

**For: Bank of Ireland**

**Document Version: 0.4**

**AISP**

**Solution Blueprint**

Contents

[1 Document Control 3](#_Toc486195213)

[1.1 Change Record 3](#_Toc486195214)

[2 Functional Overview 4](#_Toc486195215)

[2.1 Functional Block Diagram 4](#_Toc486195216)

[3 References 5](#_Toc486195217)

[4 List of PSD2 Platform APIs 6](#_Toc486195218)

[5 SCA and Consent 8](#_Toc486195219)

[6 Data Filtering 9](#_Toc486195220)

[7 Account Information API 10](#_Toc486195221)

[7.1 Overview 10](#_Toc486195222)

[7.2 Sequence / Flow Diagram 10](#_Toc486195223)

[7.3 Integration Points 10](#_Toc486195224)

[7.4 Configuration Elements 10](#_Toc486195225)

[7.5 Logging & Monitoring Requirements 10](#_Toc486195226)

[7.6 Security and Connectivity 10](#_Toc486195227)

[8 Single Account Balance API 12](#_Toc486195228)

[8.1 Overview 12](#_Toc486195229)

[8.2 Sequence / Integration Diagram 12](#_Toc486195230)

[8.3 Integration Points 12](#_Toc486195231)

[8.4 Configuration Elements 12](#_Toc486195232)

[8.5 Logging & Monitoring Requirements 12](#_Toc486195233)

[8.6 Security and Connectivity 12](#_Toc486195234)

[9 Account Transactions API 14](#_Toc486195235)

[9.1 Overview 14](#_Toc486195236)

[9.2 Sequence / Integration Diagram 14](#_Toc486195237)

[9.3 Integration Points 14](#_Toc486195238)

[9.4 Configuration Elements 14](#_Toc486195239)

[9.5 Logging & Monitoring Requirements 14](#_Toc486195240)

[9.6 Security and Connectivity 15](#_Toc486195241)

[10 Logging and Monitoring Architecture 16](#_Toc486195242)

# Document Control

|  |  |
| --- | --- |
| **Title** | AISP Solution Blueprint |
| **Original Author(s)** | Ramandeep Singh - Capgemini |

## Change Record

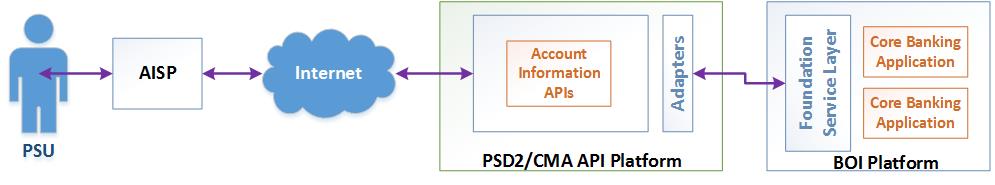
|  |  |  |  |
| --- | --- | --- | --- |
| **Version No.** | **Date** | **Author(s)** | **Revision Notes** |
| 0.1 | 17-May-17 | Ramandeep Singh - Capgemini | Initial Version |
| 0.2 | 05-Jun-17 | Sandeep Sharma - Capgemini | Added API details |
| 0.3 | 10-Jun-17 | Ramandeep Singh – Capgemini | Updated as per BOI comments |
| 0.4 | 24-Jun-17 | Ramandeep Singh – Capgemini | Updated to include updates as per latest FS XSD and CMA 1.0 RC 3 |

# Functional Overview

The AISP module will provide the required functionality and APIs for the Account Information Service Providers to access the required account information as described in PSD2 and CMA directives.

An account information service is an online service to provide consolidated information on one or more payment accounts held by a PSU with another ASPSP or multiple PSPs. An AISP acts as an aggregator of data relating to a Payment Service User’s accounts

## Functional Block Diagram



* **PSU**: *Payment Service User* is the customer who would be utilizing account aggregation functionality of AISP to access account information from the Bank (ASPSP).
* **AISP**: *Account Information Service Provider* will invoke the appropriate account information APIs passing the required Access Token and other required parameters to retrieve PSU’s account information from the Bank.
* **PSD2/CMA API Platform**: PSD2/CMA API Platform provides the required APIs to the external parties to access account details and payment services. This API platform will interface with Bank’s core banking system through Foundation Services Layer to access required account information and invoke payment services.
* **Foundation Service Layer**: *Foundation Service Layer* provides access to core banking systems through REST web services and XML based input/output.

