ITER Plant Configuration System

Experience with using EPICS 7 at ITER

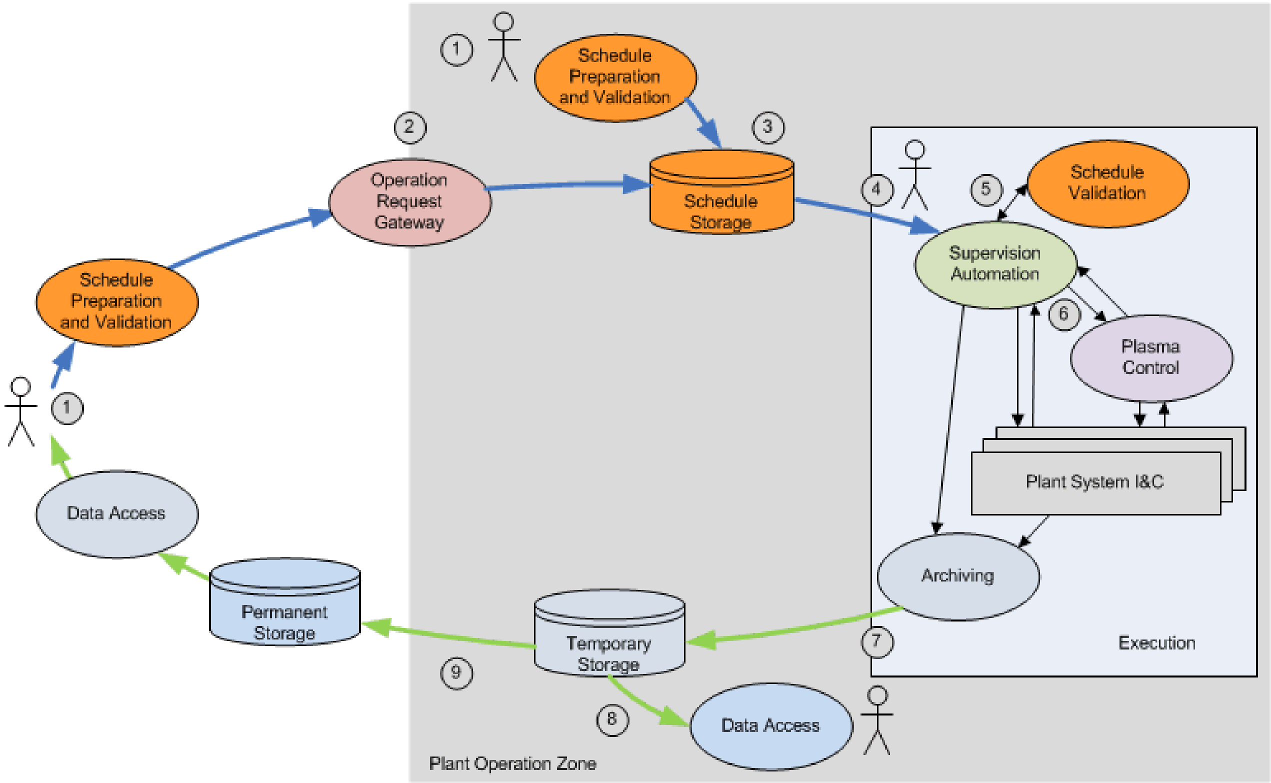
B. Bauvir (CSD)

