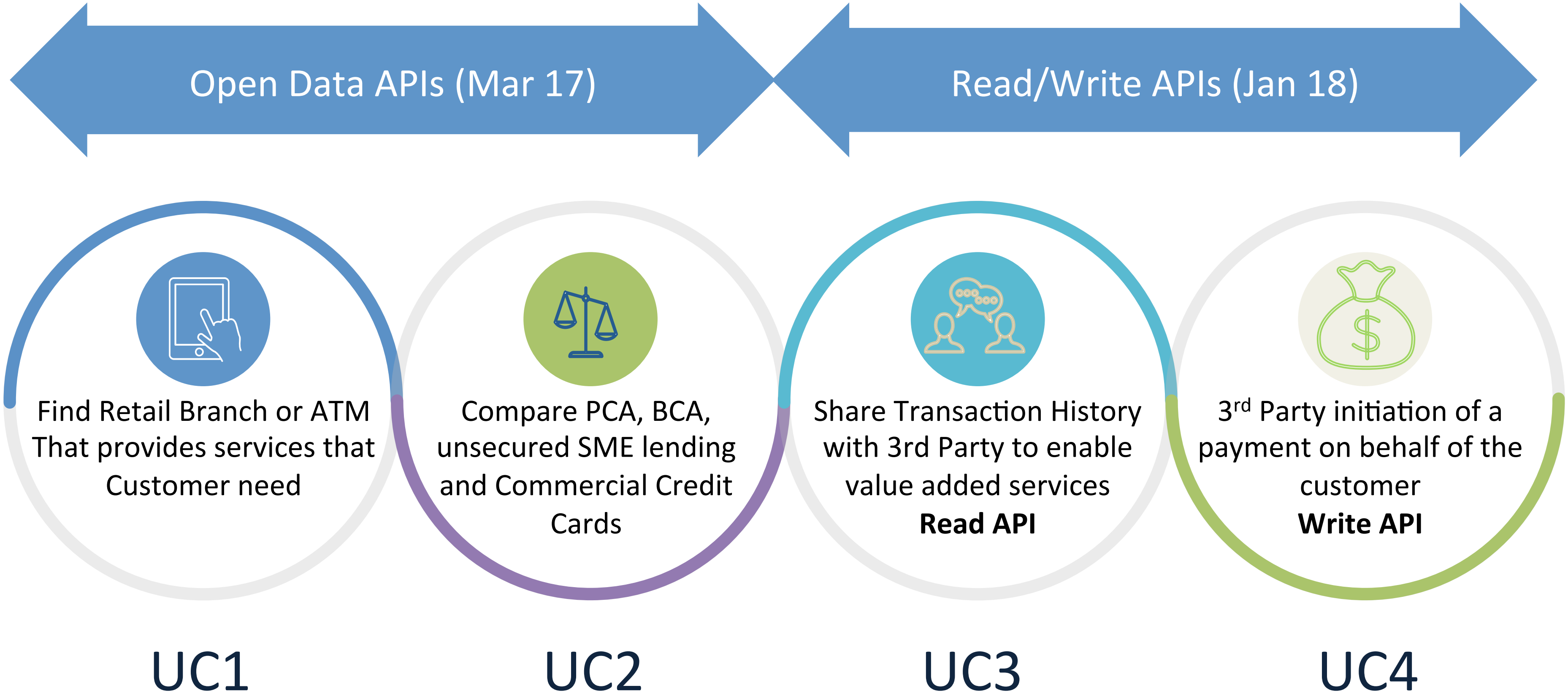
Open Banking Update

22 May 2017

Goals

- To share latest designs for Open APIs, Read/Write APIs, Trust Framework, and OB Directory.
- To give clarity on what we are (and are not) delivering for Jan 2018.
- To demonstrate the specifications and how to give feedback.
- To capture and address any questions or concerns.

CMA High Level Use Cases for APIs



Open Data APIs (Mar 17)

Read/Write APIs (Jan 18)

Find Retail Branch or ATM That provides services that Customer need

UC1

Compare PCA, BCA, unsecured SME lending and Commercial Credit Cards

UC2

Share Transaction History with 3rd Party to enable value added services
Read API

UC3

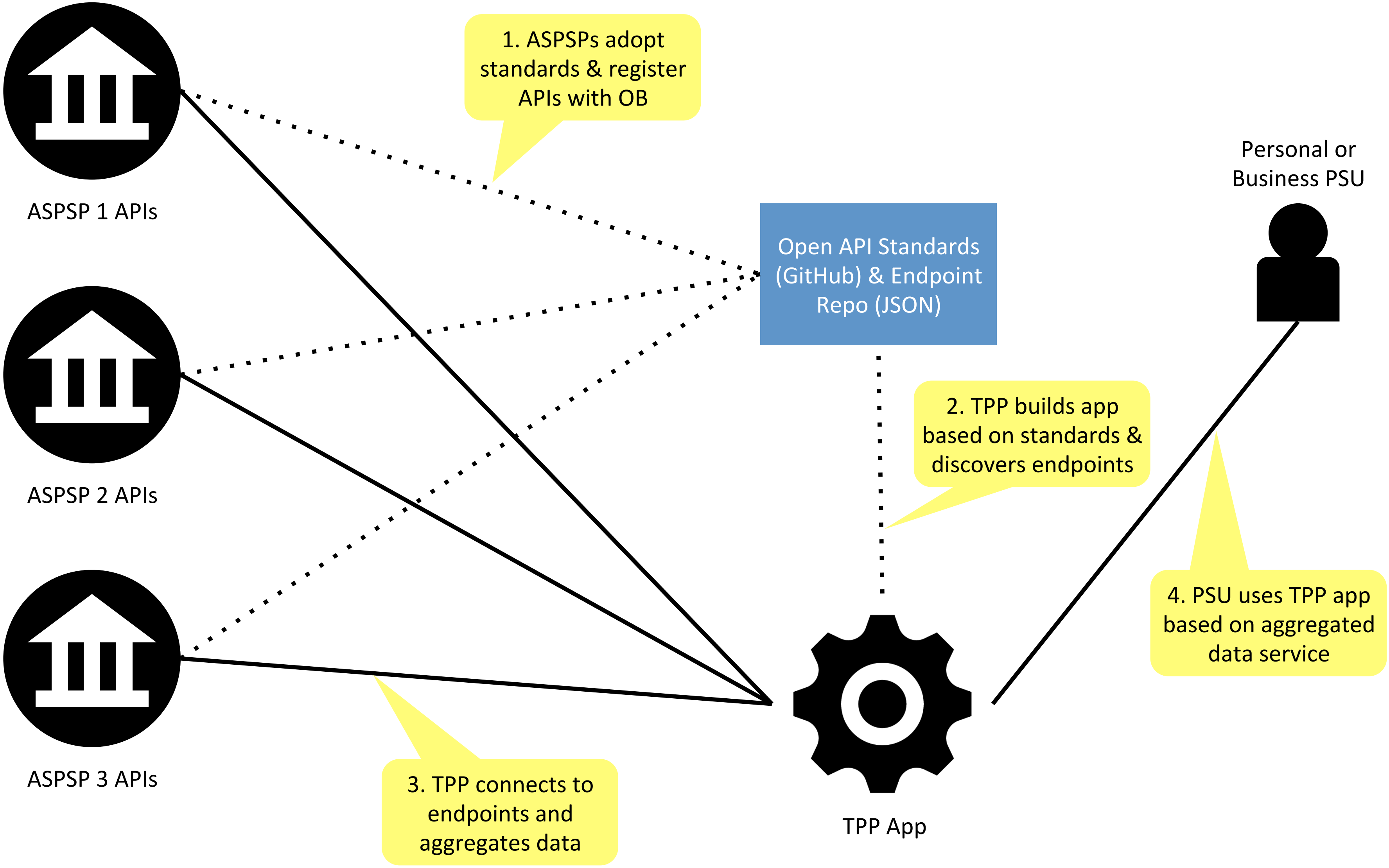
3rd Party initiation of a payment on behalf of the customer
Write API

UC4

Open APIs

Rationale for and status of v2.0

Open API ecosystem



1. ASPSPs adopt standards & register APIs with OB

2. TPP builds app based on standards & discovers endpoints

3. TPP connects to endpoints and aggregates data

4. PSU uses TPP app based on aggregated data service

Open API endpoints

/atms

Endpoint for getting ATM data

/branches

Endpoint for getting Branch data

/personal-current-accounts

Endpoint for getting Personal Current Account data

/business-current-accounts

Endpoint for getting Business Current Account data

/unsecured-sme-loans

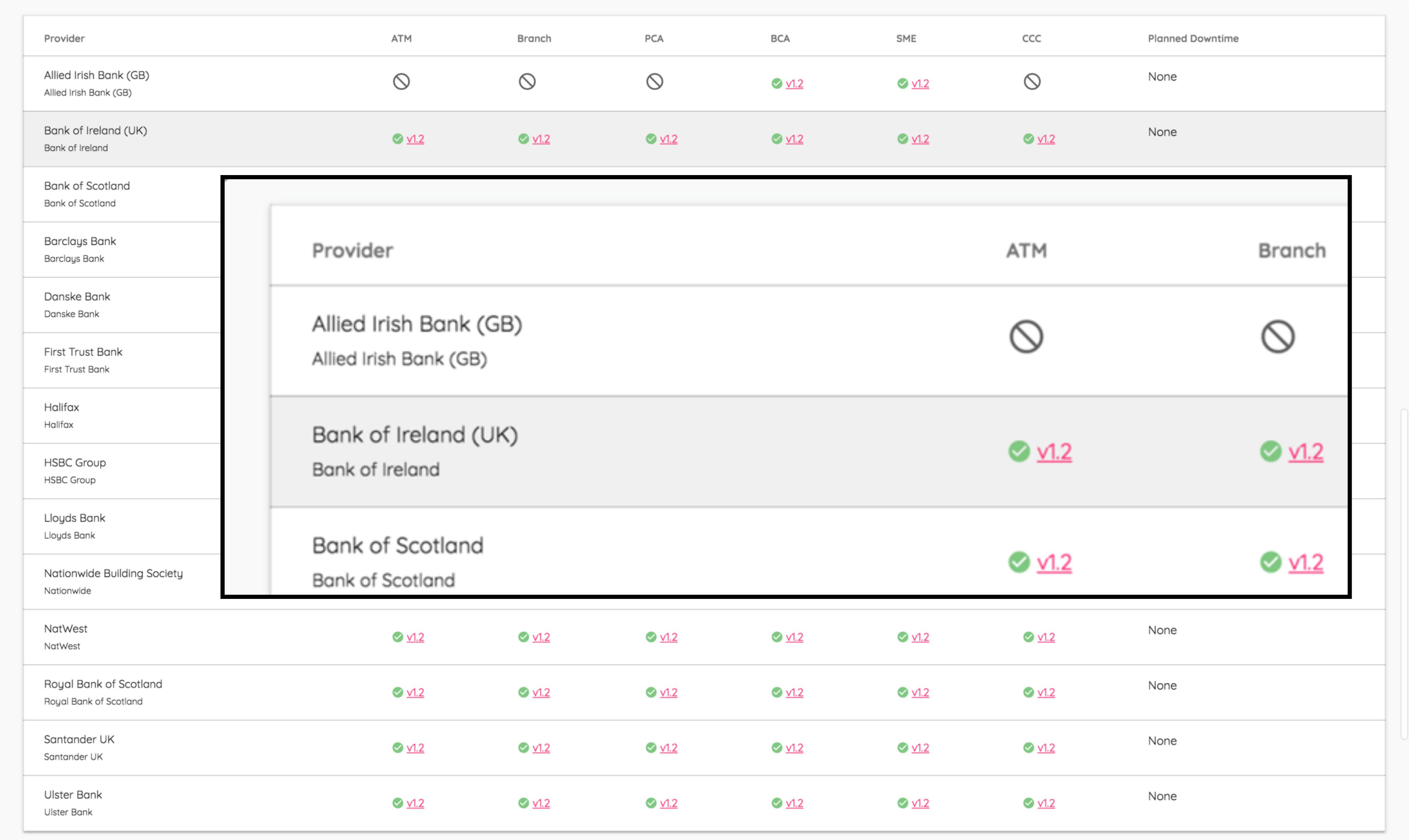
Endpoint for getting Unsecured SME Loan data

