INTERFACES

Technical - Overview

Integration Team

2018

AGENDA

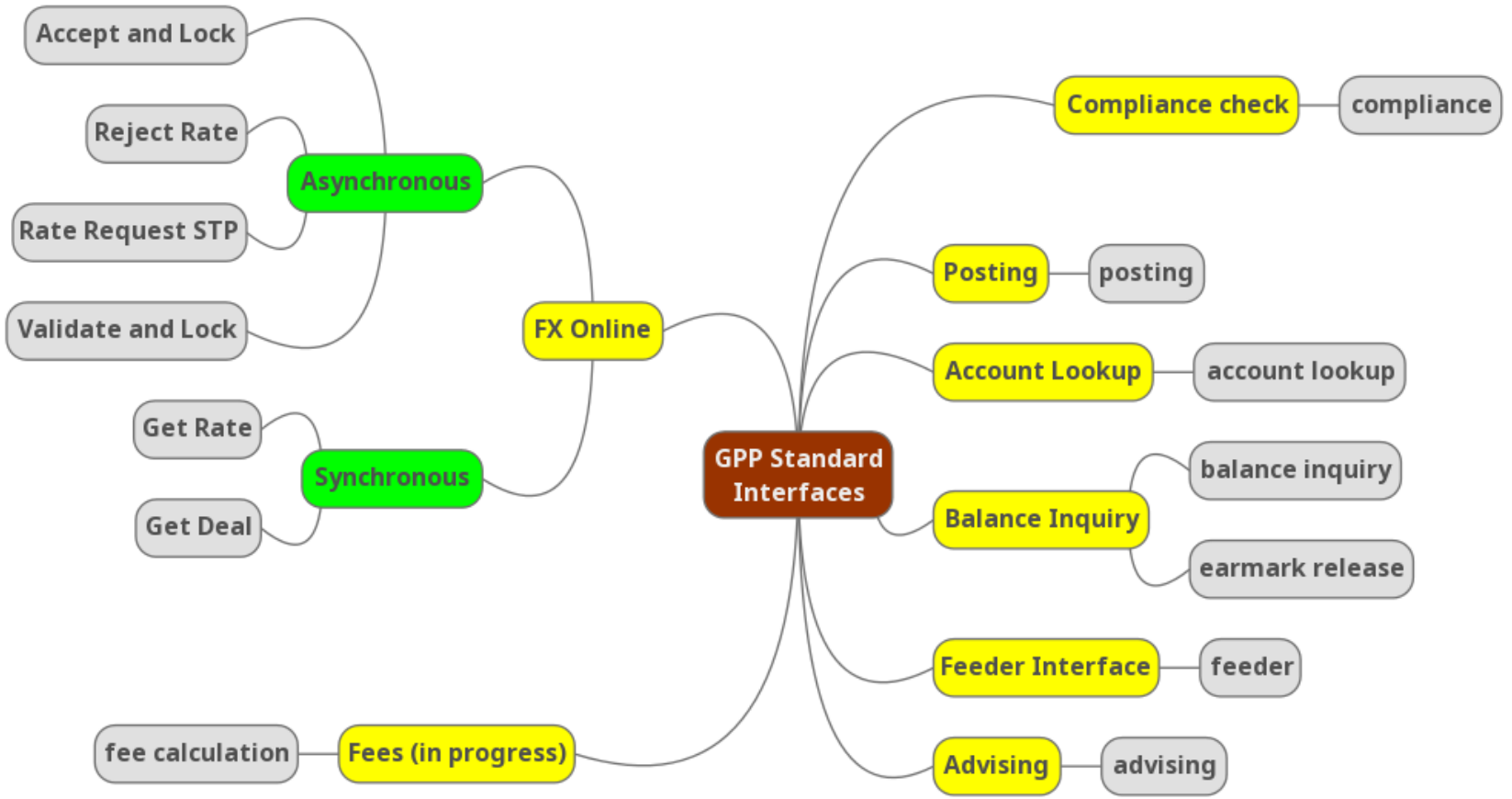
- Interfaces overview
- Common Behavior
- Request and Response
- Interface Type record structure
- UI Interface Profile

