznej. Certyfikat ten

dodatkowo powinien spełniać wymagania zdefiniowane w RTS oraz specyfikacji technicznej ETSI (TS 119 495).

e) PSU może występować w kontekście rachunku dla klientów indywidualnych lub w kontekście rachunku dla klientów korporacyjnych (firmowych). Domyślnym jest kontekst dla rachunku klienta indywidualnego. Dla wywołań w kontekście rachunku klienta korporacyjnego musi zostać przekazany znacznik „isCorporateContext" o wartości „*true*" w ciele wysyłanego żądania, zgodnie ze specyfikacją techniczną. Dodatkowymi parametrami, które pozwolą na zawężenie zakresu biznesowego informacji, zwracanych poprzez interfejs XS2A, są:
   - psuIdentifierType – typ identyfikatora PSU (dostępny zakres identyfikatorów może być różny dla każdego ASPSP i będzie przez niego zdefiniowany w szczegółowej specyfikacji interfejsu XS2A; specyfikacja PolishAPI definiuje kompletną listę dostępnych typów identyfikatorów: N - NIP, P - PESEL, R - REGON, 1 – Numer Dowodu osobistego, 2 – Numer paszportu, 3 - Inny),
   - psuIdentifierValue – wartość identyfikatora PSU.

Wymienione parametry służą do wskazania bardziej szczegółowego kontekstu wywołania metod interfejsu XS2A, którym jest identyfikator posiadacza rachunku, indywidualnego lub korporacyjnego. Parametry mogą zostać użyte w przypadku PSU, który jednocześnie jest pełnomocnikiem do rachunków wielu klientów w danym ASPSP.

### 1.4.3 Mechanizmy uwierzytelniania PSU

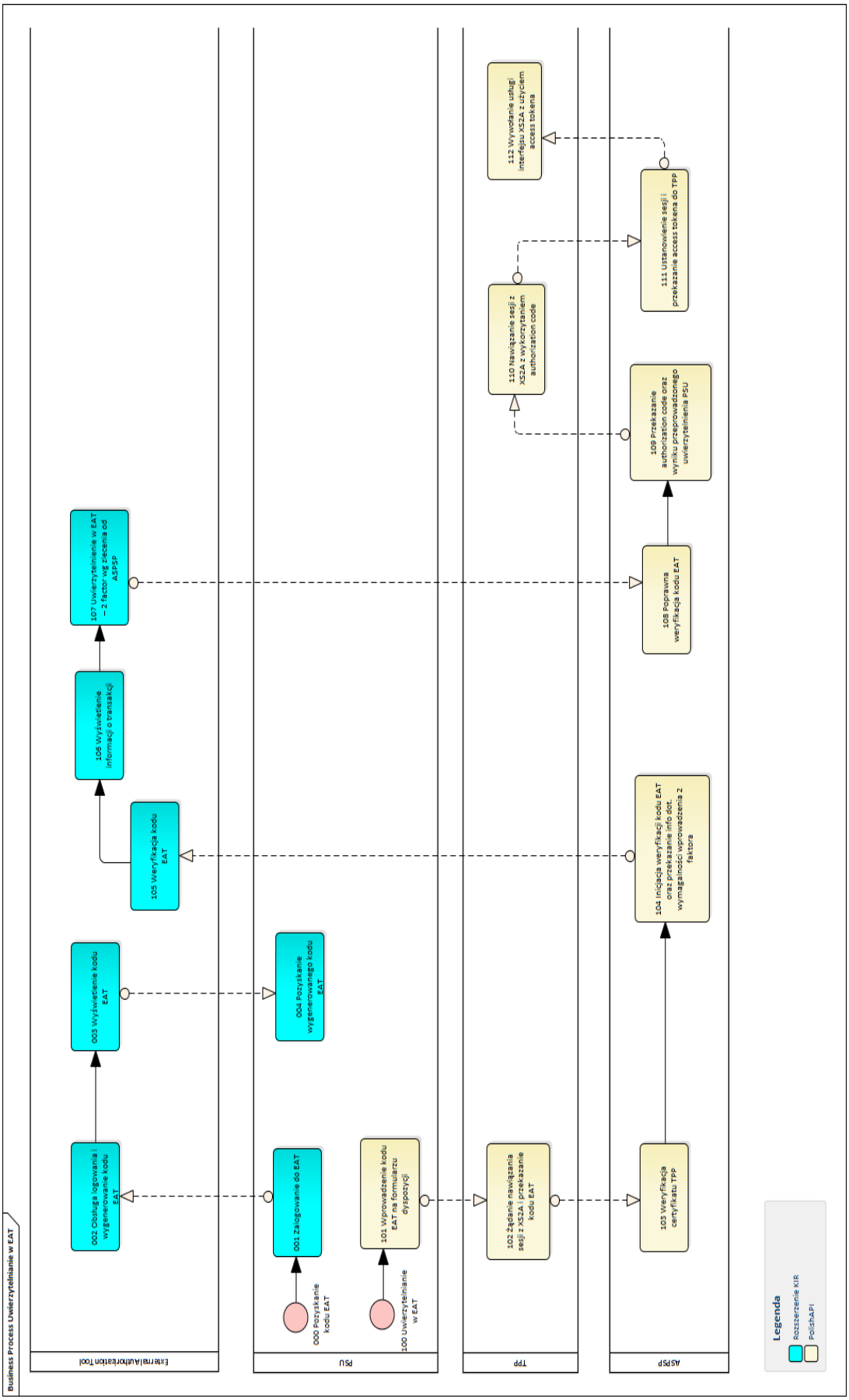
Standard PolishAPI dopuszcza poniższe mechanizmy uwierzytelnienia PSU. Wybór mechanizmów pozostaje wyłącznie w gestii ASPSP. Wybór ten powinien być zgodny z obowiązującymi regulacjami.

#### 1.4.3.1 Mechanizm uwierzytelniania po stronie ASPSP

Standard PolishAPI dopuszcza wykorzystanie mechanizmu uwierzytelniania po stronie ASPSP, zakładającego przekierowanie na stronę internetową ASPSP podczas realizowania usług AIS, PIS i CAF, co oznacza, że dane uwierzytelniające i autoryzacyjne PSU podawane są wyłącznie na stronie internetowej ASPSP. Uwierzytelnienie PSU przeprowadzane jest w interfejsie ASPSP.

#### 1.4.3.2 Mechanizm uwierzytelniania w zewnętrznym narzędziu autoryzacyjnym (*decoupled*)

Standard PolishAPI dopuszcza wykorzystanie mechanizmu uwierzytelniania w zewnętrznym narzędziu autoryzacyjnym podczas realizowania usług AIS i PIS. Mechanizm uwierzytelniania w zewnętrznym narzędziu autoryzacyjnym został wysokopoziomowo przedstawiony na poniższym diagramie. Szczegóły dotyczące jego wykorzystania zostały opisane w rozdziale 7.3.




***Ilustracja 3: Uwierzytelnianie w zewnętrznym narzędziu autoryzacyjnym***

Przyjęto następujące założenia:

- ASPSP współpracuje z dostawcą zewnętrznego narzędzia autoryzacyjnego (dalej zwanego EAT).
- PSU posiada konto w EAT oraz poczynił niezbędne kroki konieczne do korzystania z funkcji generatora kodu.
- ASPSP przygotowuje komunikat zawierający podstawowe informacje o transakcji wyświetlane w EAT przed jej potwierdzeniem.

#### 1.4.3.2.1 Pozyskanie kodu z EAT:

- 001 / PSU w celu pozyskania kodu EAT loguje się do dedykowanego narzędzia zgodnego z PSD2 i z wymogami bezpieczeństwa ASPSP.
- 002 / EAT obsługuje logowanie PSU oraz generuje kod EAT.
- 003 / EAT wyświetla kod PSU.
- 004 / PSU pozyskuje wygenerowany kod EAT.

#### 1.4.3.2.2 Uwierzytelnianie PSU z wykorzystaniem EAT:

- 101 / PSU wprowadza kod EAT na formularzu dyspozycji u TPP.
- 102 / TPP przekazuje żądanie nawiązania sesji z XS2A i przekazania kodu EAT do ASPSP.
- 103 / ASPSP dokonuje weryfikacji TPP, m.in. następuje weryfikacja certyfikatu TPP.
- 104 / ASPSP inicjuje weryfikację kodu EAT oraz przekazuje informację dot. wymagalności stosowania przez EAT 2 faktora oraz podstawowe informacje na temat transakcji.
- 105 / EAT przeprowadza weryfikację kodu.
- 106 / EAT wyświetla użytkownikowi informacje o dokonywanej transakcji (uzgodnione szczegóły).
- 107 / EAT przeprowadza uwierzytelnienie – 2 faktor stosowany jest wg. zlecenia od ASPSP.
- 108 / EAT przekazuje do ASPSP informację o poprawnie zweryfikowanym kodzie EAT.
- 109 / ASPSP przekazuje *authorization code* oraz wynik przeprowadzonego uwierzytelnienia PSU do TPP.
- 110 / TPP nawiązuje sesję z XS2A z wykorzystaniem *authorization code*.
- 111 / ASPSP ustanawia sesję i przekazuje *access token* do TPP.
- 112 / TPP wywołuje usługę interfejsu XS2A z użyciem *access tokena*.

### 1.4.3.3 Inne mechanizmy uwierzytelniające

W standardzie mogą zostać opisane inne, spełniające wymogi regulacyjne oraz uzgodnione w ramach prac grupy projektowej mechanizmy uwierzytelniania. Publikowane będą w kolejnych wersjach niniejszego dokumentu.

## 1.4.4 Zarządzanie zgodami PSU na wykonywanie usług przez TPP

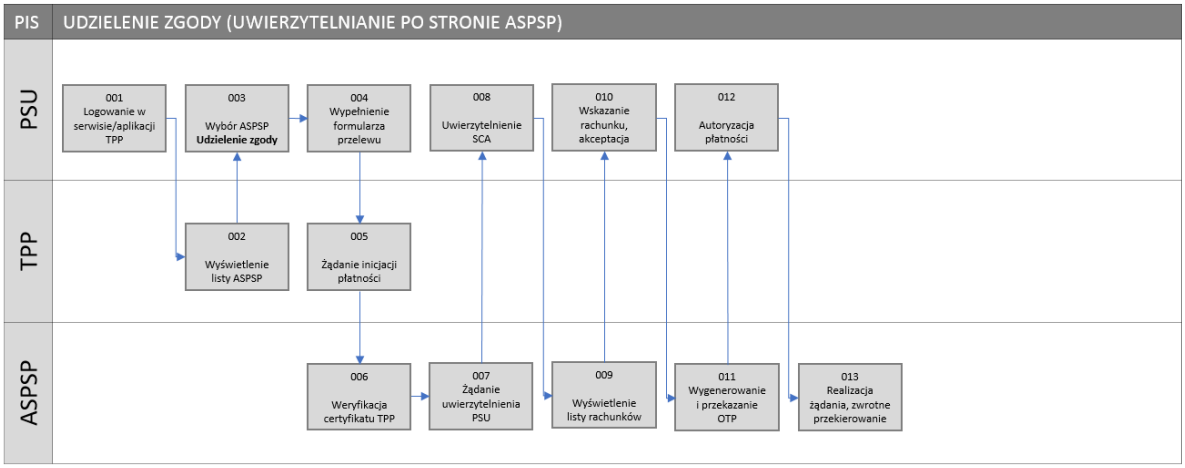
Zgodnie z PSD2, TPP może wykonywać usługi na rzecz PSU jedynie za jego zgodą i w zakresie objętym tą zgodą. Standard PolishAPI definiuje ramy udzielania oraz odwoływania zgód przez PSU.

### 1.4.4.1 Proces udzielenia zgody PSU na wykonanie usługi PIS

Zakłada się, że realizacja procesu inicjacji płatności za każdym razem jest związana z udzieleniem na to zgody przez PSU w ramach interfejsu TPP. Proces udzielenia zgody opisany w rozdziale 1.4.4.1.2 jest procesem opcjonalnym i jego implementacja zależy od decyzji ASPSP.

#### 1.4.4.1.1 Opcja w przypadku użycia mechanizmu uwierzytelniania po stronie ASPSP

- 001 / PSU inicjuje proces w interfejsie TPP
- 002 / TPP wyświetla listę ASPSP
- 003 / PSU wybiera z listy ASPSP oraz udziela TPP zgody na świadczenie usługi PIS
- 004 / PSU wypełnia formularz przelewu, który powinien zawierać przynajmniej informacje, wskazane w rozdziale 3.1.3 niniejszej specyfikacji: „Lista pól wymaganych przez ASPSP w zakresie Zgodności”
- 005 / TPP przekazuje do ASPSP żądanie inicjacji płatności i następuje przekierowanie do domeny ASPSP w celu dokonania uwierzytelnienia PSU
- 006 / ASPSP weryfikuje tożsamość TPP na podstawie certyfikatu (lub także na podstawie rejestru TPP)
- 007 / ASPSP przesyła żądanie uwierzytelnia PSU
- 008 / Uwierzytelnienie
- 009 / ASPSP wyświetla PSU szczegóły transakcji
- 010 / ASPSP generuje i przekazuje PSU dodatkowy element autoryzacyjny (np. OTP) – o ile jest to wymagane zgodnie z obowiązującymi regulacjami
- 011 / PSU autoryzuje transakcję wg metody stosowanej w relacjach z ASPSP (PSU ma możliwość niedokonania autoryzacji, co skutkuje niezrealizowaniem transakcji płatniczej)
- 012 / ASPSP realizuje żądanie, następuje przekierowanie do domeny TPP

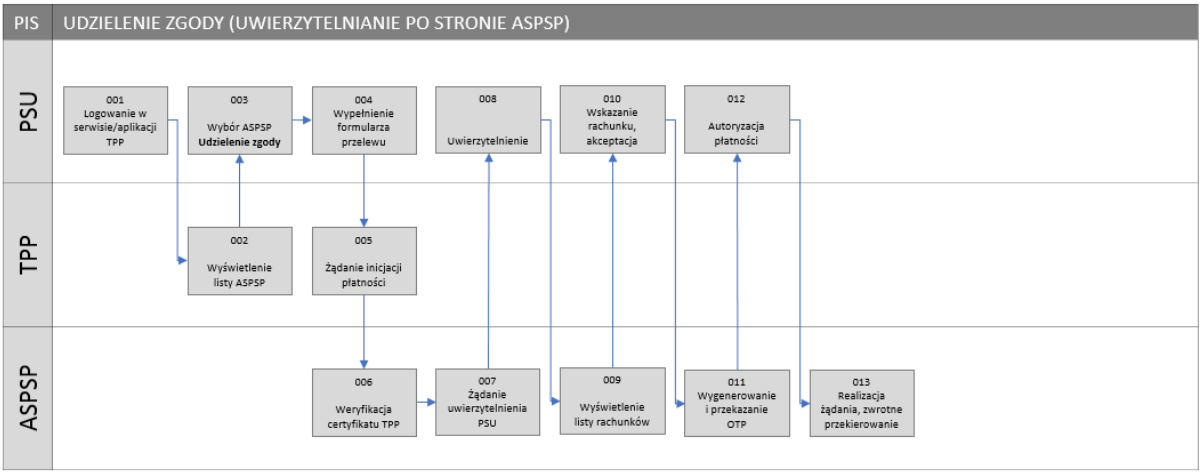



***Ilustracja 4: PIS. Udzielenie zgody (uwierzytelnianie po stronie ASPSP)***

#### 1.4.4.1.2 Opcja w przypadku użycia mechanizmu uwierzytelniania po stronie ASPSP z wyborem rachunku po stronie ASPSP

- 001 / PSU inicjuje proces w interfejsie TPP
- 002 / TPP wyświetla listę ASPSP
- 003 / PSU wybiera z listy ASPSP oraz udziela TPP zgody na świadczenie usługi PIS
- 004 / PSU wypełnia formularz przelewu, który powinien zawierać przynajmniej informacje, wskazane w rozdziale 3.1.3 niniejszej specyfikacji: „Lista pól wymaganych przez ASPSP w zakresie Zgodności”
- 005 / TPP przekazuje do ASPSP żądanie inicjacji płatności i następuje przekierowanie do domeny ASPSP w celu dokonania uwierzytelnienia PSU
- 006 / ASPSP weryfikuje tożsamość TPP na podstawie certyfikatu (lub także na podstawie rejestru TPP)
- 007 / ASPSP przesyła żądanie uwierzytelnia PSU
- 008 / Uwierzytelnienie SCA

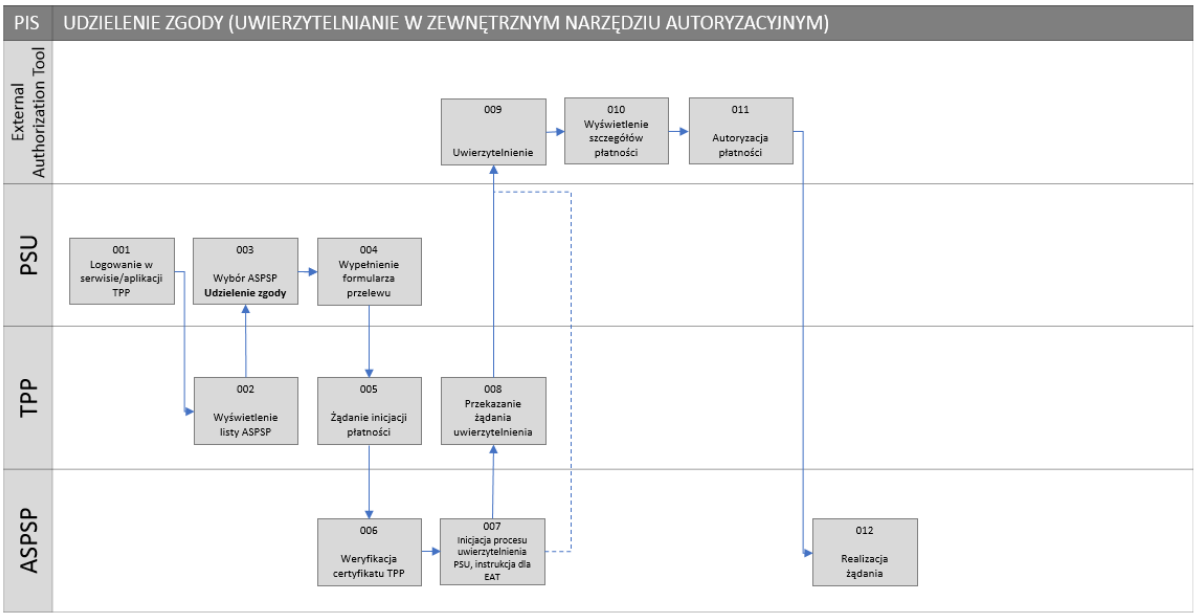
- 009 / ASPSP prezentuje PSU w swoim interfejsie do wyboru listę rachunków płatniczych, z których możliwe jest zainicjowanie transakcji płatniczej
- 010 / PSU wybiera jeden rachunek z listy i akceptuje transakcję
- 011 / ASPSP generuje i przekazuje PSU dodatkowy element autoryzacyjny (np. OTP) – o ile jest to wymagane zgodnie z obowiązującymi regulacjami
- 012 / PSU autoryzuje transakcję wg metody stosowanej w relacjach z ASPSP (PSU ma możliwość niedokonania autoryzacji, co skutkuje niezrealizowaniem transakcji płatniczej)
- 013 / ASPSP realizuje żądanie, następuje przekierowanie do domeny TPP




***Ilustracja 5: PIS. Udzielenie zgody (uwierzytelnianie po stronie ASPSP), wybór rachunku po stronie ASPSP***

#### 1.4.4.1.3 Opcja w przypadku użycia mechanizmu uwierzytelniania w zewnętrznym mechanizmie autoryzacyjnym

- 001 / PSU inicjuje proces w interfejsie TPP
- 002 / TPP wyświetla listę ASPSP
- 003 / PSU wybiera z listy ASPSP oraz udziela TPP zgody na świadczenie usługi PIS
- 004 / PSU wypełnia formularz przelewu, który powinien zawierać przynajmniej informacje, wskazane w rozdziale 3.1.3 niniejszej specyfikacji: „Lista pól wymaganych przez ASPSP w zakresie Zgodności” (w tym podaje numer rachunku, z którego płatność ma być zainicjowana)
- 005 / TPP przekazuje do ASPSP żądanie inicjacji płatności
- 006 / ASPSP weryfikuje tożsamość TPP na podstawie certyfikatu (lub także na podstawie rejestru TPP)
- 007 / ASPSP inicjuje proces uwierzytelnienia PSU, w tym przekazuje instrukcję dotyczącej użycia 2 elementu autoryzacyjnego do EAT – o ile jest to wymagane zgodnie z obowiązującymi regulacjami
- 008 / TPP przekazuje PSU żądanie uwierzytelnienia
- 009 / Uwierzytelnienie w zewnętrznym narzędziu autoryzacyjnym (por. pkt. 1.4.3.2)
- 010 / W zewnętrznym narzędziu autoryzacyjnym wyświetlane są szczegóły płatności, PSU akceptuje transakcję
- 011 / PSU autoryzuje płatność (PSU ma możliwość niedokonania autoryzacji, co skutkuje niezrealizowaniem transakcji płatniczej)
- 012 / ASPSP realizuje żądanie




***Ilustracja 6: PIS. Udzielenie zgody (uwierzytelnianie w zewnętrznym narzędziu autoryzacyjnym)***

### 1.4.4.2 Proces udzielenia zgody PSU na wykonanie usługi AIS

W poniższym rozdziale termin zgoda odnosi się wyłącznie do świadczenia usług AIS i oznacza wyrażenie zgody na usługę, bez wskazywania konkretnych rachunków (w przypadku opcji z pobraniem listy rachunków) lub z ich wskazaniem (w przypadku opcji z ręcznym wprowadzeniem numeru rachunku). Proces ten zawsze łączy się z silnym uwierzytelnianiem klienta (SCA).
Określenie parametrów dostępu (w przypadku opcji z pobraniem listy rachunków) oznacza każdą operację na konkretnych rachunkach w ramach zgody na świadczenie usług AIS, w tym:

- wskazanie konkretnego rachunku
- zmianę parametrów dla konkretnego rachunku (np. daty dostępu)
- cofnięcie wskazania konkretnego rachunku lub
- cofnięcie zgody

Operacje te nie wymagają silnego uwierzytelnienia klienta (SCA).

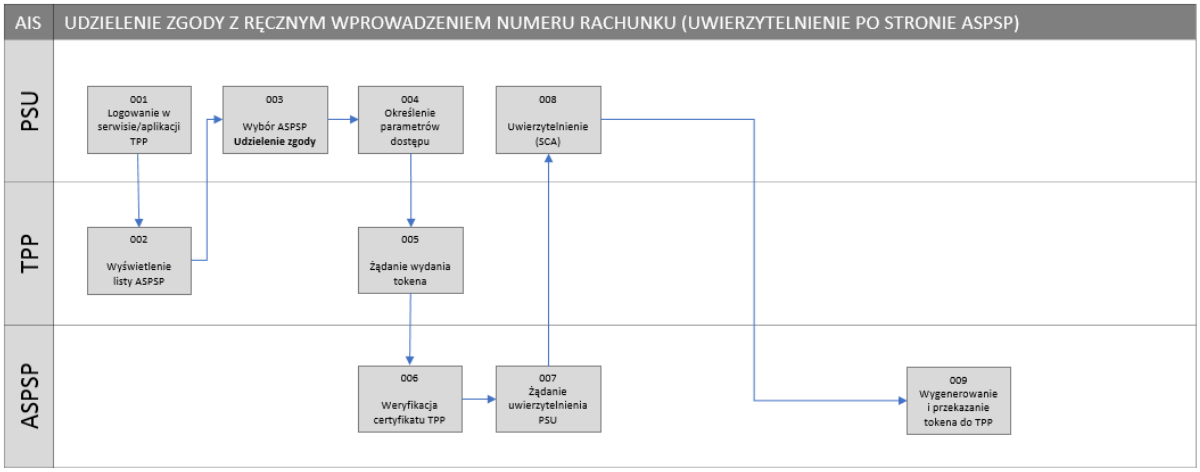
Standard dopuszcza dwa procesy udzielania zgody na usługę AIS (w opcjach uwzględniających uwierzytelnianie po stronie ASPSP oraz w zewnętrznym narzędziu autoryzacyjnym) opisane w pkt. od 1.4.4.2.1 do 1.4.4.2.4. ASPSP może zaimplementować jeden lub oba procesy. Procesy udzielenia zgody z pobraniem listy rachunków są procesami opcjonalnymi i ich implementacja zależy od decyzji ASPSP.

#### 1.4.4.2.1 Udzielenie zgody z ręcznym wprowadzeniem numeru rachunku w przypadku uwierzytelnienia po stronie ASPSP

- 001 / PSU inicjuje proces w interfejsie TPP
- 002 / TPP wyświetla listę ASPSP
- 003 / PSU wybiera z listy ASPSP oraz wyraża dla danego TPP zgodę na świadczenie usługi dostępu do informacji o rachunku prowadzonym przez danego ASPSP
- 004 / PSU wprowadza numer rachunku (-ów) oraz określa zakres dostępu
- 005 / TPP wysyła do ASPSP żądanie wydania tokena
- 006 / ASPSP weryfikuje tożsamość TPP na podstawie certyfikatu (lub także na podstawie rejestru TPP)
- 007 / ASPSP przesyła żądanie uwierzytelnia PSU

- 008 / Uwierzytelnienie SCA
- 009 / ASPSP przekazuje do TPP token dostępowy (zawierający informacje o zakresie dostępu), w przypadku AIS wielokrotnej wraz z refresh tokenem

W przypadku, gdy wprowadzony numer rachunku lub jeden lub więcej z wielu wprowadzonych numerów rachunku jest błędny lub dostęp do niego nie może być udzielony, zwrócony zostanie komunikat błędu 400.

