



API – What is it and how will this influence our standardisation world

Lisa O'Connor

October 17 2018

Types of APIs



Open Banking Approaches

Scope

- At first sight open banking initiatives appear similar however in reality there are differences.
- PSD2 is probably the best known regulation however it's remit is limited to payment accounts.
- One of the concerns in the FinTech community is that the banks may stick to this limited scope thereby limiting the potential of access to full customer data.
- While PSD2 is about accessing account data and making payments the UK, HK and AU scope may be broader in terms of financial products and data sets and certainly in initiating payments.

Legislative

- Most countries have seen privately run initiative between banks and FinTech companies to enable e-commerce / e-banking platforms to provide services to corporates.
- In the past, supervisory authorities have limited their guidance to internet security recommendations. In recent years, most countries (US is the outlier) have taken initiatives to accelerate competition and provide a better customer banking experience.

Regulator in Charge

- In countries with legislative and regulatory intervention it has been the banking or payment regulator pushing for these changes.
- Regulatory acts or recommendations are attempting to create a balance between supervision and new entrants to the payment markets.

Open Banking Approaches

Security

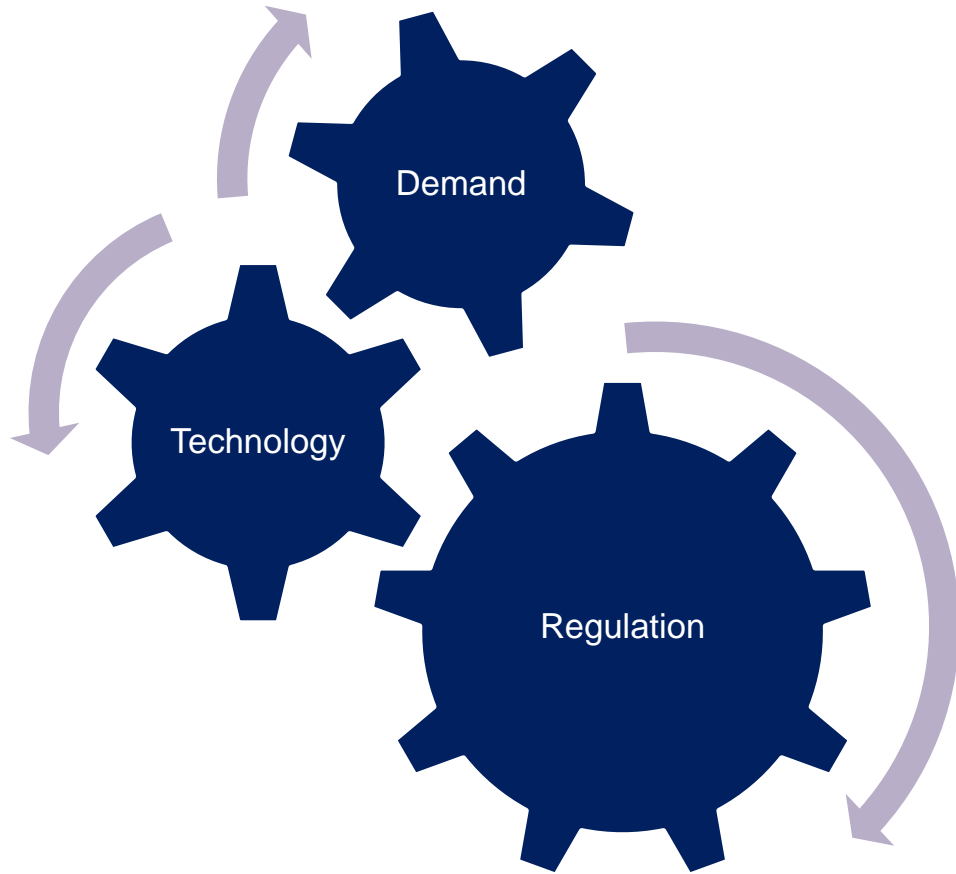
- There is a lot of consistency in the security methods proposed.
- Levels of security will typically be in the high ends for protection over the internet to guard for confidentiality and integrity of financial data.
- Security exceptions to provide increase usability and convenience could also be foreseen.
- 2 factor authentication and “qualified” FinTechs are also expected (like in EU by a qualified trust provider - eIDAS complaint).

Technology

- All initiatives have very similar technology / protocols. HTTPS message encryption, RESTful Design, JSON syntax, OAuth 2.0 authorization, etc.
- **Main differences will be found in the business payload.**



API adoption | What are the trends impacting API adoption within the financial industry? (1/2)

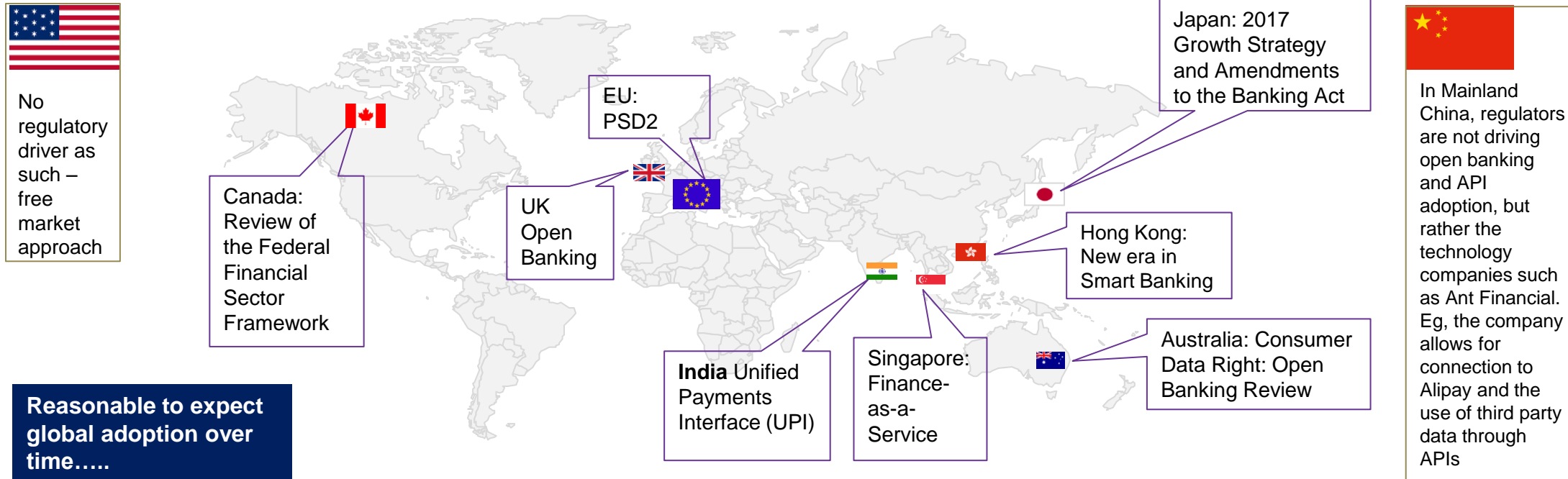


- Consumer demand for always on, instant digital services via mobile device
- New technologies fuelling consumer expectations
- New entrants/fintechs/GAAFA agile and well positioned to address consumer needs
- **Open Banking** regulation in many locations, initially in the UK and EU, now in Japan, HK, Singapore, Canada and elsewhere
- Emergence of new **instant payments** schemes leading to a possible convergence with APIs as these are fast, always-on, high throughput
- Entities from the card ecosystem looking to retain existing customer base while expanding into new customer segments with new payment solutions



API adoption | What are the trends impacting API adoption within the financial industry? (2/2)

- **Regulators** across the globe are gradually moving towards open banking albeit with local nuances
- Open banking: levelling the playing ground for new entrants, together with providing the end customer with choice
- In geographies where regulation is not the primary driver, market forces (consumer demand, competition, technology evolution) play a strong role
- Evolving landscape: a growing list of countries indicating interest in the principles of Open banking (Israel, Russia, Malaysia...)



Ambition

JPMC is transforming its markets business to deliver an end-to-end digital experience. This is seen in the bank's plan to provide services such as corporate finance on-boarding, research, analytics, execution, post-trade, clearing and settlement and prime custody through its clients digital channel of choice which include voice, online, mobile APIs and third parties

What makes them stand out

JPMorgan Chase has clear digital strategies in place across all of its business lines. The bank plans to provide corporate customer with access to its digital platforms – Chase For Business, Chase Connect and JPMorgan Access through APIs. The bank has already rolled out various digital offerings across all of its businesses.

Key learnings

JPMC is applying learnings from the retail segment for expansion in the wealth management segment

The securities industry faces similar drivers and constraints to retail. Addressing the customer demand for digital capabilities JPMC is applying the same technical (digital) strategy across all business segments

Key Products

Chase Developer

- APIs on the Chase Developer site are mainly used for retail.
- The APIs are broadly categorised into payments, data aggregation and payables and receivables.
- Access to the developer ecosystem is currently by invitation only and limited to developers and businesses who have a relationship with Chase

JPMorgan Developer

"Your source for seamless access to JP Morgan data and capabilities"

- APIs on the JPMorgan Developer site are mainly used for access to the investment bank's data.
- The APIs fall into 5 categories – Research & Analytics, Pricing & Structuring, Execution, Post-Trade and Banking
- Focused on corporate and investment banking clients, with over 100 users, the majority of which are asset management clients. Registration required

"API's give us an opportunity to break down our large programs into reusable components. That also allows us to think differently about how we want to connect with our customers." *



Ambition

Goldman Sachs’s ambition is to redesign the company around APIs. The strategy involves a shift from viewing the company as based around financial products to being built around applications. This entails a new approach to sharing data. “We’re turning all of the activities at Goldman Sachs into APIs”

What makes them stand out

Built Data Lake (a data-centric ecosystem) which pools transactional, operations and reference data in one place to serve various business purposes and includes AI technology. Looking to leverage the Data Lake to **create new revenue streams via APIs** (Marquee platform)

Key learnings

- GS is monetising their data to provide value-added services to their clients
- API developer portal is password protected: closed community APIs

Key highlight: Data Lake

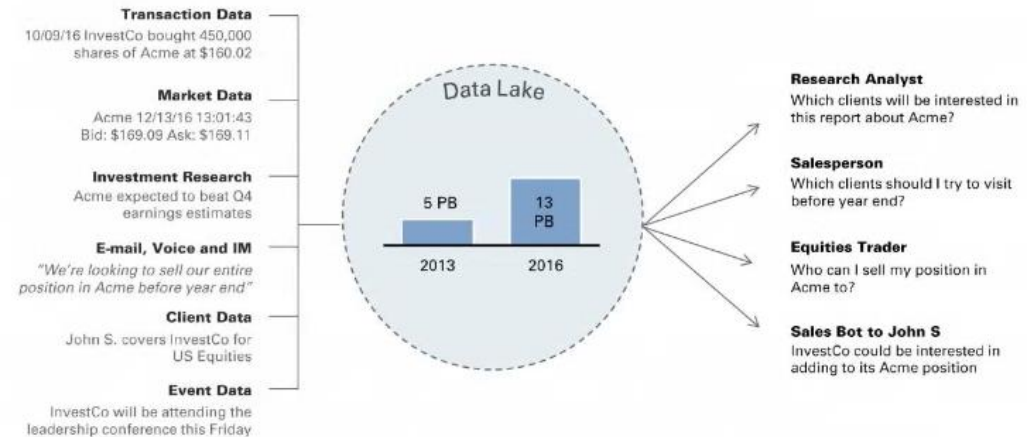
An API-based data-centric ecosystem, including AI, which pools transactional, operations and reference data in one place to serve various business purposes, e.g. applying machine learning to model/make query-able

GS is looking to leverage the Data Lake to **create new revenue streams via APIs in its Securities business**. Third parties could subscribe to audited data collected via Data Lake through an API feed to help them develop their trading/investment strategies

The infrastructure is built using open source components e.g. Hadoop, Spark and Hive

Data Lake enables entirely new data science capabilities

A single firm-wide data repository to generate new insights about clients and markets



Sources: Company website, Annual Report 2017



Future business model based on collecting data and pushing it through the analytics engine then making it available to internal and external clients.

