PolishAPI

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

Dokument opracowany przez Grupę Projektową ds. PolishAPI

12 grudnia 2019
Wersja 3.0

Licencja

Historia wersji

Nr wersji	Data publikacji	Typ zmiany	Uwagi
3.0	12.12.2019	`major`	wspierana
2.1.3	12.07.2019	`patch`	wspierana
2.1.2	19.02.2019	`patch`	wspierana
2.1.1	06.12.2018	`patch`	wspierana
2.1	18.09.2018	`minor`	wspierana
2.0	09.07.2018	`major`	nie wspierana
1.0	17.04.2018		nie wspierana

Zespół edytorski

Maciej Kostro, Związek Banków Polskich
Łukasz Jackowicz, Quercus

Wsparcie projektowe

Marcin Jankowski, Michał Podgajny, KPMG

Wsparcie prawne

Marta Stanisławska, Sławomir Szepietowski, Bird&Bird

Kontakt

info@polishapi.org

https://polishapi.org

Spis treści

Spis ilustracji

1 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.

Rachunek VAT - rachunek bankowy o ograniczonej funkcjonalności, związany z rachunkiem płatniczym.

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 –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), w wersji aktualnej w chwili publikacji niniejszego standardu.

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.

2 Wstęp

2.1 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 uprawnione 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.

2.2 Struktura dokumentu

Dokument składa się z dwóch zasadniczych części oraz załączników:

a) Części dotyczącej charakterystyki biznesowej Standardu PolishAPI (rozdziały 2 – 4)
b) Części dotyczącej rozwiązań technologicznych przyjętych w standardzie PolishAPI (rozdziały 0 – 13)
c) Załączników, których spis znajduje się w rozdziale 14

2.3 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 firmowych i korporacyjnych (np. wielopodpis).

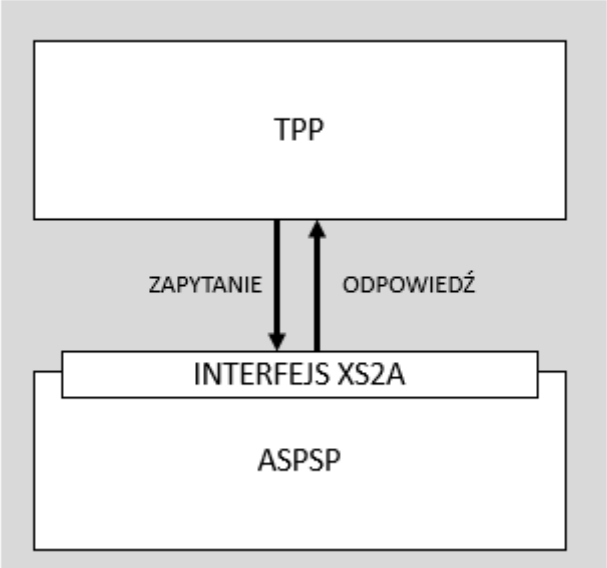
[1] Alior Bank S.A., Bank BGŻ BNP Paribas S.A., Bank Handlowy w Warszawie S.A., Bank Millennium S.A., Bank Pekao S.A., Bank Pocztowy S.A., Bank Polskiej Spółdzielczości S.A., Bank Zachodni WBK S.A., Credit Agricole Bank Polska S.A., Deutsche Bank Polska S.A., DNB Bank Polska S.A., Eurobank S.A., Getin 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.

2.4 Główne założenia

2.4.1 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



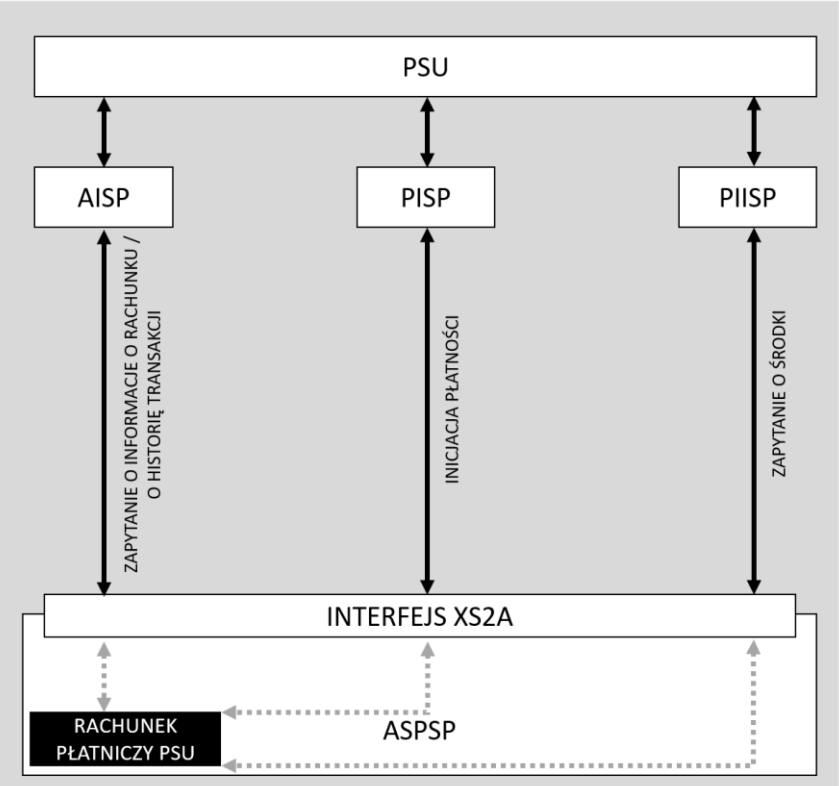
Ilustracja 1: Ogólny schemat komunikacji w Standardzie PolishAPI

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

2.4.2 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 „isCompanyContext" o wartości „*true*" w ciele wysyłanego żądania, zgodnie ze specyfikacją techniczną.

f) 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 musi być przez niego zdefiniowany, w formie słownika wartości, w szczegółowej specyfikacji interfejsu XS2A). Wartość tego parametru służy do wskazania na podstawie jakiej informacji zostanie zidentyfikowany PSU, który będzie podlegał uwierzytelnieniu. Parametr nie jest wymagany.
   - psuIdentifierValue – wartość identyfikatora PSU. Parametr jest wymagany tylko w przypadku przekazania niepustej wartości parametru psuIdentifierType.
   - psuContextIdentifierType – Typ identyfikatora kontekstu w jakim występuje PSU. (dostępny zakres identyfikatorów może być różny dla każdego ASPSP i musi być przez niego zdefiniowany, w formie słownika wartości, w szczegółowej specyfikacji interfejsu XS2A). Parametr jest wymagany warunkowo - w przypadku wysyłania żądania dla takiego PSU, który może występować w więcej niż jednym kontekście w wybranym ASPSP.,
   - psuContextIdentifierValue – wartość identyfikatora kontekstu w jakim występuje PSU. Parametr jest wymagany warunkowo - w przypadku przekazania niepustej wartości parametru psuContextIdentifierType Wymienione parametry służą do wskazania bardziej szczegółowego kontekstu wywołania metod interfejsu XS2A. Parametry powinny zostać użyte np. w przypadku takiego PSU, który jednocześnie jest pełnomocnikiem do rachunków wielu klientów w danym ASPSP.

2.4.3 Mechanizmy uwierzytelniania PSU

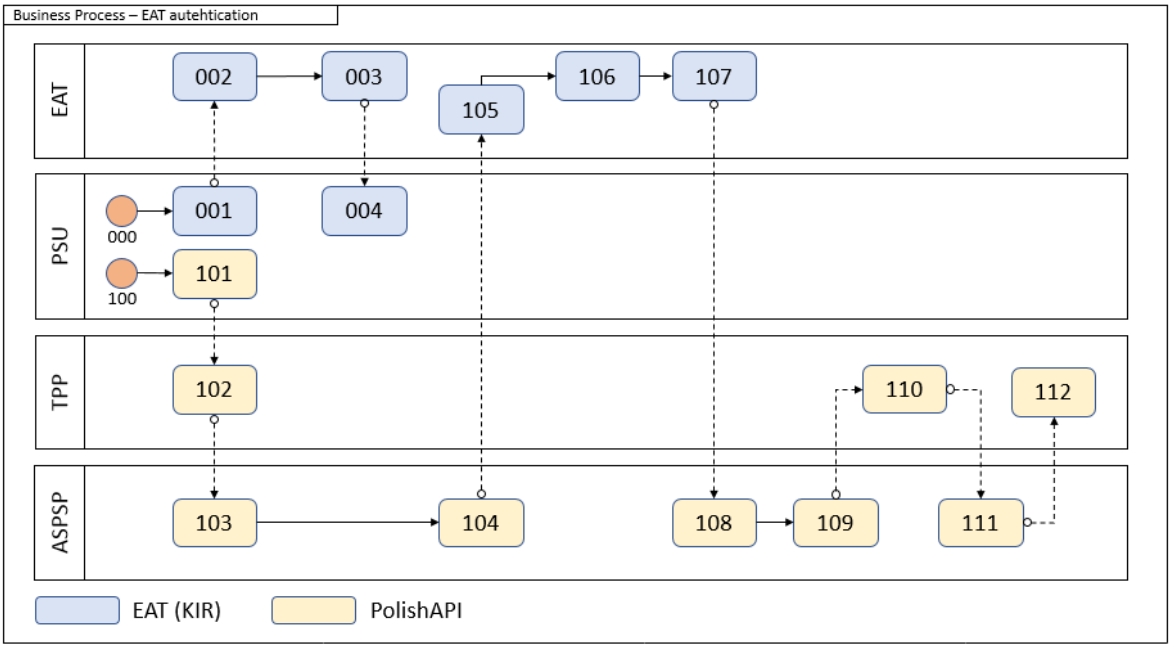
Standard PolishAPI dopuszcza poniższe mechanizmy uwierzytelnienia PSU. Wybór mechanizmów pozostaje wyłącznie w gestii ASPSP. Wybór ten powinien być zgodny z obowiązującymi regulacjami.

2.4.3.1 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.

2.4.3.2 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.

2.4.3.2.1 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.

2.4.3.2.2 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) – opcjonalnie użytkownik określa w tym kroku rachunek źródłowy (PIS) lub rachunek/rachunki objęte zgodą (AIS) jeśli nie zostało to zdefiniowane przez TPP w wywołaniu API.
- 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*.

2.4.3.3 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.

2.4.4 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.

2.4.4.1 Proces udzielenia zgody PSU na wykonanie usługi PIS

Zakłada się, że realizacja procesu inicjacji płatności (z datą bieżącą, przyszłą, płatności cyklicznych oraz płatności wielokrotnych, zgodnie z opisem w rozdziale 3.1.1) za każdym razem jest związana z udzieleniem na to zgody przez PSU w ramach interfejsu TPP. Procesy udzielania zgody, inicjacji płatności oraz pobrania statusu płatności w obligatoryjnych wariantach zakładających ręczne wprowadzenie numeru rachunku po stronie TPP oraz wybór rachunku po stronie ASPSP, a także odwoływania płatności zostały przedstawione na diagramach oraz w opisach w rozdziale 4.

2.4.4.2 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 ze wskazaniem rachunku po stronie ASPSP lub 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 ze wskazaniem rachunku po stronie ASPSP lub 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 trzy procesy udzielania zgody na usługę AIS (w opcjach uwzględniających uwierzytelnianie po stronie ASPSP oraz w zewnętrznym narzędziu autoryzacyjnym):

- z ręcznym wprowadzeniem numeru rachunku (rachunków)
- z wyborem numeru rachunku (rachunków) po stronie ASPSP (wyłącznie w opcji uwierzytelniania po stronie ASPSP)
- z pobraniem listy rachunków**.** Ten proces jest procesem opcjonalnym i jego implementacja zależy od decyzji ASPSP.

Procesy udzielania zgody oraz pobrania informacji o rachunku, w opisanych powyżej wariantach, zostały przedstawione na diagramach oraz w opisach w rozdziale 4.

2.4.4.3 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. Proces zapytania o dostępność środków został przedstawiony na diagramie oraz w opisie w rozdziale 4.

2.4.5 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.

2.4.6 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.

2.4.7 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.

2.5 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.

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, przelew cykliczny (seria przelewów), rozumiany jako definicja takiego przelewu lub wielokrotny (paczka przelewów), przy czym paczka przelewów może być utworzona wyłącznie z przelewów tego samego rodzaju z jednego numeru rachunku (np. wyłącznie z przelewów krajowych lub wyłącznie z przelewów EEA)
c) Jest to przelew z datą bieżącą lub datą przyszłą
d) 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.
e) Jeżeli jest to przelew zagraniczny, to rozliczany jest w jednym z poniższych systemów:
    a. SWIFT
    b. SEPA
    c. TARGET
f) Jest dostępny w ramach interfejsu online danego ASPSP
g) 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 uwzględniającego wybór rachunku po stronie ASPSP, wszystkie dane z wyjątkiem numeru rachunku, z którego płatność zostanie zainicjowana

Przelew wielokrotny (paczka przelewów) może być realizowana na dwa sposoby:

a) PSU definiuje n przelewów po stronie interfejsu TPP, a następnie przechodzi przez procedurę silnego uwierzytelniania, potwierdzając wszystkie zdefiniowane płatności jednocześnie,
b) PSU dokonuje uploadu ustrukturyzowanego pliku, w którym będzie umieszczona dowolna struktura, obsługiwana przez ASPSP. Przekazanie takiego pliku wiąże się z procedurą silnego uwierzytelniania (opis biznesowy oraz metoda API dla tego wariantu procesu zostanie dodana w kolejnej wersji specyfikacji).

Możliwe jest również zainicjowanie płatności podzielonej, tj. płatności, w której kwota płatności przekazywana jest na dwa rachunki odbiorcy (rachunek bieżący oraz rachunek VAT).

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 usług i danych dotyczących rachunków płatniczych dostępnych w ramach tej usługi, także w oparciu o dostępność poszczególnych usług w ramach bankowości online danego ASPSP. Realizacja usług w zakresie Zgodności nie będzie wymagała relacji umownej pomiędzy ASPSP a TPP.

3.1.2 Odwoływanie transakcji

Odwołaniu podlegają:

a) płatności pojedyncze z datą przyszłą ze statusem „zaplanowane”,
b) płatności cykliczne (definicja płatności),
c) pojedyncze płatności z datą przyszłą, definiowane w ramach płatności wielokrotnej (paczki przelewów), ze statusem „zaplanowane”.

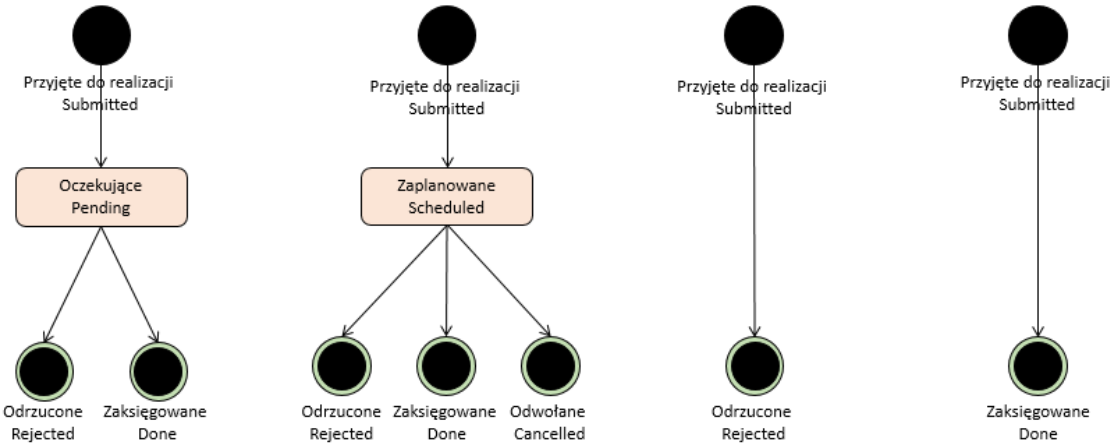
O ile ASPSP oferuje taką funkcjonalność, możliwe jest także odwołanie płatności wielokrotnej (paczki przelewów) jako całości, z zastrzeżeniem, że jeżeli w momencie odwołania w skład paczki wchodzą przelewy już wykonane, to odwołanie nie ma na nie wpływu i odwołanie skutkuje niewykonaniem wyłącznie płatności z datą przyszłą.

3.1.3 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 /{wersja}/accounts/{wersja}/paymentCallBack oraz o statusie definicji płatności cyklicznej /{wersja}/accounts/{wersja}/recurringPaymentCallBack.

Zdefiniowane są następujące statusy:

a) Przyjęte do realizacji (*submitted*)
b) Zaplanowane (*scheduled*)
c) Odwołane (*cancelled*)
d) Oczekujące (*pending*)
e) Odrzucone (*rejected*)
f) Zaksięgowane (*done*)



Ilustracja 4: Diagram statusów płatności

Płatności wielokrotne (paczki przelewów) przyjmują następujące statusy:

a) W realizacji (*inProgress*) – zawiera przynajmniej jedną transakcję z datą przyszłą;

b) częściowo zrealizowana (*partiallyDone*) – zawiera przynajmniej jedną transakcję o statusie done;
c) odwołana (*cancelled*);
d) zrealizowana (*done*) – wszystkie transakcje wchodząc w skład bundle mają status done

Definicja płatności cyklicznej przyjmuje następujące statusy:

a) Definicja płatności cyklicznej (zlecenia stałego) została wysłana prawidłowo i przyjęta po stronie ASPSP (*submitted*)
b) Płatność cykliczna (zlecenie stałe) jest prawidłowo zdefiniowana i realizowana zgodnie z harmonogramem (data końcowa w przyszłości lub bezterminowo) (*in progres*)
c) Definicja płatności cyklicznej (zlecenia stałego) została usunięta i nie jest realizowana (*cancelled*)
d) Definicja płatności cyklicznej (zlecenia stałego) miała datę końcową, która już minęła i nie jest już realizowana (*closed*).

3.1.4 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.

3.1.5 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

3.1.5.1 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	Podane przez TPP lub wybrane przez PSU po przekierowaniu do ASPSP.
Numer rachunku odbiorcy przelewu	Tak	
Pole opisujące przelew	Tak	
Tryb pilności	Tak	ExpressD0, StandardD1
Typ przelewu (system)	Tak	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	
Tryb realizacji przelewu	Tak	W jakim trybie przelew zostanie zrealizowany. Zgodnie z możliwościami opisanymi w następującym słowniku: - Natychmiastowo - Z datą przyszłą
Czy płatność z użyciem mechanizmu Split Payment	Nie	Wartość określająca czy przelew jest inicjowany z użyciem mechanizmu Split Payment. Domyślna wartość to false
Numer faktury	Warunkowo	Numer faktury, które dotyczy przelew. Wymagany gdy wykorzystywany jest mechanizm Split Payment.
Identyfikator odbiorcy przelewu	Warunkowo	Na przykład numer NIP. Wymagany gdy wykorzystywany jest mechanizm Split Payment.
Kwota podatku VAT	Warunkowo	Wymagany gdy wykorzystywany jest mechanizm Split Payment.
Opis dodatkowy	Nie	

3.1.5.2 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: 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	Podane przez TPP lub wybrane przez PSU po przekierowaniu do ASPSP.
Numer rachunku odbiorcy przelewu	Tak	
Symbol formularza	Tak	
Typ okresu	Tak	
Tryb pilności	Tak	ExpressD0, StandardD1
Typ przelewu (system)	Tak	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	
Tryb realizacji przelewu	Tak	W jakim trybie przelew zostanie zrealizowany. Zgodnie z możliwościami opisanymi w następującym słowniku: - Natychmiastowo - Z datą przyszłą

3.1.5.3 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	Podane przez TPP lub wybrane przez PSU po przekierowaniu do ASPSP.
Numer rachunku odbiorcy przelewu	Tak	
Kod kraju odbiorcy przelewu	Tak	Zgodny z normą ISO 3166-1 alfa-3
Pole opisujące przelew	Tak	
Tryb pilności	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	
Tryb realizacji przelewu	Tak	W jakim trybie przelew zostanie zrealizowany. Zgodnie z możliwościami opisanymi w następującym słowniku: - Natychmiastowo - Z datą przyszłą

3.1.5.4 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	Podane przez TPP lub wybrane przez PSU po przekierowaniu do ASPSP.
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	
Kod kraju odbiorcy przelewu	Tak	Zgodny z normą ISO 3166-1 alfa-3
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 pilności	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	
Tryb realizacji przelewu	Tak	W jakim trybie przelew zostanie zrealizowany. Zgodnie z możliwościami opisanymi w następującym słowniku: - Natychmiastowo - Z datą przyszłą

ASPSP może, dla autoryzacji przelewów, których konieczne parametry nie zostały przewidziane w powyższej tabeli, zdefiniować i udokumentować zestaw takich dodatkowych parametrów. Parametry te powinny być opcjonalne (i ewentualnie mieć podaną wartość domyślną) dla wszystkich przelewów, których realizacja jest możliwa bez ich podania (tj. dla krajów których rachunki mogą być poprawnie zaadresowane z użyciem pól z powyższej tabeli). Interpretacja parametrów dodatkowych nie może prowadzić do sprzeczności z opublikowanym znaczeniem parametrów ujętych w standardzie.