# References

|  |  |
| --- | --- |
| **Document** | **Version** |
| CMA - Account and Transaction API Specification | 1.0 RC3 |
| AISP\_FS\_XSDs\_Dtls.xlsx | 17-Jun-2017 |
|  |  |
|  |  |

# List of PSD2 Platform APIs

List of APIs exposed to external and internal parties

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| **SNO** | **Category** | **API Name** | **Consumer** | **HTTP Method** | **Endpoint URL Pattern** | **I/O Message Format** |
| 1 | AISP Consent | Create Account Info Request | AISP | POST | /account-requests | Included in the SCA Consent Blueprint |
| 2 | AISP Consent | Get Account Info Request | AISP | GET | /account-requests/{AccountRequestId} | Included in the SCA Consent Blueprint |
| 3 | AISP Consent | Delete Account Info Request | AISP | DELETE | /account-requests/{AccountRequestId} | Included in the SCA Consent Blueprint |
| 4 | AISP Account Information | Get Account Info | AISP | GET | /accounts/{AccountId} | **ICD AISP-Account-Information v0.3.docx** |
| 5 | AISP Account Information | Get Account Balances | AISP | GET | /accounts/{AccountId}/balances | **ICD AISP-Single Account-Balance v0.3.docx** |
| 6 | AISP Account Information | Get Account Transactions | AISP | GET | /accounts/{AccountId}/transactions | **ICD AISP-Single Account-Transaction v0.3.docx** |
| 7 | AISP Account Additional Information | Get Account Beneficiaries | AISP | GET | /accounts/{AccountId}/beneficiaries | Would be included in the CMA API specific Blueprint |
| 8 | AISP Account Additional Information | Get Account Direct-Debits | AISP | GET | /accounts/{AccountId}/direct-debits | Would be included in the CMA API specific Blueprint |
| 9 | AISP Account Additional Information | Get Account Product | AISP | GET | /accounts/{AccountId}/product | Would be included in the CMA API specific Blueprint |
| 10 | AISP Account Additional Information | Get Account Standing-Orders | AISP | GET | /accounts/{AccountId}/standing-orders | Would be included in the CMA API specific Blueprint |

# SCA and Consent

SCA and Consent related details are included in the SCA and Consent Blueprint.

Visual design for SCA and Consent screen is maintained separately.

# Data Filtering

Data filtering is managed in the API layer by using the configuration parameters.

Below is an example of how filtering properties are provided

app:

name: AccountBalance

filter:

reteriveAccountBalance:

ReadAccounts: accounts[accountId,currency,nickname]

ReadAccountsSensitive: accounts[account{schemeName,identification,name,secondaryIdentification},servicer{schemeName,identification}]

The filtering works based on the positive attribute specification. For an API, the attributes which should be included for a permission set are included. Only the specified attributes would be included in the result, any other data elements would be filtered out.

Above configurable logic works for field based filtering. For complex filtering of one type of transactions, either debit or credit, API would include the logic in the code to perform the filtering.

# Account Information API

## Overview

For Account Information API that is invoked by a TPP (AISP) to retrieve account Information of a customer for a single account identifier

## Sequence / Flow Diagram

Please refer to the following files for the flow details:

Capgemini API Platform - AISP APIs - Account Information v01.pdf

## Integration Points

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **SNO** | **System/Application** | **Integration Point** | **Purpose** | **ICD Document** |
| 1 | Core Banking System | Foundation Services | To retrieve account balance data | ICD AISP-Account-Information v0.2.docx |
| 2 | Logging System | Splunk | Payload logging | NA |
| 3 | Database | MongoDB | Consent permissions reference data | NA |

## Configuration Elements

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **SNO** | **Configuration Category** | **Configuration Element** | **Purpose** | **Default Value** | **BOI Value** |
| 1 | Throttling | Mule API Gateway Configuration | Throttle the request per TPP | 1000 / min |  |
| 2 | Security Policy | OAuth Mule Security Policy | Token based authorization | Enabled | Enabled |

## Logging & Monitoring Requirements

|  |  |
| --- | --- |
| Sno | Description |
| 1 | Payload logging – Implemented using Splunk |
| 2 | Audit log of API access – Implemented using Splunk |
| 5 | JVM Monitoring – Implemented using AppDynamics |
| 6 | API Analytics – Implemented using Splunk custom dashboards |

## Security and Connectivity

|  |  |
| --- | --- |
| Security | |
| 1 | MTLS |
| 2 | Token based authentication using OAuth |
| 3 | Permissions defined by CMA draft 0.1 Specifications, please refer to ICD for the details |
| Connectivity | |
| Dev & System Test | Mock data |
| SIT | VPN to BOI Network |
| UAT, Pre-Prod and Prod | AWS Direct Connect |

## Data Masking

* No data masking rules are applicable

# Single Account Balance API

## Overview

Single Account Balance API that is invoked by a TPP (AISP) to retrieve account balance of single customer account identifier

## Sequence / Integration Diagram

Please refer to the following files for the flow details:

Capgemini API Platform - AISP APIs - Balance v01.pdf

## Integration Points

Describe at a high level what all integration points required and what information they provide.