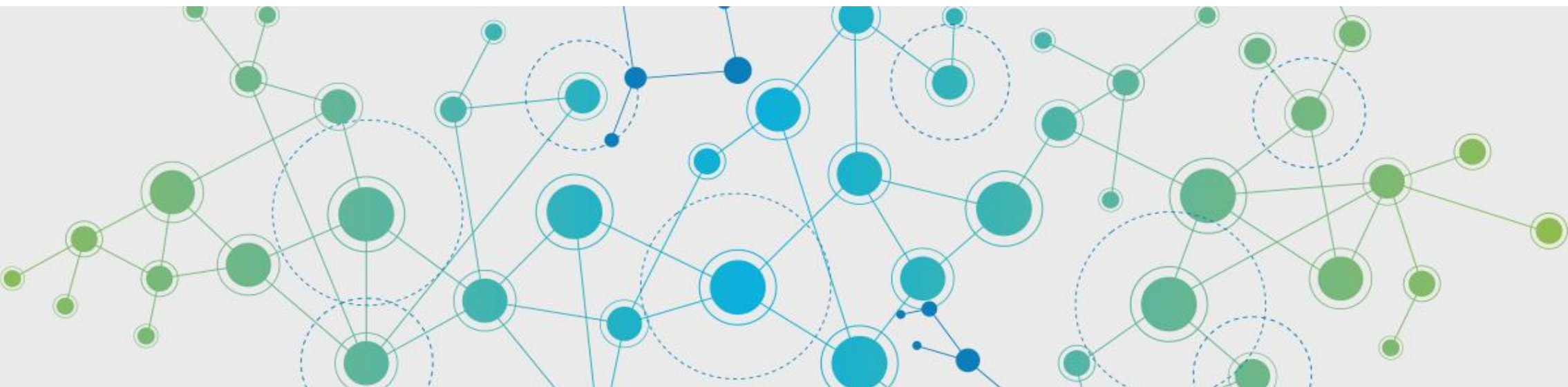
The Studio Platform

The Marquee platform allows for different businesses at Goldman Sachs to **open themselves to institutional clients in a secure and consistent way**. It essentially provides clients with direct access to the firm's internal risk management system. This is where clients can obtain information to make decisions without having to directly speak to GS sales teams.



ISO 20022 for API interoperability





“Whatever the technology, in multi-party networked business, participants need to agree on the meaning and content of shared data, business processes, roles and responsibilities. This is the domain of *business standards*”

ISO 20022 – Objective

To enable communication interoperability between financial institutions, their market infrastructures and their end-user communities



What is ISO 20022?

Single standardization approach (methodology, process, repository)
to be used by all financial standards initiatives

Recipe to create financial standards

Business / Conceptual

- Defines **business meaning** of financial concepts, e.g., 'Credit Transfer'

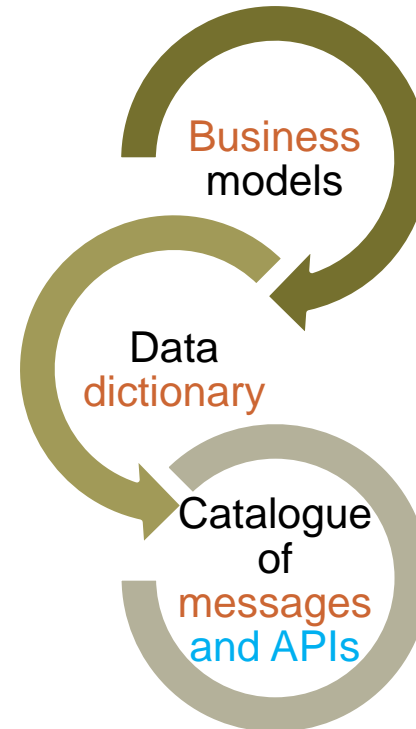
Logical

- Defines e.g. credit transfer **messages and APIs**, to execute the business process

Physical

- Defines physical **syntax**, e.g. XML **and JSON**

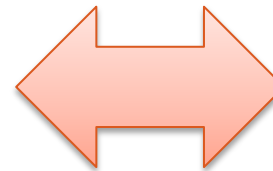
Body of content



Similarities and differences with the messaging paradigm

	Messaging	API
Architecture	Entire information set flows between all participants.	Central state maintained as web resources.
Interaction style	Point-to-point asynchronous messaging.	REpresentational State Transfer (REST); stateless operations access and manipulate centrally maintained resources.
Type of message	Messages cover many use cases and apply for different roles	Concise and focused set of API calls.
Development and maintenance	Robust development requiring predefined, precise maintenance cycles.	Rapidly changing, fast and simple implementation environment requiring agile development.
Types of services and applications	Highly automated back office applications.	Services on the edge, lightweight implementations.
Data format	Proprietary or eXtensible Markup Language (XML).	JavaScript Object Notation (JSON), OpenAPI

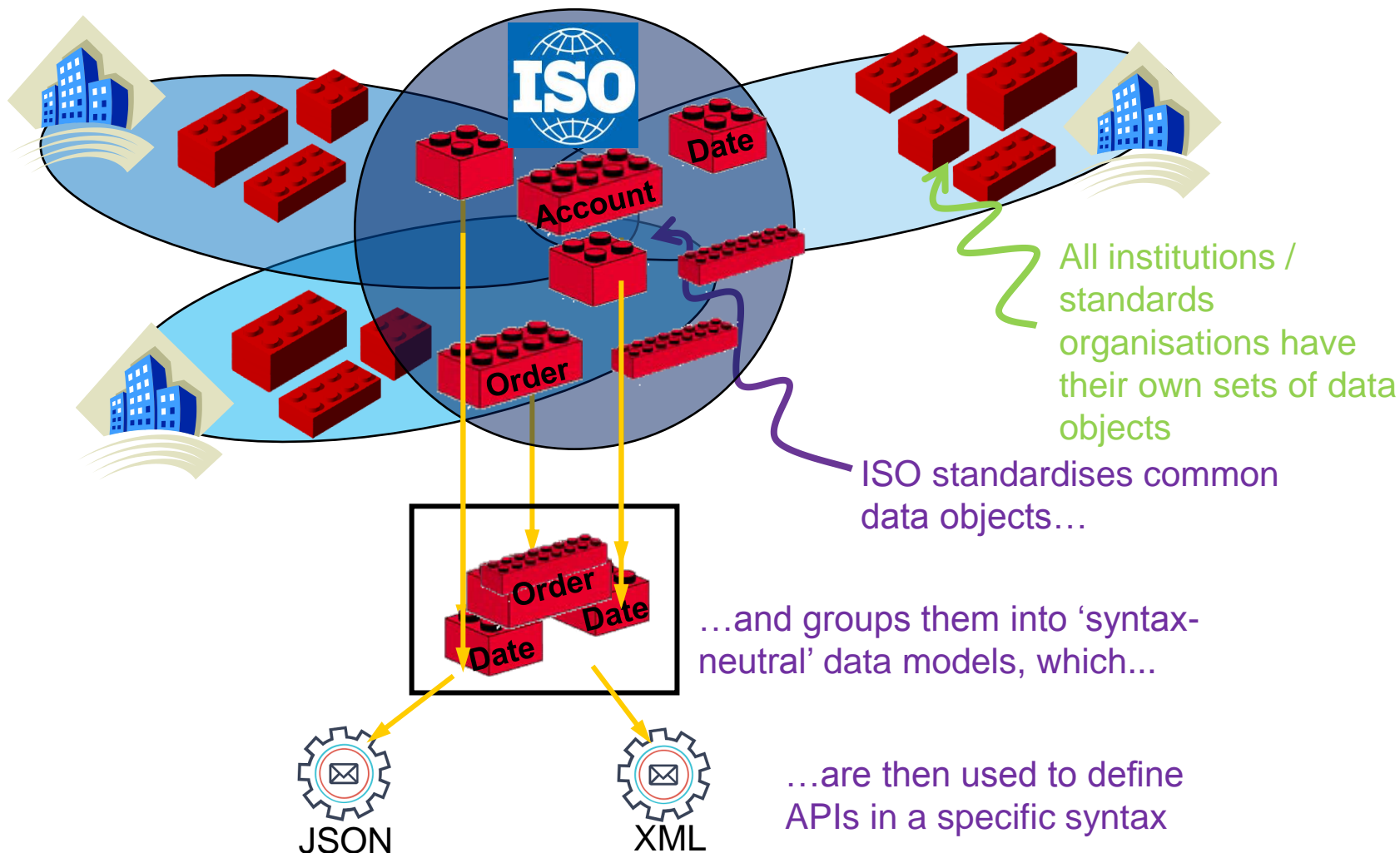
- Messaging standards are published centrally
- Strict annual review process
- ISO20022 messages are 'one-size fits all'



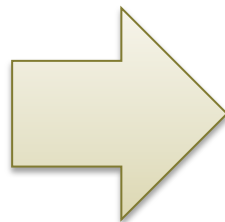
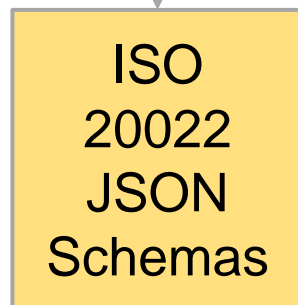
- APIs are decentralised
- More agile
- Fit for purpose



