API standards for data-sharing (account aggregator)

Report submitted by Consultative Group on Innovation and the Digital Economy

October 2022

BIS Representative Office for the Americas

This publication is available on the BIS website (www.bis.org).

ISBN 978-92-9259-606-4 (online)

Table of contents

Foreword

This report is the third part of a trilogy on enabling open finance through APIs under the auspices of the BIS Consultative Group on Innovation and the Digital Economy (CGIDE).

The CGIDE was launched in February 2020 to meet demand from Bank for International Settlements (BIS) central bank members in the Americas for greater cooperation in technological innovation and the digital economy. The group provides a forum in which senior central bank officials cooperate and it has the following objectives:

(i) Analysing and developing public technological infrastructures geared towards tackling common shortcomings in all participating jurisdictions.

(ii) Promoting an environment suitable to open banking, potentially through the development of key application programming interfaces (APIs).

(iii) Analysing the implications of these public technological infrastructures in terms of market structure and regulatory implications.

The first CGIDE report, *Enabling open finance through APIs* was published in December 2020 and explored technical issues surrounding the development of an identification and authentication API that could be used to implement privately and publicly administered open finance solutions with seamless scalability. The second report, *Enabling open finance through APIs: report on payment initiation*, analysed two alternative API architectures for payment initiation, both based on an authentication app for mobile phones developed and maintained by a central validator. The first one involves the use of a standalone app to authenticate customers for each transaction, managed by the central validator. The second one involves embedded functionality that allows customers to use their third-party app after the completion of initial onboarding with the central validator's authentication application.

This is the third report of the trilogy and deals with data-sharing models. The report was prepared by a technical task force of central bank experts that participate in the CGIDE. It aims to serve as a useful general reference for central banks seeking to develop their own data-sharing initiatives related to account aggregation in the context of open finance. Comments are welcome and should be addressed to CGIDEreport@bis.org.

Milton Vega	Miguel Diaz	Alexandre Tombini
Chair Technical Task Force Central Reserve Bank of Peru	Chair CGIDE Head BIS Innovation Hub	BIS Chief Representative for the Americas

Executive summary

Application programming interfaces (APIs) are a critical part of open finance, and they are particularly important for enabling the secure exchange of information between different parties. Yet to achieve this, a certain level of standardisation is necessary, as well as agreements on the technical model which enables data to be shared. This report dives into these technical issues. The objective is to provide central banks with important elements according to which the introduction of data-sharing infrastructures in their economies can be evaluated.

Data-sharing can be defined as the provision of data by a data holder or data provider to a third party or data consumer with the consent of the data owner. It is one of the main pillars of open banking initiatives and incorporates a collection of practices, technologies, architecture, cultural elements and legal frameworks that relate to the exchange of digital information between individuals or organisations. Introducing explicit data-sharing models has several benefits. It can promote transparency, competition and market entry, and contribute to reciprocity and cooperation in the financial ecosystem. It can improve the performance and value of services by combining data from diverse sources. Finally, it can enable better decision-making, deliver better products and empower citizen data ownership.

Account aggregators (AAs) are an intermediate technological platform responsible for managing and transferring data flows between data providers and data consumers. AAs are an important mechanism for the implementation of data-sharing. One of their functions is to develop interoperability between participants. But AAs are only intermediaries and cannot store the data or redirect it to unauthorised entities. An important feature of AAs is how they develop mechanisms to gain consent for data flows from and for the end users.

This report presents three types of data-sharing model: centralised, decentralised and trust ecosystem. In a centralised model, an AA collects the data. In a decentralised model, participating members agree to share their data with other participants individually. The trust framework is hybrid; it is decentralised for data-sharing and centralised for identity management. It integrates with a trusted third party instead of an aggregator. This last model requires operators to correctly establish the registration process for participants, as well as to ensure security in communications and agree on a standard for the exchange of information.

APIs are important to share information in the data-sharing models. To develop them, an authority must evaluate their functionalities, access levels, standards, protocols and security mechanisms. The three main access levels for APIs are public, private and partner. Access levels depend on the regulatory stance and on how the authority implements data-sharing. Public APIs are generally open and accessible. Private or internal APIs are available only to specific service consumers. Partner APIs are available for external access for pre-defined service consumers, usually from partner organisations.

APIs' security mechanisms must be robust and must keep data safe. The first process, authentication, identifies if the client and users are who they claim to be. The second process is access control, which limits API consumers' actions after correct authentication. The third is encryption. Encrypted tokens store vital information such as the username and password. These tokens expire after a certain time, strengthening the API's security. Finally, audit logging in a registry stores actions and calls made to the API. Some recommended standards for the implementation of security mechanisms for APIs are JSON (JavaScript Object Notation) Web Token, OAuth 2.0, OpenID Connect and FAPI (financial-grade API).

Central banks have a common interest in implementing data-sharing, with the aim of increasing efficiency and promoting competition in their ecosystems. The main challenges are coordination among participants, standardisation and technological infrastructure. Cooperative technical work can help to mitigate these challenges.

Introduction

This report, *API standards for data-sharing (account aggregator)*, is part of the work of the Consultative Group on Innovation and the Digital Economy (CGIDE) on open finance. The CGIDE was launched in February 2020 to meet demand from Bank for International Settlements (BIS) central bank members in the Americas for greater cooperation in the areas of technological innovation and the digital economy. This work is part of a series of projects on enabling open finance through application programming interfaces (APIs). Previous reports developed a technical flow process, outlined the characteristics of the information technology infrastructure and proposed API architecture.

In this report, the CGIDE Technical Task Force (TTF) explores multiple data-sharing models: centralised, decentralised, hybrid, trust ecosystem, regulation-driven and market-driven. Other key points developed in this report are related to data serialisation, API content, metadata, API protocols, messaging formats and access to services. This document complements the previous reports and discusses technical issues concerning APIs that could contribute to implementing privately and publicly administered open finance and data-sharing ecosystems.

As in previous reports, the work of the CGIDE TTF does not review all possible alternatives for data-sharing through APIs. Instead, this document should serve as a general reference for individual countries seeking to develop their own data-sharing projects. The report does not recommend one solution over another.

As initial work for this report, the CGIDE TTF conducted a survey, the respondents were eight central banks in the Americas (those of Argentina, Brazil, Canada, Chile, Colombia, Mexico, Peru and the United States). The aim was to gather relevant preliminary information on awareness and considerations related to API standards for secure and effective sharing of customer data between financial institutions, fintech companies and certified third parties. The results of this survey, presented in Annex A, served as a basis for discussions about the technical requirements for implementation subsequently proposed by the technical task force.

The remainder of this report is organised as follows: Section 1 defines data-sharing basics and related concepts such as API, data providers, data consumers, consent architecture and account aggregator. Section 2 describes a data-sharing implementation process for central banking purposes. Sections 3 and 4 look into associated models, interactions and data flows. Section 5 presents technological considerations for API implementations such as design patterns, protocols, technical standards and a demo. Finally, the conclusions focus on the challenges and next steps for the CGIDE TTF.

1 Background

1.1 Definitions

1.1.1 Data-sharing

This report defines data-sharing as the provision of data by a data holder to a third party with the consent of the data owner. Data-sharing also includes the reuse of data based on commercial and non-commercial data-sharing agreements. Data-sharing incorporates a collection of practices, technologies, architecture, cultural elements and legal frameworks that relate to digital transactions of any kind of information sent between individuals or organisations. It is worth noting that the data-sharing concept is not only about the data but also about the process involved in exchanging data (SCDS (2022)).

Data-sharing is one of the main pillars of open banking initiatives which are emerging in financial services. Innovations include the involvement of third-party providers, which facilitate access to banking records with the user's consent, also known as payment service providers. Data-sharing promotes transparency in a digital society and supports high levels of reciprocity and cooperation within the financial ecosystem.

Account aggregators are important mechanisms for the implementation of data-sharing in an open banking scheme. In most of the models described below, an account centraliser or aggregator is a point of concentration of information flows, suitably standardised and regulated. By contrast, in a decentralised model, intermediate onboarding or authentication mechanisms are enough. There are several benefits of data-sharing such as promoting a transparent digital society, achieving reciprocity and cooperation in the financial ecosystem, combining data from diverse sources to improve the performance and value of services, enabling better decision making, delivering better products and empowering data ownership by citizens.

1.1.2 API content

An application programming interface (API) is a set of functions used by a software program to provide an interface that allows other consumer programs (external parties) to connect to and interact with the software program. An API may contain:

(i) communication protocols;

(ii) data exchange requirements;

(iii) access and consumer policies; and

(iv) integrity and confidential management.

1.1.3 Data serialisation

Data serialisation is the process of translating one data format to another. In this process, data are serialised, but do not change, and the source and destination still have access to the same data. Software programs that exchange data but store or represent it differently require data serialisation. This mechanism is present within a data-sharing ecosystem based on API implementations.

1.1.4 API dimensions

The construction of an API for data-sharing requires a series of considerations or characteristics associated with the implementation of the API itself. These quality attributes or dimensions are important in terms of API design and construction, thereby promoting the success of the API in the context of open banking (Zachariadis (2020)). Relevant considerations or characteristics include:

- **API accesibility**. Describes the openness of the API. Available model choices are public, private or partner. This dimension requires the definition and establishishment of an onboarding process.
- **API functionality**. Establishes the granularity, categories, functionalities and scope of the service. This dimension requires discussion and defininition of read-only and transactional APIs.
- **API usage**. Evaluates and measures the bandwidth, resilience, concurrency, scalability and sizing of the infrastructure before implementing data-sharing solutions.

- **Open APIs** are "An interface that provides a means of accessing data based on a public standard. Also known as external or public API."[1] Central banks or financial authorities must define open standards ie API standards, a message format, security policies based on standards and others.
- **Alternative APIs**. Financial insitutions do not hold data on unbanked citizens. Accordingly, a financially inclusive approach should consider including complementary/alternative data such as data sourced from social networks, sensors, the internet of things (IoT) and mobile technologies among others. This would help to overcome the challenges associated with including unbanked citizens.

1.1.5 API standards

International and industry-accepted standards are necessary to implement API-based data-sharing solutions. The most common standards are the following:

- **OpenAPI:** defines a standard for describing resource-oriented and REST-based APIs. The standard was originally based on the Swagger specification and is currently developed by the OpenAPI Initiative.[2]
- **W3C:** prominent industry standards are XML, XML Schema, XQuery, XML Encryption, XML Signature, XPath, XSLT, WSDL, SOAP, WS-Addressing and WS-Policy.
- **OASIS:** prominent deliverables are WS-BPEL, WS-Security, UDDI, ebXML and SAML.
- **IETF:** prominent deliverables are HTTP, URI Template, JSON, JSON Schema and JSON Pointer.
- **Open Container Initiative:** prominent deliverables are Runtime Specification and Image Specification (based on Docker).

1.2 Messaging data formats and data models

Service consumers interact with web-based services through the exchange of messages that use industry data-format languages. The two most common data formats are Extensible Markup Language (XML) and JavaScript Object Notation (JSON).

Schema definition languages describe the vocabulary, structure and data types of message contents. Many schemas can represent common organisational documents such as invoices and budgets. A schema language essentially provides a means of expressing the definition of a data model for the message vocabulary in a manner that enables validation of the message contents against a schema definition during runtime.

XML Schema Definition (XSD) can define the structure of messages formatted in XML. JSON Schema can define data models for messages exchanged in JSON. XML messaging is relevant to both Simple Object Access Protocol (SOAP)-based web services and Representational State Transfer (REST) services. JSON messaging is primarily relevant for REST services (Deepak (2020)).

1.3 Data providers

End users' financial information is stored in banks and with insurance companies, mutual funds, stockbrokers and even government agencies. These external repositories of personal financial data are data providers (DPs). DPs have data scattered in several storage managers, which are accessible through location mechanisms (ie URL (uniform resource locator), DNS (domain name system), IP (internet protocol)

[1] BCBS (2019).

[2] For more details, see the OpenAPI Initiative site.

addresses etc). In most cases, users are not aware of the ownership they have over their own data and their associated rights.

Each DP has its own standard and there is usually no consensus. Hence, a relevant starting point for any potential data-sharing or open banking initiative is to standardise and harmonise data providers' repositories and associated services.

1.4 Data consumers

Data consumers (DCs) are any type of entity that would like access to end users' financial data. For example, loan fintechs, personal finance managers, advisory bots, banks and other financial organisations constantly require access to accurate granular data on their current and potential clients. They aim to provide customised services to their clients.

Data consumers would use services offered by data providers to access clients' data. The delivery of value is a constant driver for data consumers, and accessing client data hosted by external servers is required in order to offer suitable financial products.

1.5 Consent architecture

The health sector has extensively developed its consent architecture. It allows patients to consent to access to personal medical data. In this way, medical personnel can access patients' data at critical times and read this information under restricted authorisations (Bergmann et al (2007)).

In the context of open banking, it is a challenge for banks to share sensitive customer data in a consent architecture. Instead, third-party APIs can serve as channels for the secure sharing of data and promote trust between participants.

A consent scheme consists of:[3]

- **Consent:** a user interface displays a description of the data that the DC requires. It also displays the period of time for which the owner of the data grants consent.
- **Authentication:** participating banks are responsible for security and authentication mechanisms. Credentials of banks and third-party APIs must be autonomous.
- **Authorisation:** the user receives the details of the requested consent and can then approve or deny it. The banks are notified of the answer.