A. Neto (F4E)

Outline

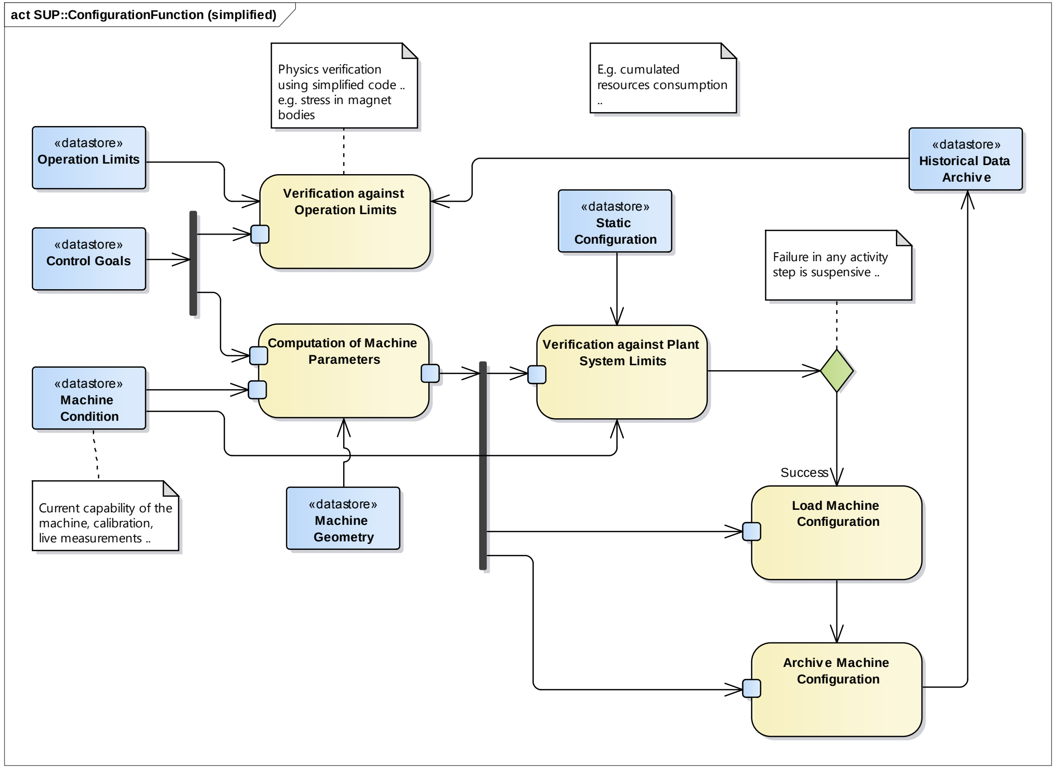
- Context
- Objectives and architecture
- Successes and difficulties
- Conclusions and future work

Context



1
Schedule Preparation and Validation
2
Operation Request Gateway
3
Schedule Storage
4
5
Schedule Validation
Supervision Automation
6
Plasma Control
Plant System I&C
Archiving
Execution
7
Temporary Storage
8
Data Access
9
Permanent Storage
Data Access
1
Schedule Preparation and Validation
Plant Operation Zone

Context



act SUP::ConfigurationFunction (simplified)
Physics verification using simplified code .. e.g. stress in magnet bodies
E.g. cumulated resources consumption ..
«datastore» Operation Limits
Verification against Operation Limits
«datastore» Historical Data Archive
«datastore» Static Configuration
«datastore» Control Goals
Failure in any activity step is suspensive ..
Computation of Machine Parameters
Verification against Plant System Limits
«datastore» Machine Condition
Success
Current capability of the machine, calibration, live measurements ..
«datastore» Machine Geometry
Load Machine Configuration
Archive Machine Configuration

Context

- Plant configuration as part of the pulse preparation
  - Derivation of machine parameters from physics/operations' goals
    - Adaptation of the domain of machine operations to the technical domain of the Plant System I&C without requiring manual translation by an expert
    - Adaptation to the current condition of the machine, reduced availability due to broken parts
  - Verification incl. potentially complex code and simulation (operating instructions, scarce resource budget planning, etc.)
    - Non-trivial validation, consistency of parameters across the machine
  - Providing Machine Operations with support to manage complexities
    - Detect and notify inconsistencies across the machine, limit conditions, etc.
    - Obviously Machine Operations are empowered to ignore, override, etc.