The building blocks



Leveraging ISO 20022 and off-the-shelf tools



Example of off-the-shelf API tools

Design

Swagger API
Framework

*For designing API according to
Open API Initiative principles*

API Designer

*For designing and testing API according to
RAML 0.8 and RAML 2.0 specs*

API
workbench

Test

Swagger UI

*For introspecting and
testing Open API*

SOAPUI

*For REST API
mocking*

POSTMAN

*API console for introspecting
(supports Swagger/RAML
imports) and testing APIs*

Document

Swagger API
Framework

RAML2HTML

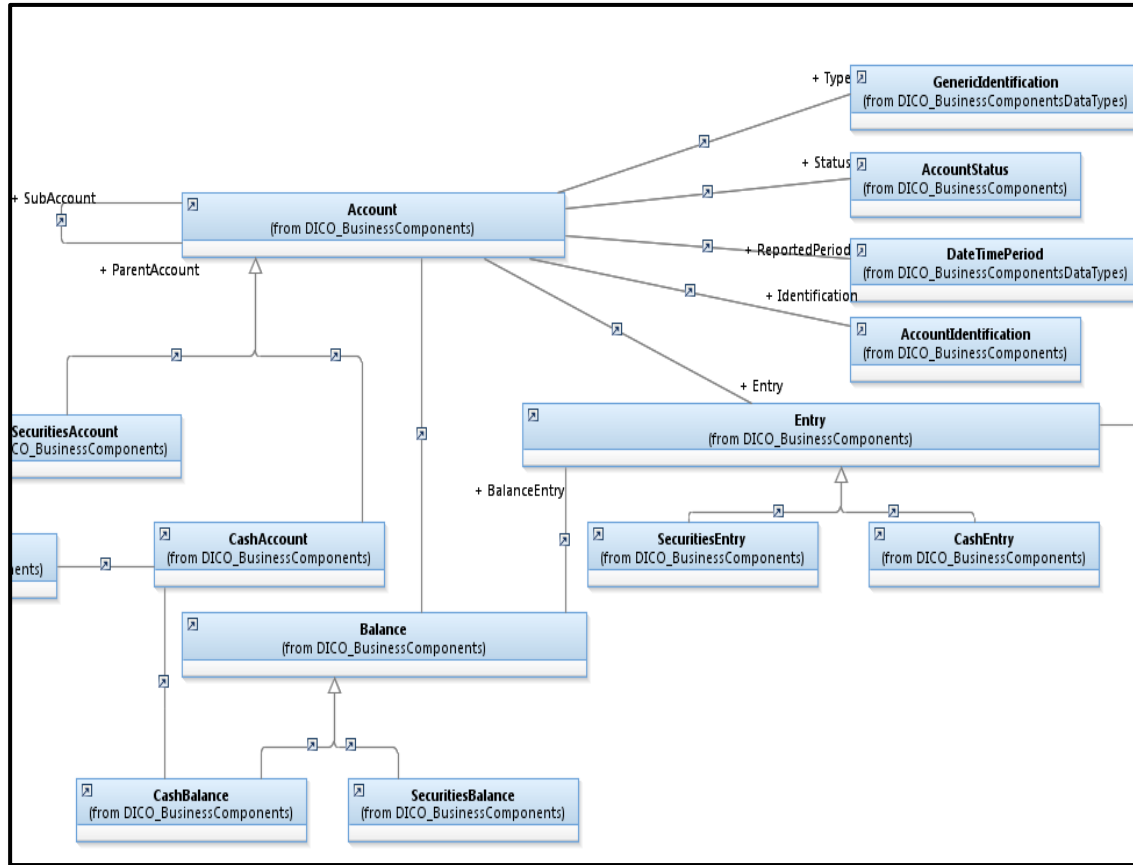
*Documentation tool that outputs a single HTML
page (based on a RAML definition)*



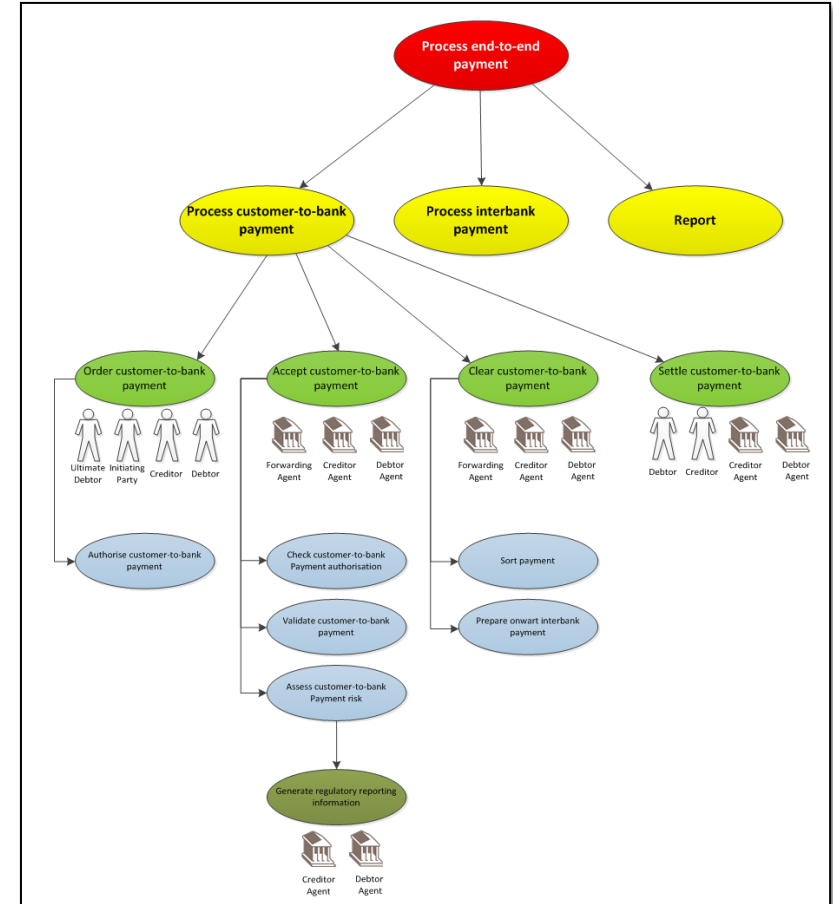
Using ISO 20022 for APIs



So there is a lot in the repository that we can reuse



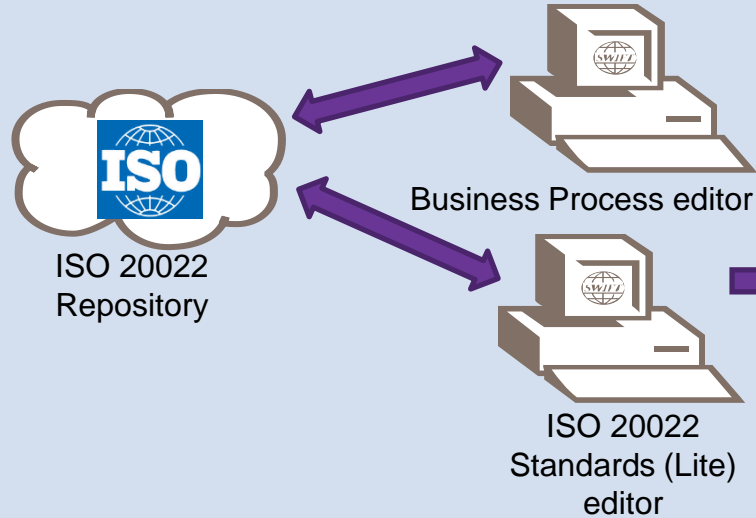
The business data



The business stories

ISO 20022 API Process and governance

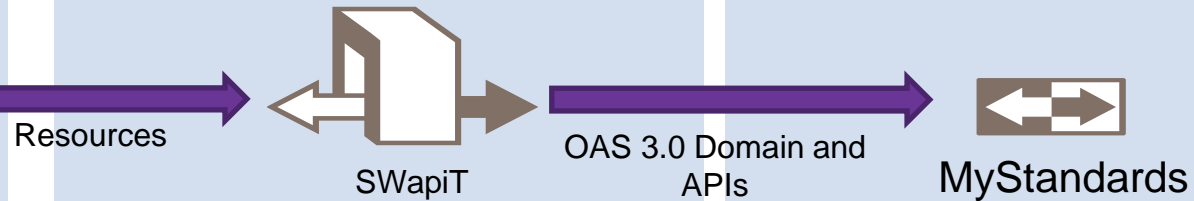
Resource definitions



- Business Transactions ('the stories')
- Resources ('the data')
- ISO 20022 Registration process and governance



Generic API definitions



- API specific information (**methods**, parameters, **endpoints**, URLs...)
- **generic** APIs
- OAS 3.0 file generation
- Governance by SWIFT or ISO 20022