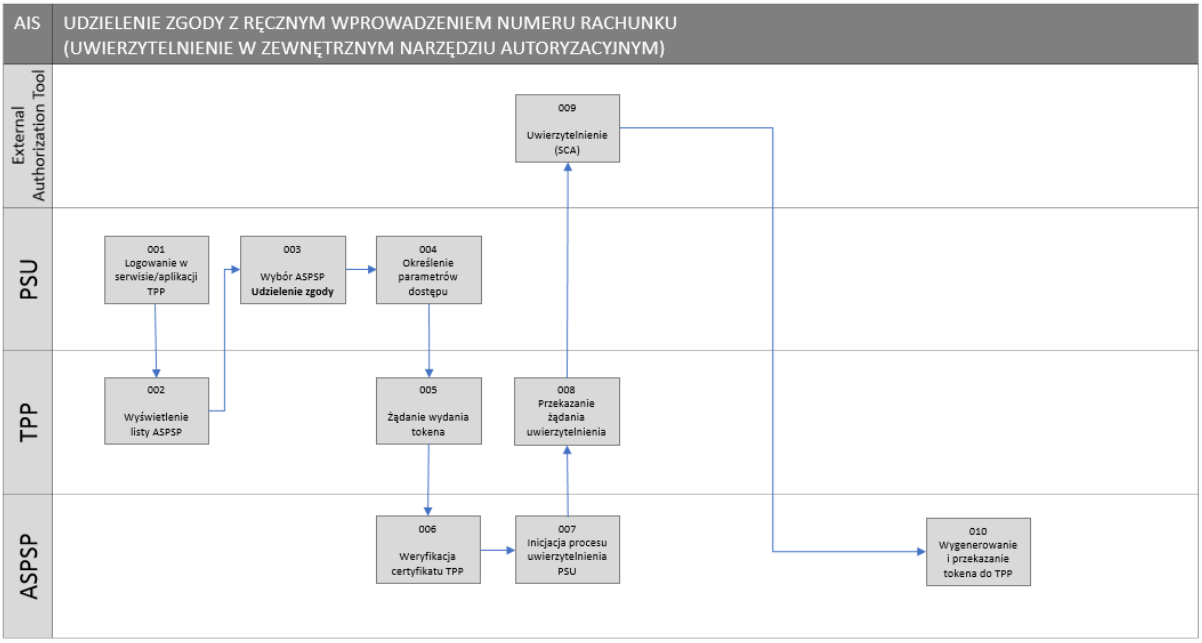



***Ilustracja 7: AIS. Udzielenie zgody z ręcznym wprowadzeniem numeru rachunku (uwierzytelnienie po stronie ASPSP)***

#### 1.4.4.2.2 Udzielenie zgody z ręcznym wprowadzeniem numeru rachunku w przypadku uwierzytelnienia w zewnętrznym narzędziu autoryzacyjnym

- 001 / PSU inicjuje proces w interfejsie TPP
- 002 / TPP wyświetla listę ASPSP
- 003 / PSU wybiera z listy ASPSP oraz wyraża dla danego TPP zgodę na świadczenie usługi dostępu do informacji o rachunku prowadzonym przez danego ASPSP
- 004 / PSU wprowadza numer rachunku (-ów) oraz określa zakres dostępu
- 005 / TPP wysyła do ASPSP żądanie wydania tokena
- 006 / ASPSP weryfikuje tożsamość TPP na podstawie certyfikatu (lub także na podstawie rejestru TPP)
- 007 / ASPSP inicjuje proces uwierzytelniania PSU
- 008 / TPP przekazuje żądanie uwierzytelnienia do PSU
- 009 / Uwierzytelnienie SCA w zewnętrznym narzędziu autoryzacyjnym (por. pkt. 1.4.3.2)
- 010 / ASPSP przekazuje do TPP token dostępowy (zawierający informacje o zakresie dostępu), w przypadku AIS wielokrotnej wraz z refresh tokenem

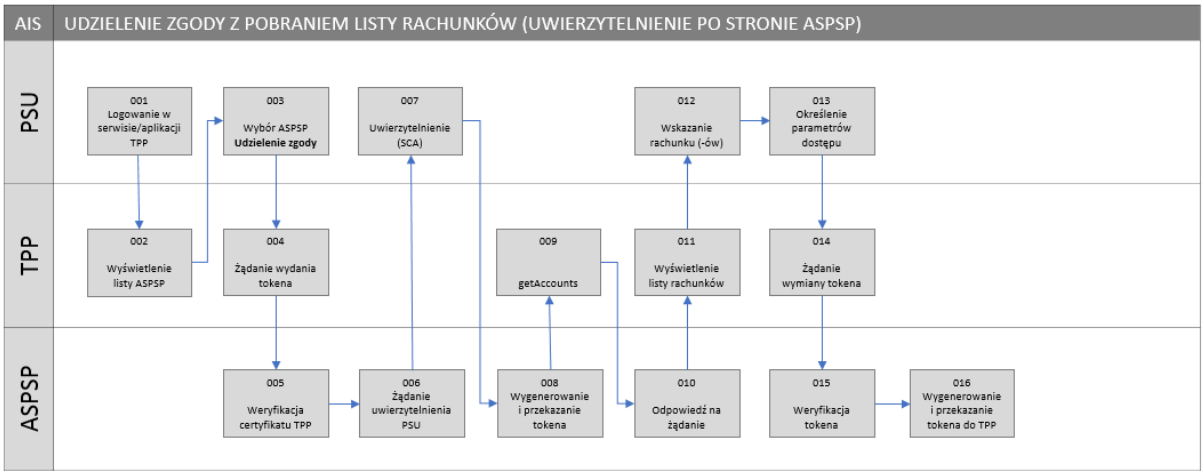
W przypadku, gdy wprowadzony numer rachunku lub jeden lub więcej z wielu wprowadzonych numerów rachunku jest błędny lub dostęp do niego nie może być udzielony, zwrócony zostanie komunikat błędu 400.




***Ilustracja 8: AIS. Udzielenie zgody z ręcznym wprowadzeniem numeru rachunku (uwierzytelnianie w zewnętrznym narzędziu autoryzacyjnym)***

#### 1.4.4.2.3 Udzielenie zgody z pobraniem listy rachunków w przypadku uwierzytelnienia po stronie ASPSP

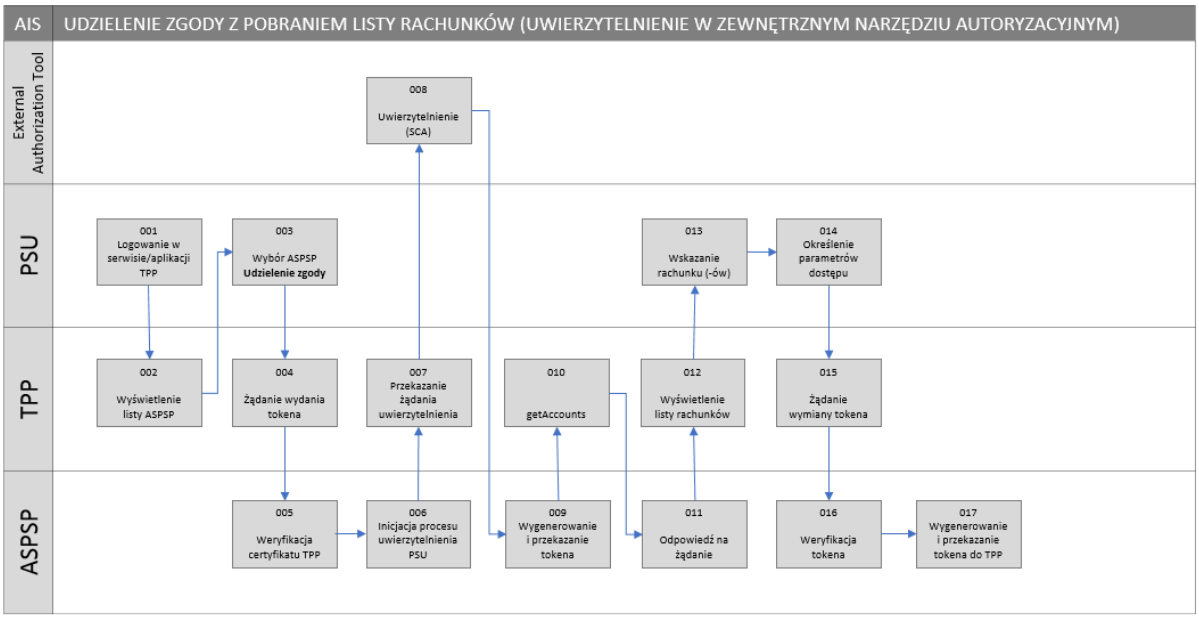
- 001 / PSU inicjuje proces w interfejsie TPP
- 002 / TPP wyświetla listę ASPSP
- 003 / PSU wybiera z listy ASPSP oraz wyraża dla danego TPP zgodę na świadczenie usługi dostępu do informacji o rachunku prowadzonym przez danego ASPSP
- 004 / TPP wysyła do ASPSP żądanie wydania tokena
- 005 / ASPSP weryfikuje tożsamość TPP na podstawie certyfikatu (lub także na podstawie rejestru TPP)
- 006 / ASPSP przesyła żądanie uwierzytelnia PSU
- 007 / Uwierzytelnienie SCA
- 008 / ASPSP przekazuje do TPP token dostępowy
- 009 / TPP wysyła do ASPSP żądanie przekazania listy rachunków PSU (wraz z tokenem dostępowym)
- 010 / ASPSP przekazuje TPP listę rachunków PSU (pełny albo częściowo maskowany numer rachunku + nazwa produktu + typ rachunku)
- 011 / TPP wyświetla listę rachunków
- 012 / PSU wskazuje rachunek (lub rachunki) w celu określenia zakresu dostępu
- 013 / PSU określa parametry dostępu
- 014 / TPP wysyła do ASPSP żądanie wymiany tokena (na token zawierający szczegóły dostępu)
- 015 / ASPSP weryfikuje tożsamość TPP oraz PSU (na podstawie pierwszego tokena dostępowego)
- 016 / ASPSP przekazuje do TPP nowy token dostępowy (zawierający informacje o zakresie dostępu), w przypadku AIS wielokrotnej wraz z refresh tokenem




***Ilustracja 9: AIS. Udzielenie zgody z pobraniem listy rachunków (uwierzytelnianie po stronie ASPSP)***

#### 1.4.4.2.4 Udzielenie zgody z pobraniem listy rachunków w przypadku uwierzytelnienia w zewnętrznym narzędziu autoryzacyjnym

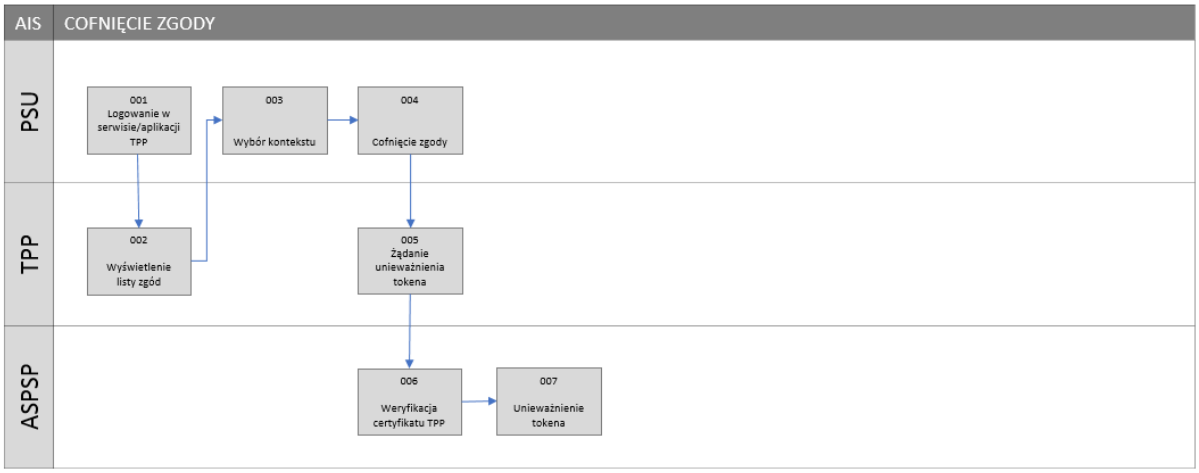
- 001 / PSU inicjuje proces w interfejsie TPP
- 002 / TPP wyświetla listę ASPSP
- 003 / PSU wybiera z listy ASPSP oraz wyraża dla danego TPP zgodę na świadczenie usługi dostępu do informacji o rachunku prowadzonym przez danego ASPSP
- 004 / TPP wysyła do ASPSP żądanie wydania tokena
- 005 / ASPSP weryfikuje tożsamość TPP na podstawie certyfikatu (lub także na podstawie rejestru TPP)
- 006 / ASPSP inicjuje proces uwierzytelnia PSU
- 007 / TPP przekazuje żądanie uwierzytelnienia do PSU
- 008 / Uwierzytelnienie SCA w zewnętrznym narzędziu autoryzacyjnym (por. pkt. 1.4.3.2)
- 009 / ASPSP przekazuje do TPP token dostępowy
- 010 / TPP wysyła do ASPSP żądanie przekazania listy rachunków PSU (wraz z tokenem dostępowym)
- 011 / ASPSP przekazuje TPP listę rachunków PSU (pełny albo częściowo maskowany numer rachunku + nazwa produktu + typ rachunku)
- 012 / TPP wyświetla listę rachunków
- 013 / PSU wskazuje rachunek (lub rachunki) w celu określenia zakresu dostępu
- 014 / PSU określa parametry dostępu
- 015 / TPP wysyła do ASPSP żądanie wymiany tokena (na token zawierający szczegóły dostępu)
- 016 / ASPSP weryfikuje tożsamość TPP oraz PSU (na podstawie pierwszego tokena dostępowego)
- 017 / ASPSP przekazuje do TPP nowy token dostępowy (zawierający informacje o zakresie dostępu), w przypadku AIS wielokrotnej wraz z refresh tokenem




***Ilustracja 10: AIS. Udzielenie zgody z pobraniem listy rachunków (uwierzytelnianie w zewnętrznym narzędziu autoryzacyjnym)***

#### 1.4.4.2.5 Cofnięcie zgody

- 001 / PSU inicjuje proces w interfejsie TPP
- 002 / TPP wyświetla listę zgód
- 003 / PSU wybiera konkretną zgodę z listy zgód, w ramach której dokonane zostaną zmiany
- 004 / PSU cofa zgodę na usługę AIS
- 005 / TPP wysyła do ASPSP żądanie unieważnienia tokena
- 006 / ASPSP weryfikuje tożsamość TPP na podstawie certyfikatu (lub także na podstawie rejestru TPP)
- 007 / ASPSP unieważnia token powiązany ze zgodą




***Ilustracja 11: AIS. Cofnięcie zgody***

### 1.4.4.3 Proces udzielenia zgody PSU na wykonanie usługi CAF

Proces udzielania zgody przez PSU dla ASPSP na wykonanie usługi CAF zostanie opracowany w kolejnej wersji dokumentu. Na potrzeby aktualnej wersji przyjmuje się, że zapytanie w ramach usługi CAF jest wykonywane wyłącznie w sytuacji uprzednio udzielonej zgody.

### 1.4.5 Zastosowanie mechanizmu silnego uwierzytelnienia (SCA)

ASPSP korzystają z dowolnego wybranego przez siebie systemu silnego uwierzytelnienia PSU *(Strong Customer Authentication* – SCA), a standard PolishAPI nie definiuje ani nie rekomenduje żadnego ze sposobów przeprowadzania tej procedury. Ponadto, decyzja o zwolnieniu danej transakcji z obowiązku realizacji procedury SCA pozostaje w wyłącznej gestii ASPSP.

### 1.4.6 Realizacja usług w zakresie Zgodności

Każde ASPSP jest zobowiązane do udostępniania usług w zakresie usług Zgodności na mocy PSD2 oraz powiązanych aktów prawnych. ASPSP udostępnia rachunki zgodne z definicją zawartą w rozdziale 3.2.1 oraz niezależnie podejmuje decyzje o zakresie udostępnianych online danych dot. rachunków płatniczych dostępnych w ramach tej usługi. Realizacja usług w zakresie Zgodności nie będzie wymagała relacji umownej pomiędzy ASPSP a TPP.

### 1.4.7 Realizacja usług w zakresie Premium

Każde ASPSP podejmuje decyzje o udostępnianiu usług w zakresie Premium oraz, w przypadku decyzji o rozpoczęciu ich oferowania, kształtuje ich zakres niezależnie. Realizacja usług w zakresie Premium będzie wymagała relacji umownej pomiędzy ASPSP a TPP.

## 1.5 Rozwój standardu PolishAPI

W chwili obecnej standard PolishAPI definiuje zakres Zgodności dla usług AIS, PIS i CAF. Zakłada się stały rozwój standardu w odpowiedzi na zmiany regulacyjne, technologiczne i biznesowe na rynku polskim oraz europejskim. Zmiany będą publikowane jako kolejne wersje specyfikacji standardu PolishAPI.

# 2 Słownik pojęć użytych w dokumencie

**Account Information Service (AIS) –** usługa dostępu do informacji o rachunku, zdefiniowana w art. 66 PSD2.

**Account Information Service Provider (AISP) –** Dostawca Świadczący Usługę Dostępu do Informacji o Rachunku – TPP używające interfejsu XS2A w celu dostępu do informacji o rachunku płatniczym PSU.

**Confirmation of the Availability of Funds (CAF)** – usługa potwierdzania dostępności na rachunku płatniczym płatnika kwoty niezbędnej do wykonania transakcji płatniczej, zdefiniowana w art. 65 PSD2.

**External Authorization Tool (EAT)** – zewnętrzne narzędzie autoryzacyjne, będące systemem zapewniającym procedurę SCA czyli silnego uwierzytelnienia PSU.

**European Banking Authority (EBA) –** Europejski Urząd Nadzoru Bankowego.

**ETSI –** Europejski Instytut Norm Telekomunikacyjnych.

**OAuth2** – Oauth2 jest otwartym standardem autoryzującym. Pozwala użytkownikom dzielić swoje prywatne zasoby (np. zdjęcia, filmy, kontakty) przechowywane na jednej stronie z inną stroną bez konieczności zagłębiania się w obsługę ich poświadczeń, dostarczając zazwyczaj nazwę użytkownika oraz token (hasło jednorazowe).

**Payment Initiation Service Provider (PISP) –** Dostawca Świadczący Usługę Inicjowania Transakcji Płatniczej – TPP używające interfejsu XS2A w celu inicjacji transakcji płatniczej w ciężar rachunku PSU.

**Payment Initiation Services (PIS) –** usługa inicjowania transakcji płatniczej, zdefiniowana w art. 67 PSD2.

**Payment Instrument Issuer Service Provider (PIISP**) – Dostawca Wydający Instrumenty Płatnicze Oparte na Karcie – TPP używające interfejsu XS2A w celu potwierdzania dostępności na rachunku płatniczym PSU kwoty niezbędnej do wykonania transakcji płatniczej realizowanej w oparciu o instrument wydany przez PIISP.

**Payment Services Directive (PSD) –** Dyrektywa 2007/64/WE Parlamentu Europejskiego i Rady w sprawie usług płatniczych w ramach rynku wewnętrznego.

**Payment Services Directive 2 (PSD2) –** Dyrektywa 2015/2366 Parlamentu Europejskiego i Rady w sprawie usług płatniczych w ramach rynku wewnętrznego, uchylająca Dyrektywę 2007/64/WE.

**Payment Services User (PSU) –** użytkownik usług płatniczych, osoba fizyczna lub prawna korzystająca z usługi płatniczej w charakterze płatnika, odbiorcy lub płatnika i odbiorcy.

**Rachunek płatniczy –** rachunek prowadzony w imieniu co najmniej jednego użytkownika usług płatniczych, wykorzystywany do wykonywania transakcji płatniczych.

**Regulatory Technical Standard (RTS) –** Rozporządzenie Delegowane Komisji (UE) nr 2018/389 z dnia 27.11.2017, uzupełniające dyrektywę Parlamentu Europejskiego i Rady 2015/2366 w odniesieniu do regulacyjnych standardów technicznych dotyczących silnego uwierzytelniania klienta i wspólnych i bezpiecznych otwartych standardów komunikacji.

**Revised Payment Services Directive (PSD2) –** Dyrektywa 2007/64/WE Parlamentu Europejskiego i Rady 2015/2366 z dnia 25 listopada 2015 r. w sprawie usług płatniczych w ramach rynku wewnętrznego (znowelizowana dyrektywa w sprawie usług płatniczych).

**Strong Customer Authentication (SCA) –** silne uwierzytelnianie klienta, oznacza uwierzytelnianie w oparciu o zastosowanie co najmniej dwóch elementów (składników) należących do kategorii: wiedza (coś, co wie wyłącznie użytkownik), posiadanie (coś, co posiada wyłącznie użytkownik) i cechy klienta (coś, czym jest użytkownik), niezależnych w tym sensie, że naruszenie jednego z nich nie osłabia wiarygodności pozostałych, które to uwierzytelnianie jest zaprojektowane w sposób zapewniający ochronę poufności danych uwierzytelniających.

**Swagger –** oprogramowanie open source które pomaga projektować, budować, dokumentować i konsumować usługi RESTful Web.

**TS 119 495** – (v. 1.1.1) specyfikacja techniczna normy odnoszącej się do profilu certyfikatów kwalifikowanych na potrzeby dyrektywy w sprawie usług płatniczych (Electronic Signatures and Infrastructures (ESI); Sector Specific Requirements; Qualified Certificate Profiles and TSP Policy Requirements under the payment services Directive 2015/2366/EU), opublikowana w maju 2018 r.

**Udzielenie zgody** – proces, w wyniku którego PSU udziela TPP zezwolenia na dostęp do jego rachunku, prowadzonego przez ASPSP w celu realizacji usługi, w tym usług AIS, PIS i CAF.

**Uwierzytelnianie** – proces, w wyniku którego ASPSP weryfikuje tożsamość PSU.

**Ustawa o usługach płatniczych (UUP) –** Ustawa z dnia 19 sierpnia 2011 r. o usługach płatniczych.

**XS2A (Access to Account)** – dostęp do rachunków płatniczych, wykorzystywany do wykonywania usług AIS, PIS, CAF oraz innych realizowanych w ramach PolishAPI.

**Zakres Premium** w ramach usług AIS, PIS i CAF **–** usługi wykraczające poza wymogi PSD2.

**Zakres Zgodności** w ramach usług AIS, PIS i CAF **–** usługi wymagane przez PSD2.

# 3 Definicja biznesowa usług z zakresu Zgodności

## 3.1 Definicja biznesowa zakresu Zgodności dla usługi PIS

Usługa inicjowania transakcji płatniczej w zakresie Zgodności polega na udostępnieniu przez ASPSP możliwości zainicjowania płatności z rachunku płatniczego przez PSU za pośrednictwem TPP, występującego w roli PISP, po uprzednim pozyskaniu odpowiednich zgód od PSU.

### 3.1.1 Rodzaje transakcji w zakresie Zgodności

W ramach usługi PIS w zakresie Zgodności ASPSP będzie umożliwiało PSU, za pośrednictwem TPP (PISP) inicjację płatności spełniających łącznie poniższe warunki:

a) Jest to przelew bankowy
b) Jest to przelew pojedynczy
c) Jest to przelew z datą bieżącą
d) Jest to przelew na IBAN (NRB w przypadku ASPSP działających w Polsce), w tym przelew do polskiego Urzędu Skarbowego
e) Jeżeli jest to przelew krajowy, to rozliczany jest w jednym z poniższych systemów (w zależności, który z systemów jest wspierany przez ASPSP):
    a. Elixir,
    b. Express Elixir,
    c. SORBNET2,
    d. Blue Cash.
f) Jeżeli jest to przelew zagraniczny, to rozliczany jest w jednym z poniższych systemów:
    a. SWIFT
    b. SEPA
    c. TARGET
g) Jest dostępny w ramach interfejsu online danego ASPSP
h) PSU wypełni wszystkie dane wymagane do złożenia zlecenia wykonania przelewu (ASPSP nie zapewnia wsparcia w postaci słowników, list rozwijalnych ani innych kreatorów), lub w przypadku procesu opisanego w pkt. 1.4.4.1.2 wszystkie dane z wyjątkiem numeru rachunku, z którego płatność zostanie zainicjowana.

Dane, które przekazuje TPP w zleceniu przelewu nie powinny być modyfikowane przez PSU w domenie ASPSP. Każde ASPSP jest zobowiązane do udostępniania usług w zakresie usług Zgodności na mocy PSD2 oraz powiązanych aktów prawnych. ASPSP niezależnie podejmuje decyzje o zakresie udostępnianych online danych dotyczących rachunków płatniczych dostępnych w ramach tej usługi. Realizacja usług w zakresie Zgodności nie będzie wymagała relacji umownej pomiędzy ASPSP a TPP.

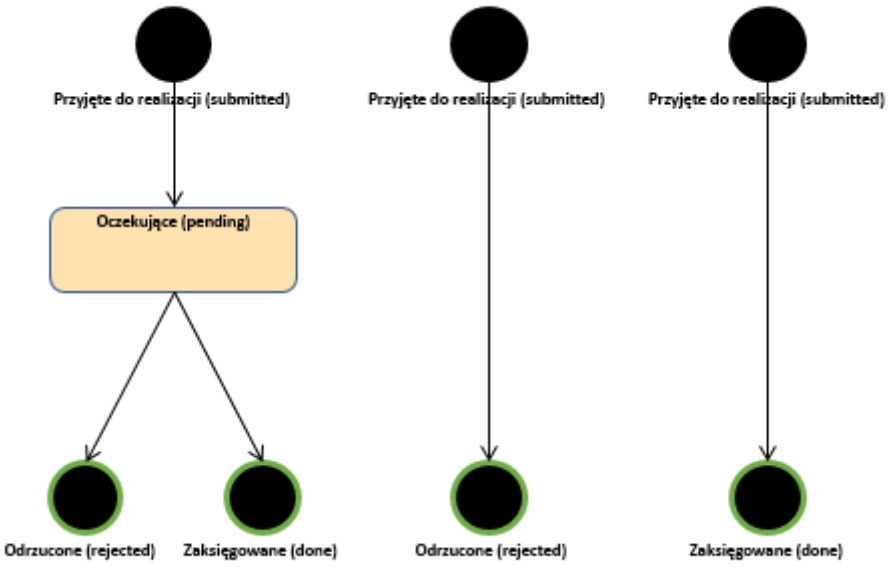
### 3.1.2 Informacja o statusie transakcji

W ramach wymiany komunikatów w usłudze PIS w zakresie Zgodności ASPSP powiadomi TPP niezwłocznie o przyjęciu bądź odrzuceniu zlecenia. Dodatkowo TPP będzie miał możliwość pobrania informacji na temat statusu płatności przy użyciu metody getPayment z opcjonalną możliwością zapytania o status wielu płatności (getMultiplePayments), o ile ASPSP będzie oferował taką funkcjonalność. ASPSP będzie miał opcjonalną możliwość przekazania do TPP (asynchronicznie) informacji o statusie płatności przy wykorzystaniu metody /v1.0/accounts/v1.0/paymentCallBack.