“

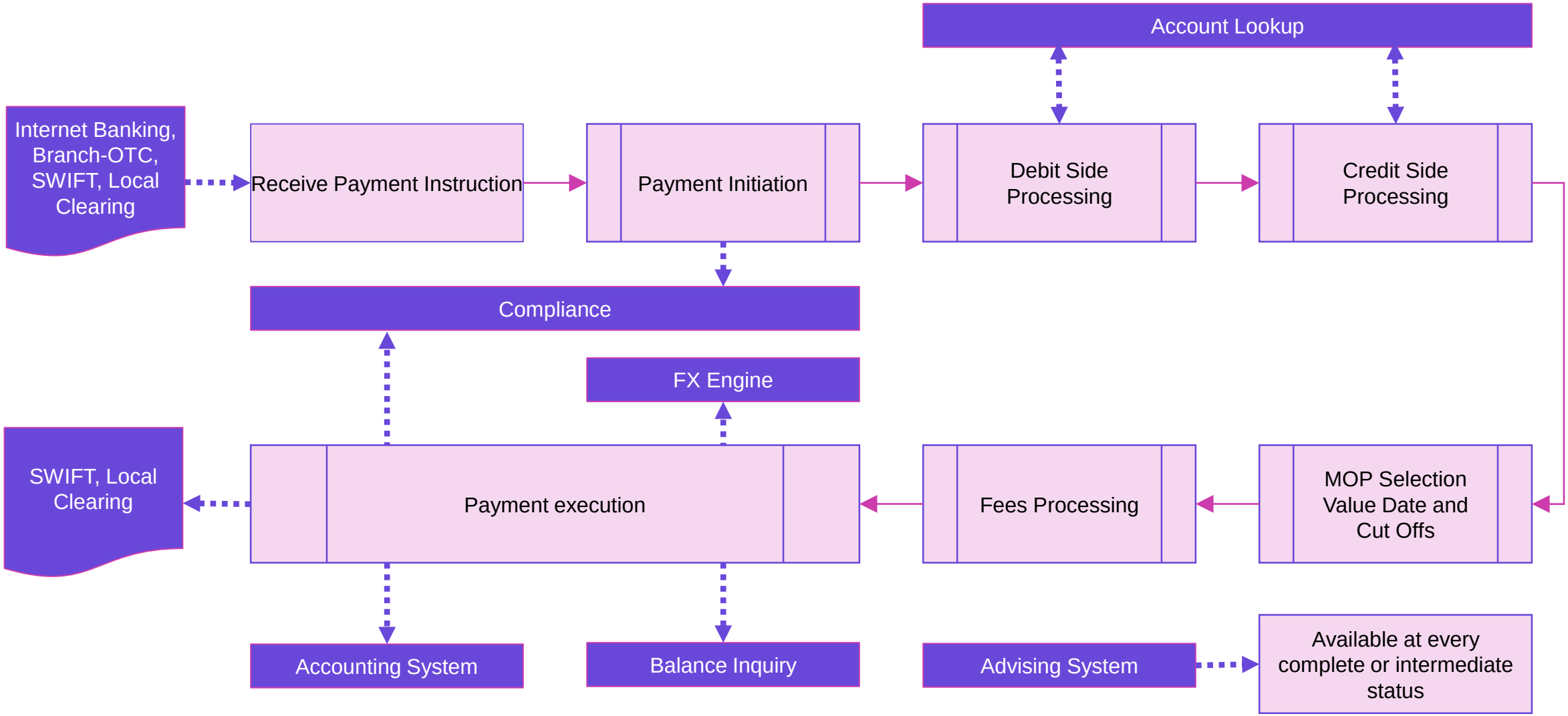
The Interfaces infrastructure is responsible for all data that enters and exits GPP.”

GPP Interfaces – Technical Guide

GPP STANDARD INTERFACES



GENERAL PAYMENT FLOW

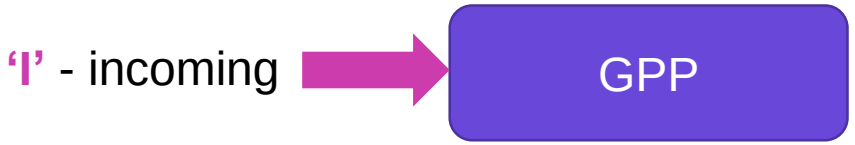


COMMON BEHAVIOR

REQUEST DIRECTION

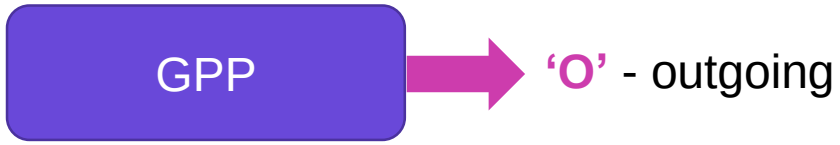
Incoming interface type

GPP functions as the server: it gets the request and returns the response.



Outgoing interface type

GPP functions as the client: it sends the request and may or may not get a response.



INTERFACE MODEL

Wait Behavior

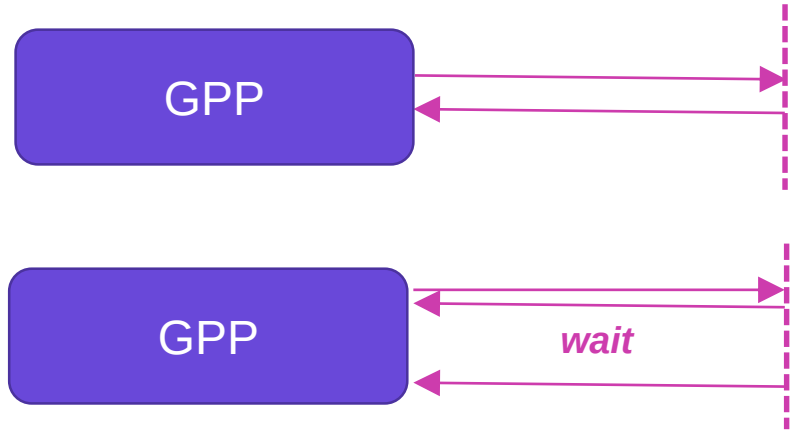
Synchronous model

Valid for incoming and outgoing types. GPP waits for a response before continuing the flow.

Asynchronous model

GPP parks the payment in a certain ‘Wait status’, as defined in the interface metadata. When a response is accepted, flow continues.