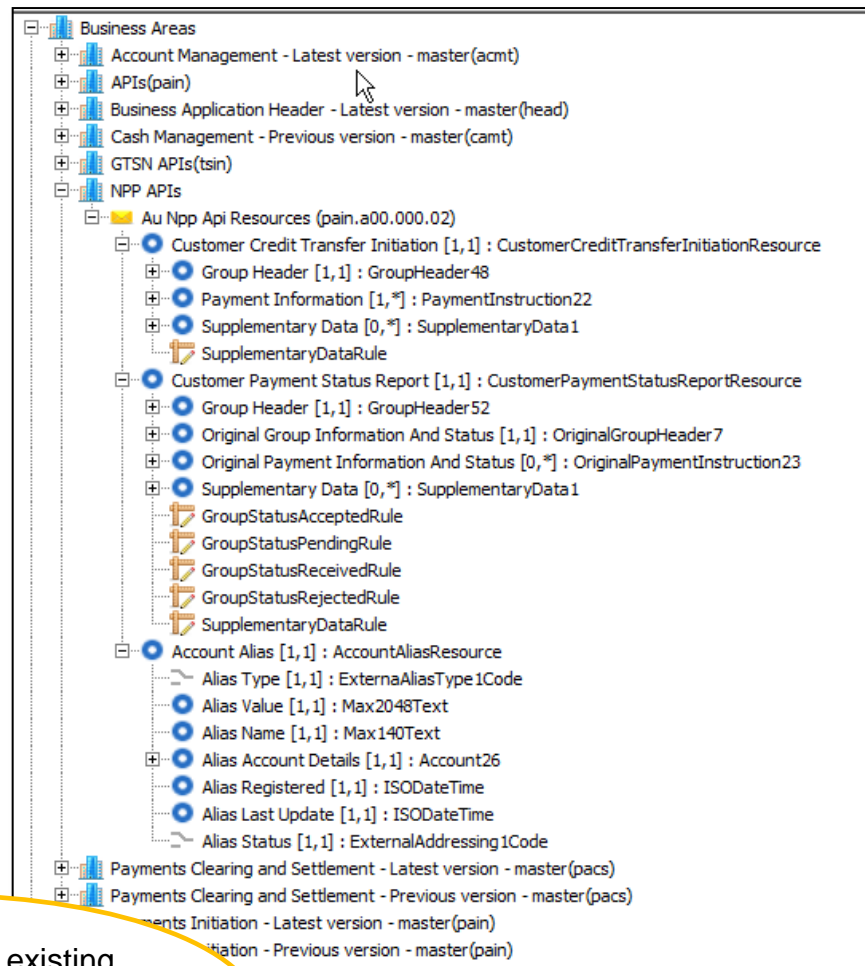
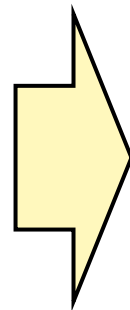
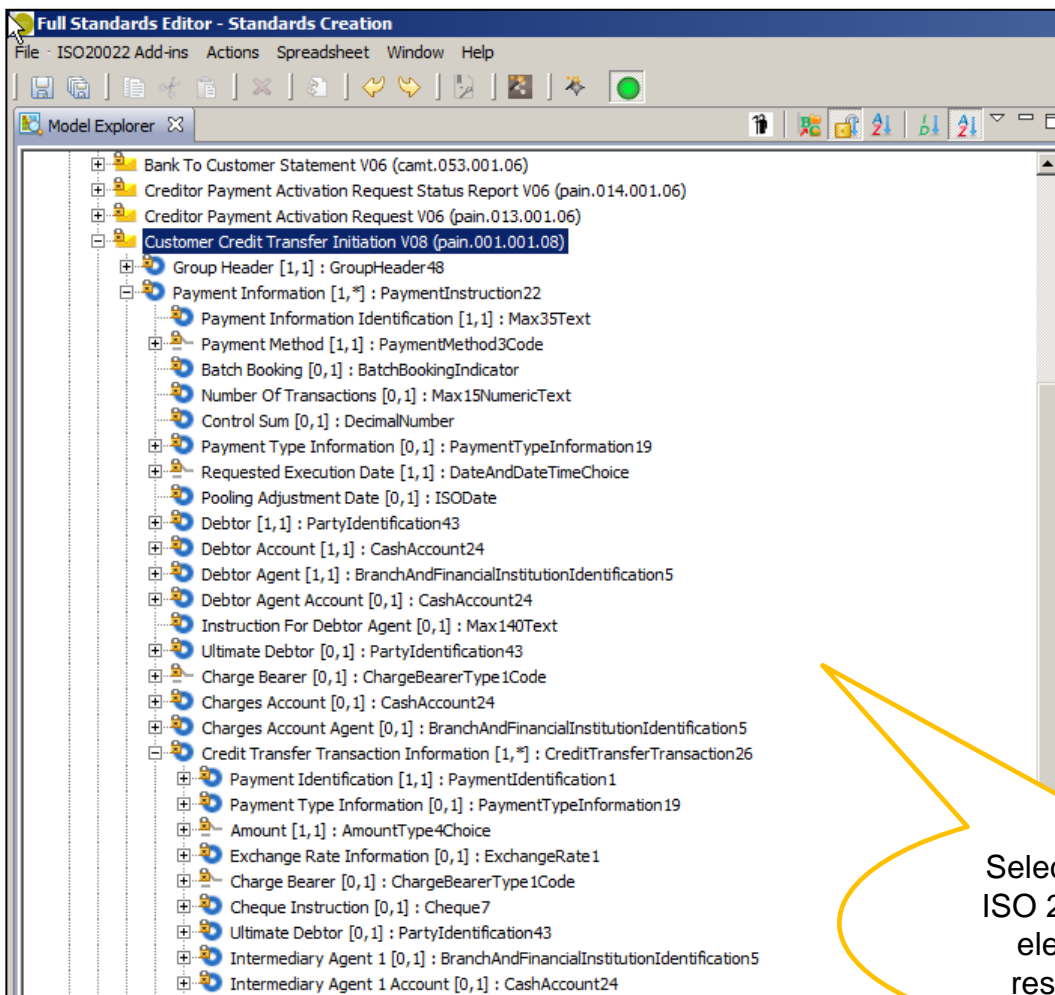


Service specific design

- Security, authentication...
- API **usage guidelines**
- **Testing** (sandbox)
- environment for **review/comments**
- **service** specific APIs
- governance by the **service provider**

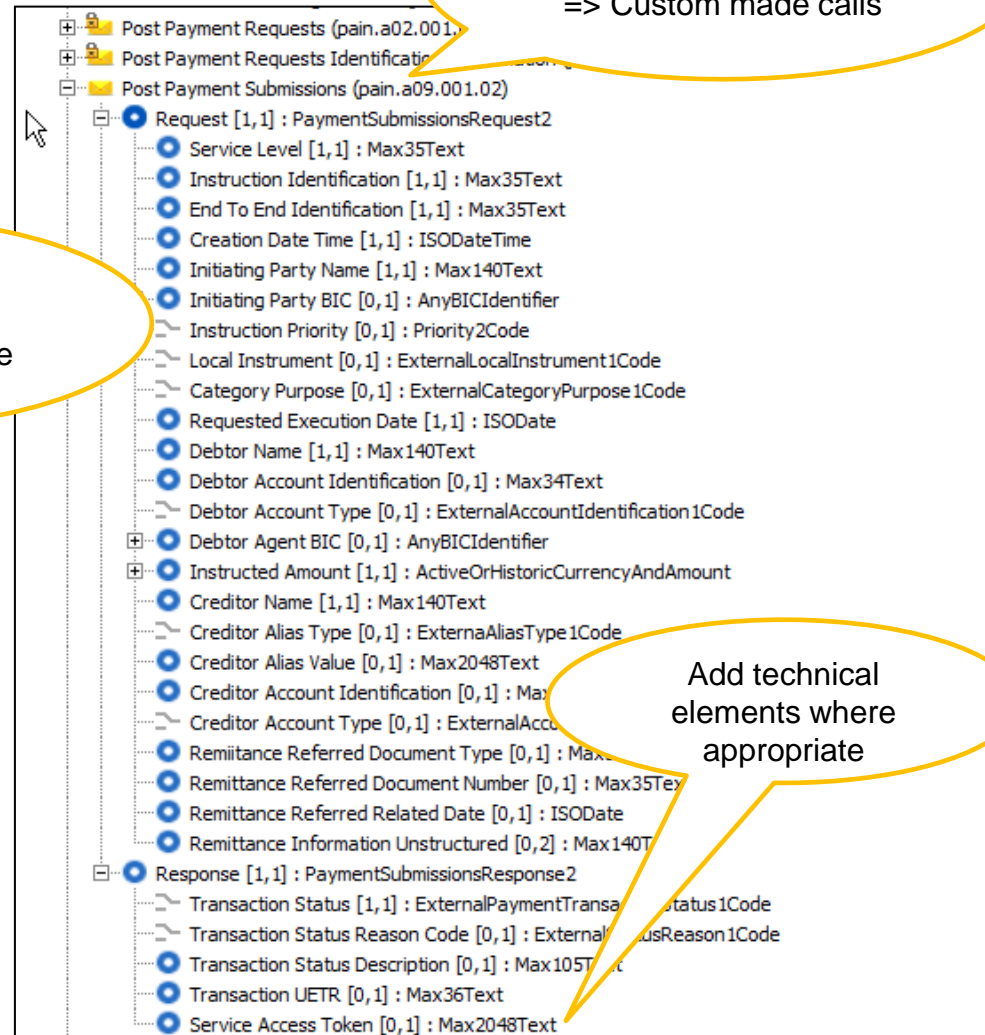
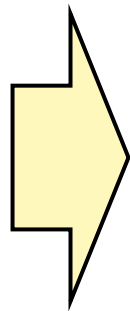
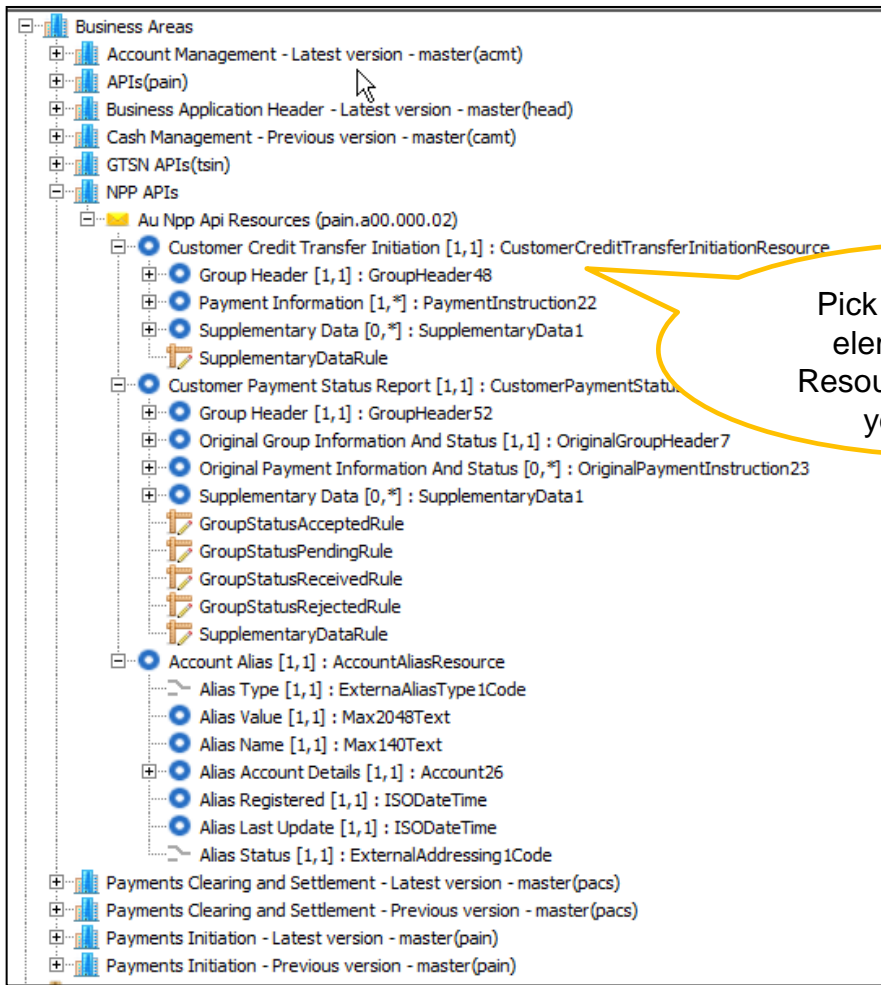


Define the API Resources



Selectively reuse existing
ISO 20022 components &
elements => Custom
resource components