Zdefiniowane są następujące statusy:

a) Przyjęte do realizacji (*submitted*)
b) Oczekujące (*pending*)
c) Odrzucone (*rejected*)

d) Zaksięgowane (*done*)




***Ilustracja 12: Diagram statusów płatności***

### 3.1.3 Definicja rachunku płatniczego

Usługa ta realizowana jest wyłącznie dla rachunków płatniczych, do których dany PSU posiada dostęp on-line. Rachunek taki musi spełniać łącznie poniższe warunki:

a) Jest to rachunek prowadzony dla jednego lub większej liczby użytkowników służący do wykonywania transakcji płatniczych (zgodnie z definicją UUP),
b) PSU posiada dostęp do rachunku on-line.

### 3.1.4 Lista pól wymaganych przez ASPSP w zakresie Zgodności

Do poprawnego zainicjowania transakcji płatniczej w ramach usługi PIS w zakresie Zgodności, ASPSP może zażądać od PSU, za pośrednictwem TPP (PISP), aby poniższe pola zostały wypełnione danymi dotyczącymi zlecenia transakcji. Każde ASPSP może oczekiwać od PSU przekazania za pośrednictwem TPP innego zestawu danych.

W odniesieniu do przelewów zagranicznych użycie niektórych pól będzie opcjonalne, uzależnione od funkcjonalności obsługiwanej przez dane ASPSP. ASPSP będzie realizował płatności pod warunkiem odpowiedniej inicjacji tej płatności przez TPP tzn. przekazaniu odpowiednich pól z właściwą zawartością.

Pola obowiązkowe, definiujące TPP:

a) Nazwa TPP

#### 3.1.4.1 Przelew krajowy

| NAZWA POLA | WYMAGALNOŚĆ | KOMENTARZE |
|---|---|---|
| Adres odbiorcy przelewu | Nie | |
| Data wykonania przelewu | Nie | Dla daty wykonania przelewu w przyszłości, tryb pilności odnosi się do tej daty |

| | | |
|---|---|---|
| Kwota przelewu | Tak | |
| Nazwa nadawcy przelewu | Nie | Nazwa nadawcy uzupełniana przez ASPSP, aby uniknąć sytuacji, gdzie w zleceniu przelewu wychodzącym z ASPSP podane są dane nadawcy inne niż właściciela obciążanego rachunku. |
| Nazwa odbiorcy przelewu | Tak | |
| Numer rachunku nadawcy przelewu | Tak | Może zostać podane przez TPP, jednak PSU powinien mieć możliwość wyboru rachunku obciążanego po przekierowaniu do ASPSP. |
| Numer rachunku odbiorcy przelewu | Tak | |
| Pole opisujące przelew | Tak | |
| Tryb pilności | Tak | ExpressD0, StandardD1 |
| Typ przelewu (system) | Nie | W przypadku przelewu krajowego Elixir, ExpressElixir, Sorbnet, BlueCash, Internal |
| Waluta | Nie | W przypadku, gdy pole jest puste, ASPSP wykona przelew w walucie rachunku. |
| Blokada | Nie | Pole typu bool, dzięki któremu klient będzie mógł explicite wyrazić życzenie, że chce założyć blokadę (w przypadku np. zlecenia przelewu w dniu wolnym). Domyślne zachowanie w przypadku nieprzekazania parametru definiuje ASPSP. |
| Identyfikator transakcji nadany przez TPP | Tak | |

#### 3.1.4.2 Przelew krajowy do Organu Podatkowego/Izby Celnej w Polsce

| NAZWA POLA | WYMAGALNOŚĆ | KOMENTARZE |
|---|---|---|
| Adres odbiorcy przelewu | Tak | |
| Dane urzędu | | |
| Data wykonania przelewu | Nie | Dla daty wykonania przelewu w przyszłości, tryb pilności odnosi się do tej daty |
| Identyfikator płatnika | Tak | |
| Typ identyfikatora płatnika | Tak | Słownik:<br><br>N - NIP, P - PESEL, R - REGON, 1 – Numer Dowodu osobistego, 2 – Numer paszportu, 3 - Inny |
| Identyfikator zobowiązania | Nie | |
| Kwota przelewu | Tak | |
| Nazwa płatnika | Nie | Nazwa nadawcy uzupełniana przez ASPSP, aby uniknąć sytuacji, gdzie w zleceniu przelewu wychodzącym z ASPSP podane są dane nadawcy inne niż właściciela obciążanego rachunku. |
| Numer okresu | Tak | |
| Numer rachunku nadawcy przelewu | Tak | Może zostać podane przez TPP, jednak PSU powinien mieć możliwość wyboru rachunku obciążanego po przekierowaniu do ASPSP. |

| | | |
|---|---|---|
| Numer rachunku odbiorcy przelewu | Tak | |
| Symbol formularza | Tak | |
| Typ okresu | Tak | |
| Typ przelewu | Tak | Wartość stała – przelew do urzędu skarbowego |
| Tryb pilności | Tak | ExpressD0, StandardD1 |
| Tryb realizacji przelewu (system) | Nie | Standard (Elixir), ekspres (ExpressElixir) |
| Waluta | Tak | |
| Blokada | Nie | Pole typu bool, dzięki któremu klient będzie mógł explicite wyrazić życzenie, że chce założyć blokadę (w przypadku np. zlecenia przelewu w dniu wolnym). Domyślne zachowanie w przypadku nieprzekazania parametru definiuje ASPSP. |
| Identyfikator transakcji nadany przez TPP | Tak | |

#### 3.1.4.3 Przelew zagraniczny EEA

| NAZWA POLA | WYMAGALNOŚĆ | KOMENTARZE |
|---|---|---|
| Adres odbiorcy przelewu | Nie | |
| Data wykonania przelewu | Nie | Dla daty wykonania przelewu w przyszłości, tryb pilności odnosi się do tej daty |
| Kwota przelewu | Tak | |
| Nazwa nadawcy przelewu | Nie | Nazwa nadawcy uzupełniana przez ASPSP, aby uniknąć sytuacji, gdzie w zleceniu przelewu wychodzącym z ASPSP podane są dane nadawcy inne niż właściciela obciążanego rachunku. |
| Nazwa odbiorcy przelewu | Tak | |
| Numer rachunku nadawcy przelewu | Tak | Może zostać podane przez TPP, jednak PSU powinien mieć możliwość wyboru rachunku obciążanego po przekierowaniu do ASPSP. |
| Numer rachunku odbiorcy przelewu | Tak | |
| Pole opisujące przelew | Tak | |
| Tryb realizacji przelewu | Tak | Standard, express |
| Typ przelewu (system) | Nie | SEPA, Instant SEPA, Target |
| Waluta | Nie | Wartość stała - EUR |
| Blokada | Nie | Pole typu bool, dzięki któremu klient będzie mógł explicite wyrazić życzenie, że chce założyć blokadę (w przypadku np. zlecenia przelewu w dniu wolnym). Domyślne zachowanie w przypadku nieprzekazania parametru definiuje ASPSP. |
| Identyfikator transakcji nadany przez TPP | Tak | |

#### 3.1.4.4 Przelew zagraniczny inny niż EEA

| NAZWA POLA | WYMAGALNOŚĆ | KOMENTARZE |
|---|---|---|
| Data wykonania przelewu | Nie | Dla daty wykonania przelewu w przyszłości, tryb pilności (realizacji przelewu) odnosi się do tej daty |
| Numer rachunku nadawcy przelewu | Tak | Nazwa nadawcy uzupełniana przez ASPSP, aby uniknąć sytuacji, gdzie w zleceniu przelewy wychodzącym z ASPSP podane są dane nadawcy inne niż właściciela obciążanego rachunku. |
| Numer rachunku odbiorcy przelewu | Tak | |
| Nazwa nadawcy przelewu | Nie | Nazwa nadawcy uzupełniana przez ASPSP, aby uniknąć sytuacji, gdzie w zleceniu przelewu wychodzącym z ASPSP podane są dane nadawcy inne niż właściciela obciążanego rachunku. |
| Nazwa odbiorcy przelewu | Tak | |
| Adres odbiorcy przelewu | Nie | |
| Pole opisujące przelew | Tak | |
| Kwota przelewu | Tak | |
| Waluta | Tak | |
| Numer BIC/SWIFT Banku odbiorcy | Nie | Pola warunkowe – wymagalność zależna od docelowej specyfikacji Banku implementującego standard PolishAPI |
| Kraj Banku odbiorcy | Nie | |
| Nazwa Banku odbiorcy | Nie | |
| Adres Banku odbiorcy | Nie | |
| Kod banku odbiorcy | Nie | |
| Klauzula kosztowa | Nie | |
| Tryb realizacji przelewu | Tak | Standard, urgent, express |
| Typ przelewu (system) | Nie | SWIFT |
| Blokada | Nie | Pole typu bool, dzięki któremu klient będzie mógł explicite wyrazić życzenie, że chce założyć blokadę (w przypadku np. zlecenia przelewu w dniu wolnym). Domyślne zachowanie w przypadku nieprzekazania parametru definiuje ASPSP. |
| Identyfikator transakcji nadany przez TPP | Tak | |

### 3.1.5 Diagram zapytania w ramach usługi PIS w zakresie Zgodności

Diagram został przedstawiony w Przypadku Użycia #1 w rozdziale 4.

### 3.1.6 Autoryzacja transakcji płatniczej zainicjowanej za pomocą usługi PIS

ASPSP zapewnia możliwość autoryzacji transakcji płatniczej zleconej przez PSU za pomocą usługi inicjowania transakcji płatniczej w rozumieniu ustawy o usługach płatniczych (UUP), bez względu na metodę autoryzacji oraz jej złożoność. Wybór metody autoryzacji jest po stronie ASPSP.

## 3.2 Definicja biznesowa zakresu Zgodności dla usługi AIS

Usługa dostępu do informacji o rachunku w zakresie Zgodności polega na udostępnieniu przez ASPSP danych dotyczących historii transakcji oraz wybranych informacji dotyczących rachunku płatniczego, do którego PSU posiada aktywny dostęp on-line. Dostęp udzielany jest dla TPP występującego jako AISP, po uprzednim pozyskaniu odpowiednich zgód od PSU. Ponadto, ASPSP udostępni mechanizmy filtrowania danych, zgodnie z kryteriami dostępnymi on-line w systemie ASPSP (czyli przez bankowość elektroniczną), np.:

a) Data księgowania transakcji (wg wskazanej konkretnej daty księgowania oraz w podanym zakresie dat)
b) Kwota transakcji
c) Obciążenia i uznania na rachunku płatniczym

### 3.2.1 Definicja rachunku płatniczego

Usługa ta realizowana jest wyłącznie dla rachunków płatniczych, do których dany PSU posiada dostęp on-line. Rachunek taki musi spełniać łącznie poniższe warunki:

c) Jest to rachunek prowadzony dla jednego lub większej liczby użytkowników służący do wykonywania transakcji płatniczych (zgodnie z definicją UUP),
d) PSU posiada dostęp do rachunku on-line.

### 3.2.2 Częstotliwość zapytań w zakresie Zgodności

W ramach usługi AIS w zakresie Zgodności, TPP (AISP) może zażądać od ASPSP przesłania historii rachunku płatniczego oraz wybranych informacji o rachunku płatniczym:

a) Do 4 razy w ciągu 24 godzin od momentu przesłania pierwszego zapytania, w przypadku, gdy pobranie danych nie jest inicjowane na żądanie PSU za pośrednictwem TPP (AISP), ale przez TPP (AISP) na mocy zgód wyrażonych uprzednio przez PSU;
b) Każdorazowo, w przypadku, gdy żądanie jest bezpośrednio inicjowane przez PSU za pośrednictwem TPP (AISP).

Większa częstotliwość zapytań w przypadku, gdy pobranie danych nie jest inicjowane na żądanie PSU za pośrednictwem TPP (AISP), ale przez TPP (AISP) na mocy zgód wyrażonych uprzednio przez PSU, może być realizowana w usłudze AIS wyłącznie w zakresie Premium i jest przedmiotem odrębnych ustaleń bilateralnych pomiędzy ASPSP i TPP (AISP).

### 3.2.3 Zakres informacji dot. historii rachunku płatniczego w zakresie Zgodności

W zakresie Zgodności usługi AIS jest udostępnienie pełnej dostępnej on-line historii rachunku w zakresie transakcji zaksięgowanych, oczekujących i odrzuconych na danym rachunku płatniczym, wraz z mechanizmami filtrowania danych (w tym zakres dat dla historii transakcji) dla PSU, oraz blokad, które są widoczne dla PSU w kanale on-line ASPSP. Przy czym transakcja oczekująca (*pending*) oznacza transakcję niezaksięgowaną, niemodyfikowalną, wpływającą na dostępne środki (saldo dostępne). Zgodnie z regulacjami, udostępnienie historii rachunku wiąże się z procesem SCA zawsze (niezależnie od zastosowanych wyłączeń od obowiązku stosowania SCA), gdy klient uzyskuje dostęp do rachunku online po raz pierwszy oraz gdy zapytanie dotyczy historii dłuższej niż 90 dni. SCA można nie stosować jeżeli zapytanie dotyczy historii transakcji płatniczych przeprowadzonych w ciągu ostatnich 90 dni, pod warunkiem że nie minęło więcej niż 90 dni odkąd ostatni raz uzyskano dostęp do historii obejmującej do 90 dni i zastosowano SCA.

### 3.2.4 Lista pól udostępnianych przez ASPSP w zakresie Zgodności

W ramach odpowiedzi na przesłane przez TPP (AISP) żądanie, ASPSP przesyła odpowiedź w zakresie poniższych pól.

| NAZWA POLA | WYMAGALNOŚĆ | KOMENTARZ |
|---|---|---|
| Numer rachunku | Tak | Numer rachunku |
| Waluta rachunku | Nie | Dla każdego rachunku płatniczego |
| Imiona i nazwisko / Nazwa PSU | Tak | Dla osoby fizycznej imiona i nazwisko, dla osoby prawnej nazwa |
| Adres PSU | Nie | Adres posiadacza rachunku |
| Dostępne środki | Tak | Dostępne środki w walucie rachunku - po wykonaniu transakcji |
| Identyfikator transakcji | Tak | Unikalny identyfikator danej transakcji nadany przez ASPSP |
| Kwota w oryginalnej walucie | Nie | Dla każdej transakcji w historii rachunku |
| Saldo księgowe rachunku | Tak | Saldo księgowe rachunku - po wykonaniu transakcji |
| Kod typu rachunku | Nie | Klucz jednoznacznie identyfikujący typ rachunku w słowniku typów rachunków, zdefiniowanym przez ASPSP |
| Typ rachunku | Warunkowo | Opis biznesowy typu rachunku powiązanego z kluczem słownikowym, przekazanym w polu „Kod typu rachunku”.<br>Np. rachunek dla konsumenta / firmowy + odniesienie do produktu, np. konto, karta kredytowa, rachunek oszczędnościowy, itd.<br>Wartość wymagana warunkowo, w przypadku przekazania pola „Kod typu rachunku”. |
| Data transakcji | Tak | Dla każdej transakcji w historii rachunku |
| Kwota | Tak | Dla każdej transakcji w historii rachunku |
| Numer rachunku nadawcy | Warunkowo | Dla każdej transakcji w historii rachunku. W zależności od typu transakcji - uznaniowa/obciążeniowa. |
| Numer rachunku odbiorcy | Warunkowo | Dla każdej transakcji w historii rachunku. W zależności od typu transakcji - uznaniowa/obciążeniowa. |
| Opis | Nie | Dla każdej transakcji w historii rachunku |
| Tytuł | Tak | Dla każdej transakcji w historii rachunku |
| Kod statusu transakcji | Nie | Dla każdej transakcji w historii rachunku. Wartość klucza słownika statusów transakcji zdefiniowana przez ASPSP. |
| Status transakcji | Warunkowo | Opis biznesowy statusu transakcji. Wymagany jeśli została przekazana wartość słownikowa w polu „Kod statusu transakcji” |
| Typ transakcji | Nie | Transakcja uznaniowa/obciążeniowa.<br>Dla każdej transakcji w historii rachunku |
| Waluta oryginalna transakcji | Nie | Dla każdej transakcji w historii rachunku |
| Nazwa typu rachunku (definiowana przez ASPSP) | | Nazwa handlowa produktu |
| Dane Urzędu Skarbowego | Nie | Tylko dla przelewu do Organu Podatkowego / Izby Celnej w Polsce |
| Data księgowania | Nie | Dla każdej transakcji w historii rachunku |
| Data kursu waluty | Nie | Dla każdej transakcji w historii rachunku |
| ID transakcji TPP | Nie | Unikalny identyfikator danej transakcji nadany przez TPP |

| | | |
|---|---|---|
| Identyfikator płatnika w US | Warunkowo | Tylko dla przelewu do Organu Podatkowego / Izby Celnej w Polsce |
| Identyfikator zobowiązania w US | Nie | Tylko dla przelewu do Organu Podatkowego / Izby Celnej w Polsce |
| Kurs transakcji | Nie | Dla każdej transakcji w historii rachunku |
| Kod waluty przed przewalutowaniem transakcji | Nie | Zgodny ze standardem ISO |
| Kod waluty po przewalutowaniu transakcji | Nie | Zgodny ze standardem ISO |
| Numer okresu | Warunkowo | Tylko dla przelewu do Organu Podatkowego / Izby Celnej w Polsce |
| Numer wirtualny rachunku odbiorcy | Nie | Dla każdej transakcji w historii rachunku |
| Rok | Warunkowo | Tylko dla przelewu do Organu Podatkowego / Izby Celnej w Polsce |
| Symbol formularza | Warunkowo | Tylko dla przelewu do Organu Podatkowego / Izby Celnej w Polsce |
| Typ okresu | Warunkowo | Tylko dla przelewu do Organu Podatkowego / Izby Celnej w Polsce |
| Typ wpłaty | Nie | Tylko dla przelewu do ZUS |
| Unikalny identyfikator instrumentu płatniczego, za którego pomocą wykonano transakcję | Nie | Np. numer karty płatniczej (maskowany, zgodnie z prezentacją danych w interfejsie on-line ASPSP) |
| Dane właściciela instrumentu płatniczego, za którego pomocą wykonano transakcję | Nie | Dane właściciela karty płatniczej |
| Rodzaj operacji | Tak | Dla każdej transakcji w historii rachunku |
| Saldo rachunku po transakcji | Nie | Dla każdej transakcji w historii rachunku |
| Typ identyfikatora płatnika w US | Warunkowo | Tylko dla przelewu do Organu Podatkowego / Izby Celnej w Polsce |
| Nazwa odbiorcy przelewu przychodzącego | Nie | Dla każdej transakcji przelewu przychodzącego w historii rachunku |
| Adres odbiorcy przelewu przychodzącego | Nie | Dla każdej transakcji przelewu przychodzącego w historii rachunku |
| Nazwa odbiorcy przelewu wychodzącego | Warunkowo | Dla każdej transakcji przelewu wychodzącego w historii rachunku. W zależności od typu transakcji - uznaniowa/obciążeniowa. |
| Adres odbiorcy przelewu wychodzącego | Nie | Dla każdej transakcji przelewu wychodzącego w historii rachunku |
| Adres Banku odbiorcy | Warunkowo | Tylko dla przelewów zagranicznych |
| Kod Banku odbiorcy | Warunkowo | Tylko dla przelewów zagranicznych |
| Nazwa Banku odbiorcy | Warunkowo | Tylko dla przelewów zagranicznych |
| Numer BIC/SWIFT Banku odbiorcy | Warunkowo | Tylko dla przelewów zagranicznych |
| Kod kraju Banku odbiorcy | Warunkowo | Tylko dla przelewów zagranicznych |
| Nazwa nadawcy przelewu przychodzącego | Tak | Dla każdej transakcji przelewu przychodzącego w historii rachunku |
| Adres nadawcy przelewu przychodzącego | Nie | Dla każdej transakcji przelewu przychodzącego w historii rachunku |
| Nazwa nadawcy przelewu wychodzącego | Tak | Dla każdej transakcji przelewu wychodzącego w historii rachunku |
| Adres nadawcy przelewu wychodzącego | Nie | Dla każdej transakcji przelewu wychodzącego w historii rachunku |
| Nazwa konta ustawiona przez klienta | Nie | O ile ASPSP udostępnia taką usługę. |

| | | |
|---|---|---|
| Nazwa inicjatora transakcji | Warunkowo | W przypadku transakcji zlecanych przez osobę inną niż właściciel rachunku (imię i nazwisko) |
| Adres inicjatora transakcji | Nie | W przypadku transakcji zlecanych przez osobę inną niż właściciel rachunku |
| Nazwa TPP | Nie | W przypadku transakcji inicjowanych w ramach usługi PIS |
| MCC | Nie | Kod dla każdej transakcji / operacji wykonanej przy użyciu karty |
| Powód odrzucenia | Nie | W przypadku transakcji odrzuconych |
| Data odrzucenia | Nie | W przypadku transakcji odrzuconych |
| Data ważności blokady | Nie | W przypadku blokad na rachunku |
| NIP | Warunkowo | Podstawowy identyfikator płatnika do ZUS będący numerem NIP. |
| Dodatkowy numer identyfikacyjny płatnika do ZUS | Warunkowo | Wartość dodatkowego identyfikatora płatnika do ZUS (wartość odpowiednia dla wybranego typu dodatkowego identyfikatora płatnika z pola „Typ dodatkowego identyfikatora płatnika") |
| Typ dodatkowego identyfikatora płatnika | Nie | Wartość słownikowa określająca typ dodatkowego identyfikatora płatnika do ZUS. |
| Numer deklaracji | Nie | Wartość numeru deklaracji dla przelewów do ZUS, zgodny z formularzem tego typu przelewów |
| Okres deklaracji | Nie | Wartość okresu deklaracji dla przelewów do ZUS, zgodny z formularzem tego typu przelewów |
| Identyfikator zobowiązania w ZUS | Nie | Wartość identyfikatora zobowiązania, z którego wynika należność dla przelewów do ZUS, zgodny z formularzem tego typu przelewów |

Opisane w powyższej tabeli pola stają się obligatoryjne dla ASPSP w relacji do zakresu informacji o rachunkach i transakcjach płatniczych, jakie dany ASPSP udostępnia w swoim interfejsie online, z zastrzeżeniem wyjątków wynikających z przepisów prawa (np. w zakresie szczególnie chronionych danych dotyczących płatności lub danych osobowych). Każdy ASPSP może dodać do udostępnianego zakresu danych na temat rachunku i transakcji dodatkowe pola, wykorzystując w tym celu pole auxData typu Mapa w strukturach AccountInfo, TransactionInfo, TransactionHoldInfo, TransactionPendingInfo oraz TransactionRejectedInfo.

Lista pól udostępnianych w przypadku, gdy ASPSP umożliwia skorzystanie z opcji pobrania listy rachunków w ramach procesu udzielania zgody na usługi AIS lub PIS.

| NAZWA POLA | WYMAGALNOŚĆ | KOMENTARZ |
|---|---|---|
| Numer rachunku | Tak | Numer rachunku w formie maskowanej, widoczne 2 pierwsze i 4 ostatnie cyfry rachunku lub bez maskowania, zgodnie z decyzją ASPSP |
| Nazwa typu rachunku (definiowana przez Bank) | Tak | Nazwa handlowa produktu |
| Typ rachunku | Tak | Np. rachunek dla konsumenta / firmowy + odniesienie do produktu, np. konto, karta kredytowa, rachunek oszczędnościowy, itd. |

### 3.2.5 Diagramy zapytań w ramach usługi AIS w zakresie Zgodności

Diagram został przedstawiony w Przypadku Użycia#2, w rozdziale 4.

## 3.3 Definicja biznesowa zakresu Zgodności dla usługi CAF

Usługa potwierdzania dostępności na rachunku płatniczym płatnika kwoty niezbędnej do wykonania transakcji płatniczej w zakresie Zgodności polega na przesłaniu zapytania przez TPP, występującego w roli PIISP do ASPSP, o potwierdzenie czy na danym rachunku płatniczym PSU znajdują się środki w określonej w zapytaniu kwocie, na podstawie zgód uprzednio udzielonych przez PSU. W odpowiedzi ASPSP zwraca odpowiedź będącą komunikatem „TAK" albo „NIE".

### 3.3.1 Lista pól wymaganych przez ASPSP w zakresie Zgodności

Do poprawnego obsłużenia zapytania o potwierdzenie dostępności na rachunku płatniczym płatnika kwoty niezbędnej do wykonania transakcji płatniczej w ramach usługi CAF w zakresie Zgodności, ASPSP może zażądać od PSU, za pośrednictwem TPP (PIISP), aby poniższe pola zostały wypełnione danymi dot. zlecenia transakcji. Szczegóły dot. pól zdefiniowane są w rozdziale 5.7 Kanoniczny model danych.

| NAZWA POLA | WYMAGALNOŚĆ | KOMENTARZE |
|---|---|---|
| Identyfikator rachunku, którego dotyczy zapytanie | Tak | Rachunek uprzednio powiązany z instrumentem płatniczym na bazie zgody wyrażonej przez PSU. |
| Kwota | Tak | |
| Waluta | Tak | Waluta transakcji |

### 3.3.2 Diagram zapytania w ramach usługi CAF w zakresie Zgodności

Diagram został przedstawiony w Przypadku Użycia #3, w rozdziale 4.

# 4 Przykładowe przypadki użycia

Bieżąca wersja Standardu PolishAPI opisuje sposób realizacji transakcji opartych o interfejs XS2A w zakresie Zgodności, zdefiniowanym w rozdziale 3 niniejszego dokumentu, a TPP może uczestniczyć w tych transakcjach w jednej ze zdefiniowanych ról.