Message Wait Status is status in which the message is parked while waiting for the response.



INTERFACE STATUS

Active status

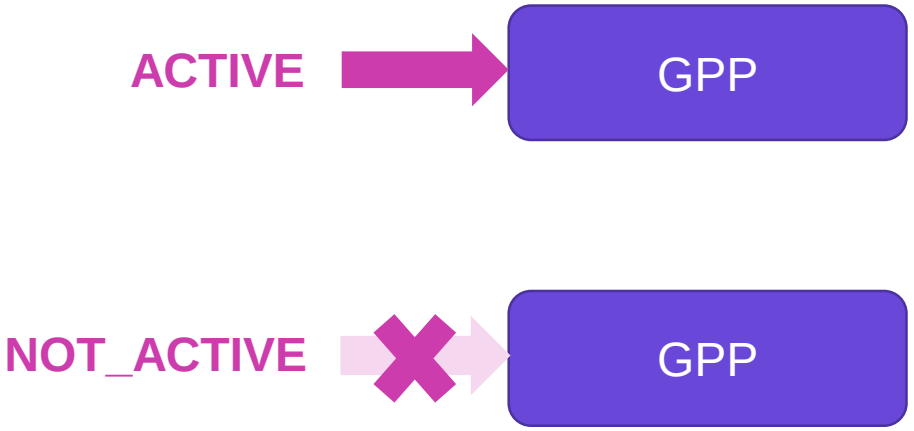
GPP communicate thru the interface. This is the default status.

Not active status

GPP does not communicate thru the interface. Status is set automatically, when GPP identifies that there is a problem in communicating with the interface.

Stop After Connectivity Exception

Number of consecutive request transmission exceptions after which the interface is marked as inactive.



Note: Switch to active is done manually by a GPP user.

CONTINGENCY MODE

Not Active behaviour

STOP_UNTIL_ACTIVE

Do not create a request, stop the flow, and change the payment message status as defined in the ‘Message Stop Status.’ Send request when reactivated. Example is ‘Account Lookup’ for posting interface.

STORE

Creates the request and saves it in database, payment message continues with the flow. Send request when reactivated. Example is ‘Stop Posting’ for the posting interface.

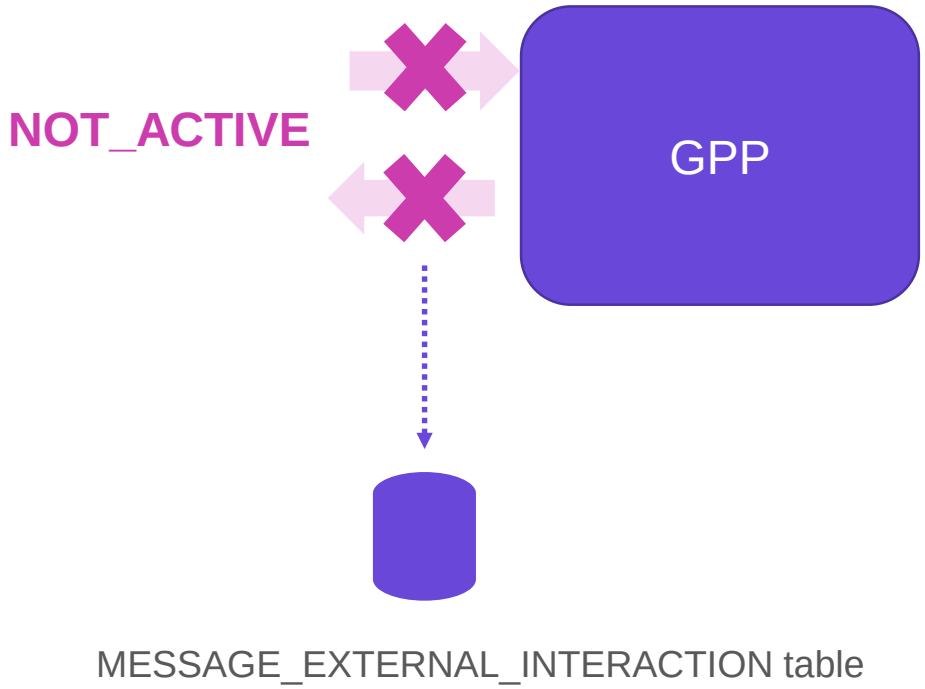
SKIP

Payment message continues with the flow and system does not create or send a request. Example is ‘Sanctions Checking’ for posting interface.

PERMANENT_STOP

Stop the flow and change the message status as defined in the ‘Message Stop Status’, no request is created. Does not send request when reactivated.

Note: Only one inactive behavior type can be applied per interface.



INTERFACE MONITOR INDEX

Track over the interface status

This monitor defines the interface status per payment message.