Define the API request/responses based on the Resources



Generate JSON Schemas

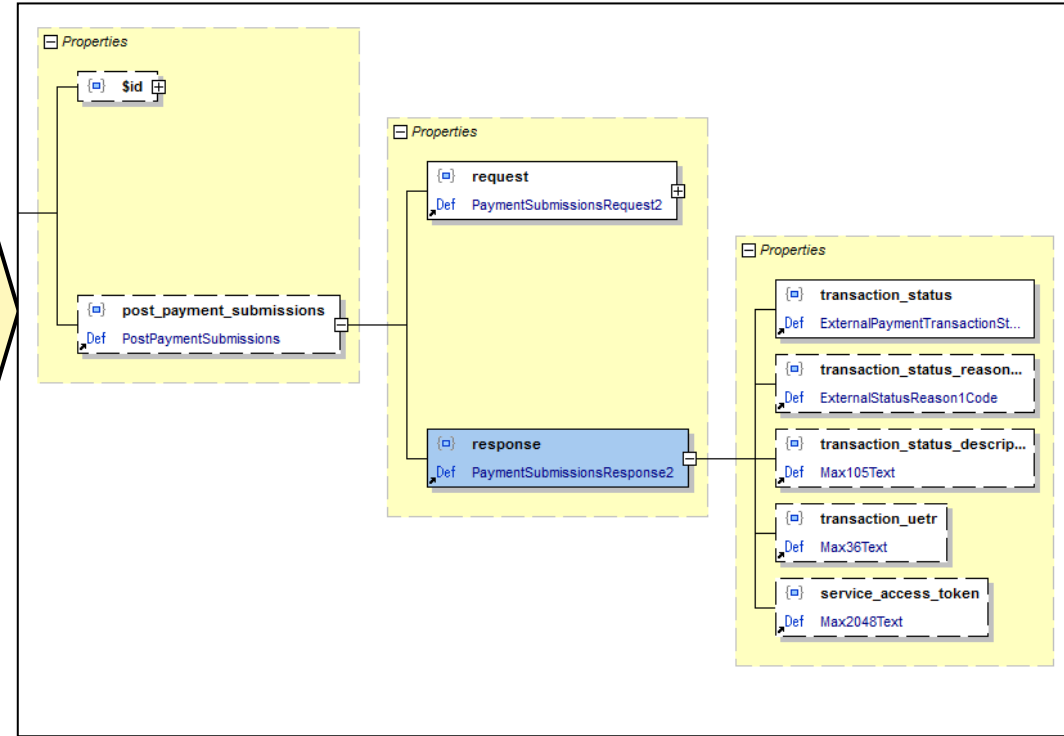


Post Accounts Coverage Control (pain.a06.001.01)
Post Payment Requests (pain.a02.001.01)
Post Payment Requests Identification Confirmation (pain.a08.001.01)
Post Payment Submissions (pain.a09.001.01)

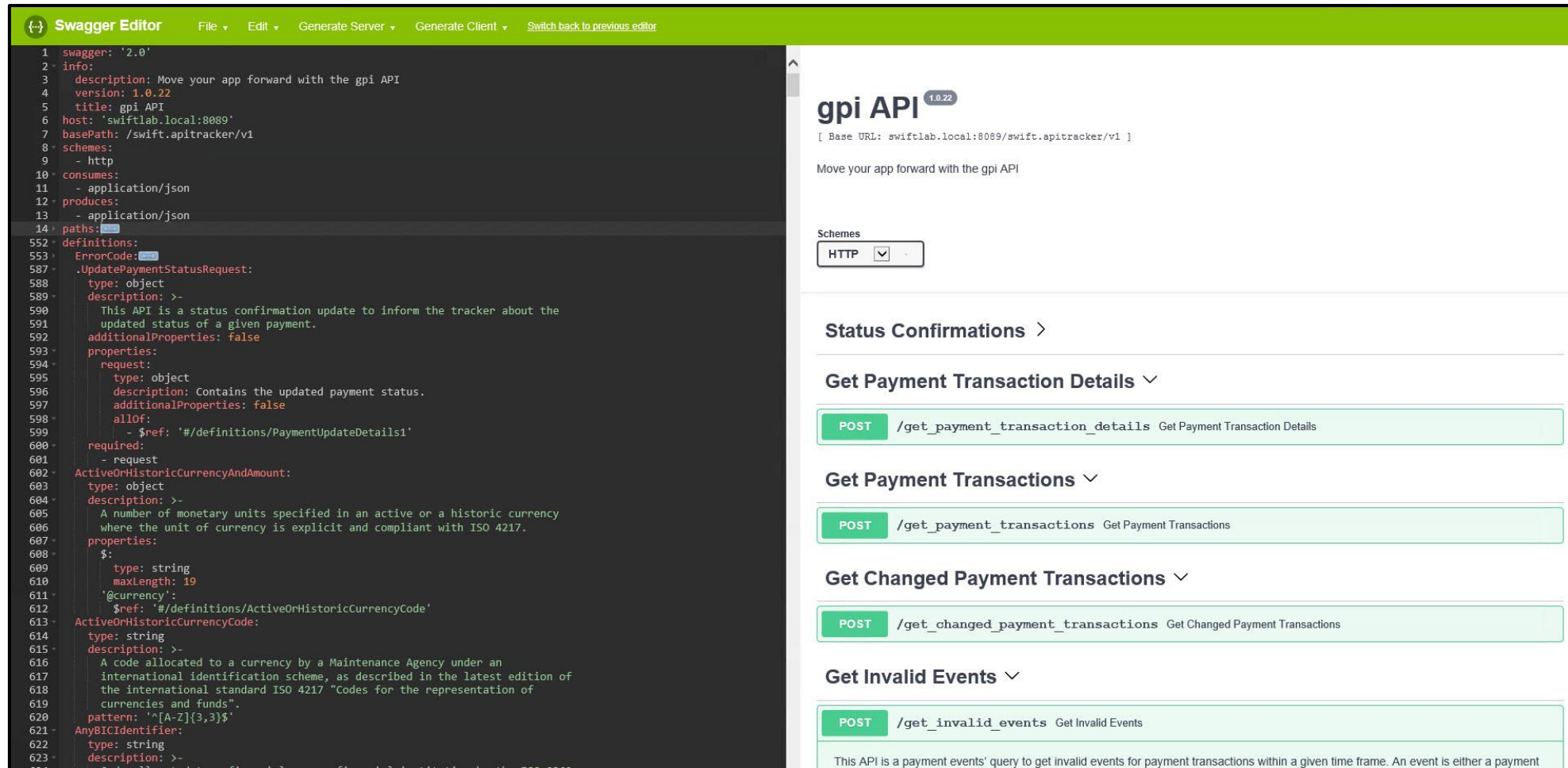
- Request [1,1] : PaymentSubmissionsRequest
- Service Level [1,1] : Max35Text
- Instruction Identification [1,1] : Max35Text
- End To End Identification [1,1] : Max35Text
- Creation Date Time [1,1] : ISODateTime
- Initiating Party Name [1,1] : Max140Text
- Initiating Party BIC [0,1] : AnyBIC
- Instruction Priority [0,1] : Priority2
- Local Instrument [0,1] : ExternalLocalInstrument
- Category Purpose [0,1] : ExternalCategoryPurpose
- Requested Execution Date [1,1] : ISODateTime
- Debtor Name [1,1] : Max140Text
- Debtor Account Identification [0,1] : ExternalAccountIdentification
- Debtor Account Type [0,1] : ExternalAccountType
- Debtor Agent BIC [0,1] : AnyBIC
- Instructed Amount [1,1] : ActiveOrPassiveAmount
- Creditor Name [1,1] : Max140Text
- Creditor Alias Type [0,1] : ExternalCreditorAliasType
- Creditor Alias Value [0,1] : Max204Text
- Creditor Account Identification [0,1] : ExternalAccountIdentification
- Creditor Account Type [0,1] : ExternalAccountType
- Remittance Referred Document Type [0,1] : ExternalRemittanceDocumentType
- Remittance Referred Document Number [0,1] : Max35Text
- Remittance Referred Related Date [0,1] : ISODateTime
- Remittance Information Unstructured [0,1] : Max1024Text
- Response [1,1] : PaymentSubmissionsResponse
- Transaction Status [1,1] : ExternalTransactionStatus
- Transaction Status Reason Code [0,1] : ExternalTransactionStatusReasonCode
- Transaction Status Description [0,1] : Max105Text
- Transaction UETR [0,1] : Max36Text
- Service Access Token [0,1] : Max2048Text

Context Menu:

- New Child
- Undo Remove from MxBusinessArea (Ctrl+Z)
- Redo (Ctrl+Y)
- Cut
- Copy
- Paste
- Delete
- Decouple (Ctrl+Shift+D)
- New Version
- Add To Message Set
- Add To SWIFT Solution
- Move To Business Area
- Generators**
 - Generate SWIFT Documentation
 - Generate ISO Documentation
 - Generate API Documentation
 - Generate XML Tags
 - Generate Excel Sheets
 - Generate Quality Review Excel
 - Generate XML Schemas and esr
 - Generate JSON Schemas**
 - Generate Diff Excel



Import the JSON Schemas into an API modelling tool



The image shows the Swagger Editor interface. On the left, a JSON schema is being edited. The schema defines an API titled 'gpi API' with version '1.0.22'. It includes a description, host, basePath, and consumes/produces media types. The 'paths' section is currently empty. The 'definitions' section contains several objects: 'ErrorCode', 'UpdatePaymentStatusRequest', 'ActiveOrHistoricCurrencyAndAmount', and 'AnyBICIdentifier'. The 'UpdatePaymentStatusRequest' object has a description, properties, and a required field 'request'. The 'ActiveOrHistoricCurrencyAndAmount' object has a description, properties, and a required field 'ActiveOrHistoricCurrencyCode'. The 'AnyBICIdentifier' object has a description and a pattern. On the right, the rendered API documentation is displayed. It shows the title 'gpi API' with version '1.0.22' and a base URL. Below the title, there is a section for 'Status Confirmations' and a list of endpoints: 'Get Payment Transaction Details', 'Get Payment Transactions', 'Get Changed Payment Transactions', and 'Get Invalid Events'. Each endpoint is listed with its HTTP method (POST) and a brief description.

```
1 swagger: '2.0'
2 info:
3   description: Move your app forward with the gpi API
4   version: 1.0.22
5   title: gpi API
6   host: 'swiftlab.local:8089'
7   basePath: /swift.apitracker/v1
8   schemes:
9     - http
10 consumes:
11   - application/json
12 produces:
13   - application/json
14 paths:
15 definitions:
16   ErrorCode:
17     type: object
18     description: >-
19       This API is a status confirmation update to inform the tracker about the
20       updated status of a given payment.
21     additionalProperties: false
22     properties:
23       request:
24         type: object
25         description: Contains the updated payment status.
26         additionalProperties: false
27         allOf:
28           - $ref: '#/definitions/PaymentUpdateDetails1'
29     required:
30       - request
31   ActiveOrHistoricCurrencyAndAmount:
32     type: object
33     description: >-
34       A number of monetary units specified in an active or a historic currency
35       where the unit of currency is explicit and compliant with ISO 4217.
36     properties:
37       $:
38         type: string
39         maxLength: 19
40         '@currency':
41           $ref: '#/definitions/ActiveOrHistoricCurrencyCode'
42   ActiveOrHistoricCurrencyCode:
43     type: string
44     description: >-
45       A code allocated to a currency by a Maintenance Agency under an
46       international identification scheme, as described in the latest edition of
47       the international standard ISO 4217 "Codes for the representation of
48       currencies and funds".
49     pattern: '^?[A-Z]{3,3}$'
50   AnyBICIdentifier:
51     type: string
52     description: >-
```

gpi API ^{1.0.22}
[Base URL: swiftlab.local:8089/swift.apitracker/v1]