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **SNO** | **System/Application** | **Integration Point** | **Purpose** | **ICD Document** |
| 1 | Core Banking System | Foundation Services | To retrieve account balance data | ICD AISP-Single Account-Balance v0.2.docx |
| 2 | Logging System | Splunk | Payload logging | NA |
| 3 | Database | MongoDB | Consent permissions reference data | NA |
|  |  |  |  |  |
|  |  |  |  |  |

## Configuration Elements

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **SNO** | **Configuration Category** | **Configuration Element** | **Purpose** | **Default Value** | **BOI Value** |
| 1 | Throttling | Mule API Gateway Configuration | Throttle the request per TPP | 1000 / min |  |
| 2 | Security Policy | OAuth Mule Security Policy | Token based authorization | Enabled | Enabled |

## Logging & Monitoring Requirements

|  |  |
| --- | --- |
| Sno | Description |
| 1 | Payload logging – Implemented using Splunk |
| 2 | Audit log of API access – Implemented using Splunk |
| 5 | JVM Monitoring – Implemented using AppDynamics |
| 6 | API Analytics – Implemented using Splunk custom dashboards |

## Security and Connectivity

|  |  |
| --- | --- |
| Security | |
| 1 | MTLS |
| 2 | Token based authentication using OAuth |
| 3 | Permissions defined by CMA draft 0.1 Specifications, please refer to ICD for the details |
| Connectivity | |
| Dev & System Test | Mock data |
| SIT | VPN to BOI Network |
| UAT, Pre-Prod and Prod | AWS Direct Connect |

## Data Masking

* No data masking rules are applicable

# Account Transactions API

## Overview

Account Transactions API that is invoked by a TPP (AISP) to retrieve account transactions of a single customer account identifier based on the consent authorization given to TPP (AISP)

## Sequence / Integration Diagram

Please refer to the following files for the flow details:

Capgemini API Platform - AISP APIs - Transactions v02.pdf

## Integration Points

Describe at a high level what all integration points required and what information they provide.

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **SNO** | **System/Application** | **Integration Point** | **Purpose** | **ICD Document** |
| 1 | Core Banking System | Foundation Services | To retrieve account transaction data | ICD AISP-Single Account-Transaction v0.2.docx |
| 2 | Logging System | Splunk | Payload logging | NA |
| 3 | Database | MongoDB | Consent permissions reference data | NA |
|  |  |  |  |  |
|  |  |  |  |  |

## Configuration Elements

Describe specific configuration items and their configuration values decided

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **SNO** | **Configuration Category** | **Configuration Element** | **Purpose** | **Default Value** | **BOI Value** |
| 1 | Throttling | Mule API Gateway Configuration | Throttle the request per TPP | 1000 / min | ? |
| 2 | Security Policy | OAuth Mule Security Policy | Token based authorization | Enabled | Enabled |

## Logging & Monitoring Requirements

|  |  |
| --- | --- |
| Sno | Description |
| 1 | Payload logging – Implemented using Splunk |
| 2 | Audit log of API access – Implemented using Splunk |
| 5 | JVM Monitoring – Implemented using AppDynamics |
| 6 | API Analytics – Implemented using Splunk custom dashboards |

## Security and Connectivity

|  |  |
| --- | --- |
| Security | |
| 1 | MTLS |
| 2 | Token based authentication using OAuth |
| 3 | Permissions defined by CMA draft 0.1 Specifications, please refer to ICD for the details |
| Connectivity | |
| Dev & System Test | Mock data |
| SIT | VPN to BOI Network |
| UAT, Pre-Prod and Prod | AWS Direct Connect |

## Transaction Date Range Handling

Transactions can be filtered based on their Booking Date using the **fromBookingDateTime** and **toBookingDateTime** parameters. The filter values will be assumed to refer to the same timezone as the timezone in which the booking date for the account is maintained.

These date range would be provided as query parameter to the API call. Both the “from” and “to” date is optional. In case of these dates are not provided, following logic would be used:

* If “from” datetime is not specified, the “TransactionFromDateTime” specified in the consent would be used. In case the consent does not have this date, the date exactly 12 months in past would be selected as “from” datetime.
* If “to” datetime is not specified, the “TransactionToDateTime” specified in the consent would be used. In case the consent does not have this date, the current datetime would selected as “to” datetime.

In case the “from” and “to” datetimes specified are beyond the one specified in the consent, the consent datetimes will override these query parameter datetimes.

The API Platform will return the data for the remaining valid period specified by the filter and/or consent date time ranges.

## Transaction Filtering

While responding to the transaction request API, the individual transaction entries would be filtered based on the permission provided to the TPP during the consent process.

A TPP can have permissions for “ReadTransactionsCredits” and “ReadTransactionsDebits” or both.

It the TPP only has permission for “ReadTransactionsCredits”, they can only read credit transactions, hence only the credit transactions would be shared with such a TPP. All debit transactions would be filtered out.

It the TPP only has permission for “ReadTransactionsDebits”, they can only read debit transactions, hence only the debit transactions would be shared with such a TPP. All credit transactions would be filtered out.

A TPP will get the fill list of transactions only if they have both the permissions

## Data Masking

Following data masking rule would be applicable:

* Any 8 numeric chars in sequence will be masked to “X” for each character (any whitespace, alphabet, punctual mark or special char will break the sequence)
* Any 12 alphabet chars in sequence will be masked to “X” for each character (any whitespace, numeric char, punctual mark or special char will break the sequence)

# Logging and Monitoring Architecture

