ASAM | Association for standardisation of automation and measuring systems

Release Presentation

ASAM MCD-2 CERP/ P2014-02 / BS / V1.0

Calibration Expert System Rule and Product Model Format

Release Date: 2016/ 02 / 22

Motivation

- Check correctness of calibration data
- Automate checks
- Check characteristics* or hardware properties versus:
  - Fix values
  - Other characteristics
  - A2L properties
  - Hardware properties
- Exchangeable rule set:
  - Applied in different tools from different vendors
  - Applied at different points in calibration lifecycle
- Later: Generate parameter values via rules

* Characteristics = calibration parameters = labels = tuning variables = software constants

Marketing

- Improve quality of calibration data
- Early quality feedback in calibration lifecycle
- Improved communication between software development, control loop design, calibration, test, quality assurance
- Automate checks
- Automatic and functional calibration in later phases

Features of Standard

- Programming language OTX used as base
- Shared measurement and calibration extensions for OTX:
  - Read or write characteristic values
  - Read database (A2L) information
  - Compare characteristic values (ECU / physical representation, consider quantization, interpolate maps)
  - Low level functions with access to all properties (expert view)
  - Comfort functions with limited but simplified access to most important properties (simple view)
- CERP private extensions for OTX:
  - Access to feature (hardware properties)
  - Rule procedure to express checks

What are CERP Rules?

Rules

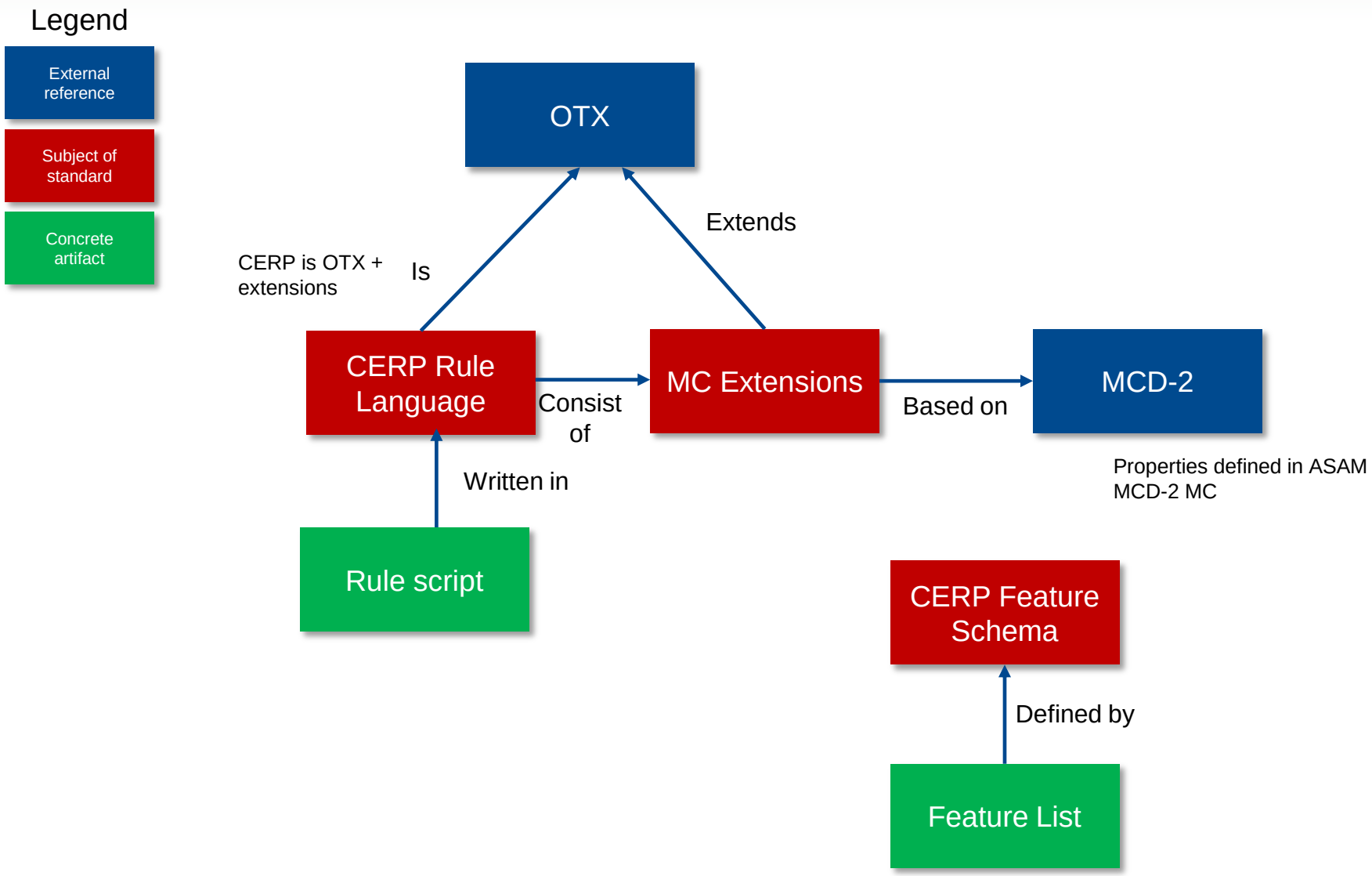
- Rules are small programs written in OTX
- Rules read calibration artifacts and calculate whether they are correct

