



EUROPEAN CENTRAL BANK

EUROSYSTEM

SDMX artefacts in a statistical production chain



01/11/2023

Ole Sørensen
Principal Economist - Statistician

Overview

- 1 Introduction
- 2 SDMX Tool
- 3 Configurations
- 4 The Process
- 5 Data Providers, Dataflows and Provision agreements
- 6 Mappings
- 7 Reporting constraints
- 8 Validation Report

1. Introduction

ECB are in the process of migrating an existing **Statistical Data Production Environment** into a new system with new and more modern technologies, this has been presented in previous conferences.

In this presentation I will focus on a specific part of this system that is related directly to SDMX and how different **SDMX** artefacts are combined and used in the system.

We are all aware of **SDMX** artefacts like dataflows, data providers, provision agreements, constraints, mappings etc., but how are they actually used in a production environment, what tool is used and what kind of services and configurations are needed.

2. SDMX Tool

For handling metadata and the different SDMX artefacts the Fusion Registry (FR) was selected as the tool.

- Already used in the ECB for managing metadata
- Fully supporting the SDMX Data Model
- Fully supported API with different services (async) e.g., handling validation and transformations (mappings)
- Kafka support for synchronization (not part of this presentation)
- Good level of support

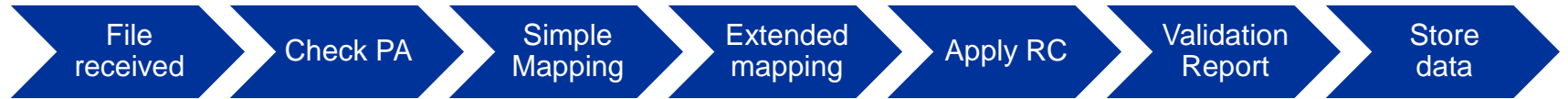
3. Configurations

Configurations is a primary functionality and used all over to handle different processes in the system. Configurations are defined either global or specific for a dataset depending on the need and can contain very granularly information. The format is done in Json.

Example of configurations related to this presentation:

- Mappings to identify if an advanced mapping is needed
- Content of the validation report
- Blocking data with errors
- Information related to dissemination

4. The process



5. Data Providers, Dataflows, Provision Agreements

Data Providers (DP) are probably the simplest artefact, but nevertheless an important component in relation to other used artefacts.

Dataflows (DF) are used for each DSD even only one Dataflow is needed, this makes it easier to distinguish the data and create specific artifacts for different purposes.

Provision Agreements (PA) are per definition a combination of the above 2 artefacts. This is used to control the received data whether this can be accepted or should be rejected. This is the minimum we need to receive data. For dissemination PA are also created to control the different targets for the dissemination.



Provision Agreements continued

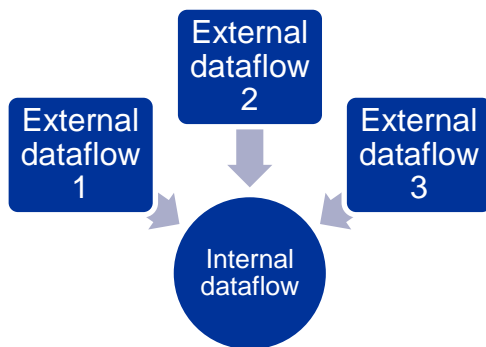
When a file is received, we check the Fusion Registry if we have a PA with a combination of the provided Sender code and the DataSetID (DataflowId) *Sender_DataflowId*, in the example below GR2_ECB_PAY4, if such combination exist, the file will be further processed and if not, it will be rejected.

```
<message:Header>
  <message:ID>IREF021099</message:ID>
  <message:Test>>false</message:Test>
  <message:Prepared>2022-03-03T07:48:21Z</message:Prepared>
  <message:Sender id="GR2">
</message:Sender>
  <message:Receiver id="4F0"/>
  <common:Name xml:lang="en">Test-H1-PEM</common:Name>
  <message:Structure dimensionAtObservation="TIME_PERIOD" structureID="ECB_PAY4" r
    <common:Structure>
      <Ref agencyID="ECB" id="ECB_PAY4" version="1.0"/>
    </common:Structure>
  </message:Structure>
  <message:DataSetAction>Replace</message:DataSetAction>
  <message:DataSetID>ECB_PAY4</message:DataSetID>
  <message:Extracted>2022-01-11T07:48:21Z</message:Extracted>
</message:Header>
```


6. Mappings

We use mappings for 2 different purposes:

1. We map the dataflow we receive from one external dataflow to an internal dataflow. We have this mapping because we in some cases have more than one external dataflow to separate the data when submitted, e.g. different frequencies. This is a simple mapping that is always provided.



Mappings

-
2. We also have more advanced mappings like series key mappings. For the system to identify if this mapping needs to be applied before it is stored in the system we have a configuration (json) for the external dataflow referencing the urn to the mapping in the FR. The system checks if such configuration exist when data is received and apply the mapping before ingested.

```
'mapping': {'STSOTHER_ECB_N':  
'urn:sdmx:org.sdmx.infomodel.mapping.StructureMap=ECB.SPACE:SPACE_DATAFLOW_MAPPING(1.0).STSOTHER_ECB_N'}
```



7. Reporting Constraints

In order to restrict the type of data we want to receive **Reporting Constraints (RC)** are created. RCs are defined as a combination of each dataflow and DP and are attached to a PA for the same combination.

We use a naming convention similar to the PA (Sender_DataflowId), in this way we can create different RCs for each DP referencing the related PA. When data is received and validated against the PA we also check if there for the specific DP and dataflow exist RCs and if so they are also used in the validation for the final validation report.



Reporting Constraints

Examples of RCs referencing the different PAs

Id	Type	Constrained Structure(s)
AT2_ECB_PAY4	ProvisionAgreement	AT2_ECB_PAY4[1.0]
BE2_ECB_PAY4	ProvisionAgreement	BE2_ECB_PAY4[1.0]
BG2_ECB_PAY4	ProvisionAgreement	BG2_ECB_PAY4[1.0]
CY2_ECB_PAY4	ProvisionAgreement	CY2_ECB_PAY4[1.0]
CZ2_ECB_PAY4	ProvisionAgreement	CZ2_ECB_PAY4[1.0]
DE2_ECB_PAY4	ProvisionAgreement	DE2_ECB_PAY4[1.0]
DK2_ECB_PAY4	ProvisionAgreement	DK2_ECB_PAY4[1.0]
EE2_ECB_PAY4	ProvisionAgreement	EE2_ECB_PAY4[1.0]
ES2_ECB_PAY4	ProvisionAgreement	ES2_ECB_PAY4[1.0]
FI2_ECB_PAY4	ProvisionAgreement	FI2_ECB_PAY4[1.0]
FR2_ECB_PAY4	ProvisionAgreement	FR2_ECB_PAY4[1.0]
GR2_ECB_PAY4	ProvisionAgreement	GR2_ECB_PAY4[1.0]



8. Validation Report

We take advantage of the validation report provided by the Fusion Registry. As mentioned previously we can configure the validation of the data and the content of the validation report.

Our DSDs (and dataflows) all have attributes that are mandatory normally used for the dissemination. When collecting data we use the same DSD/Dataflows but we don't need these attributes reported and to avoid errors when validating we can configure that the data is accepted and ingested in the system, and we can also configure that these errors are removed from the validation report.