Move your app forward with the gpi API

Schemes
HTTP

Status Confirmations >

Get Payment Transaction Details ▾

POST /get_payment_transaction_details Get Payment Transaction Details

Get Payment Transactions ▾

POST /get_payment_transactions Get Payment Transactions

Get Changed Payment Transactions ▾

POST /get_changed_payment_transactions Get Changed Payment Transactions

Get Invalid Events ▾

POST /get_invalid_events Get Invalid Events

This API is a payment events' query to get invalid events for payment transactions within a given time frame. An event is either a payment



and complement with implementation specific information

Bringing the ISO 20022 repository to the community

Response Content Type
application/json ▼

Parameters

Parameter	Value	Description	Parameter Type	Data Type
Pain001	<pre>{ "instruction_identification": "A0455", "end_to_end_identification": "A0566", "initiating_party_name": "Myself", "creation_date_time": "2018-03-27T00:00:01Z", "service_level": "NORM", "local_instrument": "INST", "category_purpose": "SUPP", "requested_execution_date": "2018-03-31", "debtor_name": "Bob Masina", "debtor_alias_type": "EMAL", "debtor_alias_value": "bob.masina@nppa.com.au", "debtor_account_identification": "855833382299", "debtor_account_type": "CURR", "debtor_agent_bic": "ANZBAU3M", "instructed_amount": { "currency": "AUD", "\$": "500" }, "creditor_name": "Qantas", "creditor_account_identification": "38484890", "creditor_account_type": "Current", "remittance_referred_document_type": "LIST", "remittance_referred_document_number": "12345", "remittance_referred_related_date": "2018-03-31", "remittance_information_unstructured": ["smiley here"] }</pre>	Resource to instruct movement of funds from the debtor account to creditor.	body	Model Example Value NPPPostCustomerCreditTransferInitiation { instruction_identification (string): Unique identification as assigned by an instructing party for an instructed party to unambiguously identify the instruction., end_to_end_identification (string): Unique identification assigned by the initiating party to unambiguously identify the transaction. This identification is passed on, unchanged, throughout the entire end-to-end chain., creation_date_time (string): Date and time at which the request was created., initiating_party_name (string): Name by which the Initiating Party is known and which is usually used to identify that party., initiating_party_bic (string, optional): Code allocated to a financial or non-financial institution by the ISO 9362 Registration Authority, as described in ISO 9362 "Banking - Banking telecommunication messages - Business identifier code (BIC)", instruction_priority (string, optional): Indicator of the urgency or order of importance that the instructing party would like the instructed party to apply to the processing of the instruction. = ['HIGH', 'NORM']

Parameter content type: application/json ▼



Documentation Approach

WG Circulated Draft in Jun 2018

Technical Reference

- ❑ Overview of Web Application Interface
- ❑ WAPI Basic Terminology
 - Terms and Definition
 - Communication Protocol
 - HTTP/ HTTPS
 - Web Socket
 - Web Hook
 - JSON
 - Data Structure
 - JSON
 - XML
 - HTML
 - Security Authentication
 - Infrastructure Layer
 - Transportation
 - Application Layer
 - OAuth 2.0
 - Server
 - Public Client
 - Confidential Client

WG Consolidating Feedback

Implementation Guide

- ❑ Overview of Web Application Integration
- ❑ Design Principles
 - Communication (tabulate pros and cons of each approach. i.e. HTTP, Webhook, JSON)
- ❑ WAPI Rule
 - Name Convention
 - Data Payload
- ❑ Security and Authentication
 - Authorization Profile
 - Data Access
 - Threat Model & Security Considerations
 - Data Integrity, Authentication & Non-repudiation
- ❑ Version Control

ISO 2022 TSG Published Mapping Rule

Use Case – Example ISO 2022

- ❑ Metamodel
- ❑ Business Transaction
 - Actors
 - Transaction Flow
- ❑ API Call Design
 - Recommendation for each Communication Protocol
- ❑ Modelling Guideline
 - Mapping Rule



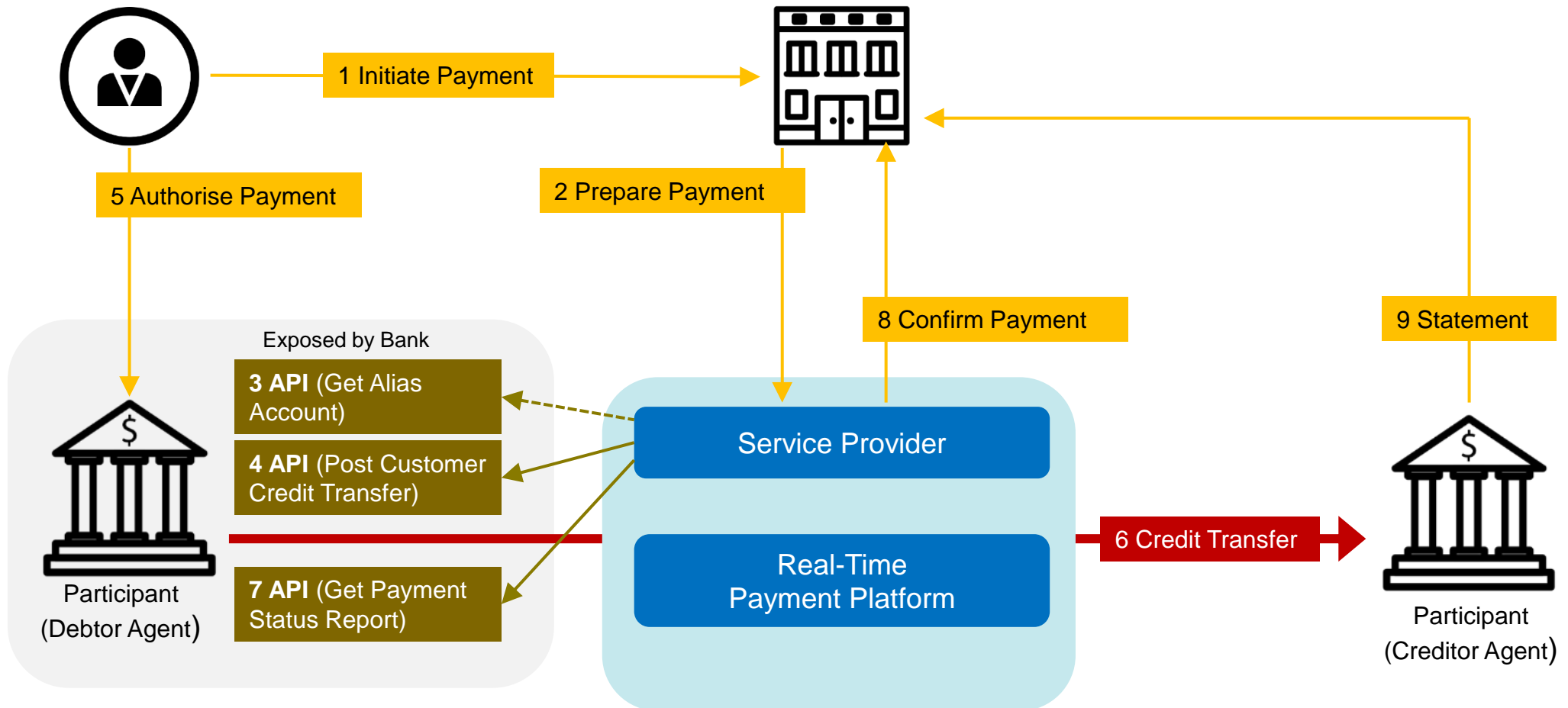
ISO 20022 for APIs Use Cases



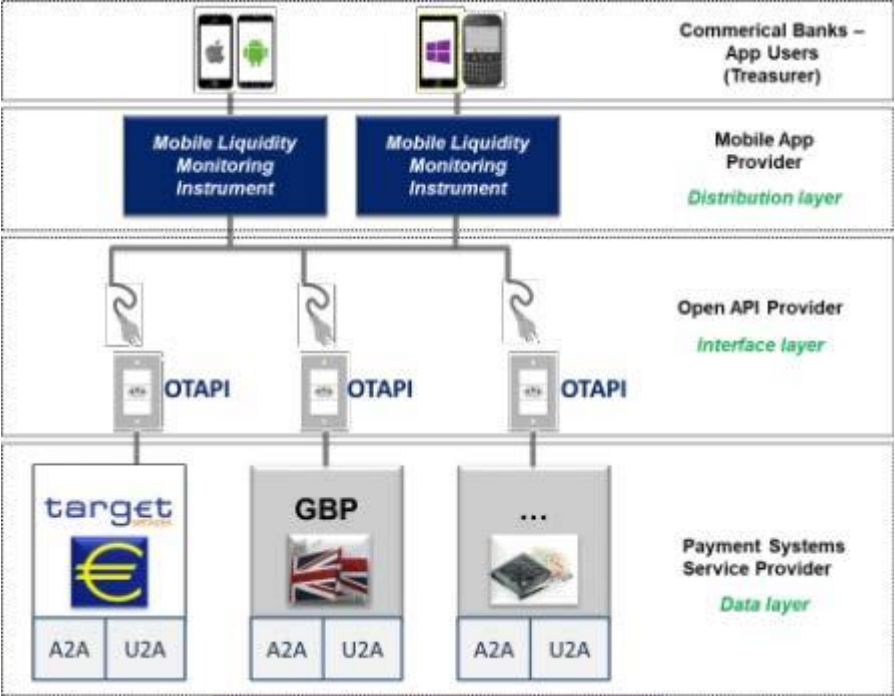
An API Use Case

Buyer (Debtor - Payer)

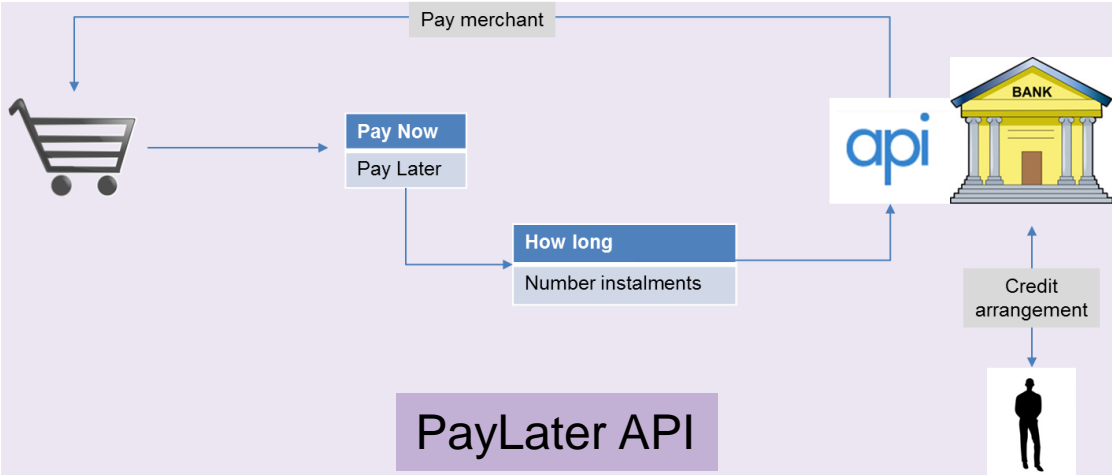
Merchant (Creditor - Biller)



SWIFT third party API Proof Of Concepts



Open Treasury API



Why is DLT a valid solution to support the voting process?

