OSLC-based Integration
Traceability Architecture <> Code

smartfacts

Our Experience: MID has implemented various OSLC Connectors
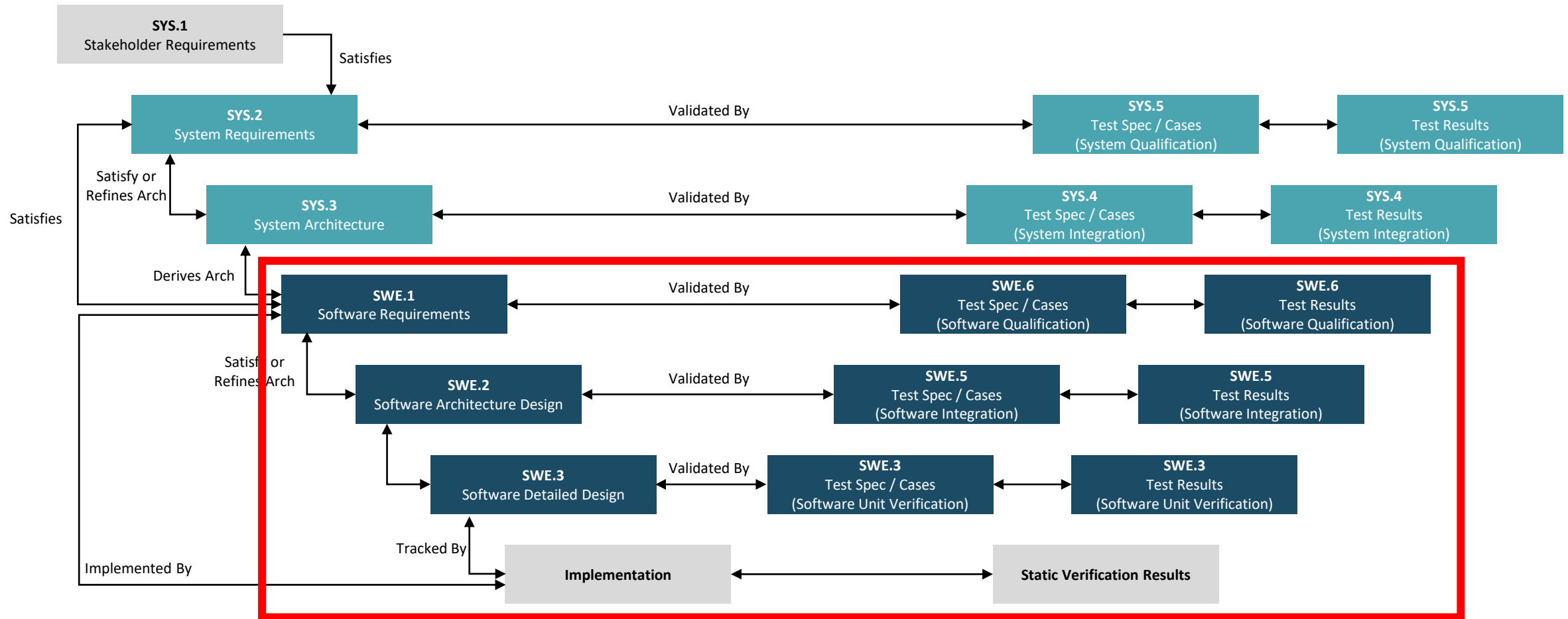
OSLC Connectors

- codeBeamer (RM, QM)

- Jama Connect (RM)

- Jira (CCM)

- Confluence (Consumer only)

- Smartfacts (Modeling Tool Connector for Cameo, Enterprise Architect, Innovator, MagicDraw, Simulink, Rhapsody)

Frameworks

- #genOSLC (OSLC provider and consumer programming framework)