Objectives

- Further and enhance current methods, tools, interface for Plant System configuration
  - Plant system configuration from Central I&C should adhere to defined Quality of Service criteria
    - Ensure data integrity through to remote controllers
    - Provide exception handling and reporting mechanism
- Manage interfaces with data repositories, with Plant Systems
  - Plant System manufactured and delivered without information about central services
    - Data must be delivered to Plant Systems in the form they expect
- Provide means to accommodate changes with affordable resources

Architecture

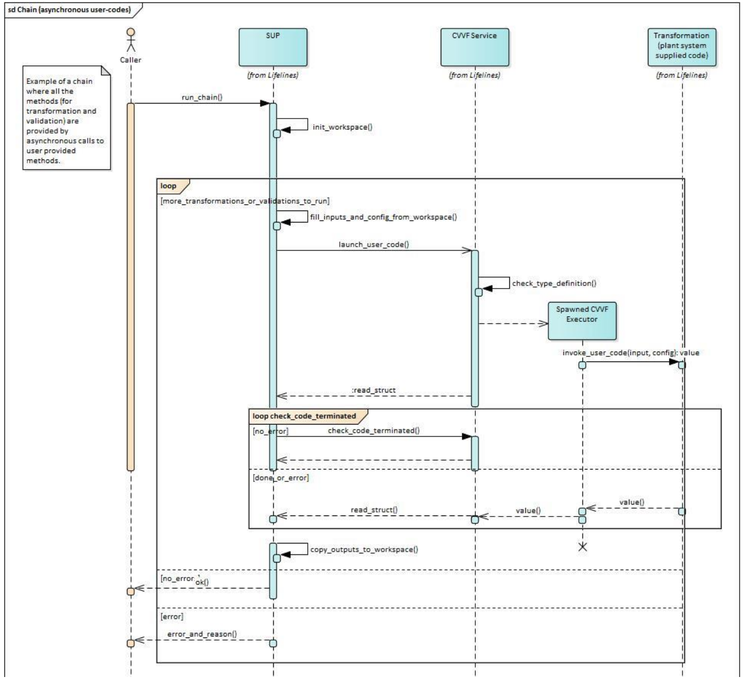
- Middle-layer services atop delivered Plant System I&C interfaces
  - Domain and interface adapters
- Separate services for chain data processing, and User-supplied code execution
  - Compiled code (at least strict version controlled) vs. interpreted workflow logic (adapt to current User, intended use), diverse languages
- Asynchronous/concurrent execution model
  - Capability of executing User-supplied code which might take longer time to execute
- Synchronous transaction model to load plant system
  - Atomic load operation, parameters acceptance/rejection en-bloc

Configuration data model

- Domain-specific language describing a chain of data processing
  - Variables and named data types of any complexity
  - Relationships though named transformation code
    - E.g. ‘55a0::cvvf::ComputeBestIPWeights/v1.0’
  - Verified through named code
  - Satisfied through interfaces
    - Text files, databases, data archive, process variables (pvAccess, Channel Access), etc.