Examples of possible monitor values:

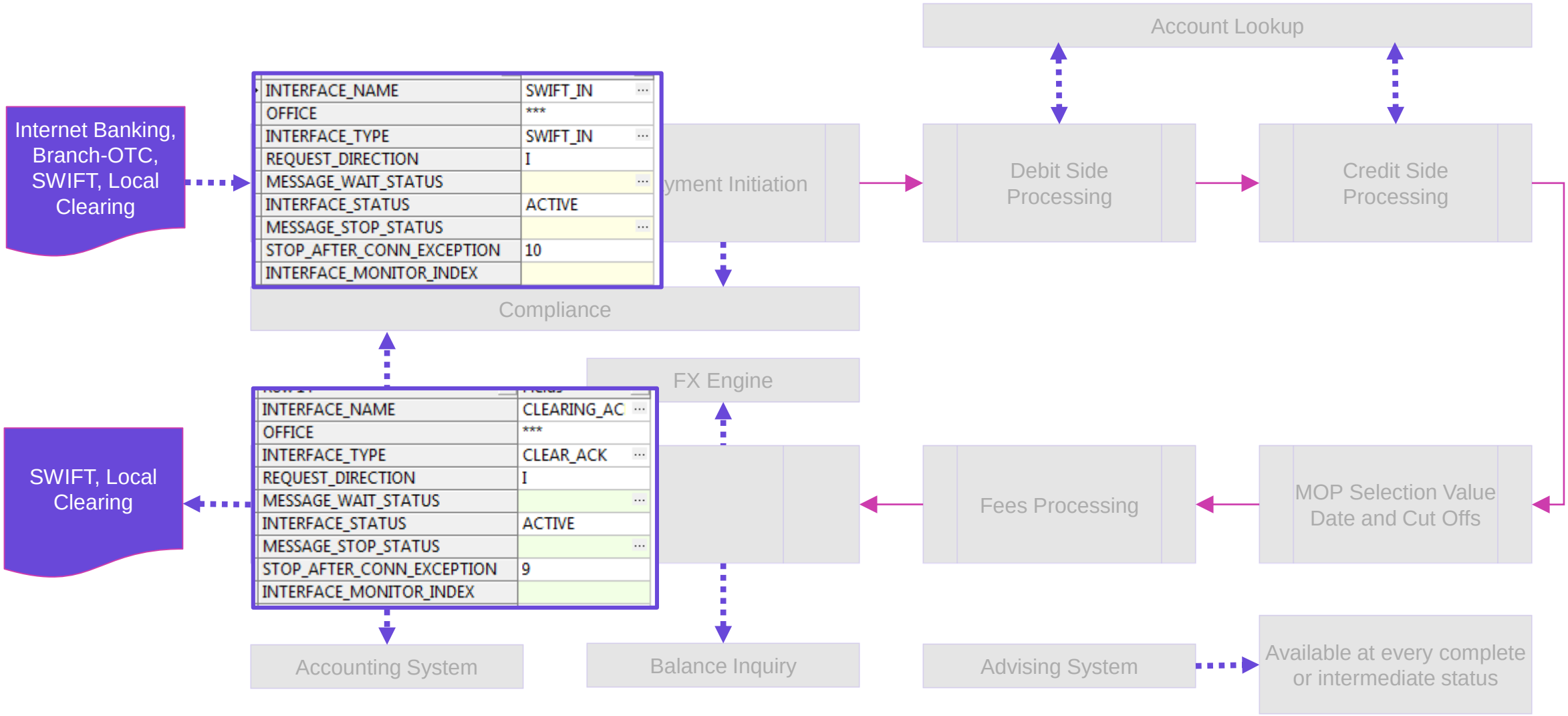
- **H -** Hold (i.e. when Not Active Behavior is set to ‘STOP’)
- **W -** Wait (i.e. when Not Active Behavior is set to ‘STORE’)
- **S -** Skipped (i.e. when Not Active Behavior is set to ‘SKIP’)
- **P –** Processed
- **X –** default value