By leveraging blockchain, we should be able to interface all the stakeholders of the General Meeting: Issuer, Issuer Agent, Intermediary and Shareholders.

The solution and the technology will allow to reduce friction and manual intervention in the voting process and also ensure that all information is transparent to stakeholders when required and with the proper security, governance and risk procedures in place

Benefits to Shareholders, Issuers and Issuer Agents

- Secure identification of shareholders and their representatives
- Information about meetings, agenda items and management proposals is easily reachable
- Full track record of casted votes
- Online/real-time access to voting results
- Data is saved in an immutable ledger
- Transparency and auditability of voting results for companies (listed and unlisted)
- Compliance with SRD2 provisions

Benefits to Financial Market Infrastructures & regulators

- Reduced complexity through end-to-end general meeting administration in a single solution – from notification and material distribution to voting management and reporting at the general meeting
- Enhanced insight through real-time data access to securities holders, holdings and votes and ability to view full voting history on a single account
- Safety, security and speed benefits of leveraging blockchain technology with a fully auditable view of actions recorded in the immutable ledger








Scope of the Proof-of-Concept




Key functional highlights

- Meeting notices provided through the ledger or via ISO 20022 messages
- Upload of the voting rights on the ledger by the Issuer / Issuer Agent
- Retail shareholders' votes thru the ledger
- Issuers and/or Issuer Agent have access to all the votes via the ledger
- Results of the general meeting are made available through the ledger or via ISO 20022 messages

In scope

-  Creation of a general meeting on the Blockchain
-  Shareholder, Issuer (and/or "Centralisateurs", Administrator
-  Manual Upload of shareholder reference data with voting rights
-  Pre Voting Phase
-  Management of confidentiality

Out of scope

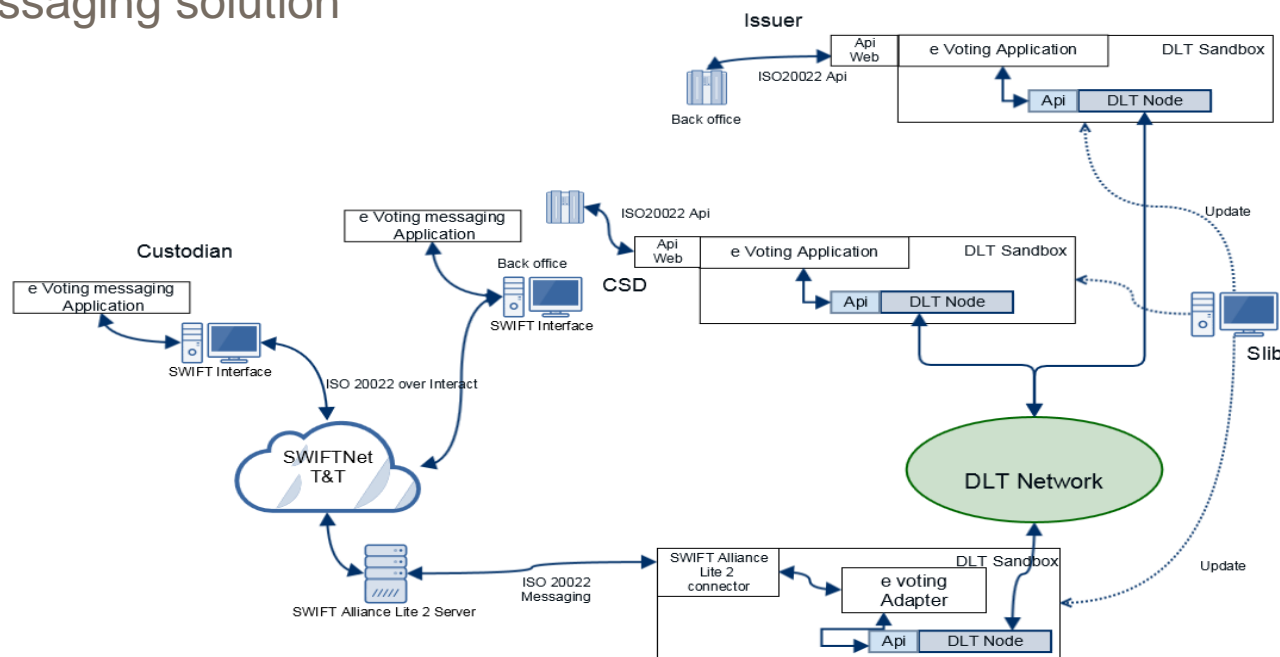
-  Custodian / Account keeper
-  Update & calculation of voting rights
-  Registration & Vote during session



How to access the e-Voting DLT based solution?

The system will support two communication channels:

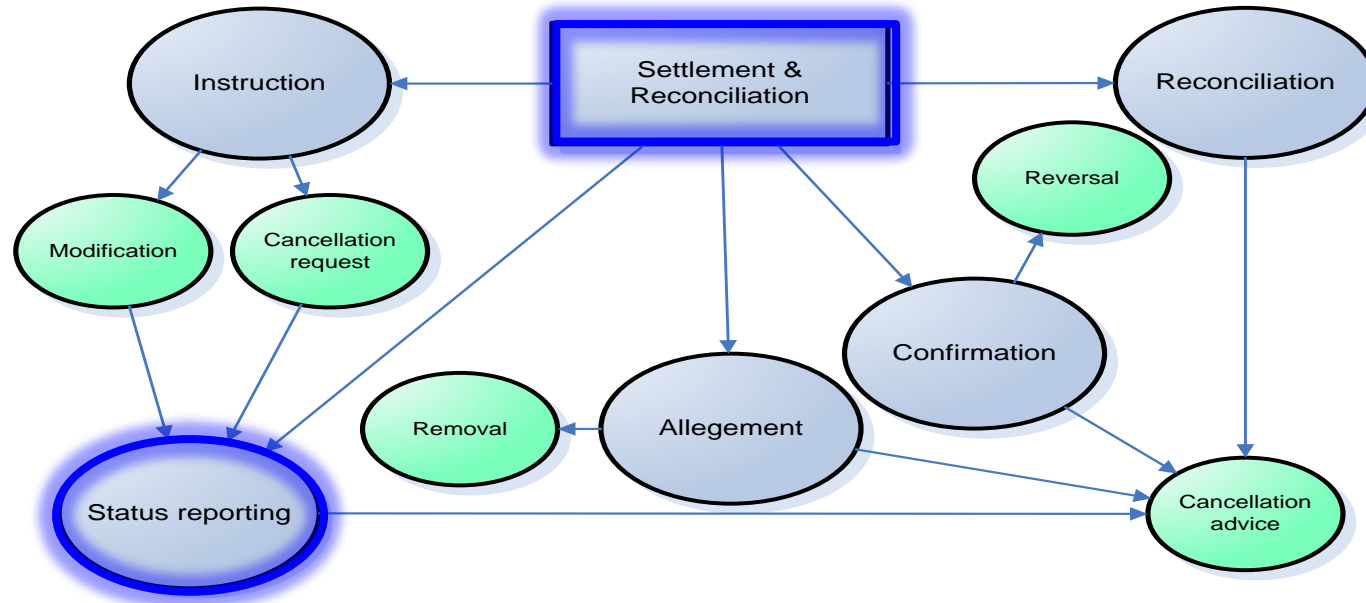
1. **DLT channel:** the system will be implemented on Hyperledger Fabric. Each participant will have a dedicated DLT Node running on a SWIFT sandbox
2. **Messaging channel:** the system will in parallel with the DLT based solution offer a ISO 20022 messaging solution



Business Scope & Flows (1)

ISO 20022 Business Process & Roles:

Settlement And Reconciliation → Status Reporting

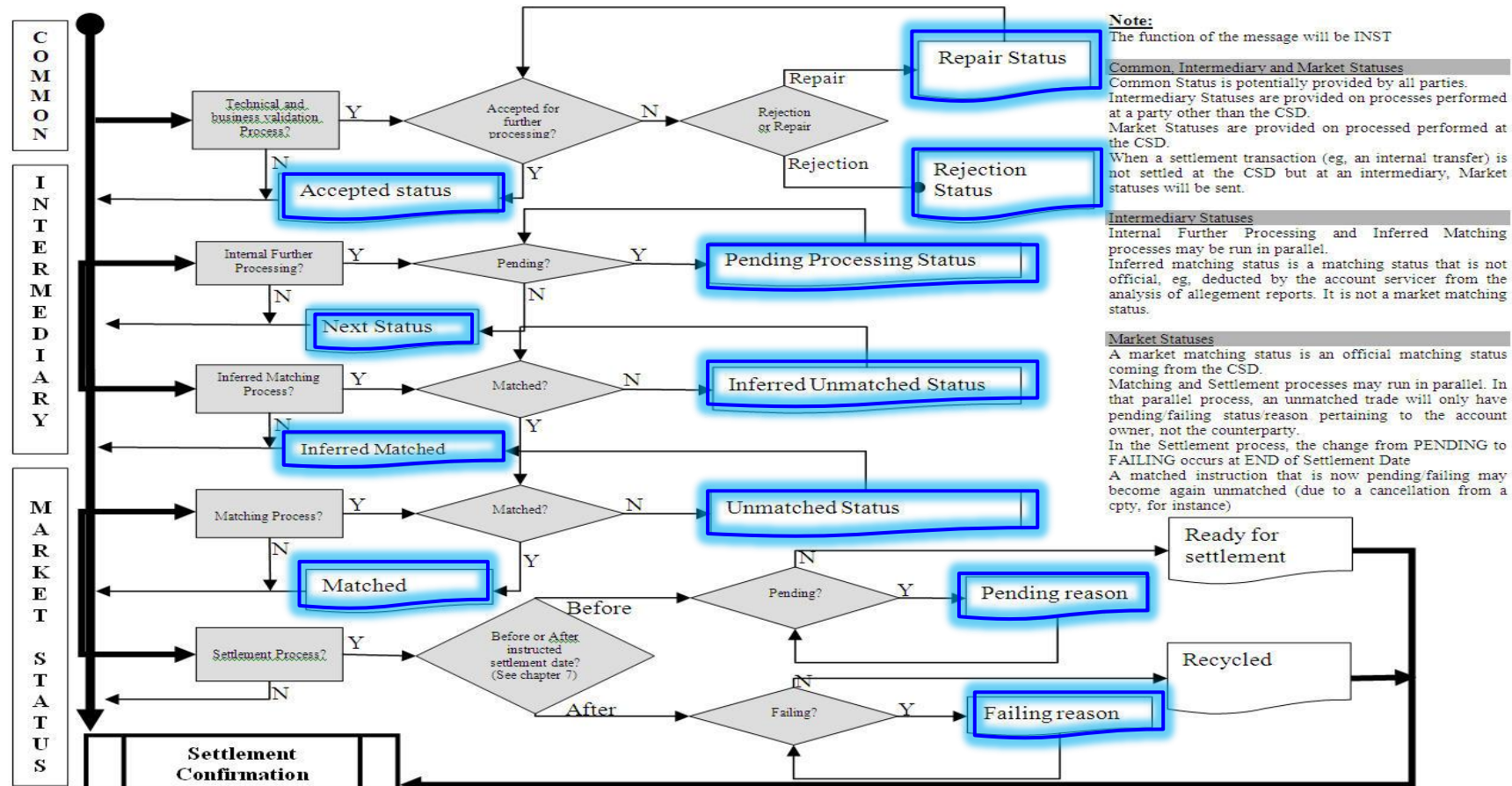


Status Reporting Definition: The process of providing or relaying the status of a transaction during the various phases of its lifecycle.

