Open-source CIM-CGMES compliant power flow.

iPST/PowSyBl day

Luis María Zamarreño
Aplicaciones en Informática avanzada (AIA)

Outline

- → Motivation and goals
- → Functional requirements
- → Design considerations
- → Implementation
- → Workflow for main use cases
- → Validation
- → Next steps & challenges

Motivation

- ➔ Previous versions of iPST provided support of ENTSO-E **CIM Profile 1** (approved by ENTSO-E in **2009**) for importing power system network models.
- ➔ In **2014**, the ENTSO-E approved the Common Grid Model Exchange Specification (**CGMES**) that would replace CIM Profile 1 for grid model exchanges between European TSOs.
- ➔ The implementation of CGMES in iPST/PowSyBl project became mandatory to ensure **compliance with ENTSO-E standards**

Goals

- The **open-source CIM-CGMES compliant power flow** has been developed as a collaboration project between RTE and AIA.
- This project exploits the experience gained throughout the **FP7 EU iTesla**, as well as developments made in the **post-iTesla** open source project **iPST**.
- The **main goal** is the implementation of the **Common Grid Exchange Model Standard (CGMES)** in the iPST open source project, now **PowSyBl**.
- The **ultimate goal** is to go with iPST/PowSyBl through the **ENTSO-E Conformity Assessment** as a power flow tool (old IOP tests: interoperability for grid model exchange).

Functional requirements

- ➔ Support for **file exchange rules** defined in the standard.
- ➔ Support for **individual** (single authority) and **assembled** models.
- ➔ Preserve non-standard information (private **extensions**).
- ➔ Support for **bus-branch** and **node-breaker**. Validation of given topology / computation of topology if not given.
- ➔ Support for **strict** and **loose** import.
- ➔ **Update** and **full** export modes.
- ➔ Design for processing of **difference files** (although they will not be implemented in a first version).

Functional requirements

Main changes with respect to ENTSO-E Profile 1

- ➔ Improvements in **power system equipment modeling**: Transformers, Static VAR compensator, HVDC equipment, Equivalent injections, Equivalent Branches, etc.
- ➔ **Tabular data for ratio and phase tap changers** to allow description of change in phase/ratio as an explicit curve.
- ➔ **Steady State Hypothesis profile** (SSH) that contains input data for power flow calculations.
- ➔ Other profiles: Dynamics, Diagram Layout, Geographical Location. Data should be preserved in import/export, but will not be processed.

Functional requirements

Focus on initial use cases

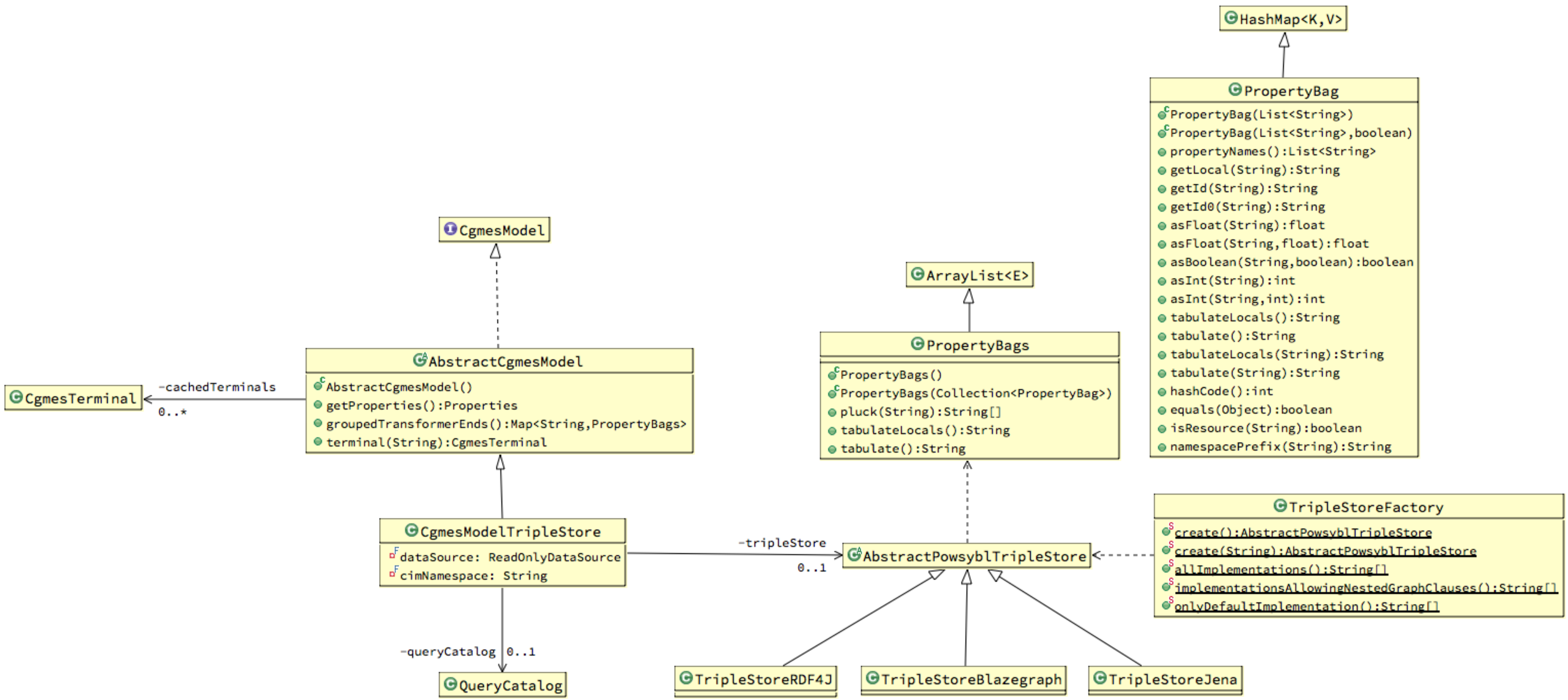
- ➔ Run a power flow over a CGMES model, put results back in CGMES State Variables (SV profile).
- ➔ Full export an iIDM model (built from any other source) to CGMES.