3.1.5.5 Płatność cykliczna (zlecenie stałe)

Każdy z typów płatności może zostać zdefiniowany jako płatność cykliczna (zlecenie stałe), przy założeniu, że jest realizowany na rzecz tego samego beneficjenta i w tej samej kwocie. W takim przypadku, oprócz danych zdefiniowanych w powyższych rozdziałach, niezbędne będzie podanie następujących danych:

NAZWA POLA	WYMAGALNOŚĆ	KOMENTARZE
Start date (Data wykonania pierwszej płatności)	Tak	Data wykonania pierwszej płatności w definiowanym cyklu
Frequency (Częstotliwość)	Tak	Określa jak często ma być wykonywany przelew cykliczny.
periodType	Tak	Typ jednostki okresu czasu
periodValue	Tak	Wartość jednostki okresu czasu
End date (Ostatnia możliwa data przelewu)	Nie	Określa ostatnią możliwą datę, kiedy przelew cykliczny może zostać wykonany.
Realizacja w dzień wolny	Tak	Określa zachowanie w przypadku, gdy data zlecenia przypada na dzień wolny, możliwe wartości to „przed” i „po” dniu wolnym.

3.1.5.6 Odwołanie zainicjowanej płatności

NAZWA POLA	WYMAGALNOŚĆ	KOMENTARZE
Identyfikator płatności	Warunkowo	Wymagany dla żądania odwołania pojedynczej płatności
Identyfikator paczki przelewów	Warunkowo	Wymagany dla żądania odwołania paczki przelewów

3.1.5.7 Odwołanie definicji płatności cyklicznej

NAZWA POLA	WYMAGALNOŚĆ	KOMENTARZE
Identyfikator płatności cyklicznej	Tak	Identyfikator płatności cyklicznej nadany przez ASPSP

3.1.6 Diagramy zapytania w ramach usługi PIS w zakresie Zgodności

Diagramy zostały przedstawione w Przypadku Użycia #1 w rozdziale 4.

3.1.7 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).

Jeżeli zapytanie, które nie jest inicjowane na żądanie PSU, zawiera w sobie stronicowane wyniki, powinno być traktowane jako pojedyncze zapytanie. Funkcjonalność zliczania zapytań w zadanym przedziale czasu pozostaje w gestii ASPSP na poziomie implementacji, przy czym musi zostać uwzględniona logika biznesowa zapytania, które może być realizowane przez wywołanie kilku metod interfejsu (np. zapytanie o historię transakcji na rachunku wiąże się z wywołaniem metod:

getTransactionsDone, getTransactionsPending, getTransactionsScheduled, getTransactionsCancelled, getHolds oraz getTransactionDetail – wywołanie wszystkich wymienionych metod interfejsu powinno być potraktowane jako pojedyncze zapytanie.

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), operacja zaplanowana (*scheduled*) oznacza płatność zleconą z datą przyszłą. 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. Ponowne silne uwierzytelnienie PSU po upływie 90-dniowego okresu od ostatniego silnego uwierzytelnienia, jeżeli zakres usługi nie ulega żadnym zmianom, jest przeprowadzane przy użyciu identyfikatora zgody (consentId), bez konieczności ponownego przekazywania parametrów dostępu do rachunku, zdefiniowanych wcześniej przez PSU. TPP może zainicjować taki proces przed upływem 90-dniowego okresu, przedłużając tym samym okres, przez który PSU będzie miał dostęp do danych bez konieczności silnego uwierzytelnienia.

3.2.4 Lista pól udostępnianych przez ASPSP w zakresie Zgodności

W ramach odpowiedzi na przesłane przez TPP (AISP) żądania, ASPSP przesyła odpowiedzi w zakresie poniższych pól, uporządkowanych według metod interfejsu.
/getAccount

NAZWA POLA	WYMAGALNOŚĆ	KOMENTARZ
Numer rachunku	TAK	
Typ rachunku	NIE	Wartość słownikowa
Nazwa typu rachunku	NIE	Definiowana przez ASPSP
Rodzaj posiadacza rachunku	TAK	Osoba fizyczna lub osoba prawna
Nazwa konta	NIE	Ustawiona przez klienta
Waluta rachunku	TAK	
Dostępne środki	TAK	
Saldo księgowe rachunku	TAK	
Numer BIC/SWIFT Banku	NIE	
Nazwa Banku	NIE	
Imię Nazwisko lub Nazwa PSU	NIE	
Typ relacji PSU do rachunku	Tak	Wartość słownikowa
Typ pełnomocnictwa PSU do rachunku	Nie	Wartość słownikowa
Numer rachunku VAT	Nie	Numer, który jest powiązany z danym rachunkiem

		rozliczeniowym/lokacyjnym.
Numery rachunków rozliczeniowych/lokacyjnych	Nie	Lista wszystkich numerów rachunków rozliczeniowych/lokacyjnych, powiązanych z danym rachunkiem VAT

/getTransactionsDone

NAZWA POLA	WYMAGALNOŚĆ	KOMENTARZ
Identyfikator elementu	TAK	Identyfikator transakcji lub blokady, nadany przez ASPSP
Kwota transakcji	TAK	
Waluta transakcji	NIE	Kod ISO waluty transakcji
Tytuł transakcji	TAK	
Data operacji	NIE	
Kod mcc	NIE	Merchant Category Code dla każdej transakcji/operacji wykonanej przy użyciu karty
Typ transakcji	NIE	
Kategoria transakcji	TAK	uznanie/obciążenie
Status transakcji	NIE	Wartość słownikowa
Dane podmiotu inicjującego	NIE	W przypadku transakcji zlecanych przez osobę inną niż właściciel rachunku, nazwa i adres
Numer konta nadawcy	NIE	
Numer wirtualny rachunku nadawcy	NIE	W formacie IBAN
Numer BIC/SWIFT Banku nadawcy	NIE	Tylko dla przelewów zagranicznych
Nazwa Banku nadawcy	NIE	Tylko dla przelewów zagranicznych
Kod Banku nadawcy	NIE	Tylko dla przelewów zagranicznych
Kod kraju nadawcy	NIE	Tylko dla przelewów zagranicznych
Adres Banku nadawcy	NIE	Tylko dla przelewów zagranicznych
Dane nadawcy	NIE	Nazwa i adres
Numer konta odbiorcy	NIE	
Numer wirtualny rachunku odbiorcy	NIE	W formacie IBAN
Numer BIC/SWIFT Banku odbiorcy	NIE	Tylko dla przelewów zagranicznych
Nazwa Banku odbiorcy	NIE	Tylko dla przelewów zagranicznych
Kod Banku odbiorcy	NIE	Tylko dla przelewów zagranicznych
Kod kraju odbiorcy	NIE	Tylko dla przelewów zagranicznych
Adres Banku odbiorcy	NIE	Tylko dla przelewów zagranicznych
Dane odbiorcy	NIE	Nazwa i adres
Data księgowania	NIE	
Saldo księgowe	NIE	Saldo księgowe rachunku po transakcji

/getTransactionsPending

NAZWA POLA	WYMAGALNOŚĆ	KOMENTARZ
Identyfikator elementu	TAK	Identyfikator transakcji lub blokady, nadany przez ASPSP
Kwota transakcji	TAK	
Waluta transakcji	NIE	Kod ISO waluty transakcji
Tytuł transakcji	TAK	
Data operacji	NIE	
Kod mcc	NIE	Merchant Category Code dla każdej transakcji/operacji wykonanej przy użyciu karty
Typ transakcji	NIE	

Kategoria transakcji	TAK	uznanie/obciążenie
Dane podmiotu inicjującego	NIE	W przypadku transakcji zlecanych przez osobę inną niż właściciel rachunku, nazwa i adres
Numer konta nadawcy	NIE	
Numer wirtualny rachunku nadawcy	NIE	W formacie IBAN
Numer BIC/SWIFT Banku nadawcy	NIE	Tylko dla przelewów zagranicznych
Nazwa Banku nadawcy	NIE	Tylko dla przelewów zagranicznych
Kod Banku nadawcy	NIE	Tylko dla przelewów zagranicznych
Kod kraju nadawcy	NIE	Tylko dla przelewów zagranicznych
Adres Banku nadawcy	NIE	Tylko dla przelewów zagranicznych
Dane nadawcy	NIE	Nazwa i adres
Numer konta odbiorcy	NIE	
Numer wirtualny rachunku odbiorcy	NIE	W formacie IBAN
Numer BIC/SWIFT Banku odbiorcy	NIE	Tylko dla przelewów zagranicznych
Nazwa Banku odbiorcy	NIE	Tylko dla przelewów zagranicznych
Kod Banku odbiorcy	NIE	Tylko dla przelewów zagranicznych
Kod kraju odbiorcy	NIE	Tylko dla przelewów zagranicznych
Adres Banku odbiorcy	NIE	Tylko dla przelewów zagranicznych
Dane odbiorcy	NIE	Nazwa i adres

/getTransactionsRejected

NAZWA POLA	WYMAGALNOŚĆ	KOMENTARZ
Identyfikator elementu	TAK	Identyfikator transakcji lub blokady, nadany przez ASPSP
Kwota transakcji	TAK	
Waluta transakcji	NIE	Kod ISO waluty transakcji
Tytuł transakcji	TAK	
Data operacji	NIE	
Kod mcc	NIE	Merchant Category Code dla każdej transakcji/operacji wykonanej przy użyciu karty
Typ transakcji	NIE	
Kategoria transakcji	TAK	uznanie/obciążenie
Dane podmiotu inicjującego	NIE	W przypadku transakcji zlecanych przez osobę inną niż właściciel rachunku, nazwa i adres
Numer konta nadawcy	NIE	
Numer wirtualny rachunku nadawcy	NIE	W formacie IBAN
Numer BIC/SWIFT Banku nadawcy	NIE	Tylko dla przelewów zagranicznych
Nazwa Banku nadawcy	NIE	Tylko dla przelewów zagranicznych
Kod Banku nadawcy	NIE	Tylko dla przelewów zagranicznych
Kod kraju nadawcy	NIE	Tylko dla przelewów zagranicznych
Adres Banku nadawcy	NIE	Tylko dla przelewów zagranicznych
Dane nadawcy	NIE	Nazwa i adres
Numer konta odbiorcy	NIE	
Numer wirtualny rachunku odbiorcy	NIE	W formacie IBAN
Numer BIC/SWIFT Banku odbiorcy	NIE	Tylko dla przelewów zagranicznych

Nazwa Banku odbiorcy	NIE	Tylko dla przelewów zagranicznych
Kod Banku odbiorcy	NIE	Tylko dla przelewów zagranicznych
Kod kraju odbiorcy	NIE	Tylko dla przelewów zagranicznych
Adres Banku odbiorcy	NIE	Tylko dla przelewów zagranicznych
Dane odbiorcy	NIE	Nazwa i adres
Powód odrzucenia	NIE	
Data odrzucenia	NIE	

/getTransactionsCancelled

NAZWA POLA	WYMAGALNOŚĆ	KOMENTARZ
Identyfikator elementu	TAK	Identyfikator transakcji lub blokady, nadany przez ASPSP
Kwota transakcji	TAK	
Waluta transakcji	NIE	Kod ISO waluty transakcji
Tytuł transakcji	TAK	
Data operacji	NIE	
Kod mcc	NIE	Merchant Category Code dla każdej transakcji/operacji wykonanej przy użyciu karty
Typ transakcji	NIE	
Kategoria transakcji	TAK	uznanie/obciążenie
Status transakcji	NIE	Wartość słownikowa
Dane podmiotu inicjującego	NIE	W przypadku transakcji zlecanych przez osobę inną niż właściciel rachunku, nazwa i adres
Numer konta nadawcy	NIE	
Numer wirtualny rachunku nadawcy	NIE	W formacie IBAN
Numer BIC/SWIFT Banku nadawcy	NIE	Tylko dla przelewów zagranicznych
Nazwa Banku nadawcy	NIE	Tylko dla przelewów zagranicznych
Kod Banku nadawcy	NIE	Tylko dla przelewów zagranicznych
Kod kraju nadawcy	NIE	Tylko dla przelewów zagranicznych
Adres Banku nadawcy	NIE	Tylko dla przelewów zagranicznych
Dane nadawcy	NIE	Nazwa i adres
Numer konta odbiorcy	NIE	
Numer wirtualny rachunku odbiorcy	NIE	W formacie IBAN
Numer BIC/SWIFT Banku odbiorcy	NIE	Tylko dla przelewów zagranicznych
Nazwa Banku odbiorcy	NIE	Tylko dla przelewów zagranicznych
Kod Banku odbiorcy	NIE	Tylko dla przelewów zagranicznych
Kod kraju odbiorcy	NIE	Tylko dla przelewów zagranicznych
Adres Banku odbiorcy	NIE	Tylko dla przelewów zagranicznych
Dane odbiorcy	NIE	Nazwa i adres

/getTransactionsScheduled

NAZWA POLA	WYMAGALNOŚĆ	KOMENTARZ
Identyfikator elementu	TAK	Identyfikator transakcji lub blokady, nadany przez ASPSP
Kwota transakcji	TAK	
Waluta transakcji	NIE	Kod ISO waluty transakcji

Tytuł transakcji	TAK	
Data operacji	NIE	
Kod mcc	NIE	Merchant Category Code dla każdej transakcji/operacji wykonanej przy użyciu karty
Typ transakcji	NIE	
Kategoria transakcji	TAK	uznanie/obciążenie
Status transakcji	NIE	Wartość słownikowa
Dane podmiotu inicjującego	NIE	W przypadku transakcji zlecanych przez osobę inną niż właściciel rachunku, nazwa i adres
Numer konta nadawcy	NIE	
Numer wirtualny rachunku nadawcy	NIE	W formacie IBAN
Numer BIC/SWIFT Banku nadawcy	NIE	Tylko dla przelewów zagranicznych
Nazwa Banku nadawcy	NIE	Tylko dla przelewów zagranicznych
Kod Banku nadawcy	NIE	Tylko dla przelewów zagranicznych
Kod kraju nadawcy	NIE	Tylko dla przelewów zagranicznych
Adres Banku nadawcy	NIE	Tylko dla przelewów zagranicznych
Dane nadawcy	NIE	Nazwa i adres
Numer konta odbiorcy	NIE	
Numer wirtualny rachunku odbiorcy	NIE	W formacie IBAN
Numer BIC/SWIFT Banku odbiorcy	NIE	Tylko dla przelewów zagranicznych
Nazwa Banku odbiorcy	NIE	Tylko dla przelewów zagranicznych
Kod Banku odbiorcy	NIE	Tylko dla przelewów zagranicznych
Kod kraju odbiorcy	NIE	Tylko dla przelewów zagranicznych
Adres Banku odbiorcy	NIE	Tylko dla przelewów zagranicznych
Dane odbiorcy	NIE	Nazwa i adres

/getHolds

NAZWA POLA	WYMAGALNOŚĆ	KOMENTARZ
Identyfikator elementu	TAK	blokady, nadany przez ASPSP
Kwota blokady	TAK	
Waluta	NIE	Kod ISO waluty
Tytuł transakcji	TAK	
Data operacji	NIE	
Kod mcc	NIE	Merchant Category Code dla każdej transakcji/operacji wykonanej przy użyciu karty
Typ transakcji	NIE	uznanie/obciążenie
Data ważności blokady	NIE	
Dane podmiotu inicjującego	NIE	W przypadku transakcji zlecanych przez osobę inną niż właściciel rachunku, nazwa i adres
Numer konta nadawcy	NIE	
Numer wirtualny rachunku nadawcy	NIE	W formacie IBAN
Numer BIC/SWIFT Banku nadawcy	NIE	Tylko dla przelewów zagranicznych
Nazwa Banku nadawcy	NIE	Tylko dla przelewów zagranicznych
Kod Banku nadawcy	NIE	Tylko dla przelewów zagranicznych
Kod kraju nadawcy	NIE	Tylko dla przelewów zagranicznych

Adres Banku nadawcy	NIE	Tylko dla przelewów zagranicznych
Dane nadawcy	NIE	Nazwa i adres
Numer konta odbiorcy	NIE	
Numer wirtualny rachunku odbiorcy	NIE	W formacie IBAN
Numer BIC/SWIFT Banku odbiorcy	NIE	Tylko dla przelewów zagranicznych
Nazwa Banku odbiorcy	NIE	Tylko dla przelewów zagranicznych
Kod Banku odbiorcy	NIE	Tylko dla przelewów zagranicznych
Kod kraju odbiorcy	NIE	Tylko dla przelewów zagranicznych
Adres Banku odbiorcy	NIE	Tylko dla przelewów zagranicznych
Dane odbiorcy	NIE	Nazwa i adres

/getTransactionDetail

NAZWA POLA	WYMAGALNOŚĆ	KOMENTARZ
Identyfikator elementu	TAK	Identyfikator transakcji lub blokady, nadany przez ASPSP
Kwota transakcji	TAK	
Waluta transakcji	NIE	Kod ISO waluty transakcji
Tytuł transakcji	TAK	
Data operacji	NIE	
Kod mcc	NIE	Merchant Category Code dla każdej transakcji/operacji wykonanej przy użyciu karty
Typ transakcji	NIE	
Kategoria transakcji	TAK	uznanie/obciążenie
Status transakcji	NIE	Wartość słownikowa
Dane podmiotu inicjującego	NIE	W przypadku transakcji zlecanych przez osobę inną niż właściciel rachunku, nazwa i adres
Numer konta nadawcy	NIE	
Numer wirtualny rachunku nadawcy	NIE	W formacie IBAN
Numer BIC/SWIFT Banku nadawcy	NIE	Tylko dla przelewów zagranicznych
Nazwa Banku nadawcy	NIE	Tylko dla przelewów zagranicznych
Kod Banku nadawcy	NIE	Tylko dla przelewów zagranicznych
Kod kraju nadawcy	NIE	Tylko dla przelewów zagranicznych
Adres Banku nadawcy		Tylko dla przelewów zagranicznych
Dane nadawcy	NIE	Nazwa i adres
Numer konta odbiorcy	NIE	
Numer wirtualny rachunku odbiorcy	NIE	W formacie IBAN
Numer BIC/SWIFT Banku odbiorcy	NIE	Tylko dla przelewów zagranicznych
Nazwa Banku odbiorcy	NIE	Tylko dla przelewów zagranicznych
Kod Banku odbiorcy	NIE	Tylko dla przelewów zagranicznych
Kod kraju odbiorcy	NIE	Tylko dla przelewów zagranicznych
Adres Banku odbiorcy	NIE	Tylko dla przelewów zagranicznych
Dane odbiorcy	NIE	Nazwa i adres
Data księgowania	NIE	
Saldo	NIE	Saldo rachunku po transakcji
Identyfikator elementu	TAK	blokady, nadany przez ASPSP
Numer Identyfikacji Podatkowej	NIE	Podstawowy identyfikator płatnika do ZUS będący numerem NIP.
Dodatkowy numer	NIE	Wartość dodatkowego identyfikatora płatnika do ZUS

identyfikacyjny płatnika		(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.
Typ wpłaty	NIE	Tylko dla przelewu 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 typu płatności	NIE	Wartość identyfikatora zobowiązania, z którego wynika należność dla przelewów do ZUS, zgodny z formularzem tego typu przelewów
Numer tytułu wykonawczego	NIE	Wartość numeru deklaracji dla przelewów do ZUS, zgodny z formularzem tego typu przelewów
Identyfikator płatnika	TAK	Tylko dla przelewu do Organu Podatkowego / Izby Celnej w Polsce
Typ identyfikatora płatnika	TAK	Tylko dla przelewu do Organu Podatkowego / Izby Celnej w Polsce
Symbol formularza Urzędu Skarbowego lub Izby Celnej	TAK	Tylko dla przelewu do Organu Podatkowego / Izby Celnej w Polsce
Numer okresu	WARUNKOWO	Wymagany warunkowo - w zależności od wartości parametru w polu Symbol formularza. Tylko dla przelewu do Organu Podatkowego / Izby Celnej w Polsce.
Typ okresu	WARUNKOWO	Wymagany warunkowo - w zależności od wartości parametru w polu Symbol formularza. Tylko dla przelewu do Organu Podatkowego / Izby Celnej w Polsce.
Rok okresu	WARUNKOWO	Wymagany warunkowo - w zależności od wartości parametru w polu Symbol formularza. Tylko dla przelewu do Organu Podatkowego / Izby Celnej w Polsce.
Identyfikator zobowiązania, z którego wynika należność podatku np. decyzja, tytuł wykonawczy, postanowienie	NIE	Tylko dla przelewu do Organu Podatkowego / Izby Celnej w Polsce
Właściciel karty	NIE	
Numer karty	NIE	
Data kursu waluty	NIE	
Kursy przewalutowania	NIE	
Kod waluty przed przewalutowaniem transakcji	NIE	Kod ISO
Kod waluty po przewalutowaniu transakcji	NIE	
Waluta oryginalna transakcji	NIE	Kod ISO
Kwota w oryginalnej walucie	NIE	
Unikalny identyfikator instrumentu płatniczego, za którego pomocą wykonano transakcję	NIE	
Unikalny identyfikator transakcji po stronie TPP	NIE	W przypadku transakcji inicjowanych w ramach usługi PIS
Nazwa TPP	NIE	W przypadku transakcji inicjowanych w ramach usługi PIS
Przyczyna odrzucenia	NIE	W przypadku transakcji odrzuconych
Data ważności blokady	NIE	W przypadku blokad na rachunku
Czy płatność z użyciem mechanizmu Split Payment	Nie	Wartość określająca czy przelew został inicjowany z użyciem mechanizmu Split Payment. Domyślna wartość to false

Numer faktury	Warunkowo	Numer faktury, która dotyczy zainicjowanego przelewu. Wymagany gdy wykorzystany został mechanizm Split Payment.
Identyfikator odbiorcy przelewu	Warunkowo	Na przykład numer NIP. Wymagany gdy wykorzystany został mechanizm Split Payment.
Kwota podatku VAT	Warunkowo	Wymagany gdy wykorzystany został mechanizm Split Payment.
Opis dodatkowy	Nie	

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, HoldInfo, 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.