/commercial-credit-cards

Endpoint for getting Commercial Credit Card data

https://github.com/OpenBankingUK/opendata-api-spec-compiled

Current adoption of v1.2/1.3



Provider	ATM	Branch	PCA	BCA	SME	CCC	Planned Downtime
Allied Irish Bank (GB) Allied Irish Bank (GB)	⊘	⊘	⊘	✓ v1.2	✓ v1.2	⊘	None
Bank of Ireland (UK) Bank of Ireland	✓ v1.2	✓ v1.2	✓ v1.2	✓ v1.2	✓ v1.2	✓ v1.2	None
Bank of Scotland Bank of Scotland							
Barclays Bank Barclays Bank							
Danske Bank Danske Bank							
First Trust Bank First Trust Bank							
Halifax Halifax							
HSBC Group HSBC Group							
Lloyds Bank Lloyds Bank							
Nationwide Building Society Nationwide							
NatWest NatWest	✓ v1.2	✓ v1.2	✓ v1.2	✓ v1.2	✓ v1.2	✓ v1.2	None
Royal Bank of Scotland Royal Bank of Scotland	✓ v1.2	✓ v1.2	✓ v1.2	✓ v1.2	✓ v1.2	✓ v1.2	None
Santander UK Santander UK	✓ v1.2	✓ v1.2	✓ v1.2	✓ v1.2	✓ v1.2	✓ v1.2	None
Ulster Bank Ulster Bank	✓ v1.2	✓ v1.2	✓ v1.2	✓ v1.2	✓ v1.2	✓ v1.2	None

Provider	ATM	Branch
Allied Irish Bank (GB) Allied Irish Bank (GB)	⊘	⊘
Bank of Ireland (UK) Bank of Ireland	✓ v1.2	✓ v1.2
Bank of Scotland Bank of Scotland	✓ v1.2	✓ v1.2

Version 2

Version 2 is a significant upgrade of the Open API standard, and will include the following enhancements:

- Fix all known defects, implement all agreed CRs, and address other CRs under discussion.
- Remodel data structure to make it more generic, and thus better suited to both back book and future products.
- Implement ISO20022 elements wherever possible and relevant, to simplify/speed up the ISO submission process.
- Build in validation from Third Parties, to ensure it is optimised for API users as far as possible.
- Review and potential simplification of user licence and terms, to remove any unnecessary restrictions, and thereby increase take-up.
- Review and update functional and non-functional requirements, including updates to ciphers, to ensure the standard is secure and fit for purpose.
- Review and update to API console and API dashboard, to reduce complexity and cost of ownership.
- Overall make it a more stable and useable standard, which is easier to support for both API providers (not just the CMA9), but also API users.

Read/Write APIs have a dependency on this

- In order to meet the CMA order for product comparison (primarily for back book products), there is a need for the Read/Write Account API to expose the relevant entities from the Open API standard, as back book products are not included in the Open Data standard.
- We have defined this as a separate endpoint (Product API) with data specific to each PCA/BCA customer account.
- This will need to have a similar data structure to the Open API standard (to enable a like for like comparison).
- Some ASPSPs may chose to use this Product API for populating and exposing front as well as back book products for their customers.
- In any case, we cannot complete the design of this API until Open API 2.0 is published.

Schedule

- Started Apr 2017
- Series of 2 week sprints with release candidate at end of each
- Publish v2 Draft Aug 2017
- Live with CMA9 by Jan 2018

What is not included in this release

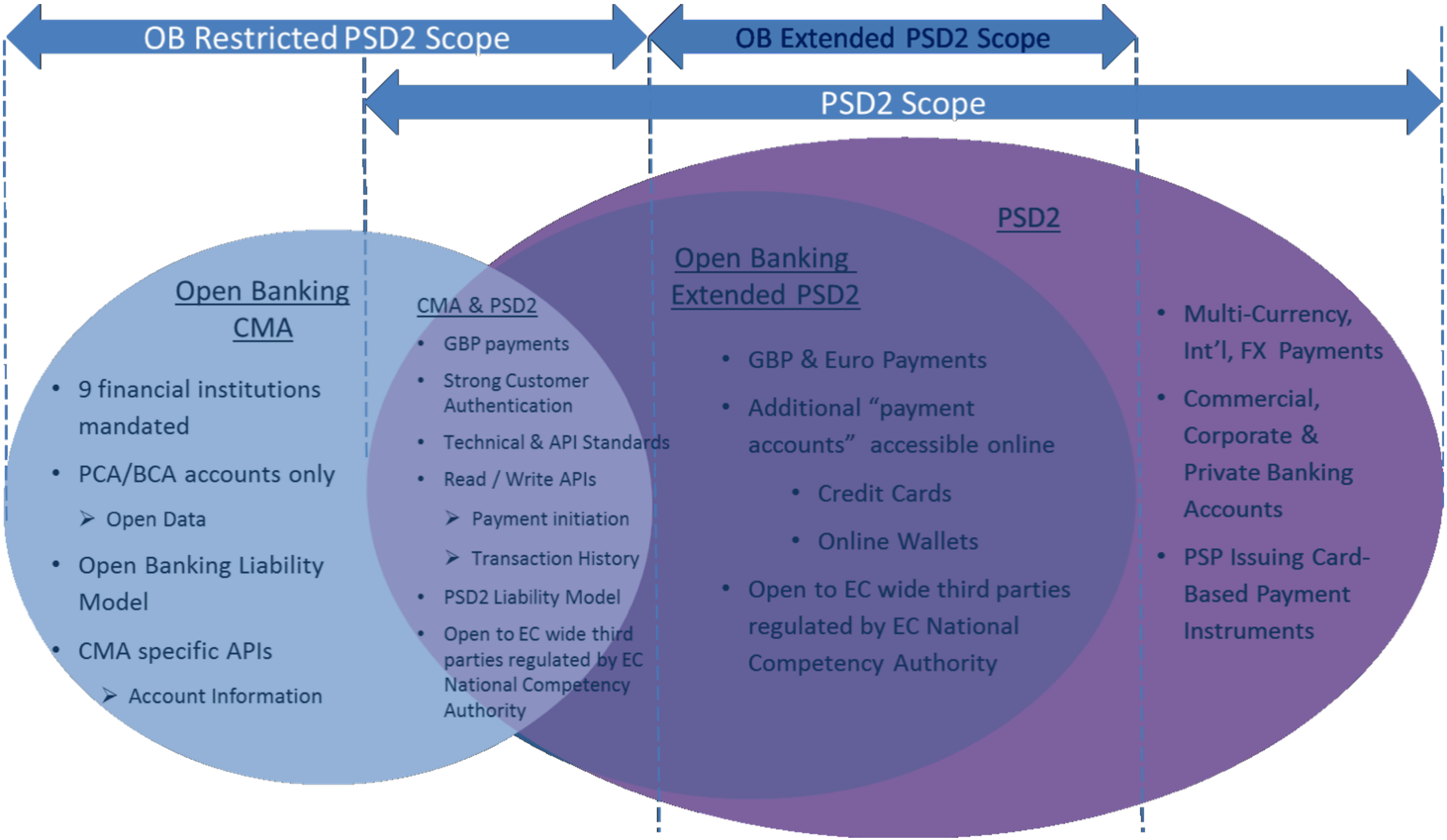
There are a number of further enhancements which are not planned for v2 and, and these could be considered candidates for potential future releases:

- Updates based on ‘real world’ feedback from all users (API providers and API users) one these APIs are used in conjunction with the Read/Write APIs from Jan 2018 onwards.
- Updates based on feedback from ISO submission process.
- Extension to other account types (e.g. Mortgages).
- Extension to non UK products.
- Aggregation of all API provider end points to create a central service.

Read/Write APIs

Initial Scope for Jan 2018 Delivery

CMA v PSD2 Scope



OB Restricted PSD2 Scope

OB Extended PSD2 Scope

PSD2 Scope

Open Banking CMA

- 9 financial institutions mandated
- PCA/BCA accounts only
  - ➢ Open Data
- Open Banking Liability Model
- CMA specific APIs
  - ➢ Account Information

CMA & PSD2

- GBP payments
- Strong Customer Authentication
- Technical & API Standards
- Read / Write APIs
  - ➢ Payment initiation
  - ➢ Transaction History
- PSD2 Liability Model
- Open to EC wide third parties regulated by EC National Competency Authority

Open Banking Extended PSD2

- GBP & Euro Payments
- Additional "payment accounts" accessible online
  - Credit Cards
  - Online Wallets
- Open to EC wide third parties regulated by EC National Competency Authority

PSD2

- Multi-Currency, Int'l, FX Payments
- Commercial, Corporate & Private Banking Accounts
- PSP Issuing Card-Based Payment Instruments

Use case summary for Read/Write APIs

	TPP Proposition	Enabling Features	Use Cases Enabled
UC3 Read API	Consumer/SME Product Comparison	• **Read API Endpoints** : Transaction & Personalised Product Information (aligned to v2.0 of Open Data Product API (UC2) • Single access token, 12 months minimum historical data.	Access to personalised product meta data that is comparable to Open Data API + transaction history enables accurate PCA/BCA product comparison. NB: Personalised Product Info is **critical** to meet this **Key** CMA remedy.
	Consumer Account Aggregation	• **Read API Endpoints** : Transaction, Balance, SO, DD and Beneficiaries'. • Multiple access token, 12 months minimum historical data.	Single Aggregated account dashboard, analytics on spending and income patterns across multiple accounts. Provides customers with insights on opportunities to budget and reduce undue charges
	SME Accountancy Package	• **Read API Endpoints** : Transaction, Balance, SO, DD and Beneficiaries • Multiple access token, 24 months minimum historical data.	Single aggregated account dashboard, analytics on spending, income, Cash flow, liquidity forecasting, invoice reconciliation.
	SME : Lending	• **Read API Endpoints** : Transaction • Single access token, up-to 36 months historical data.	Access to historical transaction data over a long period (up-to 36 months) allows lender to model seasonal variation and produce more accurate risk profiles before making lending decisions
	SME Soft ID verification / Fraud Checking for Lending	• **Read API Endpoints** : Account ID (Sort Code etc.), Account Name, Personal Information (Account Holder Name, DoB and Addresses)	*Access to Account name, DoB & communications details in addition to transaction history would reduce friction in loan application process and aid fraud checking efforts by the loan provider.* ***DOB & Address are not in v0.1***
UC3 + UC4 Read + Write API	Consumer PFM (Budget Tools , Debt Advice, micro-saving)	• **Read API Endpoints** : Transaction, Balance, SO, DD , Product Info, & Beneficiaries', multiple access token, 12 months min historical data. • **Write API Endpoints** : Single Immediate Payment, single token	Aggregated account dashboard, analytics on spending and income patterns across multiple accounts. Forecasting when a customer is likely to go overdrawn and alerting them to transfer money using payment initiation from one account to another.
	SME Liquidity Management	• **Read API Endpoints** : Transaction, Balance, SO, DD , Product Info, and Beneficiaries' APIs, multiple access token, 24+ months minimum historical data. • **Write API Endpoints** : Single Immediate Payment, single token	Single aggregated account dashboard, analytics on spending, income, Cash flow, liquidity forecasting, and alerting the financial controller to transfer surplus funds (or savings for taxation) to better interest paying account .
	Balance display before Payment	• **Read API Endpoints** : Balance API • **Write API Endpoints** : Single Immediate Payment, single token	TPP gets balance from the customer account displays balance before asking payment from the customer. NB : this will require SCA twice which is not ideal. Banks may choose to implement Balance display on payment initiation which is better experience but is in the competitive space
UC4 Write API	Money transfer between accounts	• **Write API Endpoints** : Single Immediate Payment, single token	Move money from my current account from customers Bank A to Bank B
	Con/SME Retail Payment	• **Write API Endpoints** : Single Immediate Payment, single token	An Immediate Payments transfer from a customer's account to the retailer beneficiary bank for payment of purchased of goods
	SME Payment	• **Write API Endpoints** : Single immediate Payment from Accounting Package to 3rd Party Beneficiary	SME is able to pay VAT owed to HMRC via his accounting package.