Design considerations

- ➔ Alternatives for handling CGMES data: CIM gateway, triple store.
- ➔ **CIM gateway** was used for ENTSO-E Profile 1. A **set of Java classes** corresponding to a specific version of the CIM object model is generated from XMI files (XML Metadata Interchange) by an external tool.
- ➔ **Triple store**. A **specific purpose database** oriented to the storage and retrieval of triples, the data predicates used by the RDF serialization of CIM objects. **RDF/XML CIM files** are directly ingested, stored and exposed as triplets. Database can queried/modified using a query language (**SPARQL**).
- ➔ Triple store was selected: easy **support for extensions**, easier maintainability for **supporting different CGMES versions**, open-source libraries available, **no big performance penalty**.

Implementation

CGMES model on triple store



Implementation

Triple store

- ➔ Abstraction for a triple store: uses a catalog of queries, returns data as lists of property bags.
- ➔ Available implementations for triple store: Jena, RDF4J, Blazegraph.
- ➔ CGMES data is kept in separate datasets (graphs): EQ, TP, SV, SSH, ...
- ➔ Found limitations on SPARQL queries with GRAPH clauses in Blazegraph.
- ➔ Not using GRAPH clauses (to keep maximum compatibility between all possible engines) adds a small performance penalty.

Implementation
SPARQL Queries

```
# query: terminals
SELECT *
WHERE {
{ GRAPH ?graph {
    ?Terminal
        a cim:Terminal ;
        cim:Terminal.ConductingEquipment ?ConductingEquipment .
    ?ConductingEquipment a ?conductingEquipmentType .
}}
OPTIONAL { GRAPH ?graphSSH {
    ?Terminal cim:ACDCTerminal.connected ?connected
}}
OPTIONAL { GRAPH ?graphSV {
    ?SvPowerFlow
        a cim:SvPowerFlow ;
        cim:SvPowerFlow.Terminal ?Terminal ;
        cim:SvPowerFlow.p ?p ;
        cim:SvPowerFlow.q ?q
}}
}
```

Implementation
CGMES model

- Abstraction for a CGMES model: collections of objects returned as property bags.
- Only Terminals that connect conducting equipment are implemented as first class Java classes.
- Only available implementation is a triple store backed CGMES model.
- Ability to handle CIM14 (ENTSO-e Profile 1) and CIM16 (CGMES) using a different catalog of queries.