/getAccounts

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. W przypadku maskowania numeru rachunku ASPSP powinno zapewnić mechanizm odszyfrowania tego numeru, tak, aby możliwa była realizacja usługi AIS.
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.
Typ relacji PSU do rachunku	Tak	Wartość słownikowa
Typ pełnomocnictwa PSU do rachunku	Nie	Wartość słownikowa

3.2.5 Diagramy zapytań w ramach usługi AIS w zakresie Zgodności

Diagramy zostały przedstawione 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..

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 płatności przez PISP (PIS)

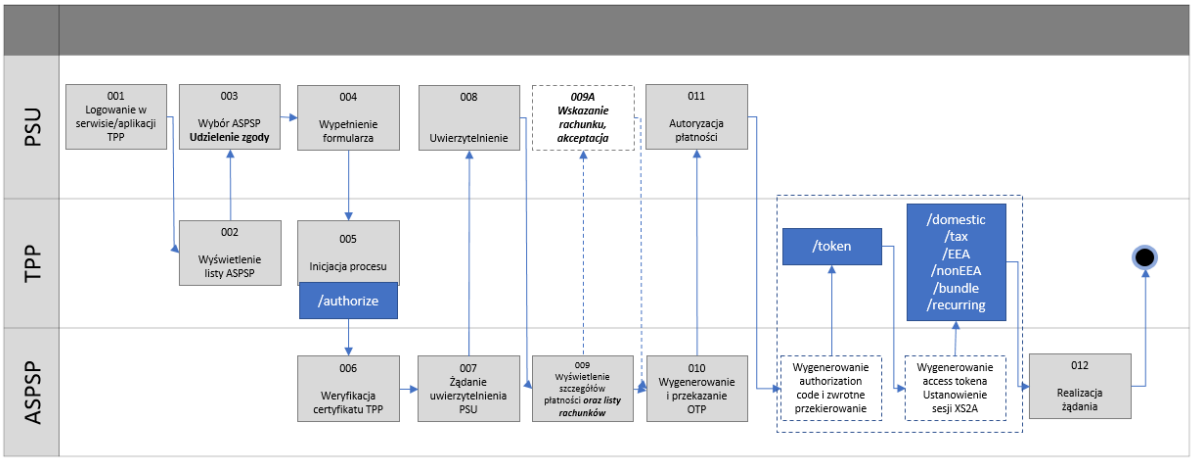
Wykorzystanie usługi PIS w zakresie Zgodności zaprezentowane w tym Przypadku Użycia polega na zainicjowaniu przez TPP występującego w roli PISP transakcji płatniczej w ciężar rachunku płatniczego PSU prowadzonego przez ASPSP, w oparciu o stosowne zapisy UUP. ASPSP może odrzucić transakcję, jeżeli TPP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

4.1.1 Udzielenie zgody i realizacja inicjacji płatności (płatność pojedyncza z datą bieżącą lub przyszłą, płatność cykliczna, płatność wielokrotna – paczka przelewów) – uwierzytelnianie 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, w tym zgodę na inicjację przelewu, przelewu cyklicznego lub paczki przelewów oraz na odpytanie o aktualny status przelewu, przelewu cyklicznego lub paczki przelewów, już po ich zainicjowaniu
- 004 / PSU wypełnia formularz przelewu, przelewu cyklicznego lub wielu przelewów, spełniający wymagania opisane w rozdziale 3.1.1, który powinien zawierać przynajmniej informacje, wskazane w rozdziale 3.1.4 niniejszej specyfikacji: „Lista pól wymaganych przez ASPSP w zakresie Zgodności” – w zależności od opcji wraz z numerem rachunku lub bez niego

- 005 / TPP inicjuje proces PIS (użycie metody /authorize, w tym przekazanie scope i scope_details), 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 o ile jest wymagane)
- 009 i 009A / ASPSP wyświetla PSU szczegóły transakcji oraz listę rachunków (w opcji z wyborem rachunku po stronie ASPSP), PSU wskazuje rachunek, z którego zostanie zainicjowana płatność
- 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). Po zakończonej sukcesem autoryzacji przelewu, przelewu cyklicznego lub paczki przelewów przez PSU, na linii ASPSP-TPP dochodzi do ustanowienia sesji interfejsu XS2A, czego efektem jest przekazanie przez ASPSP do TPP access tokena oraz refresh tokena (użycie metody /token oraz, po ustanowieniu sesji XS2A jednej z metod: /domestic, /tax, /EEA, /nonEEA, /bundle, /recurring)
- 012 / ASPSP realizuje żądanie, następuje przekierowanie do domeny TPP

proces inicjacji płatności kończy się



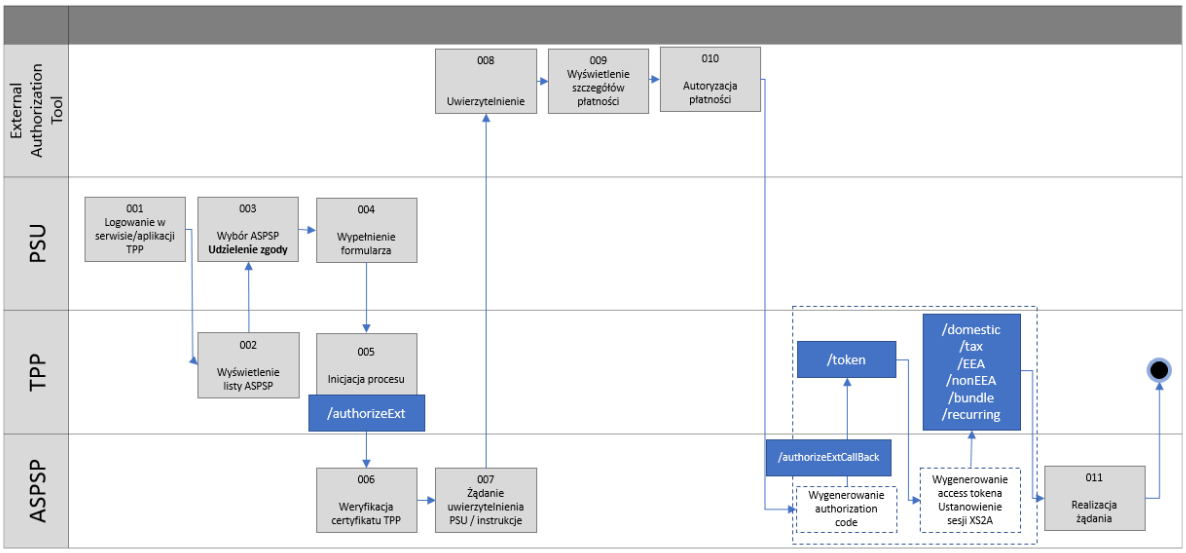
Ilustracja 5: PIS – uwierzytelnianie po stronie ASPSP

4.1.2 Udzielenie zgody i realizacja inicjacji płatności (płatność pojedyncza z datą bieżącą lub przyszłą, płatność cykliczna, płatność wielokrotna – paczka przelewów) – uwierzytelnianie w zewnętrznym narzędziu 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, w tym zgodę na inicjację przelewu, przelewu cyklicznego lub paczki przelewów oraz na odpytanie o aktualny status przelewu lub paczki przelewów, już po ich zainicjowaniu
- 004 / PSU wypełnia formularz przelewu, przelewu cyklicznego lub wielu przelewów, spełniający wymagania opisane w rozdziale 3.1.1, który powinien zawierać przynajmniej informacje, wskazane w rozdziale 3.1.4 niniejszej specyfikacji: „Lista pól wymaganych przez ASPSP w zakresie Zgodności" – wraz z numerem rachunku
- 005 / TPP inicjuje proces PIS (użycie metody /authorizeExt, w tym przekazanie scope i scope_details)
- 006 / ASPSP weryfikuje tożsamość TPP na podstawie certyfikatu (lub także na podstawie rejestru TPP)
- 007 / ASPSP inicjuje proces uwierzytelnienia PSU, w tym przekazuje do EAT instrukcję dotyczącą użycia 2 elementu autoryzacyjnego – o ile jest to wymagane zgodnie z obowiązującymi regulacjami
- 008 / Uwierzytelnienie (SCA o ile jest wymagane)
- 009 / W zewnętrznym narzędziu autoryzacyjnym wyświetlane są szczegóły płatności, PSU akceptuje transakcję
- 010 / PSU autoryzuje transakcję (PSU ma możliwość niedokonania autoryzacji, co skutkuje niezrealizowaniem transakcji płatniczej). Po zakończonej sukcesem autoryzacji przelewu, przelewu cyklicznego lub paczki przelewów przez PSU, na linii ASPSP-TPP dochodzi do ustanowienia sesji interfejsu XS2A, czego efektem jest przekazanie przez ASPSP do TPP access tokena oraz refresh tokena (użycie metody /token oraz, po ustanowieniu sesji XS2A jednej z metod: /domestic, /tax, /EEA, /nonEEA, /bundle, /recurring)
- 011 / ASPSP realizuje żądanie, następuje przekierowanie do domeny TPP

proces inicjacji płatności kończy się



Ilustracja 6: PIS – uwierzytelnianie w zewnętrznym narzędziu autoryzacyjnym

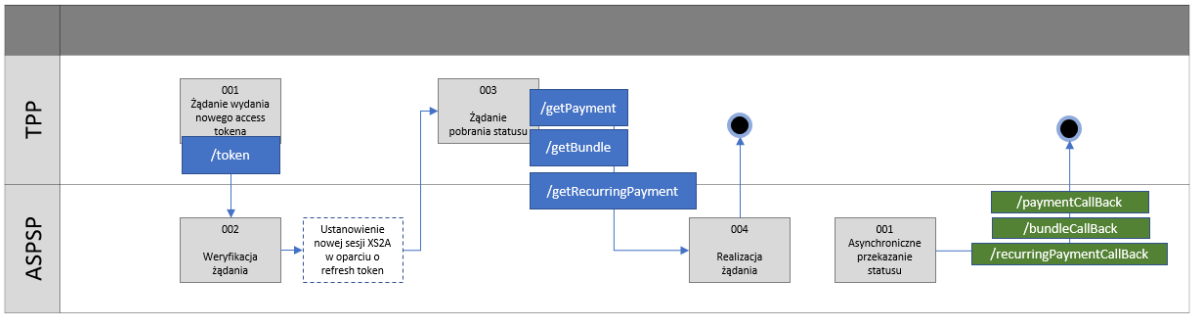
4.1.3 Zapytanie o status płatności (płatność pojedyncza z datą bieżącą lub przyszłą, płatność cykliczna, płatność wielokrotna – paczka przelewów)

- 001 / TPP wysyła żądanie wydania nowego access tokena (nawiązania nowej sesji komunikacyjnej zgodnie z opisem w punkcie 11.3), w oparciu o wartość refresh tokena uzyskanego w kroku 011 (uwierzytelnianie po stronie ASPSP) lub 010 (uwierzytelnianie po stronie zewnętrznego narzędzia autoryzacyjnego), oraz nowy zakres zgód, który obejmuje możliwość odpytywania o status przelewu lub paczki przelewów
- 002 / ASPSP weryfikuje żądanie TPP, w szczególności przekazaną wartość refresh tokena oraz dane opisujące żądaną zgodę, po czym nawiązuje nową sesję komunikacyjną interfejsu XS2A i przekazuje do TPP nowe wartości access tokena i refresh tokena, które tę sesję identyfikują
- 003 / TPP wysyła żądania usługi PIS dotyczące pobrania statusu przelewu, przelewu cyklicznego lub paczki przelewów, z wykorzystaniem opisanego wyżej access tokena
- 004 / ASPSP realizuje żądanie

proces zapytania o status płatności kończy się

- 001 / ASPSP asynchronicznie przekazuje status płatności

proces przekazania statusu płatności kończy się



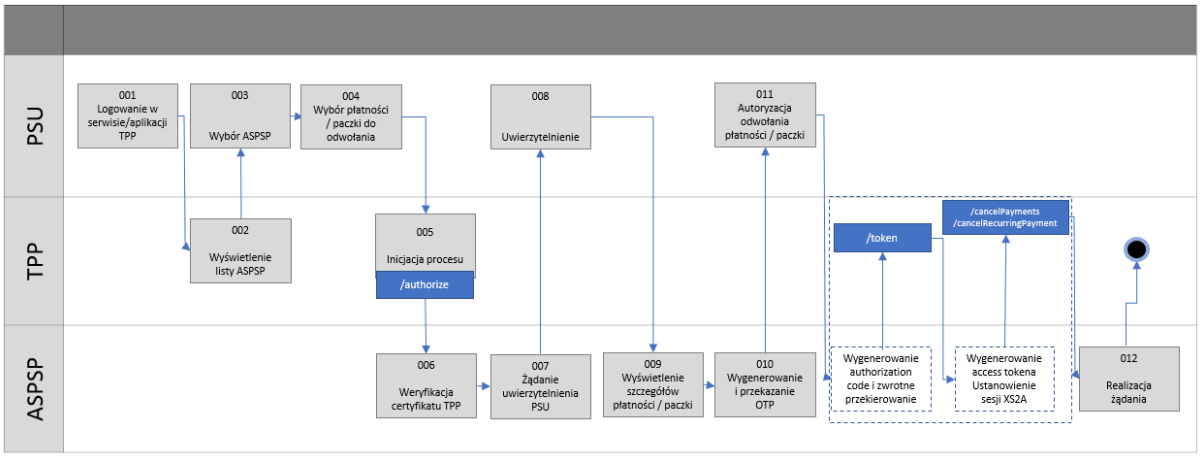
Ilustracja 7: PIS – zapytanie o status

4.1.4 Odwołanie płatności (płatność pojedyncza z datą przyszłą, płatność cykliczna z datą przyszłą, pojedyncza płatność w ramach płatności wielokrotnej (z datą przyszłą) lub płatność wielokrotna – paczka przelewów) – uwierzytelnianie po stronie ASPSP

- 001 / PSU inicjuje proces w interfejsie TPP
- 002 / TPP wyświetla listę ASPSP
- 003 / PSU wybiera z listy ASPSP
- 004 / PSU wybiera płatność, definicję płatności cyklicznej lub paczkę do odwołania
- 005 / TPP inicjuje proces odwołania (użycie metody /authorize, w tym przekazanie scope i scope_details)

- 006 / ASPSP weryfikuje tożsamość TPP na podstawie certyfikatu (lub także na podstawie rejestru TPP)
- 007 / ASPSP inicjuje proces uwierzytelnienia PSU
- 008 / Uwierzytelnienie
- 009 / ASPSP wyświetla PSU szczegóły odwoływanej płatności
- 010 / ASPSP generuje i przekazuje PSU dodatkowy element autoryzacyjny (np. OTP)
- 011 / PSU autoryzuje odwołanie płatności wg metody stosowanej w relacjach z ASPSP. Po zakończonej sukcesem autoryzacji odwołania, na linii ASPSP-TPP dochodzi do ustanowienia sesji interfejsu XS2A, czego efektem jest przekazanie przez ASPSP do TPP access tokena (użycie metody /token oraz, po ustanowieniu sesji XS2A metody / cancelPayments lub /cancelReccurringPayment
- 012 / ASPSP realizuje żądanie, następuje przekierowanie do domeny TPP

proces odwołania płatności / paczki kończy się



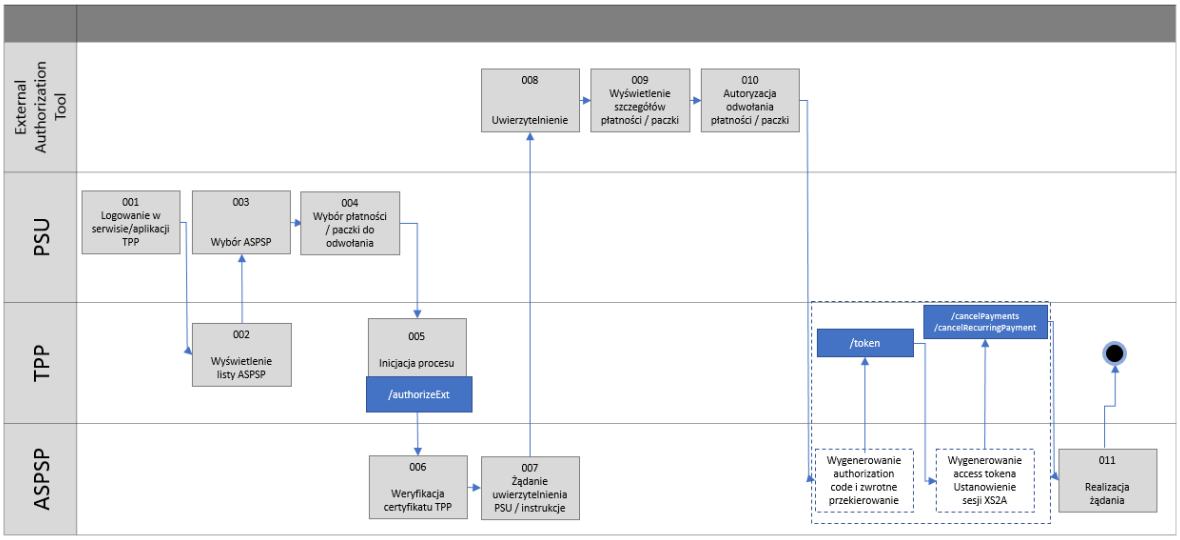
Ilustracja 8: PIS – odwołanie płatności – uwierzytelnianie po stronie ASPSP

4.1.5 Odwołanie płatności (płatność pojedyncza z datą przyszłą, płatność cykliczna z datą przyszłą, pojedyncza płatność w ramach płatności wielokrotnej (z datą przyszłą) lub płatność wielokrotna – paczka przelewów) – uwierzytelnianie w zewnętrznym narzędziu autoryzacyjnym

- 001 / PSU inicjuje proces w interfejsie TPP
- 002 / TPP wyświetla listę ASPSP
- 003 / PSU wybiera z listy ASPSP
- 004 / PSU wybiera płatność, definicję płatności cyklicznej lubpaczkę do odwołania
- 005 / TPP inicjuje proces odwołania (użycie metody /authorizeExt, w tym przekazanie scope i scope_details)
- 006 / ASPSP weryfikuje tożsamość TPP na podstawie certyfikatu (lub także na podstawie rejestru TPP)

- 007 / ASPSP inicjuje proces uwierzytelnienia PSU, w tym przekazuje do EAT instrukcję dotyczącą użycia 2 elementu autoryzacyjnego
- 008 / Uwierzytelnienie
- 009 / EAT wyświetla PSU szczegóły odwoływanej płatności
- 010 / PSU autoryzuje odwołanie płatności wg metody stosowanej w relacjach z ASPSP. Po zakończonej sukcesem autoryzacji odwołania, na linii ASPSP-TPP dochodzi do ustanowienia sesji interfejsu XS2A, czego efektem jest przekazanie przez ASPSP do TPP access tokena (użycie metody /token oraz, po ustanowieniu sesji XS2A metody / cancelPayments lub /cancelRecurringPayment
- 011 / ASPSP realizuje żądanie

proces odwołania płatności / paczki kończy się



Ilustracja 9: PIS – odwołanie płatności – uwierzytelnianie w zewnętrznym narzędziu autoryzacyjnym

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

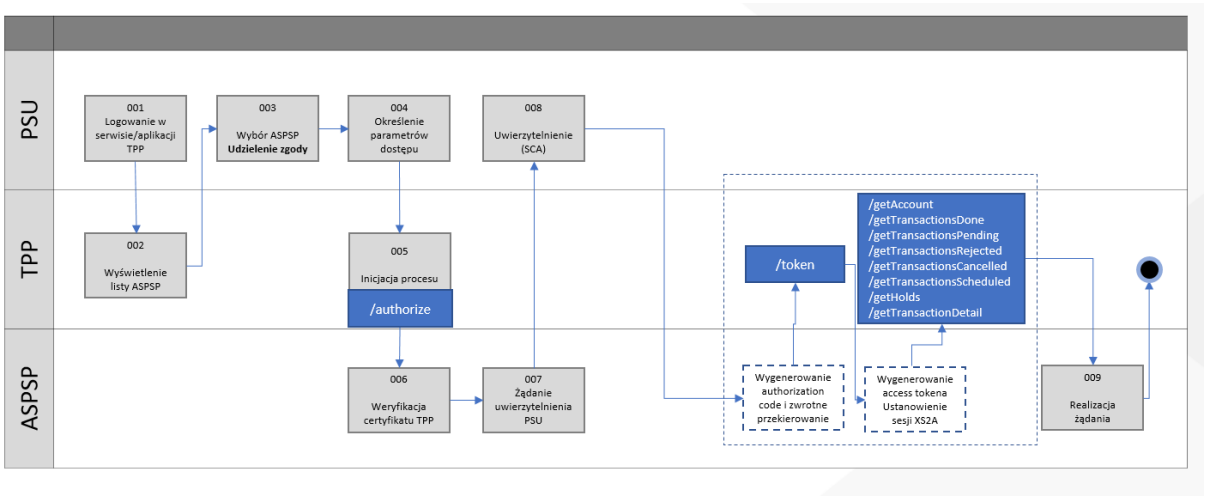
Wykorzystanie usługi AIS w zakresie Zgodności zaprezentowane w tym Przypadku Użycia polega na pozyskaniu przez TPP występującego w roli AISP informacji na temat rachunku płatniczego PSU, prowadzonego przez ASPSP, w oparciu o stosowne zapisy UUP.