OTX programming language

- OTX is a Turing-complete programming language
- OTX programs are exchanged in standardized XML notation
- OTX extension address calibration specific functionality

CERP extensions

- Core element of ASAM standard
- Extensions to read and write calibration artifacts


Legend
External reference
Subject of standard
Concrete artifact
OTX
Extends
CERP is OTX + extensions
Is
CERP Rule Language
Consist of
MC Extensions
Based on
MCD-2
Properties defined in ASAM MCD-2 MC
Written in
Rule script
CERP Feature Schema
Defined by
Feature List

Rule Lifecycle

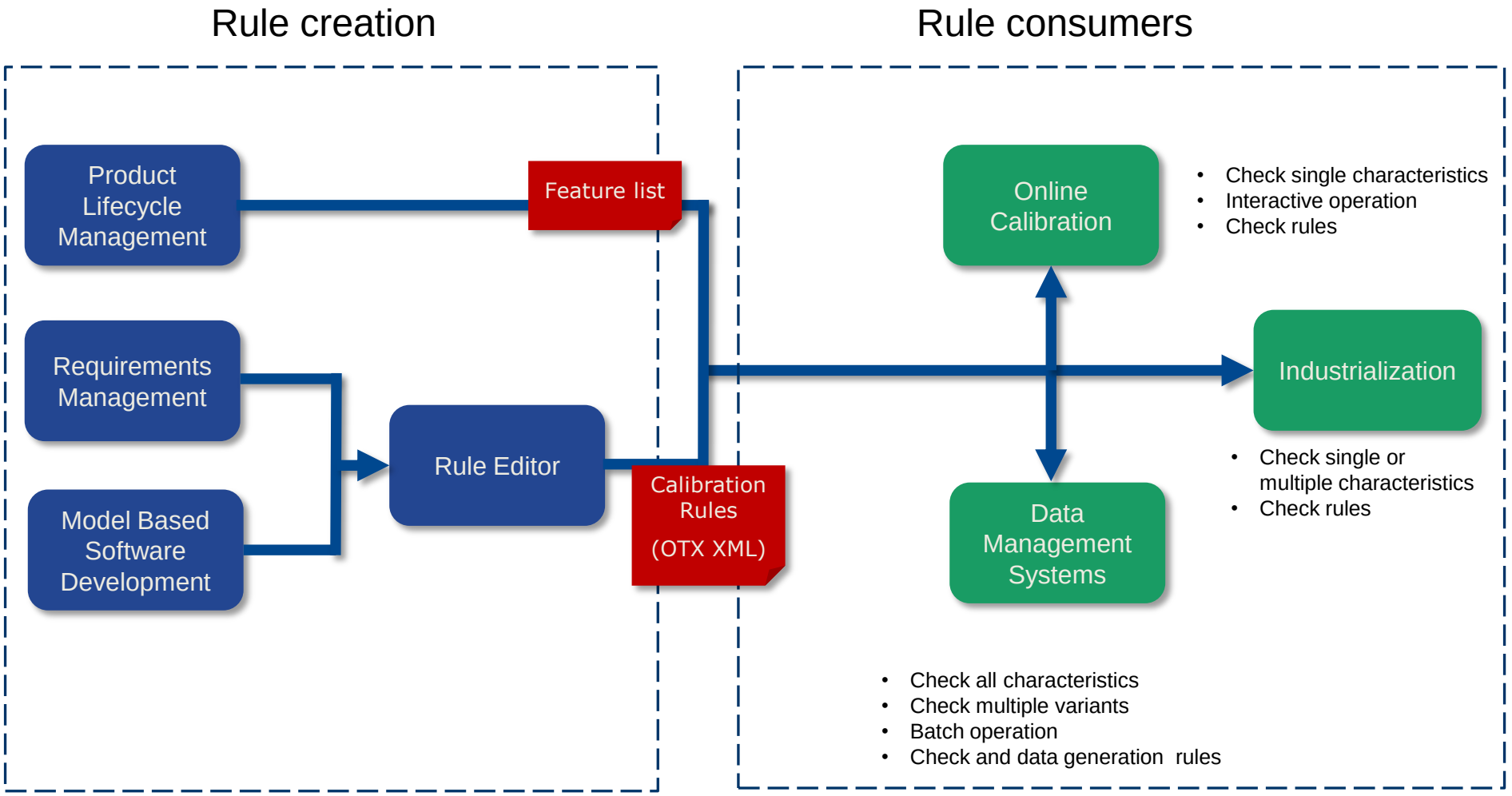
Rules are created in:

- Authoring tools
  - Rule created in rule editors
  - Feature list could be derived from product lifecycle management systems and bill of material

Rules are executed in:

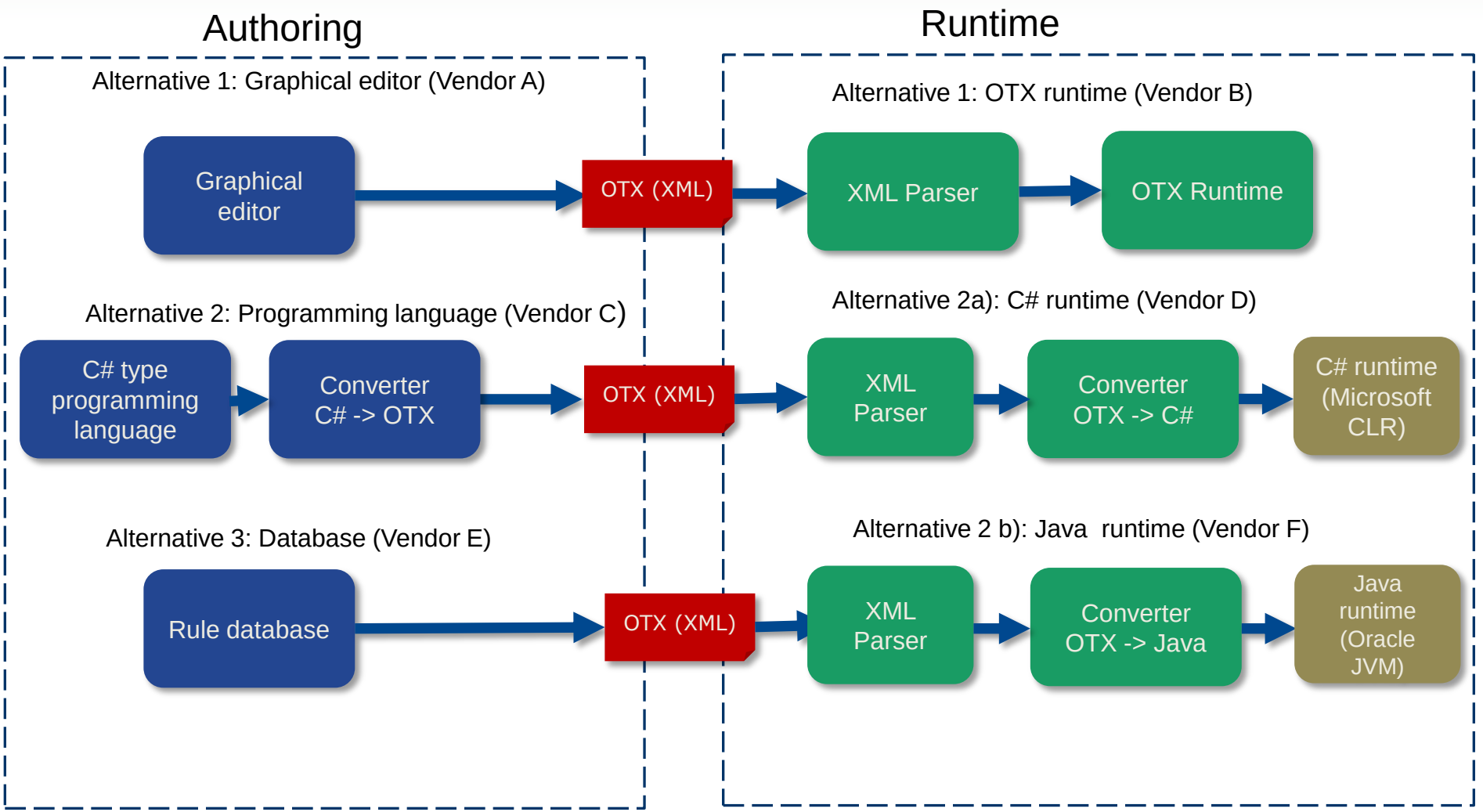
- Online calibration tools
  - Check individual parameter
  - Usage of rules: Component protection, early feedback if tuning questionable
- Data management systems
  - Check complete datasets
  - Usage of rules: Verify overall consistency, release preparations, later: generate default calibration for new calibration variants
- Industrialization
  - Final verification