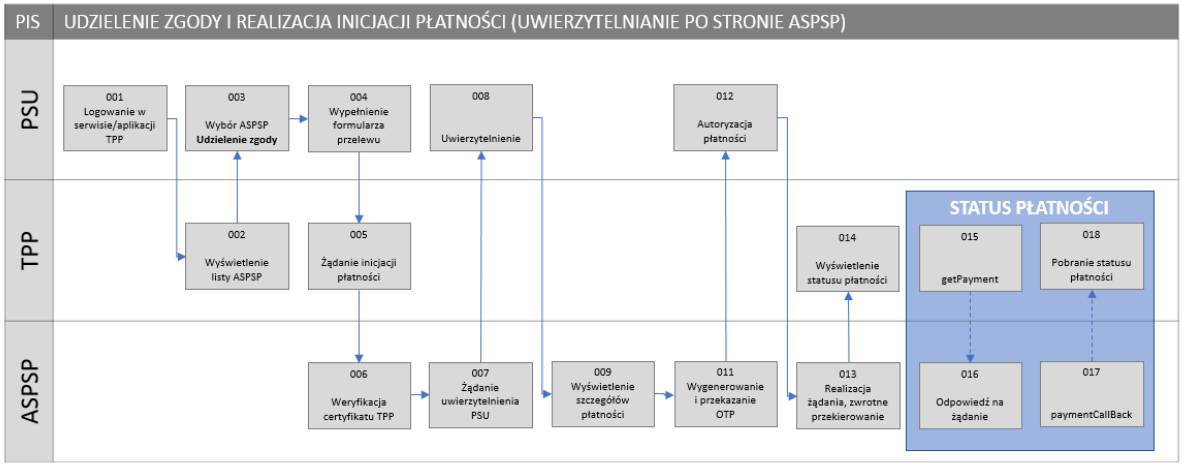
Przykłady obrazujące wykorzystanie poszczególnych usług zostały przedstawione w niniejszym rozdziale. Mają one na celu wyłącznie zilustrowanie kroków dla poszczególnych usług i nie powinny być traktowane jako wyczerpująca lista dopuszczalnych przypadków użycia.

## 4.1 Przypadek Użycia #1: inicjacja pojedynczej płatności przez PISP (PIS)

Wykorzystanie usługi PIS w zakresie Zgodności zaprezentowane w tym Przypadku Użycia polega na zainicjowaniu przez TPP występującego w roli PISP transakcji płatniczej w ciężar rachunku płatniczego PSU prowadzonego przez ASPSP, w oparciu o stosowne zapisy UUP. ASPSP może odrzucić transakcję, jeżeli TPP (PISP) nie zostanie zidentyfikowany jako podmiot uprawniony do realizacji usługi PIS.

### 4.1.1 Inicjacja pojedynczej płatności przez PISP z wykorzystaniem mechanizmu uwierzytelniania po stronie ASPSP

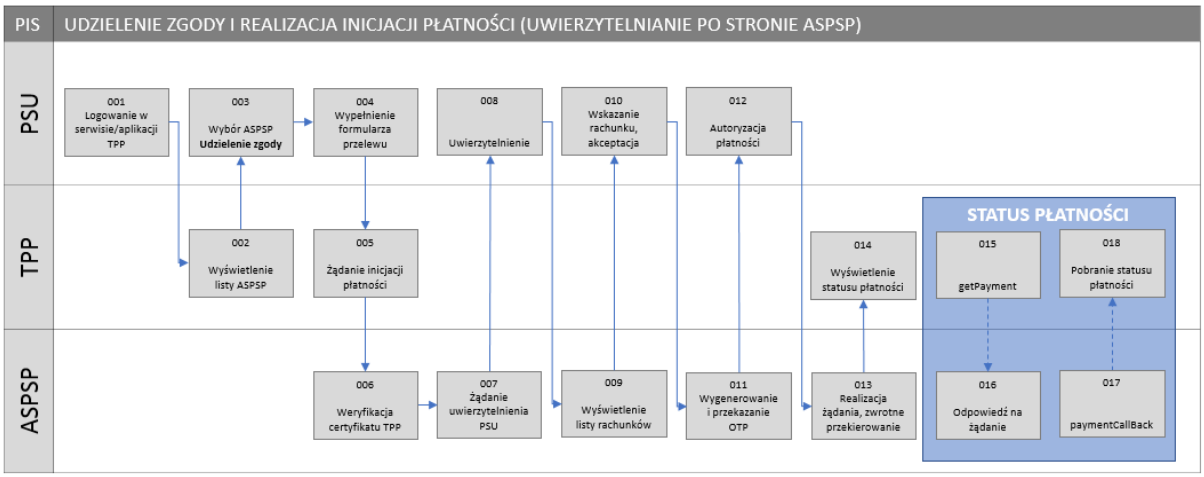
Poniższy diagram odnosi się do procesów opisanych w rozdziale 1.4.4.1.1 (Proces udzielenia zgody PSU na wykonanie usługi PIS – uwierzytelnianie po stronie ASPSP) oraz 3.1.2 (Informacja o statusie transakcji).




***Ilustracja 13: PIS / Udzielenie zgody i realizacja inicjacji płatności oraz pobranie statusu płatności***

### 4.1.2 Inicjacja pojedynczej płatności przez PISP z wykorzystaniem mechanizmu uwierzytelniania po stronie ASPSP z wyborem rachunku po stronie ASPSP

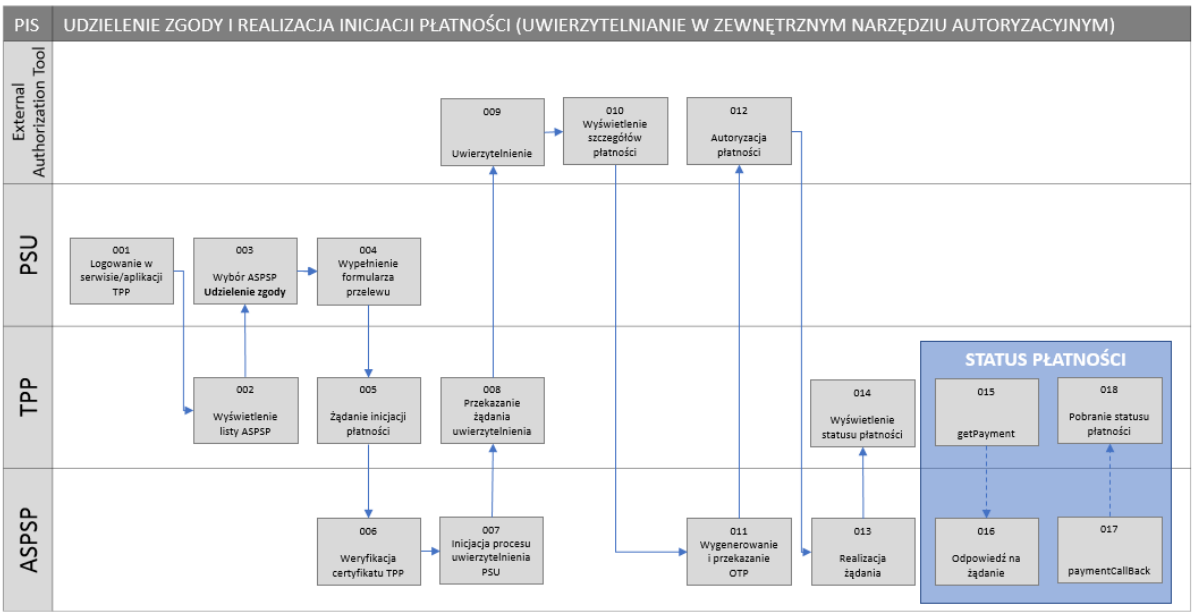
Poniższy diagram odnosi się do procesów opisanych w rozdziale 1.4.4.1.1 (Proces udzielenia zgody PSU na wykonanie usługi PIS – uwierzytelnianie po stronie ASPSP) oraz 3.1.2 (Informacja o statusie transakcji).




*Ilustracja 14: PIS / Udzielenie zgody i realizacja inicjacji płatności (wybór rachunku po stronie ASPSP) oraz pobranie statusu płatności*

### 4.1.3 Inicjacja pojedynczej płatności przez PISP z wykorzystaniem mechanizmu uwierzytelniania w zewnętrznym narzędziu autoryzacyjnym

Poniższy diagram odnosi się do procesów opisanych w rozdziale 1.4.4.1.3 (Proces udzielenia zgody PSU na wykonanie usługi PIS – uwierzytelnianie w zewnętrznym narzędziu autoryzacyjnym) oraz 3.1.2 (Informacja o statusie transakcji).



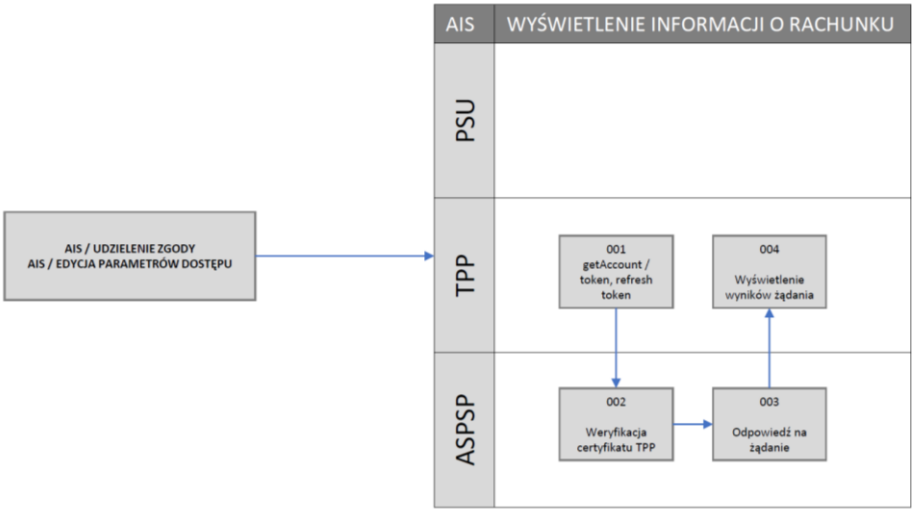

*Ilustracja 15: PIS / Udzielenie zgody i realizacja inicjacji płatności (uwierzytelnienie w zewnętrznym narzędziu autoryzacyjnym) oraz pobranie statusu płatności*

## 4.2 Przypadek Użycia #2: wyświetlenie informacji o rachunku płatniczym przez AISP (AIS)

Wykorzystanie usługi AIS w zakresie Zgodności zaprezentowane w tym Przypadku Użycia polega na pozyskaniu przez TPP występującego w roli AISP informacji na temat rachunku płatniczego PSU, prowadzonego przez ASPSP, w oparciu o stosowne zapisy UUP.

Proces udzielenia zgody na skorzystanie z usługi AIS został opisany w rozdziale 1.4.4.2. W opisanym poniżej Przypadku przyjęto założenie, że PSU udzielił AISP zgody na pobranie danych w określonym zakresie. Zapytanie jest inicjowane przez AISP w imieniu PSU.

Proces ten został wysokopoziomowo pokazany na poniższym diagramie.



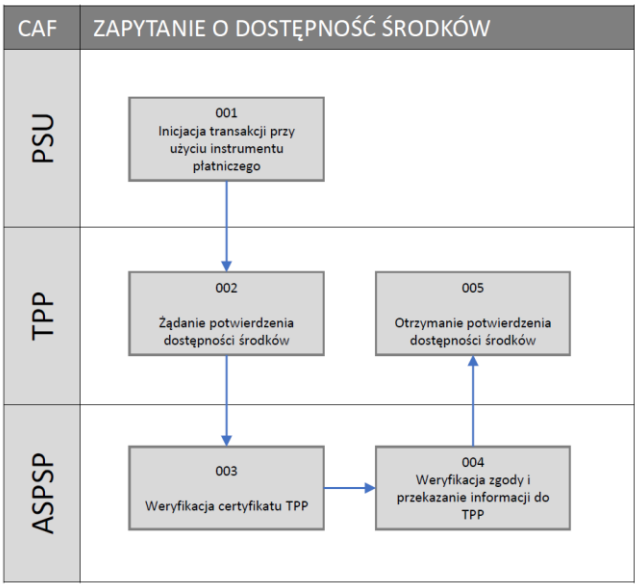

***Ilustracja 16: AIS / Wyświetlenie informacji o rachunku płatniczym przez AISP***

## 4.3 Przypadek użycia #3: zapytanie o dostępność środków przez PIISP (CAF)

Wykorzystanie usługi CAF w zakresie Zgodności zaprezentowane w tym Przypadku Użycia polega na zainicjowaniu przez TPP występującego w roli PIISP zapytania o dostępność środków w kwocie transakcji na rachunku płatniczym PSU, w oparciu o stosowne zapisy UUP.

PSU musi uprzednio wskazać PIISP rachunek płatniczy, który będzie każdorazowo odpytywany o dostępność środków oraz udziela uprzednich zgód na udzielanie odpowiedzi przez ASPSP prowadzącego dany rachunek płatniczy. PSU inicjuje proces biznesowy wymagający weryfikacji, czy na wskazanym uprzednio przez PSU rachunku płatniczym znajdują się dostępne środki w kwocie co najmniej równiej kwocie zapytania. W celu realizacji usługi, PIISP nawiązuje sesję XS2A z ASPSP, dokonuje zapytania i uzyskuje odpowiedź „TAK” lub „NIE”.

Proces ten został wysokopoziomowo pokazany na poniższym diagramie.




***Ilustracja 17: Ilustracja 16: CAF / Zapytanie o dostępność środków***

Proces ten może być wykorzystany np. do dokonywania autoryzacji transakcji instrumentami płatniczymi niepowiązanymi z rachunkiem płatniczym prowadzonym przez wydawcę instrumentu.

# 5 Specyfikacja techniczna PolishAPI

## 5.1 Założenia techniczne

Poniższa tabela prezentuje założenia techniczne przyjęte dla PolishAPI:

| LP | ZAŁOŻENIE | OPIS | UZASADNIENIE |
|---|---|---|---|
| 1 | Bezpośrednia komunikacja TPP-ASPSP | TPP i ASPSP w ramach podstawowego wariantu PolishAPI komunikują się bezpośrednio. | Stosowana architektura peer-to-peer zwiększa bezpieczeństwo, wydajność i pozwala na uniknięcie pojedynczego punktu awarii. |
| 2 | Rola HUB PSD2 | W przypadku korzystania przez ASPSP z usług HUB PSD2 jest to neutralne dla TPP. HUB PSD2 przedstawia się certyfikatem ASPSP, z perspektywy TPP nie ma różnicy czy łączy się z HUB PSD2 czy z ASPSP bezpośrednio. | Ułatwienie wdrożenia API i standardu PolishAPI w sposób wydajny i bezpieczny. |
| 3 | Komunikacja TPP-ASPSP to serwer-serwer | Nie jest dopuszczalna bezpośrednia komunikacja urządzenia klienta (np. aplikacji mobilnej) z serwerami PolishAPI ASPSP. Należy prawnie zobligować TPP do zabezpieczenia kluczy dostępowych (tzw certyfikat dostępowy). W szczególności certyfikaty dostępowe nie mogą być instalowane w aplikacjach mobilnych udostępnianych dla PSU) | |
| 4 | Rozdzielenie kroku wyrażenia zgody klienta od realizacji operacji | Krok wyrażenia zgody klienta na realizację usługibędzie oddzielony od samej realizacji operacji. Jednym ze skutków jest to, iż samo wyrażenie zgody nie rodzi skutków finansowych. | Elastyczność we wdrażaniu nowych usług, w tym Usług Premium. |
| 5 | Zakres PolishAPI | Zakres PolishAPI specyfikuje:<br>- sposób wyrażenia zgody na wykonywanie przez TPP operacji w imieniu klienta<br>- zakres operacji i uprawnień<br>- URL, pod jakim dana usługa jest dostępna<br>- standardowy zakres parametrów per usługa<br>- mechanizmy zabezpieczeń<br>- zasady komunikacji<br>- obsługę błędów<br>PolishAPI nie specyfikuje<br>- pełnego zakresu funkcjonalności, które mają być udostępnione przez ASPSP, oraz które z nich będą w Usługach Zgodności<br>- pełnej specyfikacji pól per usługa dla każdego ASPSP | RTS uzależnia zakres funkcjonalności i zakres danych od zakresu funkcjonalności udostępnianych w bankowości internetowej, która jest różna u każdego ASPSP |

## 5.2 Nawiązanie sesji XS2A

Wykorzystanie przez TPP usług biznesowych (AIS, PIS, CAF), udostępnianych po stronie ASPSP, wymaga nawiązania tzw. sesji komunikacyjnej po stronie rozwiązań technicznych wymienionych podmiotów.

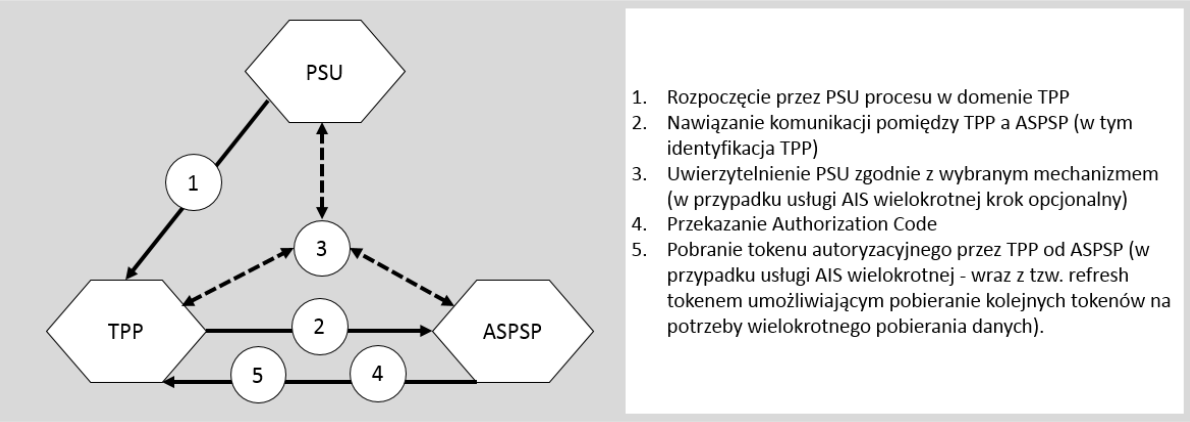
Proces nawiązywania sesji komunikacyjnej z interfejsem XS2A obejmuje żądania i odpowiedzi przekazywane pomiędzy TPP i ASPSP przy użyciu usług technicznych tego interfejsu (AS – Authorization Service), którego efektem jest ustanowienie sesji komunikacyjnej po stronie ASPSP i przekazanie do TPP jej technicznej reprezentacji, w tym meta danych takich jak czas jej ważności.

Nawiązanie sesji komunikacyjnej może uwzględniać konieczność dokonania silnego uwierzytelnienia PSU. Ze względu na wybraną metodę SCA (po stronie *ASPSP* lub tzw. *decoupled*) proces nawiązywania sesji komunikacyjnej może się różnić. Bez względu na wspomniane różnice w metodach SCA, nawiązanie sesji komunikacyjnej opiera się na założeniach standardu OAuth 2.0 w następujących kwestiach:

a) Wymaganym sposobem autoryzacji dostępu do zasobów ASPSP, udostępnianych poprzez interfejs biznesowy XS2A, jest zwrócenie przez serwer po stronie ASPSP, w odpowiedzi na żądanie wysłane przez TPP, jednorazowego kodu autoryzacyjnego (*authorisation code*) – w rozumieniu Art. 4 RTS, który zostanie w kolejnym kroku wykorzystany przez TPP do uzyskania tzw. tokena dostępu (ang. *access token*) - zgodnie z zapisami standardu OAuth 2.0
b) Parametr *state*, wysyłany przez TPP w żądaniu autoryzacji (punkt a), musi być unikalny dla każdego procesu autoryzacji realizowanego przez danego TPP
c) Sugeruje się aby po stronie serwera ASPSP jednorazowy kod uwierzytelniający oraz token dostępu były identyfikatorem zasobu w bazie danych, w którym wskazane dane tego zasobu posłużą do identyfikacji PSU, na rzecz którego generowany jest token dostępu lub realizowana jest określona operacja biznesowa.
   Zastosowanie tzw. *stateless token* (np. JWT Token - RFC 7519) powinno być stosowane tylko w przypadku, gdy ujawnienie danych o kliencie (w tym identyfikatora) ASPSP jest zgodne z polityką bezpieczeństwa
d) Wraz z tokenem dostępu przekazywany jest do TPP również parametr *scope* (taki sam, jak w żądaniu wysłanym przez TPP)

Posługiwanie się przez TPP ważną sesją komunikacyjną jest warunkiem koniecznym otrzymywania poprawnych odpowiedzi na żądania wysyłane do usług biznesowych interfejsu XSS2A.

Schemat przebiegu procesu nawiązywania sesji komunikacyjnej XS2A przedstawia poniższy diagram.




***Ilustracja 18: Wysokopoziomowy diagram nawiązywania sesji XS2A***

Realizacja transakcji w ramach usług biznesowych interfejsu XS2A odbywa się w ramach dedykowanej, odrębnej sesji komunikacyjnej, przy czym dla wybranych metod w ramach usług AIS i PIS, i przy spełnieniu warunków określonych w częściach biznesowej i technicznej dyrektywy PSD2, dopuszczalne jest wielokrotne wykorzystanie tej samej sesji komunikacyjnej, przy wysyłaniu żądań do usług biznesowych interfejsu XS2A, bez konieczności każdorazowego przeprowadzenia procedury SCA dla PSU oraz przy zachowaniu czasu ważności tej sesji komunikacyjnej, po którym zostanie ona automatycznie unieważniona przez ASPSP.

Usługa techniczna interfejsu XS2A, może oferować alternatywny, automatyczny (nie wymagający interakcji z PSU) mechanizm nawiązania sesji komunikacyjnej, którym jest tzw. *refresh token*. Mechanizm ten umożliwia odnowienie uprzednio unieważnionej przez ASPSP sesji komunikacyjnej, bez konieczności ponownego przeprowadzenia procedury SCA, w oparciu o osobny identyfikator sesji (*refresh token*) przekazywany do TPP w odpowiedzi na żądanie nawiązania pierwotnej sesji komunikacyjnej. Ten mechanizm nawiązywania sesji komunikacyjnej może być użyty tylko dla tych usług biznesowych interfejsu XS2A, dla których zostało to dozwolone zapisami regulacyjnymi dyrektywy PSD2, np. AIS wielokrotny. Mechanizm ten jest dodatkowo opisany w rozdziale 11.3.

## 5.3 Definicja tokena dostępu

Token dostępu (ang. *access token*) stanowi techniczną reprezentację sesji komunikacyjnej, o ustalonym czasie ważności, nawiązanej pomiędzy TPP i ASPSP w kontekście ściśle określonego PSU i dla ściśle określonego zakresu usług i zasobów po stronie ASPSP, do których TPP uzyskał dostęp. Token dostępu jest ciągiem znaków, którego rolą jest potwierdzenie autoryzacji dostępu do zabezpieczonych zasobów, udostępnianych przez usługi interfejsu XS2A. Token dostępu może posiadać różne formaty i sposoby interpretacji. Ostateczne właściwości tokena dostępu zależą od systemów autoryzacyjnych po stronie ASPSP, które dokonują implementacji standardu Polish API.

Zgodnie z regulacjami opisanymi w RTS dyrektywy PSD2, w zależności od usługi biznesowej interfejsu XS2A (AIS, PIS), dla której została ustanowiona sesja komunikacyjna, token dostępu może być wykorzystywany jednokrotnie lub wielokrotnie, zanim ulegnie on unieważnieniu przez ASPSP, co będzie się wiązało z koniecznością ponownego przeprowadzenia procedury SCA dla PSU, w przypadku zamiaru ponownego skorzystanie z tej usługi.

## 5.4 Wzajemne uwierzytelnienie TPP i ASPSP

Uwierzytelnienie wzajemne TPP i ASPSP następuje na podstawie certyfikatów X.509v3 wystawionych przez zaufaną stronę trzecią. Zaufaną trzecią stroną, w szczególności może być instytucja pełniąca funkcję Hub tożsamości. Mogą nim być także inne podmioty powiązane relacjami zaufania opartymi o mechanizmy infrastruktury klucza publicznego.

Wszystkie operacje składające się na wyszczególnione i opisane w standardzie przepływy są możliwe jedynie w sytuacji poprawnego uwierzytelnienia w procesie, na który składa się wzajemne uwierzytelnienie serwera jak i klienta (*Mutual authentication*). TPP i ASPSP mogą występować w roli zarówno serwera, jak i klienta, ale w każdym przypadku wymagane jest wzajemne uwierzytelnienie stron komunikacji.

Opis infrastruktury klucza publicznego wykorzystywanej na potrzeby uwierzytelnień stron (TPP, ASPSP, PISP) nie stanowi części standardu PolishAPI. Powinien być opisany w oddzielnych dokumentach (standardach roboczych) z uwzględnieniem struktury relacji zaufania pomiędzy urzędami certyfikacji oraz interoperacyjności PolishAPI z innymi tego rodzaju rozwiązaniami działającymi w innych krajach.

## 5.5 Protokół komunikacyjny

Jako protokół komunikacyjny zastosowany zostanie HTTP/2 lub HTTP 1.1, zabezpieczony za pomocą protokołu TLS 1.2+ z wzajemnym uwierzytelnianiem klienta i serwera za pomocą certyfikatów X.509v3 (*Mutual authentication*). Ze względu na wymóg zapewnienia niezaprzeczalności (podpisywanie żądań i odpowiedzi) w komunikacji http stosowana będzie jedynie metoda POST.