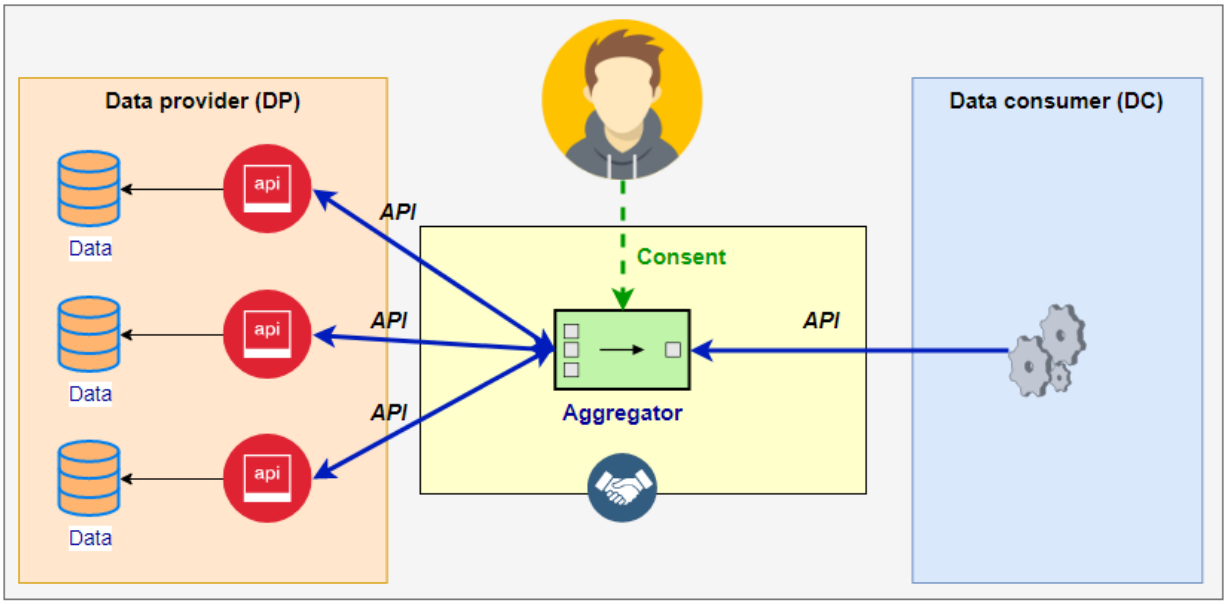
1.6 The account aggregator

An account aggregator (AA) is an intermediate technological platform responsible for managing and transferring data flows between DPs and DCs. One function of AAs is to develop interoperability between participants. But they are only intermediaries and cannot store the data or redirect it to unauthorised entities. An important feature of account aggregators is how they develop mechanisms to gain consent for data flows from and for the end users.[4] Central authorities could play the role of AAs.

[3] CGIDE (2021) provides an example of an authentication and authorisation process. For example, for the payment initiation process, the user approves the action and the financial institution validates the consent. Another example relates to the onboarding process in which the authentication factor maintains knowledge of the user's financial institution on an exclusive basis, independent of the third party involved in the process.

[4] For more details, see Press Information Bureau, Government of India (2021).

Consent architecture with account aggregator Graph 1



There are examples of AAs in which third parties certified by central authorities carry out developments and implementations. For instance, in India there are currently three companies regulated by the Reserve Bank of India (RBI) that provide services as AAs. India's AA ecosystem requires API schema, financial information schema conformance, compliance with security specifications and a test plan.[5] In other words, the RBI does not implement a public account aggregator platform for data-sharing. Instead it regulates, certifies and licenses third-party providers that implement the account aggregator. AA ecosystems contain the following flows:[6]

(i) account discovery and linking flow;

(ii) consent flow;

(iii) consent handling management flow;

(iv) financial institutions data flow;

(v) notification flow; and

(vi) monitoring flow.

In South Korea the account aggregator's base is the Korea Financial Telecommunications and Clearings Institute (KFTC). The KFTC is a unique platform provided by the Bank of Korea (BOK) and a consortium of commercial banks. It centralises all management and API calls, acts as the account aggregator and manages consent. The KFTC is also a retail payment network operator, which is not the case for the Indian AA. The following services implement transaction and query APIs for the data aggregator:[7]

[5] More details and account aggregator flows are available at https://sahamati.org.in/certification/.

[6] Detailed list of flows and APIs are available at https://sahamati.org.in/account-aggregator-key-resources/#technical.

[7] More details about KFTC: https://olc.worldbank.org/system/files/Korea%27s%20Open%20Banking_KFTC_Yunjong%20Moon.pdf.

(i) receipt information inquiry;

(ii) remitter identification;

(iii) account holder identification;

(iv) transaction information inquiry;

(v) account balance inquiry;

(vi) credit transfer; and

(vii) debit transfer.

2 Data-sharing implementation process

APIs are technological components implemented under a software development approach, and have their own life cycle. The API life cycle consists of planning and design, development, testing, deployment and production, and retirement stages (Nolle (2021)). However, this approach may not suffice. From a data-sharing perspective, central banks may start with an exploratory or investigatory stage. Then, they could move on to proofs of concept (PoCs), prototypes, pilots and subsequent mass production in the case of an innovation model (Sonin (2022)). Therefore, central banks may take on a comprehensive model that covers the initial stages of core banking, from an innovation approach to the final phases of start-up and data exchange operation (API implementation). Graph 2 shows a diagram of the proposed model.

(i) **Initiation:** the motivation to start this innovation project within a specialised group. A central bank can propose a plan with objectives and milestones and identify other financial authorities.

   a. Participants: central banks, financial authorities.
   b. Deliverables: kick-off report, schedule.

(ii) **Research:** the stage of conceptual research through an established team. The central bank reviews the literature on data-sharing. Mutual collaboration and working groups with peers could be necessary to learn from other central banks' experiences.

(iii) **Model design:** central banks evaluate the alternative data-sharing models available for open banking. This report presents the centralised, decentralised and trust ecosystem models. One model would be more suitable than another, depending on whether it requires strict regulation or not (see Annex B).

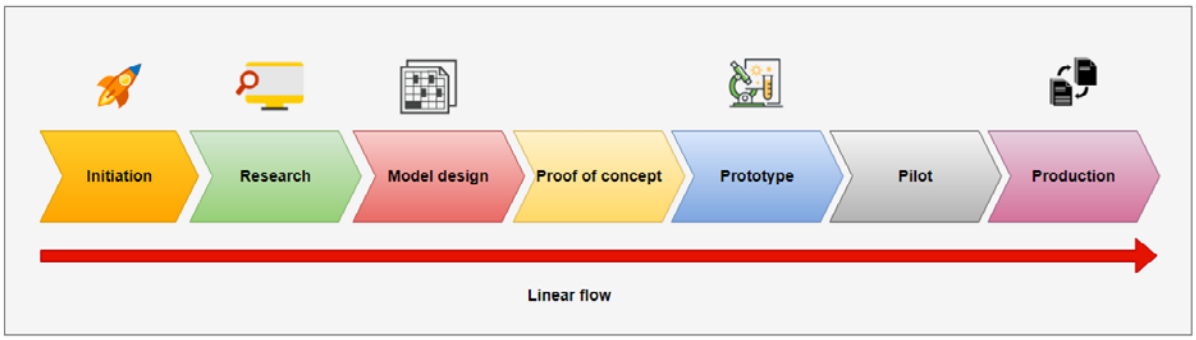
(iv) **Proof of concept:** depending on the choice of data-sharing model, this stage could involve building a public platform for data-sharing purposes. The PoC implies conducting an exercise to verify if the proposed solution aligns with the theoretical concepts and that it will be possible to implement.

(v) **Prototype:** the iterative process of implementing the data-sharing platform. In this phase, the authority completes the API life cycle in full.

(vi) **Pilot:** fully developed solution. The authority can opt to start working with a small target user group to get feedback.

(vii) **Production:** final stage. Full implementation of data-sharing model.

Data-sharing implementation process Graph 2



When the central bank does not undertake the development of a centralised data-sharing platform, it may not execute the steps of PoC, prototype, pilot and production. Instead, the central bank would focus on legal and regulatory aspects and not on implementing its own specific technological solution.

3 Data-sharing flow models

The data-sharing flow model is largely determined by the choice of a rigid or flexible regulatory framework. The greatest challenges in selecting an adequate model are establishing where to store the data, identifying the consumers and determining which communication interfaces to use. Two other factors which play a role in establishing an adequate model are determining the responsibilities of the parties involved and obtaining the consent of mandatory users. Finally, other important decisions before any kind of implementation can take place involve communication technologies, protocols, standardised messages, infrastructures and security mechanisms. There are three viable alternative models: centralised, decentralised and trust-ecosystem.

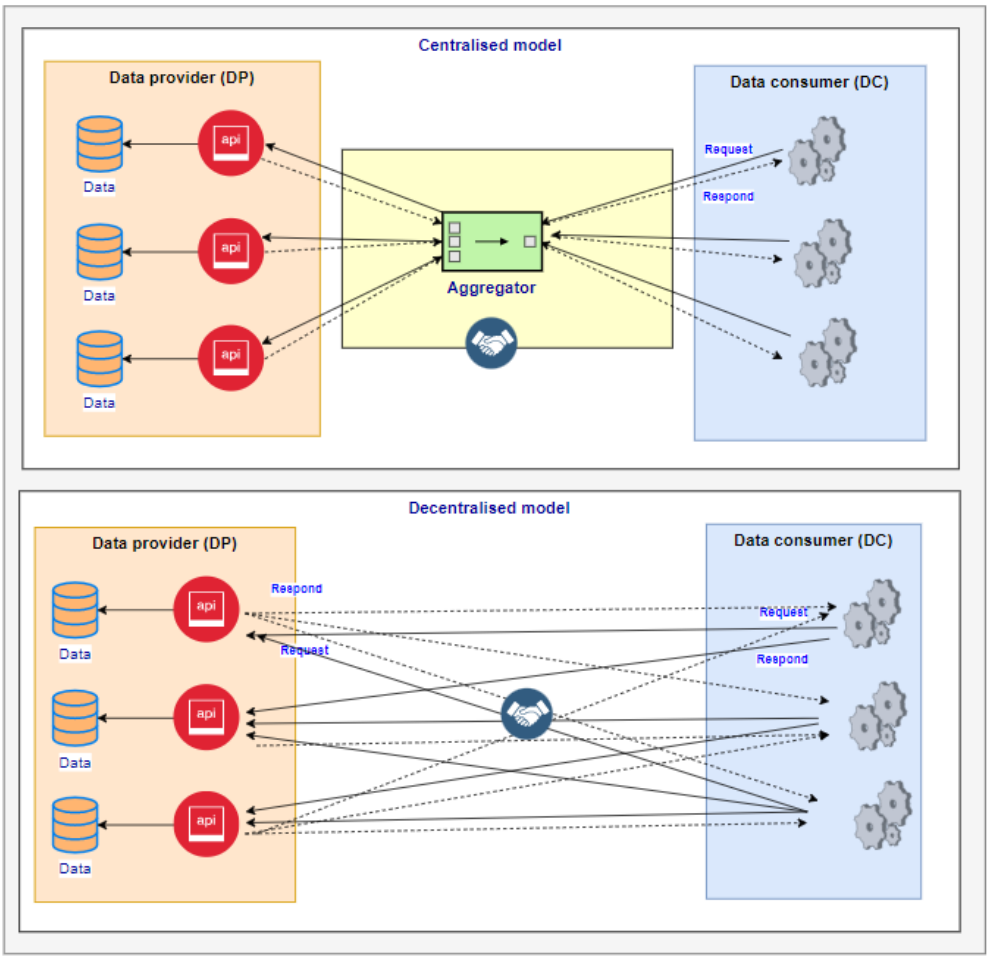
3.1 Centralised model

In a centralised model, an aggregator collects the data. The institution in charge of the exchange (the data provider) has full control over data-sharing. This includes control over the authorisation and authentication process to access the data through the aggregator. One of the key benefits of this model is the short response time for data returns as it is quicker to obtain data from a central aggregator (ie a consolidated source) than from multiple sources.

However, some challenges emerge in a centralised model:

- It relies on consistent, timely data transfers from third-party providers. Hence, it does not guarantee data availability. Service level agreements and a monitoring mechanism for DPs are required.
- There are risks related to data mismatches, omissions or duplicates. To reduce these risks, effective user-matching algorithms are required.
- In the case of a centralised data repository, the institution in charge of centralising the data is responsible for privacy and information technology security.

Data-sharing models Graph 3



3.2 Decentralised model

In a decentralised model, data also remain at the source or point of service. However, participating members agree to share their data with other participants individually. An alternative name for this model is the federated exchange model. Each participating organisation maintains ownership and control of the data within its source databases. One of the main benefits is that it guarantees access to up-to-date data and each participant can negotiate the list of data shared.

An administrative or governance group may maintain a centralised list of citizens or a directory. This facilitates data transfers using standard integration methods. However, as there are point-to-point transfers between the participants it is not necessarily possible to follow a precise standard for the exchange of data. A challenge is that a standard and predefined field about how and what to share between participants does not necessarily exist. Each participant individually negotiates based on the possibilities available. However, one positive factor is that response times are faster than in the centralised model since it is a direct point-to-point connection without a centraliser.