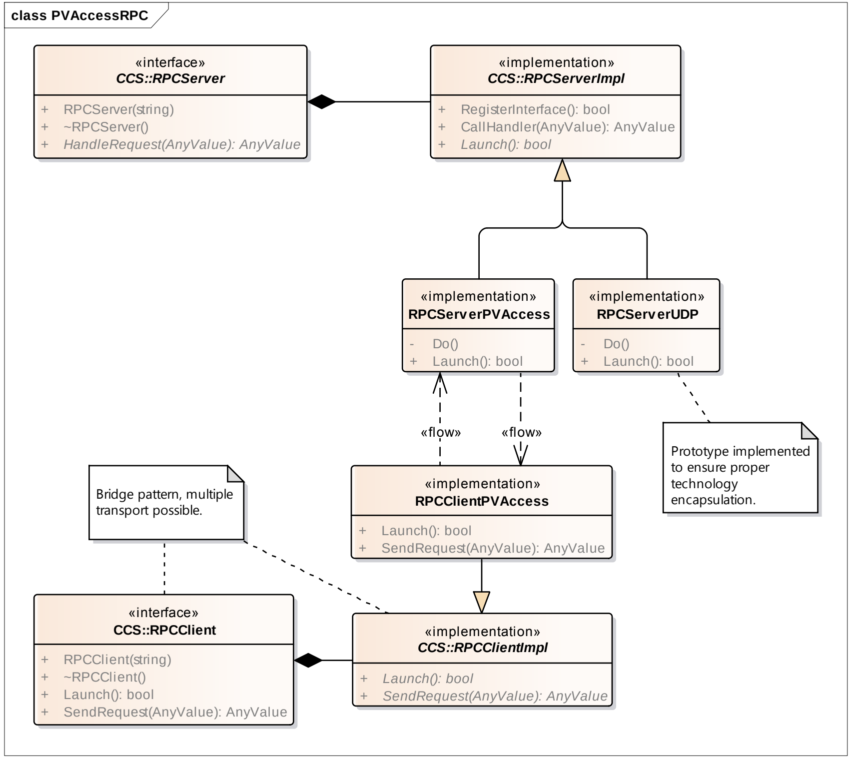
Chain workflow executor

- MARTe2 (https://vcis.f4e.europa.eu/marte2-docs/master/html)
  - Data-driven configuration
  - Perfect architecture match
    - Processing organized along the variable dependency path
  - Strong decoupling
    - Between signals and data processing
    - .. and with data providers
  - Support for built-in code execution
    - Data types transformation, composition
    - Simple verification (e.g. within bounds, etc.)



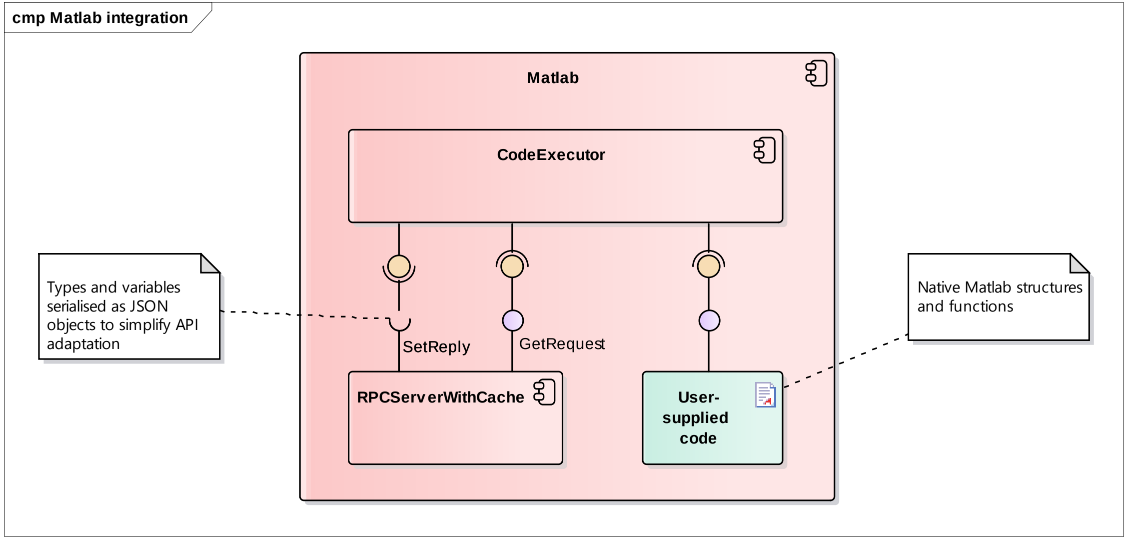
Remote Procedure Call (RPC) services

- Named RPC services providing
  - Dynamic services discovery (TBD)
  - Introspection of data types and function prototypes
  - Asynchronous/concurrent execution model