4.2.1 Udzielenie zgody oraz pobranie informacji o rachunku z ręcznym wprowadzeniem numeru rachunku – uwierzytelnienie 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 inicjuje proces AIS (użycie metody /authorize, w tym przekazanie scope i scope_details), 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. Po zakończonym sukcesem uwierzytelnieniu przez PSU, na linii ASPSP-TPP dochodzi do ustanowienia sesji interfejsu XS2A, czego efektem jest przekazanie przez ASPSP do TPP access tokena oraz refresh tokena (użycie metody /token oraz, po ustanowieniu sesji XS2A jednej z metod: /getAccount, /getTransactionsDone, /getTransactionsPending, /getTransactionsRejected, /getTransactionsCancelled, /getTransactionsScheduled, /getHolds, /getTransactionDetail)
- 009 / ASPSP realizuje żądanie, następuje przekierowanie do domeny TPP

proces pobrania informacji o rachunku kończy się



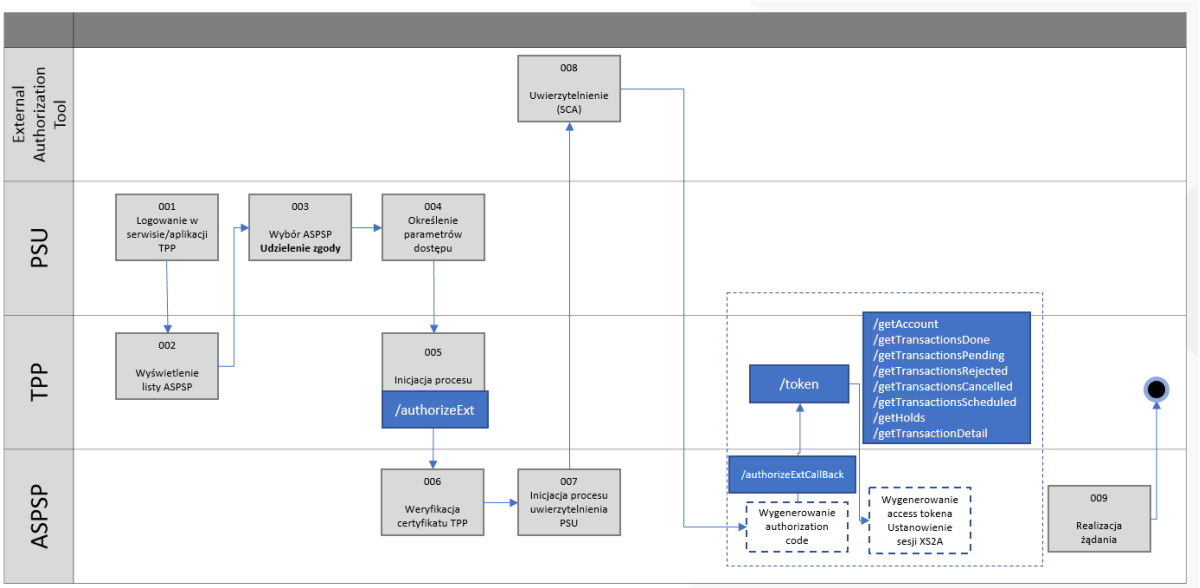
Ilustracja 10: AIS – ręczne wprowadzenie nr rachunku – uwierzytelnianie po stronie ASPSP

4.2.2 Udzielenie zgody oraz pobranie informacji o rachunku z ręcznym wprowadzeniem numeru rachunku – uwierzytelnienie 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 inicjuje proces AIS (użycie metody /authorizeExt, w tym przekazanie scope i scope_details)

- 006 / ASPSP weryfikuje tożsamość TPP na podstawie certyfikatu (lub także na podstawie rejestru TPP)
- 007 / ASPSP inicjuje proces uwierzytelnienia PSU
- 008 / Uwierzytelnienie SCA. Po zakończonym sukcesem uwierzytelnieniu przez PSU, na linii ASPSP-TPP dochodzi do ustanowienia sesji interfejsu XS2A, czego efektem jest przekazanie przez ASPSP do TPP access tokena oraz refresh tokena (użycie metody /token oraz, po ustanowieniu sesji XS2A jednej z metod: /getAccount, /getTransactionsDone, /getTransactionsPending, /getTransactionsRejected, /getTransactionsCancelled, /getTransactionsScheduled, /getHolds, /getTransactionDetail)
- 009 / ASPSP realizuje żądanie

proces pobrania informacji o rachunku kończy się



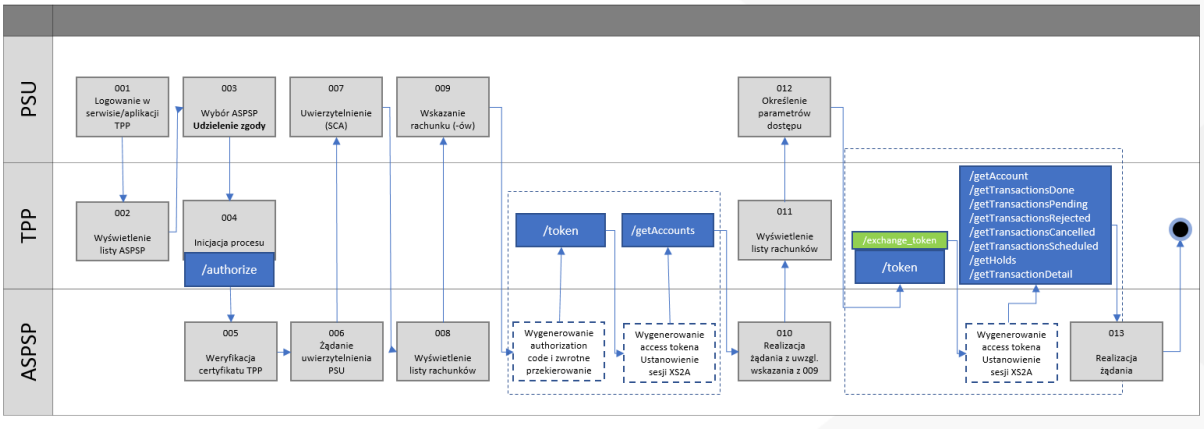
Ilustracja 11: AIS – ręczne wprowadzenie nr rachunku – uwierzytelnianie w zewnętrznym narzędziu autoryzacyjnym

4.2.3 Udzielenie zgody oraz pobranie informacji o rachunku z wyborem rachunku po stronie ASPSP – uwierzytelnienie 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 inicjuje proces AIS (użycie metody /authorize, w tym przekazanie scope i scope_details)
- 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 wyświetla listę rachunków
- 009 / PSU wskazuje rachunek (rachunki), w kontekście których chce uzyskać informacje. Po zakończonym sukcesem uwierzytelnieniu oraz wskazaniu rachunków przez PSU, na linii ASPSP-TPP dochodzi do ustanowienia sesji interfejsu XS2A, czego efektem jest przekazanie przez ASPSP do TPP access tokena oraz refresh tokena (użycie metody /token oraz, po ustanowieniu sesji XS2A metody /getAccounts)
- 010 / ASPSP realizuje żądanie z uwzględnieniem wskazania z kroku 009
- 011 / TPP wyświetla listę wskazanych rachunków
- 012 / PSU określa parametry dostępu dla wskazanych rachunków, TPP wywołuje metodę /token z użyciem metody exchange_token, czego efektem jest przekazanie przez ASPSP do TPP nowych access tokena oraz refresh tokena, a następnie wywołanie jednej z metod: : /getAccount, /getTransactionsDone, /getTransactionsPending, /getTransactionsRejected, /getTransactionsCancelled, /getTransactionsScheduled, /getHolds, /getTransactionDetail
- 013 / ASPSP realizuje żądanie

proces pobrania informacji o rachunku kończy się



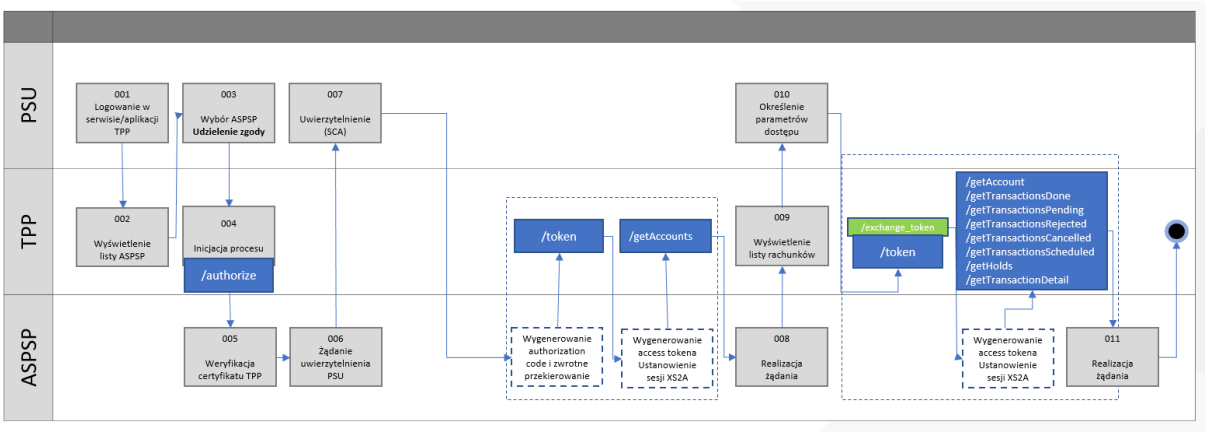
Ilustracja 12: AIS – wybór rachunku po stronie ASPSP – uwierzytelnianie po stronie ASPSP

4.2.4 Udzielenie zgody oraz pobranie informacji o rachunku z pobraniem listy rachunków – uwierzytelnienie 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 inicjuje proces AIS (użycie metody /authorize, w tym przekazanie scope i scope_details)
- 005 / ASPSP weryfikuje tożsamość TPP na podstawie certyfikatu (lub także na podstawie rejestru TPP)
- 006 / ASPSP przesyła żądanie uwierzytelnia PSU

- 007 / Uwierzytelnienie SCA. Po zakończonym sukcesem uwierzytelnieniu oraz wskazaniu rachunków przez PSU, na linii ASPSP-TPP dochodzi do ustanowienia sesji interfejsu XS2A, czego efektem jest przekazanie przez ASPSP do TPP access tokena oraz refresh tokena (użycie metody /token oraz, po ustanowieniu sesji XS2A metody /getAccounts)
- 008 / ASPSP realizuje żądanie
- 009 / TPP wyświetla listę rachunków
- 010 / PSU określa parametry dostępu dla wskazanych rachunków, TPP wywołuje metodę /token z użyciem metody exchange_token, czego efektem jest przekazanie przez ASPSP do TPP nowych access tokena oraz refresh tokena, a następnie wywołanie jednej z metod: /getAccount, /getTransactionsDone, /getTransactionsPending, /getTransactionsRejected, /getTransactionsCancelled, /getTransactionsScheduled, /getHolds, /getTransactionDetail
- 011 / ASPSP realizuje żądanie

proces pobrania informacji o rachunku kończy się



Ilustracja 13: AIS – pobranie listy rachunków – uwierzytelnianie po stronie ASPSP

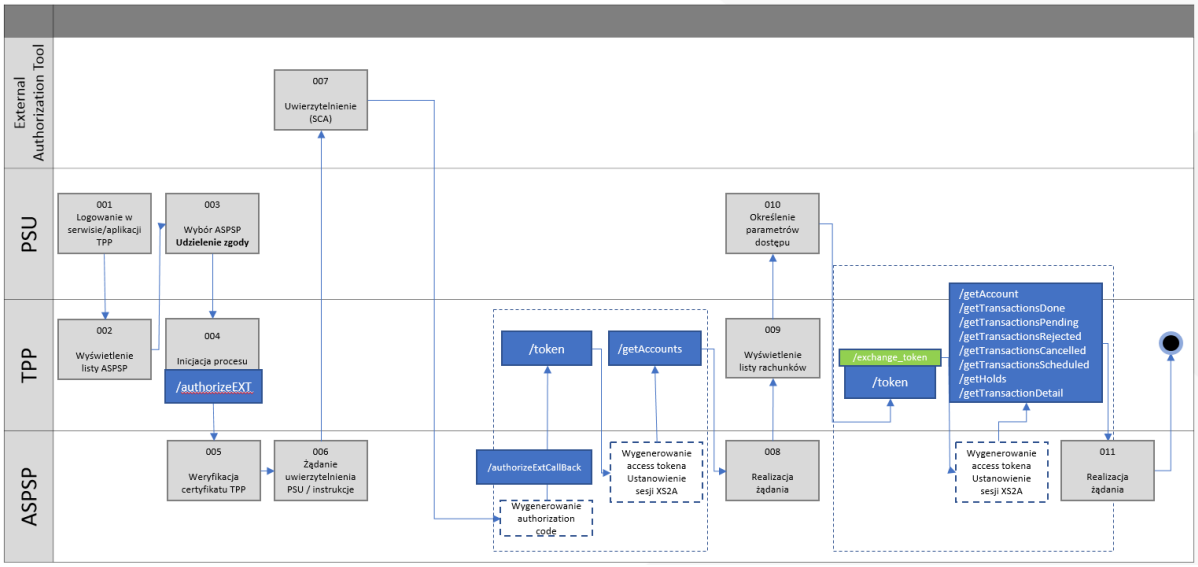
4.2.5 Udzielenie zgody oraz pobranie informacji o rachunku z pobraniem listy rachunków – uwierzytelnienie w zewnętrznym narzędziu autoryzacyjnym

- 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 inicjuje proces AIS (użycie metody /authorizeExt, w tym przekazanie scope i scope_details)
- 005 / ASPSP weryfikuje tożsamość TPP na podstawie certyfikatu (lub także na podstawie rejestru TPP)
- 006 / ASPSP inicjuje proces uwierzytelnienia PSU
- 007 / Uwierzytelnienie SCA. Po zakończonym sukcesem uwierzytelnieniu oraz wskazaniu rachunków przez PSU, na linii ASPSP-TPP dochodzi do ustanowienia sesji interfejsu XS2A, czego

efektem jest przekazanie przez ASPSP do TPP access tokena oraz refresh tokena (użycie metody /token oraz, po ustanowieniu sesji XS2A metody /getAccounts)

- 008 / ASPSP realizuje żądanie
- 009 / TPP wyświetla listę rachunków
- 010 / PSU określa parametry dostępu dla wskazanych rachunków, TPP wywołuje metodę /token z użyciem metody exchange_token, czego efektem jest przekazanie przez ASPSP do TPP nowych access tokena oraz refresh tokena, a następnie wywołanie jednej z metod: /getAccount, /getTransactionsDone, /getTransactionsPending, /getTransactionsRejected, /getTransactionsCancelled, /getTransactionsScheduled, /getHolds, /getTransactionDetail
- 011 / ASPSP realizuje żądanie

proces pobrania informacji o rachunku kończy się

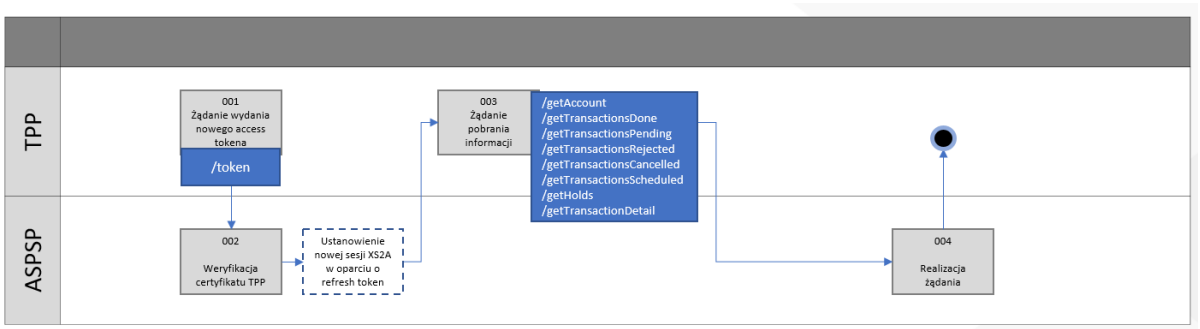


Ilustracja 14: AIS – pobranie listy rachunków – uwierzytelnianie w zewnętrznym narzędziu autoryzacyjnym

4.2.6 Pobranie informacji o rachunku bez udziału PSU

- 001 / TPP przekazuje żądanie wydania nowego tokena w oparciu o refresh token
- 002 / ASPSP weryfikuje tożsamość TPP na podstawie certyfikatu (lub także na podstawie rejestru TPP). Po pozytywnej weryfikacji na linii ASPSP-TPP dochodzi do ustanowienia sesji interfejsu XS2A, czego efektem jest przekazanie przez ASPSP do TPP access tokena w oparciu o refresh token
- 003 / TPP wywołuje jedną z metod: /getAccount, /getTransactionsDone, /getTransactionsPending, /getTransactionsRejected, /getTransactionsCancelled, /getTransactionsScheduled, /getHolds, /getTransactionDetail
- 004 / ASPSP realizuje żądanie

proces pobrania informacji o rachunku kończy się

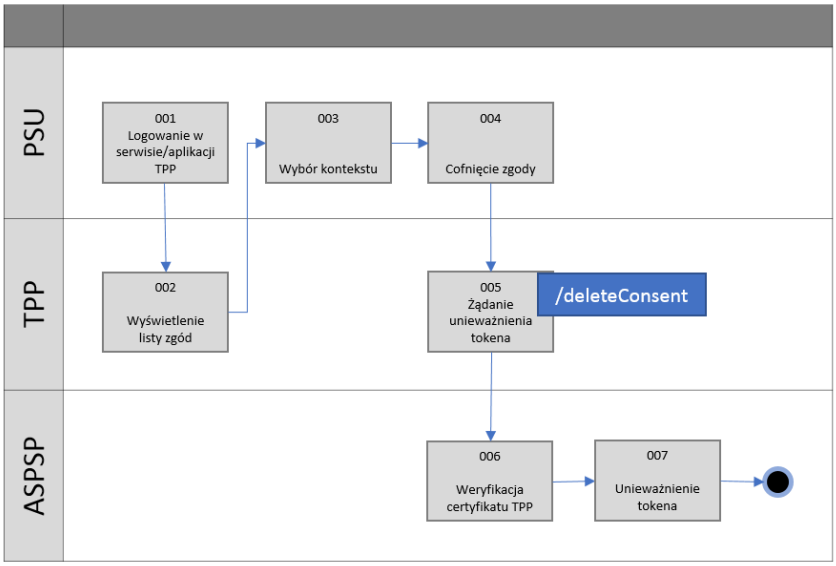


Ilustracja 15: AIS – pobranie informacji o rachunku bez udziału PSU

4.2.7 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 (użycie metody /deleteConsent)

proces cofnięcia zgody kończy się



Ilustracja 16: AIS – cofnięcie zgody

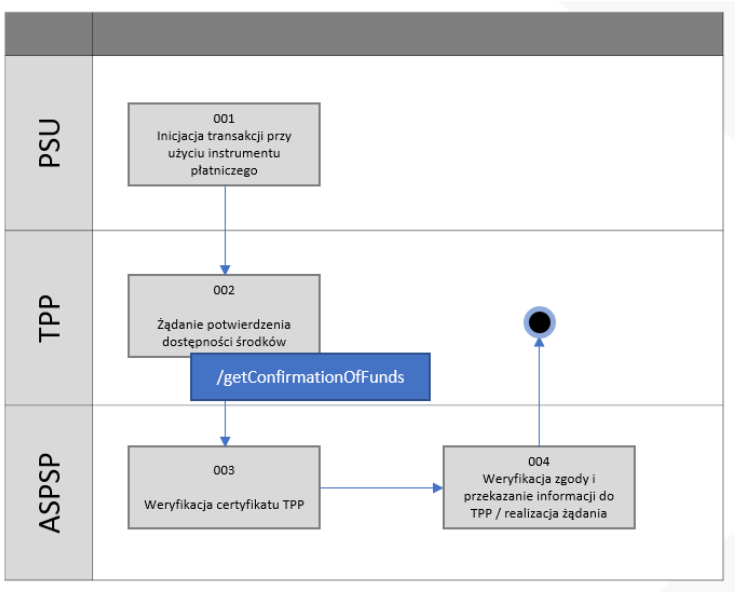
4.3 Przypadek użycia #3: zapytanie o dostępność środków przez PIISP (CAF)

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: CAF – zapytanie o dostępność środków

- 001 / Inicjacja transakcji przy użyciu instrumentu płatniczego
- 002 / TPP wywołuję metodę /getConfiramtionOfFunds
- 003 / ASPSP weryfikuje tożsamość TPP na podstawie certyfikatu (lub także na podstawie rejestru TPP).
- 004 / ASPSP weryfikuje zgodę na usługę po swojej stronie, po pozytywnej weryfikacji realizuje żądanie

proces zapytania o dostępność środków kończy się

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 usługi będzie 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: - sposób wyrażenia zgody na wykonywanie przez TPP operacji w imieniu klienta - zakres operacji i uprawnień - URL, pod jakim dana usługa jest dostępna - standardowy zakres parametrów per usługa - mechanizmy zabezpieczeń - zasady komunikacji - obsługę błędów PolishAPI nie specyfikuje - pełnego zakresu funkcjonalności, które mają być udostępnione przez ASPSP, oraz które z nich będą w Usługach Zgodności - 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.