Note: The above Use Cases are limited PCA & BCA GBP accounts with payments restricted to GBP and within the UK only.

Scope of Payment Types

Existing API design has capability to support the following payment request types but delivery for Jan18 is limited

#	Payment Request Type	Example Use Case
1	**Single Immediate Payment** • *Currency: GBP* • *Account Type: PCA, BCA* • *Scheme: Faster Payments* • *Authorisation: Single* JAN-18	A) **As a Consumer**, **I want to** have the ability to use a TPP tool to move money from one PCA account in credit to another PCA account in debit, **so that** I can avoid overdraft charges. B) **As a Retailer**, **I want to** create an Immediate Payments transfer from a customer's account to mine, **so that** I can quickly create and verify their payment and can fulfil their purchase quickly. **Exception1: No SCA for Payments up to EUR 30 and a cumulative amount of EUR 100 or 5 consecutive individual electronic transactions** **Exception2: No SCA for Payments where originator and beneficiary are the same natural person and both accounts held with same ASPSP**
2	**Single Future Dated Payment:** **i) TTP Schedule** **ii) ASPSP Schedule**	**As a Retailer**, **I want to** create a single future dated payments transfer from a customer's account to mine, **so that** I can receive the funds for the purchase on the agreed delivery date.
3	**Standing Order Payment: (Mandate with ASPSP)**	**As a Subscription Service Provider**, **I want to** efficiently setup a standing order payment from a new customer's account to mine as part of the online signup process, **so that** I can make the signup and subscription process simpler for the customer.
4	**Bulk on-line payments**	**As a TPP/PISP**, **I want to** be able to offer to my SME customers the same bulk payment capabilities they have today from their banks' online channels through my own service offering (e.g. single debit multiple credits & beneficiaries for payroll), **so that** I can offer value added services and use my channel to manage the full customer experience.
5	**Recurring Fixed Payments:** **i) with end date** **ii) without end date**	**As a Subscription Service Provider**, **I want to** get the new customer's consent to quickly and efficiently setup a recurring payment from the customer's account to mine for an initial finite subscription period, as part of the online signup process, **so that** I can make the signup and subscription process simpler for the customer.
6	**Trusted Beneficiary payments**	**As a Consumer**, **I want to** initiate a single payment or single future dated payment to a trusted beneficiary without having to go through SCA from my ASPSP, **so that** my online payment journey is frictionless. *Note 1: Trusted beneficiary is someone whose account I explicitly add to my ASPSP list after following SCA.* *Note 2: Decision to exempt consumer from SCA in case of payment to a trusted beneficiary finally rests with ASPSP due to monitoring requirements.*

Out of Scope

As per the Draft RTS articles (10-18), application of Strong Customer Authentication by the PSPs can be exempt, subject to transaction monitoring, in the case of: i) fixed recurring schedule/date, AND ii) fixed amount, AND iii) fixed payee.
Therefore, the following payment request types were classified as Out of Scope for Open Banking due to non-compliance with RTS and/or PSD2.

#	Payment Request Type	Details
1	**<u>Single Deferred Payment,</u> initiated for online purchase from Retailer** ***(UC#10, UC#94)***	Merchant executes payment on dispatch of goods. Execution date not fixed but within certain time period. Not explicitly stated in CMA/PSD2, but in the spirit (API adoption would be low if not available) <u>TPP cannot execute payment in Customer Not Present (CNP) mode and without customer SCA.</u> **Why Out of Scope: Customer cannot authorise a payment without knowing the date of execution.**
2	**<u>Multiple Deferred Payment,</u> initiated for online purchase from Retailer** ***(UC#11, UC#95)***	Paying for several items upfront at checkout, from one merchant, but items delivered separately. E.g. Customer authorises payment for 3 items of £150 total on Amazon. TPP executes multiple smaller payments totalling the full authorised amount (e.g. £150) when items are dispatched (e.g. 1 x £150 or 3 x £50 or 1 x £100 + 1 x £50) <u>TPP cannot execute each payment in Customer Not Present (CNP) mode and without customer SCA.</u> **Why Out of Scope: Customer cannot authorise a payment without knowing the date of execution AND cannot authorise the total amount of payment without knowing the actual amount of each individual payment.**
3	**Open-ended payment instruction** ***(UC#96)***	Consumer/SME giving full access of payment initiation to a third party. Third party can initiate payments of various amounts to different beneficiaries as and when they wish. Would require a bi-lateral consent but outside of PSD2. <u>TPP cannot execute each payment in Customer Not Present (CNP) mode and without customer SCA.</u> **Why Out of Scope: Cannot authorise a payment without knowing the date of execution AND the amount of the payment AND the payment beneficiary.**
4	**Recurring Regular Variable Payments:** **i) without end date** **ii) with end date** ***(UC#97)***	Consumer/SME giving consent to a TPP to debit utility bill amount every month (RTS implies every execution would need SCA for variable amounts) for an unlimited or limited period. <u>TPP cannot execute each payment in Customer Not Present (CNP) mode and without customer SCA.</u> **Why Out of Scope: Cannot authorise a payment without knowing the amount of the payment.** **Exception: No SCA for payments up to EUR 30 and a cumulative amount of EUR 100 or 5 consecutive individual electronic transactions**
5	**Recurring Irregular Variable Payments:** **i) without end date** **ii) with end date** ***(UC#98)***	i) Consumer/SME giving consent to a TPP to monitor their two different bank accounts and initiate payment from one account to another as and when required based on pre-defined rule set for an unlimited or limited period to avoid overdraft charges ii) Consumer/SME giving consent to a TPP (online wallet/pre-paid service) to top up their account as and when it goes below certain value for an unlimited or limited period. <u>TPP cannot execute each payment in Customer Not Present (CNP) mode and without customer SCA.</u> **Why Out of Scope: Cannot authorise a payment without knowing the date of execution AND the amount of the payment.**

Out of Scope

The following functionality items and Use Cases have been classified as out of scope for Open Banking Payments APIs technical implementation

#	Payment Type	Details
6	**Direct debits (instruction via payee)** ***(UC#98)***	Initiate direct debit collections
	Payment Instruction confirmation ***(UC#27)***	**As a Consumer or Business, I want to** ensure that I am paying the right person when setting up a new Open Banking payment instruction (i.e. seeing the recipient name), **so that** I avoid paying the incorrect beneficiary. *Note: Open Banking flows will be playing back the details of beneficiary but not confirming they are correct. This will require Confirmation of Payee (CoP) functionality which is part of the future delivery of The New Payments Architecture for the Payments Strategy Forum.*
	Customer Refund ***(UC#66)***	**As a Retailer**, I **want to** be able easily initiate a refund directly to the customer's bank account which they had used to make the payment for the purchase, **so that** I can meet my return and refund policies with my customers
	Other functionality: • **Reversals, Returns** • **Earmarking** • **Funds check**	Reversals (funds sent to payee bank but hasn't reached payee account and payee bank returns funds) Refund (return of previously received funds from payee)
		Why Out of Scope: Not required by CMA or PSD2

Any questions?

Payment Initiation API

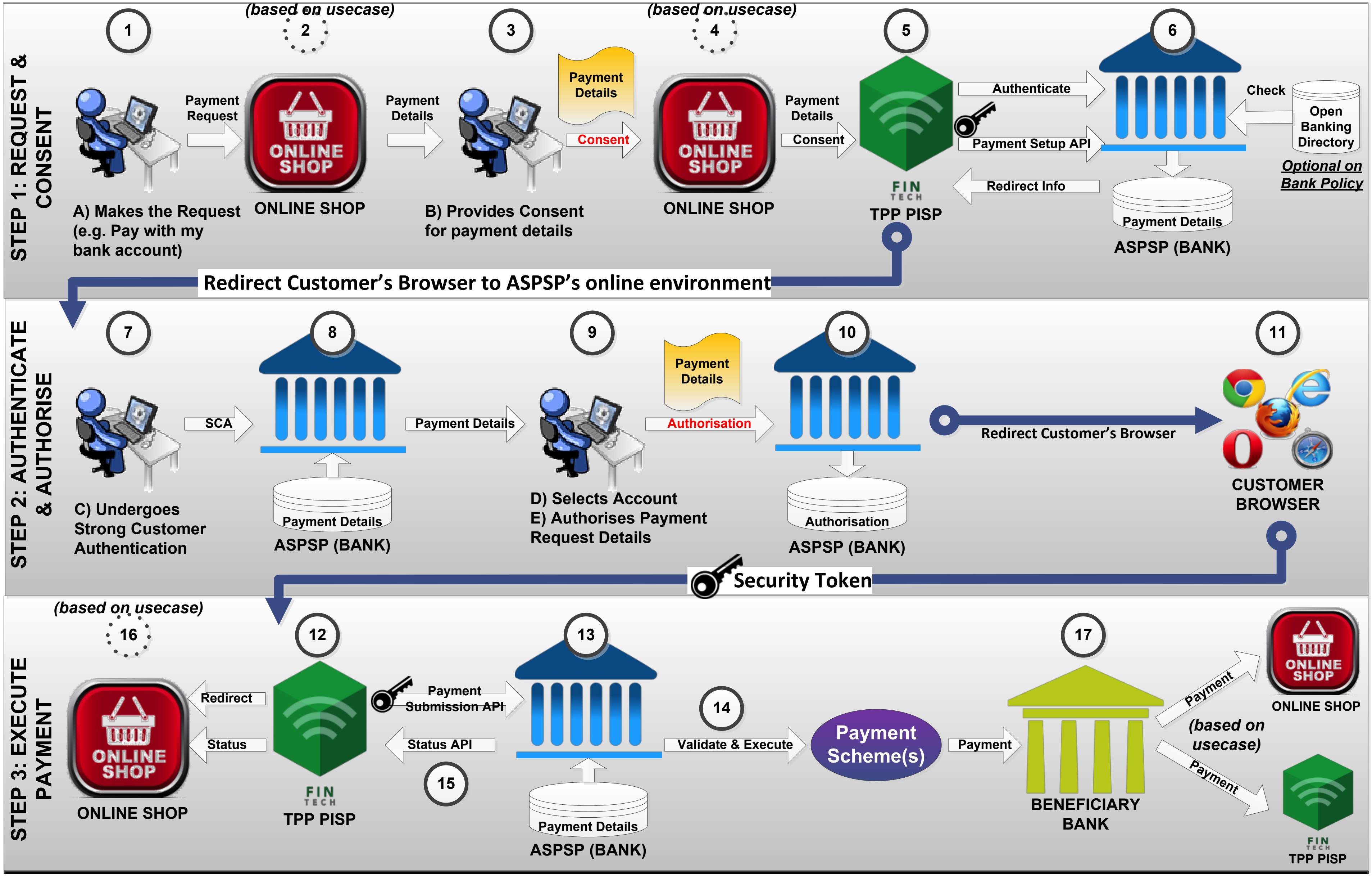
Customer Journey

Flows and Data Model

Example user journey

- Customer purchasing on Argos where Argos is the PISP
- https://projects.invisionapp.com/share/WYAOFBZ5Z#/screens