Database location

MINF.P_INTERFACE_STATE_MONITOR

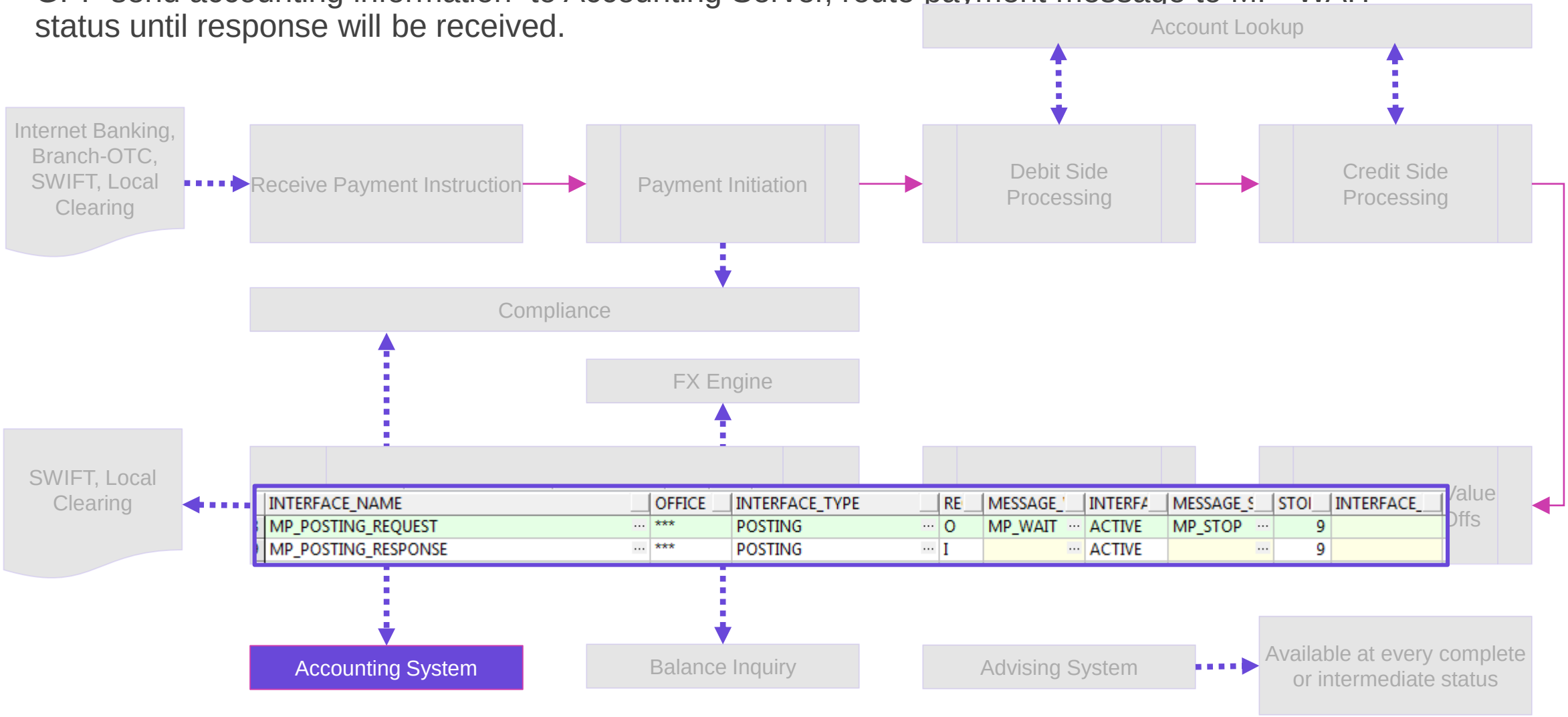
Example

FROM SWIFT TO LOCAL CLEARING ACK



ASYNCHRONOUS MODEL

GPP send accounting information to Accounting Server, route payment message to MP WAIT status until response will be received.



REQUEST AND RESPONSE

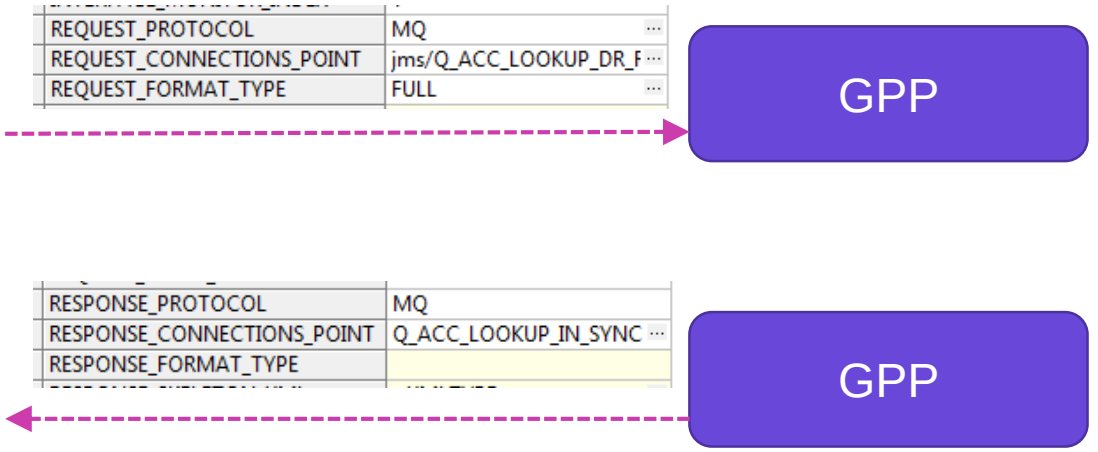
TRANSPORT PROTOCOL

- **MQ -** Java Message Service (JMS) can **guarantee message delivery. JMS** supports BACKOUT queues (failures) and listeners failover mechanism.
- **WEB_SERVICE -** SOAP over HTTP (SOAP 1.1, 1.2). Security is supported on both client and server sides (WS-Security).
- **SOAP_JMS –**SOAP over JMS used for reliability, scalability, and asynchronous messaging support.
- **FILE -** File drop is based on share folder approach.
- **MQFTE1 -** MQ FTE (File Transfer Edition) enables secure and reliable managed file transfers.

  Important: MQFTE server is required at bank side.
- **Email -** Messages with string attachment
- **SFTP2 -** Secure File Transfer protocol.