## 5.6 Schemat nazewnictwa zasobów

Usługi PolishAPI będą udostępniane pod adresami zgodnymi z następującym wzorcem:

https://{domenaDNS}/{v{numerWersjiZasobu 1}/{nazwaZasobu 1}}/.../v{numerWersjiZasobu n}/{nazwaZasobu n}

Opis pól:

a) Domena DNS/adres – pod którym ASPSP udostępnia usługi PolishAPI (informacja udostępniana w rejestrze PSD2)
b) Numer wersji zasobu – numer wersji zgodnie ze specyfikacją PolishAPI (liczba przed kropką) i kolejnej wersji interfejsu w ramach danego ASPSP (liczba po kropce)
c) Nazwa zasobu – nazwa zasobu, którego dotyczy zapytanie; dopuszczalne są ścieżki zagnieżdżające zasoby, np. /v{numer wersji zasobu accounts}/accounts/v{numer wersji zasobu transactionsDone}/transactionsDone

## 5.7 Kanoniczny model danych

W celu unifikacji typów danych został zaproponowany kanoniczny model danych przedstawiony w tabeli. Szczegółową definicję modelu zawiera Załącznik nr 1.

| NAZWA KLASY | PRZEZNACZENIE | API |
|---|---|---|
| AccountIban | Numer konta | Podstawowe |
| AccountBaseInfo | Klasa zawierająca podstawowe dane o rachunku | Podstawowe |
| AccountForeign | Numer konta w formacie dla przelewów zagranicznych | Podstawowe |
| AccountInfo | Klasa informacji o koncie | Podstawowe / CallBack |
| AccountInfoRequest | Klasa zapytania o pojedynczy rachunek | Podstawowe |
| AccountResponse | Klasa odpowiedzi na zapytania o konto PSU | Podstawowe |
| AccountsRequest | Klasa zapytania o rachunki | Podstawowe |
| AccountsRequest | Klasa żądania zwrotnego z listą rachunków | CallBack |
| AccountsResponse | Klasa odpowiedzi na zapytania o wiele kont PSU | Podstawowe |
| AddPaymentResponse | Klasa odpowiedzi na żądanie inicjacji płatności | Podstawowe |
| Address | Klasa zawierająca dane adresu pocztowego | Podstawowe / CallBack |
| AuthorizationDataRequest | Klasa ze zwrotną informacją z kodem autoryzacyjnym zgodnym z standardem OAuth 2.0 dla metody autohorization code | CallBack |
| AuthorizeRequest | Klasa zawierająca dane wymagane do autoryzacji TPP | Podstawowe |
| BankAccountInfo | Klasa zawierająca dane banku | Podstawowe / CallBack |
| Bank | Klasa zawierająca dane banku wykorzystywana w | Podstawowe / |

| | | |
|---|---|---|
| | żądaniach AIS | CallBack |
| CallBackResponse | Klasa bazowa dla odpowiedzi na żądania zwrotne | CallBack |
| ConfirmationOfFundsRequest | Klasa zapytania o dostępne środki płatnicze na rachunku | Podstawowe |
| ConfirmationOfFundsResponse | Klasa odpowiedzi na zapytanie o dostępne środki płatnicze na rachunku | Podstawowe |
| CurrencyRate | Klasa zawierająca kursy przewalutowań | Podstawowe |
| DeleteConsentRequest | Klasa zawierająca dane umożliwiające identyfikację zgód do usunięcia po stronie ASPSP | Podstawowe |
| DictionaryItem | Klasa zawierająca dane elementu/pozycji słownika | Podstawowe / CallBack |
| EatCodeRequest | Klasa zapytania o uzyskanie kodu autoryzacyjnego zgodnego z OAuth 2.0 na podstawie przekazanego kodu jednorazowego, wygenerowanego w EAT | Podstawowe |
| Error | Klasa informacji zawierająca dane o zwracanym błędzie | Podstawowe / CallBack |
| GetPaymentResponse | Klasa odpowiedzi na zapytanie o płatność | Podstawowe |
| Map | Klasa mapy <string, string> | Podstawowe / CallBack |
| NameAddress | Klasa zawierająca dane nazwy i adresu w postaci czterech lini danych | Podstawowe / CallBack |
| PageInfo | Klasa zawierająca dane pozwalające na korzystanie z mechanizmu stronicowania | Podstawowe |
| PaymentDomesticRequest | Klasa zapytania dotyczącego inicjacji standardowego przelewu krajowego | Podstawowe |
| PaymentEEARequest | Klasa zapytania dotyczącego inicjacji przelewu zagranicznego SEPA | Podstawowe |
| PaymentInfo | Klasa informacji o płatność | Podstawowe |
| PaymentNonEEARequest | Klasa zapytania dotyczącego inicjacji przelewu zagranicznego innego niż SEPA | Podstawowe |
| PaymentRequest | Klasa zapytania o status płatności | Podstawowe |
| PaymentStatus | Słownik statusów płatności | Podstawowe / CallBack |
| PaymentStatusInfoRequest | Klasa żądania zwrotnego z informacjami o statusie płatności | CallBack |
| PaymentTaxRequest | Klasa zapytania dotyczącego inicjacji przelewu podatkowego | Podstawowe |
| PaymentsRequest | Klasa zapytania o statusy wielu płatności | Podstawowe |
| PaymentsResponse | Klasa odpowiedzi na pytanie o wiele płatności | Podstawowe |
| Payor | Klasa informacji o płatniku do ZUS i US | Podstawowe |
| RecipientPIS | Klasa zawierająca dane odbiorcy używana w żądaniach PIS | Podstawowe |
| RecipientPISTax | Klasa zawierająca dane odbiorcy używana w żądaniach PIS dla operacji podatkowych | Podstawowe |
| RecipientPISForeign | Klasa zawierająca dane odbiorcy używana w żądaniach PIS dla operacji zagranicznych | Podstawowe |
| RequestHeader | Klasa zawierająca informacje o PSU | Podstawowe |
| RequestHeader | Klasa zawierająca metadane żądania do interfejsu wywołań zwrotnych | CallBack |
| RequestHeaderAIS | Klasa zawierająca informacje o PSU dla zapytań do usługi AIS interfejsu XS2A | Podstawowe |
| RequestHeaderAISCallback | Klasa zawierająca informacje o PSU dla zapytań | Podstawowe |

| | | |
|---|---|---|
| | do usługi AIS interfejsu XS2A umożliwiających przesłanie odpowiedzi w formie żądania do interfejsu wywołań zwrotnych | |
| RequestHeaderCallback | Klasa zawierająca informacje o PSU dla zapytań umożliwiających przesłanie odpowiedzi w formie żądania do interfejsu wywołań zwrotnych | Podstawowe |
| RequestHeaderWithoutToken | Klasa zawierająca informacje o PSU dla zapytań nie wymagających tokena dostępu | Podstawowe |
| RequestHeaderWithoutTokenCallback | Klasa zawierająca informacje o PSU dla zapytań nie wymagających tokena dostępu i umożliwiających przesłanie odpowiedzi w formie żądania do interfejsu wywołań zwrotnych | Podstawowe |
| ResponseHeader | Klasa zawierająca metadane odpowiedzi | Podstawowe |
| ResponseHeader | Klasa zawierająca metadane odpowiedzi | CallBack |
| SenderPIS | Klasa zawierająca dane nadawcy używana w żądaniach PIS | Podstawowe |
| SenderRecipient | Klasa zawierająca dane nadawcy/odbiorcy używana w żądaniach AIS | Podstawowe / CallBack |
| SocialSecurityPayor | Klasa zawierająca informację o płatniku składek do ZUS | Podstawowe |
| TokenRequest | Klasa zawierające dane wymagane do uzyskania tokena dostępu | Podstawowe |
| TokenResponse | Klasa odpowiedzi zawierająca m.in. token dostępu | Podstawowe |
| TransactionDetailRequest | Klasa zapytania o pojedynczą transakcję | Podstawowe |
| TransactionDetailResponse | Klasa odpowiedzi na zapytanie o transakcję | Podstawowe |
| TransactionsDoneInfoResponse | Klasa odpowiedzi zawierająca listę transakcji done | Podstawowe |
| TransactionDoneInfoRequest | Klasa żądania zwrotnego zawierająca listę transakcji done | CallBack |
| TransactionHoldInfo | Klasa zawierające dane o blokadach na rachunku | Podstawowe |
| TransactionHoldInfoResponse | Klasa odpowiedzi zawierająca listę blokad na rachunku | Podstawowe |
| TransactionHoldInfoRequest | Klasa żądania zwrotnego zawierająca listę blokad na rachunku | CallBack |
| TransactionHoldRequest | Klasa zapytania o blokady na rachunku | Podstawowe |
| TransactionInfo | Klasa opisująca zaksięgowaną transakcję płatniczą | Podstawowe / CallBack |
| TransactionInfoBase | Klasa bazowa opisująca transakcję płatniczą | Podstawowe / CallBack |
| TransactionInfoCard | Klasa reprezentująca informacje o karcie w ramach transakcji | Podstawowe |
| TransactionInfoRequest | Klasa zapytania o transakcje | Podstawowe |
| TransactionInfoRequestBase | Bazowa klasa dla zapytań o transakcje | Podstawowe |
| TransactionInfoTax | Klasa informacji danych dla przelewu do Organu Podatkowego/Izby Celnej | Podstawowe |
| TransactionInfoZUS | Klasa informacji danych dla przelewu do ZUS | Podstawowe |
| TransactionPendingInfo | Klasa opisująca oczekującą transakcję płatniczą | Podstawowe / CallBack |
| TransactionPendingInfoResponse | Klasa odpowiedzi zawierająca listę transakcji pending | Podstawowe |
| TransactionPendingInfoRequest | Klasa żądania zwrotnego zawierająca listę transakcji pending | CallBack |

| | | |
|---|---|---|
| TransactionRejectedInfo | Klasa opisująca odrzuconą transakcję płatniczą | Podstawowe / CallBack |
| TransactionRejectedInfoResponse | Klasa odpowiedzi zawierająca listę transakcji rejected | Podstawowe |
| TransactionRejectedInfoRequest | Klasa żądania zwrotnego zawierająca listę transakcji rejected | CallBack |
| TransferData | Klasa zawierająca dane przelewu | Podstawowe |
| TransferDataBase | Klasa zawierająca ogólne dane przelewu | Podstawowe |
| TransferDataCurrencyRequired | Klasa zawierająca dane przelewu z wymaganą informacją o walucie | Podstawowe |

## 5.8 Operacje

Ze względu na wymóg zapewnienia niezaprzeczalności w komunikacji http stosowana będzie jedynie metoda POST pozwalającą na złożenie podpisu w formacie *JWS Signature*. W ramach operacji kontekst konkretnego użytkownika określany jest na podstawie tokena dostępowego. Ta zasada dotyczy zarówno żądań wysyłanych przez TPP do interfejsu XS2A ASPSP, jak i żądań przesyłanych z ASPSP do interfejsu XS2A wywołań zwrotnych, udostępnianego przez TPP.

## 5.9 Sortowanie

Zwracane rekordy sortowane są chronologicznie (odwrotnie) wg daty transakcji.

## 5.10 Filtrowanie

Filtrowanie w usłudze AIS odbywa się przez ustawienie odpowiednich właściwości w obiekcie klasy TransactionInfoRequest:

a) transactionIdFrom -transakcje od podanego identyfikatora transakcji „chronologicznie”
b) transactionDateFrom – początkowa data transakcji żądanego zakresu danych
c) transactionDateTo – końcowa data transakcji żądanego zakresu danych
d) bookingDateFrom – początkowa data księgowania żądanego zakresu danych
e) bookingDateTo – końcowa data księgowania żądanego zakresu danych
f) type – CREDIT lub DEBIT
g) minAmount – minimalna kwota operacji w żądanym zakresie danych
h) maxAmount – maksymalna kwota operacji w żądanym zakresie danych

## 5.11 Stronicowanie

Wyniki zapytań zawierające wiele rekordów (gdzie wiele > 100) powinny być stronicowane. Kolejne strony będą pobierane poprzez ustawianie atrybutu pageId w klasie TransactionInfoRequest. Atrybut pageId należy ustawić na wartość zwróconą w klasie PageInfo poprzedniego żądania. Dla nawigacji w przód nextPage, dla nawigacji w tył previousPage. Liczba rekordów na stronie definiowana jest za pomocą atrybutu perPage w klasie TransactionInfoRequest. Sugeruje się maksymalna wartość, jaką można ustawić w parametrze perPage to 100. Numerowanie stron rozpoczyna się od 1. Pominięcie parametru pageId zwróci pierwszą stronę.

## 5.12 Statusy odpowiedzi

Statusy techniczne będą zwracane poprzez następujące kody http:

| STATUS | OPIS |
|---|---|
| 200 OK | Operacja się powiodła |
| 304 Not Modified | Używane, jeżeli użyto nagłówków cache'owania |
| 400 Bad Request | Zapytanie jest niepoprawne syntaktycznie |
| 401 Unauthorized | Niepoprawnie uwierzytelniony użytkownik |
| 403 Forbidden | Błąd autoryzacji (brak uprawnień dostępu do zasobu) |
| 404 Not Found | Odwołanie do nieistniejącego zasobu |
| 405 Method Not Allowed | Użycie niewłaściwej metody – metoda zawarta w żądaniu nie jest dozwolona dla wskazanego zasobu (Używany jest tylko POST) |
| 406 Not Acceptable | Nieprawidłowy nagłówek accept w zapytaniu (serwer nie jest w stanie obsłużyć) |
| 415 Unsupported Media Type | Jeżeli nieprawidłowy content type został ustawiony w zapytaniu |
| 422 Unprocessable Entity | Błąd walidacji |
| 429 Too Many Requests | Zapytanie odrzucone ze względu na przekroczenie maksymalnej liczby żądań dostępu do zasobu |

## 5.13 Nagłówki HTTP

W zapytaniach zostaną użyte następujące nagłówki http:

| NAGŁÓWEK | TYP | OPIS |
|---|---|---|
| Authorization | String | Nagłówek uwierzytelnienia (używany przy przesyłaniu tokenu). Wartość nagłówka Authorization powinien składać się z „type" + „credentials", gdzie w przypadku podejścia z wykorzystaniem tokenu „type" powinien mieć wartość „Bearer". |
| Date | Date | Timestamp żądania w formacie RFC 5322 date and time format. |
| Accept | Content type | Należy ustawić na application/json W przeciwnym razie aplikacja powinna zwrócić 406 Not Acceptable HTTP. |
| Accept-Encoding | Gzip, deflate | Operacja powinna wspierać GZIP oraz kodowanie DEFLATE, może również zwrócić dane nieskompresowane. |
| Accept-Language | „pl", „en", etc. | Określa, preferowany język w jakim ma być zwrócona odpowiedź. Operacja nie musi wspierać tego nagłówka |
| Accept-Charset | Charset type like „UTF-8" | UTF-8 |
| Content-Type | application/json | Należy ustawić na application/json. W przeciwnym razie operacja zwraca 415 Unsupported Media Type HTTP status code |
| X-JWS-SIGNATURE | String | Podpis JWS Signature (Detached) |

Nagłówki odpowiedzi:

| NAGŁÓWEK | WYMAGALNOŚĆ | OPIS |
|---|---|---|
| Date | Tak | Timestamp na bazie czasu serwera GMT zgodnie z RFC 5322 |
| Content-Type | Tak | application/json |
| Content-Encoding | Tak | GZIP lub DEFLATE |
| Expires | Nie | Określa politykę cacheowania dla wolnozmiennych obiektów np. Expires: Mon, 25 Jun 2012 21:31:12 GMT |
| Size | Tak | Wielkość odpowiedzi w bajtach |
| ETag | Nie | Identyfikator wersji zasobu |
| Last-Modified | Nie | Data ostatniej modyfikacji zasobu |
| X-JWS-SIGNATURE | Tak | Podpis JWS Signature (Detached) |
| Location | Nie | URI, na który powinno nastąpić przekierowanie żądania autoryzacji, w przypadku odpowiedzi 302 od ASPSP |

## 5.14 Format wiadomości

Formatem wymiany danych będzie JSON z kodowaniem UTF-8. Wszystkie komunikaty mają zdefiniowaną JSON schema draft #4. Nazwy parametrów będą zapisane camelCase.

## 5.15 Podstawowe formaty danych

| FORMAT | FORMAT JSON | OPIS |
|---|---|---|
| Tekst | String | Tekst kodowany w UTF-8 |
| Daty | String | Zgodnie z ISO8601.<br>Data i czas będą reprezentowane w postaci YYYY-MM-DD na YYYY-MM-DDThh:mm:ss.ccczzzzzz z obowiązkowym podaniem strefy czasowej<br>Oznaczenia:<br>YYYY – rok, MM – miesiąc, DD – dzień, hh – godzina, mm – minuta, ss – sekunda, ccc – milisekundy (opcjonalne) zzzzzz – np. +02:00 lub Z dla oznaczenia czas uniwersalnego<br>Przykładowo 2016-10-10T12:00:05.342+01:00 |
| Kwoty | String | Zapisane jako liczby ze znakiem oddzielającym część całkowitą od części ułamkowej 2 miejsca (znak kropki). Przy kwotach dodatnich nie dodajemy żadnych dodatkowych znaków. Przy liczbach ujemnych dodajemy przed liczbą „-„ |
| Liczba całkowita | Number | Liczby całkowite reprezentowane są bez separatorów grupowych |
| Liczba rzeczywista | String | Liczby rzeczywiste reprezentowane są bez separatorów grupowych i ze znakiem ‘.’ jako separatorem dziesiętnym |
| Oznaczenia krajów | String | Zgodnie z ISO 3166 |
| Waluty | String | Oznaczenia walut zgodnie z ISO 4217 |
| Numery rachunków | String | Numery IBAN zgodnie z ISO 13616 |
| Identyfikatory banków | String | Bank Identifier Codes (BIC) zgodnie z ISO 9362 |
| Wartość logiczna | boolean | Flagi i znaczniki logiczne, które mogą przyjmować jedną z dwóch wartości: true lub false |

# 6 Bezpieczeństwo informacji

Niniejszy rozdział obejmuje ogólne wymagania bezpieczeństwa, istotne z punktu widzenia kreowania standardu oraz projektowania na jego podstawie ekosystemu rozwiązań informatycznych zgodnych z PolishAPI. Szczegółowe wymagania bezpieczeństwa, obejmujące dodatkowo kwestie bezpieczeństwa implementacji, operacji i utrzymania systemów opartych na PolishAPI zostaną opisane w oddzielnym dokumencie, a jego opracowanie zostanie poprzedzone przygotowaniem szczegółowego modelu zagrożeń. Będą zatem odpowiedzią na zidentyfikowane konkretne zagrożenia, miejsca potencjalnej materializacji tych zagrożeń, a także ocenę poziomu istotności oraz prawdopodobieństwa i wpływu przypadków materializacji zagrożeń na bezpieczeństwo i ciągłość działania ekosystemu PolishAPI.

Poszczególne komponenty systemów informatycznych opartych na PolishAPI powinny mieć jasno zdefiniowany rozdział pomiędzy warstwą danych, warstwą kontrolera i warstwą prezentacyjną. Komponenty powinny być odseparowane od siebie poprzez zdefiniowane zabezpieczenia takie jak segmentacja sieci lub reguły zapory sieciowej.

## 6.1 Uwierzytelnienie TPP

Podmioty TPP muszą zostać poprawnie uwierzytelnienie przed udzieleniem im dostępu do interfejsu XS2A tak, aby zapewnić wysoki poziom ochrony zarówno przed podszyciem się nieuprawnionych podmiotów pod właściwych TPP, jak i przed nieuprawnioną eskalacją poziomu autoryzacji przez TPP mających legalny dostęp do interfejsu XS2A. Uwierzytelnienie następuje w oparciu o certyfikaty klucza publicznego w procesie wzajemnego uwierzytelnienia (*Mutual authentication*) za pomocą protokołu TLS 1.2+.

Błędy uwierzytelnienia muszą skutkować odmową dostępu do interfejsu XS2A.
Dane uwierzytelniające użytkownika oraz sesję, a także tokeny do autoryzowania operacji nie mogą być przekazywane w postaci parametrów URI.

## 6.2 Autoryzacja TPP

Autoryzacja TPP musi być oparta na modelu RBAC (*Role Based Access Control*), w którym poziom i zakres dostępu do poszczególnych zasobów API zależy od roli użytkownika PolishAPI.

Użycie poszczególnych metod musi być autoryzowane w taki sposób, aby uprawnienia były zależne od roli użytkownika. W szczególności, poziom i zakres autoryzacji powinien być różny dla TPP w zależności od zakresu ich uprawnień.

## 6.3 Autoryzacja PSU dla operacji wykonywanych przez TPP

Niezależnie od zastosowanego mechanizmu uwierzytelniania PSU w ramach usług AIS/PIS/CAF zakłada się, iż proces ten kończy się wydaniem przez ASPSP *access tokenu* zdefiniowanego w rozdziale 5.3 specyfikacji. Zlecanie operacji przez TPP odbywa się zawsze z wykorzystaniem ważnego *access tokenu.*

## 6.4 Bezpieczeństwo w przypadku aplikacji mobilnych

Ze względu na bezpieczeństwo w modelu wykorzystującym mechanizm uwierzytelniania po stronie ASPSP, przekierowanie na stronę ASPSP i z powrotem na stronę TPP będzie odbywało się w przeglądarce systemowej (nie będą dopuszczone przeglądarki inne niż systemowa, nie będzie dopuszczone stosowanie WebView) a nie w samej aplikacji mobilnej. TPP może zarejestrować

odpowiedni URL w systemie operacyjnym urządzenia, aby po przekierowaniu do TPP automatycznie wznowić aplikację mobilną.

## 6.5 Walidacja i zapewnienie integralności danych

Dane muszą być poddane procedurom walidacji w kontekście typu zmiennych, zakresu i wzorca dopuszczalnych wartości. W szczególności ustrukturyzowane dane JSON muszą być parsowane zgodnie z formalnymi procedurami walidacyjnymi z zastosowaniem podejścia opartego na listach dopuszczalnych wartości (*white list*). Walidacji muszą być także poddane nagłówki Content-type i Accept (application/json) na Zgodność wartości nagłówka z rzeczywistą treścią komunikatu HTTP.

Podczas walidacji musi być zwalidowany podpis cyfrowy w nagłówku (X-JWS-SIGNATURE) w kontekście danych przekazywanych zarówno w żądaniach jak i odpowiedziach protokołu http, wymienianych w komunikacji na linii ASPSP-TPP.  danych. Należy podkreślić, iż ta reguła ma zastosowanie również w przypadku komunikacji inicjowanej przez stronę ASPSP, w przypadku wykorzystania interfejsu XS2A wywołań zwrotnych, który jest udostępniany przez TPP.

W razie błędów walidacji treści Content-type i Accept powinien zostać zwrócony komunikat http numer 406 (Not Acceptable).

Błędy walidacji danych wejściowych muszą być rejestrowane w logach.

Błędy walidacji muszą być sygnalizowane komunikatem HTTP 400 (Bad Request) i dane muszą być odrzucane.

Dane niezwalidowane bądź niepoprawnie zwalidowane muszą być odrzucane.

## 6.6 Kryptografia

Komunikacja przez PolishAPI musi być zabezpieczona kryptograficznie na dwóch poziomach:

a) Na poziomie transportu za pomocą(https/TLS). Renegocjacja parametrów połączenia TLS musi być wykonywana bezpiecznie, zgodnie z RFC 5746
b) Na poziomie komunikatu do zapewniania niezaprzeczalności należy zastosować podpis JSON Web Signature (JWS https://tools.ietf.org/html/rfc7515) sygnatura podpisu musi znajdować się w nagłówku X-JWS-SIGNATURE

Każda ze stron komunikacji (TPP, ASPSP) musi posiadać własne unikatowe dwie pary kluczy (do transmisji i podpisu).

Do zabezpieczania transmisji na poziomie https oraz podpisu JWS-SIGNATURE muszą być zastosowane odrębne certyfikaty. Dla https certyfikat musi posiadać rozszerzone użycie klucza (*Client Authentication*) dla podpisu (*Digital signature*).

Certyfikaty użyte do zestawienia transmisji oraz podpisu muszą zostać walidowane pod względem:

a) Ważności (daty ważności certyfikatu od i do)
b) Braku odwołania (crl/ocsp)
c) Weryfikacji ścieżki (https://tools.ietf.org/html/rfc4158)

Informacje szczególnie wrażliwe, w tym poświadczenia tożsamości oraz klucze autoryzacyjne nie mogą podlegać buforowaniu oraz zapisywania w logach.

Certyfikaty powinny być wydawane z uwzględnieniem specyfikacji ETSI TS 119 495.

## 6.7 Ochrona przed nadużyciami API

Implementacja API powinno uwzględniać mechanizmy ochrony przez nadmiarem żądań ze strony użytkowników (uprawnionych i nieuprawnionych), w szczególności celowo wygenerowanych z zamiarem spowodowania niedostępności zasobu (DoS/DDoS), przez zastosowanie mechanizmów limitujących liczbę obsługiwanych żądań w jednostce czasu. Wartości limitów winny być ustalane na podstawie rozpoznania konkretnych warunków operacyjnych. Limity tego rodzaju powinny podlegać parametryzacji. Mierzenie liczby żądań dostępu do zasobów powinno bazować na zastosowaniu jednoznacznie identyfikującego danego TPP klucza (Klasa RequestHeader.tppID) oraz liczników zaimplementowanych per TPP po stronie serwera. Przekroczenie limitów musi być sygnalizowane komunikatem HTTP numer 429 (*Too Many Requests*).

Zabezpieczenia powinny być zrealizowane w oparciu o zalecenia *OWASP – REST Security Cheat Sheet* (https://www.owasp.org/index.php/REST_Security_Cheat_Sheet).

## 6.8 Logowanie informacji audytowych

Zaleca się, aby źródła czasu wszystkich podmiotów korzystających z PolishAPI powinny być synchronizowane, aby zapewnić, że wpisy w logach mają poprawny czas.

Logowanie kluczowych operacji biznesowych powinno zapewniać niezaprzeczalność i integralność wpisów przez wykorzystanie danych z podpisu JWS Signature.

Log powinien zawierać niezbędne informacje, które pozwolą na precyzyjną analizę czasową w przypadku wystąpienia zdarzenia pozwalającą na złączenie poszczególnych wpisów w jedną transakcje. Elementem łączącym poszczególne wpisy może być np. skrót z tokenu autoryzacyjnego.