Payment Initiation API Flow



STEP 1: REQUEST & CONSENT

1
A) Makes the Request (e.g. Pay with my bank account)

Payment Request

(based on usecase)
2
ONLINE SHOP

Payment Details

3
B) Provides Consent for payment details

Payment Details
Consent

(based on usecase)
4
ONLINE SHOP

Payment Details
Consent

5
FIN TECH
TPP PISP

Authenticate
Payment Setup API
Redirect Info

6
ASPSP (BANK)
Payment Details

Check
Open Banking Directory
Optional on Bank Policy

Redirect Customer's Browser to ASPSP's online environment

STEP 2: AUTHENTICATE & AUTHORISE

7
C) Undergoes Strong Customer Authentication

SCA

8
Payment Details
ASPSP (BANK)

Payment Details

9
D) Selects Account
E) Authorises Payment Request Details

Payment Details
Authorisation

10
Authorisation
ASPSP (BANK)

Redirect Customer's Browser

11
CUSTOMER BROWSER

Security Token

STEP 3: EXECUTE PAYMENT

(based on usecase)
16
ONLINE SHOP

Redirect
Status

12
FIN TECH
TPP PISP

Payment Submission API
Status API
15

13
Payment Details
ASPSP (BANK)

14
Validate & Execute

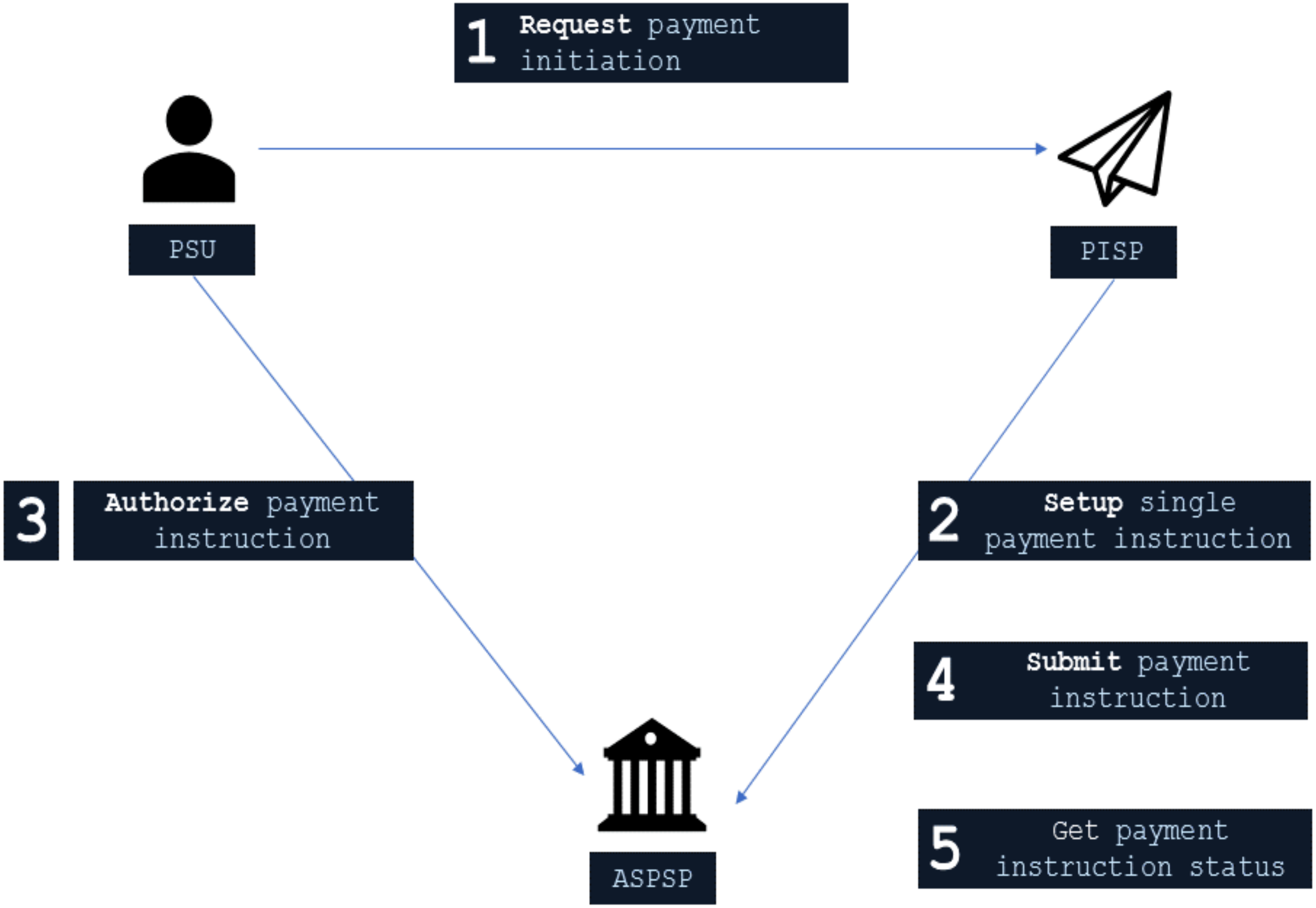
Payment Scheme(s)

Payment

17
BENEFICIARY BANK

Payment
ONLINE SHOP
(based on usecase)
Payment
FIN TECH
TPP PISP

Payment flow


1 Request payment initiation
PSU
PISP
3 Authorize payment instruction
2 Setup single payment instruction
4 Submit payment instruction
ASPSP
5 Get payment instruction status

Specifications

- Latest specifications can be found here https://openbanking.atlassian.net/wiki/x/JVIP

Any questions?

Account & Transaction APIs

Customer journey

Flows and Data Model

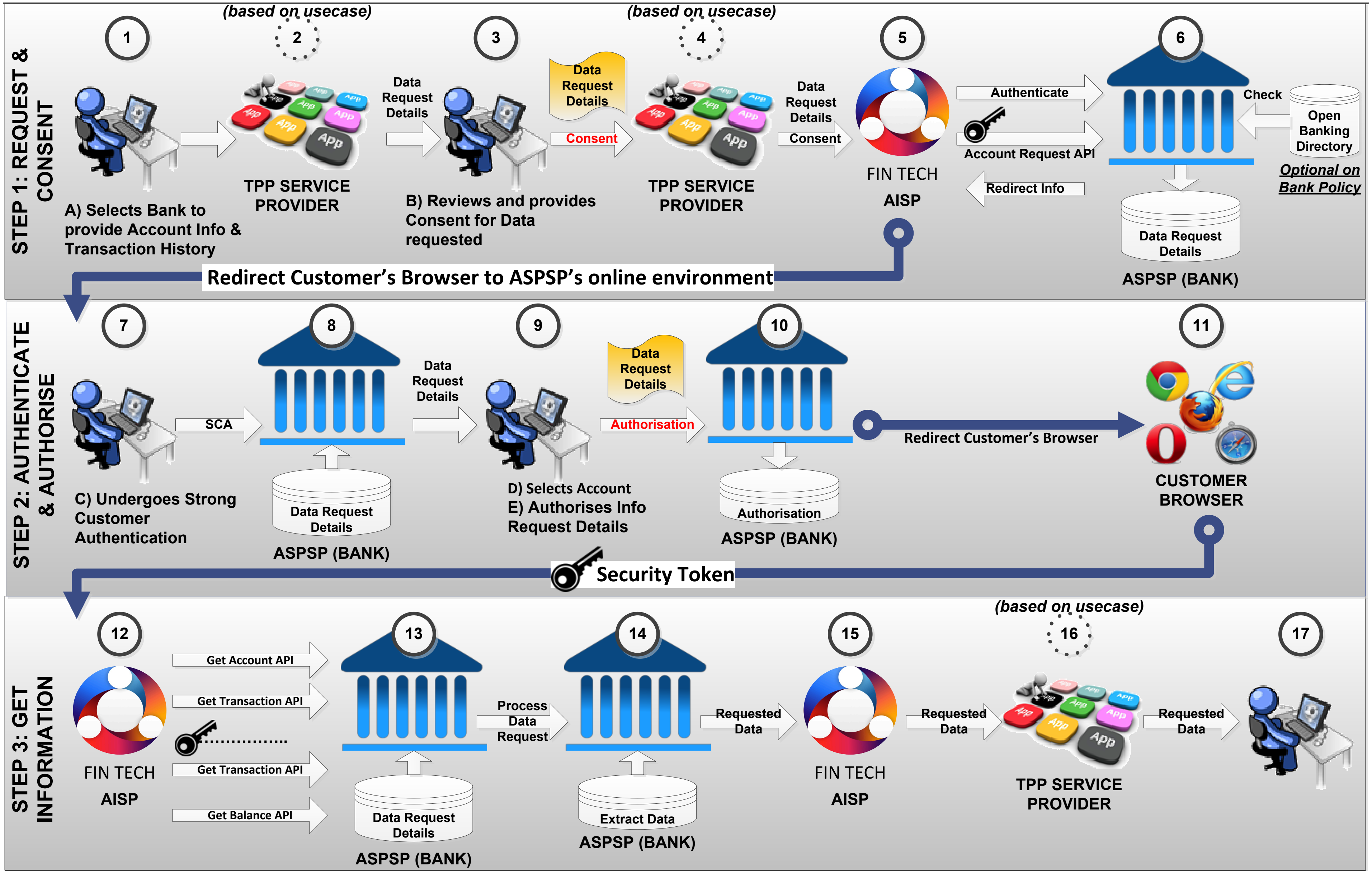
Example customer journey

Business account price comparison

- Business customer looking to switch accounts. AISP registered with OB is an aggregator.
- End service provider is not. Consent is for Open Banking data as well as for another service (Companies House Data).
- No account selection as customer has only one BCA.