Rule Lifecycle



Authoring and Runtime

- Exchange format is always standardized XML
  - OTX has only standardized XML notation
  - Not suitable to be read by humans
- Authoring tools
  - Design rules with graphical notation, domain specific language or general purpose language
  - Convert to standardized XML for exchange
- Runtime
  - Runtime executed rules in online and offline calibration tools
  - Possibility 1: Create OTX runtime and read XML directly into data model
  - Possibility 2: Use existing runtime (JVM, CLR) and convert OTX to C# / Java for execution



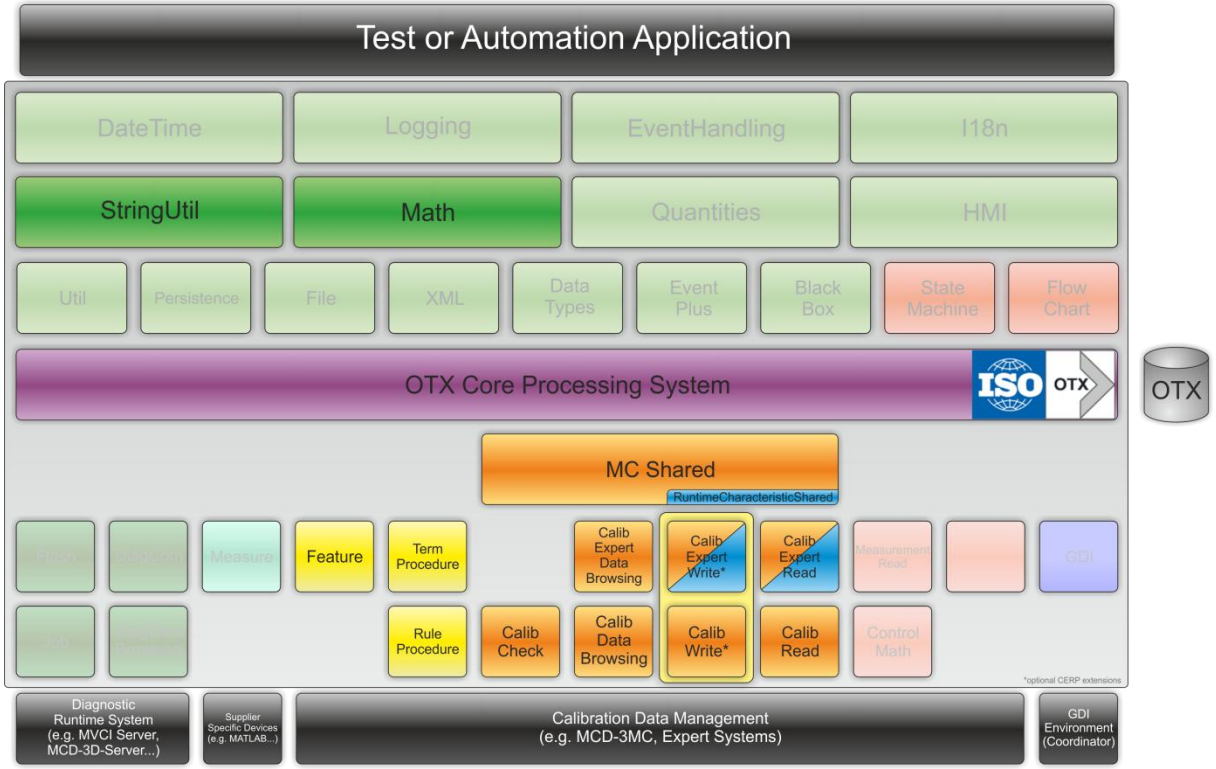
These are just example scenarios!
Tool vendors could use programming language and runtime of choice.

Available Functionality

- Read information from characteristic database (ASAM MCD-2 MC)
  - Layout of characteristic in ECU
  - Ranges, data types, max. dimension of arrays, units
  - Functions, variant coding, system constants
  - Simple but restricted “comfort” access; detailed “expert” access functions
- Read value information from calibration tool runtime model
  - Inspired by ASAM MCD-3 model (simplified and converted to procedures)
  - Characteristic values, value units
  - ECU and physical representation
  - Simple but restricted “comfort” access; detailed “expert” access functions
- Read feature values
  - Read feature values from exchange feature XML file
- Rule procedures
  - Special OTX procedure to realize rules
  - Rule procedure have predefined return type and supports precondition checks