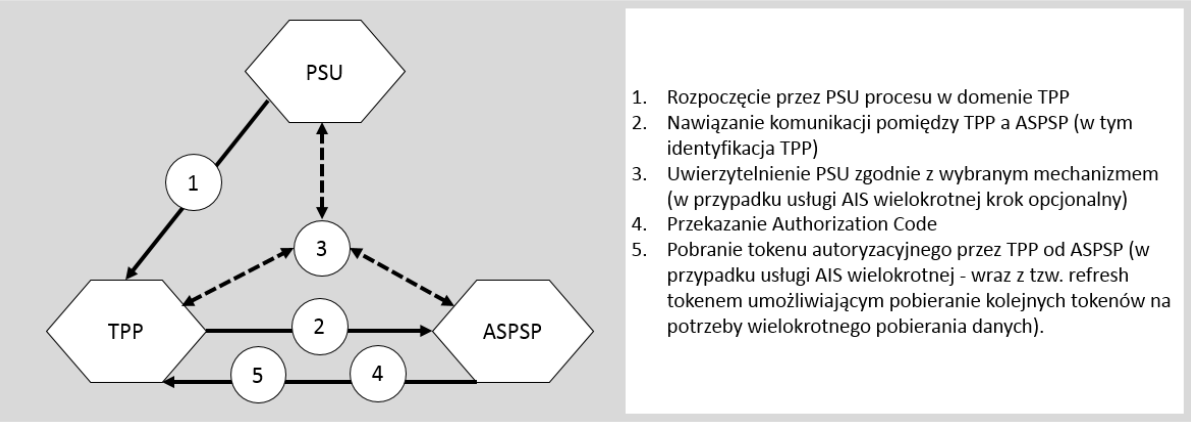
Proces nawiązywania sesji komunikacyjnej z interfejsem XS2A obejmuje żądania i odpowiedzi przekazywane pomiędzy TPP i ASPSP przy użyciu usług technicznych tego interfejsu (AS – Authorization Service), którego efektem jest ustanowienie sesji komunikacyjnej po stronie ASPSP i przekazanie do TPP jej technicznej 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 mieć miejsce 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.

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 PolishAPI.

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 rozdzielana znakiem „_") 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 Wersjonowanie

a) Zasada wersjonowania interfejsu API, specyfikacji PolishAPI, bazuje na konwencji nazewniczej *Semantic Versioning v2.0.0* (https://semver.org/). Zakłada ona wykorzystanie oznaczenia wersji przy użyciu trzech liczb oddzielonych separatorem: `X_Y_Z`.
b) Każda z kolejnych części rozdzielonych separatorem jest liczbą naturalną (na przykład 1_12_1). Pierwszy segment oznacza tzw. wersję `major`, środkowy `minor`, a ostatni `patch`.
c) Wersja `major` jest używana do określania zmian niekompatybilnych wstecznie lub przełomowych względem opublikowanej aktualnie, obowiązującej wersji interfejsu API.
d) Wersja `minor` jest używana do określenia kolejnych przyrostów funkcjonalności interfejsu API. Zgodnie z konwencją nazewniczą funkcjonalności w interfejsie API, dla danej jego wersji, powinny wyłącznie przyrastać, chyba że mamy do czynienia z jasno określoną zmianą dotyczącą wyprowadzania funkcjonalności z użycia (ang. deprecation). Wprowadzone zmiany w wersji `minor` nie powinny powodować niekompatybilności.
e) Wersja `patch` jest przeznaczona do użytku wyłącznie dla poprawek bezpieczeństwa oraz funkcjonalności, nie powodujących niekompatybilności wstecznej ani przyrostów funkcjonalności.
f) Przyjmuje się zasadę nie publikowania jednoczesnych zmian rozszerzenia funkcjonalności (`minor`) i poprawy błędów (`patch`).
g) Zmiany typu `minor` będą wprowadzane nie częściej niż 2 razy w ciągu roku.
h) W przypadku zmiany typu `major`, aby zapewnić zgodność ze standardem, należy wdrożyć ją w okresie maksymalnie 6 miesięcy od daty publikacji.
i) W przypadku zmiany `minor`, aby zapewnić zgodność ze standardem, należy wdrożyć ją w okresie maksymalnie 12 miesięcy od daty publikacji lub w zależności od daty publikacji kolejnej wersji `major`.

j) W przypadku zmiany typu `major`, o której mowa w pkt h) powyżej oraz zmiany `minor`, o której mowa w pkt i) powyżej, wskazany maksymalny okres wprowadzenia zmian dotyczy zmian w środowisku testowym (*sandbox*). Zmiany w środowisku produkcyjnym mogą zostać wprowadzone w terminie późniejszym po przetestowaniu interfejsu przez TPP, w tym w szczególności z zachowaniem 3-miesięcznego okresu, o którym mowa w art. 30 ust. 4 RTS.

5.8 Kanoniczny model danych

Szczegółowy opis struktur danych jest dostępny w dokumentacji swagger, w zakładce „Models".

5.9 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.

5.10 Sortowanie

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

5.11 Filtrowanie

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

a) itemIdFrom – transakcje lub blokady od podanego identyfikatora „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) transactionCategory – CREDIT lub DEBIT
g) minAmount – minimalna kwota operacji w żądanym zakresie danych
h) maxAmount – maksymalna kwota operacji w żądanym zakresie danych

5.12 Stronicowanie

Wyniki zapytań zawierające wiele rekordów (więcej niż 100) powinny być stronicowane. Kolejne strony będą pobierane poprzez ustawianie atrybutu pageId w strukturach danych odpowiedzialnych za przesyłanie żądań o pobranie listy rachunków oraz listy transakcji.

W odpowiedzi na żądania pobrania rachunków lub transakcji będzie zwracana struktura o nazwie PageInfo, zawierająca opcjonalne parametry o nazwach nextPage oraz previousPage. Parametry te, o ile zostaną zwrócone, zawierają identyfikatory odpowiednio poprzedniej i następnej strony w stosunku do strony zwróconej w żądaniu, a poza tym znaczeniem nie podlegają interpretacji przez aplikację kliencką (są dowolnym dozwolonym napisem, interpretowalnym przez stronę dostawcy API w sposób pozwalający na identyfikację zawartości strony, w szczególności mogą być zakodowanymi wewnętrznymi identyfikatorami rekordów z jednego lub więcej systemów lub reprezentacją tzw. kursora).

Aby pobrać określoną stronę należy przekazać, w kolejnym żądaniu, w parametrze pageId, jedną z wartości otrzymanych w poprzednim żądaniu, we wspomnianych atrybutach nextPage lub previousPage. Atrybut pageId zawierać może wyłącznie wartość literalnie równą wartości zwróconej w parametrze nextPage lub previousPage poprzedniego żądania, i nie podlega interpretacji przez aplikację kliencką (w szczególności nie jest tożsamy z numerem kolejnym, o ile wystawca API nie określi możliwości takiej interpretacji).

Liczba rekordów na stronie definiowana jest za pomocą atrybutu perPage wysyłanych żądań.

W przypadku wykorzystania parametru pageId zawierającego wartość zwróconą poprzednio w nextPage lub previousPage żadana liczba rekordów na stronie powinna mieć tą wartość co użyta w żądaniu, z którego pochodzi użyty identyfikator strony (dodatkowo zmianie nie powinny podlegać parametry określające filtrowanie, np.: itemIdFrom, transactionDateFrom, transactionDateTo, transactionCategory itd.). API ma prawo zwrócić błąd stronicowania w przypadku zmiany żądanej wielkości strony lub parametrów stronicowania pomiędzy żądaniami, jeśli żądanie wykorzystuje parametr pageId.

Pusta wartość parametru pageId, lub jego pominięcie oznacza żądanie zwrócenia pierwszej strony.

5.13 Statusy odpowiedzi

Statusy techniczne będą zwracane poprzez następujące kody http:

STATUS	OPIS
200 OK	Operacja się powiodła
204 No Content	Operacja się powiodła. Odpowiedź nie zawiera dodatkowych informacji.
400 Bad Request	Zapytanie jest niepoprawne syntaktycznie
401 Unauthorized	Niepoprawnie uwierzytelniony użytkownik
403 Forbidden	Błąd autoryzacji (brak uprawnień dostępu do 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
500 Internal Server Error	Wystąpił nieznany, wewnętrzny błąd serwera API
501 Not Implemented	Interfejs XS2A po stronie ASPSP nie wspiera użytej przez TPP funkcjonalności
503 Service Unavailable	Serwer API jest tymczasowo niedostępny

5.14 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 cache'owania 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)

5.15 Format wiadomości

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

5.16 Podstawowe formaty danych

FORMAT	FORMAT JSON	OPIS
Tekst	String	Tekst kodowany w UTF-8
Daty	String	Zgodnie z ISO8601. Data i czas będą reprezentowane w postaci YYYY-MM-DD na YYYY-MM-DDThh:mm:ss.ccczzzzzz z obowiązkowym podaniem strefy czasowej Oznaczenia: 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 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

5.17 Unikalny identyfikator żądania i algorytm jego generowania

Każde żądanie wysyłane przez TPP do interfejsu XS2A po stronie ASPSP musi zawierać unikalny identyfikator (parametr o nazwie requestId w strukturze nagłówkowej przekazywanej w ciele każdego żądania). Unikalność tego identyfikatora musi być zachowana w skali wszystkich żądań wysyłanych przez wszystkie TPP do wybranego ASPSP.

Wymóg przekazywania unikalnego identyfikatora żądania wynika z konieczności przeprowadzania weryfikacji wszystkich otrzymanych przez ASPSP żądań, w celu identyfikacji i odrzucania tych samych żądań otrzymanych wielokrotnie np. w wyniku błędu po stronie TPP lub powtórnego, intencjonalnego wysłania komunikatu przez TPP w przypadku braku odpowiedzi ze strony ASPSP.

Standard PolishAPI definiuje wymagany format identyfikatora żądania, który zapewnia tak zdefiniowaną unikalność oraz pozwala dokonywać opisanej weryfikacji żądań po stronie ASPSP w sposób wybiórczy tzn. z uwzględnieniem niewielkiego podzbioru otrzymanych wcześniej identyfikatorów żądań, co znacząco pozytywnie wpływa na wydajność takiej weryfikacji i pośrednio na zapewnienie szybszej odpowiedzi interfejsu XS2A.

Wymaganym formatem identyfikatora żądania jest UUID (*ang.* Universally Unique Identifier), który jest standardem opisanym w dokumencie RFC 4122 (https://tools.ietf.org/html/rfc4122). Dodatkowo wymaga się aby identyfikator żądania był generowany w wariancie numer 1 (patrz punkt 4.1.1 RFC 4122) oraz wersji nr 1 (patrz punkt 4.1.3 RFC 4122), co zapewnia uwzględnienie w wartości identyfikatora składnika monotonicznego opartego o czas wysłania żądania oraz informację identyfikującą podmiot wysyłający żądanie (TPP).

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 i PIS 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. 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.

Błędy walidacji danych wejściowych muszą być rejestrowane w logach.

Błędy walidacji muszą być sygnalizowane komunikatem HTTP 400 (Bad Request) i dane muszą być odrzucane. Dotyczy to również negatywnej walidacji podpisu JWS-SIGNATURE.

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

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, dla zapewniania niezaprzeczalności, należy zastosować podpis JSON Web Signature, zgodnie ze standardem RFC 7515 (https://tools.ietf.org/html/rfc7515). Sygnatura podpisu musi być umieszczana w każdym żądaniu w nagłówku o nazwie 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.

6.6.1 Rejestracja aplikacji klienckich TPP po stronie ASPSP

Specyfikacja PolishAPI, wychodząc naprzeciw wymaganiom dyrektywy PSD2, definiuje dedykowaną metodę usługi AS (ang. *Authorization Services*) o nazwie **/register**, która umożliwia przeprowadzenie automatycznej rejestracji aplikacji klienckich TPP, które mają uzyskać dostęp do interfejsu XS2A.

Rejestracja aplikacji klienckiej jest wykorzystuje standard RFC 7591 (https://tools.ietf.org/html/rfc7591), rozszerzony o dodatkowe informacje, i dzięki temu pozwalaja na zapewnienie:

- podwyższonego bezpieczeństwa wynikającego z braku nieograniczonej dostępności kluczy publicznych (znaczące ograniczenie możliwości ataków brute force oraz ataków w oparciu o błędy implementacji procesu weryfikacji JWT/JWS, takie jak użycie klucza publicznego jako klucza symetrycznego w wyniku wstrzyknięcia wartości "alg" w JWT)
- możliwości użycia osobnych kluczy per TPP i per aplikacja (wymiana kluczy publicznych), zapewniające podwyższoną rozłączalność działań
- możliwości użycia przez TPP więcej niż jednej, osobnej, aplikacji klienckiej, lub działania aplikacji w różnych trybach, z wymuszeniem ograniczenia uprawnień per aplikacja (a nie per TPP) – np. jeden TPP może mieć wydzieloną aplikację do inicjacji płatności, różną niż stworzona przez tego samego TPP aplikacja prezentująca stan finansów klienta – i aplikacje te nie będą mogły wykonać działań spoza ich zakresu uprawnień, mimo iż mieszczą się one w uprawnieniach TPP.
- przekazania danych dodatkowych pozwalających na spójne i zgodne z intencją TPP prezentowanie klientowi banku informacji o TPP (nazwa, logo, opis przedmiotu działania, odnośniki do warunków świadczenia usług etc.) oraz jego aplikacji (nazwa, opis przeznaczenia aplikacji, logo, URL do pobrania itp.)
- ograniczenia podatności związanych z przekierowaniem na arbitralne adresy, tzw. open redirector

Opisywany mechanizm przewiduje możliwość przekazania pomiędzy TPP a ASPSP:

a) Nazwy, opisu i innych informacji (np. logo) służących przedstawieniu danej aplikacji użytkownikowi (w kontekście otwartych API bankowych tworzonych na potrzeby dyrektywy PSD2 - użytkownikiem jest klient banku - PSU)

b) danych identyfikacji aplikacji w wywołaniach API, w tym API autoryzacji, ze szczególnym uwzględnieniem wydania unikalnego identyfikatora aplikacji client_id (a nie tylko - organizacji - TPP - co pozwala na posiadanie przez jednego TPP wielu aplikacji klienkich o różniących się uprawnieniach lub działających w różnych "trybach"), przy czym wsparcie dla wielu aplikacji per TPP pozostaje decyzją ASPSP (dokument zakłada, że ASPSP może ograniczyć możliwość rejestracji do pojedynczej aplikacji klienckiej per TPP)

c) materiału kryptograficznego (kluczy i certyfikatów) do użycia w późniejszej komunikacji, przy czym protokół wspiera zarówno możliwość wymiany kluczy współdzielonych symetrycznych) jak i przekazania kluczy publicznych służących weryfikacji podpisów i szyfrowaniu asymetrycznemu, i przeprowadzenia dowodu kryptograficznego posiadania kluczy prywatnych im odpowiadających.

d) Przekazania certyfikatów klienta służących potwierdzeniu tożsamości wystawcy aplikacji (wraz z przeprowadzeniem kryptograficznego dowodu tożsamości poprzez dowód posiadania klucza prywatnego odpowiadającego zaufanemu certyfikatowi), a co za tym idzie jego uprawnień wynikających z umocowania regulacyjnego (certyfikaty eIDAS zgodne z ETSI TS 119 495

e) Ustalenia adresów przekierowań z usług autoryzacji OAuth2

Przebieg samodzielnej rejestracji aplikacji klienckiej przez TPP

Wsparcie dla samodzielnej rejestracji aplikacji klienckiej (dalej: samorejestracja TPP) przez TPP w oparciu o certyfikat pieczęci jest wymagane dla scenariuszy regulacyjnych PSD2.

W scenariuszach regulacyjnych PSD2 warunkiem wymaganym i wystarczającym dla rejestracji jest przekazanie certyfikatu eIDAS zgodnego z ETSI TS 119 495, wskazującego na dane umocowanie regulacyjne (identyfikator TPP nadany przez regulatora, role AISP / PISP / ASPSP etc.) oraz dowiedzenie kryptograficznie jego własności (dowód posiadania klucza prywatnego).

Metodą dowiedzenia posiadania klucza prywatnego pieczęci jest podpisanie przez TPP tokena *software_statement* tym kluczem, przy czym certyfikat z nim korespondujący musi być na liście *jwks*.

Zawartość żądania i odpowiedzi metody /register

Żądanie rejestracji wysyłane jest szyfrowanym kanałem (użycie TLS 1.2 lub nowszego standardu), z użyciem metody HTTP POST, na adres /register. Zawartość żądania stanowi dokument JSON (Content-type: application/json), zgodny z wymaganiami punktu 3.1.1. RFC 7591 - "Client Registration Request Using a Software Statement". Zakłada się, że wszystkie fakty ujęte w żądaniu powinny zostać przekazane jako treść podpisanego tokenu JWT stanowiącego wartość pola *software_statement*.

ASPSP może uwzględnić fakty przekazane jako pola poza tokenem *software_statement* (w polach głównego dokumentu JSON nie będących wewnątrz treści *software_statement*), o ile nie stoją one w sprzeczności z treścią *software_statement*.

ASPSP nie może odrzucić żądania zawierającego wszystkie dane wymagane w obrębie *software_statement* ze względu na wartość przekazaną poza tym tokenem.

Pola żądania rejestracji zawarte w tokenie software_statement

Identyfikator faktu	Znaczenie	Wymagany	Uwagi
iat	Moment wystawienia software_statement	Nie	W celu zapobieżenia ponownemu użyciu tego samego żądania, z prawem odrzucenia przez ASPSP żądań po upływie określonego czasu
aud	Zakładany odbiorca danego żądania	Nie / Warunkowo do decyzji ASPSP	ASPSP ma prawo wprowadzenia wymogu wystawienia *software_statement* wymieniającego danego ASPSP jako odbiorcę – element wzmacniający bezpieczeństwo poprzez uniemożliwienie użycia tego samego żądania z innym dostawcą, oraz zapewniający lepszą rozłączalność. Przykładowa wartość – identyfikator ASPSP.

iss	Identyfikator wystawcy *software_statement*	Tak	Jest identyfikatorem TPP wystawionym przez regulatora i zakodowany zgodnie z wymogami ETSI TS 119 495.
iss_name	Nazwa wystawcy *software_statement*	Nie	
sub	Identyfikator organizacji żądającej dostępu dla swojej aplikacji. Dla scenariuszy regulacyjnych PSD2 – identyfikator TPP nadany przez regulatora	Nie w scenariuszach regulacyjnych, w których może zostać odczytany z certyfikatu	O ile podany, musi być tożsamy z identyfikacją organizacji, której dotyczy certyfikat
sub_name	Nazwa (pełna) organizacji żądającej dostępu dla swojej aplikacji – pełna nazwa TPP	Nie	O ile podany nie może pozostawać w sprzeczności (być różny) od nazwy podanej w certyfikacie przedstawionym przez organizację
sub_descr	Opis organizacji żądającej dostępu dla swojej aplikacji	Nie	
sub_logo	URL logo organizacji żądającej dostępu dla swojej aplikacji	Nie	
Sub_contact_name	Imię i nazwisko osoby kontaktowej	Nie	
Sub_contact_email	Email do osoby kontaktowej	Nie	
Sub_org_number	NIP w formacie europejskim organizacji rejestrującej	Nie	
Sub_country	Kraj rejestracji organizacji	Nie	
client_name	Nazwa aplikacji klienckiej	Nie	W razie nie podania, przyjmuje się nazwę TPP (z pola *sub_name* lub z certyfikatu)
response_types	Zgodnie z RFC 7591	Nie	Wartość stała: “code”
grant_types	Zgodnie z RFC 7591, rozszerzone o wartość „exchange_token”	Nie	Może przyjmować jedną z wartości: • authorization_code • refresh_token • exchange_token
redirect_uris	Adresy URL na które dopuszczalne jest przekierowanie klienta po zakończeniu wywołania metody /authorize	Tak	Zakłada się, że niedopuszczalne jest przekierowanie na inny URL niż wskazany w tym parametrze.
response_types		Nie	Wartość stała: “code”

jwks	Kolekcja kluczy (certyfikatów - dla kryptografii asymetrycznej), które mogą zostać użyte przez TPP dla podpisu żądań (i opcjonalnie nawiązania szyfrowanej komunikacji TLS). Zgodnie RFC 7591 i RFC 7517	Warunkowo TAK (o ile nie podano jwks_uri)	Musi zawierać certyfikat (lub miejsce pobrania certyfikatu) pieczęci zgodny z ETSI TS 119 495, który może zostać wykorzystany do podpisu żądań wysyłanych do interfejsu XS2A. Bank może opcjonalnie wspierać obsługę wielu certyfikatów per TPP/ aplikacja, w takiej sytuacji może zawierać więcej niż jeden certyfikat pieczęci. O ile TPP zamierza używać parametru "kid" dla identyfikacji klucza użytego do podpisu żądań API, wartość tego parametru musi być podana dla każdego klucza umieszczonego w jwks/jwks_uri (parametr kid). Jeśli nie podano parametru kid, zakłada się, że TPP użyje jednoznacznego identyfikatora klucza (fingerprint) w żądaniach, zalecane jest użycie x5t#256.
jwks_uri		Warunkowo TAK (o ile nie podano jwks)	
scope	Lista (separowana spacjami) nazw zakresów uprawnień. Zgodnie z RFC 7591 I RFC 6749.	Nie	Jedna lub więcej wartości z listy: - ais-accounts - ais - pis W przypadku nie podania przyjmowane jest na podstawie uprawnień na podstawie ról opisanych w certyfikacie TPP.