- Tool Independent Link Index

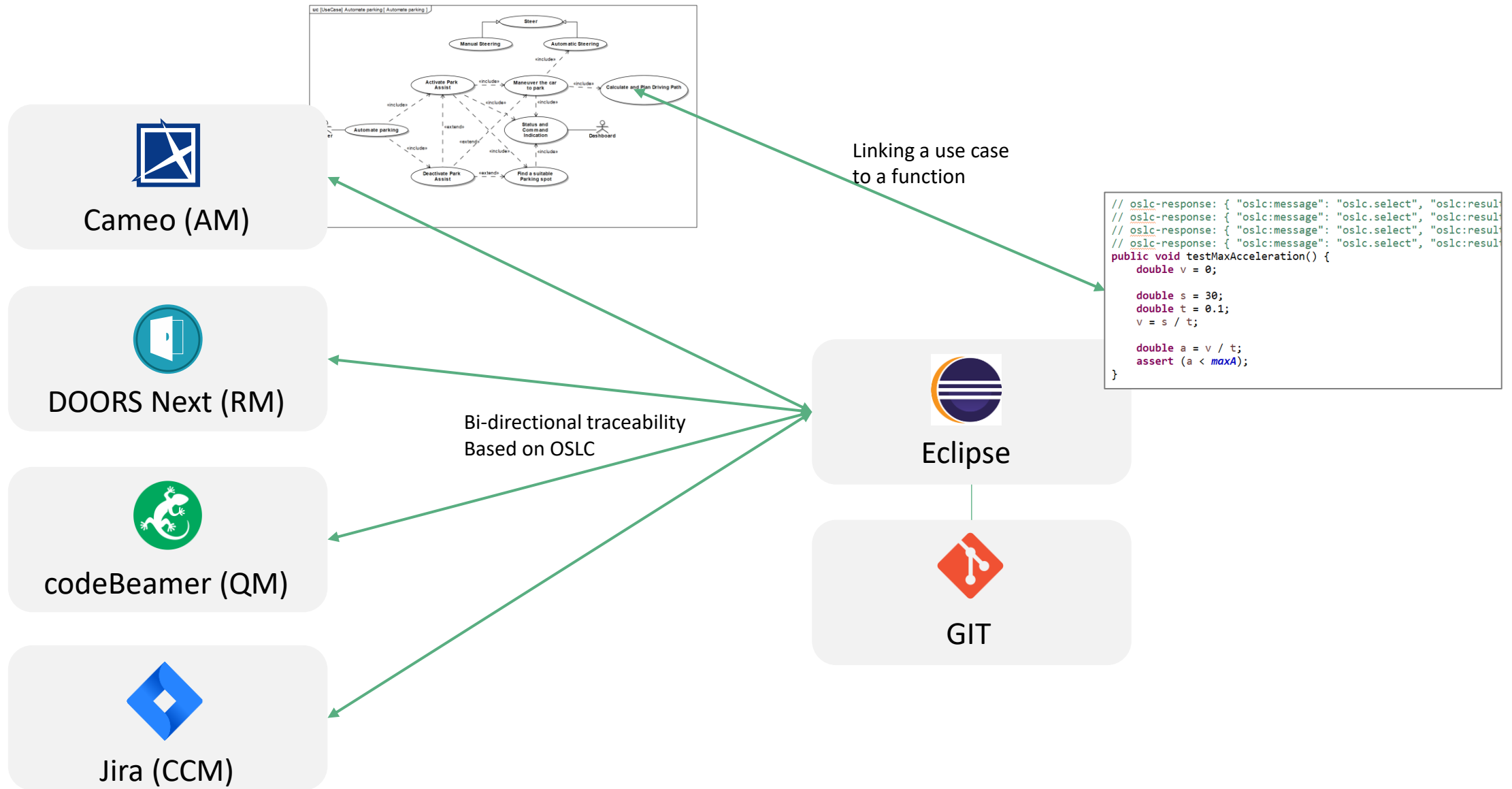# Realizing a digital thread information model with OSLC vocabulary
Example: Automotive Spice (ASPICE)
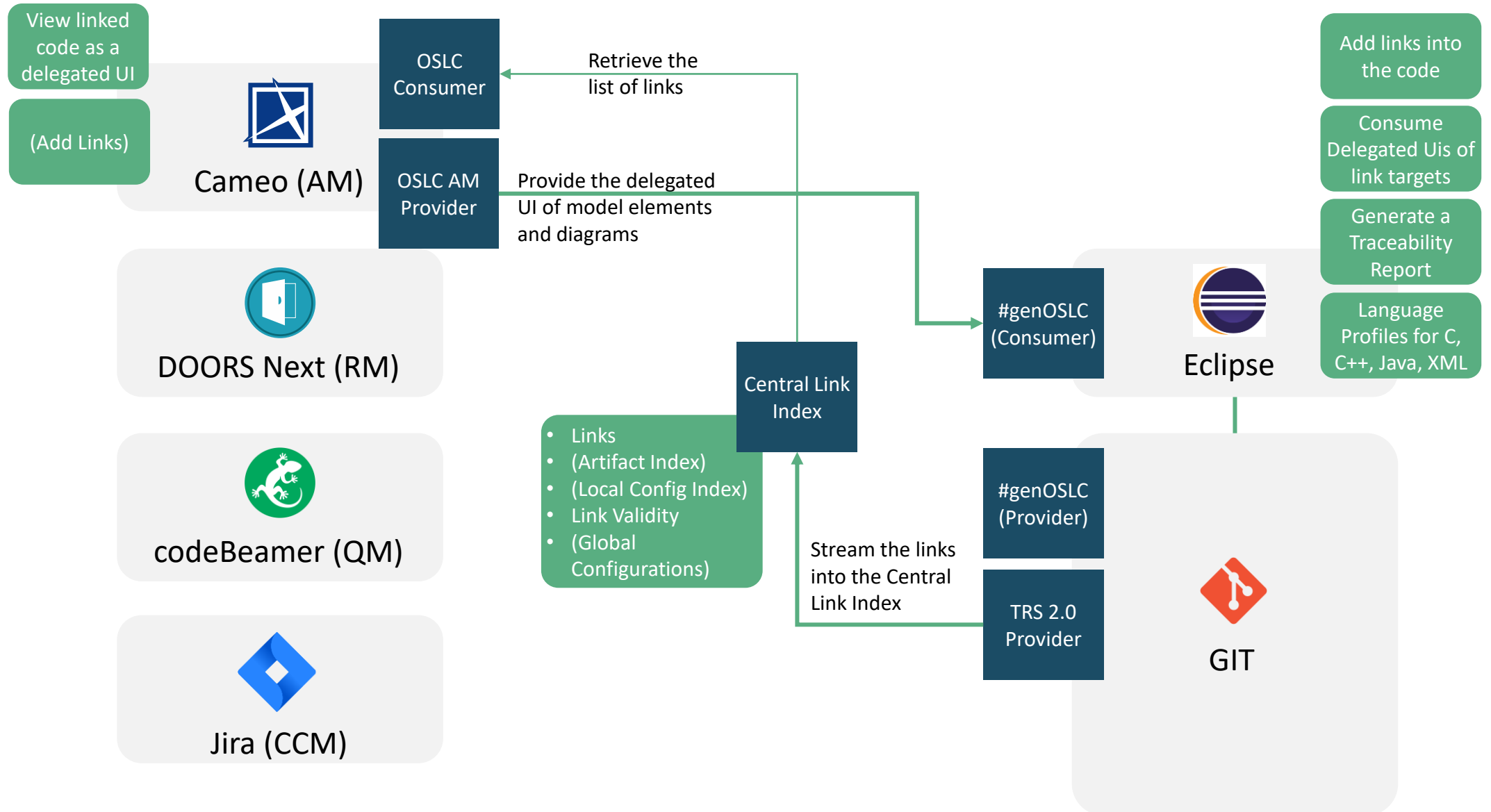
Motivation