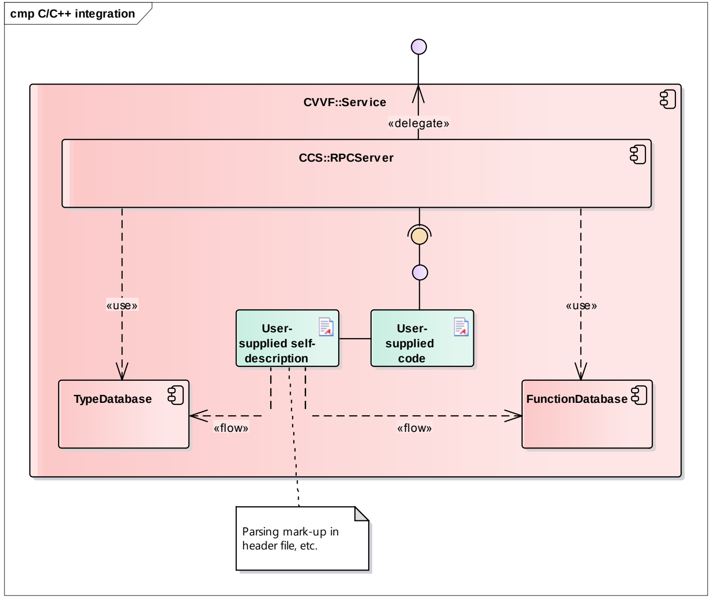
Code delivery and execution

- Code encapsulation, no dependency to any framework
- Promote usage of programming language native types
  - C/C++, Matlab and Python



Code delivery and execution

- C/C++ code and structures require additional introspection capabilities, e.g.
  - Parsing mark-up in header files
  - Generated from Plant System Self-Description Data (SDD) model

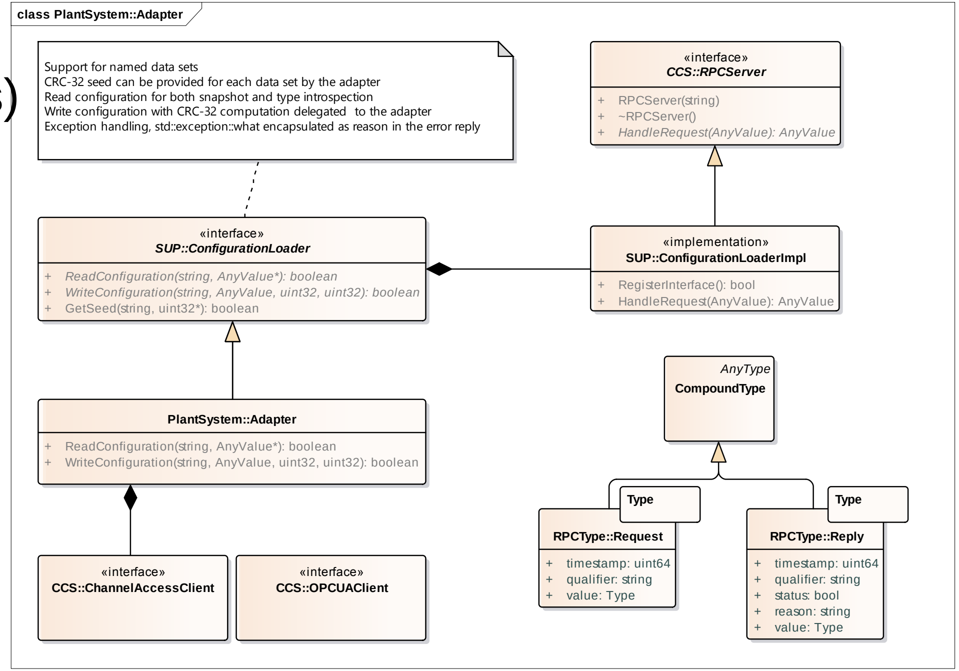


Plant System interface

- Plant System architecture
  - Large systems composed of distributed devices and controllers
  - Complex hierarchies and compositions, mutable assemblies
  - Both top-down or bottoms-up views of the plant system favour a structured representation of the data model
- Plant Operation Network (PON) interface
  - EPICS IOC and records database are deployed as mailboxes between CODAC and remote controllers (PLC, RTC, DAQ, FPGA)
  - Concerns about the data integrity through to remote controllers