# 7 Opis techniczny procesu uwierzytelniania i autoryzacji

## 7.1 Parametry scope oraz scope_details

Parametr scope definuje następujące zakresy dostępowe (odpowiadające poszczególnym zasobom (*paths*) specyfikacji:

- ais:accounts: Uprawnienie do wykonania AIS-Accounts
- ais:account: Uprawnienie do wykonania AIS-Account
- ais:holds: Uprawnienie do wykonania AIS-Holds – wymaga jednoczesnego wystąpienia o zgodę na dostęp do rachunku czyli ais:account
- ais:transactionsDone: Uprawnienie do wykonania AIS-TransactionsDone – wymaga jednoczesnego wystąpienia o zgodę na dostęp do rachunku czyli ais:account
- ais:transactionsPending: Uprawnienie do wykonania AIS-TransactionsPending – wymaga jednoczesnego wystąpienia o zgodę na dostęp do rachunku czyli ais:account
- ais:transactionsRejected: Uprawnienie do wykonania AIS-TransactionsRejected – wymaga jednoczesnego wystąpienia o zgodę na dostęp do rachunku czyli ais:account
- ais:transactionDetail: Uprawnienie do wykonania AIS-TransactionDetail – wymaga jednoczesnego wystąpienia o zgodę na dostęp do rachunku czyli ais:account oraz jednej lub wielu zgód na dostęp do historii transakcji czyli ais:holds, ais:transactionsDone, ais:transactionsPending, ais:transactionsRejected
- pis:multiplePayments: Uprawnienie do wykonania PIS-MultiplePayments
- pis:payment: Uprawnienie do wykonania PIS-Payment
- pis:domestic: Uprawnienie do wykonania PIS-Domestic
- pis:EEA: Uprawnienie do wykonania PIS-EEA
- pis:nonEEA: Uprawnienie do wykonania PIS-NonEEA
- pis:tax: Uprawnienie do wykonania PIS-Tax
- CAF:confirmationOfFunds: Uprawnienie do wykonania CAF-ConfirmationOfFunds

W przypadku przekazywania w parametrze scope wielu identyfikatorów zgód powinny być one oddzielone znakiem spacji np.:

```
{
        ”scope” : ”ais:account ais:transactionsDone”
}
```

Parametr scope_details określa zakresy czasowe, ograniczenia, szczegóły danego uprawnienia:

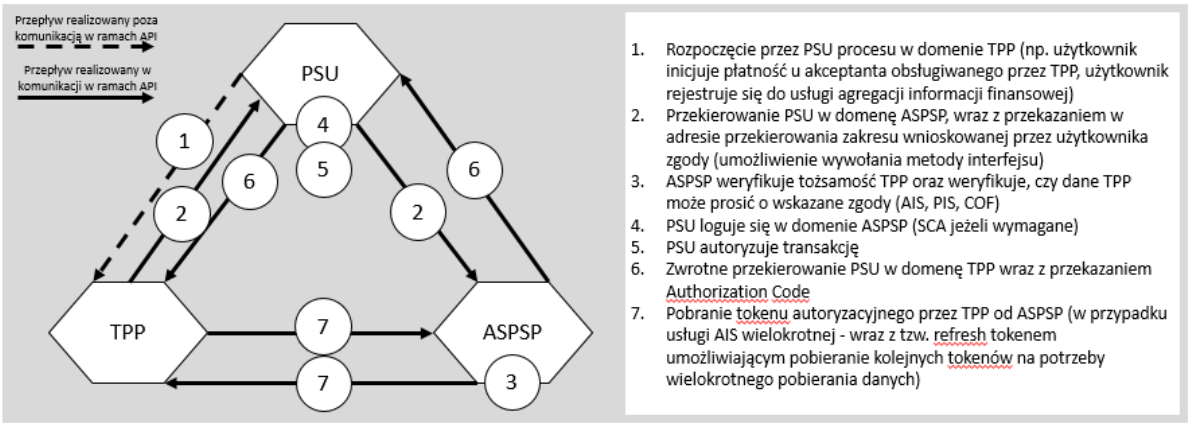
a) co do zakresu zasobów jakie są udostępniane (np. lista kont)
b) czasu na jaki są udostępniane
c) limitu liczby użycia
d) listy operacji jakich dotyczy
e) wybranych parametrów operacji np. długości historii wstecz, parametrów przelewu itp.

Specyfikacja struktury parametru scope_details znajduje się w załączniku nr 1.

Powyższe parametry są przesyłane przez TPP jako POST (ze względu na możliwą wielkość scope_details) w formacie JSON zakodowane i podpisane przy użyciu JSON Web Signature zgodnie z RFC 7515.

## 7.2 Mechanizm uwierzytelniania po stronie ASPSP

Proces uwierzytelnienia PSU po stronie ASPSP został opracowany w oparciu o metodę *authorization code*, zdefiniowaną w standardzie OAuth 2.0. Wysokopoziomowy aspekt biznesowy tego mechanizmu został zobrazowany na poniższym schemacie, zaś szczegółowy przebieg procesu uwierzytelniania i uzyskiwania autoryzacji do zasobów ASPSP opisano w rozdziale 11.1.




***Ilustracja 19: Mechanizm uwierzytelniania po stronie ASPSP***

Poniżej opisano kroki wraz ze zmianami, w odniesieniu do standardu OAuth 2.0, wprowadzonymi przez PolishAPI i wynikającymi z wymogów prawnych oraz bezpieczeństwa.

### 7.2.1 Przekierowanie z TPP do ASPSP

Przekierowanie obejmuje następujące parametry:

| PARAMETR | WYMAGALNOŚĆ | KOMENTARZ |
|---|---|---|
| response_type | Wymagane | Wartość „code” |
| client_id | Wymagane | Unikalny identyfikator TPP |
| redirect_uri | wymagane | |
| scope | Wymagane | |
| scope_details | Wymagane | |
| state | Wymagane | Losowa, unikalna w ramach TPP wartość – zabezpieczenie przed atakiem Cross-Site Request Forgery |

### 7.2.2 Uwierzytelnienie PSU i autoryzacja

Realizacja po stronie ASPSP.

### 7.2.3 Zwrotne przekierowanie przeglądarki PSU do TPP

Po udzieleniu autoryzacji dla TPP przez PSU, serwer autoryzacji dostarcza tę informację do TPP poprzez przekazanie wygenerowanego kodu autoryzacyjnego (*authorization code*), który ma charakter jednorazowy, co oznacza, iż może zostać wykorzystany przez TPP do uzyskania dostępu do zasobów ASPSP (uzyskania tokena dostępu) dokładnie jeden raz. Przekazanie tego kodu jest wykonywane wewnątrz żądania przekierowania przeglądarki PSU na adres redirect_uri (z użyciem parametru Content-Type o wartości "application/x-www-form-urlencoded"). Dodatkowo w żądaniu może zostać przekazany parametr state, który jest wymagany tylko jeśli został uprzednio wysłany przez TPP w żądaniu autoryzacji.

Przykładowe przekierowanie zwrotne do TPP po dokonaniu uwierzytelnienia PSU i autoryzacji dostępu TPP do zasobów ASPSP:

HTTP/1.1 302 Found

Location: https://[redirect_uri]?code=[authorization_code]

&state=[state]

gdzie:

[redirect_uri] – adres po stronie TPP, przekazany w żądaniu autoryzacji, na który zostaje wykonane przekierowanie przeglądarki PSU, po zakończeniu procesu uwierzytelniania tego PSU oraz autoryzacji dostępu do zasobów ASPSP

[authorization_code] – jednorazowy kod autoryzacyjny potwierdzający poprawne uwierzytelnienie PSU i nadanie przez niego autoryzacji dostępu do zasobów ASPSP dla TPP

[state] – dodatkowy parametr, pozwalający na dopasowanie żądania autoryzacji z żądaniem przekserowania po zakończeniu uwierzytelnienia PSU i nadaniu przez niego autoryzacji dostępu do zasobów ASPSP dla TPP, wykorzystywany w celu zapobiegania atakom typu „*cross-site request forgery*".

### 7.2.4 Pobranie tokenu na podstawie Auhtorization Code

| PARAMETR | WYMAGALNOŚĆ | KOMENTARZ |
|---|---|---|
| grant_type | wymagane | Wartość „authorization_code" |
| Code | wymagane | Zgodna z wartością przekazaną w kroku 7.1.3 |
| redirect_uri | wymagane | Wartość zgodna z wartością z kroku7.1.1 |
| client_id | wymagane | Unikalny identyfikator TPP |

Uwierzytelnienie TPP odbywa się na podstawie certyfikatu użytego do połączenia TLS

Zwracane dane także z uwzględnieniem pola o nazwie *scope_details*, zawierającego szczegóły zgód, jakie wyraził PSU.

| PARAMETR | WYMAGALNOŚĆ | KOMENTARZ |
|---|---|---|
| access_token | wymagane | |
| token_type | wymagane | |
| expires_in | wymagane | |
| refresh_token | opcjonalne | |
| scope | wymagane | |
| scope_details | wymagane | |

### 7.2.5 Wycofanie zgody

Wycofanie zgody jest realizowane za pomocą metody /v1.0/accounts/v1.0/deleteConsent

### 7.2.6 Stosowanie struktury scope_details

- Zgodę jednorazową obsługujemy za pomocą parametru scopeUsageLimit.

## 7.3 Mechanizm uwierzytelniania w zewnętrznym narzędziu autoryzacyjnym (*decoupled*)

Podstawowym założeniem opisywanej metody uwierzytelnienia PSU jest wykorzystanie EAT (ang. *External Authorization Tool*) czyli tzw. zewnętrznego narzędzia autoryzacyjnego. Jest to narzędzie, którego minimalną funkcjonalnością jest zdolność do przeprowadzenia silnego uwierzytelnienia PSU, w rozumieniu technicznych wymogów dyrektywy PSD2. Ponadto, narzędzie EAT może być oprogramowaniem zewnętrznym względem infrastruktury technicznej ASPSP i w takim przypadku musi zapewniać takie połączenie z infrastrukturą ASPSP, które zapewnienia bezpieczną wymianę informacji autoryzacyjnych.

Nawiązywanie sesji pomiędzy TPP i ASPSP, z uwzględnieniem silnego uwierzytelnienia PSU i w oparciu o metodę *Decoupled*, w celu umożliwienia TPP wykorzystania interfejsu XS2A, musi zostać przeprowadzane w zgodzie z procesem, opisanym w poniższych punktach. Opisany proces został opracowany w oparciu o założenia protokołu OAuth 2.0 co oznacza, iż posługuje się pojęciami tam zdefiniowanymi (jak „*authorization code*", „*access token*") ale stanowi odrębny sposób uzyskania dostępu do interfejsu XS2A, ze względu na brak wykorzystania przekierowań (ang. *redirections*), w rozumieniu protokołu http, które są mechanizmem wymaganym przez ten standard w metodzie „*Authorization Code Grant*". Takie podejście zostało zastosowane w celu zapewnienia spójności procesu uzyskiwania dostępu do interfejsu XS2A, bez względu na wybraną metodę uwierzytelnienia PSU, i ma na celu ułatwienie czynności integracyjnych związanych z wykorzystaniem interfejsu XS2A przez TPP.

1. TPP inicjuje proces nawiązania sesji z interfejsem XS2A po stronie ASPSP poprzez wywołanie następującej metody interfejsu XS2A:

**POST /[VER_A]/auth/[VER_B]/authorizeExt**

Dane wysyłane w żądaniu powinny być zgodne ze specyfikacją techniczną interfejsu XS2A, opisaną w załączniku nr 1. Należy podkreślić jednak, iż te parametry, w przeważającej większości, są tożsame z parametrami żądania inicjującego sesję z interfejsem XS2A z wykorzystaniem mechanizmu uwierzytelniania po stronie ASPSP, opisanego w punkcie 7.1.1. Najistotniejsze parametry tego żądania zostały opisane w poniższej tabeli.

| PARAMETR | WYMAGALNOŚĆ | KOMENTARZ |
|---|---|---|
| response_type | Wymagane | Wartość stała: code |
| eatCode | Wymagane | Jednorazowy kod autoryzacyjny wygenerowany przez narzędzie EAT |
| client_id | Wymagane | Unikalny identyfikator TPP |
| callbackURL | Wymagane | Adres funkcji zwrotnej w interfejsie TPP, na który zostanie przesłane żądanie zawierające wynik uwierzytelnienia PSU |
| apiKey | | Klucz zabezpieczający i dopasowujący odpowiedź na żądanie przesyłane w formie funkcji zwrotnej<br>Wartość klucza pełni dwie funkcje: Stanowi wartość identyfikującą ASPSP, na podstawie której TPP określa czy stroną wysyłającą żądanie zwrotne jest tą, do której zostało wysłane pierwotne żądanie Pozwala na dopasowanie żądania zwrotnego do żądania wysłanego pierwotnie przez TPP. Konieczne w przypadku wielu żądań wysyłanych do ASPSP, dla których odpowiedzi są przesyłane w postaci żądań do interfejsu zwrotnego TPP<br>Specyfikacja sposobu przekazywania atrybutu apiKey, |

| | | |
|---|---|---|
| | | zarówno w żądaniach do interfejsu XS2A, jak i do interfejsu wywołań zwrotnych, została zapisana w formacie *swagger* (wersja 2.0) w załącznikach 1 i 2. |
| scope_details | Wymagane | Struktura parametru opisana w załączniku nr 1 |

W wyniku wywołania tej metody i po dokonaniu przez ASPSP pozytywnej weryfikacji TPP (Mutual TLS Authentication, weryfikacja client_id) oraz potwierdzeniu niezaprzeczalności otrzymanej wiadomości (JWS Signature), zostanie zwrócona informacja na temat potwierdzenia rozpoczęcia procesu uwierzytelniania PSU.

Uwagi:

Jednorazowy kod autoryzacyjny, który jest wymagany jako parametr wejściowy metody authorizeExt, musi zostać wygenerowany w narzędziu EAT, na żądanie PSU, który uprzednio zostanie przez to narzędzie uwierzytelniony.

PSU musi uprzednio aktywować dostęp do narzędzia EAT zgodnie z procedurą opracowaną i wymaganą przez każdy z ASPSP.

Narzędzie EAT zapewnia procedurę silnego uwierzytelnienia PSU, Wynik przeprowadzonej procedury SCA musi zostać przekazany powiadomieniem do właściwego ASPSP. Sposób przekazania wyniku silnego uwierzytelnienia PSU do ASPSP, przeprowadzonego przez narzędzie EAT, nie jest przedmiotem specyfikacji PolishAPI.

Wynik przeprowadzonej procedury silnego uwierzytelnienia PSU musi następnie zostać przekazany przez ASPSP do TPP przy użyciu następującej metody interfejsu wywołań zwrotnych, po stronie TPP: **[callBackURL]/[VER_A]/auth/[VER_B]/authorizeExtCallBack**

Zakres danych żądania został opisany szczegółowo w załączniku nr 2. Najważniejszymi parametrami tego żądania są:

| PARAMETR | WYMAGALNOŚĆ | KOMENTARZ |
|---|---|---|
| authorized | Wymagane | Znacznik logiczny oznaczający wynik przeprowadzonego silnego uwierzytelnienia PSU przez narzędzie EAT.<br>- true – PSU został uwierzytelniony<br>- false – PSU nie został uwierzytelniony |
| code | Warunkowe | Jest to wartość kodu autoryzacyjnego, w rozumieniu standardu OAuth 2.0 i metody „authorization code", wygenerowanego przez ASPSP tylko i wyłącznie w wyniku uwierzytelnienia PSU w narzędziu EAT. |

Na podstawie otrzymanego w poprzednim kroku kodu autoryzacyjnego, TPP powinien zainicjować sesję interfejsu XS2A poprzez użycie następującej metody tego interfejsu, w której jednym z wymaganych parametrów jest kod autoryzacyjny, a informację zwrotną stanowi m.in. tzw. „Access token" (w rozumieniu standardu OAuth 2.0): **/[VER_A]/auth/[VER_B]/token**

Sposób wywołania tej metody jest zgodny z punktem 7.2.4, opisującym sposób zainicjowania sesji w metodzie uwierzytelnienia PSU po stronie ASPSP. Szczegółowa specyfikacja techniczna tej metody jest opisana w załączniku nr 1.

## 7.4 Pobranie *access tokena* na podstawie *refresh tokena*

Po wygaśnięciu ważności *access tokenu*, TPP może pobrać nowy *access token* korzystając z *refresh tokenu* (o ile został wydany). Taka sytuacja będzie miała miejsce w przypadku usługi AIS wielokrotny.

Poniżej przedstawiono żądanie TPP i odpowiedź serwera ASPSP

| PARAMETR | WYMAGALNOŚĆ | KOMENTARZ |
|---|---|---|
| grant_type | wymagane | Wartość „refresh_token” |
| refresh_token | wymagane | Zgodna z wartością przekazaną przez ASPSP w kroku 7.1.4 |
| scope | opcjonalny | Żądany zakres nie może być większy niż ten, który przekazano w kroku 7.1.4 |
| scope_details | opcjonalny | Żądany zakres nie może być większy niż ten, który przekazano w kroku 7.1.4 |
| is_user_session | opcjonalne | Określa czy dana sesja jest związana z interakcją z PSU – wartości true/false. Rozszerzenie standardu OAuth2 |
| user_ip | Wymagane, jeśli is_user_session=true | IP przeglądarki użytkownika (informacja na potrzeby fraud detection)<br>Rozszerzenie standardu OAuth2 |
| user_agent | Wymagane, jeśli is_user_session=true | Informacja dotycząca wersji przeglądarki użytkownika (informacja na potrzeby fraud detection)<br>Rozszerzenie standardu OAuth2 |

Odpowiedź odsyłana przez ASPSP jest taka sama jak w pkt. 7.2.4

## 7.5 Pobranie nowego *access tokena* na podstawie *exchange tokena*

Jest to metoda nawiązania sesji komunikacyjnej z interfejsem XS2A, której celem jest zapewnienie możliwości wymiany tokena dostępu, bez konieczności ponownego przeprowadzenia procedury SCA, w przypadku zmiany zakresu zgód, zgodnie ze scenariuszem opisanym w pkt. 1.4.4.2.3 specyfikacji. Ten scenariusz zakłada uzyskanie dostępu do ściśle określonego podzbioru rachunków PSU, który został przez niego wskazany w oparciu o listę wszystkich jego rachunków w danym ASPSP, którą TPP uprzednio uzyskał w oparciu o inny rodzaj zgody PSU i po przeprowadzeniu procedury SCA.

Dla realizacji tej metody nawiązania sesji konieczne jest użycie dedykowanej metody autoryzacji, wskazanej w atrybucie „grant_type” metody /token o wartości „exchange_token”, oraz przekazanie w dedykowanym atrybucie o tej samej nazwie (exchange_token) wartości tokena dostępu, uzyskanego podczas wcześniejszego występowania o zgodę na pobranie listy rachunków, skojarzonego z ważną sesją komunikacyjną interfejsu XS2A.

Poniżej przedstawiono żądanie TPP i odpowiedź serwera ASPSP

| PARAMETR | WYMAGALNOŚĆ | KOMENTARZ |
|---|---|---|
| grant_type | wymagane | Wartość „exchange_token” |
| exchange_token | wymagane | Token dostępu uzyskany podczas występowania o zgodę na pobranie listy rachunków |
| scope | opcjonalny | Żądany zakres musi zostać zawężony do wybranych przez PSU rachunków oraz uprawnień dotyczących zakresu żądanych informacji np. szczegóły rachunku, historia transakcji czy szczegóły transakcji |
| scope_details | opcjonalny | Żądany zakres musi zostać zawężony do wybranych |

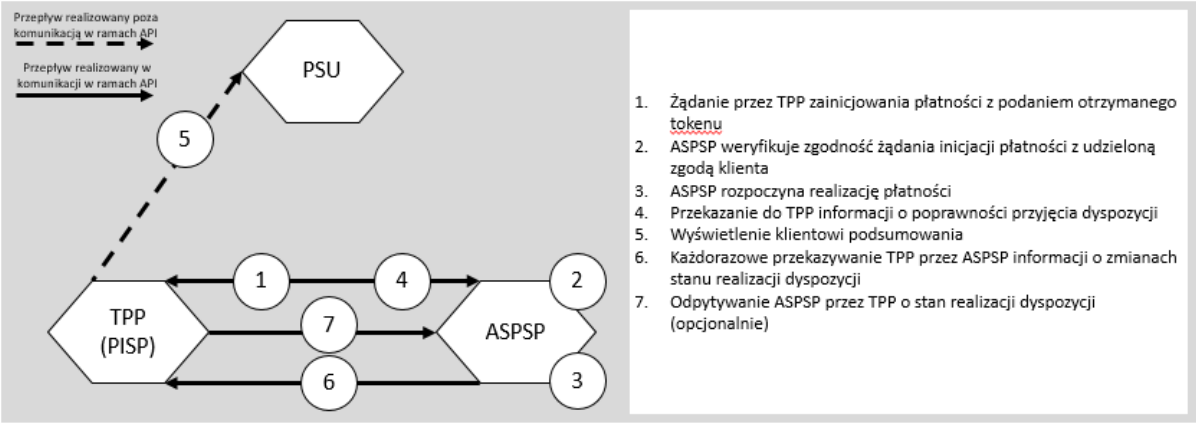
| | | |
|---|---|---|
| | | przez PSU rachunków oraz uprawnień dotyczących zakresu żądanych informacji np. szczegóły rachunku, historia transakcji czy szczegóły transakcji |
| is_user_session | opcjonalne | Określa czy dana sesja jest związana z interakcją z PSU – wartości true/false. Rozszerzenie standardu OAuth2 |
| user_ip | Wymagane, jeśli is_user_session=true | IP przeglądarki użytkownika (informacja na potrzeby fraud detection)<br>Rozszerzenie standardu OAuth2 |
| user_agent | Wymagane, jeśli is_user_session=true | Informacja dotycząca wersji przeglądarki użytkownika (informacja na potrzeby fraud detection)<br>Rozszerzenie standardu OAuth2 |

Odpowiedź odsyłana przez ASPSP jest taka sama jak w pkt. 7.2.4

# 8 Opis techniczny usługi PIS

Rozdział stanowi streszczenie specyfikacji API w formacie swagger zdefiniowanej w Załączniku nr 1 oraz Załączniku nr 2.

## 8.1 Diagram aktywności w usłudze PIS




***Ilustracja 20: Wysokopoziomowy diagram aktywności w usłudze PIS***

## 8.2 Struktura zapytań interfejsu XS2A

Poniższa tabela zawiera podstawowe informacje na temat wszystkich metod usługi PIS interfejsu XS2A z uwzględnieniem klas obiektów kanonicznego modelu danych przekazywanych w żądaniach i i otrzymywanych w odpowiedziach.

| METODA INTERFEJSU | OPIS | KLASA OBIEKTU KMD |
|---|---|---|
| /payments/{wersja}/domestic | Inicjuje przelew krajowy | PaymentDomesticRequest/ AddPaymentResponse |
| /payments/{wersja}/EEA | Inicjuje przelew zagraniczny SEPA | PaymentEEARequest/ AddPaymentResponse |
| /payments/{wersja}/nonEEA | Inicjuje przelew zagraniczny poza SEPA | PaymentNonEEARequest / AddPaymentResponse |
| /payments/{wersja}/tax | Inicjujeprzelew do US | PaymentTaxRequest / AddPaymentResponse |
| /payments/{wersja}/getPayment | Pobiera status realizacji przelewu | PaymentRequest/ GetPaymentResponse |
| /payments/{wersja}/getMultiplePayments | Pobiera statusy realizacji wielu płatności. Wywołanie nie wymaga podania tokenu. | PaymentsRequest/ PaymentResponse |

## 8.3 Struktura zapytań interfejsu wywołań zwrotnych - CallBack

Specyfikacja usługi PIS obejmuje również definicję interfejsu wywołań zwrotnych (ang *CallBack*), dzięki której ASPSP ma możliwość powiadamiania TPP, w sposób asynchroniczny, o zmianach statusu płatności, zainicjowanej z użyciem wybranej metody usługi PIS interfejsu XS2A. Do tego celu została

zdefiniowana pojedyncza metoda interfejsu CallBack o nazwie paymentCallBack. Szczegółową specyfikację techniczną interfejs wywołań zwrotnych dla usługi PIS zdefiniowano w załączniku nr 2, dlatego poniższa tabela opisuje jedynie podstawowe elementy tego interfejsu.

| METODA INTERFEJSU | OPIS | KLASA OBIEKTU KMD |
|---|---|---|
| /payments/{wersja}/paymentCallBack | Przekazuje status realizacji pojedynczej płatności | PaymentStatusInfoRequest/ CallBackResponse |

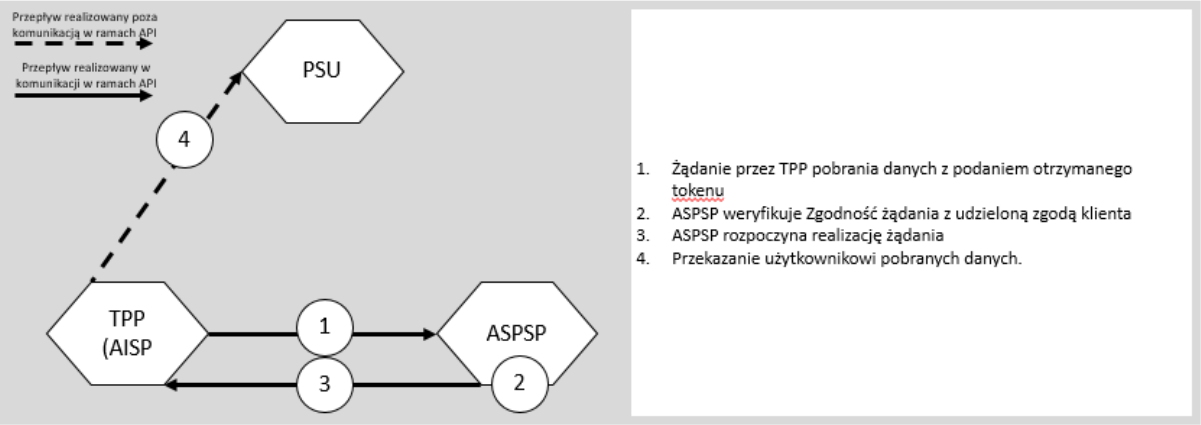
Jako metodę zabezpieczającą API zastosowano typ „apiKey” (https://swagger.io/docs/specification/2-0/authentication/) oraz dodatkowo weryfikowany jest odcisk palca certyfikatu serwerowego TPP, który używany jest do zestawiania połączenia TLS wywołania zwrotnego - przekazywany w parametrze keyID. W wywołanych PIS TPP przekazuje do ASPSP wartość apiKey oraz callbackURL, które są użyte w wywołaniach zwrotnych.

W przypadku nieudanego wywołania, TPP może je ponowić, a liczba ponownych wywołań zostanie zdefiniowana przez ASPSP w ramach dokumentacji implementacyjnej.