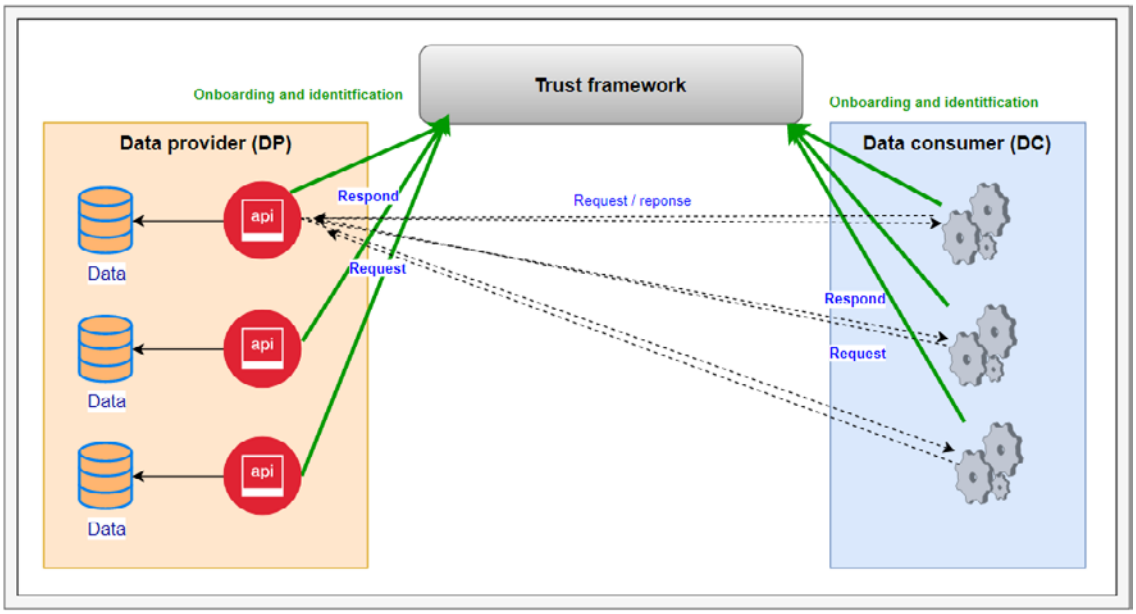
3.3 Trust ecosystem model

In this model, the AA is not necessary if standards are very well defined. Accordingly, there is no centraliser. The key pillars of this model are standardisation, testing and an accurate certification process. The model is decentralised for data-sharing and centralised for identity management. The proposal relies on a trust framework that dynamically registers both data providers and consumers (Graph 4).

The trust framework integrates with a trusted third party (TTP), using TTP libraries instead of an aggregator.[8] One key benefit is that third parties can develop their libraries in multiple programming languages.

The model requires the establishment of a registration process for participants, security in communications and a standard for the exchange of information through an API. Since the implementation of standards often vary, even if only slightly, the model states that a certification process managed by certifying authorities is necessary

Trust framework — Graph 4



4 Interaction and data flow

Enabling open finance through APIs: report on payment initiation (CGIDE (2021)) developed two schemes: a central validator (CV) auth-app scheme and an in-app scheme. The former requires the installation of an application from a third-party provider. The app is responsible for the authentication of credentials and confirmation of payment transaction (or a consent for data-sharing). The latter scheme uses an in-app model integrated natively without the need to install an additional app. These models allow the development of similar alternatives for account aggregation in the context of data-sharing.

[8] Proposed by Raidiam Services Limited for the open banking models of Brazil and the United Kingdom.

Next, the report developed three main user interactions and data flows based on the models previously described:

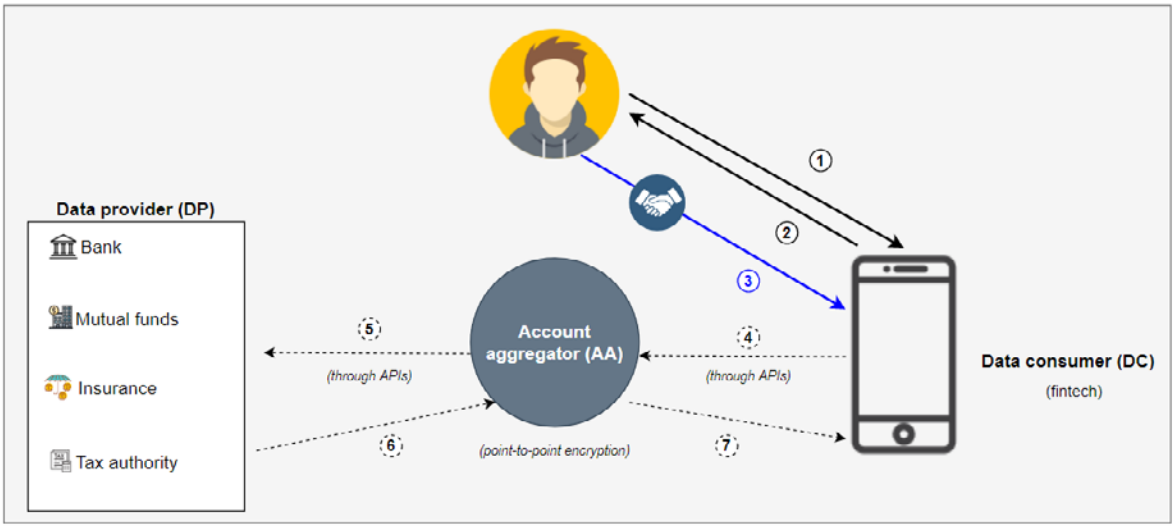
- a fully centralised model via APIs;
- a centralised model with a third-party consent app; and
- a trust model without centraliser.

4.1 A fully centralised model via APIs

Preconditions:

- The user has already installed the fintech app (one app only).
- The user has created a PIN or credentials with the fintech app and completed the onboarding process.[9]

Centralised model (account aggregator) via APIs — Graph 5



Data flow and interaction (Graph 5):

1. The user requests a financial service from a fintech, for instance, a loan.
2. The fintech processes the request and, within its flow, it requires access to a third-party data provider holding user information. The fintech shows the user the list of sources from which they need to add information (banks, insurance companies, government agencies and others).
3. The user gives consent (providing an authentication factor) to access the data from the requested sources.
4. The fintech app connects to its back-end server, which has a secure, private and encrypted connection to the account aggregator. The back-end server sends the query request to the aggregator, to whom it then provides meta-data which guarantee that the user has given consent.

[9] A feasible alternative for the onboarding process is defined in CGIDE (2020)).

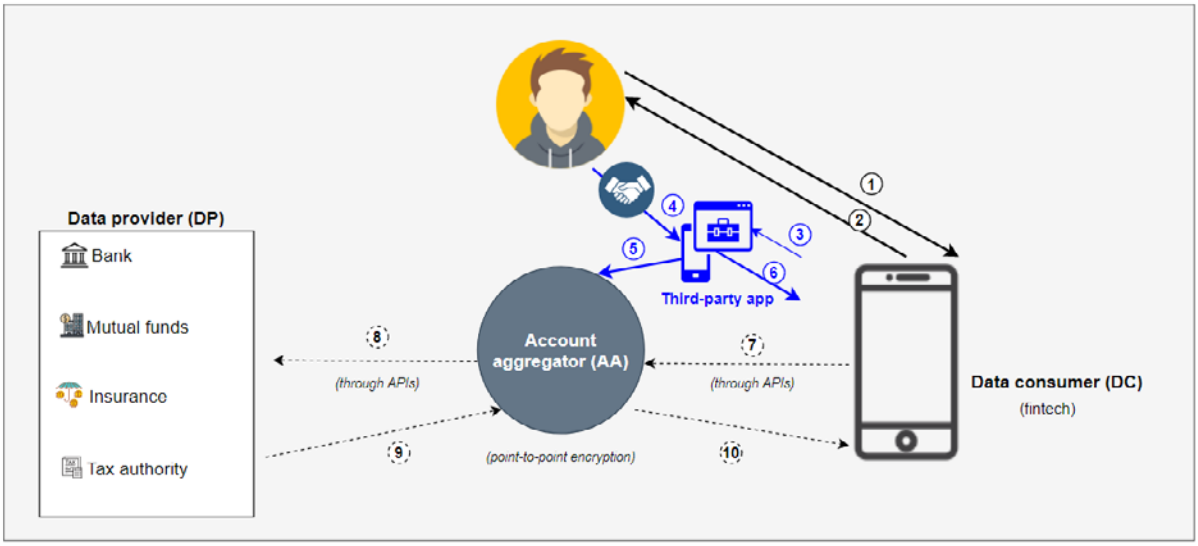
5. The account aggregator validates the received data. It then proceeds to consume the APIs of all the data providers involved.
6. The data providers respond to the account aggregator. The aggregator proceeds to consolidate the data received.
7. The account aggregator responds to the fintech's back-end server with the results obtained. Finally, the fintech can process the financial service offered to the client.

4.2 A centralised model via a third-party consent app

Preconditions:

- The user has already installed the fintech app.
- The user has already installed the consent app (third-party app).
- The user has created a PIN or credentials with the fintech app and the consent app and completed the onboarding process in both environments.

Centralised model (account aggregator) via a third-party consent app — Graph 6



Data flow and interaction (Graph 6):

1. The user requests a financial service from a fintech, for instance, a loan.
2. The fintech processes the request and, within its flow, it requires access to a third-party data provider holding user information. The fintech shows the user the list of sources from which they need to add information (banks, insurance companies, government agencies and others).
3. The fintech also invokes the third-party app and displays it with the consent interface.
4. The user gives consent to access the data from the requested sources.
5. The third-party app notifies the account aggregator.
6. The third-party app notifies the data consumer app, provides consent and invokes the parties.
7. The fintech app connects to its back-end server, which has a secure, private and encrypted connection to the account aggregator. The back-end server sends the query request to the aggregator.

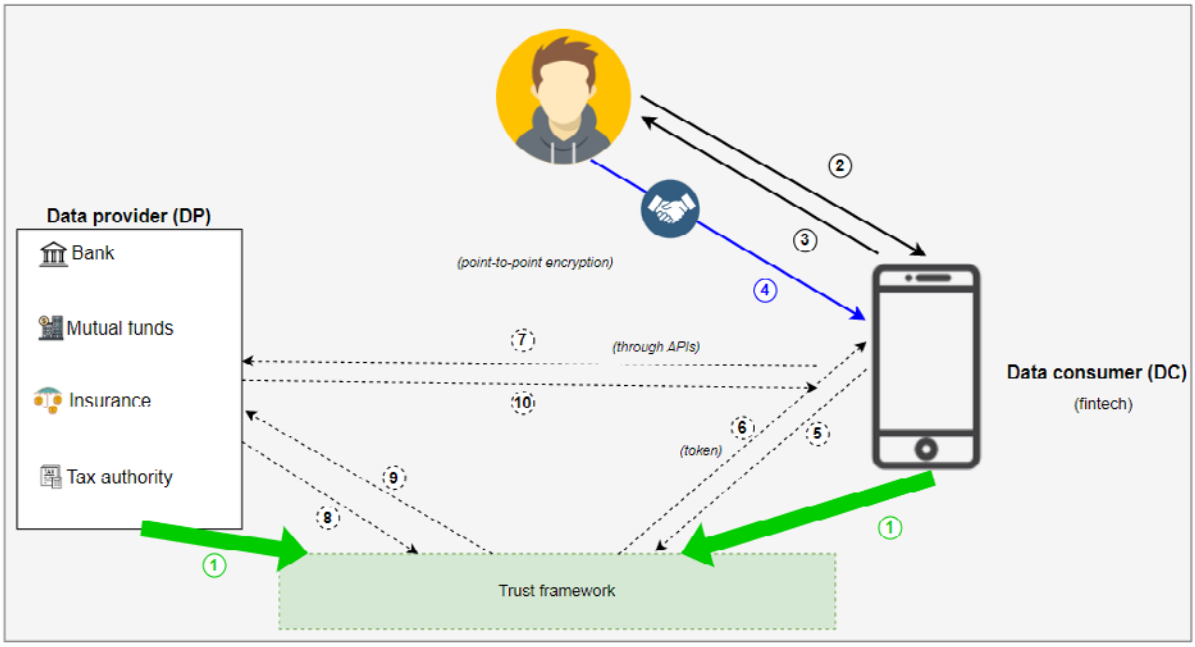
8. The account aggregator validates the received data. It then proceeds to consume the APIs of all the data providers involved.
9. The data providers respond to the account aggregator. The aggregator proceeds to consolidate the data received.
10. The account aggregator responds to the fintech's back-end server with the results obtained. Finally, the fintech can process the financial service offered to the client.

4.3 A trust model without centraliser

Preconditions:

- The user has already installed the fintech app.
- Both data provider and data consumer have already performed the onboarding process (step 1 in Figure 7) for the trust framework.

Trust model without centraliser — Graph 7



Data flow and interaction (Graph 7):

1. Data provider (DP) performs an onboarding process. Data consumer (DC) also performs the onboarding process.
2. The user requests a financial service from a fintech, for instance, a loan.
3. The fintech processes the request, and within its flow, it requires access to a third-party data provider holding user information. The fintech shows the user the list of sources from which they need to add information (banks, insurance companies, government agencies and others).
4. The user gives consent to access the data from the requested sources.

5. The fintech app connects to its back-end server, which has a secure, private and encrypted connection to the trust framework. The back-end server sends the token request to the trust framework.
6. The trust framework authenticates and validates the fintech back-end server and generates a token. The fintech app receives the token.
7. The fintech back-end server makes a direct connection, via API, to the DP using the token provided.
8. The DP connects to the trust framework to validate the token.
9. The trust framework answers the token validation.
10. If the token is valid, then the DP returns the data requested by the DC.

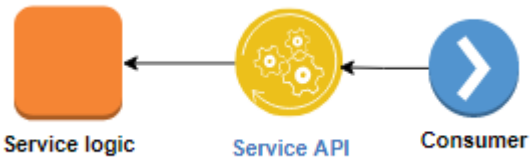
The token can remain valid for a prolonged period (expiration). Therefore, the process could include mechanisms such as time stamps and cryptography so that steps 8 and 9 are not redundant. Alternative models could improve this exercise.