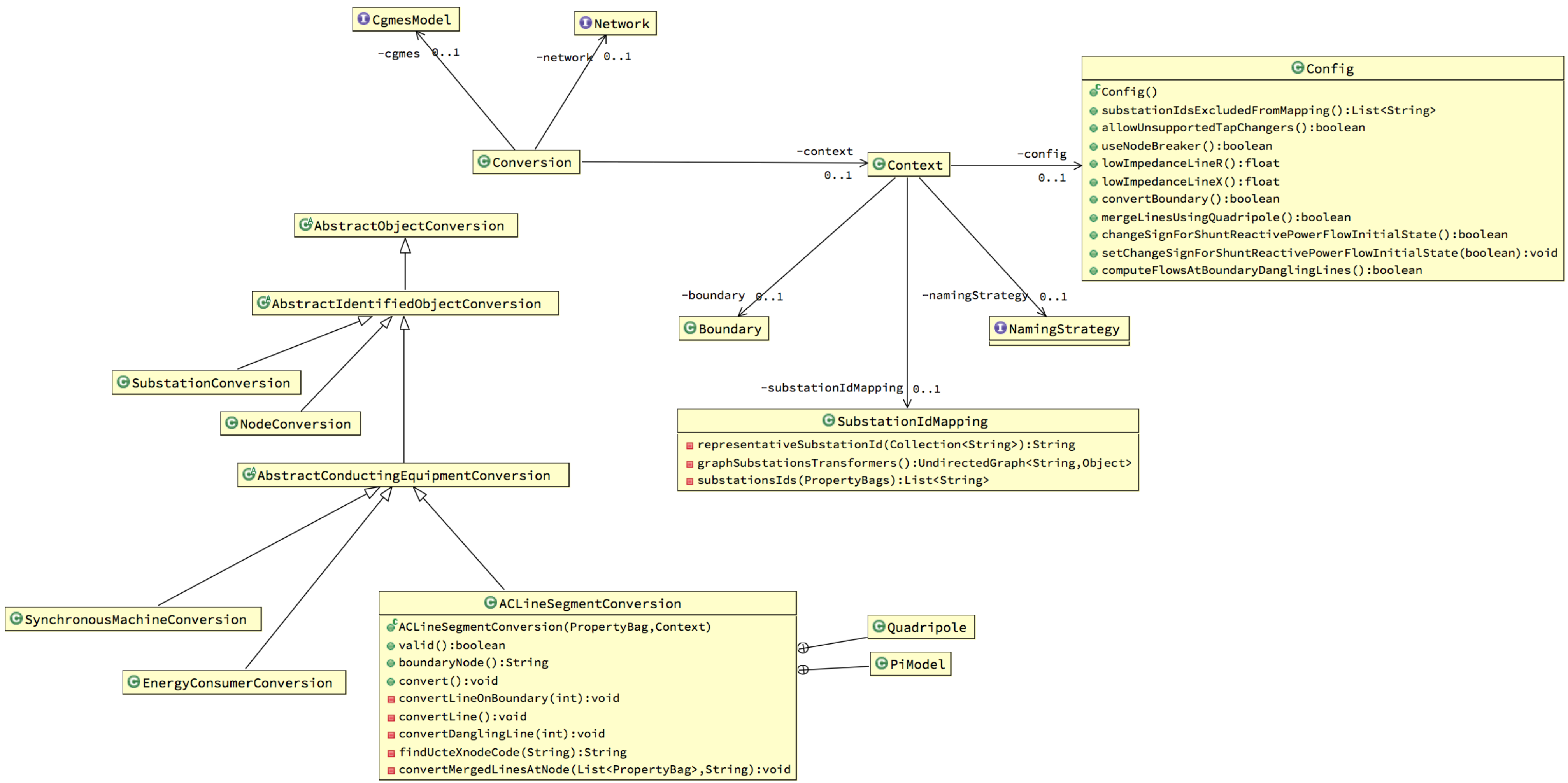
Implementation

Conversion to iIDM

- ➔ Implementation of standard PowSyBl interface for Importer and Exporter.
- ➔ Boundaries can be imported explicitly for debugging purposes. Boundary nodes and related elements (Equivalent Injections, External Network Injections, ...) will be mapped to iIDM objects.
- ➔ Specific Conversion classes for every relevant CGMES class that has to be mapped to iIDM, extending an abstract Conversion.
- ➔ AbstractConductingEquipmentConversion deals with Terminals and associated data (related bus, voltage level, power flow).