# 9 Opis techniczny usługi AIS

Rozdział stanowi streszczenie specyfikacji API w formacie swagger zdefiniowanej w Załączniku nr 1 oraz Załączniku nr 2.

## 9.1 Diagram aktywności w usłudze AIS




***Ilustracja 21: Wysokopoziomowy diagram aktywności w usłudze AIS***

## 9.2 Struktura zapytań interfejsu XS2A

Poniższa tabela zawiera podstawowe informacje na temat wszystkich metod usługi AIS interfejsu XS2A z uwzględnieniem klas obiektów kanonicznego modelu danych przekazywanych w żądaniach i i otrzymywanych w odpowiedziach.

| METODA INTERFEJSU | OPIS | KLASA OBIEKTU KMD |
|---|---|---|
| /accounts/{wersja}/deleteConsent | Usuwa/unieważnia zgodę | DeleteConsentRequest/ string |
| /accounts/{wersja}/getAccounts | Pobiera wszystkie rachunki PSU | AccountsRequest/ AccountsResponse |
| /accounts/{wersja}/getAccount | Pobiera pojedynczy rachunek płatniczy | AccountInfoRequest/ AccountInfo |
| /accounts/{wersja}/getTransactionsDone | Pobiera transakcje zrealizowane na rachunku | TransactionInfoRequest/ TransactionDoneInfoResponse |
| /accounts/{wersja}/getTransactionsPending | Pobiera transakcje oczekujące na rachunku | TransactionInfoRequest/ TransactionPendingInfoResponse |
| /accounts/{wersja}/getTransactionsRejected | Pobiera transakcje odrzucone na rachunku | TransactionInfoRequest/ TransactionRejectedInfoResponse |
| /accounts/{wersja}/getHolds | Pobiera blokady na rachunku | TransactionInfoRequest/ TransactionHoldInfoResponse |
| /accounts/{wersja}/getTransationDetail | Pobiera szczegóły pojedynczej transakcji/blokady | TransactionDetailRequest/ TransactionDetailResponse |

## 9.3 Struktura zapytań interfejsu wywołań zwrotnych - CallBack

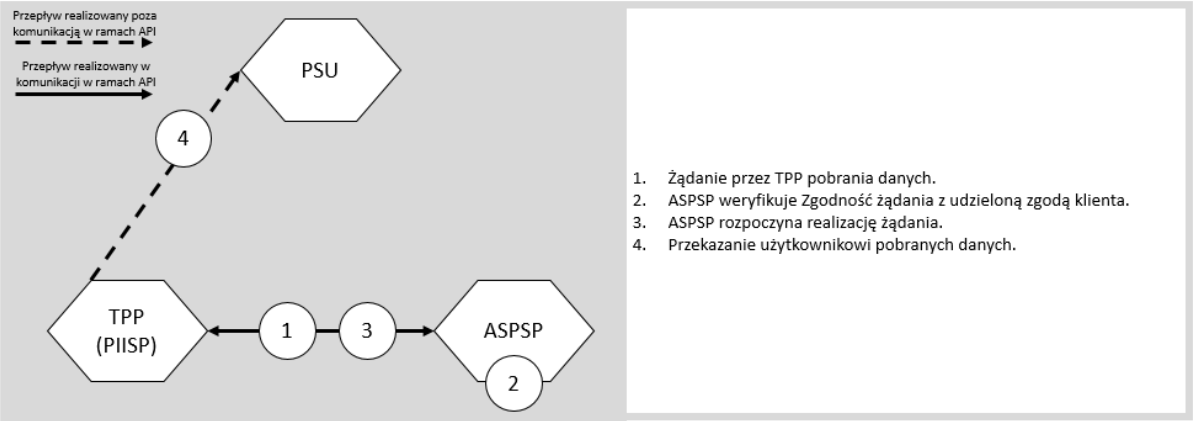
Specyfikacja usługi AIS obejmuje również definicję interfejsu wywołań zwrotnych (ang *CallBack*), dzięki której ASPSP ma możliwość przekazywania do TPP, w sposób asynchroniczny, informacji o rachunku, transakcjach i blokadach, o których udostępnienie wystąpił TPP poprzez wywołanie odpowiednich metod interfejsu XS2A. Do tego celu został zdefiniowany szereg metod interfejsu CallBack, których szczegółową specyfikacja techniczna została zdefiniowana w załączniku nr 2. Poniższa tabela opisuje jedynie podstawowe elementy interfejsu wywołań zwrotnych dla usługi AIS.

| METODA INTERFEJSU | OPIS | KLASA OBIEKTU KMD |
|---|---|---|
| /accounts/{wersja}/accountsCallBack | Przekazuje informacje o szczegółach wybranego rachunku ołatniczego | AccountsRequest / CallBackResponse |
| /accounts/{wersja}/transactionsDoneCallBack | Przekazuje informacje o transakcjach zrealizowanych dla danego rachunku płatniczego | TransactionDoneInfoRequest / CallBackResponse |
| /accounts/{wersja}/transactionsPendingCallBack | Przekazuje informacje o transakcjach oczekujących na realizację dla danego rachunku płatniczego | TransactionPendingInfoRequest / CallBackResponse |
| /accounts/{wersja}/transactionsRejectedCallBack | Przekazuje informacje o transakcjach odrzuconych dla danego rachunku płatniczego | TransactionRejectedInfoRequest / CallBackResponse |
| /accounts/{wersja}/transactionsHoldCallBack | Przekazuje informacje o blokadach dla danego rachunku płatniczego | TransactionHoldInfoRequest / CallBackResponse |

# 10 Opis techniczny usług CAF

Rozdział stanowi streszczenie specyfikacji API w formacie swagger zdefiniowanej w Załączniku nr 1.

## 10.1 Diagram aktywności w usłudze CAF




***Ilustracja 22: Wysokopoziomowy diagram aktywności w usłudze CAF***

## 10.2 Struktura zapytania interfejsu XS2A(w tym opis pól i wymagalność)

| METODA INTERFEJSU | OPIS | KLASA OBIEKTU KMD |
|---|---|---|
| /confirmation/{wersja}/getConfirmationOfFunds | Potwierdzenia dostępności środków | confirmationOfFundsRequest/ confirmationOfFundsResponse |

# 11 Utylizacja metod interfejsu XS2A oraz usług autoryzacyjnych – diagramy sekwencji

Przedstawione w notacji UML diagramy sekwencji, opisują interakcje zachodzące pomiędzy PSU, TPP, systemami po stronie ASPSP oraz systemami zewnętrznymi, które obrazują pełen zakres scenariuszy użycia interfejsu XS2A, będący przedmiotem specyfikacji PolishAPI. Na diagramach zostały pokazane jedynie podstawowe ścieżki sekwencji i interakcji prowadzące do osiągnięcia zamierzonego celu. Oznacza to, iż poszczególne interakcje mogą zakończyć się niepowodzeniem, czego efektem będą komunikaty i kody błędów, zwracane przez metody interfejsu XS2A lub usług autoryzacyjnych, a które nie zostały uwzględnione na diagramach dla zachowania przejrzystości przekazu. Z tego samego względu, poszczególne interakcje zawierają tylko niektóre parametry żądań i odpowiedzi, które są istotne z punktu widzenia poprawności zaprezentowanych sekwencji (pełna specyfikacja, obejmująca wszystkie parametry żądań i odpowiedzi usług interfejsu XS2A oraz usługi autoryzacyjnej, została opisana w specyfikacji technicznej tych interfejsów.

Na diagramach zostały wykorzystane następujące skróty i oznaczenia:

**ASPSP Auth** – interfejs komunikacyjny zapewniany przez ASPSP, zgodnie z specyfikacją PolishAPI (usługa AS - Authorization Service), którego rolą jest dostarczanie metod autoryzacji dostępu TPP do usług interfejsu XS2A i w efekcie nawiązywanie sesji z tym interfejsem.

**ASPSP XS2A** – interfejs komunikacyjny zapewniany przez ASPSP, którego rolą jest zapewnienie realizacji usług biznesowych opisanych w specyfikacji PolishAPI (AIS - Account Information Service, PIS – Payment Initiation Service i CAF – Confirmation of the Availability of Funds).

**EAT** – ang. *External Authorization Tool*, czyli tzw. zewnętrzne narzędzie autoryzacyjne, będące systemem zapewniającym procedurę SCA czyli silnego uwierzytelnienia PSU.

**eatCode** – kod jednorazowy generowany w narzędziu EAT na żądanie PSU, który służy do autoryzacji dostępu do interfejsu XS2A jako jeden z faktorów procedury SCA.

**sync / async** – oznaczenie rodzaju komunikacji (synchroniczna, asynchroniczna), pomiędzy aktorami uwzględnionymi na diagramie.

**tpp_redirect_url** – adres URL, na który powinno nastąpić przekierowanie przeglądarki internetowej PSU, po zakończeniu uwierzytelniania PSU i autoryzacji dostępu TPP do zasobów tego PSU po stronie ASPSP, w przypadku metody *Redirection* uwierzytelnienia PSU.

**auth_redirect_url** – adres na jaki powinno nastąpić przekierowanie przeglądarki internetowej PSU. W celu jego uwierzytelnienia przy użyciu metody *Redirection.*

**authorization_code** – jednorazowy kod autoryzacyjny, który stanowi potwierdzenie autoryzacji TPP do dostępu do zasobów PSU.

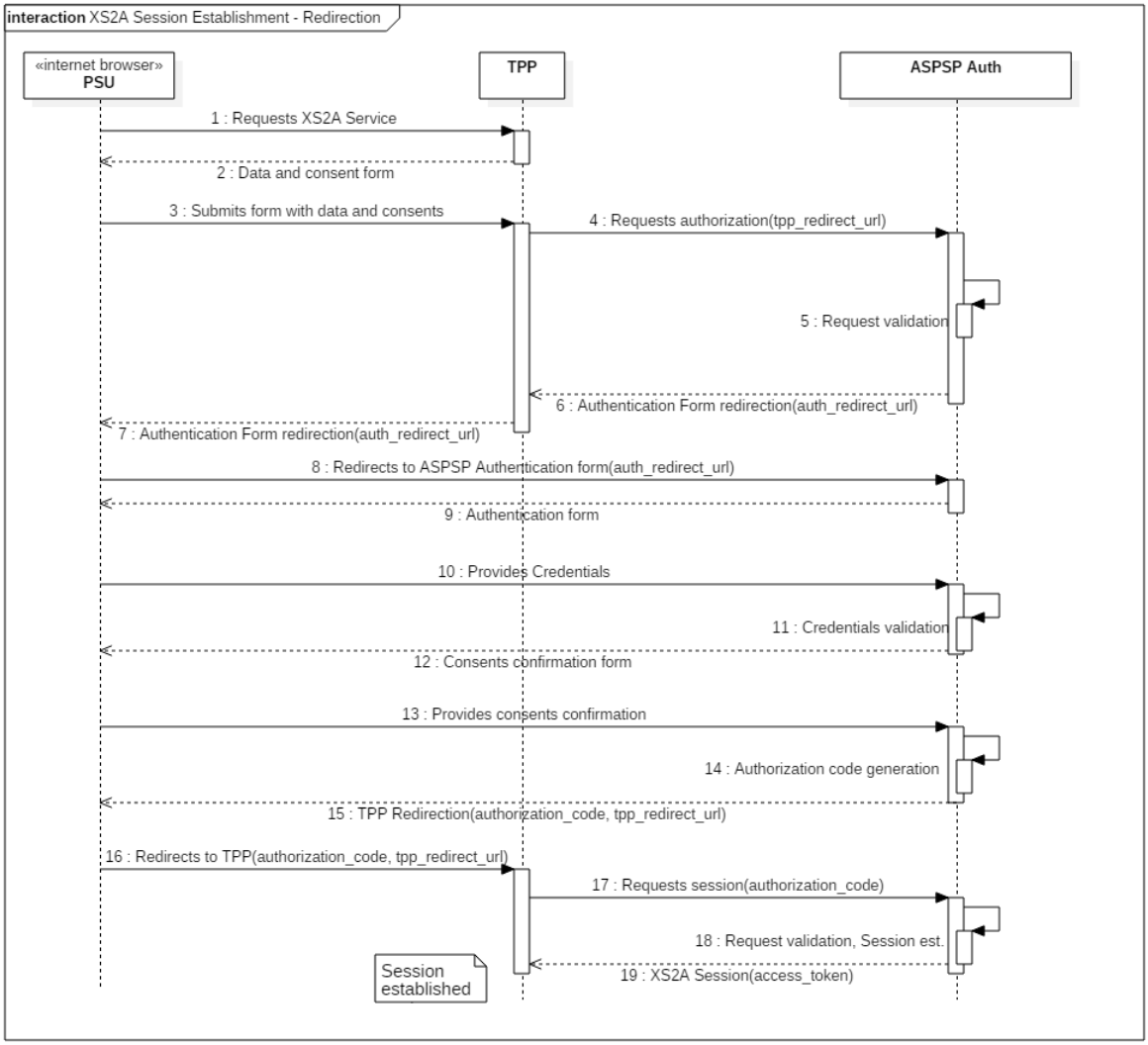
**access_token** – token dostępu umożliwiający utylizację usług interfejsu XS2A, opisany szerzej w punkcie 5.3 Definicja tokena dostępu.

**callback_url** – adres interfejsu zwrotnego po stronie TPP wskazujący cel wysyłania asynchronicznych odpowiedzi.

**apiKey** – rodzaj tokena wysyłanego w żądaniu w celu zabezpieczenia komunikacji asynchronicznej z interfejsem XS2A.

## 11.1 Nawiązywanie sesji XS2A z uwierzytelnieniem PSU metodą *redirection*

Diagram obrazuje sekwencję komunikacyjną, która prowadzi do nawiązania sesji z interfejsem XS2A, z uwzględnieniem uwierzytelnienia PSU metodą *redirection*, opisaną w rozdziale 7.1 Mechanizm uwierzytelniania po stronie ASPSP




Opis interakcji wg kolejności ich występowania:

1: PSU inicjuje wykorzystanie wybranej usługi interfejsu XS2A po stronie aplikacji TPP

2: TPP prezentuje formularz z danymi wymaganymi do identyfikacji ASPSP, wywołania usługi interfejsu XS2A oraz uzyskania dostępu do tego interfejsu

3: PSU wprowadza i zatwierdza dane wymagane w formularzu TPP

4: TPP żąda autoryzacji dostępu do interfejsu XS2A poprzez wywołanie następującej metody usługi autoryzacyjnej:

**/[VER_1]/auth/[VER_2]/authorize**

Jednym z parametrów tej metody jest adres url (tpp_redirect_url) powrotu do interfejsu TPP po zakończeniu procedury uwierzytelniania PSU i autoryzacji dostępu TPP do jego zasobów w ASPSP.

5: ASPSP waliduje poprawność otrzymanego żądania autoryzacyjnego pod różnymi względami, w tym poprawność podpisu, tożsamość TPP, zgodność przekazanych zgód z uprawnieniami TPP

6: ASPSP, w przypadku pozytywnego wyniku walidacji żądania autoryzacyjnego, zwraca dopowiedź w postaci przekierowania http do swojego interfejsu (auth_redirect_url) służącego do uwierzytelnienia PSU i jego autoryzacji w kontekście żądania wysłanego przez TPP

7: TPP interpretuje odpowiedź z ASPSP i zwraca do przeglądarki PSU odpowiedź w formie przekierowania do interfejsu ASPSP, który otrzymał w odpowiedzi na żądanie autoryzacji

8: Przeglądarka PSU dokonuje automatycznego przekierowania do interfejsu ASPSP przy użyciu otrzymanego auth_redirect_url

9: ASPSP zwraca do przeglądarki stronę zawierającą formularz do uwierzytelnienia PSU

10: PSU wprowadza dane uwierzytelniająca do zaprezentowanego formularza, które po jego zatwierdzeniu są przesyłane do ASPSP

11: ASPSP waliduje poprawność otrzymanych danych uwierzytelniających w ramach zapewnienia procedury SCA

12: Po potwierdzeniu tożsamości PSU ASPSP zwraca do przeglądarki stronę opisującą zakres zgód o jakie wystąpił TPP w celu realizacji usługi interfejsu XS2A, z formularzem służącym do zatwierdzenia żądania TPP

13: PSU akceptuje żądane przez TPP zgody poprzez zatwierdzenie zaprezentowanego formularza i przekazanie tej informacji do ASPSP

14: Po otrzymaniu akceptacji zgód, ASPSP generuje i zachowuje jednorazowy kod autoryzacyjny

15: ASPSP zwraca do przeglądarki odpowiedź w formie przekierowania do interfejsu TPP czyli na otrzymany w żądaniu autoryzacji adres url powrotu do TPP (tpp_redirect_url) i przekazuje jako parametr tej odpowiedzi wartość wygenerowanego, jednorazowego kodu autoryzacyjnego

16: Przeglądarka PSU dokonuje automatycznego przekierowania do interfejsu TPP przy użyciu otrzymanego adres url powrotu (tpp_redirect_url), wraz z jednorazowym kodem autoryzacyjnym

17: TPP na podstawie otrzymanego żądania z jednorazowym kodem autoryzacyjnym prosi ASPSP o nawiązanie sesji z interfejsem XS2A, w kontekście otrzymanej od PSU autoryzacji. W tym celu dokonuje wywołania następującej metody usługi autoryzacyjnej, której jednym z wymaganych elementów jest jednorazowy kod autoryzacyjny (authorization_code):

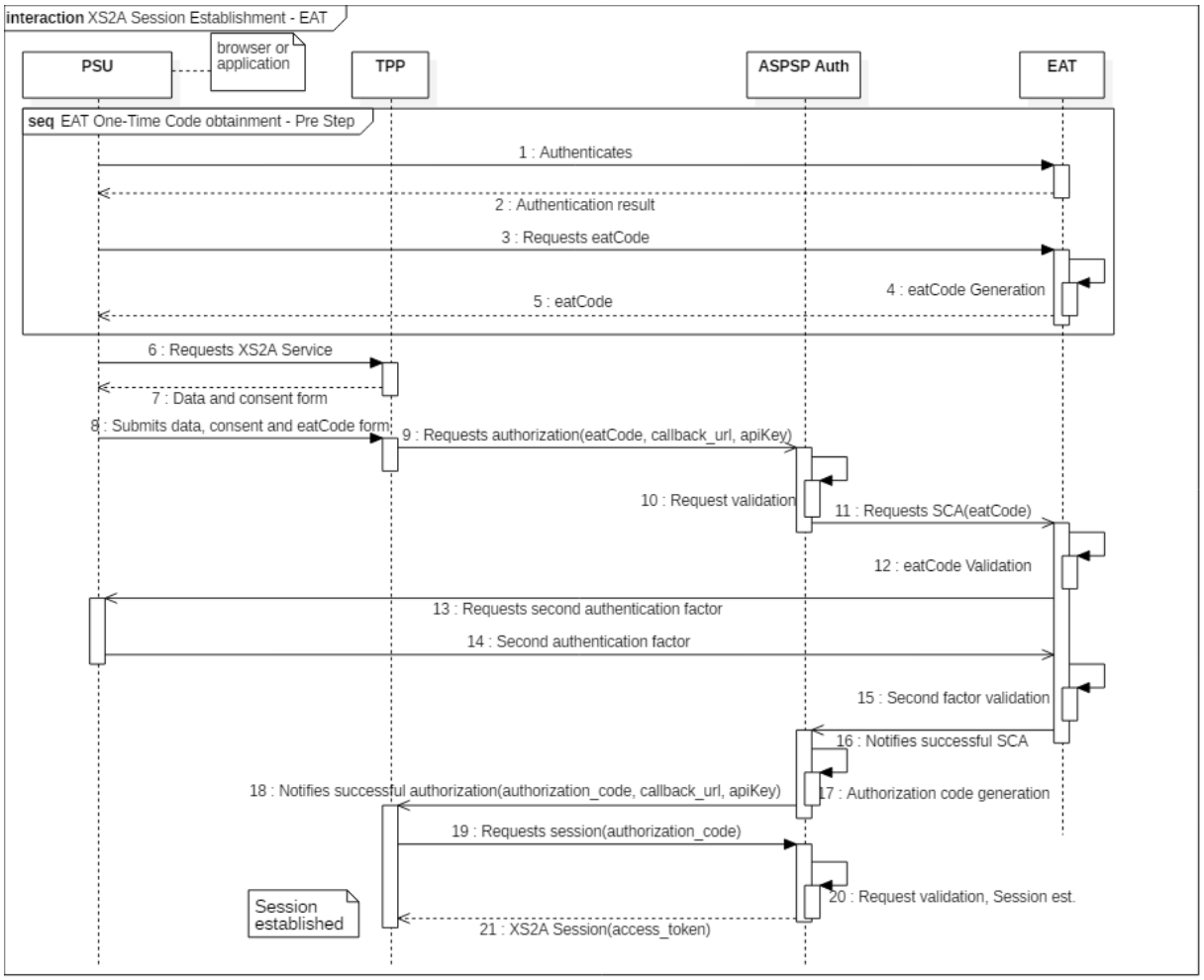
**/[VER_1]/auth/[VER_2]/token**

18: ASPSP waliduje otrzymane żądanie nawiązania sesji XS2A poprzez weryfikację otrzymanego kodu autoryzacyjnego (authorization_code) oraz danych o udzielonych przez PSU zgodach. Po pozytywnej weryfikacji, ASPSP ustanawia nową sesję interfejsu XS2A czego efektem jest wygenerowanie unikalnego tokena dostępu (access_token).

19: ASPSP zwraca do TPP odpowiedź na żądanie nawiązania sesji, zawierającą m.in. wartość wygenerowanego tokena dostępu, potwierdzając tym samym nawiązanie sesji z interfejsem XS2A

## 11.2 Nawiązywanie sesji XS2A z uwierzytelnieniem PSU metodą *decoupled*

Diagram obrazuje sekwencję komunikacyjną, która prowadzi do nawiązania sesji z interfejsem XS2A, z uwzględnieniem uwierzytelnienia PSU metodą *Decoupled*, opisaną w rozdziale 7.2 Mechanizm uwierzytelniania w zewnętrznym narzędziu autoryzacyjnym (decoupled)




Opis interakcji wg kolejności ich występowania:

1: PSU poprzez przeglądarkę lub aplikację przesyła dane do uwierzytelnia się w narzędziu EAT

2: Narzędzie EAT weryfikuje dane uwierzytelniające i nadaje dostęp PSU do swojego interfejsu

3: PSU żąda wydania kodu jednorazowego (eatCode)

4: Narzędzie EAT generuje kod jednorazowy (eatCode)

5: Narzędzie EAT zwraca kod jednorazowy do przeglądarki lub aplikacji PSU

6: PSU inicjuje wykorzystanie wybranej usługi interfejsu XS2A po stronie aplikacji TPP

7: TPP prezentuje formularz z danymi wymaganymi do identyfikacji ASPSP, wywołania usługi interfejsu XS2A oraz uzyskania dostępu do tego interfejsu (m.in. do wprowadzenia wartość kodu eatCode)

8: PSU wprowadza i zatwierdza dane wymagane w formularzu TPP

9: TPP żąda autoryzacji dostępu do interfejsu XS2A poprzez wywołanie następującej metody usługi autoryzacyjnej:

**/[VER_1]/auth/[VER_2]/authorizeExt**

Ze względu na asynchroniczny charakter odpowiedzi na to żądanie, wśród parametrów metody wymagane są adres url (callback_url) interfejsu zwrotnego XS2A oraz token zabezpieczający (apiKey). Ponadto wymaganiem do uzyskania autoryzacji jest również przekazanie kodu jednorazowego otrzymanego z narzędzia EAT (eatCode).

10: ASPSP waliduje poprawność otrzymanego żądania autoryzacyjnego pod różnymi względami, w tym poprawność podpisu, tożsamość TPP, zgodność przekazanych zgód z uprawnieniami TPP

11: ASPSP wysyła żądanie do narzędzia EAT w celu przeprowadzenia procedury SCA wobec PSU, w tym weryfikacji poprawności otrzymanego od PSU kodu jednorazowego (eatCode), wygenerowanego przez EAT

12: EAT sprawdza poprawność kodu jednorazowego (eatCode) otrzymanego z ASPSP

13: W przypadku poprawności kodu jednorazowego narzędzie EAT żąda od PSU drugiego faktora w celu ukończenia procedury SCA

14: PSU wykonuje drugi faktor w narzędziu EAT

15: Narzędzie EAT dokonuje weryfikacji poprawności dostarczonego przez PSU, drugiego faktora

16: Narzędzie EAT powiadamia ASPSP o wyniku przeprowadzonej procedury SCA

17: ASPSP wobec pozytywnego wyniku silnego uwierzytelnienia PSU i przeprowadzonej autoryzacji dostępu TPP do zasobów PSU (w tym uzyskaniu zgód od PSU), generuje i zachowuje jednorazowy kod autoryzacyjny (authorization_code)

18: ASPSP powiadamia TPP o wyniku żądania autoryzacji dostępu do zasobów PSU poprzez wywołanie następującej metody interfejsu wywołań zwrotnych po stronie TPP:

**/[VER_1]/auth/[VER_2]/authorizeExtCallBack**

W przypadku uzyskania przez TPP autoryzacji dostępu do zasobów PSU w tym żądaniu przekazywany jest jednorazowy kod autoryzacyjny (authorization_code) .