https://projects.invisionapp.com/share/F7AX8CESN#/224393366_Home

Account Information API Flow



STEP 1: REQUEST & CONSENT

1 A) Selects Bank to provide Account Info & Transaction History

(based on usecase) 2 TPP SERVICE PROVIDER

Data Request Details

3 B) Reviews and provides Consent for Data requested

Data Request Details

Consent

(based on usecase) 4 TPP SERVICE PROVIDER

Data Request Details

Consent

5 FIN TECH AISP

Authenticate

Account Request API

Redirect Info

6 ASPSP (BANK)

Check

Open Banking Directory

Optional on Bank Policy

Data Request Details

Redirect Customer's Browser to ASPSP's online environment

STEP 2: AUTHENTICATE & AUTHORISE

7 C) Undergoes Strong Customer Authentication

SCA

8 ASPSP (BANK)

Data Request Details

Data Request Details

9 D) Selects Account
E) Authorises Info Request Details

Data Request Details

Authorisation

10 ASPSP (BANK)

Authorisation

Redirect Customer's Browser

11 CUSTOMER BROWSER

Security Token

STEP 3: GET INFORMATION

12 FIN TECH AISP

Get Account API

Get Transaction API

Get Transaction API

Get Balance API

13 ASPSP (BANK)

Data Request Details

Process Data Request

14 ASPSP (BANK)

Extract Data

Requested Data

15 FIN TECH AISP

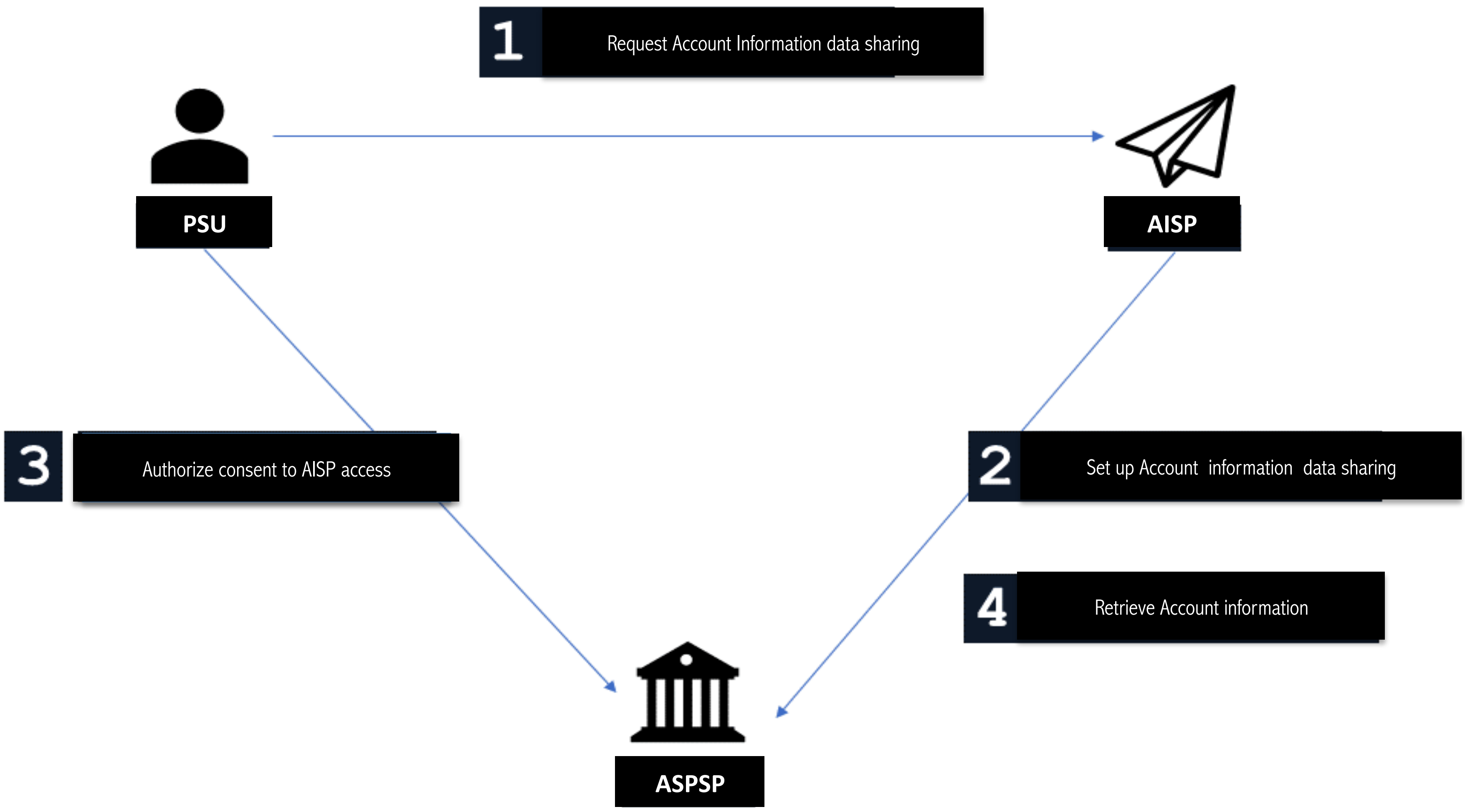
Requested Data

(based on usecase) 16 TPP SERVICE PROVIDER

Requested Data

17

Flows



1 Request Account Information data sharing

2 Set up Account information data sharing

3 Authorize consent to AISP access

4 Retrieve Account information

Specifications

- Latest specifications can be found here https://openbanking.atlassian.net/wiki/x/U_QI

Any questions?

Trust Framework

API Security Flows

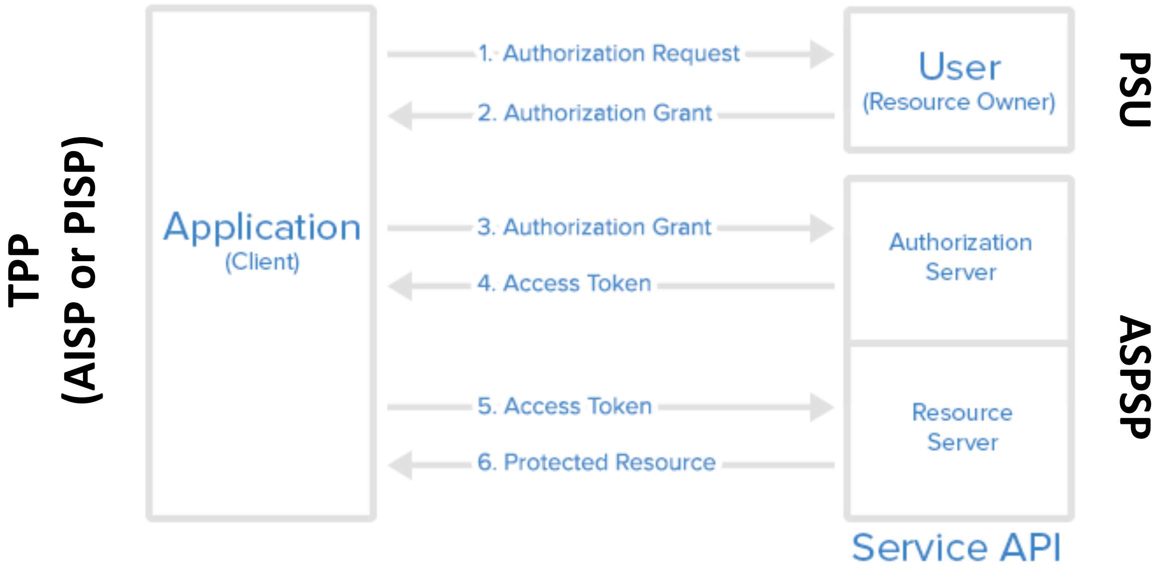
Principles

Find the most appropriate security protocol(s) for Open Banking System Participants to communicate, and to find an appropriate balance between Risk to all Participants and fixed regulatory delivery dates.

- Leverage open international standards.
- Apply appropriate separation of concerns.
- Support evolution.
- Support interoperability.
- ASPSPs supporting more appropriate and secure protocols should not be forced to downgrade.
- TPPs should not be forced to support 9 different security protocols to interact with the CMA9.

OAuth2 Functional Roles & Protocol Flow

1. **Resource Owner**: An entity capable of granting access to a protected resource. When the resource owner is a person, it is referred to as an end-user.
2. **Resource Server**: The server hosting the protected resources, capable of accepting and responding to protected resource requests using access tokens.
3. **Client**: An application making protected resource requests on behalf of the resource owner with its authorization. The term "client" does not imply any particular implementation characteristics (e.g., whether the application executes on a server, a desktop, or other devices).
4. **Authorisation Server**: The server issuing access tokens to the client after successfully authenticating the resource owner and obtaining authorization.



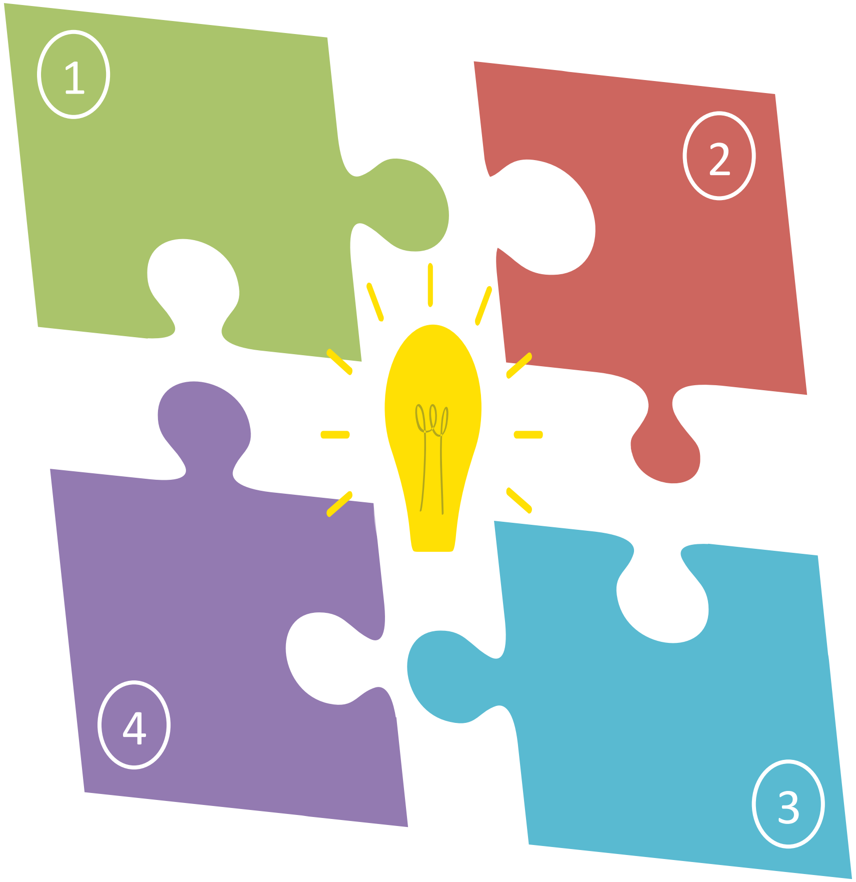
Known issues with OAuth 2.0

- All known issues with Oauth 2.0 are documented here https://tools.ietf.org/html/rfc6819/
- This RFC also defines how each of these threats can be addressed.
  - Using TLS 1.2 MA.
  - Not permitting public clients.
  - Ensuring certain parameters are always passed for certain grant types.
- The OIDC profile we are defining and adopting as a standard addresses all of these threats and also provides several additional benefits, in particular defining how the standards should be implemented.