5 Technological considerations for API design

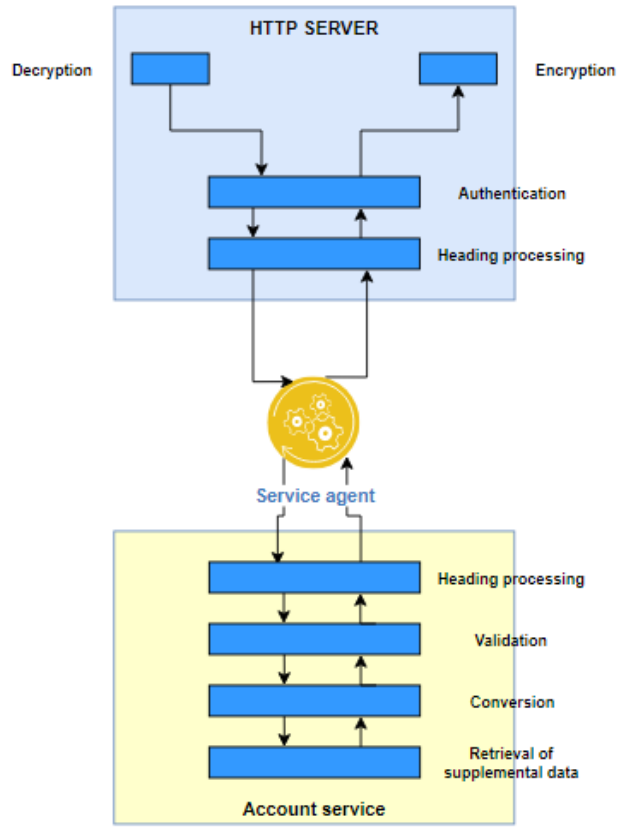
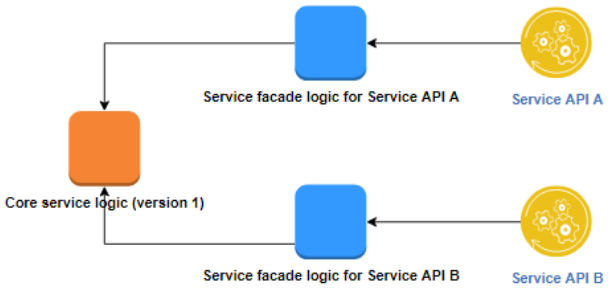
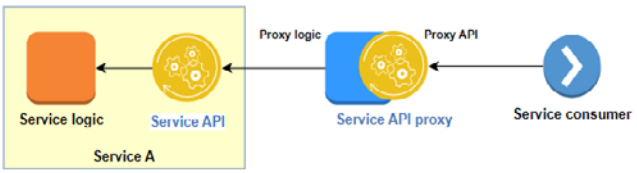
Graph 1 showed that in the account aggregation implementation scheme, those in charge of implementing APIs that publish services are the DP and the AAs. By contrast, the trust framework requires reciprocity which implies that both the DP and the DC must implement and publish API services. This section explains the design alternatives available for API implementations.

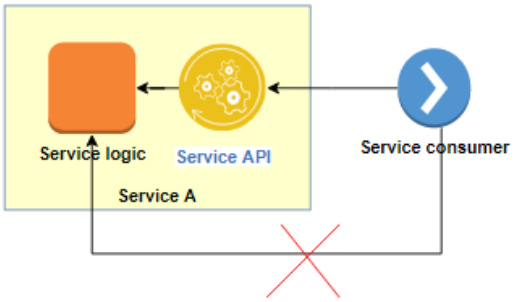
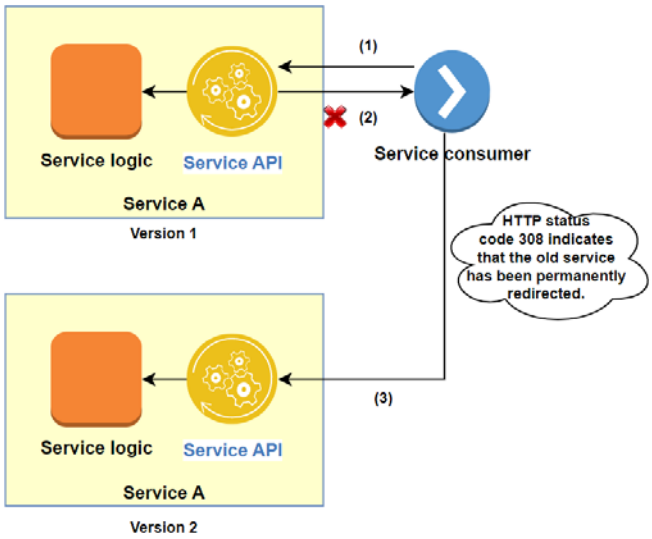
The API literature provides a series of basic patterns to achieve robust interfaces and technologies for data-sharing. The following section presents a series of API service patterns proposed by Arcitura.[10]

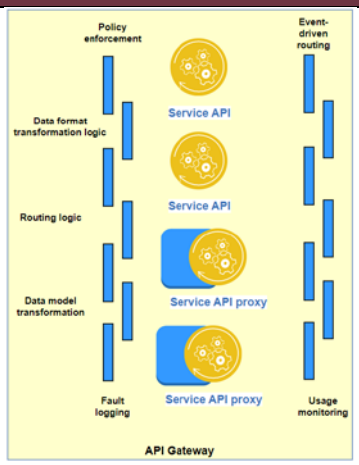
5.1 Service API design patterns

Pattern	Definition	Interaction
Decoupled service API	Separates the service logic and service API. It allows service logic to functionally scale without having a negative impact. It promotes proper future maintenance.	

[10] Arcitura provides and develops technology certification programmes. Arcitura offers an API Specialist certification in which participants can develop a series of patterns. See https://patterns.arcitura.com/service-api-patterns.

Pattern	Definition	Interaction
Service agent	Uses a program as an intermediary to manage messages received and sent between two parties.	
Service facade	Coupling service logic and standardised APIs generates software maintenance issues. This pattern inserts a facade that isolates the abstraction from the service logic component. This can promote the use of multiple facades pointing to a service logic.	
Service API proxy	Coupling of service logic and API can be strong. A solution is to couple a proxy and the API. As a result, changes in consumers of a service do not affect their interface.	

Pattern	Definition	Interaction
Service API centralisation	Programs' consumption could occur through multiple paths, resulting in distinct types of outputs. An API that centralises calls for, or consumption of, the service would create a unique entry point.	
Endpoint redirection	REST API service may change its implementation over time, leading to major changes involving an updated version of the system. Therefore, it is necessary for the new implementation and the API to be isolated and have their own ecosystem. Then, the old API will implement the endpoint redirection pattern that, on its consumption, will redirect to the newly published API, preventing consumers from changing its implementation.	

Pattern	Definition	Interaction
API gateway	The API gateway is a service consumption entry point pattern involving redirection, routing, messaging monitoring, security, authentication and transformation. API gateways are an improved design way of processing APIs.	
Source: Arcitura (2022 a, b, c, d, e, f, g).		

5.2 API protocols and styles

Communication protocols for APIs show a predominant market steer toward REST, with a considerable number of legacies developed in SOAP. Below are some of the most relevant protocols.

Protocol	Definition	Interaction
SOAP	Web services use a specific set of established industry standards that generally include WSDL (for the definition of service contracts), SOAP (for the definition of messages) and XML Schema (for the definition of messaging data models).	In this architectural style, a SOAP client formulates a request for a service. The client transfers an XML document to a SOAP server, using HTTP or HTTPS. Then, the web server receives the SOAP message as an XML document using the SOAP request handler servlet. The server then dispatches the message as a service invocation to an appropriate server-side application providing the requested service. Finally, the SOAP request handler servlet, and then the caller, receive a response using the standard SOAP XML payload format.
REST	Representational state transfer (REST) is a distributed architectural style based on the underlying architecture of the world wide web (Kerry (2022)).	In REST, clients and the server are separate, which helps to establish a fundamentally distributed architecture and supports the independent evolution of the server logic and its clients. From a service contract perspective, this requires that a service offer one or more capabilities. Service consumers invoke a capability by sending a request message

Protocol	Definition	Interaction
		for the service to perform a task. The service subsequently replies with a response message.
Traditional RPC	A remote procedural call (RPC) is one of the oldest and simplest forms of API interaction. The system sends a client's request to a server. The result is then executed on the server and sent back to the client. Client-side and server-side stubs are involved in conducting this communication. A stub is a segment of code that converts parameters sent from the client to the server and vice versa.	The client-side logic sends a request to the client's RPC stub. It then forwards the request through the network via a network protocol, such as TCP/IP. The server's RPC stub receives the request and forwards it to the server logic. The server logic processes the request and sends a response to the server's RPC stub. The server's RPC stub forwards the response back to the client's RPC stub. Finally, the client's RPC stub forwards the response back to the requesting client logic.
gRPC	Originally developed by Google. The gRPC is a protocol designed to support distributed systems with high scalability requirements. The gRPC framework provides a built-in authentication API with native support for SSL/TLS.	A gRPC client sends a datum serialised into a hex value to the server as part of a protocol buffer (protobuf) message. The gRPC server processes the message and responds with a result also serialised into binary format as part of a protobuf message.
GraphQL	Originally developed by Facebook, this is an RPC-based protocol designed for large environments with high volumes of diverse types of data. It is a query language like SQL.	Similar to standard RPC, GraphQL requires installed server-side and client-side software to carry out communication. Unlike standard RPC, GraphQL clients and servers commonly communicate over HTTP. The client and server have a copy of the JSON schema used to define and validate the data in an exchanged JSON message.
Falcor	Originally developed by Netflix, Falcor is an RPC-based protocol designed to optimise data access across various data sources. Like GraphQL, Falcor is a query-centric API commonly used with JSON as the message data-serialisation format.	The Falcor RPC framework requires installed server-side and client-side software to carry out communication. JSON is a common messaging format used to exchange non-binary data over HTTP.

5.3 Service API access levels

Service API access levels depend on how the authority implements data-sharing, as well as the level of regulation. Below are the access levels in the context of open banking (Brodsky and Oakes (2021)).

Level	Definition	Access
Public	Public APIs are generally open and accessible, published with services hosted in a cloud-based environment and may have built-in subscription and monetisation features.	This access level commonly employs API proxies and/or an API gateway.
Private	Private or internal APIs are only available to specific service consumers within a predefined boundary (organisational).	The use of API proxies depends on the number of service consumers and the processing logic.
Partner	A service API published for external access for pre-defined service consumers, usually from partner organisations. Such partner APIs often require additional security controls to regulate access.	Can consider using API proxies or an API gateway.

5.4 Security considerations

Below are the main components of API security mechanisms.[11]

1. **Authentication:** this is the process of identifying whether the client and users are who they claim to be. It is the first step in the secure implementation and execution of the API.
2. **Access control:** control mechanisms limit API consumers' actions after correct authentication. It validates and grants authorised accesses, while for those not deemed suitable it responds with HTTP code 401 Unauthorised or 403 Forbidden.
3. **Encryption:** security mechanisms use tokens for encryption. They are simple data structures essential to the functionality of APIs. Encrypted tokens store vital information such as the username and password. These tokens expire after a certain time, strengthening the API's security.
4. **Audit logging:** a registry that stores actions and calls made to the API. This log promotes accountability. Records store all key activities after authentication and control, both the positive ones and the failures or falls.

Below are some recommended standards for the implementation of security mechanisms for REST APIs.

[11] For more information see Madden (2020). This section complements "Annex D: Cybersecurity in the API ecosystem" in CGIDE (2021) which provides guidelines related to governance, asset risks and other concepts.

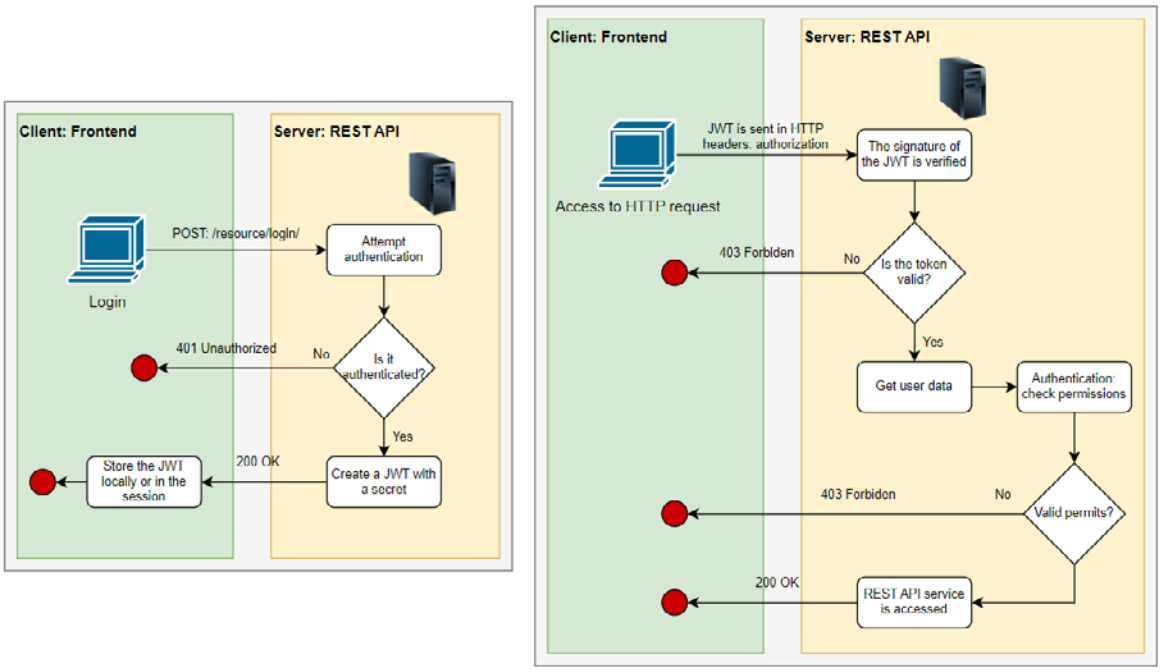
5.4.1 JSON Web Token

In security, tokens are scalable and can integrate multiple applications (desktop, mobile, other servers, etc). JSON Web Token (JWT) is an open standard (RFC 7519) to implement security in a safe way for REST APIs projects. JWT allows the user to send an alphanumeric code to the server and then the server is responsible for deciphering and validating if the user exists and what their permissions are according to their roles.