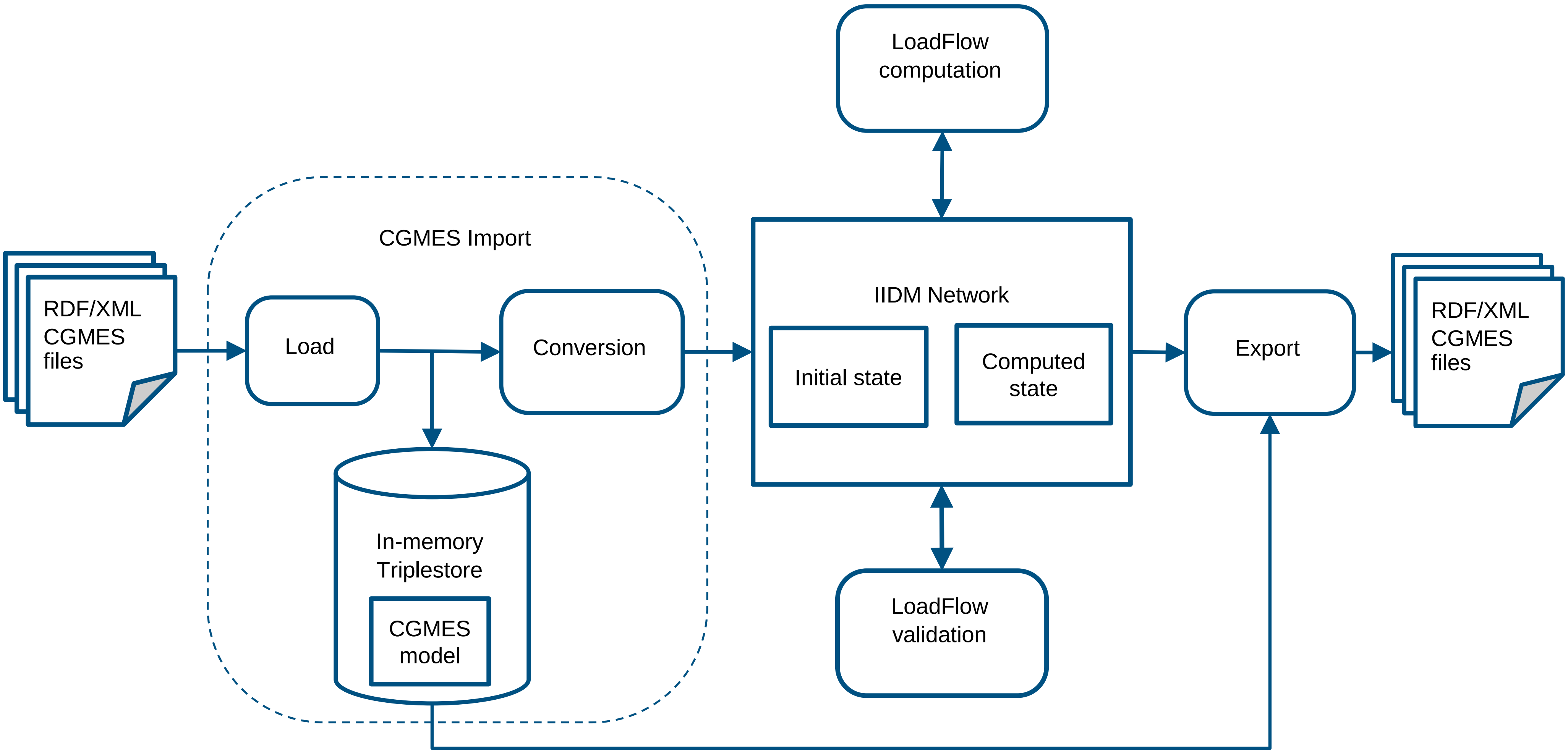
Implementation

Conversion to iIDM



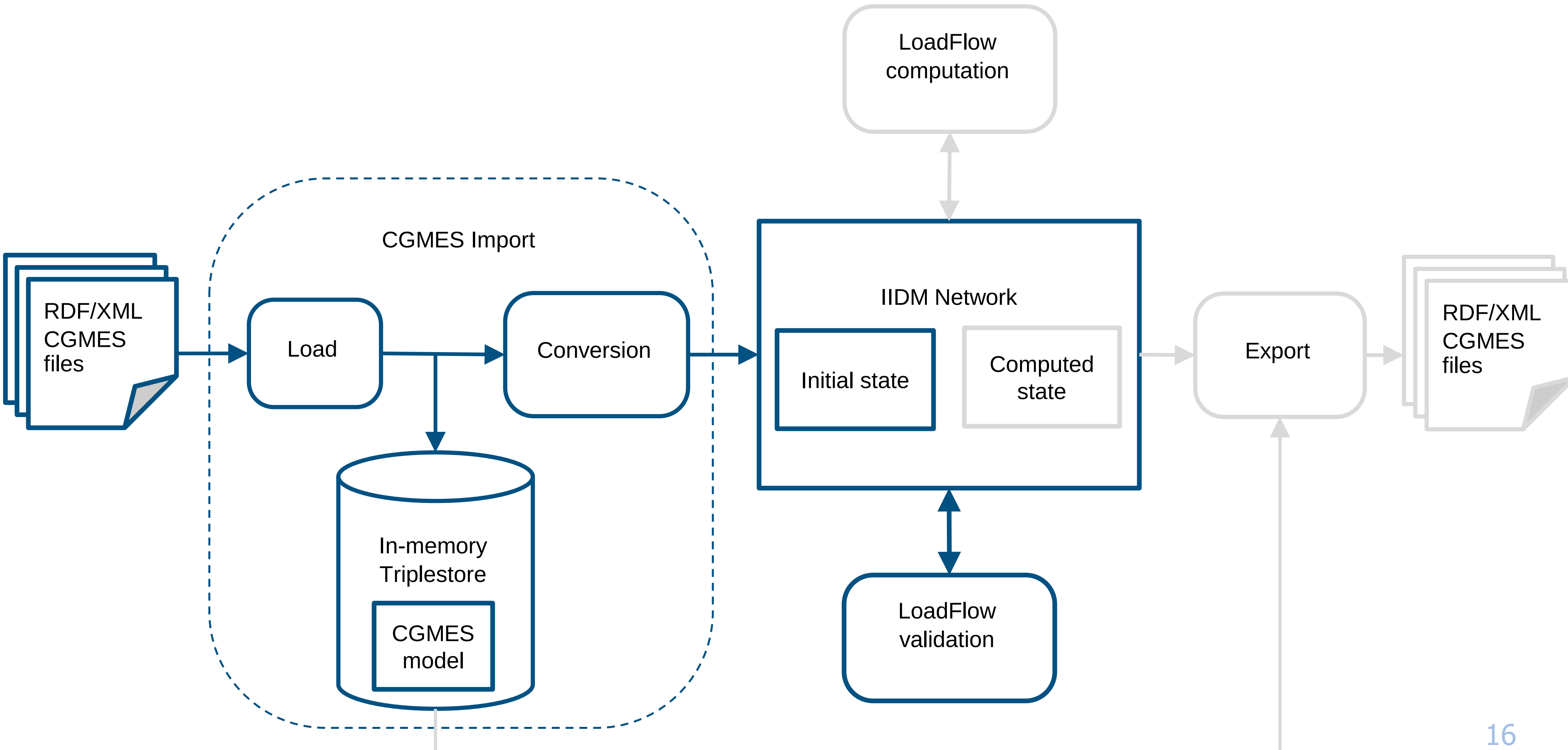
Workflows for main use cases

CGMES - IIDM conversion



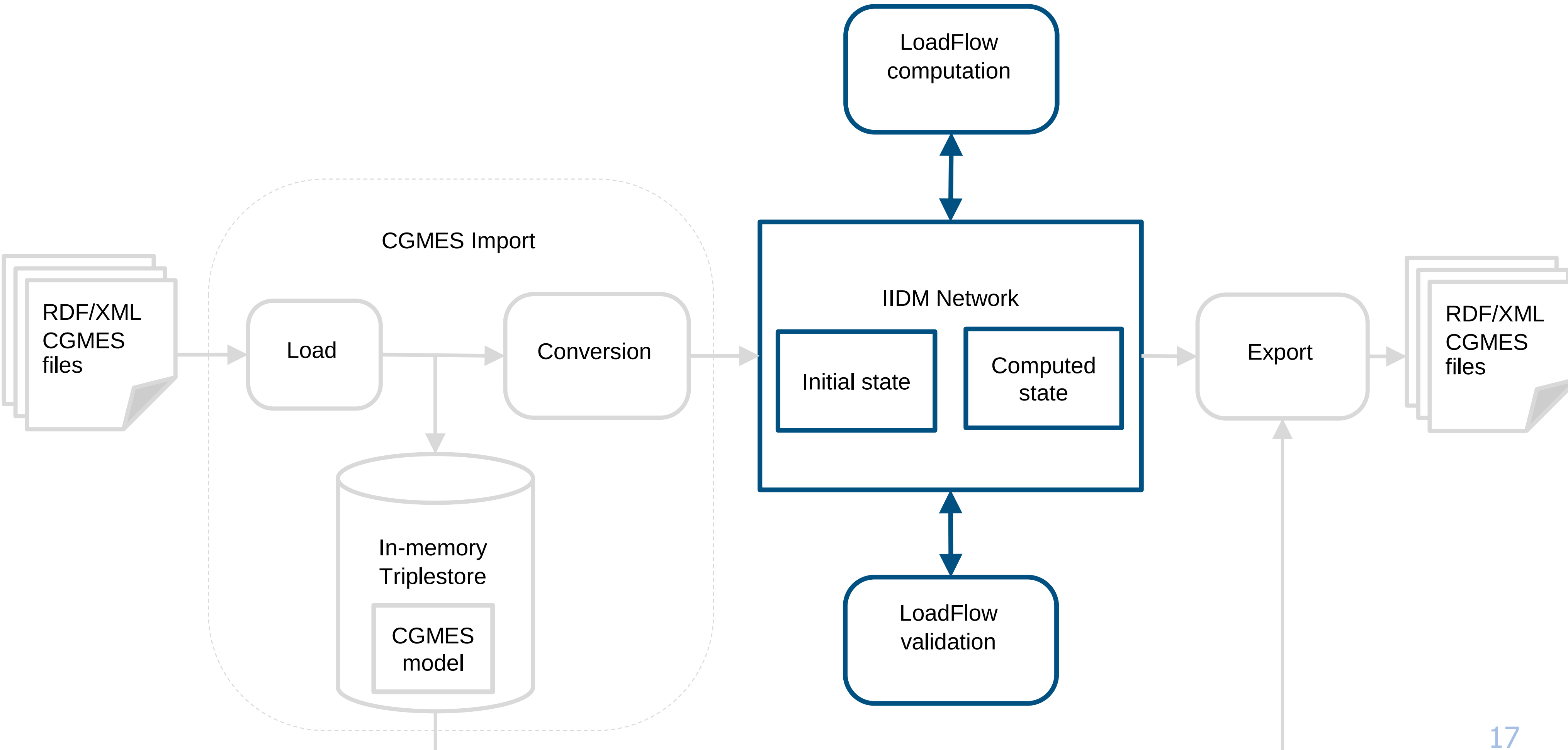
Workflows for main use cases

Validation of CGMES data for a solved power flow exchange



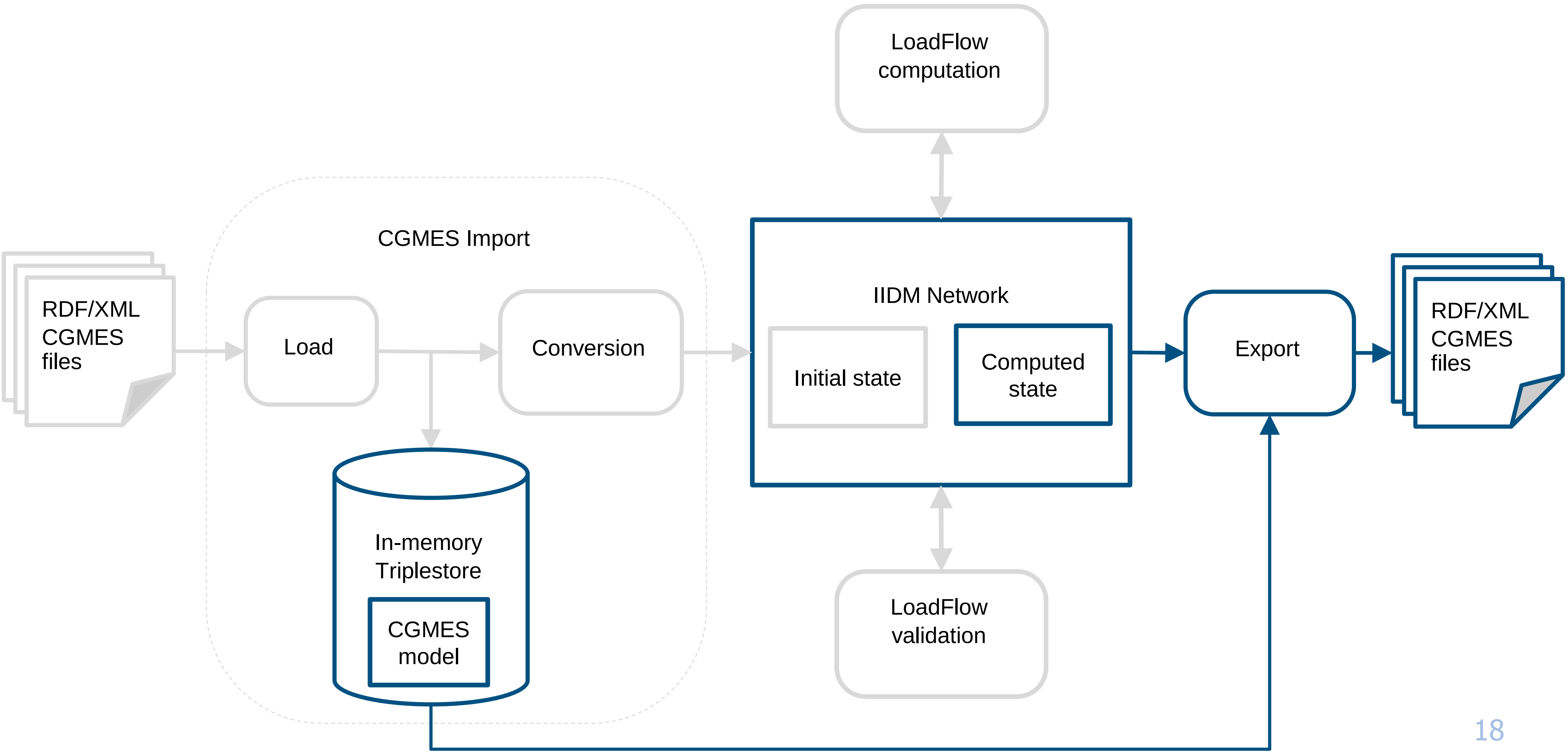
Workflows for main use cases

LoadFlow computation and validation



Workflows for main use cases

Export computed power flow results to CGMES



Validation

Steady state validation

- ➔ Steady state validation function from PowSyBl is used to check input data and verify computed power flow results (will be explained in detail by N. Omont).
- ➔ Some solved power flow cases (SV data) only have voltages. Corresponding flows have not been exported to SV file.
- ➔ A new LoadFlow results completion tool was implemented in PowSyBl by TechRain to handle this situation and compute missing flows on all branches.
- ➔ It is of great help to spot problems in test cases and helped identify errors during conversion implementation:
  - Assigning power flows at dangling lines created at boundaries.
  - Computing the merge of AC Line Segments that connect at a boundary in an assembled model (must use Quadripoles so computed flows match with given voltages).
  - Verify the interpretation of magnetizing branch conductance/susceptance for transformers (different test cases assume different criteria).