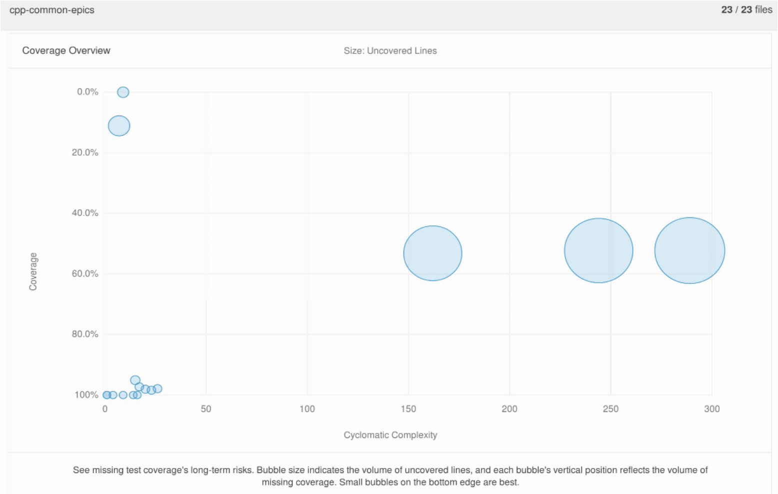
Plant system interface

- RPC service providing
  - Named data sets (discriminate between devices, life-cycle phases)
  - Introspection of data types
  - Synchronous transaction model
  - Support for integrity verification
  - Exception handling
- Adapters required and foreseen
  - Serialization to EPICS records
  - .. to OPC UA servers
  - And bespoke implementations



Successes

- EPICS 7 (pvAccess) matches very well the need
  - Support for structured data of any complexity
  - Dynamic discovery of services
  - Built-in data type introspection
- Able to implement a representative scale proof of concept in about 4 man-months
  - Incl. tests and intensive code coverage
    - Handling PVArray is challenging
  - C/C++ and Matlab code encapsulation



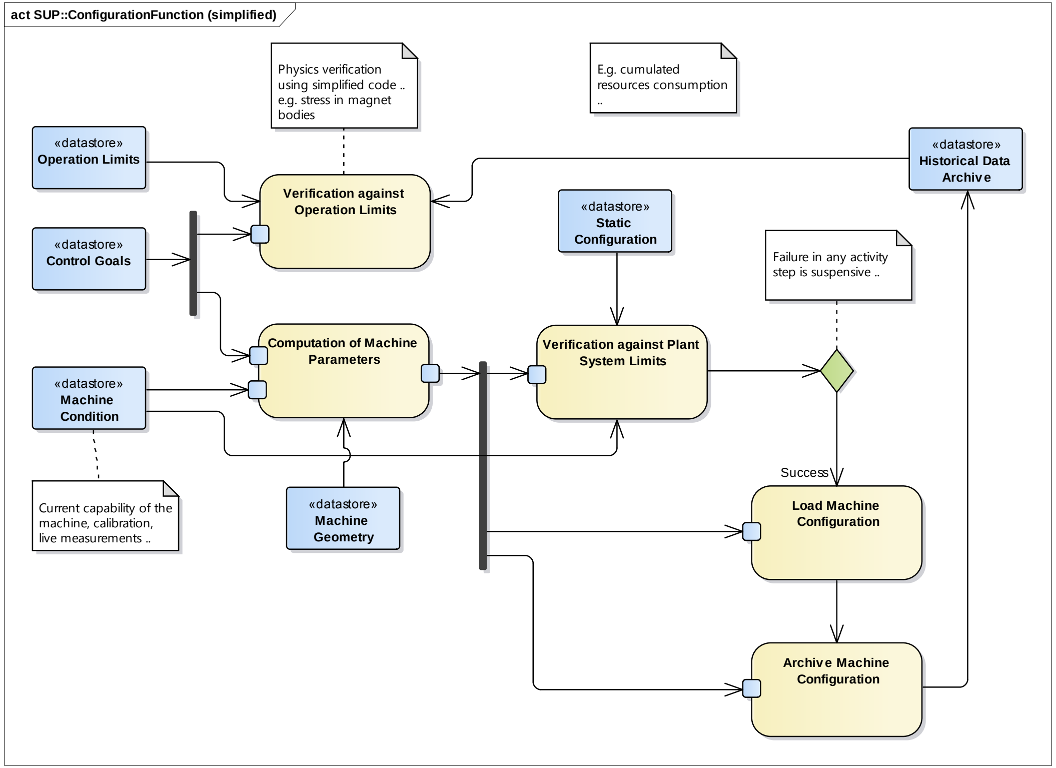
Difficulties

- Documentation
- Multiple interfaces available
  - Concerns about evolutions, maintainability
    - ‘epics::pvAccess::RPCClient’ vs ‘epics::pvaClient::PvaClientRPC’
- Arrays of structure
  - Command line tools, CS-Studio (pvmanager) do not support addressing beyond reaching array
  - Concerns about evolutions, pvAccess support at large in the EPICS ecosystem, mismatch between intended and actual use