Trust Framework Technologies

TLS 1.2 – encryption and Transport Security Layer

Industry standard for securing, encrypting and optionally authenticating communications and actors exchanging data.



OAUTH2 – industry authorization protocol

Authorization protocol. An API receives "access tokens" representing a client and the user that client may be operating on behalf of, rather than actual secrets like API keys or passwords.

OpenID Connect – industry identity protocol, enhanced OAuth 2.0 security profile.

Authorization protocol enhancements coupled with the Internet Standard Identity Token. Closes known vulnerabilities; supplies a mechanism to communicate information about the access token or the identity for whom an access token has been issued. (Non Repudiation achieved for all parties).

JSON web token security suite

Standardised and industry wide payload definitions supporting encryption and signing. Supports validation, authentication and non repudiation of individual message payloads.

Design Rationale

Two-Step Approach for APIs

The APIs use a two-step setup-then-submission approach because of the functional requirement to be able to authorize a transaction that can be submitted on some date, or dates, in the future.

OAuth 2.0 Family for Security, Authorization, and Privacy

The APIs need a consistent, powerful, and standard mechanism. OAuth is widely supported for API security and to curtail the sharing of user passwords. OIDC is built on OAuth, allows access tokens to be accompanied by an identity, and is increasingly advocated as a powerful OAuth security layer in its own right.

Consent Compliance and Authorization Grant

PSD2 says the TPP must explain to the PSU the purpose of what they want to do and gather consent for it. But in OAuth the authorization server gathers authorization from the user. The two-step paradigm provides value for compliance, and because the consent details are finer in grain than an OAuth scope as typically designed and implemented, there is business value in having the TPP “broker” the consent details.

Securing Consent Details

Given the functional requirement for the two-step approach and the consent compliance requirement, the OIDC signed request object by value (signed JWT, possibly to be encrypted in future) is the most robust method for security and efficiency. It has IT, business trust, and standardization benefits.

Benefits of OpenID Connect

- Mitigation of security risks in OAuth 2.0 implementation
- PISP scenarios become much more supportable
  - **Discovery attribute is there to say how Third Parties should interact with the ASPSP.**
  - JWT can be sent indicating that the id_token should come back with a unique transaction / intent reference.
  - **Signed assertion** with a TXID comes back as a result. Supporting Non Repudiation Use cases.
- AISP scenarios become much more supportable
  - Id_tokens can be configured to provide “Authorization” data and can cater for authorization elements that can change.
- Additional Authorization Request features are available
  - claims_locales - specify a preferred language.
- OpenID Certification performed by the OpenID Foundation, TPP's and ASPSP's can be certified by the OpenID Certification for compliance.
- The OpenID Foundation provides a test framework for OpenID compliance, this could be forked and potentially collaborated on by all OB participants. New security issues could be added to the test suite and TTP's request to re-test to ensure compliance and achieve security.
- **OpenID Connect Financial API profile alignment does not compel or require ASPSPs to provide Federated Identity Services nor Identity Attestations About ASPSPs PSUs.**

1. Pass Intent ID via Request URI + OAuth Only

Vulnerabilities:

No way to definitively tie Authorization Code returned to an Authorization Request. Raises MIM vulnerability where a TPP compromised client could be used to obtain access to account information from another user.

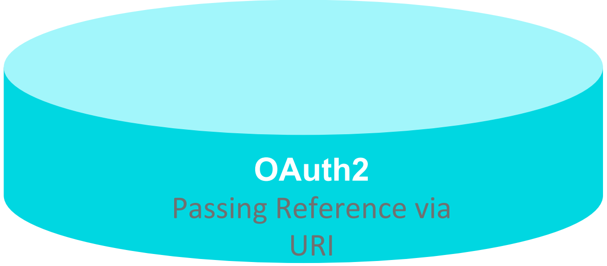
No definitive way to tie Authorization Code to initial Intent Request ID. **Mitigation**: Will potentially require the creation of an additional API to allow TPPs to confirm mapping between Authorization Grant and IntentID. **Mitigation 2:** Execution APIs will always require reconfirmation of Intent Details to allow Access Token to Intent Mapping.

No discovery capability for TPPs increasing onboarding complexity. **Mitigation**: Create a well-known/configuration endpoint as per OpenIDC specification.

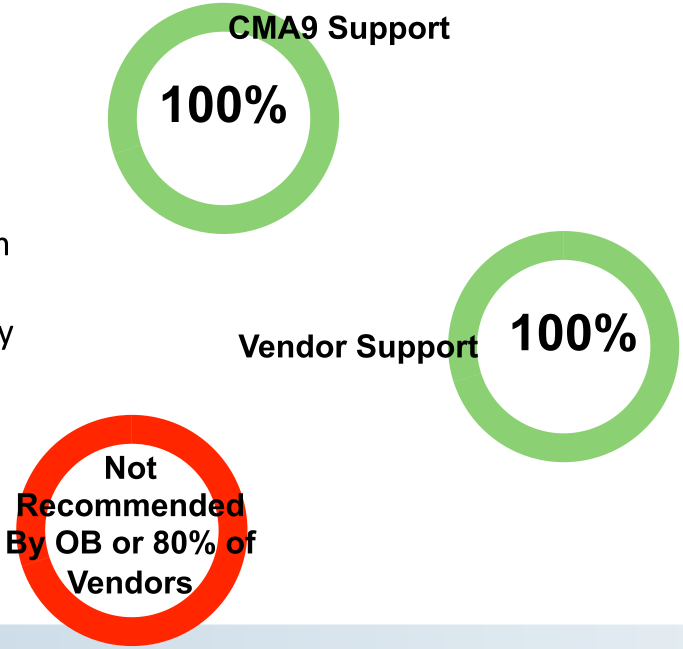
No conformance testing to security profile mandated for providers or available as reference for TPPs.

IntentID can not be signed in a way that would 100% prevent a malicious PSU from obtaining information on Intent on a random PSU via brute force. **Mitigation**: Intent ID should be appropriately randomly generated and IntentID's appropriately namespaced.

General: OAuth explicitly notes identity-based attacks against clients, recommending use of identity protocols for PSU protection. Does not prevent payee data leakage to PSU.



OAuth2
Passing Reference via URI



2. Pass Intent ID via Request URI + OAuth + OpenIDC

Vulnerabilities:

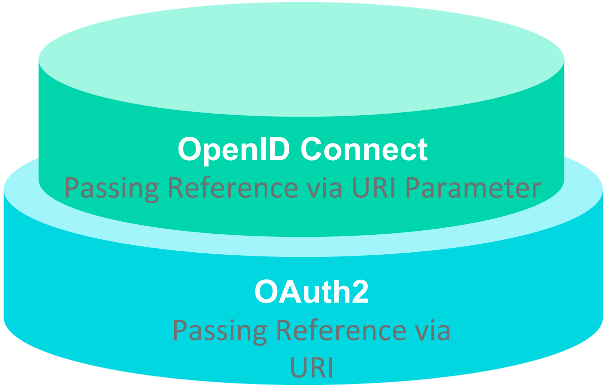
No way to definitively tie Authorization Code returned to an Authorization Request. Raises MIM vulnerability where a TPP compromised client could be used to obtain access to account information from another user.

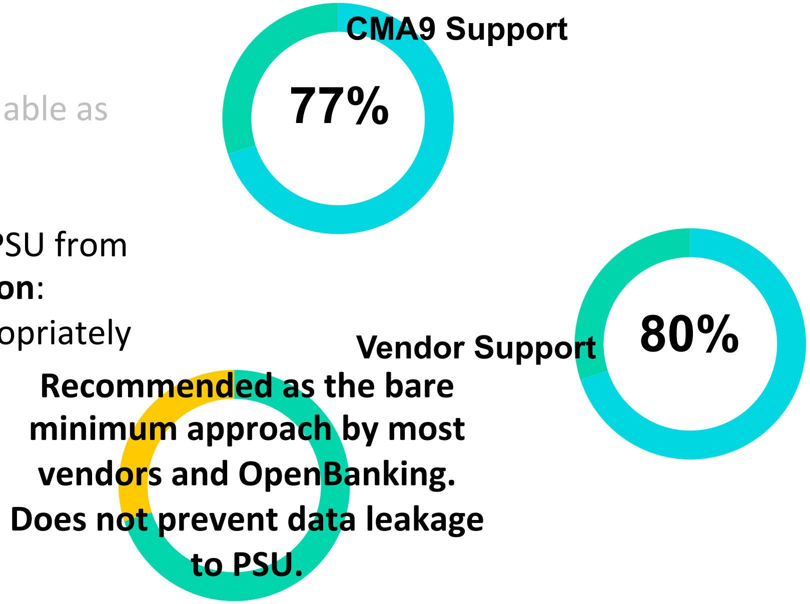
No definitive way to tie Authorization Code to initial Intent Request ID. **Mitigation**: Will potentially require the creation of an additional API to allow TPPs to confirm mapping between Authorization Grant and IntentID. **Mitigation 2:** Execution APIs will always require reconfirmation of Intent Details to allow Access Token to Intent Mapping.

No discovery capability for TPPs increasing onboarding complexity. **Mitigation**: Create a well-known/configuration endpoint as per OpenIDC specification.

No conformance testing to security profile mandated for providers or available as reference for TPPs.

IntentID can not be signed in a way that would 100% prevent a malicious PSU from obtaining information on Intent on a random PSU via brute force. **Mitigation**: Intent ID should be appropriately randomly generated and IntentID's appropriately namespaced.


OpenID Connect
Passing Reference via URI Parameter
OAuth2
Passing Reference via URI



Recommended as the bare minimum approach by most vendors and OpenBanking. Does not prevent data leakage to PSU.

3. Pass Intent ID via Signed JWT + OAuth + OpenIDC - TARGET

Vulnerabilities:

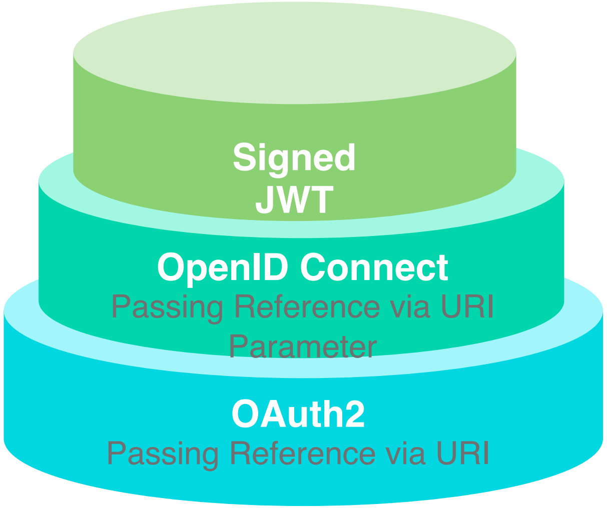
No way to definitively tie Authorization Code returned to an Authorization Request. Raises MIM vulnerability where a TPP compromised client could be used to obtain access to account information from another user.

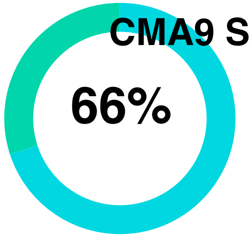
No definitive way to tie Authorization Code to initial Intent Request ID. **Mitigation**: Will potentially require the creation of an additional API to allow TPPs to confirm mapping between Authorization Grant and IntentID. **Mitigation 2:** Execution APIs will always require reconfirmation of Intent Details to allow Access Token to Intent Mapping.