Przykład żądania:

```
POST /register HTTP/1.1
     Content-Type: application/json
     Accept: application/json
     Host: server.example.com
     {
       "software_statement": "
eyJhbGciOiJSUzI1NiIsInR5cCI6IkpXVCJ9.eyJqd2tzIjp7ImtleXMiOlt7Imt0eSI6IlBLSV
giLCJ4NWMiOlsiTUlJRTNqQ0NBOGFnQXdJQkFnSUNBd0V3RFFZSktvWklodmNOQVFFRkJRQXdZe
kVMTUFrR0ExVUVCaE1DVlZNVlp3PT0iLCJBQTRHQkFEdC9VRzl2VUpTWlNXSTRPQjlMK0tYSVBx
ZUNnZllyeCtqRnoiXSwidXNlIjoic2lnbiIsImtpZCI6ImtpZF9naXZlbl9ieV90cHAifV19LCJ
yZWRpcmVjdF91cmlzIjpbImh0dHBzOi8vdHBwLmRvbWFpbi5leGFtcGxlL29hdXRoMi1jYWxsYm
FjayJdfQ.J5ZWE6kKuIN2YxTUk14yyXWT4Ka72jQiXsx6BzxZrRM"
```

```
    }
```

Przykład parametru software_statement:

```
{
  "redirect_uris": [
    "https://client.example.org/callback",
    "https://client.example.org/callback2"
  ],
  "client_name": "My Money Planner",
  "logo_uri": "https://client.example.org/logo.png",
  "jwks_uri": "https://client.example.org/my_public_keys.jwks",
  "client_desc": "One app to manage all your personal finances!",
  "sub": "PL-PFSA-43124312414",
  "sub_name": "Money Planning Sp. z o.o.",
  "scope": "ais ais-accounts"
}
```

Odpowiedź pozytywna

Prawidłowa rejestracja aplikacji klienckiej sygnalizowana jest kodem odpowiedzi HTTP 201 Created. Zawartość treści odpowiedzi stanowi dokument JSON (Content-type: application/json), zawierający prezentowane poniżej pola, przy czym dopuszczalne jest udzielenie odpowiedzi zawierającej dowolne inne pola, z tym że zakłada się, że wartości spoza listy poniżej nie powinny być wymagane do konstrukcji prawidłowego żądania API (powinny mieć wartość czysto informacyjną).

Pola odpowiedzi

Identyfikator faktu	Znaczenie	Wymagany	Uwagi
client_id	Unikalny identyfikator aplikacji TPP	Nie	Jeśli ASPSP nie wspiera rejestracji więcej niż jednej aplikacji per TPP, możliwe jest pominięcie przez niego w odpowiedzi wartości client_id. Zaleca się, by client_id przybierał wartość równą identyfikatorowi TPP dla aplikacji domyślnej danego TPP (tzn. takiej dla której TPP nie podał specyficznej identyfikacji aplikacji poprzez wskazanie software_id). Pozwala to też na użycie takiej wartości konwencyjnie gdy

			ASPSP nie wspiera rejestracji wielu aplikacji. Jeśli ASPSP zwraca wartość client_id, TPP winien podać ta wartość w żądaniach API, które to umożliwiają (przede wszystkim metody /authorize i /token)
client_secret	Wartość tajna służąca uwierzytelnieniu aplikacji klienckiej względem serwera autoryzacji	Nie	W zależności od sposobu autoryzacji wywołań metody /token, wartość może lub nie zostać podana. O ile serwer rejestracji zwraca wartość client_secret, powinna ona zostać podana na wejściu metody /token
api_key	Wartość tajna służąca identyfikacji aplikacji klienckiej względem API	Nie	O ile serwer rejestracji zwraca wartość api_key, powinna ona zostać podana na wejściu w wywołaniach XS2A wspierających jej podanie
jwks	Kolekcja kluczy (certyfikatów - dla kryptografii asymetrycznej), które mogą zostać użyte przez ASPSP dla podpisu odpowiedzi API. Zgodnie RFC 7591 i RFC 7517	Warunkowo TAK (o ile nie podano jwks_uri)	Musi zawierać certyfikat (lub miejsce pobrania certyfikatu) pieczęci zgodny z ETSI TS 119 495, który może zostać wykorzystany do podpisu odpowiedzi. ASPSP może opcjonalnie wspierać obsługę wielu certyfikatów per TPP/ aplikacja, w takiej sytuacji kolekcja może zawierać więcej niż jeden certyfikat pieczęci. O ile ASPSP zamierza używać parametru “kid” dla identyfikacji klucza użytego do podpisu odpowiedzi API, wartość tego parametru musi być podana dla każdego klucza umieszczonego w jwks/jwks_uri (parametr kid). Jeśli nie podano parametru kid, zakłada się, ze ASPSP użyje jednoznacznego identyfikatora klucza (fingerprint) w odpowiedziach, zalecane jest użycie x5t#256.
jwks_uri		Warunkowo TAK (o ile nie podano jwks)	

Przykład odpowiedzi:

```
HTTP/1.1 201 Created
     Content-Type: application/json
     Cache-Control: no-store
     Pragma: no-cache

{
  "jwks": {
    "keys": [
      {
        "kty": "PKIX",
        "x5c": [

"XcFvvE3jCCA8ZasDDAgICAwEwDQYJKoZIhvcNAQEFBQAwYzELMAkGA1UE[int) w
odpowiedz
          "xcAADt/UG9vUJSZSWI4OB9L+KX[YJKoZIhvcNAQEFBQA
        ],
        "use": "sign",
        "kid": "kid_given_by_bank"
      }
    ]
  },
  "client_id": "s6BhdRkqt3"
}
```

Odpowiedź negatywna

Odpowiedź błędna sygnalizowana jest kodem odpowiedzi innym niż HTTP 201 Created. W ogólnym wypadku błąd rejestracji sygnalizowany kodami:

- 401 - w przypadku braku możliwości potwierdzenia tożsamości aplikacji klienckiej lub jej wystawcy,

- 403 - w przypadku gdy żądanie dotyczy nadania zakresu uprawnień wykraczającego poza dozwolony, np. wynikający z umocowania regulacyjnego.

Pola odpowiedzi negatywnej – zgodne co do struktury z odpowiedziami negatywnymi na żądania do innych metod interfejsu XS2A

6.6.2 Zarządzanie certyfikatami do podpisu JWS-SIGNATURE

Standard PolishAPI wymaga podpisywania wszystkich żądań i odpowiedzi zgodnie ze standardem JWS-SIGNATURE, w ramach interfejsu XS2A (po stronie ASPSP) oraz interfejsu wywołań zwrotnych (po stronie TPP). Z tego faktu wynika konieczność zarządzania certyfikatami, zarówno w kontekście

podpisywania wysyłanych komunikatów, jak i ich uzgadniania oraz udostępniania drugiej stronie komunikacji. Taka konieczność jest symetryczna tzn. dotyczy jednakowo ASPSP oraz TPP.

Standard PolishAPI, mając na uwadze względy wydajnościowe komunikacji poprzez interfejsy XS2A oraz wywołań zwrotnych, a także potencjalne trudności w zarządzaniu infrastrukturą kryptograficzną, związane z możliwością wykorzystania nieograniczonej ilości certyfikatów, wprowadza następujący wymóg, będący rozszerzeniem standardu RFC 7515 w kontekście parametrów nagłówka podpisu JWS-SIGNATURE:

- użycie parametru nagłówka o nazwie „kid" jest wymagane (rozszerzenie punktu 4.1.4 RFC 7515)

- użycie parametru nagłówka o nazwie „x5t#S256" jest wymagane (rozszerzenie punktu 4.1.8 RFC 7515)

Łączne spełnienie obu wymienionych wymogów przy konstruowaniu każdego podpisu JWS-SIGANTURE pozwala na:

a) jednoznaczne zidentyfikowanie certyfikatu, po stronie odczytującej, odnalezienie go w wewnętrznej infrastrukturze kryptograficznej i użycie do odczytania treści podpisu

b) pominięcie konieczności uzgodnienia certyfikatu przy każdym wysłanym i odebranych komunikacie poprzez interfejsy XS2A oraz wywołań zwrotnych

Implikacją wprowadzenia opisanych wymogów jest konieczność jednorazowego, wcześniejszego lub równoległego (względem komunikatu podpisanego przy użyciu JWS-SIGNATURE), uzgodnienia certyfikatu pomiędzy stronami komunikacji. Ze względu na nieliczny charakter takiej operacji, w stosunku do liczby komunikatów podpisanych przy użyciu wybranego certyfikatu, a co za tym idzie niski koszt jej przeprowadzenia, standard PolishAPI nie narzuca w tym względzie żadnych wymogów ale określa jedynie rekomendowane implementacje, którymi są:

a) użycie parametru nagłówka podpisu JWS-SIGNATURE o nazwie „x5u" (punkt 4.1.5 RFC 7515); pozwala na przekazanie URL do zasobu będącego publicznym kluczem certyfikatu X.509, w tym samym komunikacie, w którym podpis JWS-SIGANTURE został zbudowany po raz pierwszy przy użyciu tego certyfikatu

b) zastosowanie procedury w oparciu o protokół „OAuth 2.0 Dynamic Client Registration" (RFC 7591), pozwalającej na wcześniejsze (w stosunku do faktycznej komunikacji przy użyciu interfejsu XS2A lub wywołań zwrotnych) uzgodnienie certyfikatu pomiędzy stronami – może do tego celu posłużyć metoda /register, opisywana w specyfikacji PolishAPI – patrz punkt 6.6.1.

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 zakres dostępowy (odpowiadający zgodzie udzielonej TPP przez PSU na świadczenie usług dostępnych przy użyciu interfejsu XS2A). Dostępne wartości parametru to (należy przekazać dokładnie jedną):

- ais-accounts – uprawnienie do pobrania listy rachunków PSU;
- ais – uprawnienie do pobierania informacji o jednym lub wielu wskazanych przez PSU rachunkach;
- pis – uprawnienie do zainicjowana pojedynczej płatności lub wielu płatności w postaci paczki przelewów oraz do pobierania informacji o statusie zainicjowanych transakcji oraz paczki przelewów;

Szczegółowy zakres oraz warunki usług, świadczonych przez TPP w oparciu o wyżej wymienione uprawnienia, zostały opisane w postaci zestawu metod interfejsu XS2A, w specyfikacji technicznej standardu Polish API (załącznik nr 1), z osobna dla każdej z usług AIS i PIS.

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.

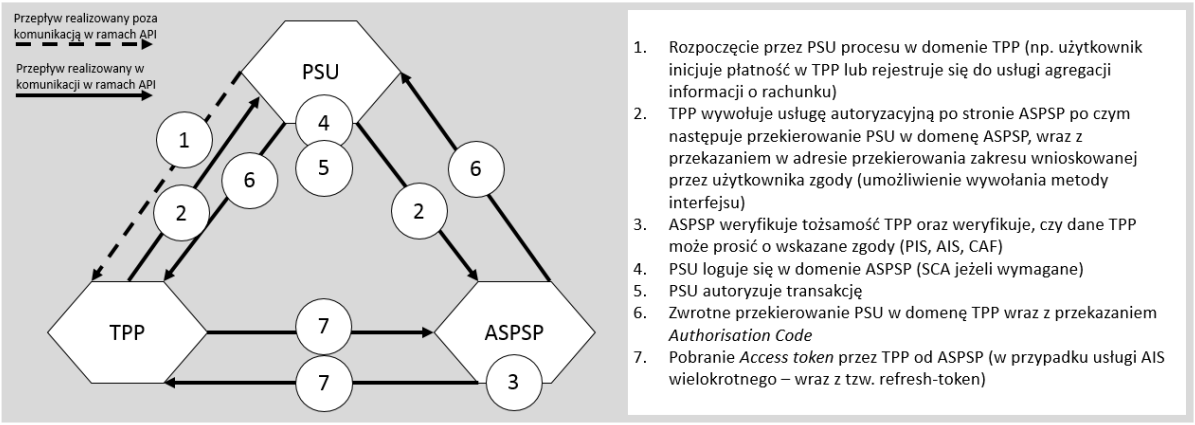
Powyższe parametry są przesyłane przez TPP jako POST (ze względu na możliwą wielkość scope_details) w formacie JSON zakodowane i podpisane przy użyciu JSON Web Signature zgodnie z RFC 7515.

Standard PolishAPI nie przewiduje możliwości podania na wejściu dwóch niezależnych od siebie list uprawnień (privilege_list) dla tego samego rachunku, o ile listy te zawierają to samo uprawnienie (np. możliwość wyświetlenia historii operacji zaksięgowanych). Dodatkowo standard zakłada, że wybór rachunku przez użytkownika (PSU) po stronie ASPSP, jeśli jest umożliwiony przez ASPSP, nie może w rezultacie doprowadzić do pojawienia się w zgodzie dwóch list uprawnień zawierających to samo uprawnienie (z różnymi parametrami) dla tego samego rachunku - mechanizm autoryzacji powinien uwzględniać konieczność ograniczenia wyborów PSU w celu zapobieżenia wystąpienia takiej sytuacji (np. poprzez uniemożliwienie wybrania niektórych rachunków w kontekście już dokonanych wyborów lub walidację uniemożliwiającą zaakceptowanie tak wypełnionej zgody).

Ograniczenie powyższe jest niezbędne dla zapewnienia jednoznacznego przyporządkowania występujących po autoryzacji wywołań API do konkretnego uprawnienia i zapewnienia rozstrzygalności faktu, czy żądanie jest autoryzowane.

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 przekierowania 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*".

W przypadku błędu jaki może wystąpić po stronie ASPSP, przed lub w trakcie procesu uwierzytelnienia PSU lub autoryzacji dostępu do zasobów, ASPSP musi poinformować o tym fakcie TPP oraz wskazać rodzaj sytuacji wyjątkowej, która ten błąd spowodowała. W tym celu również wykorzystywane jest zwrotne żądanie przekierowania przeglądarki PSU na adres redirect_uri wskazny przez TPP. Rodzaj błędu oraz opcjonalne informacje szczegółowe na jego temat umieszczane są w dedykowanych parametrach, zlokalizowanych w nagłówku *Location* tego przekierowania. Następujące parametry nagłówka *Location* służą do przekazania informacji o błędzie:

[error] – parametr wymagany, jego wartość określa rodzaj sytuacji wyjątkowej, która spowodowała błąd; parametr musi przyjmować jedną z następujących wartości słownikowych:

- invalid_request – w przypadku gdy żądanie przekierowania przeglądarki PSU do ASPSP było błędne, w szczególności gdy nie było zgodne z adresem przekazanym przez ASPSP w odpowiedzi na żądanie /authorize interfejsu XS2A
- invalid_authentication – w przypadku gdy PSU nie był w stanie poprawnie zakończyć procesu uwierzytelnienia po stronie ASPSP lub wystąpiła niezgodność parametru *psuIdentifierValue* z tożsamością uwierzytelnionego PSU
- context_mismatch – w jednym z następujących przypadków:
    a. uwierzytelniony PSU nie jest klientem indywidualnym, a w żądaniu /authorize nie wskazano kontekstu korporacyjnego
    b. uwierzytelniony PSU nie jest klientem korporacyjnym, a w żądaniu /authorize wskazano kontekst korporacyjny
    c. kontekst uwierzytelnionego PSU jest niezgodny z kontekstem przekazanym w żądaniu /authorize
    d. uwierzytelniony PSU posiada więcej niż jeden kontekst, a w żądaniu /authorize nie wskazano żadnego kontekstu
- access_denied – w przypadku gdy PSU nie udzielił autoryzacji TPP dostępu do swoich zasobów lub taka autoryzacja nie została dopuszczona przez ASPSP
- server_error – w przypadku wystąpienia nieoczekiwanego błędu, wynikającego z awarii systemu ASPSP, który uniemożliwił wykonanie procesu uwierzytelnienia PSU lub autoryzacji dostępu do zasobów

- temporarily_unavailable – w przypadku gdy system po stronie ASPSP jest tymczasowo niedostępny, co uniemożliwia przeprowadzenie procesu uwierzytelnienia PSU i autoryzacji dostępu do zasobów

[error_description] – parametr opcjonalny, pozwala na przekazanie dodatkowej informacji biznesowej o szczegółach błędu

[state] – parametr wymagany, wartość parametru musi być zgodna z wartością analogicznego parametru onazwie *state*, przekazanego przez TPP w żądaniu autoryzacji.

Przykładowe przekierowanie zwrotne, wysyłane przez ASPSP w celu przekazania informacji o błędzie:

HTTP/1.1 302 Found

Location: https://[redirect_uri]?error= access_denied&state=[state]

7.2.4 Pobranie tokenu na podstawie *Authorization Code*

PARAMETR	WYMAGALNOŚĆ	KOMENTARZ
grant_type	wymagane	Wartość „authorization_code”
Code	wymagane	Zgodna z wartością przekazaną w kroku 7.2.3
redirect_uri	wymagane	Wartość zgodna z wartością z kroku7.2.1
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	opcjonalne	
scope_details	opcjonalne	

7.2.5 Wycofanie zgody

Wycofanie zgody jest realizowane za pomocą metody /{wersja}/accounts/{wersja}/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.

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 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 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. Opcjonalnie narzędzie EAT zapewnia również określenie przez użytkownika rachunku źródłowego (PIS) lub rachunku/rachunków objętych zgodą (AIS) jeśli nie zostało to zdefiniowane przez TPP w wywołaniu. 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. - true – PSU został uwierzytelniony - 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*

TPP może pobrać nowy *access token* korzystając z *refresh tokenu* (o ile został wydany). Taka sytuacja może mieć miejsce w przypadku usług AIS i PIS, dla których wymagane jest przekazanie access tokena, w następujących sytuacjach:

a) Ważność wydanego pierwotnie access tokena wygasła i dozwolone jest jego odnowienie dla identycznego zakresu zgód

Została zlecona inicjacja przelewu lub paczki przelewów, przy użyciu wybranej metody usługi PIS, i konieczne jest uzyskanie przez TPP nowego access tokena dla innego zakresu zgód tj. na potrzeby sprawdzenia statusu zleconego przelewu lub paczki przelewów, bez ponownego przeprowadzania procedury SCA.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 7.2.4
scope	opcjonalny	Żądany zakres nie może być większy niż ten, który przekazano w kroku 7.2.4
scope_details	opcjonalny	Żądany zakres nie może być większy niż ten, który przekazano w kroku 7.2.4
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) Rozszerzenie standardu OAuth2
user_agent	Wymagane, jeśli is_user_session=true	Informacja dotycząca wersji przeglądarki użytkownika (informacja na potrzeby fraud detection) 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

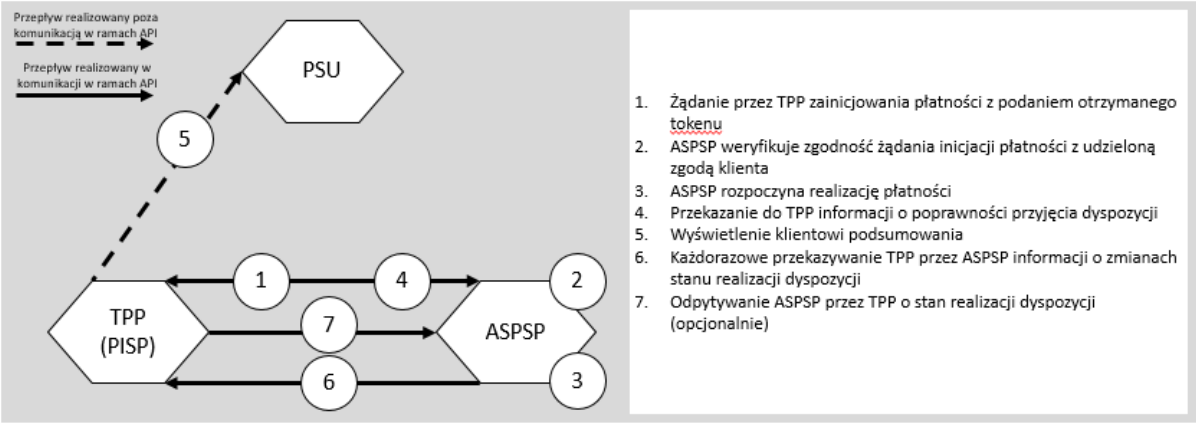
		zgodę na pobranie listy rachunków
scope	opcjonalny	Żądany zakres musi zostać zawężony do wybranych przez PSU rachunków oraz uprawnień dotyczących zakresu żądanych informacji np. szczegóły rachunku, historia transakcji czy szczegóły transakcji
scope_details	opcjonalny	Żądany zakres musi zostać zawężony 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) Rozszerzenie standardu OAuth2
user_agent	Wymagane, jeśli is_user_session=true	Informacja dotycząca wersji przeglądarki użytkownika (informacja na potrzeby fraud detection) 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 otrzymywanych w odpowiedziach.