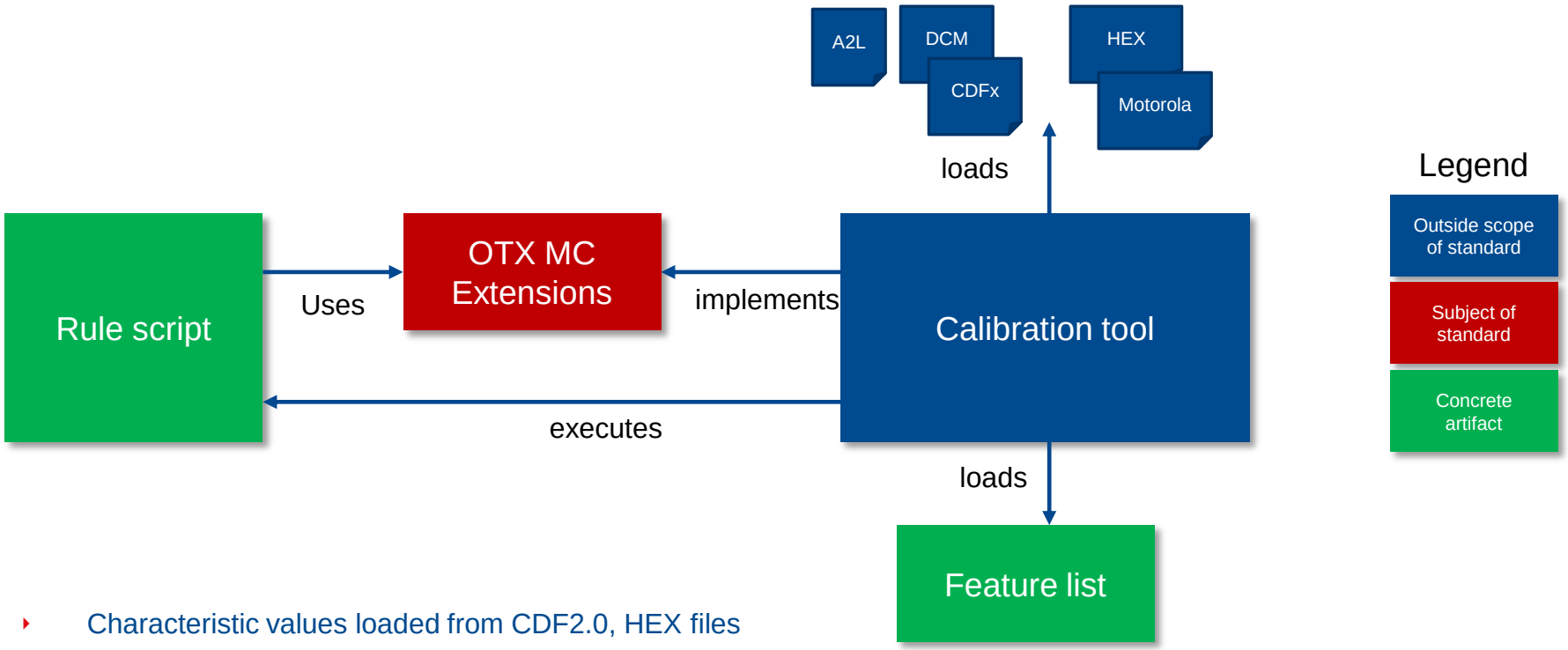
Available Functionality

OTX Extension Overview



- Orange: Shared extensions with CPX
- Yellow: CERP extensions
- StringUtil, Math: Shall be supported by CERP runtime

Artifacts in Context of Tool



- Characteristic values loaded from CDF2.0, HEX files
- Characteristic database loaded from A2L
- Feature list loaded from feature list XML
- Rule script executed in calibration tool context
- Rule scripts reads values, database and feature information from calibration tool model via standardized interface

Example

Non-standardized pseudo code

```
private MyRule (out RuleResult result, in Integer anInputVar)
{
  // Get ID of characteristic
  CharacteristicIdentifier idA = CreateCharacteristicIdentifier(Name  = "ParamA");

  // The reference value
  PhysicalValue refValue = CreatePhysicalValueByFloat (FloatValue = 17.3);

  string injector = GetFeature("Injector");

  // The actual comparison
  CompareRelation compRes = CompareCharacteristicPhysicalValue (characteristicIdentifier  = idA, compareValue = refValue);


  if (injector  == "TypA" && compRes == e_CR_Greater)
  {
    result.RuleState = Successful;

  }
  else if (injector == "TypB" && compRes == eCR_Less)
  {
    result.RuleState = Successful;
  }
  else
  {
    result.RuleState = Failed;
    result.RuleSeverity = Error;
    result.RuleMessage = "ParamA not in valid range for given injector type"
  }
}
```

Rule procedure

Unique identifier of characteristic

Reference value

Read feature

Current physical value of "ParamA"

Set value of ParamA in relation with feature

Error result

Deliverables

- CalibExtension Interface Definition
  - Shared with CPX
  - Describes 9 OTX extensions for calibration access
- CERP Reference Manual
  - Describes 3 OTX extensions only used by CERP
- User Guide
  - Intention and design principles of standard
  - Implementation guide for tool vendors and rule editors
- XSDs of OTX extensions
  - XSDs contain OTX extensions
  - Can be used to validate exchanged files
  - One XSD that contains feature model
- UML model
  - UML model is master for reference guide and XSDs
- XML examples
  - Simple check rule examples in OTX exchange format

Compatibility

ASAM Standards

- Database access for properties described in ASAM MCD-2 MC (ASAP2) V1.7

ISO

- CERP is a set of extensions according to OTX (ISO 13209 1st edition)
- OTX base language to express variables, sequences, loops, conditions, type system, runtime model