- Software Code is an important part of the digital thread

- Regulatory standards like ISO 26262 demand full traceability from the requirement into the implementation details of the code

- We use the OSLC standard, to establish traces from code to other OSLC domains (AM, CCM, QM, RM)

- We aim to support the use of various IDEs and various programming languages

Challenge: Linking OSLC Artifacts to Code

Linking a use case to a function

Bi-directional traceability
Based on OSLC

Cameo (AM)

DOORS Next (RM)

codeBeamer (QM)

Jira (CCM)

Eclipse

GIT

```
// oslc-response: { "oslc:message": "oslc.select", "oslc:resul
// oslc-response: { "oslc:message": "oslc.select", "oslc:resul
// oslc-response: { "oslc:message": "oslc.select", "oslc:resul
// oslc-response: { "oslc:message": "oslc.select", "oslc:resul
public void testMaxAcceleration() {
    double v = 0;

    double s = 30;
    double t = 0.1;
    v = s / t;

    double a = v / t;
    assert (a < maxA);
}
```

Solution:

View linked code as a delegated UI

(Add Links)

Cameo (AM)

DOORS Next (RM)

codeBeamer (QM)

Jira (CCM)

OSLC Consumer

OSLC AM Provider

Retrieve the list of links

Provide the delegated UI of model elements and diagrams

Central Link Index

- Links
- (Artifact Index)
- (Local Config Index)
- Link Validity
- (Global Configurations)

Stream the links into the Central Link Index

#genOSLC (Consumer)