Roles: Executing / servicing Party

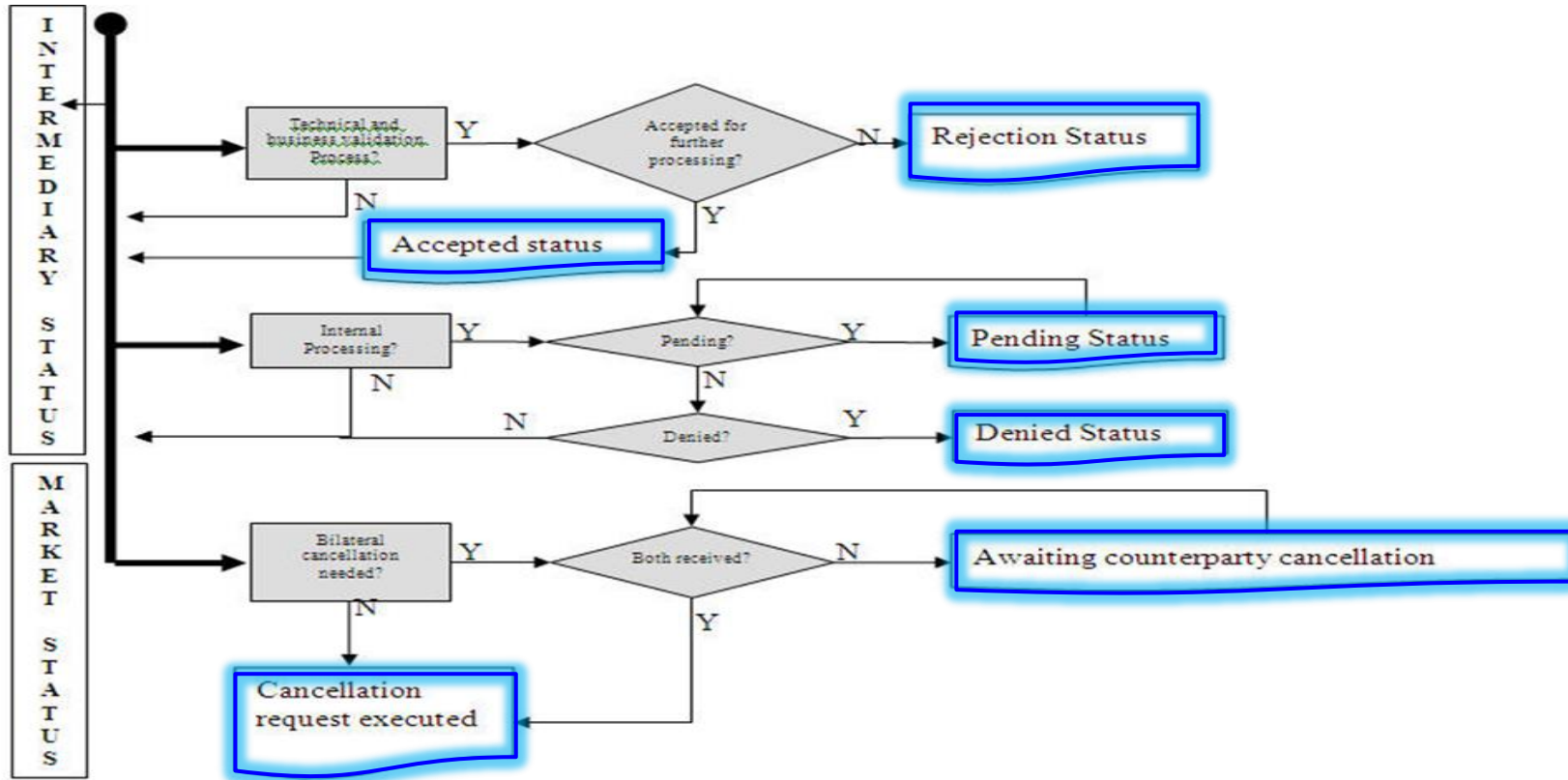
Business Scope & Flows (2)

Instruction Status Advice - ISO 20022 Business Activities:



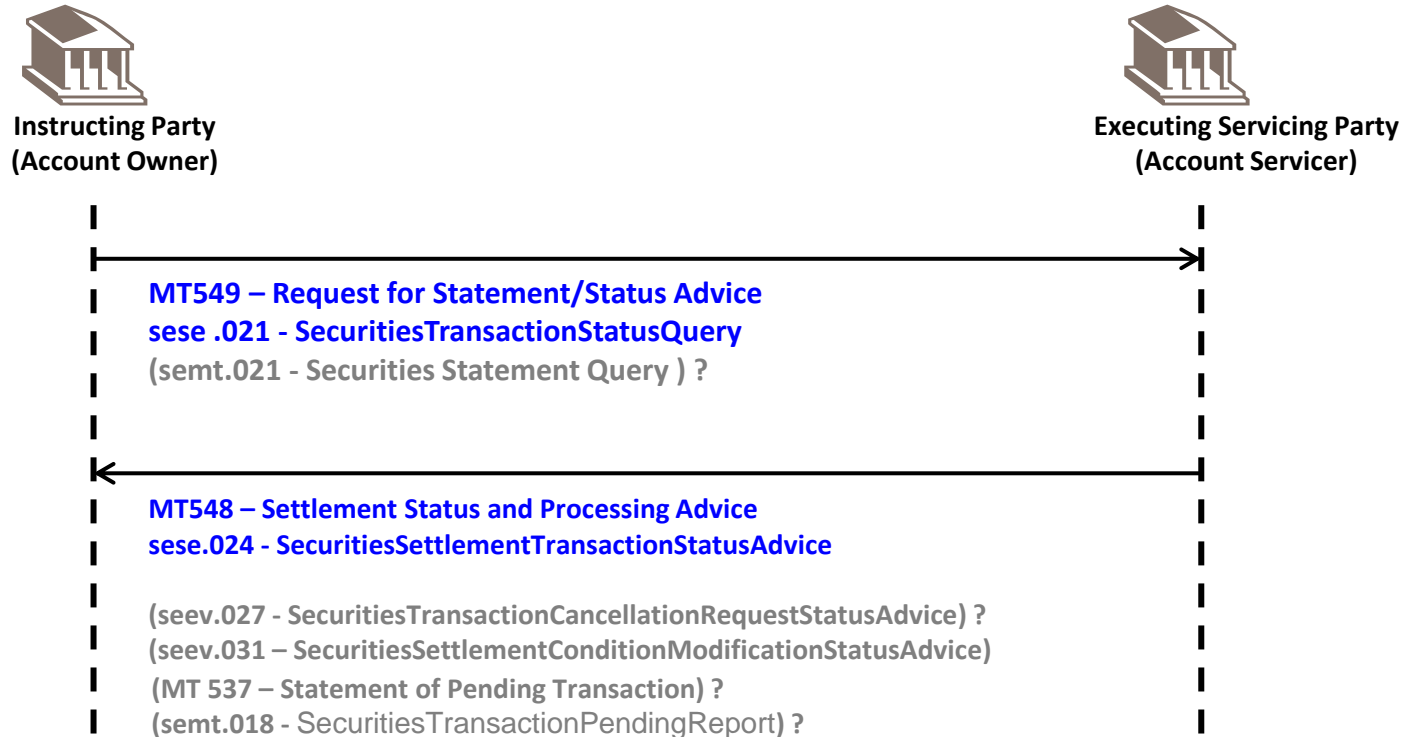
Business Scope & Flows (2)

Cancellation Request Status Advice - ISO 20022 Business Activities:



Business Scope & Flows (3)

Status Reporting – Message Flows Covered (replaced by APIs):



Business Scope & Flows (4)

Coverage of Settlement and Processing Statuses

Instruction Status Types to cover:	Statuses & Reason codes to cover for:
Instruction Processing Status / IPRC	Rejected, Acknowledged / Accepted, Pending Processing, CancellationRequest, PendingCancellation, Cancelled
Matching Status / MTCH	Matched, Unmatched
Settlement Status / SETT	Pending Settlement, Failing Settlement
Cancellation Processing Status / CPRC	Cancellation Completed, Pending Cancellation, Denied, Rejected
Processing Change Command Status / TPRC	Denied, Modification Completed, Modification Pending, Rejected



Proposed Query

Search Parameters	Date Type	Type of Returned Results
<u>Get Securities Settlement Transaction Status (1)</u>		1 single Securities Settlement Transaction
AccountOwnerTransactionIdentification [1..1]	Max35Text	
<u>Get Securities Settlement Transaction Status (2)</u>		One or more Securities Settlement Transaction depending upon the combination of values provided parameters
Status [0..1]	Code	
Reason [0..1]	Code	
SafekeepingAccount [0..1]	Max35Text	
FinancialInstrumentIdentification [0..1]	ISINOct2015Identifier	
SecuritiesMovementType [0..1]	Code	
Payment [0..1]	Code	
SecuritiesTransactionType [0..1]	Code	
SettlementQuantity [0..1]	DecimalNumber	
FromSettlementDate [0..1]	ISODateTime	
ToSettlementDate [0..1]	ISODateTime	





SWIFT Standards explores how to standardise APIs for the financial community.



Evaluate re-use of
ISO20022
methodology and
repository for API
definition



Provide tools for the
creation of APIs
'design time'



Unparalleled
expertise on
ISO20022 for your
own API design



Provide a self-service
portal with interactive
documentation and
sandboxing for your
developer community



www.swift.com