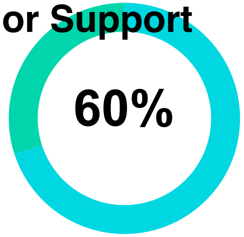
No discovery capability for TPPs increasing onboarding complexity. **Mitigation**: Create a well-known/configuration endpoint as per OpenIDC specification.

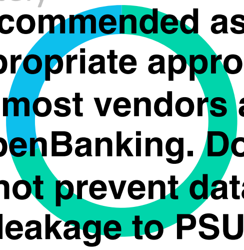
No conformance testing to security profile mandated for providers or available as reference for TPPs.

IntentID can not be signed in a way that would 100% prevent a malicious PSU from obtaining information on Intent on a random PSU via brute force. **Mitigation**: Intent ID should be appropriately randomly generated and IntentID's appropriately namespaced.









Recommended as an appropriate approach by most vendors and OpenBanking. Does not prevent data leakage to PSU.

4. Pass Intent ID via Signed & Encrypted JWT + OAuth + OpenIDC

Vulnerabilities:

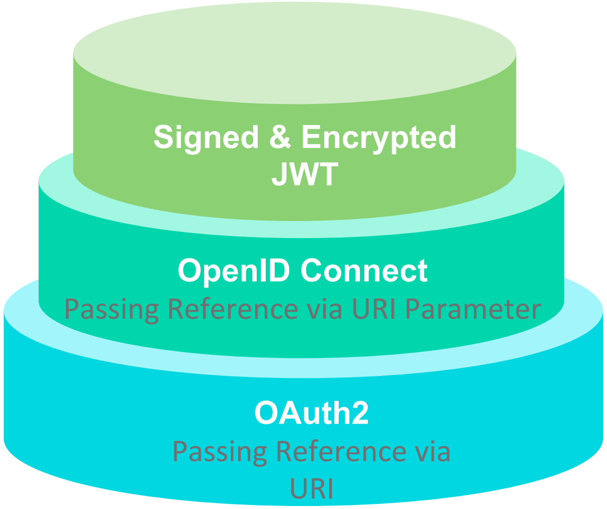
No way to definitively tie Authorization Code returned to an Authorization Request. Raises MIM vulnerability where a TPP compromised client could be used to obtain access to account information from another user.

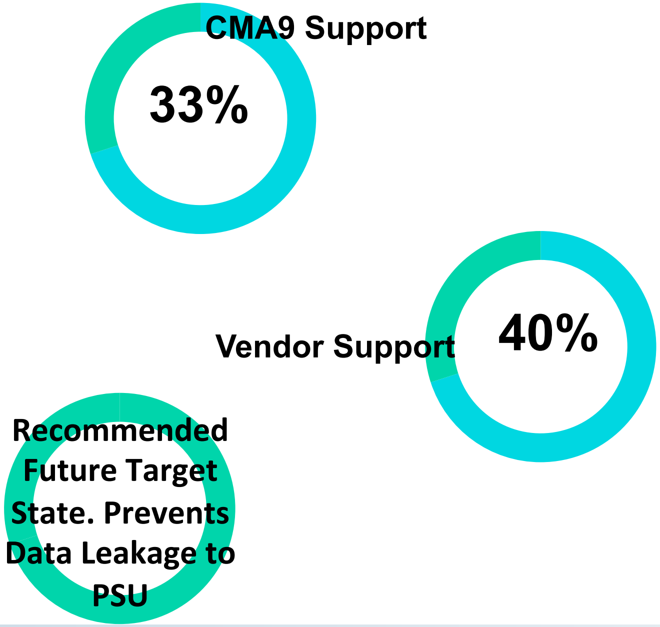
No definitive way to tie Authorization Code to initial Intent Request ID. **Mitigation**: Will potentially require the creation of an additional API to allow TPPs to confirm mapping between Authorization Grant and IntentID. **Mitigation 2:** Execution APIs will always require reconfirmation of Intent Details to allow Access Token to Intent Mapping.

No discovery capability for TPPs increasing onboarding complexity. **Mitigation**: Create a well-known/configuration endpoint as per OpenIDC specification.

No conformance testing to security profile mandated for providers or available as reference for TPPs.

IntentID can not be signed in a way that would 100% prevent a malicious PSU from obtaining information on Intent on a random PSU via brute force. **Mitigation**: Intent ID should be appropriately randomly generated and IntentID's appropriately namespaced.


Signed & Encrypted JWT
OpenID Connect
Passing Reference via URI Parameter
OAuth2
Passing Reference via URI



Security Framework – Options Overview

Option	Description	OB Security Assessment
1	Oauth 2.0: Passing Reference via URI Parameter	**Least Secure** **Not Recommended Target State**
2	OpenID Connect + Oauth 2.0: Passing Reference via URI Parameter	**Some Vulnerabilities remain.** **Not Recommended Target State**
3	OpenID Connect + Oauth 2.0: Signed JWT	**Limited Vulnerabilities** **Recommended**
4	OpenID Foundation FAPI Working Group Read-Write Specifications	**Strategic Alignment With Global International Standards – Target State on Ratification by OID Foundation.**

Summary

- Communications between PSU and TPP, and between PSU and ASPSP should be secured as defined by RTS SCA, however the exact methods are in the competitive space and not covered by the OB standard.
- The OB standard specifically covers the protocols for securing all ‘back channel’ communications between TPPs and ASPSPs as follows:
  - All communications should be Server to Server, using TLS 1.2 MA, as this is the most up to date standard for Transport Layer Security.
  - OAuth 2.0 should be used as the authorisation framework for all Open Banking Read/Write APIs, as this is the most widely adopted and supported open standard which enables Internet users to authorise websites or applications to access their information without handing over their passwords.
  - Signed JWTs should be used for payloads to support validation and non-repudiation.
  - OpenID Connect should be used to mitigate known vulnerabilities in OAuth 2.0, and provide well known end points to enable ‘discovery’.
  - OB will develop an OB OIDC profile to enable conformance testing to enable TPPs to ‘self-test’ their applications meet the standard.
- This approach is an enabler, not a barrier, and has the following benefits:
  - Sets the bar high to protect all parties (meeting CMA remedies and conforms to RTS).
  - Is supported by most major vendors (including CMA9 vendors).
  - Will be easy for all competent TPPs to adopt and implement.
- These standards should evolve over time:
  - To provide even greater protection, for example to include encryption of signed JWT once supported by more vendors.
  - Align with ISO/FAPI.

Decision

Option 3 was approved by TDA in Apr 2017

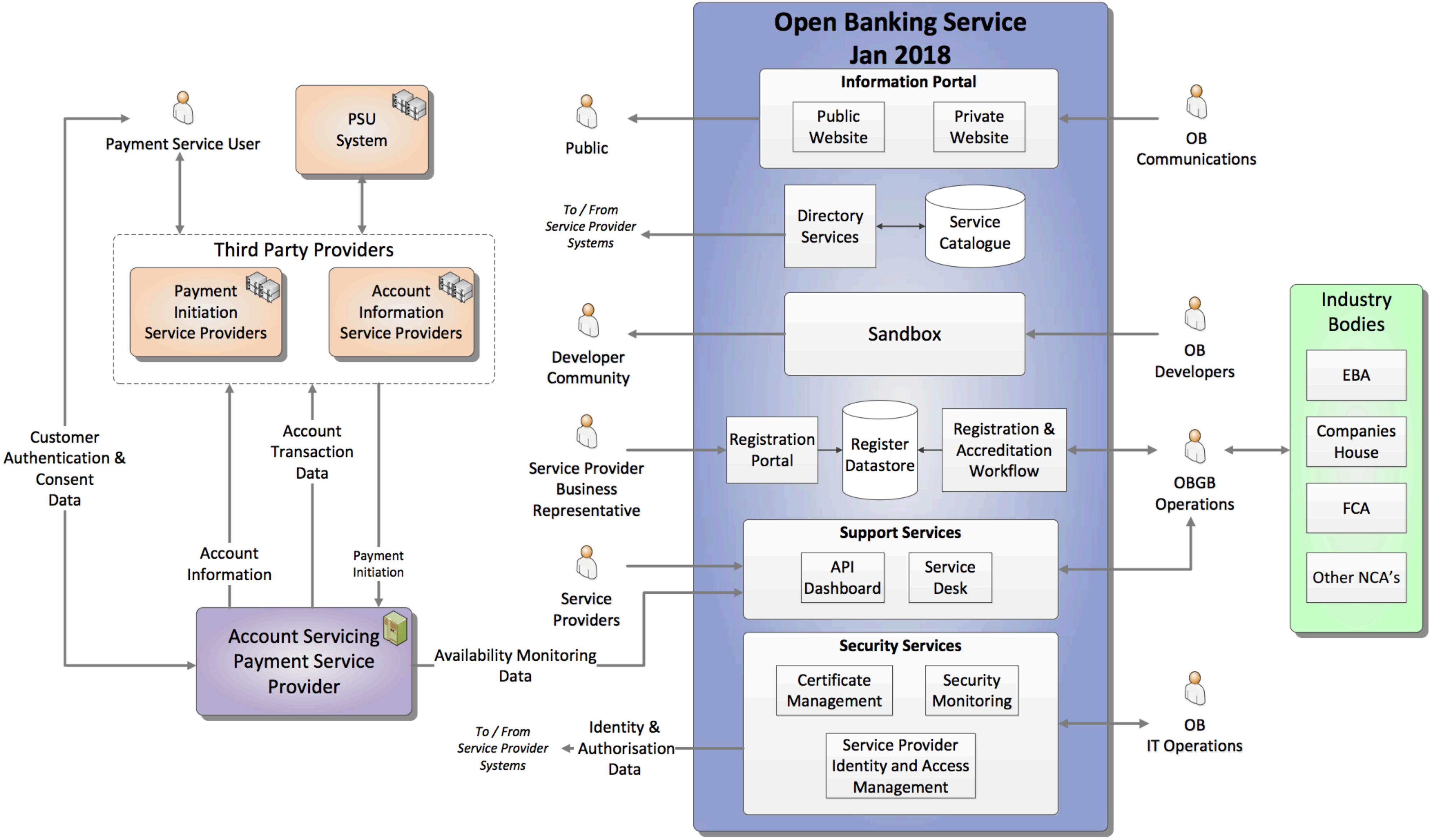
More detailed specifications can be found at https://openbanking.atlassian.net/wiki/x/av4j