  Important: SFTP server is required at bank side

Example - **INTERFACE_TYPES** table

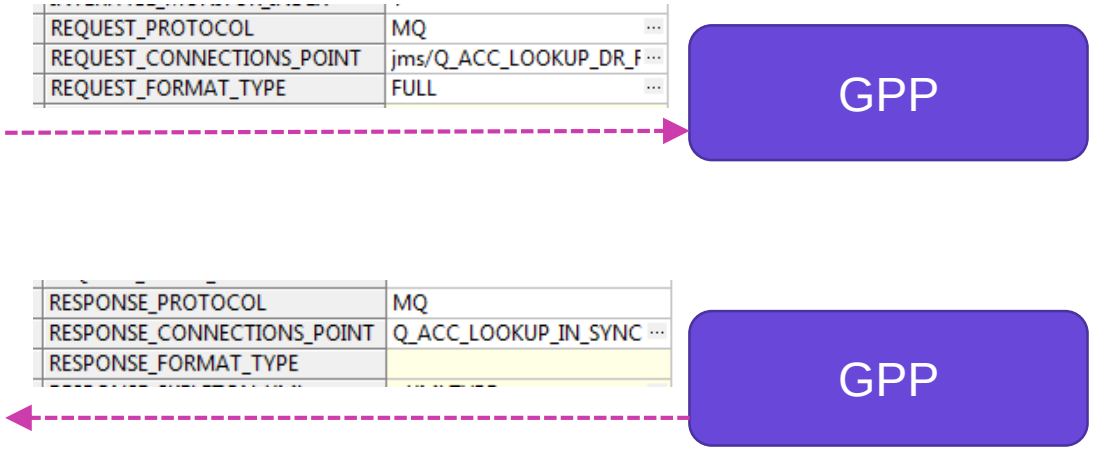


CONNECTION POINT

The actual connection point to the external systems

- JNDI name for the JMS resource (for MQ or SOAP _JMS)
- Queue name (MQ) for non JMS message queues
- Web service end point (for WEB_SERVICE and SOAP _JMS)
- Folder path (for FILE or SFTP) with permissions setup