Validation

Test cases

- ENTSO-E Conformity test cases (v4.0.3 of test cases were used).
  - **Micro**: BE, NL, Assembled. 3 substations in the Assembled.
  - **Mini**. Bus-branch and node-breaker variants. 5 substations.
  - **Small**. Bus-branch and node-breaker variants. 105 substations.
  - **Real**. Bus-branch. 4875 substations.
- Small CIM14 test cases: IEEE14, 7buses, Nordic32. Used to validate queries and conversion works for ENTSO-E Profile 1.
- Real grid data from Europe DACF cases (21-02-2018):
  - 15 bus-branch.
  - 13 node-breaker.

Validation

Findings

➔ **Precision** in data, computations and results.

- Currently all numeric data in iIDM is stored using type float.
- Some LoadFlow engines may compute using type double.
- Voltages and flows computed by a LoadFlow engine using double floating point precision will be converted to single when stored in IIDM objects.
- Validation of steady state using converted float values may fail. Precision mismatch is more evident with branches with low impedance.
- Example: conformity test case Real, line _922078528_ACLS.

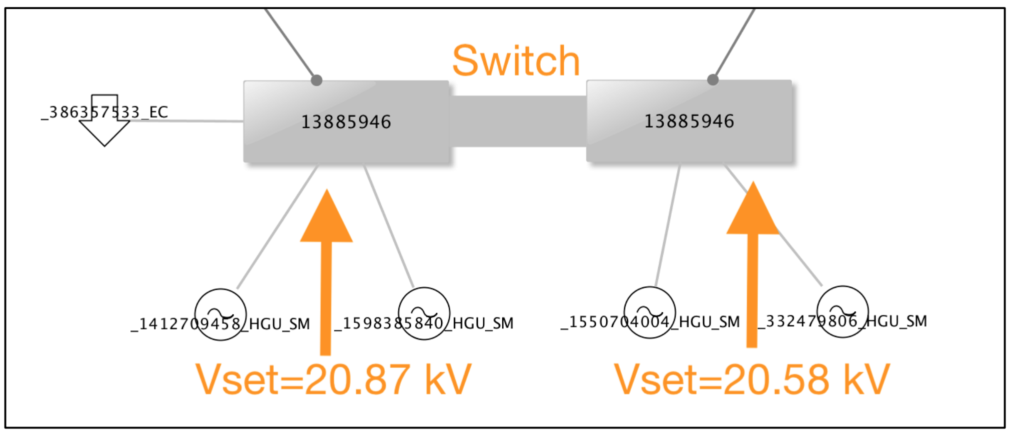
➔ **Recommended to move to double precision in iIDM**.

Validation

Findings

➔ **Conflicting setpoints in generators.**

- Different generators regulating voltage at buses connected by switches.
- Generators have different setpoints. So, not all setpoints can be satisfied at the same time.
- Currently validation reports that some generators fail to regulate voltage.
- Examples in conformity Real grid test case: 1550704004_HGU_SM, 1572135050_HGU_SM, 1942932982_HGU_SM, 745344672HGU_SM.



➔ **Steady state validation could be enhanced to warn about these situations.**

Validation

Findings

➔ Some TSOs use **EquivalentBranch** objects.

- Equivalent branches are the result of network reduction.
- They may define a dissymetrical branch (Z12 <> Z21), a modeling not yet available in iIDM.
- Observed in DACF cases.

➔ Different criteria **interpreting (g, b) in transformers**.

- Magnetizing branch conductance/susceptance.
- Some TSOs seem to compute power flow assuming g/2, b/2 at each end of PI model.
- Other DACF cases are computed considering (g, b) at only one end.

➔ In some samples, **connected buses have voltage = 0**.

- Buses are connected to energized part of the network through a breaker.
- They have no other equipment connected.
- Adjacent bus has voltage > 0.
- Wrong voltage at bus, v = 0. Also, angle <> 0 and different of the angle of adjacent bus.

Validation

Findings

- ➔ Wrong flows in **low reactance** lines.
    - Seems power flow was solved applying a correction for minimum value allowed for reactance.
- ➔ **Inconsistent data in tap changer** position.
    - Some of the DACF samples had tap position = 11, when the tap changer was defined with positions [low, high] = [-8, 8].
- ➔ **Unexpected connectivity**.
    - Found a switch connecting one node inside the model with a node located in the boundary.