METODA INTERFEJSU	OPIS	KLASA OBIEKTU KMD
/payments/{wersja}/domestic	Inicjuje przelew krajowy	DomesticRequest/ AddPaymentResponse
/payments/{wersja}/EEA	Inicjuje przelew zagraniczny SEPA	EEARequest/ AddPaymentResponse
/payments/{wersja}/nonEEA	Inicjuje przelew zagraniczny poza SEPA	NonEEARequest / AddPaymentResponse
/payments/{wersja}/tax	Inicjuje przelew do US	TaxRequest / AddPaymentResponse
/payments/{wersja}/bundle	Inicjuje wiele przelewów w formie paczki	BundleRequest / BundleResponse
/payments/{wersja}/getPayment	Pobiera status realizacji przelewu	GetPaymentRequest/ GetPaymentResponse
/payments/{wersja}/getBundle	Pobiera status realizacji paczki przelewów	GetBundleRequest / GetBundleResponse
/payments/{wersja}/getMultiplePayments	Pobiera statusy realizacji wielu płatności. Wywołanie nie wymaga podania tokenu.	GetMultiplePaymentsRequest / GetMultiplePaymentsResponse
/payments/{wersja}/cancelPayment	Odwołuje zainicjowany przelew lub paczkę przelewów	CancelPaymentsRequest/ CancelPaymentsResponse
/payments/{wersja}/recurring	Definiuje nową płatność cykliczną	RecurringRequest/ RecurringResponse
/payments/{wersja}/getRecurringPayment	Pobiera status płatności cyklicznej	GetRecurringPaymentRequest / GetRecurringPaymentResponse

/payments/{wersja}/cancelRecurringPayment	Pozwala na anulowanie płatności cyklicznej	CancelRecurringPaymentRequest/ CancelRecurringPaymentResponse

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 oraz paczek płatności, zainicjowanych z użyciem wybranej metody usługi PIS interfejsu XS2A. Do tego celu zostały zdefiniowane metody interfejsu CallBack o nazwach: paymentCallBack, bundleCallBack. Szczegółową specyfikację techniczną interfejsu 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	PaymentCallBackRequest / CallBackResponse
/payments/{wersja}/bundleCallBack	Przekazuje status realizacji paczki płatności	BundleCallBackRequest / CallBackResponse
/payments/{wersja}/recurringPaymentCallBack	Przekazuje status płatności cyklicznej	RecurringPaymentCallBackRequest / CallBackResponse

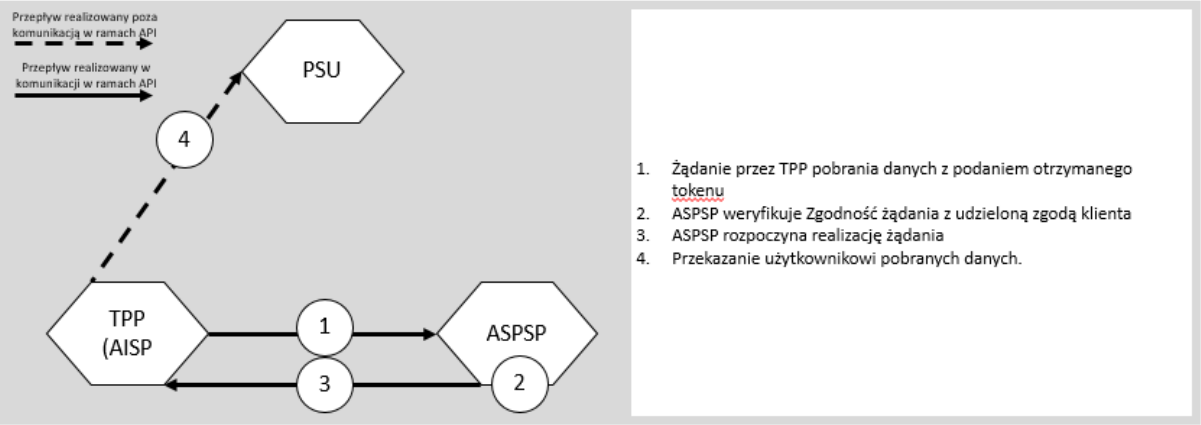
Jako metodę zabezpieczającą API zastosowano typ „apiKey” (https://swagger.io/docs/specification/2-0/authentication/) oraz dodatkowo weryfikowany jest odcisk palca certyfikatu serwerowego TPP, który używany jest do zestawiania połączenia TLS wywołania zwrotnego - przekazywany w parametrze keyID. W wywołanych 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 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}/getTransactionsCancelled	Pobiera transakcje anulowane na rachunku	TransactionInfoRequest/ TransactionsCancelledInfoResponse
/accounts/{wersja}/getTransactionsScheduled	Pobiera transakcje zaplanowane na rachunku	TransactionInfoRequest/ TransactionsScheduledInfoResponse
/accounts/{wersja}/getHolds	Pobiera blokady na rachunku	TransactionInfoRequest/ HoldInfoResponse
/accounts/{wersja}/getTransactionDetail	Pobiera szczegóły pojedynczej transakcji/blokady	TransactionDetailRequest/ TransactionDetailResponse

9.3 Struktura zapytań interfejsu wywołań zwrotnych - *CallBack*

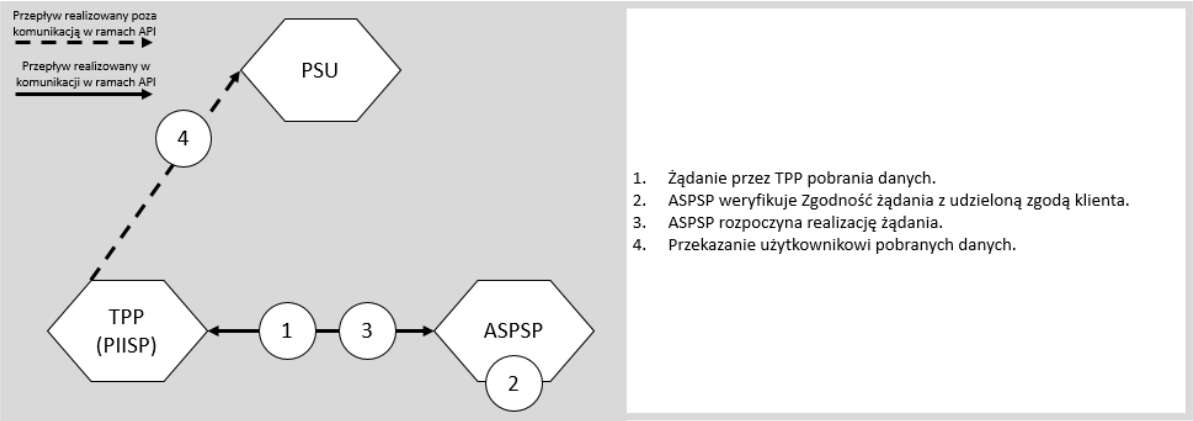
Specyfikacja usługi AIS obejmuje również definicję interfejsu wywołań zwrotnych (ang *CallBack*), dzięki której ASPSP ma możliwość przekazywania do TPP, w sposób asynchroniczny, informacji o rachunku, transakcjach i blokadach, o których udostępnienie wystąpił TPP poprzez wywołanie odpowiednich metod 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 płatniczego	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}/transactionsCancelledCallBack	Przekazuje informacje o transakcjach odwołanych dla danego rachunku płatniczego	TransactionCancelledInfoRequest / CallBackResponse
/accounts/{wersja}/transactionsScheduledCallBack	Przekazuje informacje o transakcjach zaplanowanych dla danego rachunku płatniczego	TransactionScheduledInfoRequest / CallBackResponse
/accounts/{wersja}/transactionsHoldCallBack	Przekazuje informacje o blokadach dla danego rachunku płatniczego	HoldInfoRequest / 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.

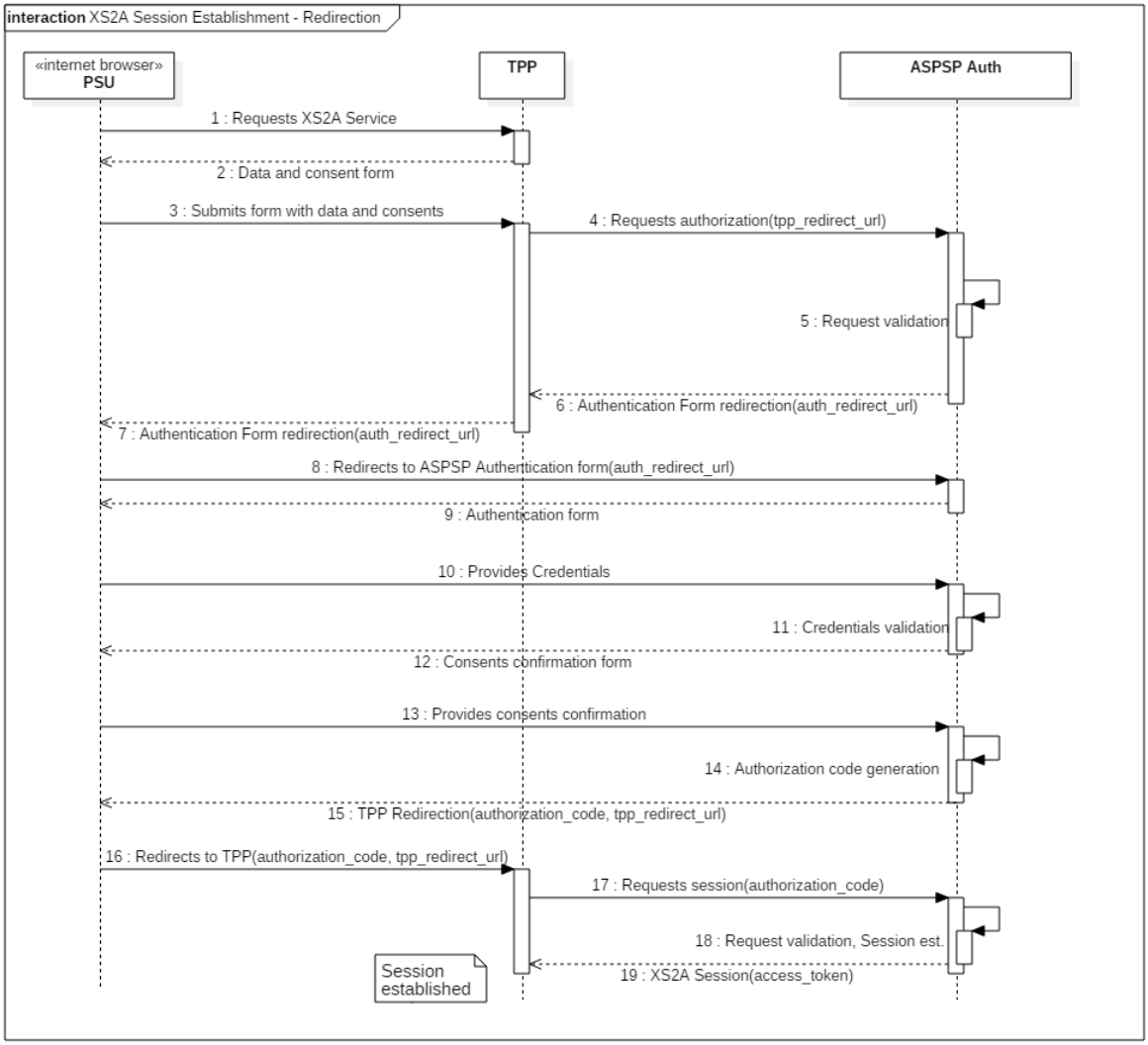
access_token – token dostępu umożliwiający utylizację usług interfejsu XS2A, opisany szerzej w punkcie 5.3 Definicja tokena dostępu.

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 po stronie ASPSP

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



Ilustracja 23: Nawiązywanie sesji XS2A – metoda 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 odpowiedź zawierającą adres URL 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 np. prezentuje formularz z listą wyboru rachunków PSU lub dane inicjowanej transakcji.

13: PSU akceptuje żądane przez TPP zgody poprzez zatwierdzenie zaprezentowanego formularza poprzedzone ewentualnym wyborem podzbioru rachunków (możliwe w przypadku wybranych usług interfejsu XS2A) 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):

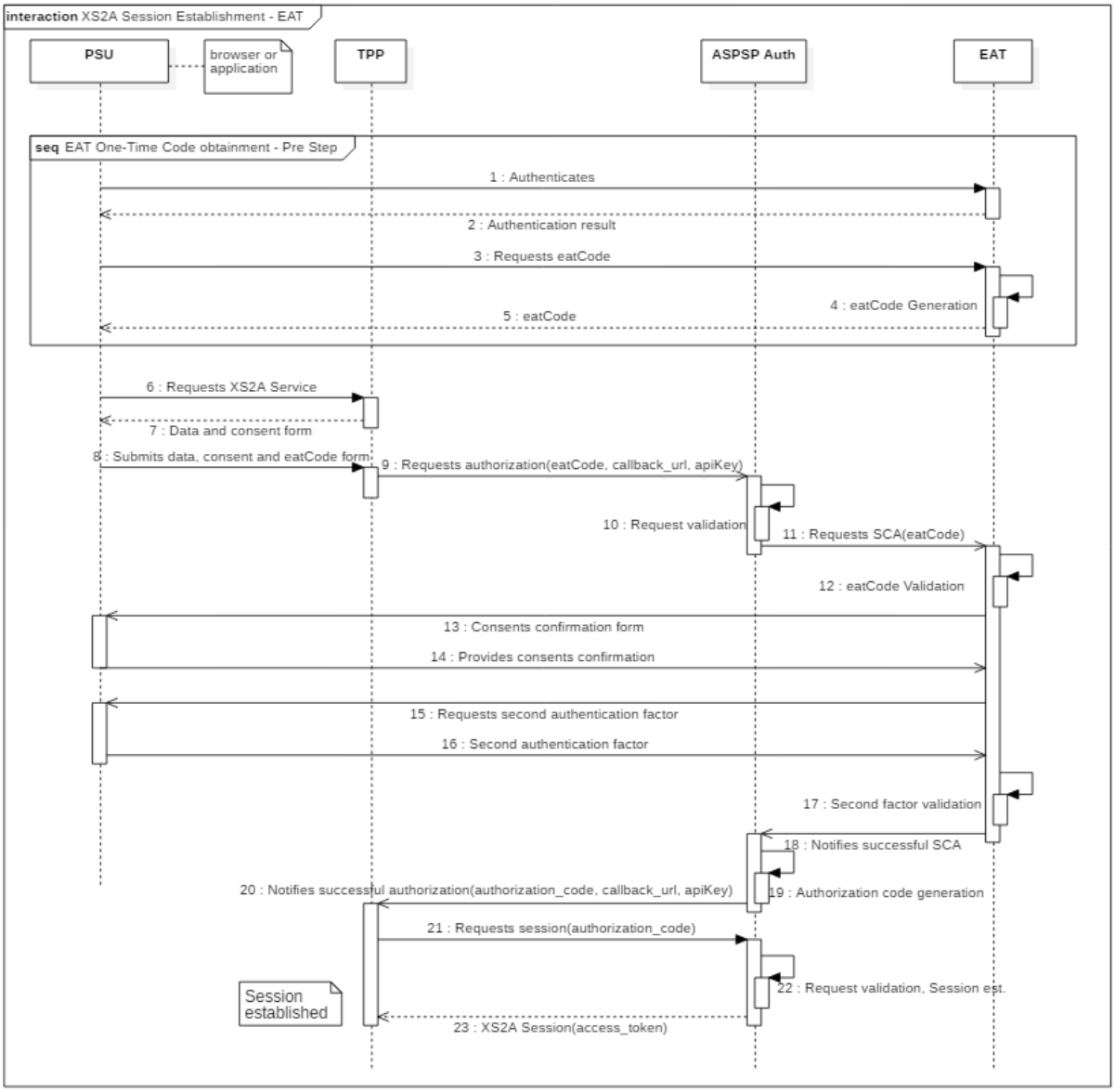
/[VER_1]/auth/[VER_2]/token

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 w zewnętrznym narzędziu autoryzacyjnym (*decoupled)*

Diagram obrazuje sekwencję komunikacyjną, która prowadzi do nawiązania sesji z interfejsem XS2A, z uwzględnieniem uwierzytelnienia PSU metodą *Decoupled*, opisaną w rozdziale 7.3 Mechanizm uwierzytelniania w zewnętrznym narzędziu autoryzacyjnym (decoupled)



Ilustracja 24: Nawiązywanie sesji XS2A – metoda uwierzytelniania w zewnętrznym narzędziu autoryzacyjnym

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. W żądaniu przekazywane są również dane biznesowe określające zakres udzielanej przez PSU zgody. W zależności od usługi interfejsu XS2A mogą to być np. lista rachunków PSU, dane inicjowanej płatności czy numery rachunków do pobrania historii transakcji.

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

13: EAT żąda od PSU udzielenia zgody na realizację usługi interfejsu XS2A przez TPP, w zakresie biznesowym o jaki wystąpił ten TPP, np. prezentuje formularz z listą wyboru rachunków PSU lub dane inicjowanej transakcji.

14: PSU akceptuje żądane przez TPP zgody poprzez zatwierdzenie zaprezentowanego formularza poprzedzone ewentualnym wyborem podzbioru rachunków (możliwe w przypadku wybranych usług interfejsu XS2A)

15: Narzędzie EAT żąda od PSU drugiego faktora w celu ukończenia procedury SCA

16: PSU wykonuje drugi faktor w narzędziu EAT

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

18: Narzędzie EAT powiadamia ASPSP o wyniku przeprowadzonej procedury SCA

19: 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)

20: 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).

21: 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):

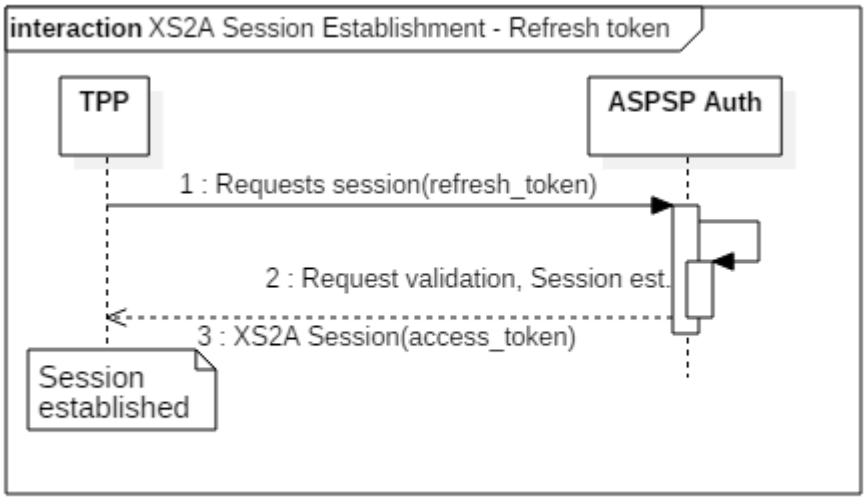
/[VER_1]/auth/[VER_2]/token

22: 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).

23: 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.



Ilustracja 25: Nawiązywanie sesji XS2A – refresh token

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 lub w przypadku konieczności zmiany (zawężenia) zakresu zgód), a z którą związany jest token dodatkowy (refresh_token), zwrócony do TPP podczas pierwotnej procedury nawiązywania sesji. W tym celu TPP wywołuje następującą metodę usługi autoryzacyjnej, zgodnie z opisem w punkcie 7.4, z przekazaniem tokena dodatkowego (refresh_token):

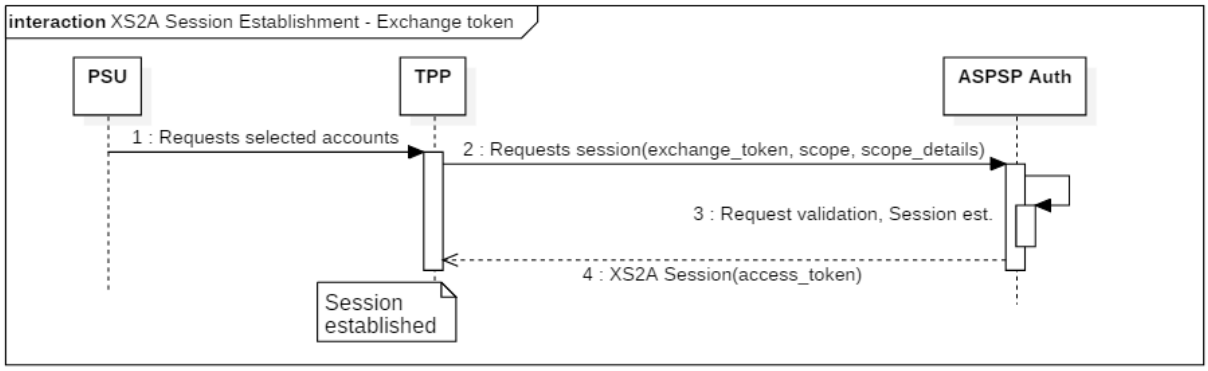
/[VER_1]/auth/[VER_2]/token

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 nowego 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.