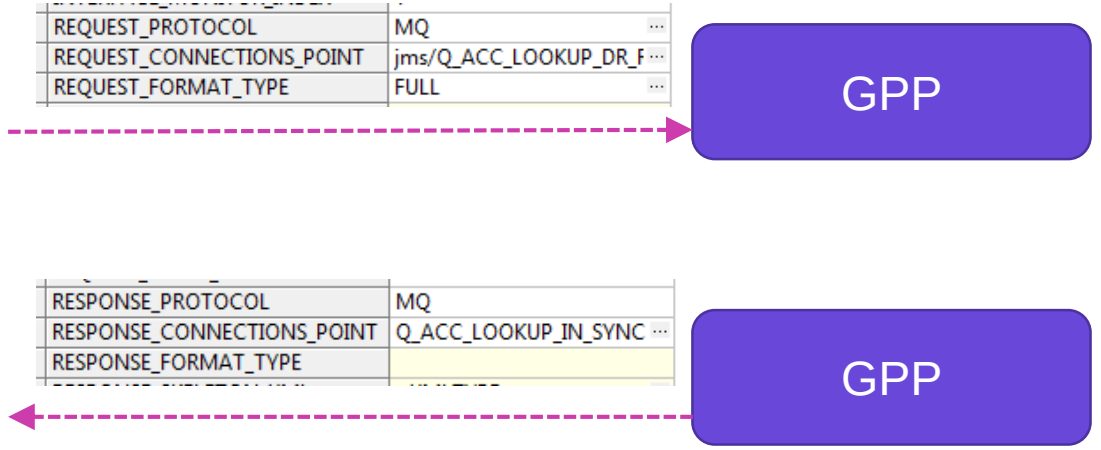
Example - **INTERFACE_TYPES** table



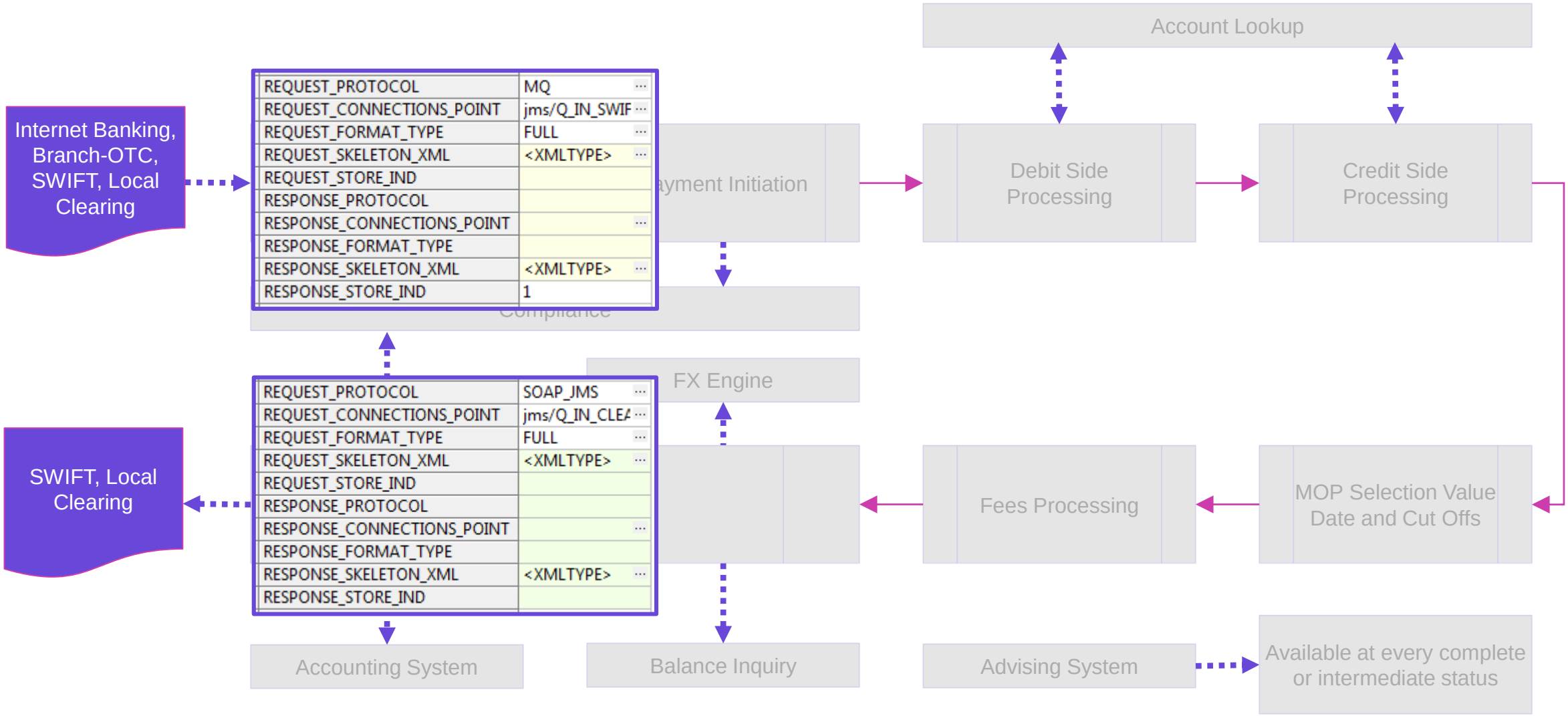
MESSAGE FORMAT TYPE

- **FULL -** full (all existing message information) Fndt Message is sent out in XML format
- **JSON –** formatted Fndt Message is sent out in JSON format
- A **subset** of Fndt Message (only a part of existing messages information) is sent out in the request
  - ACK_NOTIFY
  - Pain_012
  - CDB_OUT_CR_DEFAULT (fields list)
- **PROPRIETRY -** A proprietary structure defined for a specific customer, where Fndt Message mapped into the customer proprietary format (handled by code and ☺)

Example - **INTERFACE_TYPES** table

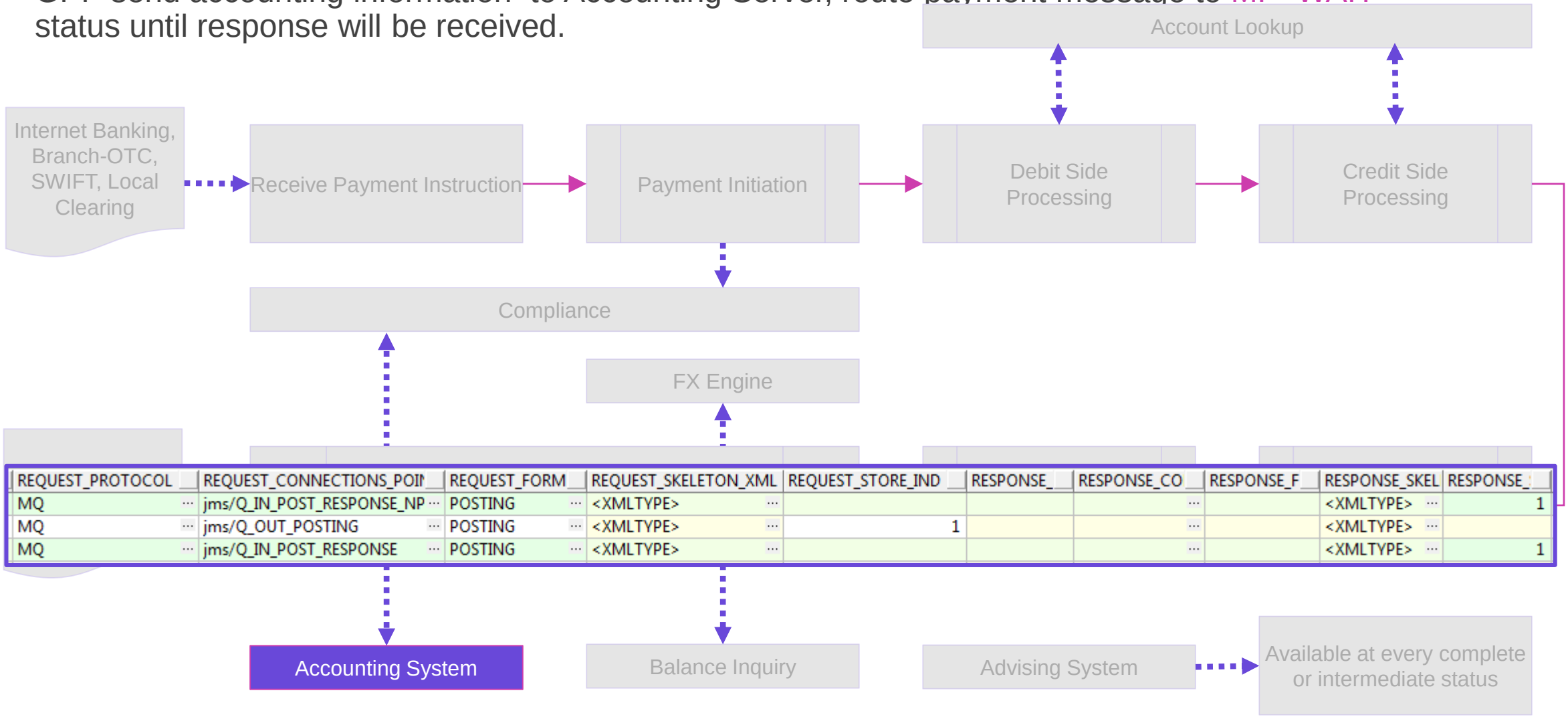


FROM SWIFT TO LOCAL CLEARING ACK



ASYNCHRONOUS MODEL

GPP send accounting information to Accounting Server, route payment message to MP WAIT status until response will be received.