Next steps & Challenges

- ➔ Fine tuning of conversion for 3-winding transformers and 2-winding transformer with multiple tap changers.
- ➔ Extensive testing on real data (both node-breaker and bus-branch samples).
- ➔ Update CGMES Conformity test configurations to Conformity Assessment Scheme (CAS) v2.0.
- ➔ Merging use cases. Validation of IGM (Individual Grid Models). Splitting of CGM (Common Grid Model).
- ➔ Full export mode.
- ➔ DC modelling (current CGMES flexibility in DC network modelling can not be directly mapped to iIDM objects).

Additional content

Design considerations

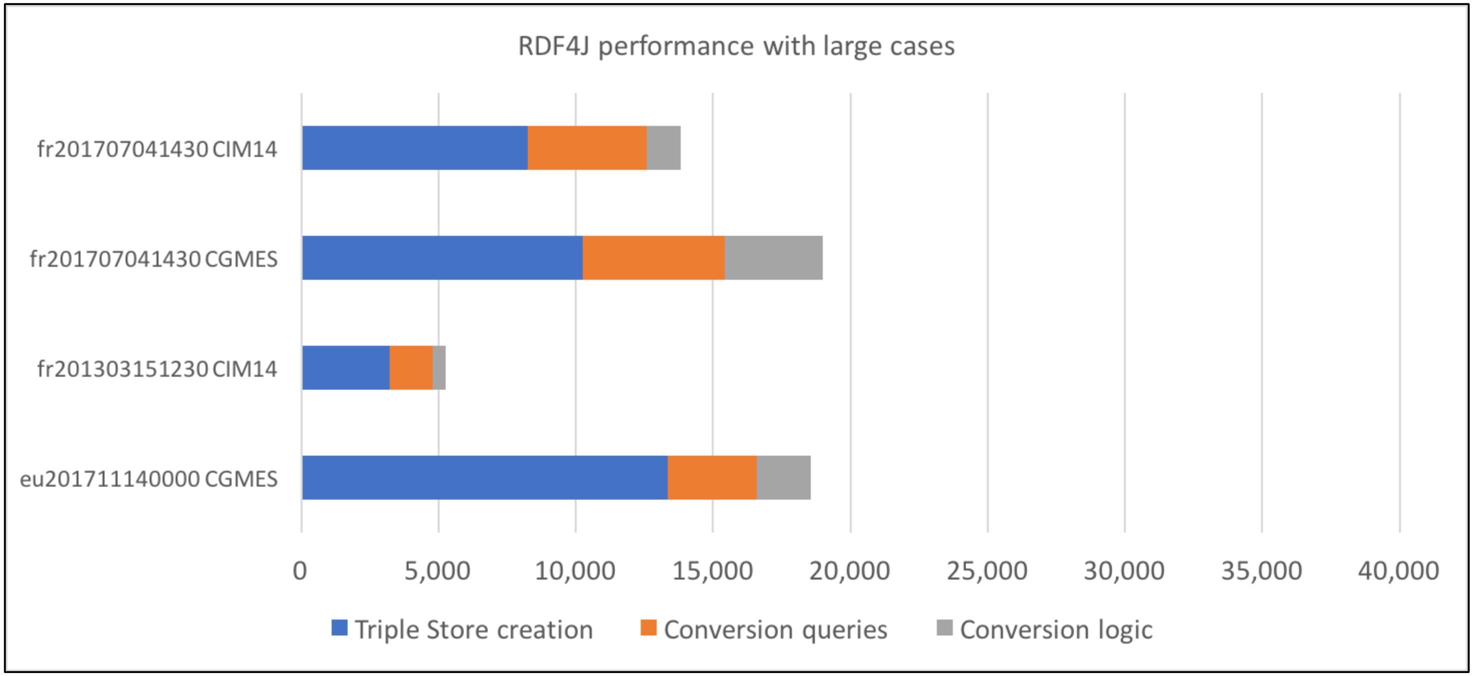
CIM gateway.

+ Performance (considering only data ingestion).
+ Simplicity of the converter (RDF to grid data complexity hidden in gateway).
- The CIM gateway is a complex tool, very difficult to maintain and evolve.
- No support for extensions.
- Handling different (even closer) versions requires lots of duplicated code.

Triple store.

- Performance (although penalty is not critical).
- Complexity of the converter (but queries can help hide RDF to grid data).
+ Multiple open-source libraries for database available.
+ Support for extensions.
+ Handling different versions is simpler (only adapt queries and conversion).

Performance



Machine: Intel Core i5 760 2.8GHz, 8GB of RAM. Windows 7 64 bit.
Different real-size networks (RTE and EU).
Performance measurements made in January 2018.

Next steps & Challenges
Merging use cases

- ➔ Snapshot merging.
- ➔ DACF merging using net position data.
- ➔ Advanced hybrid DACF-SN merging using methods developed during the iTesla FP7 project.