Graph 8 (left-hand panel) shows the authentication process for access to an API Rest service through POST. The server validates authentication and if correct (code 200) creates the JWT. It then sends the token to the client that made the request. Otherwise, if authentication fails, the server rejects the request with a message (code 401).

Graph 8 (right-hand panel) shows access to a service using the token the client obtained. If successful, the server authorises access to the required service.

JWT flow — Graph 8



5.4.2 OAuth 2.0

OAuth 2.0 is the industry standard open framework for authorisation. It provides a set of standardised JSON and HTTP-based message flows that enable developers to create authentication and authorisation protocols. This framework allows third-party applications to request a user's authorisation for access to a specific service.[12] If authorised, the third-party application obtains an access token that it uses to obtain the protected data or resources.[13]

[12] OAuth 2.0.

[13] See details of the framework at datatracker.ietf.org/doc/html/rfc6749.

5.4.3 OpenID Connect

OpenID Connect is an authentication standard based on REST/JSON message flows and specified by OAuth 2.0.[14] OpenID Connect is more secure as it allows user authentication from different types of client (web, mobile etc), without the need to manage user passwords. The OpenID Connect framework is based on public-key-encryption and uses JWT as data structures for signature schemes.[15]

5.4.4 Financial-grade API (FAPI)

Financial-grade API (FAPI) is a technical specification based on OAuth 2.0 authorisation and OpenID Connect authentication. It includes additional technical requirements geared towards industries that need a higher level of API security. For example, the use of bank, insurance or credit card accounts in the financial sector. Use case applications include:[16]

- Applications using a standard-based secure data model (JSON) for levels of access to financial data stored in accounts.
- Applications using a standard-based program interface (REST) for sharing financial data between banks, institutions and third parties.
- Application and user security controls and privacy settings to be consistently implemented with open standards (OAuth) and providers (OpenID Connect).

The first version, FAPI 1.0, consists of "FAPI 1.0 Baseline", which is suitable for protecting APIs with a moderate inherent risk and is intended for read only functionality; and "FAPI 1.0 Advanced" for a higher level of security and intended for read-write functionality (both have read-write functionality). A new version (FAPI 2.0) was published as an implementer's draft, meaning that it is locked, stable and ready for implementation. FAPI 2.0 has wider scope to achieve greater interoperability and security in authorisation flows than the previous version.

For communications between a client and server, best practice is to use TLS for encrypts at the transport layer for all API traffic. For both client and server API architecture it is recommended to maximise the safe handling and safe storage of JWT Tokens, Access Token OAuth etc.

[14] OpenID (2022a).

[15] See more details on The OpenID Foundation and OpenID Connect sites.

[16] OpenID (2022b).

Open finance in Brazil

Aligned with the principle that consumers own their personal data and therefore should be able to use them to their own benefit, including sharing with other parties, the Central Bank of Brazil (BCB) and the National Monetary Council (CMN) defined the main principles and rules for open banking (which evolved into open finance). These principles allow the standardised sharing of data and services between financial institutions and other institutions licensed by the BCB.

The data-sharing occurs through standardised APIs in a safe, agile, precise and convenient process covering the steps of consent, authentication and confirmation. This data-sharing must have been previously consented to by the customer. Each consent must be tied to specific purposes and has a validity period limited to 12 months, which could be revoked by the customer at any given time through either of the institutions involved in the data-sharing.

The Brazilian model has a broader scope than the initiatives in other countries. Participation in data-sharing is mandatory for the largest institutions (S1 and S2 firms, according to prudential regulation), while other licensed firms can participate by observing a data reciprocity requirement. The scope of the data includes the standardisation of APIs for sharing open data on available products and access channels (phase 1), customer registry and transactional data on payment and accounts (phase 2) and payment initiation service (phase 3). However, as in other jurisdictions, the Brazilian model also includes data-sharing on credit operations, and has done so from the start. Participants are expected to soon have APIs in place to share data on investment, insurance and foreign exchange operations (phase 4), among others.

A secure environment is key to reducing information asymmetry by de-monopolising data, leading to increased competition and, hopefully, cheaper and better financial services for consumers. In this regard, the BCB set rules to build an initial governance structure with the aim of reducing players' often conflicting interests while seeking non-discriminatory access and regulatory compliance. In this structure, the participating segments share equal status and voting powers to present the technical standards and build a common tech infrastructure (eg a participant directory, service desk and sandbox).

This framework sets technical requirements for open finance sharing, including OpenID Connect 2.0 and a national Financial-grade API (FAPI) profile. The governance structure submitted these requirements to the BCB, and the BCB ultimately included them in the regulatory framework that all participants must observe.

The BCB expects open finance to promote a more competitive and efficient financial system, by generating new opportunities for all parties involved. Many use cases will only become clear over time. Some are already apparent such as financial counselling and payment initiation. Others such as credit risk analysis and customer onboarding by participating institutions show improvement. In numbers, open finance in Brazil has recently completed one year of implementation, with over 5 million active consents, over 2 billion successful API calls and around 800 participating institutions in total.

As for the main challenges, the BCB points out that participants must agree on a definite governance structure in 2022, and the ecosystem should continue to evolve towards phase 4. In 2023 the BCB expects to issue requirements for interoperability in the open insurance framework. Furthermore, raising public awareness of the project is another key challenge in the years to come.

In conclusion, open finance is not meant as a static model but rather as an evolving one. Although defining the technical standards is a challenge in the short-term, its scope should be understood as dynamic in nature, enabling new solutions in the long run.

Technical remarks

The BCB has decided that each financial institution (data provider) is legally responsible for authenticating the customers (data owners) and third parties (data recipients) asking for their customers' data. They are also legally responsible for ensuring that third parties access only data and capabilities for which access has previously been authorised (consented to) by customers. These legal requirements naturally led Brazil's open finance to a decentralised architecture.

Aligned with jurisdictions such as the United Kingdom and Australia, Brazil's open finance architecture is based upon open standards like OAuth, Open ID Connect and FAPI. In the context of Brazil's open finance, OAuth is a framework through which customers authorise third parties to access their data. Open ID Connect is an extension of OAuth which allows third parties to ask data providers they trust to authenticate customers. Finally, FAPI is a security profile that restricts the mechanisms (flows, algorithms etc) of OAuth and Open ID Connect to those deemed to be adequately secure for financial purposes.

Besides OAuth, Open ID Connect and FAPI, the major remaining building blocks of Brazil's open finance architecture are a directory, a consent API and FAPI Brazil (OFBIS GT Security)[17]. The first allows financial institutions and third parties to establish trust relationships between themselves in a scalable way. Also, financial institutions advertise the address of their Open ID provider and the addresses of their APIs' implementations through the directory. The consent API and the integration of its implementations in authorisation servers provide a mechanism for dealing with fine-grained authorisations, which OAuth currently lacks. Finally, there are legal identification and privacy requirements in Brazil's open finance that are met by FAPI Brazil by minor modifications to the FAPI standard.

The initial governance structure specifies that Brazil's open finance APIs are to be in the OpenAPI format, in a public collaborative environment, starting from a conceptual description of the data and capabilities that the APIs should make available to third parties (Portal do Open Banking Brasil) (BCB (2021a))[18].

Brazil´s open finance APIs are JSON-based, tentatively adopt ISO 20022 terminology and are semantically versioned[19]. They also follow guidelines set both by the BCB (BCB (2021b)) and by the initial governance structure (Portal do Open Banking Brasil) regarding URI structures, HTTP headers and status codes, naming conventions, common data types, pagination etc. Such guidelines help keep the APIs consistent, not only internally but also with one another. Sample implementations of the APIs are available, which help financial institutions to develop their own.

Financial institutions are only allowed to advertise their Open ID providers in the directory after a comprehensive automated security test suite deems them FAPI Brazil-compliant. In a similar fashion, implementations cannot be advertised in the directory before an extensive automated functional test suite asserts their compliance with the APIs. These automated test suites are key to ensuring the interoperability of financial institutions and third parties in Brazil's open finance.

[17] See more details on https://openbanking-brasil.github.io/specs-seguranca/open-banking-brasil-financial-api-1_ID3.html.

[18] See more details on https://openbankingbrasil.atlassian.net/wiki/spaces/OB and https://in.gov.br/en/web/dou/-/instrucao-normativa-bcb-n-184-de-12-de-novembro-de-2021-359803029.

[19] See SemVer (Semantic Versioning 2.0.0).

Open finance in Mexico

The financial authorities in Mexico seek to promote an open finance architecture with a robust data-sharing infrastructure to foster competition and transparency, increase efficiency and offer services that are more customised to clients' needs. This requires a secure framework for and management of the data being shared. Data ownership remains with the customers, as stipulated by Mexico's Data Protection Law. This law sets out how data are managed, as well as the approval and removal of consent required for data-sharing.

Mexico's Fintech Law requires financial entities, such as banks and non-bank financial institutions, to establish APIs for sharing three types of customer data: financial open data (eg information on products and services), aggregated data and transactional data. This allows access to other financial entities and specialised third parties that may include bigtechs and fintechs.

Secondary regulation for data-sharing applicable to clearing houses and credit information societies, published by the Bank of Mexico in its 2/2020 Circular,[20] establishes the characteristics that APIs for these entities must comply with. These include an interoperability standard and the requirements they must fulfil, given that only two types of data may be provided through these entities (aggregated and transactional) once a customer has given consent. The secondary regulation for the rest of the financial entities included in the Fintech Law is still a work in progress by Mexico's other financial authorities.

Technical remarks

The Bank of Mexico's secondary regulation for clearing houses and credit information societies establishes a decentralised model. For the case of clearing houses, the architecture includes the following key technological features in terms of interoperability, communication, authentication, access control and telecommunications, as included in Annex 1 of the 2/2020 Banxico Circular.

- APIs must be able to receive a request for services through the IETF protocol.
- APIs must be available for execution on a public IP address.
- Privacy and integrity mechanisms during transfer must protect data shared via APIs.
- The computer systems and applications that validate the identity of the API access points must establish secure communication through the TLS protocol and a valid digital certificate.
- To authenticate and control access, the clearing house must implement access control lists by an IP address.
- Interfaces must execute synchronously.
- The message format used for these APIs will be JSON, using a REST API specification format.
- TLS is issued as the security standard for authentication, authorisation and encryption.

[20] Regulation available in Spanish: www.banxico.org.mx/marco-normativo/normativa-emitida-por-el-banco-de-mexico/circular-2-2020/%7B4FDD6B5E-8DFA-F095-6325-68C388AAEAA0%7D.pdf.

6 API aggregator implementation (demo)

The following section describes the implementation of a software architecture to exemplify a data aggregator, for the case of extracting balances from three different banks. The programming language is Java, while the aggregator uses a microservices architecture based on scalability and high availability.

6.1 Preconditions

Before running and testing the solution, users must install the following tools and preconditions:

Technology	Description
Java	At least Java version 11. This language and virtual machine will allow the execution of the microservices and the simulation.
Spring Tool Suite 4	Development IDE that will speed up coding, debugging and testing.
Spring Boot 2.7.0	Java-based framework that provides libraries and utilities for the implementation of web applications, microservices and API Rest.

The recovered balance from each bank will have the following structure:

Field	Description
id	User balance identifier
bank_id	Bank identifier in the schema
label	A label given by the owner of the account
number	Balance number
product code	Product code provided by the bank
Overall: currency	National currency
Overall: balance	Balance value
Account: scheme	A list that might include IBAN or national account identifiers
Account: address	A list that might include IBAN or national account identifiers

6.2 Software architecture

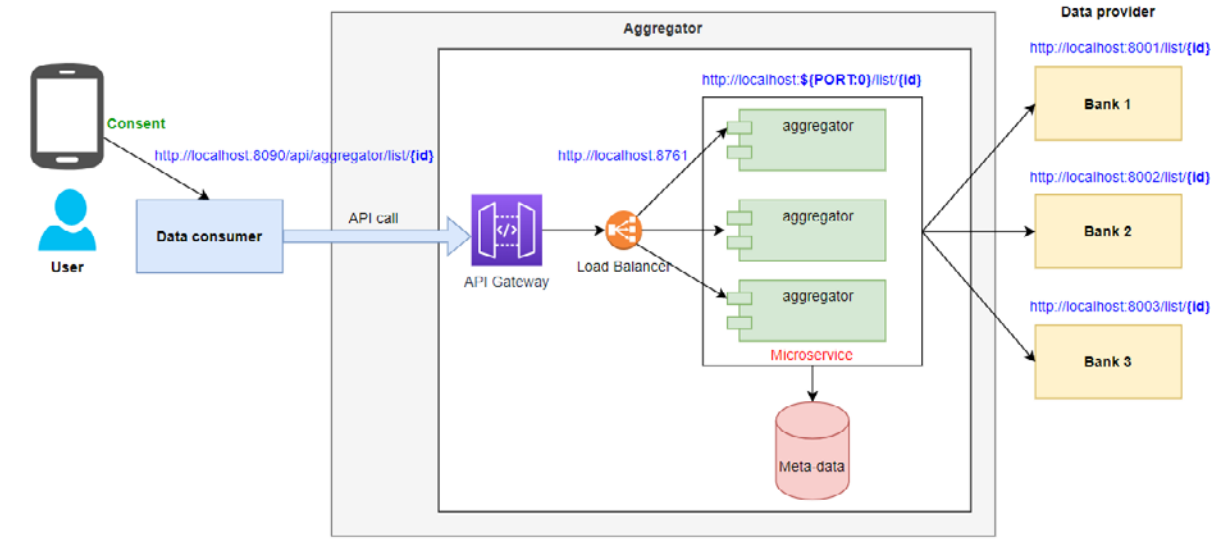
The software architecture includes the publication of an API Rest that will be consumed by the **data consumer**, through the address: http://localhost:8090/api/aggregator/list/**{id}**, where the id is the identifier of the citizen with which the balances in each of the three banks will be searched.