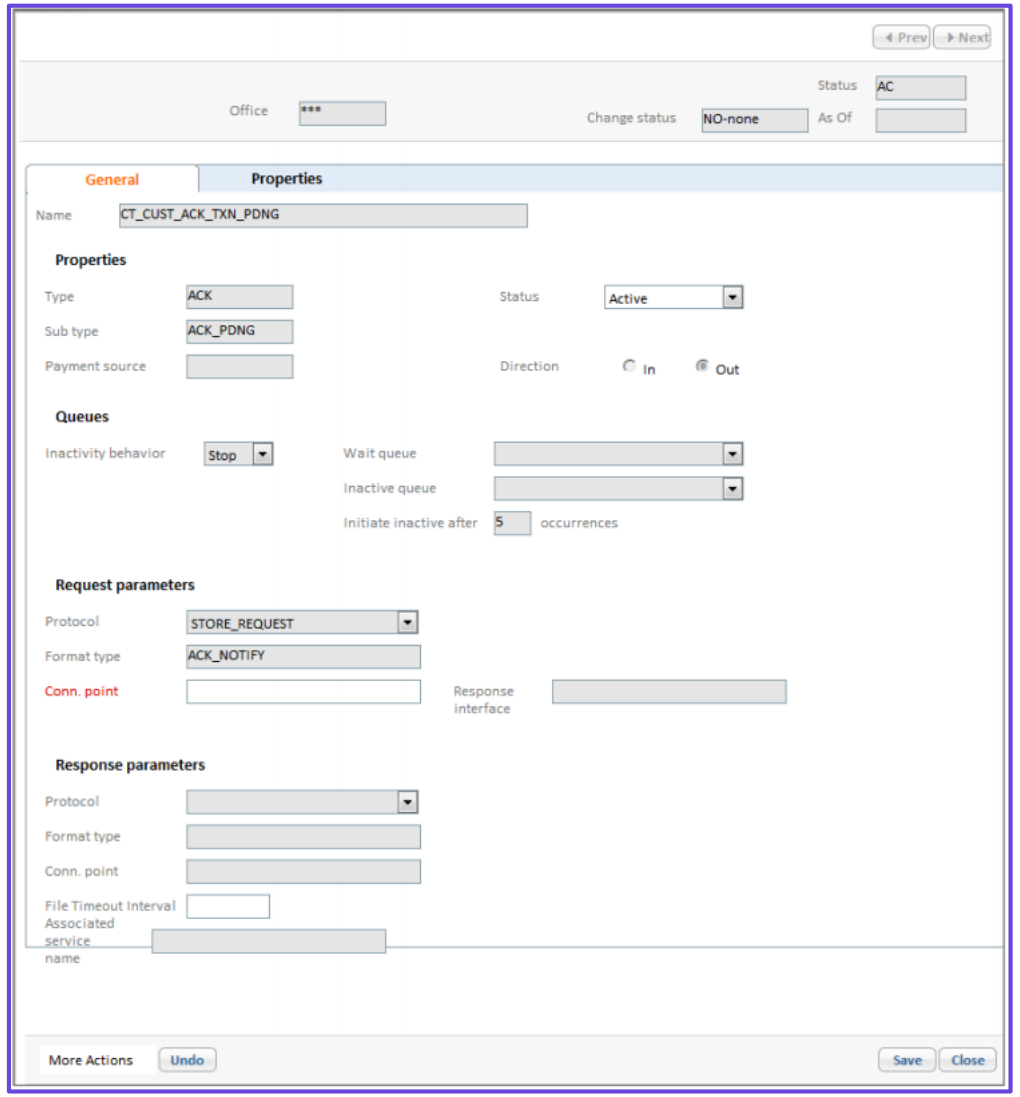
REQUEST_PROTOCOL	REQUEST_CONNECTIONS_POI	REQUEST_FORM	REQUEST_SKELETON_XML	REQUEST_STORE_IND	RESPONSE_	RESPONSE_CO	RESPONSE_F	RESPONSE_SKEL	RESPONSE_
MQ	jms/Q_IN_POST_RESPONSE_NP	POSTING	<XMLTYPE>					<XMLTYPE>	1
MQ	jms/Q_OUT_POSTING	POSTING	<XMLTYPE>	1				<XMLTYPE>	
MQ	jms/Q_IN_POST_RESPONSE	POSTING	<XMLTYPE>					<XMLTYPE>	1

INTERFACE PROFILE

User Interface Setup

The **Interface profile** in the user interface specifies, for example, the inactivity status, where payments are parked if the service is inactive, and the number of malfunction events that automate the service to Inactive status

Note: The only attributes that are open to GPP users are the **interface status** and the **interface connection** point.



Thank you

Integration Team

alexander.perman@finastra.com


@FinastraFS


Finastra LinkedIn


Finastra YouTube