19: TPP na podstawie otrzymanego żądania z jednorazowym kodem autoryzacyjnym prosi ASPSP o nawiązanie sesji z interfejsem XS2A, w kontekście otrzymanej od PSU autoryzacji. W tym celu dokonuje wywołania następującej metody usługi autoryzacyjnej, której jednym z wymaganych elementów jest jednorazowy kod autoryzacyjny (authorization_code):

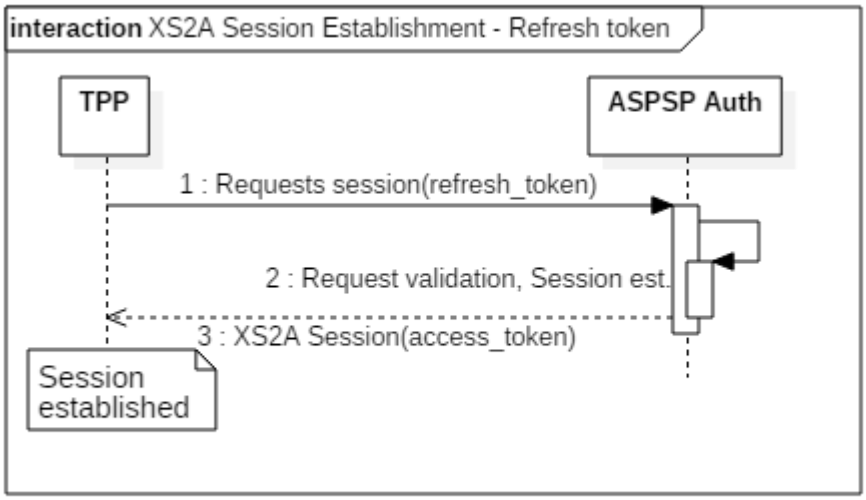
**/[VER_1]/auth/[VER_2]/token**

20: ASPSP waliduje otrzymane żądanie nawiązania sesji XS2A poprzez weryfikację otrzymanego kodu autoryzacyjnego (authorization_code) oraz danych o udzielonych przez PSU zgodach. Po pozytywnej weryfikacji, ASPSP ustanawia nową sesję interfejsu XS2A czego efektem jest wygenerowanie unikalnego tokena dostępu (access_token).

21: ASPSP zwraca do TPP odpowiedź na żądanie nawiązania sesji, zawierającą m.in. wartość wygenerowanego tokena dostępu, potwierdzając tym samym nawiązanie sesji z interfejsem XS2A

## 11.3 Nawiązywanie sesji XS2A z uwierzytelnieniem PSU metodą *refresh token*

Diagram obrazuje sekwencję komunikacyjną, która prowadzi do nawiązania sesji z interfejsem XS2A, z wykorzystaniem metody tzw. refresh tokena.




Opis interakcji wg kolejności ich występowania:

1: TPP wysyła żądanie do ASPSP o nawiązanie sesji z interfejsem XS2A, w kontekście wcześniej nawiązanej sesji, która uległa unieważnieniu, a z którą związany jest token dodatkowy (refresh_token), zwrócony do TPP podczas pierwotnej procedury nawiązywania sesji, która się unieważniła. W tym celu TPP wywołuje następującą metodę usługi autoryzacyjnej, z przekazaniem tokena dodatkowego (refresh_token):

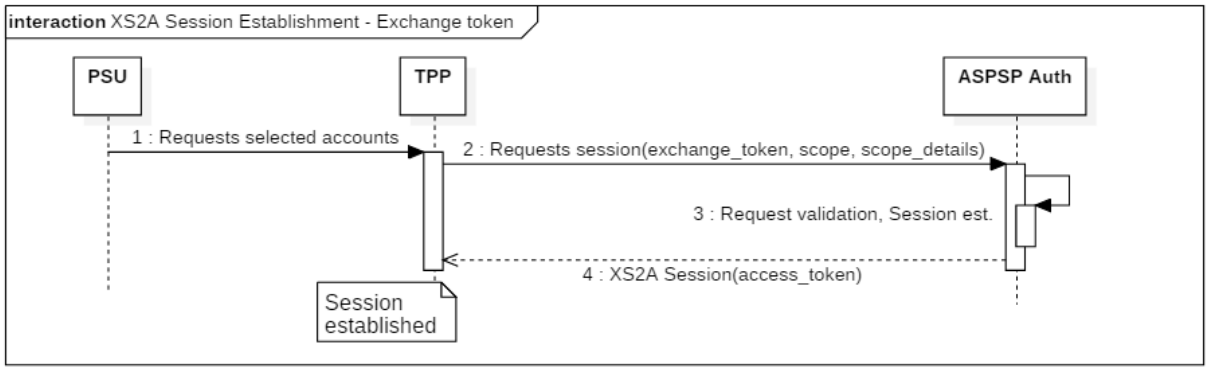
**/[VER_1]/auth/[VER_2]/token**

2: ASPSP waliduje otrzymane żądanie nawiązania sesji XS2A poprzez weryfikację otrzymanego tokena dodatkowego (refresh_token) oraz danych o udzielonych przez PSU zgodach. Po pozytywnej weryfikacji, ASPSP ustanawia nową sesję interfejsu XS2A czego efektem jest wygenerowanie unikalnego tokena dostępu (access_token).

3: ASPSP zwraca do TPP odpowiedź na żądanie nawiązania sesji, zawierającą m.in. wartość wygenerowanego tokena dostępu, potwierdzając tym samym odnowienie sesji z interfejsem XS2A.

## 11.4 Nawiązywanie sesji XS2A z uwierzytelnieniem PSU metodą *exchange token*

Diagram obrazuje sekwencję komunikacyjną, która prowadzi do nawiązania sesji z interfejsem XS2A, z wykorzystaniem metody tzw. exchange tokena.




Opis interakcji wg kolejności ich występowania:

1: PSU dokonuje uszczegółowienia udzielonych zgód dla TPP poprzez wybór podzbioru rachunków, spośród pobranych uprzednio przez TPP z ASPSP, oraz określenie zakresu uprawnień do danych związanych z tymi rachunkami, takich jak szczegóły rachunku, historia rachunku oraz jej zasięg czasowy czy szczegóły transakcji.

2: TPP wysyła żądanie do ASPSP o nawiązanie sesji z interfejsem XS2A, w kontekście wcześniej nawiązanej sesji, która została ustanowiona w celu pobrania listy rachunków PSU, i z którą związany jest token dostępu (access_token), zwrócony do TPP podczas pierwotnej procedury nawiązywania sesji z użyciem silnego uwierzytelnienia PSU. W tym celu TPP wywołuje następującą metodę usługi autoryzacyjnej, z przekazaniem wspomnianego tokena dostępowego używając parametru exchange_token:

**/[VER_1]/auth/[VER_2]/token**

Wymaganymi parametrami tego żądania są również parametry *scope* i *scope_details*, które muszą zawierać szczegółowy zakres zgód, z uwzględnieniem wybranych przez PSU numerów rachunków.

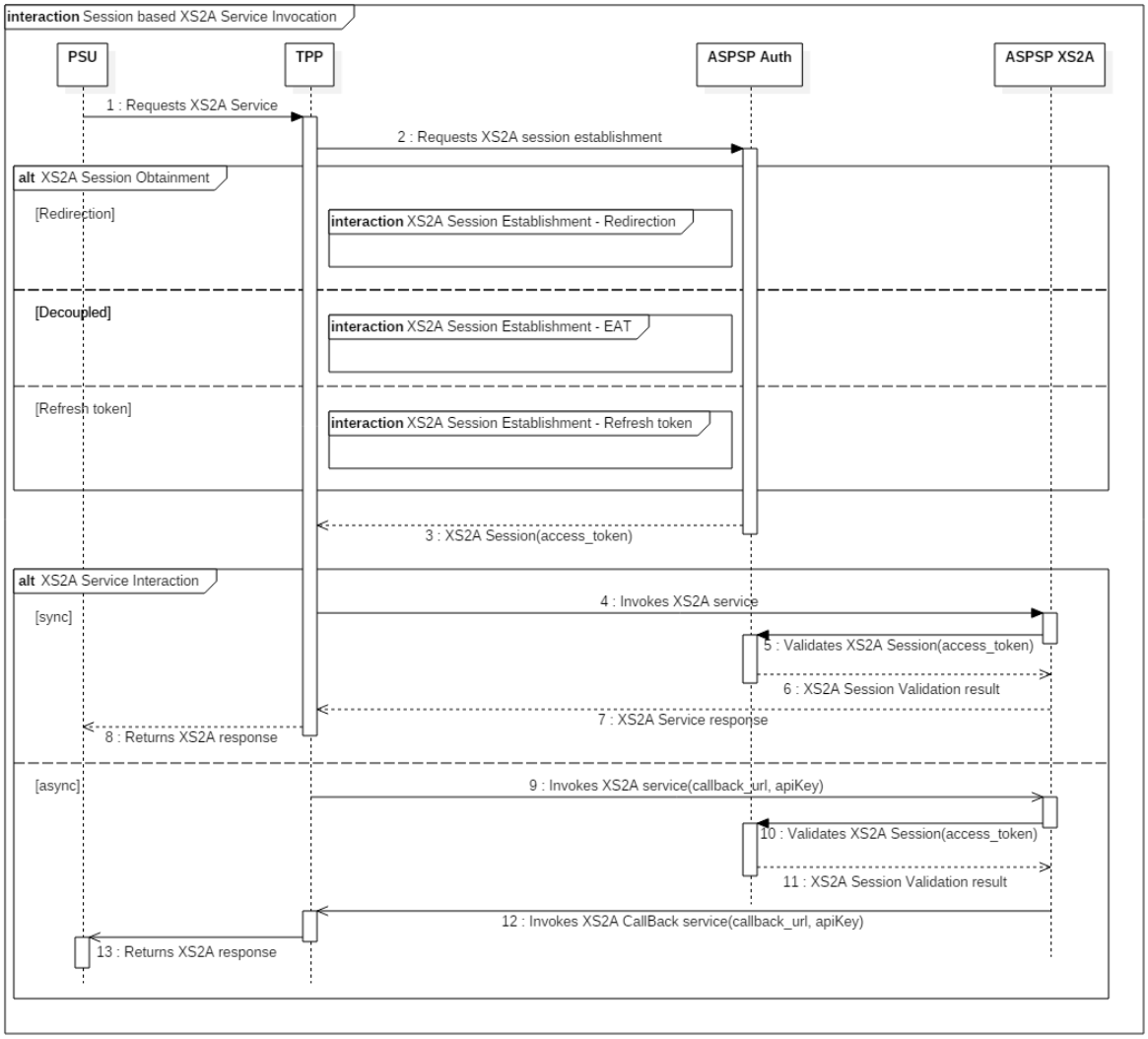
3: : ASPSP waliduje otrzymane żądanie nawiązania sesji XS2A poprzez weryfikację otrzymanego tokena dostępowego (przekazanego w atrybucie exchange_token) oraz danych o udzielonych przez PSU zgodach. Po pozytywnej weryfikacji, ASPSP ustanawia nową sesję interfejsu XS2A czego efektem jest wygenerowanie unikalnego, nowego tokena dostępu (access_token), w kontekście nowego zakresu zgód.

4: ASPSP zwraca do TPP odpowiedź na żądanie nawiązania sesji, zawierającą m.in. wartość wygenerowanego tokena dostępu, potwierdzając tym samym ustanowienie nowej sesji z interfejsem XS2A.

## 11.5 Wywołanie metod interfejsu XS2A z użyciem sesji

Diagram obrazuje sekwencję komunikacyjną pozwalającą na wywołanie usług interfejsu XS2A, dla których wymagana jest ważna sesja tego interfejsu. Poniższa tabela zawiera spis metod, w ramach usług AIS i PIS, dla których zaprezentowana sekwencja jest obowiązującą.

| **AIS** |
|---|
| /[VER_1]/accounts/[VER_2]/ getAccounts |
| /[VER_1]/accounts/[VER_2]/ getAccount |
| /[VER_1]/accounts/[VER_2]/ getTransactionsDone |
| /[VER_1]/accounts/[VER_2]/ getTransactionsPending |
| /[VER_1]/accounts/[VER_2]/ getTransactionsRejected |
| /[VER_1]/accounts/[VER_2]/ getHolds |
| /[VER_1]/accounts/[VER_2]/ getTransactionDetail |
| **PIS** |
| /[VER_1]/payments/[VER_2]/ domestic |
| /[VER_1]/payments/[VER_2]/ EEA |
| /[VER_1]/payments/[VER_2]/ nonEEA |
| /[VER_1]/payments/[VER_2]/ tax |
| /[VER_1]/payments/[VER_2]/ getPayment |




Opis interakcji wg kolejności ich występowania:

1: PSU inicjuje wykorzystanie wybranej usługi interfejsu XS2A po stronie aplikacji TPP

2: TPP żąda nawiązania sesji z interfejsem XS2A. Procedura nawiązania sesji może zostać przeprowadzona w oparciu o każdy z dostępnych wariantów, dla których diagramy sekwencji zostały opisane w poprzednich punktach niniejszego rozdziału.

3: ASPSP zwraca do TPP odpowiedź na żądanie nawiązania sesji, zawierającą m.in. wartość wygenerowanego tokena dostępu, potwierdzając tym samym nawiązanie sesji z interfejsem XS2A

**Wariant 1 – synchroniczne usługi interfejsu XS2A**

4: TPP wysyła żądanie do interfejsu XS2A w celu wykorzystania usługi tego interfejsu wybranej przez PSU (jedna z metod wymienionych w tabeli powyżej). W parametrach żądania przekazuje dane wejściowe, wymagane do realizacji usługi oraz token dostępu (access_token) w celu weryfikacji uzyskanej autoryzacji do wykorzystania tej usługi.

5: ASPSP waliduje poprawność i ważność otrzymanego tokena dostępu (access_token) poprzez komunikację usługą autoryzacyjną.

6: ASPSP otrzymuje wynik walidacji tokena dostępu

7: W przypadku pozytywnego wyniku walidacji tokena dostępu, ASPSP zwraca wynik realizacji usługi XS2A w postaci odpowiedzi na żądanie wysłane przez TPP do interfejsu XS2A.

8: TPP prezentuje PSU wynik realizacji usługi interfejsu XS2A

**Wariant 2 – asynchroniczne usługi interfejsu XS2A**

9: TPP wysyła żądanie do interfejsu XS2A w celu wykorzystania usługi tego interfejsu wybranej przez PSU (jedna z metod wymienionych w tabeli powyżej). W parametrach żądania przekazuje dane wejściowe - wymagane do realizacji usługi, token dostępu (access_token) - w celu weryfikacji uzyskanej autoryzacji do wykorzystania tej usług, oraz wartości parametrów callback_url i apiKey - wymagane do przesłania odpowiedzi na to żądanie w formie żądania do interfejsu wywołań zwrotnych po stronie TPP.

10: ASPSP waliduje poprawność i ważność otrzymanego tokena dostępu (access_token) poprzez komunikację usługą autoryzacyjną.

11: ASPSP otrzymuje wynik walidacji tokena dostępu

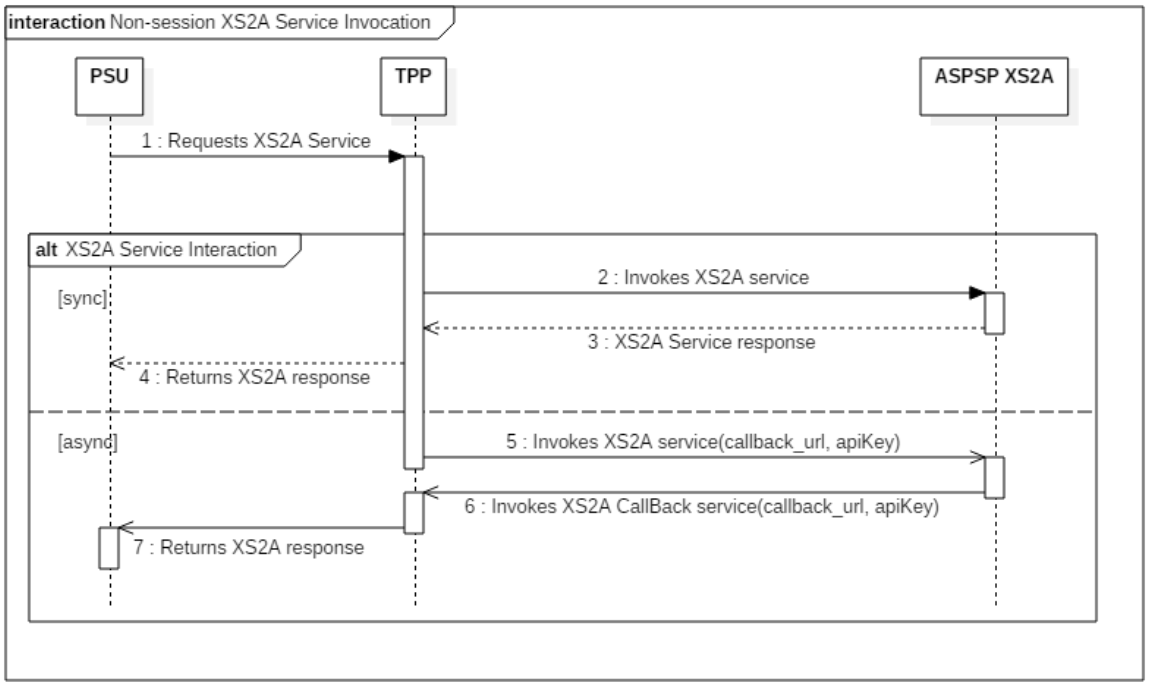
12: W przypadku pozytywnego wyniku walidacji tokena dostępu, ASPSP zwraca wynik realizacji usługi XS2A poprzez wysłanie żądania do interfejsu wywołań zwrotnych interfejsu XS2A po stronie TPP (na adres wskazany w callback_url).

13: TPP prezentuje PSU wynik realizacji usługi interfejsu XS2A

## 11.6 Wywołanie metod interfejsu XS2A bez użycia sesji

Diagram obrazuje sekwencję komunikacyjną pozwalającą na wywołanie usług interfejsu XS2A, dla których nie wymagana jest ważna sesja tego interfejsu. Poniższa tabela zawiera spis metod, w ramach usług AIS, PIS i CAF, dla których zaprezentowana sekwencja jest obowiązującą.

| **AIS** |
|---|
| /[VER_1]/accounts/[VER_2]/deleteConsent |
| **PIS** |
| /[VER_1]/payments/[VER_2]/getMultiplePayments |
| **CAF** |
| /[VER_1]/confirmation/[VER_2]/getConfirmationOfFunds |




Opis interakcji wg kolejności ich występowania:

1: PSU inicjuje wykorzystanie wybranej usługi interfejsu XS2A po stronie aplikacji TPP. Ta usługa nie wymaga sesji po stronie interfejsu XS2A. Tego typu usługi zostały wymienione w powyższej tabeli.

**Wariant 1 – synchroniczne usługi interfejsu XS2A**

2: TPP wysyła żądanie do interfejsu XS2A w celu wykorzystania usługi tego interfejsu wybranej przez PSU. W parametrach żądania przekazuje dane wejściowe, wymagane do realizacji usługi.

3: ASPSP zwraca wynik realizacji usługi XS2A w postaci odpowiedzi na żądanie wysłane przez TPP do interfejsu XS2A.

4: TPP prezentuje PSU wynik realizacji usługi interfejsu XS2A

**Wariant 2 – asynchroniczne usługi interfejsu XS2A**

5: TPP wysyła żądanie do interfejsu XS2A w celu wykorzystania usługi tego interfejsu wybranej przez PSU. W parametrach żądania przekazuje dane wejściowe - wymagane do realizacji usługi oraz wartości parametrów callback_url i apiKey - wymagane do przesłania odpowiedzi na to żądanie w formie żądania do interfejsu wywołań zwrotnych po stronie TPP.

6: ASPSP zwraca wynik realizacji usługi XS2A poprzez wysłanie żądania do interfejsu wywołań zwrotnych interfejsu XS2A po stronie TPP (na adres wskazany w callback_url).

7: TPP prezentuje PSU wynik realizacji usługi interfejsu XS2A

# 12 Kody błędów

| ETAP | BŁAD | KOD HTTP | SPOSÓB OBSŁUGI |
|---|---|---|---|
| Przekierowanie klienta w domenę ASPSP, wraz z przekazaniem w adresie przekierowania parametrów inicjowanej płatności | Przekazanie niepoprawnych parametrów | 400 | Zwrotne przekierowanie klienta w domenę TPP i zakończenie procesu (zgodnie z RFC6749 4.1.2.1) |
| ASPSP weryfikuje tożsamość TPP w Hub Tożsamości oraz weryfikuje, czy dane TPP może prosić o wskazane zgody (AIS czy PIS) | Niepoprawna weryfikacja tożsamości TPP | 401 | Wyświetlenie komunikatu użytkownikowi |
| | Niepoprawna weryfikacja licencji TPP (np. tylko AIS) | 403 | Zwrotne przekierowanie klienta w domenę TPP i zakończenie procesu (zgodnie z RFC6749 4.1.2.1) |
| Klient loguje się w domenie ASPSP | Niepoprawne logowanie 1 raz | 401 | Oczekiwanie na poprawne zalogowanie klienta |
| | Niepoprawne logowanie kilkukrotnie | 401 | Oczekiwanie na poprawne zalogowanie klienta |
| | Brak aktywacji usług TPP (opt-out) | 403 | Zwrotne przekierowanie klienta w domenę TPP i zakończenie procesu (zgodnie z RFC6749 4.1.2.1) |
| | Brak środków | 422 | Zwrotne przekierowanie klienta w domenę TPP i zakończenie procesu (odpowiedni kod błędu zwracany przy wywołaniu usługi płatności) |
| Klient wyraża zgodę w żądanym lub ograniczonym zakresie, ewentualnie odrzuca prośbę | Odrzucenie żądania przez klienta | 403 | Zwrotne przekierowanie klienta w domenę TPP i zakończenie procesu (zgodnie z RFC6749 4.1.2.1) |
| | Niepoprawna autoryzacja przez klienta | 403 | Zwrotne przekierowanie klienta w domenę TPP i zakończenie procesu (zgodnie z RFC6749 4.1.2.1) |
| Zwrotne przekierowanie klienta w domenę TPP | Wyłączenie urządzenia końcowego klienta bezpośrednio przed przekierowaniem | - | TPP ma możliwość dokończenia operacji poprzez pobranie tokenu przy pomocy contextId |
| Pobranie tokenu autoryzacyjnego przez TPP od ASPSP | Nieznany TPP | 401 | Zakończenie procesu bez przyjęcia dyspozycji |
| | Niepoprawne parametry żądania tokenu | 400 | Zakończenie procesu bez przyjęcia dyspozycji |
| | Brak możliwości nawiązania komunikacji z ASPSP | - | Zakończenie procesu bez przyjęcia dyspozycji |

| | | | |
|---|---|---|---|
| ASPSP weryfikuje zgodność żądania inicjacji płatności z udzieloną zgodą klienta | Niezgodność z udzieloną zgodą | 401 | Zakończenie procesu bez przyjęcia dyspozycji |
| | Przekroczono limit żądania dla żądanej usługi | 429 | Zakończenie procesu bez przyjęcia dyspozycji |
| ASPSP rozpoczyna realizację płatności | Brak środków | 422 | Zakończenie procesu (zwracany kod błędu) |
| Przekazanie do TPP informacji o poprawności przyjęcia dyspozycji | Brak możliwości nawiązania komunikacji z TPP | - | Ponowienie komunikatu x3<br>Informacja o awarii<br>Możliwość ponowienia wszystkich komunikatów |
| Przekazywanie TPP przez ASPSP informacji o zmianach stanu realizacji dyspozycji | Brak możliwości nawiązania komunikacji z TPP | - | Ponowienie komunikatu x3<br>Informacja o awarii<br>Możliwość ponowienia wszystkich komunikatów |
| ASPSP weryfikuje zgodność żądania danych o rachunku płatniczym z udzieloną zgodą klienta | Niezgodność z udzieloną zgodą | 403 | Zakończenie procesu |
| ASPSP otrzymuje dowolne żądanie | Dostęp do serwisu zablokowany przez ASPSP | 403 | Zakończenie procesu |
| ASPSP otrzymuje dowolne żądanie | Dostęp do serwisu zablokowany przez PSU | 403 | Zakończenie procesu |

## 12.1 Kody błędów dla kodu odpowiedzi HTTP 403

| KOD | KOMUNIKAT |
|---|---|
| 1 | Niepoprawna weryfikacja licencji TPP / Incorrect verification of TPP licenses |
| 2 | Brak aktywacji usług TPP / No activation of TPP services |
| 3 | Odrzucenie żądania przez klienta / Rejection of the request by the client |
| 4 | Niepoprawna autoryzacja przez klienta / Incorrect authorization by the client |
| 5 | Niezgodność z udzieloną zgodą / Non-compliance with the consent given |
| 6 | Dostęp do serwisu zablokowany przez ASPSP / Access to the service blocked by ASPSP |
| 7 | Dostęp do serwisu zablokowany przez PSU / Access to the website blocked by the PSU |
| 8 | Zablokowany dostęp TPP do usług bankowych przez bank / Blocked TPP's access to bank services by the bank |
| 9 | Zablokowany dostęp TPP do usług bankowych przez klienta / Blocked TPP's access to bank services by the client |

# 13 Rekomendacje implementacji standardu

## 13.1 Obsługa przekroczenia maksymalnego dozwolonego czasu (*timeout*)

Ze względu na mogące wystąpić w trakcie przetwarzania żądania http zdarzenia typu *timeout*, ASPSP musi zapewnić weryfikację unikalności na w warstwie serwerowej na poziomie id wywołania (requestId). Po stwierdzeniu nie-unikalności wywołania ASPSP zwraca błąd 400.1 (Powtórzone wywołanie).

Rekomendowana wartość *timeout* (czasu przetwarzania żądania ) to 30 sekund.

## 13.2 Weryfikacja TPP

Autentykację TPP należy przeprowadzać w oparciu o certyfikaty komunikacyjny (TLS) oraz podpisujący (JSON Web Signature) sprawdzając jednocześnie czy certyfikaty odpowiadają identyfikatorowi TPP (tppId) w bazie danych ASPSP. Wartość tppId jest ustalana przez ASPSP podczas technicznej rejestracji TPP, sugerowaną wartością jest EUNIP TPP.

## 13.3 Serwer Autoryzacji

Zaleca się aby ASPSP posiadał w swojej konfiguracji dla danego client_id listę redirect_uri, które mogą być wykorzystane. Dzięki czemu ASPSP nie przekieruje klienta na niewłaściwy adres URI, który może być podłożony przez niezaufaną stronę.

## 13.4 Antyfraud

W celach przeciwdziałania potencjalnym oszustwom wprowadzona została dedykowana Klasa RequestHeader, przekazywana w każdym żądaniu, zawierająca informacje o PSU takie jak: adres IP oraz userAgent. Struktura będzie pomocna ASPSP w zaimplementowaniu mechanizmów zabezpieczających.

Dodatkowo rekomenduje się, aby podmioty uczestniczące w projekcie dokonywały wymiany informacji w zakresie podejrzeń nieautoryzowanych transakcji/skompromitowanych IP, itp.

# 14 Spis Załączników

Załącznik nr 1: PolishAPI-ver2.0.yaml
Załącznik nr 2: PolishAPI-CallBack-ver2.0.yaml