The aggregator will initially implement an **API gateway** that serves as the main entrance of the architecture. We developed this gateway in **Zuul**, a specialised server for this purpose. This server

implements the previous path. The gateway will then communicate with a **Eureka server** that implements the high availability balance sheet microservices. Eureka is a server that registers each new instance created. This server is responsible for monitoring each registered microservice, as well as serving as a load balancer for requests. The Eureka server address is: http://localhost:8761/.

The aggregator microservice is a project called **aggregator**. We developed this project in Spring Boot. It can be initialised several times and each time the Eureka server will create a register. The path of each microservice is: http://localhost:**${PORT:0}**/api/aggregator/list/**{id}**, the server randomly generates each port and the id is the user identifier.

Aggregator software architecture[21] Graph 9



Finally, in this architecture, each of the three banks must implement its own API, under the standards and guidelines defined by the regulator.

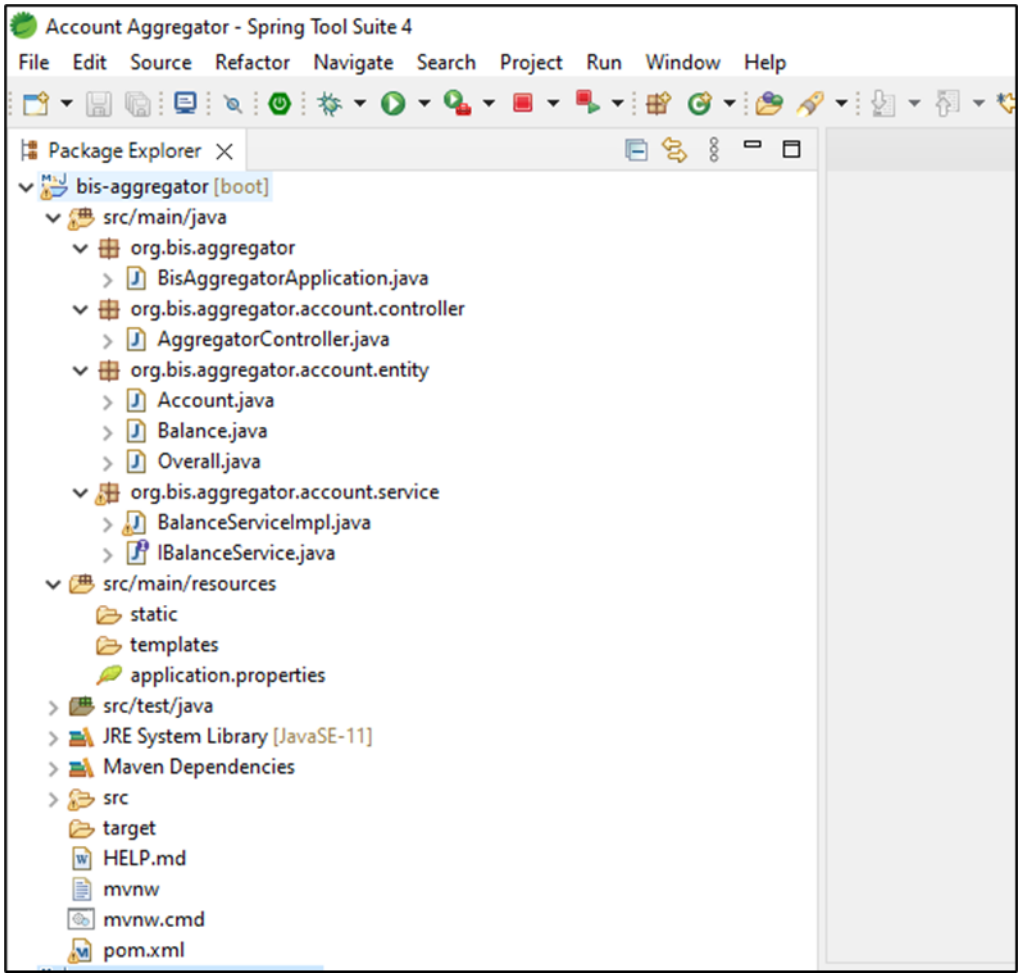
- **Bank 1:** http://localhost:8001/list/aggregator/**{id}**
- **Bank 2:** http://localhost:8002/list/aggregator/**{id}**
- **Bank 3:** http://localhost:8003/list/aggregator/**{id}**

6.3 Implementation

The aggregator's implementation occurs in a layered architecture. A controller is in charge of the API publishing interface. The class **BalanceServiceImpl** is responsible for managing the business logic ie connecting to each of the banks, consolidating the information and responding to the request.

[21] **Disclaimer:** this demo environment does not consider the API security mechanisms mentioned above.

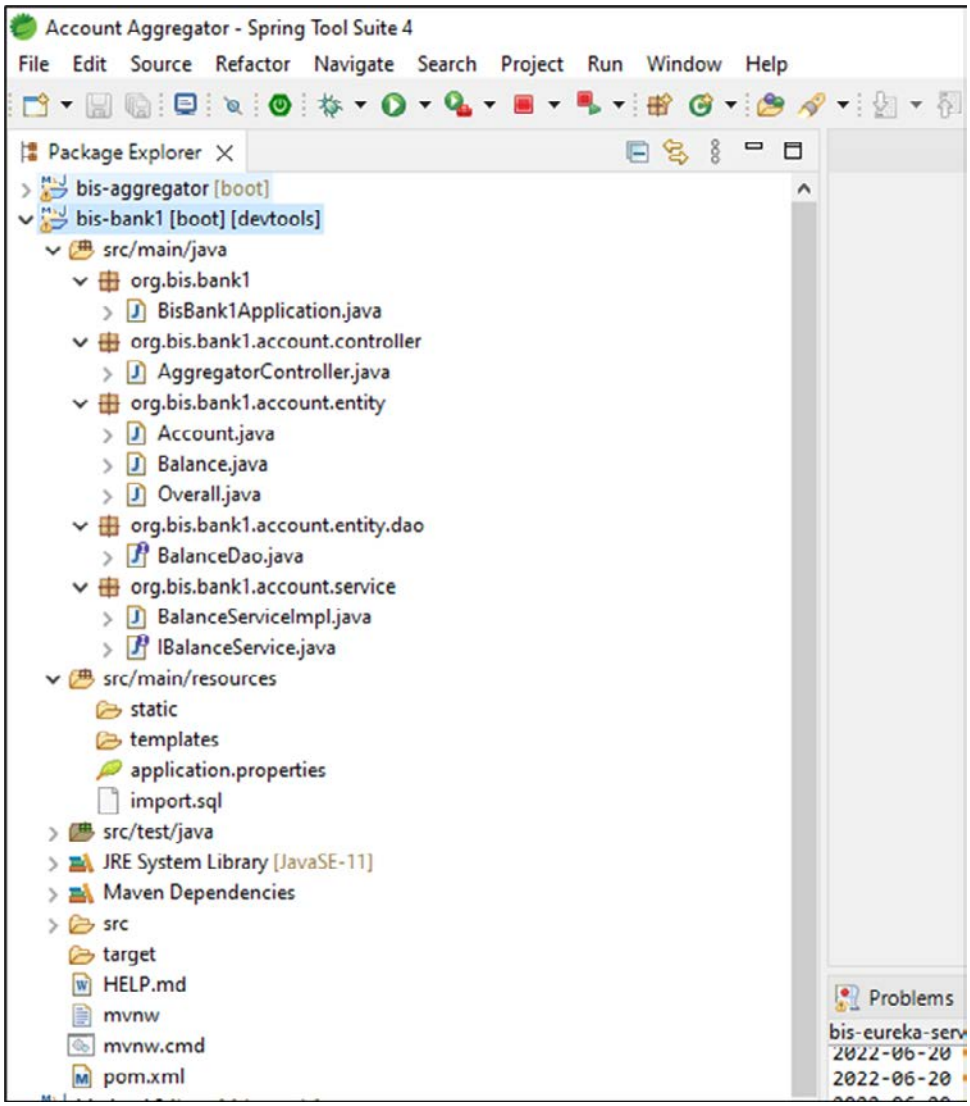
Account aggregator project Graph 10



On the other hand, each bank has its own implementation project similar to the aggregator's implementation project. However, it does have an implementation and connection to an in-memory database for testing purposes. Therefore, it requires a **DAO class**, and **import.sql** file for data loading.