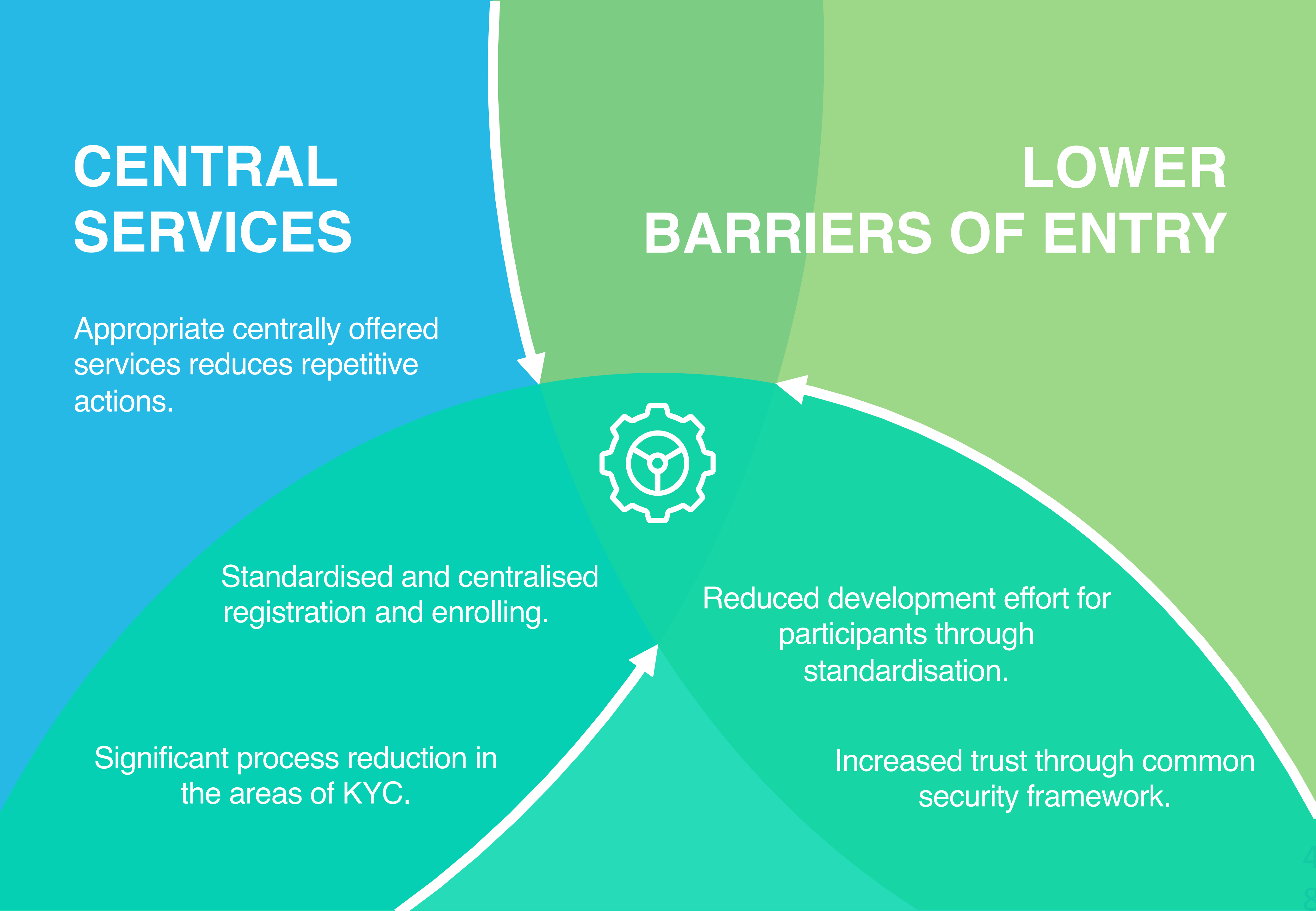
Any questions?

Open Banking Directory

Summary of functionality and design

Original solution design





CENTRAL SERVICES

Appropriate centrally offered services reduces repetitive actions.

LOWER BARRIERS OF ENTRY

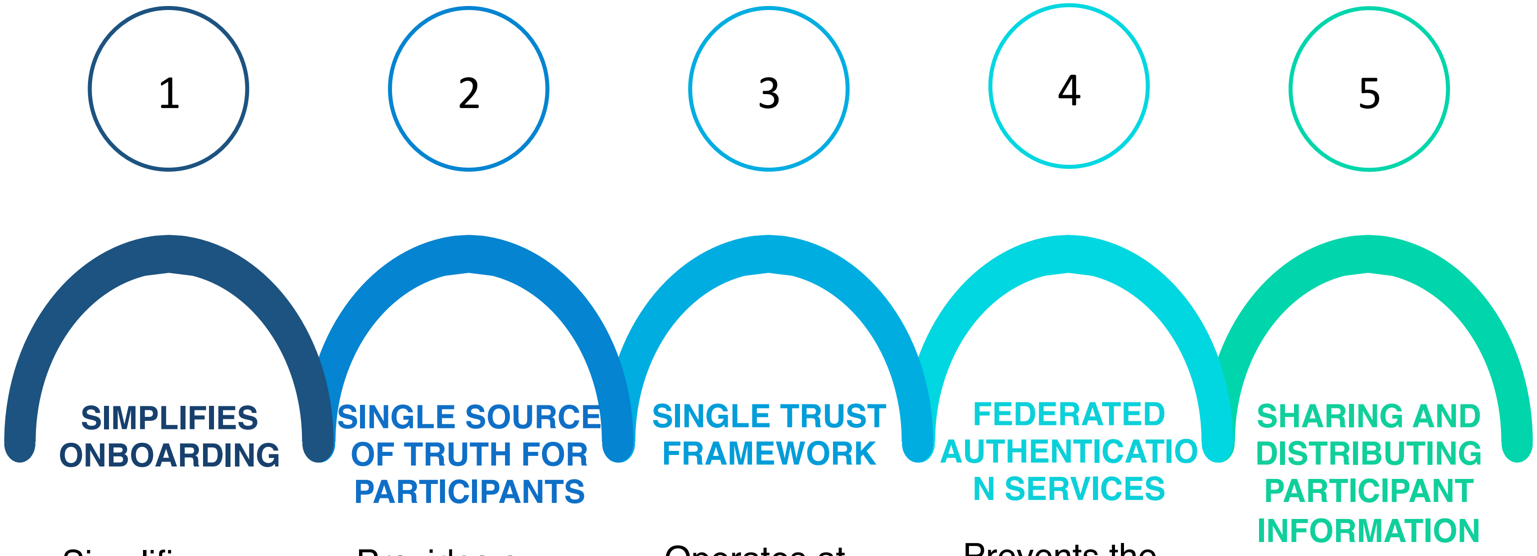
Standardised and centralised registration and enrolling.

Reduced development effort for participants through standardisation.

Significant process reduction in the areas of KYC.

Increased trust through common security framework.

Why an Open Banking Directory?



1

SIMPLIFIES ONBOARDING

Simplifies registration for all participants.

Simplifies service discovery.

Centralises KYC of TPPs reducing economic waste.

2

SINGLE SOURCE OF TRUTH FOR PARTICIPANTS

Provides a single API that ASPSPs can use to determine the roles and status of TPPs.

3

SINGLE TRUST FRAMEWORK

Operates at multiple levels (layered security model).

Protects all participants.

Centralised PKI reduces certificate management.

4

FEDERATED AUTHENTICATIO N SERVICES

Prevents the proliferation of credentials for TPPs and Human Agents.

A single authentication mechanism for ASPSPs.

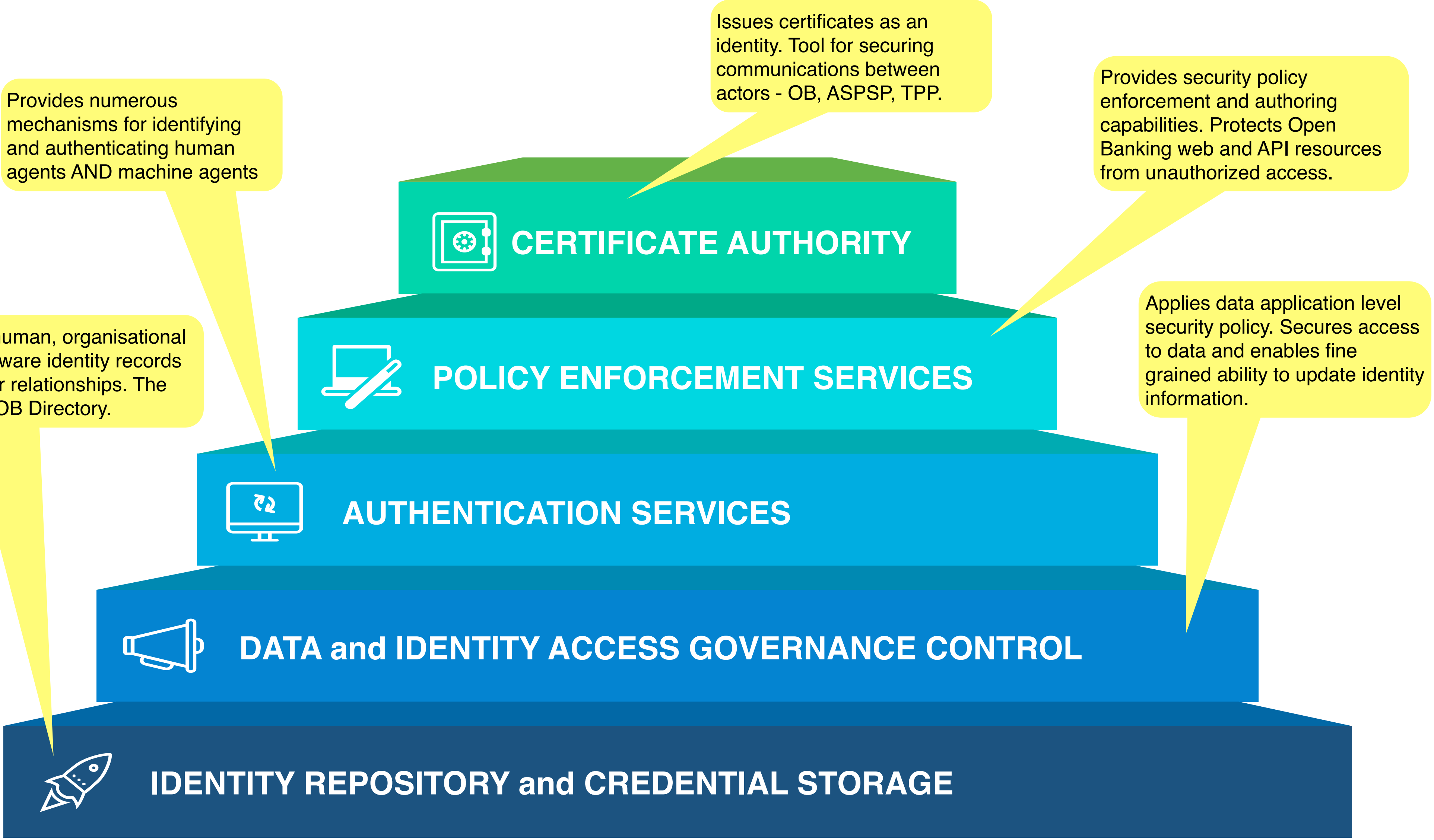
Lowers risk of credential compromise

5

SHARING AND DISTRIBUTING PARTICIPANT INFORMATION

Rapid sharing of new members, FCA revocation, retirement of members or their apps.

Open Banking Directory Services


Issues certificates as an identity. Tool for securing communications between actors - OB, ASPSP, TPP.
Provides numerous mechanisms for identifying and authenticating human agents AND machine agents
Provides security policy enforcement and authoring capabilities. Protects Open Banking web and API resources from unauthorized access.
Stores human, organisational and software identity records and their relationships. The core of OB Directory.
Applies data application level security policy. Secures access to data and enables fine grained ability to update identity information.
CERTIFICATE AUTHORITY
POLICY ENFORCEMENT SERVICES
AUTHENTICATION SERVICES
DATA and IDENTITY ACCESS GOVERNANCE CONTROL
IDENTITY REPOSITORY and CREDENTIAL STORAGE

Thank you.
Any Questions?