Forecast

- EPICS 7 full part of CODAC Supervision and Automation (SUP) components
  - Middle-layer services atop delivered Plant System I&C interfaces
    - Domain and interface adapters, homogenization of interfaces, automation
  - Careful with technology adoption
    - Full encapsulation, no implementation detail to transpire through to the application code
  - Attentive to evolutions in the EPICS ecosystem (CS-Studio)
- Objective
  - Homogeneous handling of structured variables in application code
    - Identical application-side API, whether PON, or SDN, or ..

Thank you for your attention



Back-up slides

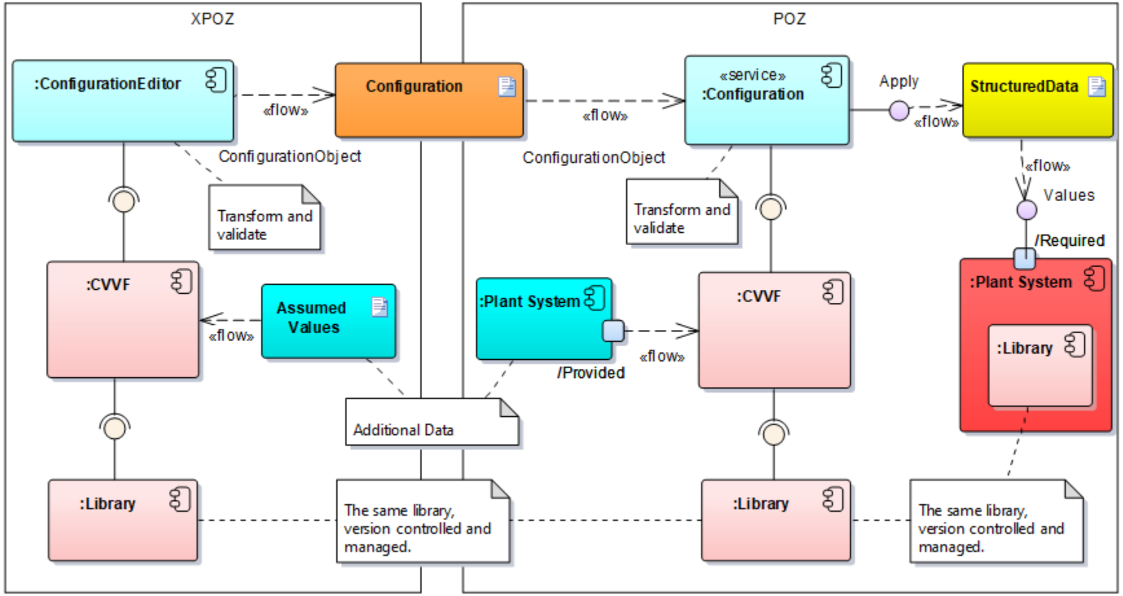
Architecture goals

- Operability
  - Run experiments according to a robust and well-established format
- Automation
  - Run autonomously with minimal human intervention; robust, routine and repetitive procedures are automated
  - Limit the need for human involvement to high added value tasks, e.g. definition of the experiment schedule, analysis of data, investigation and recovery of failures

Architecture goals

- Containment of complexity
  - Provide abstraction mechanisms, manageable information and control means
- Scalability
  - Support integration of new components with limited impact on existing ones
- Changeability
  - Adapt to changing requirements with affordable resources
- Robustness
  - Prevent propagation of failures through the control system

POZ and XPOZ



XPOZ
POZ
:ConfigurationEditor
«flow»
Configuration
«flow»
«service»
:Configuration
Apply
«flow»
StructuredData
ConfigurationObject
ConfigurationObject
Transform and validate
Transform and validate
«flow»
Values
/Required
:CVVF
Assumed Values
«flow»
:Plant System
«flow»
/Provided
:CVVF
:Plant System
:Library
Additional Data
:Library
The same library, version controlled and managed.
:Library
The same library, version controlled and managed.