Ilustracja 26: Nawiązywanie sesji XS2A – exchange token

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.

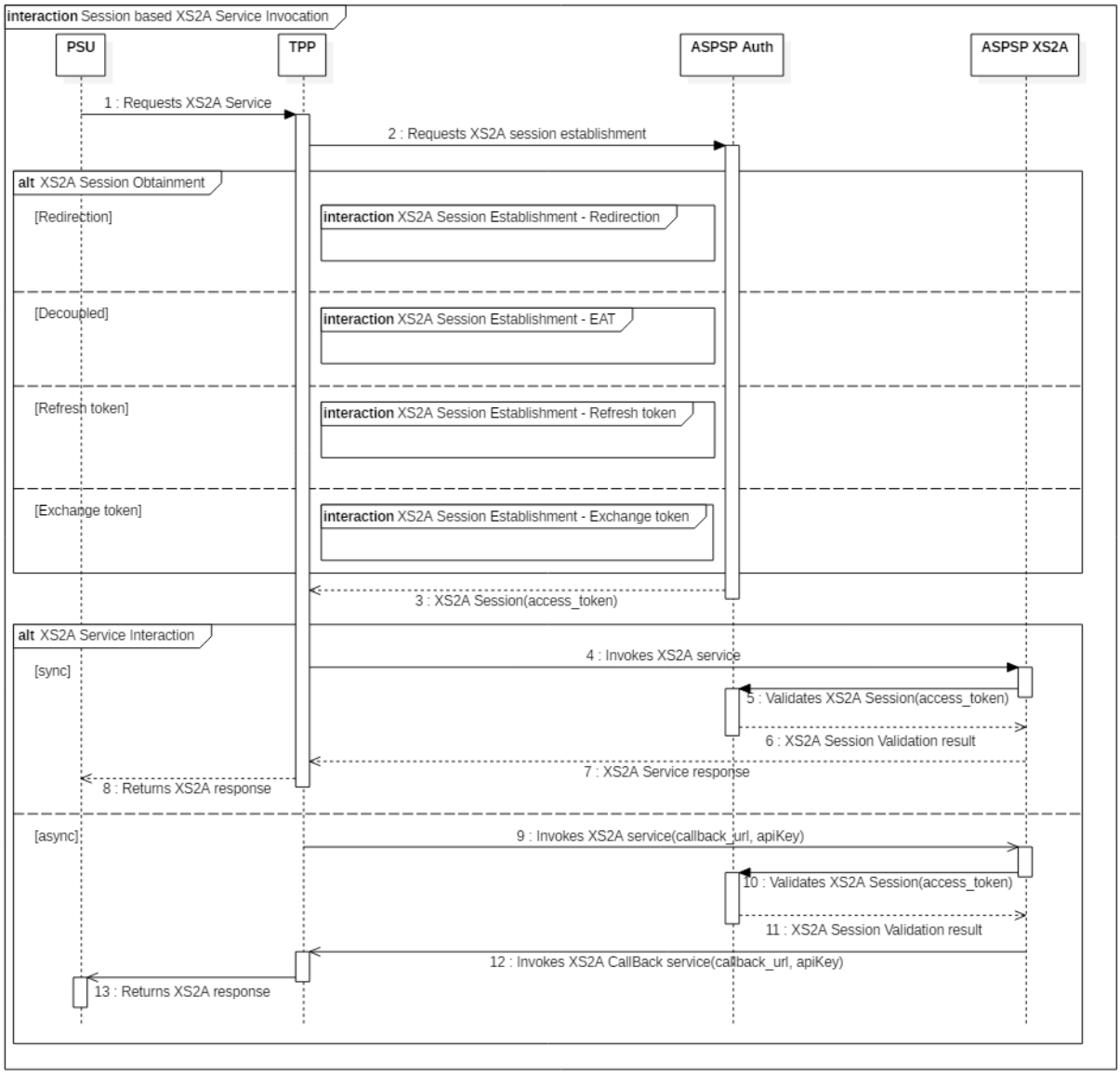
3: ASPSP waliduje otrzymane żądanie nawiązania sesji XS2A poprzez weryfikację otrzymanego tokena dostępowego (przekazanego w atrybucie exchange_token) oraz danych o udzielonych przez PSU zgodach. Po pozytywnej weryfikacji, ASPSP ustanawia nową sesję interfejsu XS2A czego efektem jest wygenerowanie unikalnego, 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]/ getTransactionsCancelled
/[VER_1]/accounts/[VER_2]/ getTransactionsScheduled
/[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]/ bundle
/[VER_1]/payments/[VER_2]/ getPayment
/[VER_1]/payments/[VER_2]/ getBundle
/[VER_1]/payments/[VER_2]/ cancelPayment
/[VER_1]/payments/[VER_2]/ recurring
/[VER_1]/payments/[VER_2]/ getRecurringPayment
/[VER_1]/payments/[VER_2]/ cancelRecurringPayment



Ilustracja 27: Wywoływanie metod interfejsu XS2A z użyciem sesji

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

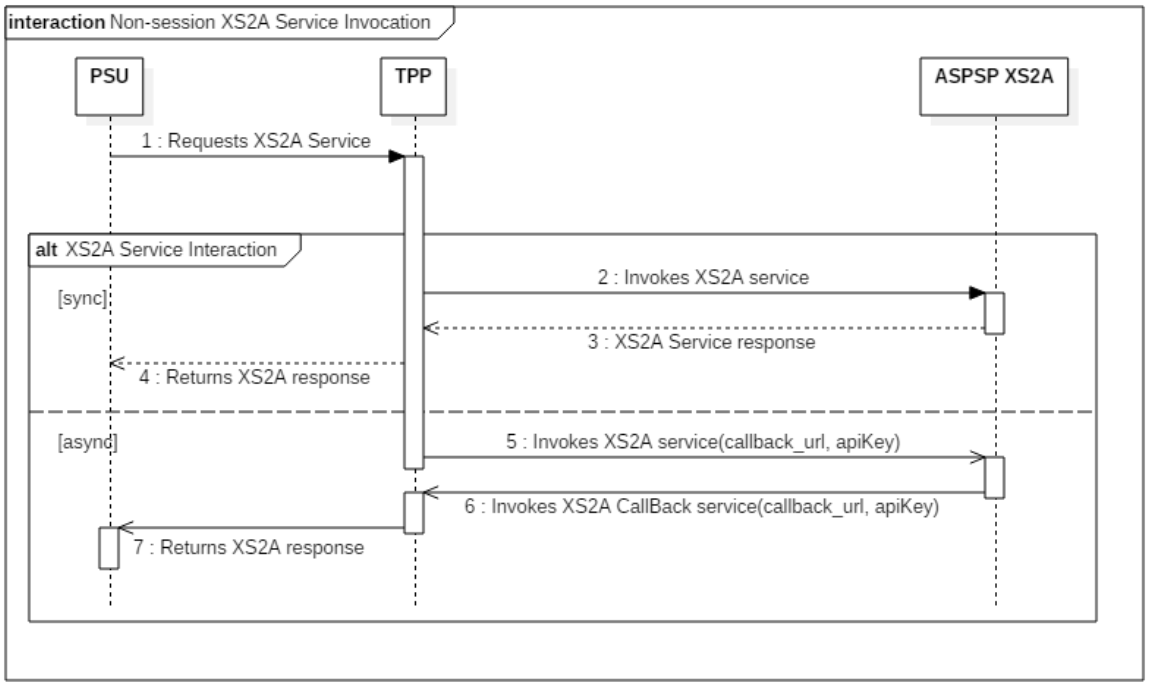
12: W przypadku pozytywnego wyniku walidacji tokena dostępu, ASPSP zwraca wynik realizacji usługi XS2A poprzez wysłanie żądania do interfejsu wywołań zwrotnych interfejsu XS2A po stronie TPP (na adres wskazany w callback_url).

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



Ilustracja 28: Wywoływanie metod interfejsu XS2A bez użycia sesji

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
Wszystkie	Nieprawidłowe żądanie. Brak wymaganych nagłówków lub błędy syntaktyczne w dostarczonym ciele żądania.	400	Obsługa po stronie TPP. Zmiana implementacji klienta interfejsu XS2A prowadząca do budowania żądań zgodnych z obowiązującą specyfikacją tego interfejsu, opublikowaną przez ASPSP.
	Niepoprawna weryfikacja tożsamości TPP. Wynikająca z braku certyfikatu TPP lub braku możliwości nawiązania połączenia przy użyciu mechanizmu wzajemnego uwierzytelnienia protokołu TLS	401	Obsługa po stronie TPP. Weryfikacja poprawności i kompletności certyfikatu wykorzystywanego do nawiązania połączenie z interfejsem XS2A.
	Wysyłanie żądań, które, co do zakresu biznesowego, są niezgodne z uzyskanymi zgodami. Np. próba zainicjowania płatności z wykorzystaniem tokena dostępu uzyskanego w oparciu o zgodę PSU na korzystanie z usług AIS interfejsu XS2A.	403	Obsługa po stronie TPP. Weryfikacja poprawności implementacji. Wysyłanie żądań zgodnych co do zakresu biznesowego z uzyskanymi zgodami.
	Użyto niedozwolonej metody protokołu http. Dozowolona jest tylko metoda POST.	405	Obsługa po stronie TPP. Zmiana implementacji klienta interfejsu XS2A prowadząca do budowania żądań z użyciem metody POST.
	Nieprawidłowy nagłówek accept w żądaniu (serwer nie jest go w stanie obsłużyć)	406	Obsługa po stronie TPP. Zmiana implementacji klienta interfejsu XS2A prowadząca do wysyłania żądań z poprawną wartością nagłówka *accept*, zgodną z obowiązującą specyfikacją tego interfejsu, opublikowaną przez ASPSP.
	Nieprawidłowy nagłówek Content-Type został ustawiony w żądaniu	415	Obsługa po stronie TPP. Zmiana implementacji klienta interfejsu XS2A prowadząca do wysyłania żądań z poprawną wartością nagłówka *Content-Type*, zgodną z obowiązującą specyfikacją tego interfejsu, opublikowaną przez ASPSP.

	Błędy walidacji danych biznesowych przesłanych w ciele żądania.	422	Obsługa po stronie TPP. Zmiana implementacji klienta interfejsu XS2A prowadząca do wysyłania żądań zawierających poprawne dane biznesowe, zgodne z obowiązującą specyfikacją tego interfejsu, opublikowaną przez ASPSP.
	Wewnętrzny błąd serwera interfejsu XS2A nieujawnionego pochodzenia	500	Obsługa po stronie ASPSP. Eliminacja przyczyn wystąpienia błędu i jak najszybsze przywrócenia poprawnego działania interfejsu XS2A.
	Interfejs XS2A po stronie ASPSP nie wspiera użytej przez TPP funkcjonalności	501	Obsługa po stronie TPP – użyta przez TPP funkcjonalność nie jest wspierana przez usługi interfejsu XS2A po stronie ASPSP i w związku z tym żądanie użycia tej funkcjonalności nie powinno być ponawiane przez TPP
	Interfejs XS2A jest tymczasowo niedostępny ze względu na zwiększone obciążenie serwera lub prowadzone prace utrzymaniowo-konserwacyjne.	503	Obsługa po stronie ASPSP – powiadomienie TPP, w ramach wysyłanej odpowiedzi na żądanie, o planowanym czasie niedostępności i przywrócenie działania interfejsu XS2A w deklarowanym terminie. Obsługa po stronie TPP – ponowienie wysłania żądania po zadeklarownym przez ASPSP terminie przywrócenia działania interfejsu XS2A. Powiadomienia PSU o chwilowej niedostępności usług związanych z korzystaniem z interfejsu XS2A.
Inicjacja procesu uwierzytelniania i autoryzacji (metody /authorize i /authorizeExt)	Niepoprawna weryfikacja licencji TPP, który ubiega się o uzyskania zgody, której zakres nie jest zgodny z posiadaną licencją.	403	Obsługa po stronie TPP. Weryfikacja uzyskanej licencji. Wysyłanie żądań zgodnych z posiadaną licencją.
Uwierzytelnienie PSU oraz autoryzacja dostępu do zasobów ASPSP w domenie ASPSP (po przekierowaniu przeglądarki PSU w domenę	Żądanie przekierowania przeglądarki PSU do ASPSP było błędne, w szczególności gdy nie było zgodne z adresem przekazanym przez ASPSP w	302	Obsługa po stronie ASPSP – wysłanie zwrotnego przekierowania w domenę TPP, z przekazaniem w nagłówku *Location*

ASPSP w metodzie autoryzacji „po stronie ASPSP")	odpowiedzi na żądanie /authorize interfejsu XS2A		informacji o przyczynie wystąpienia błędu. Obsługa po stronie TPP - Weryfikacja implementacji klienta interfejsu XS2A w celu poprawienia mechanizmu przekierowania PSU w domenę ASPSP.
	PSU nie udzielił autoryzacji TPP dostępu do swoich zasobów lub taka autoryzacja nie została dopuszczona przez ASPSP	302	Obsługa po stronie ASPSP – wysłanie zwrotnego przekierowania w domenę TPP, z przekazaniem w nagłówku *Location* informacji o przyczynie wystąpienia błędu. Obsługa po stronie TPP – przekazanie informacji PSU o uzyskanym od ASPSP wyniku autoryzacji.
	PSU nie był w stanie poprawnie zakończyć procesu uwierzytelnienia po stronie ASPSP lub wystąpiła niezgodność parametru *psuIdentifierValue* z tożsamością uwierzytelnionego PSU	302	Obsługa po stronie ASPSP – wysłanie zwrotnego przekierowania w domenę TPP, z przekazaniem w nagłówku *Location* informacji o przyczynie wystąpienia błędu. Obsługa po stronie TPP – przekazanie informacji PSU o uzyskanym od ASPSP wyniku uwierzytelnienia tego PSU.
	ASPSP zidentyfikował błędy podczas weryfikacji kontekstu uwierzytelnionego PSU	302	Obsługa po stronie ASPSP – wysłanie zwrotnego przekierowania w domenę TPP, z przekazaniem w nagłówku *Location* informacji o przyczynie wystąpienia błędu. Obsługa po stronie TPP - Weryfikacja implementacji klienta interfejsu XS2A w celu poprawienia procesu uwierzytelniania PSU lub zebranie dodatkowych, poprawnych informacji na temat kontekstu od PSU, i ponowienie procesu jego uwierzytelnienia i autoryzacji dostępu do zasobów ASPSP.
Nawiązywanie sesji interfejsu XS2A (metoda /token)	Kod autoryzacji, dla grant_type= authorization_code, jest pusty lub niewłaściwy syntaktycznie.	400	Obsługa po stronie TPP – weryfikacja implementacji klienta interfejsu XS2A w celu poprawienia mechanizmu nawiązywania sesji w

			metodzie grant_type= authorization_code.
	Kod autoryzacji, dla grant_type= authorization_code, jest nieważny.	403	Obsługa po stronie TPP – ponowna inicjalizacja procesu uwierzytelnienia PSU i autoryzacji dostępu do zasobów ASPSP, w celu uzyskania nowego kodu autoryzacyjnego.
	Wartość tokena refresh_token, dla grant_type=refresh_token, jest pusta lub niepoprawna syntaktycznie.	400	Obsługa po stronie TPP – weryfikacja implementacji klienta interfejsu XS2A w celu poprawienia mechanizmu nawiązywania sesji w metodzie grant_type= refresh_token.
	Wartość tokena refresh_token, dla grant_type=refresh_token, nie pozwoliła na zidentyfikowanie ważnej sesji interfejsu XS2A po stronie ASPSP.	403	Obsługa po stronie TPP – ponowna inicjalizacja procesu uwierzytelnienia PSU i autoryzacji dostępu do zasobów ASPSP, w celu nawiązanie nowej sesji interfejsu XS2A.
	Wartość tokena exchange_token, dla grant_type=exchange_token, jest pusta lub niepoprawna syntaktycznie.	400	Obsługa po stronie TPP – weryfikacja implementacji klienta interfejsu XS2A w celu poprawienia mechanizmu nawiązywania sesji w metodzie grant_type= exchange_token.
	Wartość tokena exchange_token, dla grant_type=exchange_token, nie pozwoliła na zidentyfikowanie ważnej sesji interfejsu XS2A po stronie ASPSP.	403	Obsługa po stronie TPP – ponowna inicjalizacja procesu uwierzytelnienia PSU i autoryzacji dostępu do zasobów ASPSP, w celu nawiązanie nowej sesji interfejsu XS2A.
	Niepoprawna weryfikacja licencji TPP, który ubiega się o uzyskanie zgody, której zakres nie jest zgodny z posiadaną licencją. Sytuacja dotyczy niepoprawnych wartości parametrów scope lub scope_details, w przypadkach gdy ich wypełnienie jest zasadne (np. uzyskanie zgody na zapytanie o status zainicjowanej płatności lub zmiana zgody w celu uszczegółowienia zapytań do usługi AIS) dla metod grant_type=exchange_token oraz grant_type=refresh_token.	403	Obsługa po stronie TPP – weryfikacja implementacji klienta interfejsu XS2A w celu poprawienia mechanizmu nawiązywania sesji w oparciu o zawężony zakres zgód.

Wywołanie usług biznesowych (AIS, PIS) interfejsu XS2A	ASPSP otrzymuje żądanie, w którym wartość tokena dostępu jest pusta lub brak nagłówka *Authorization* (nie dotyczy metod /deleteConsent oraz /getMultiplePayments)	401	Obsługa po stronie TPP - weryfikacja implementacji klienta interfejsu XS2A w celu poprawienia mechanizmów wysyłania żądań do interfejsu XS2A.
	ASPSP otrzymuje żądanie, w którym token dostępu utracił ważność	401	Obsługa po stronie TPP - odświeżenie tokena przy użyciu metody /token i grant_type=refresh_token.
	ASPSP otrzymuje żądanie, dla którego zgoda klienta (sesja intefejsu XS2A) utraciła ważność	403	Obsługa po stronie TPP - rozpoczęcie procesu uwierzytelniania PSU i autoryzacji dostępu do zasobów ASPSP w celu odświeżenia zgody lub uzyskania nowej z wykorzystaniem metody /authorize lub /authorizeExt.
	ASPSP otrzymuje żądanie bez podpisu JWS-SIGNATURE	400	Obsługa po stronie TPP - weryfikacja implementacji klienta interfejsu XS2A w celu poprawienia mechanizmów generowania podpisu JWS-SIGNATURE i/lub wysyłania żądań do interfejsu XS2A.
	ASPSP otrzymuje żądanie z podpisem JWS-SIGNATURE, który nie został poprawnie zwalidowany.	422	Obsługa po stronie TPP - weryfikacja implementacji klienta interfejsu XS2A w celu poprawienia mechanizmu generowania podpisu JWS-SIGNATURE. Weryfikacja poprawności i ważności certyfikatu (pieczęci elektronicznej) używanego przez TPP do generowanie podpisu.
	ASPSP otrzymuje żądanie, które nie może zostać obsłużone ze względu na przekroczenie obowiązującego limitu żądań (dotyczy usługi AIS i żądań wysyłanych bez udziału PSU).	429	Obsługa po stronie TPP - ponowienie wysłania żądania po resecie wartości licznika żądań

W przypadku błędów z rodziny 4xx oraz 5xx, szczegóły biznesowe, opisujące przyczyny wystąpienia błędu, mogą zostać opisane przy użyciu struktury JSON o nazwie Error i przekazane zwrotnie do TPP w ciele odpowiedzi.

W przypadku błędów, które ujawniły się po stronie ASPSP, po przekierowaniu przeglądarki PSU do domeny ASPSP (w metodzie uwierzytelniania po stronie ASPSP), szczegóły biznesowe, opisujące przyczyny wystąpienia błędu, mogą zostać opisane w nagłówku *Location* przekierowania zwrotnego (kod 302) do domeny TPP, w zgodzie z regułami opisanymi w punkcie 7.2.3 specyfikacji PolishAPI.

Treści biznesowe zwracanych przez ASPSP błędów, oraz ich dodatkowe identyfikatory, zostaną opisane indywidulanie, przez każde ASPSP, w specyfikacji interfejsu XS2A, bazującej na standardzie PolishAPI, z zachowaniem opisanych sytuacji wyjątkowych i kodów błędów HTTP.

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-ver3_0.yaml

Załącznik nr 2: PolishAPI-CallBack-ver3_0.yaml