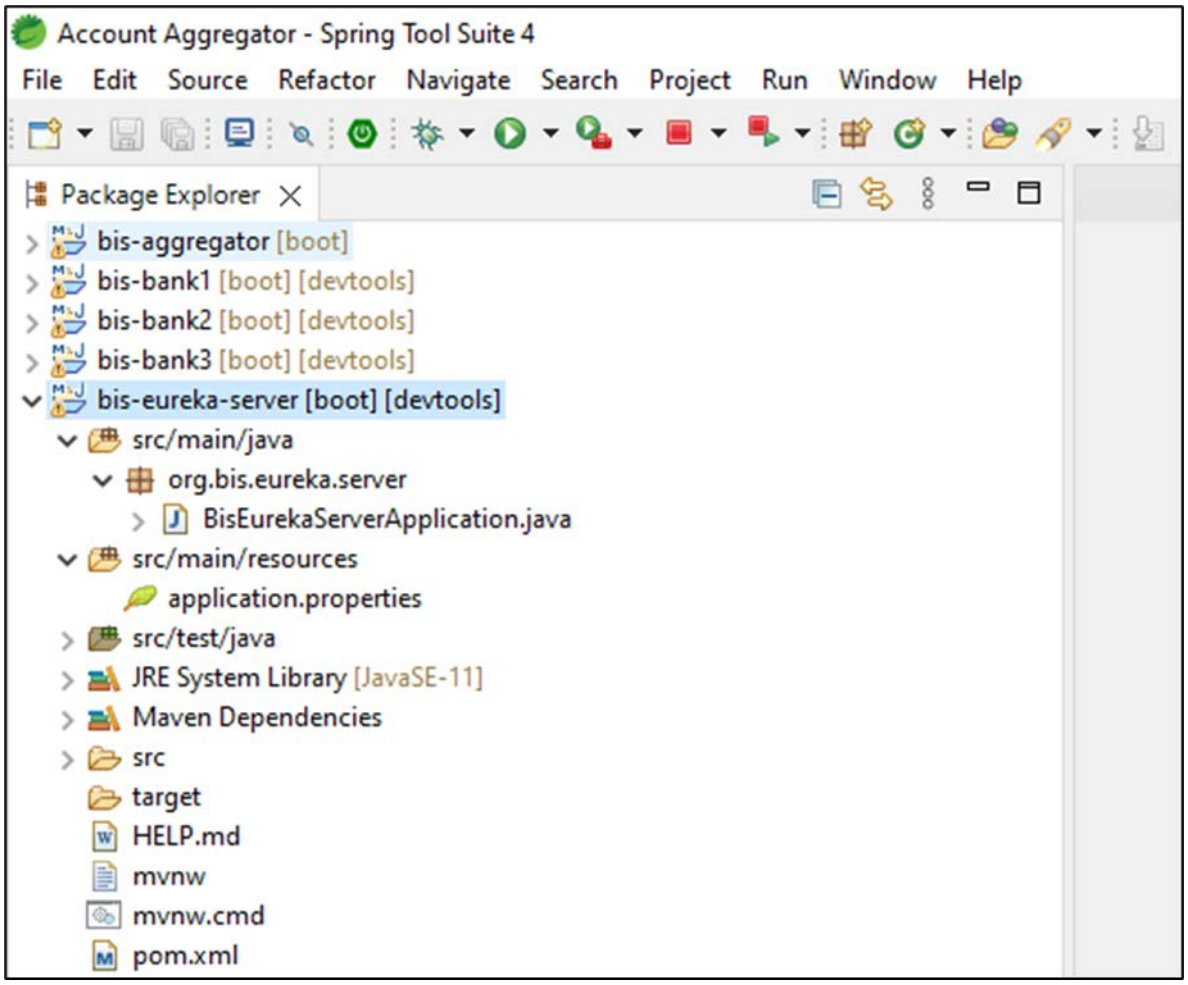
Bank 1 project

Graph 11



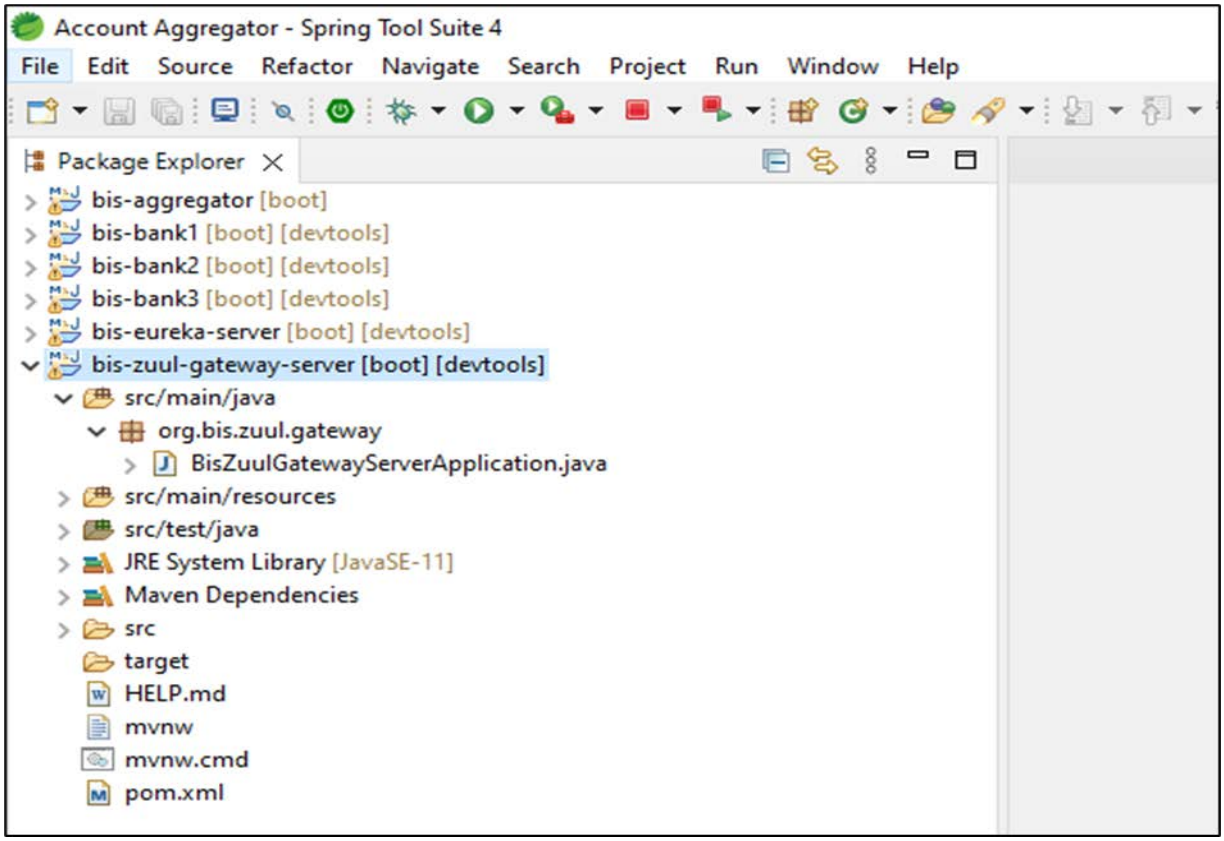
The Eureka server does not have custom implementations since its function is to register the microservice instances and to be able to monitor them. As it does not add any additional logic, it only contains one class for its startup.

Eureka server project Graph 12



Finally, similarly to Eureka Server, **Zuul** as an API gateway is just a server that serves as the main entrance for all requests and as the API publication route. Therefore, it requires no further implementation, only specific configurations.

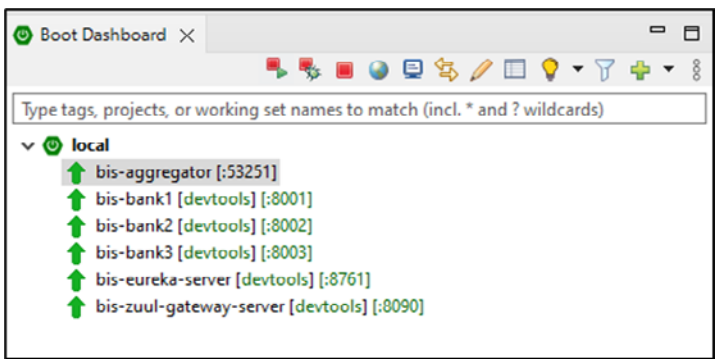
Zuul gateway project Graph 13



6.4 Testing

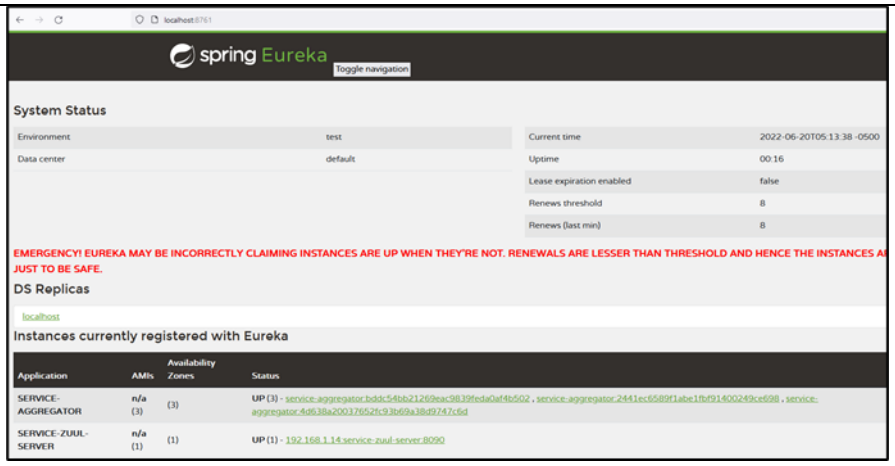
To start, the user must initialise each of the projects previously explained. First, the **Eureka** server, then the **Zuul** and finally the **Aggregator** (three instances, for example). Then, the user starts the projects for the three banks. Finally, the user verifies that they are all active.

Testing Graph 14



When entering the Eureka server, the three instances of the aggregator, as well as the API gateway, must be visible.

Eureka server Graph 15



Finally, the user enters the path http://localhost:8090/api/aggregator/list/42848906 in a web browser. Then, the aggregator displays the results of the three added banks.

JSON result

Graph 16

7 Conclusions

Data-sharing is one of the main pillars of the open banking initiatives that are emerging in financial services. Innovations include the involvement of third-party providers, also known as payment service providers, which facilitate access to banking records with users' consent. Data-sharing promotes transparency in a digital society, and supports high levels of reciprocity and cooperation in the financial ecosystem.

The results of a survey of CGIDE central bank members show that there is a common interest in implementing data-sharing with the aim of increasing efficiency and promoting competition within their ecosystems. The main challenges are coordination among participants, standardisation and technological infrastructure.

The scheme for data-sharing is largely determined by whether a rigid or flexible regulatory framework is preferred. The greatest challenges in selecting an adequate model are: establishing where the data will be stored, who the consumers are and which communication interfaces to use. This report presents three models: centralised, decentralised and trust – and develops the user interactions and their data flows.

Additionally, the report presents account aggregator functionality and possible arrangements for implementation in the open finance ecosystem. The report shows the successful implementation of a demo based on a microservices architecture that promotes high availability, scalability and resilience. The designs and results (JSON) are part of the report. The implementation of an aggregator uses Java-based frameworks and API REST. The use case was based on the accessing of personal account balances by data consumers, which were made available by three banks (also simulated in the demo).

Other organisations, mainly central banks, participated through presentations about their national experiences. These were the cases of Australia, Brazil, India, Korea and the United Kingdom. Additionally, Raidiam Services Limited presented its experiences and perspectives as a private technology provider for open banking and data-sharing initiatives.

Annex A: Survey on API standards for data-sharing

As an initial step, the CGIDE TTF APIs for data-sharing conducted a survey to collect relevant information from the eight participating central banks in this group (Argentina, Brazil, Canada, Chile, Colombia, Mexico, Peru and the United States). The objective was to collect information on considerations around the implementation of APIs for the secure and effective sharing of customers' data between financial institutions, fintech firms and certified third parties.

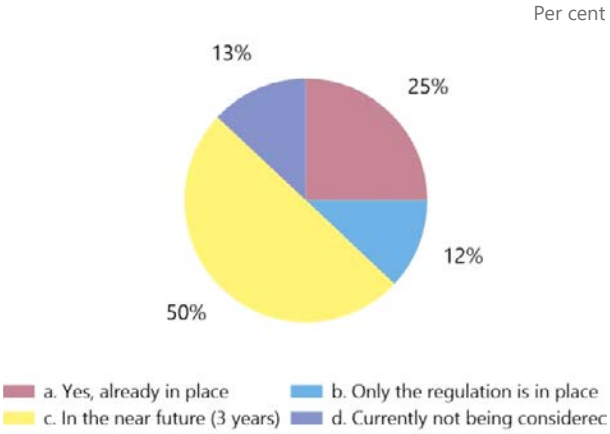
The design elements covered in this survey include the types of entity that would be part of the data-sharing process, design considerations for API standards, the type of shared information and its use, and the main drivers for implementing data-sharing. The information served as a basis for discussion on the technical requirements for the implementation of data-sharing.

The data from the survey show that central banks in the group are interested in open finance, and specifically in data-sharing. Most of the participants have already implemented data-sharing as part of their open finance initiatives or plan to implement it within the next three years (Graph 17, left-hand panel). Most participants consider that increasing efficiency and promoting competition are the main drivers for adopting and implementing data-sharing in their respective jurisdictions (Graph 17, right-hand panel). Other drivers are the need to regulate access to data, improve financial inclusion, promote innovation and improve data security and privacy, among others.

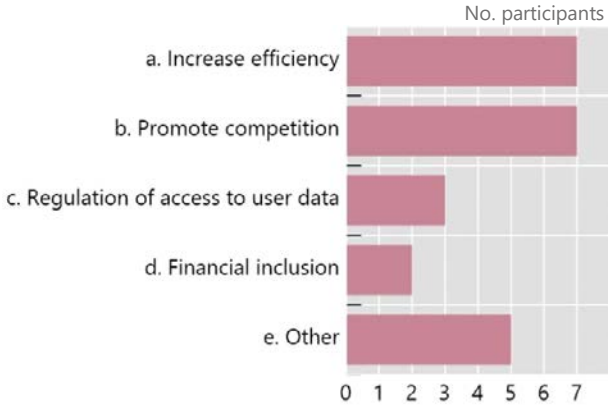
Interests and motivations for data-sharing

Graph 17

Is your country considering data-sharing as part of its open finance initiatives?



What would be the key drivers for data-sharing adoption/implementation?



Source: Central bank survey

As data-sharing can adopt a model depending on the needs and desired characteristics of each country, the results do not show a particular interest in any specific model. Half of the participants indicated that they would be interested in adopting a centralised model, while three other participants would consider a decentralised model. Another important consideration is the need to restrict the use of the information obtained within an open finance and data-sharing ecosystem.

Having an API standard is a major step in implementing data-sharing. Only one participant already has a standard in place in the financial system for this purpose, while six participants plan to introduce one in the next three years.

All participants agree that financial institutions should participate in the ecosystem. Most of them consider that banks should participate, while others indicate that PSPs and e-money issuers, among other financial institutions, should also participate. Six participants also state that fintechs should be part of the

ecosystem. Some participants mentioned payment card providers, public sector entities and bigtechs, among other entities that could participate in data-sharing.

The types of information considered for data-sharing include product information, deposit and other account information, and transaction and payment history. This information can have multiple purposes, and many participants consider that it could help to provide loans, access to insurance and the purchase of other financial services. Further purposes noted in the survey were access to investment services and savings accounts, the transfer of transactional information to financial entities and third parties, know-your-customer (KYC), and the creation of new services with transaction data, payments initiation and account aggregators, among others.

The adoption of data-sharing using APIs comes with its own challenges. The main ones are having the necessary technological infrastructure for its implementation, standardisation and coordination among participants in the ecosystem. Six participants also consider that defining an API standard is an important challenge that goes together with the challenge of coordination. Other challenges noted by some of the participants are issues relating to the current development of connectivity and telecommunications in their jurisdictions, cybersecurity, regulation and whether market participants would adopt a voluntary approach.

Considerations regarding the development of APIs and associated standards also entail technology choices. The most popular messaging format among survey participants is JSON. For API specification, four participants prefer the Open API standard, while some other participants might consider RAML, AsyncAPI and REST standards. With respect to API security, participants mentioned several protocols as options in the survey, of which OAuth, TLS, VNP and PKI certificates are the most popular technologies to be considered by participants.

Annex B: Data-sharing regulatory models

According to international experiences such as those in Australia, Brazil, India, Korea and the United Kingdom, central banks can establish flexible, rigid or no regulatory frameworks for the promotion of data-sharing. Hence, a free market-led approach or one controlled by regulation emerge as viable alternatives to be analysed by participating central banks. Smith and Lehane (2022) classified the regulatory models as follows.

Market-driven

In this type of model, regulatory frameworks do not exist or are smooth, with no requirements regarding data-sharing. Third-party providers (TPP) do not have to deal with complex privacy frameworks or costly compliance rules. Open banking in this free market version allows TPPs to access banks' APIs to provide new services to customers. However, this usually presents a risk for banks since they retain the responsibility for handling the data. The decentralised model is a good example of a trust ecosystem model that tries to promote a market-driven approach by seeking to avoid too much intervention.

Regulatory-driven

According to international experience, an approach based on regulation is appropriate when there are competition issues in the banking market. In that context, regulation is mandatory to open up the market so that TPPs and small financial institutions can access banks' APIs with the consent of the data owner. This model responds to the problems of API consumption overload by the big banks with the largest volumes of data. These problems could worsen if such banks do not act on a reciprocal basis with smaller institutions. Centralised and hybrid models are examples of regulatory-driven approaches. The trust ecosystem model can also promote some regulation by defining the standards that participants in the financial sector must follow for the implementation of APIs.

Annex C: Lessons learned from other initiatives

The CGIDE TTF hosted webinars with speakers from various jurisdictions on their data-sharing initiatives. The objective was to learn about the model implemented, the considerations taken into account for the choice of model, the lessons learned from the implementations and the environment. A second objective was to open channels of communication between members of the CGIDE TTF and the organisations in charge of the initiatives from the different jurisdictions. The invited guests were the National Payments Corporation of India, the private company Raidiam and the authorities responsible for the initiatives in the Australia, Brazil (which provided a box for this report), Korea and the United Kingdom.

Australia

Australia established the Consumer Data Right (CDR) in 2010, a pioneering regulatory framework for consumer data privacy spanning multiple sectors. CDR aims to give consumers the right to use data about them (product and consumer data) held by the country's businesses, for their own benefit. The design of CDR allows for its implementation in all sectors of the economy, starting with banking (open banking) and later in the energy and telecommunications sectors.

The authority developed CDR rules and standards in parallel with a sector-agnostic approach. Only 20% of the rules are specific to the banking and energy sectors and apply to the data sets and who are the target consumers. The rules detail the consent, authorisation and accreditation processes. The main actors under the CDR framework are the consumer, the data holder, and the accredited data recipient (ADR). The consumer is the individual or small business. The data holder is the provider that holds the consumer's data (eg banks or energy providers). Authorised deposit-taking institutions (ADIs) are banks. Data holders are required to provide an infrastructure to enable product and consumer data requests, arrange general data on the products they offer (rates, discounts etc), transmit the information securely in machine-readable format and manage consumer authorisations to share their data. ADRs receive consumer data after the consumer provides consent for the specific purpose requested.

The phased implementation model started in July 2020 with the four largest banks, and simple products such as savings and credit card accounts. The next stage considered the other banks and fintechs. Currently, implementation of open banking under the CDR framework is making excellent progress, with 99% of consumers covered and 112 active banks.

The key considerations of the Australian implementation model are that it has a centralised model based on registration. Prior to connecting to data holders and receiving data, ADRs must register through the trusted central authority. After registration, it is a distributed model because the ADRs connect directly to the data holders. Furthermore, it is prescriptive in the sense of making data-sharing mandatory for large participants in conjunction with establishing the minimum technical standards necessary to make the process consistent and secure. Finally, the model takes into account a broad approach to the economy to be applied to different sectors.

India

The National Payments Corporation of India (NPCI) is a non-profit company responsible for operating retail payment and settlement systems in India, currently with 66 banking and non-banking partners. The NPCI has developed a multi-rail network and a diversified product suit to cater to different use cases (P2P, B2B, B2C etc), starting in 2008 with the National Financial Switch (NFS) product, a single rail platform. In 2010, it launched Immediate Payment Service, a real-time credit platform. In 2016, they moved to a platform strategy and launched the Unified Payments Interface (UPI), an important milestone for the digitisation of payments, financial inclusion and the start of the open banking ecosystem.

The open banking model covers account information services (AIS) under the account aggregators (AAs) initiative and payment initiation services (PIS) through the UPI real-time payment platform. An important factor for the success of the model is India Stack, a universal framework of open APIs for the development of multiple use cases, including biometric authentication mechanisms, a national digital identity system through Aadhaar, a digital retail payment product (IMPS, UPI and BHIM, among others), and a centralised consumer consent framework for data-sharing through the AA model.

An inter-regulatory decision created the AA framework.[22] Account aggregators are RBI-authorised institutions that enable data-sharing between financial information providers (eg banks and insurance providers, among others) and financial information users (including fintechs and bigtechs) in a manner that ensures that the information shared is consensual, for a limited purpose and for a limited time. AAs share data requested through UPI after obtaining user authorisation. AAs cannot access, store or sell the data, only collect and transfer it. The user can revoke consent to, or change the access period, among other things.

Korea

The Financial Services Commission of Korea's definition of open banking implies the opening of banks' financial payment systems to fintechs. The Korea Financial Telecommunications and Clearing Institute (KFTC), established in 1986, operates the country's retail payment networks and was responsible for designing the open banking model, consisting of an integrated platform in which KFTC serves as a data aggregator for fintechs and banks. That is, it can collect data including customer-authorised data through APIs, screen extraction or other means; and can offer services directly to the customer or to other parties providing services to the customer.

Entities joining the platform are subject to the KFTC's criteria for financial soundness and operational and risk management capability. As of September 2021, 68 fintechs and eight credit card companies were part of the ecosystem, and there were 19 direct participants on the existing payment system network side. The fintechs carry out authorisation, authentication and information request processes through centralised APIs on the KFTC platform.

Raidiam

Raidiam, a data-sharing ecosystem specialist, points out that the underlying model of successful open banking ecosystems around the world has no central functionality through which data flow. There is, however, a core function that provides capabilities for identifying third-party providers (TPPs) and financial services institutions. This model, building on initiatives in Australia, Brazil, New Zealand and the United Kingdom, as well as recent studies in Canada, covers payment initiation and data-sharing services. The model does not need to have a national digital identity system to achieve a successful data-sharing model.

The main component of this model that enables data-sharing is a trust framework that includes identification, authorisation and authentication processes – and which involves registration, security and standards requirements. These processes become complex in ecosystems with multiple sources and participants. In such an environment, a model is successful when the trust framework consists of a central platform that enables initial registration between service providers and data providers, but data-sharing is still distributed. Regulators are part of this ecosystem by specifying whom they allow to participate, while data and service providers comply with the technical requirements to ensure secure connection with each other and with consumers. In addition, there is a need for standards, whereby each bank implements

[22] The authorities that created the AA framework were the RBI and other regulators including Securities and Exchange Board of India (SEBI), Insurance Regulatory and Development Authority (IRDA), and Pension Fund Regulatory and Development Authority (PFRDA) through and initiative of the Financial Stability and Development Council (FSDC)

standardised APIs, resulting in a distributed but consistent and secure data-sharing model. To achieve consistency, it requires robustness testing and a certification programme by regulators.

United Kingdom

The United Kingdom was a global pioneer in the adoption of open banking, beginning in 2017, when the Competition and Markets Authority (CMA) ordered the CMA9[23] to set up an Open Banking Implementation Entity (OBIE). The OBIE would act as governance, systems and data architect, and standards setter. The open banking model allows consumers and SMEs to share their banking and credit card transaction data securely with third parties and to initiate payments directly from their payment accounts to the beneficiary's bank account, without the need for cards. OBIE developed standardised API specifications, starting with core APIs. The specifications and guidelines include detailed sequence/interaction diagrams, customer experience guidelines, security profiles and operational guidelines, in addition to developer-focused guidelines with sandboxes for testing, comprehensive documentation and other support to help create a dynamic software ecosystem.

The implementation shows the importance of maintaining regular dialogue and collaboration for iterative development of APIs, learning from what does not work and adapting and implementing solutions. It is also easier when governance and leadership is centralised. It is advisable to have common API standards and engineering systems (sequence diagrams, flows, data model etc) under centralised oversight. It is essential to ensure regional cooperation on governance, harmonisation of legal, regulatory and technological standards, and adherence to guidelines on user experience, mandatory usage approach, clear rules of the game and codes of conduct.

The United Kingdom has been working towards an open finance model and has recently developed a conceptual model focused on SMEs for the use case of obtaining better credit. In addition to the recommendations for open banking, for an open finance implementation, the United Kingdom adds that digital identity solutions are important, including for the development of an open economy. The informal economy can limit available data, reliability and quality. It may need to start with small and easily available use cases, with support from the government.

[23] The CMA9 is the name of the nine largest current account providers in Great Britain and Northern Ireland.

Annex D: Members of the Consultative Group on Innovation and the Digital Economy (CGIDE)

Members

BIS Innovation Hub	Miguel Díaz (Chair)
Central Bank of Argentina	María Daniela Bossio Mara Misto Macias
Central Bank of Brazil	Angelo Duarte Fabio Araujo
Bank of Canada	Eric Santor
Central Bank of Chile	Pablo Furche
Central Bank of Colombia	Carlos Arango
Bank of Mexico	Othón Moreno Angel Salazar
Central Reserve Bank of Peru	Milton Vega Felix Santos Jushua Baldoceda
Board of Governors of the Federal Reserve System	Francesca Carapella Peter Lone
Bank for International Settlements	Alexandre Tombini Carlos Cantu (Secretary) Carolina Velasquez (Secretary)

Observers

Bank for International Settlements	Jaime Cortina Jon Frost Christian Upper

Annex E: Members of the Technical Task Force (TTF) of the CGIDE

Members

Central Reserve Bank of Peru	Milton Vega (Chair) Felix Santos Jushua Baldoceda
Central Bank of Argentina	Mara Misto Macias Gustavo Pereyra Mariano Vazquez Silvina Ojeda
Central Bank of Brazil	Fabio Araujo Saulo Medeiros de Araujo
Bank of Canada	Eric Santor Scott Hendry Alin Dan
Central Bank of Chile	Claudia Sotelo Enrique Gonzalez
Central Bank of Colombia	Carlos Arango Samuel Gutiérrez
Bank of Mexico	Othón Martino Moreno González Salazar Sotelo Ángel Daniel Garrido Delgadillo Aurelio Martín Reyes Montoya
Board of Governors of the Federal Reserve System	Francesca Carapella Peter Lone Franklin Ervin
BIS Innovation Hub	Miguel Díaz
Bank for International Settlements	Alexandre Tombini Jon Frost Carlos Cantú (Secretary) Carolina Velasquez (Secretary)

References

Arcitura (2022a): "Decoupled service API", https://patterns.arcitura.com/service-api-patterns/fundamental-service-api-patterns/decoupled-service-api.

——— (2022b): "Service agent", https://patterns.arcitura.com/service-api-patterns/service-agent.

——— (2022c): "Service facade", https://patterns.arcitura.com/service-api-patterns/service-facade.

——— (2022d): "Service API proxy", https://patterns.arcitura.com/service-api-patterns/fundamental-service-api-patterns/service-api-proxy.

——— (2022e): "Service API centralisation" https://patterns.arcitura.com/service-api-patterns/fundamental-service-api-patterns/service-api-centralization.

——— (2022f): "Endpoint redirection", https://patterns.arcitura.com/service-api-patterns/fundamental-service-api-patterns/endpoint-redirection.

Bank of Mexico (2020): "CIRCULAR 2/2020", https://www.banxico.org.mx/marco-normativo/normativa-emitida-por-el-banco-de-mexico/circular-2-2020/%7B4FDD6B5E-8DFA-F095-6325-68C388AAEAA0%7D.pdf

Basel Committee on Banking Supervision (BCBS) (2019): *Report on open banking and application programming interfaces*, November.

Bergmann, J, O Bott, D Pretschner and R Haux (2007): "An e-consent-based shared EHR system architecture for integrated healthcare networks", *International Journal of Medical Informatics*, vol 76, no 2–3, pp 130–6.

Brodsky, L and L Oakes (2021): "Data-sharing and open banking", McKinsey & Company, www.mckinsey.com/industries/financial-services/our-insights/data-sharing-and-open-banking.

Central Bank of Brazil (2021a): "INSTRUÇÃO NORMATIVA BCB Nº 95, DE 14 DE ABRIL DE 2021", https://in.gov.br/web/dou/-/instrucao-normativa-bcb-n-95-de-14-de-abril-de-2021-314703508.

———(2021b): "INSTRUÇÃO NORMATIVA BCB Nº 184, DE 12 DE NOVEMBRO DE 2021", https://in.gov.br/en/web/dou/-/instrucao-normativa-bcb-n-184-de-12-de-novembro-de-2021-359803029.

Consultative Group on Innovation and the Digital Economy (CGIDE) (2020): *Enabling open finance through APIs*, December, www.bis.org/publ/othp36.pdf.

——— (2021): *Enabling open finance through APIs: report on payment initiation*, September, www.bis.org/publ/othp41.pdf.

Deepak, V (2020): "Comparing XML and JSON: what's the difference?", *Techwell*, 27 April, www.techwell.com/techwell-insights/2020/04/comparing-xml-and-json-what-s-difference.

Kerry, D (2022): "REST (Representational State Transfer)", TechTarget, www.techtarget.com/searchapparchitecture/definition/REST-REpresentational-State-Transfer.

Madden, N (2020): *API security in action*, Manning Publications.

Nolle, T (2021): "API lifecycle management", TechTarget, www.techtarget.com/searchapparchitecture/definition/API-lifecycle-management

OAuth 2.0, https://oauth.net/2/ (accessed 1 June 2022).

Open Finance Brasil GT Security (OFBIS GT Security): "Open Finance Brasil Financial-grade API Security Profile 1.0 Implementers Draft 3", retrieve from https://openbanking-brasil.github.io/specs-seguranca/open-banking-brasil-financial-api-1_ID3.html.

OpenAPI Initiative: "OpenAPI Initiative FAQ", www.openapis.org/faq (accessed 1 June 2022).

OpenID (a): "OpenID Connect FAQ and Q&As", https://openid.net/connect/faq/ (accessed 1 June 2022).

———(b): "FAPI – Financial Grade API", https://fapi.openid.net/ (accessed Jun 1,2022).

Press Information Bureau, Government of India (2021): "Know all about account aggregator network – a financial data-sharing system", 10 September, https://pib.gov.in/PressReleaseIframePage.aspx?PRID=1753713.

Portal do Open Banking Brasil: Área do Desenvolvedor, https://openbankingbrasil.atlassian.net/wiki/spaces/OB/overview (accessed 1 June 2022).

———"Padrões" https://openbankingbrasil.atlassian.net/wiki/spaces/OB/pages/9634480/Padr+es (accessed 1 June 2022).

Semantic Versioning 2.0.0: "Semantic Versioning Specification (SemVer)", https://semver.org/

Support Centre for Data-Sharing (SCDS) (2022): "What is data-sharing?", https://eudatasharing.eu/what-data-sharing.

Smith, S and D Lehane (2022): "Blurring the lines: creating an open banking data-sharing ecosystem", *Deloitte*,www2.deloitte.com/ie/en/pages/financial-services/articles/Creating_an_Open_Banking_data_sharing_ecosystem.html.

Sonin (2022): "The 4 Ps innovation model: POC, prototype, pilot, production", https://sonin.agency/the-4ps-innovation-model-poc-prototype-pilot-production/.

Zachariadis, M (2020): *Data-sharing frameworks in financial services: discussing open banking regulation for Canada*, Global Risk Institute, August.