#genOSLC (Provider)

TRS 2.0 Provider

Eclipse

GIT

Add links into the code

Consume Delegated Uis of link targets

Generate a Traceability Report
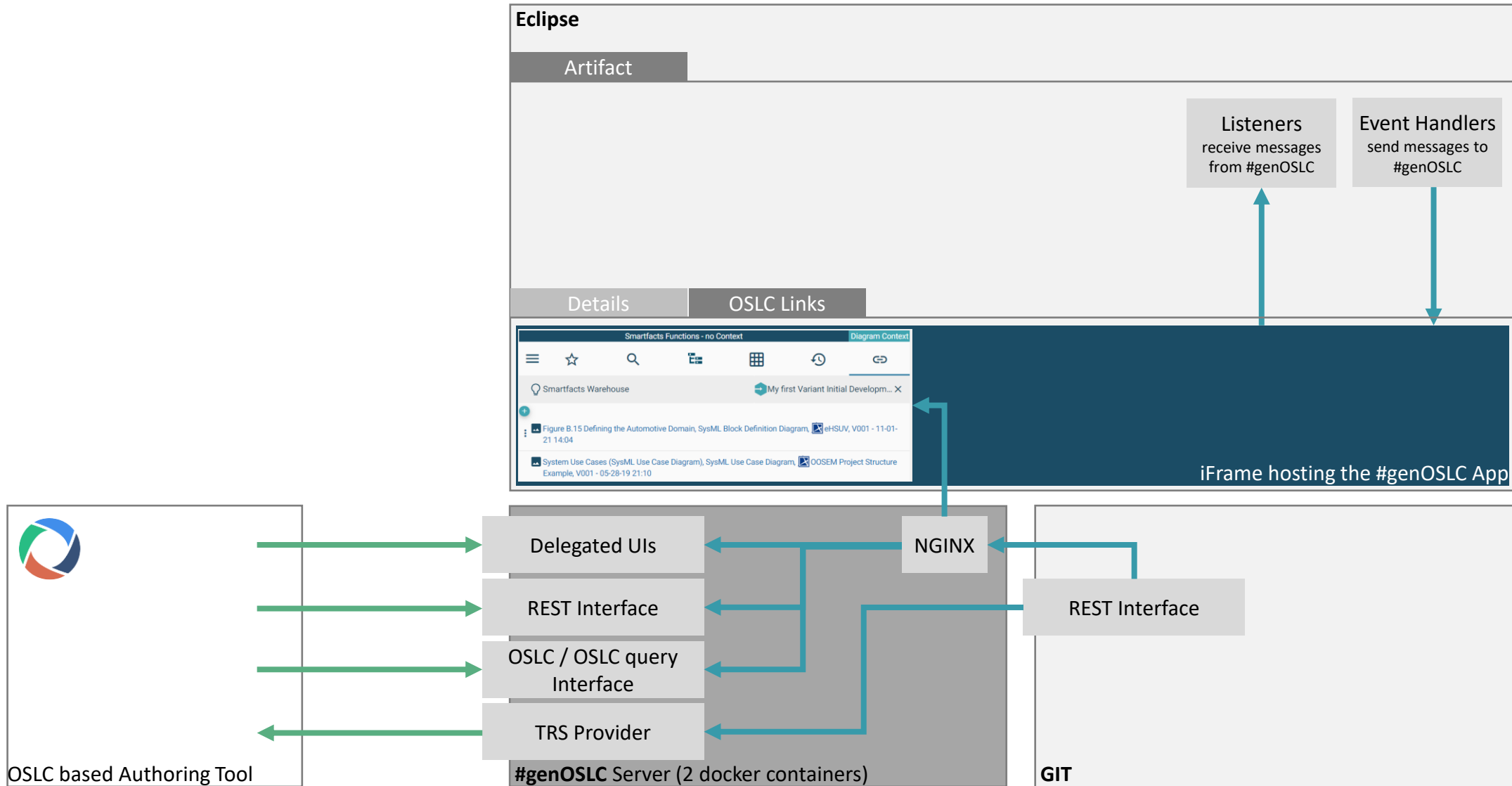
Language Profiles for C, C++, Java, XML

Demo

Architecture

The #genOSLC Programming Framework allows to add an OSLC consumer and a provider to a tool with only a few days of effort

#genOSLC Provides Large Parts of the Framework

Framework functionality

Configuration
- Manage rootservices
- Manage link types
- Manage friends
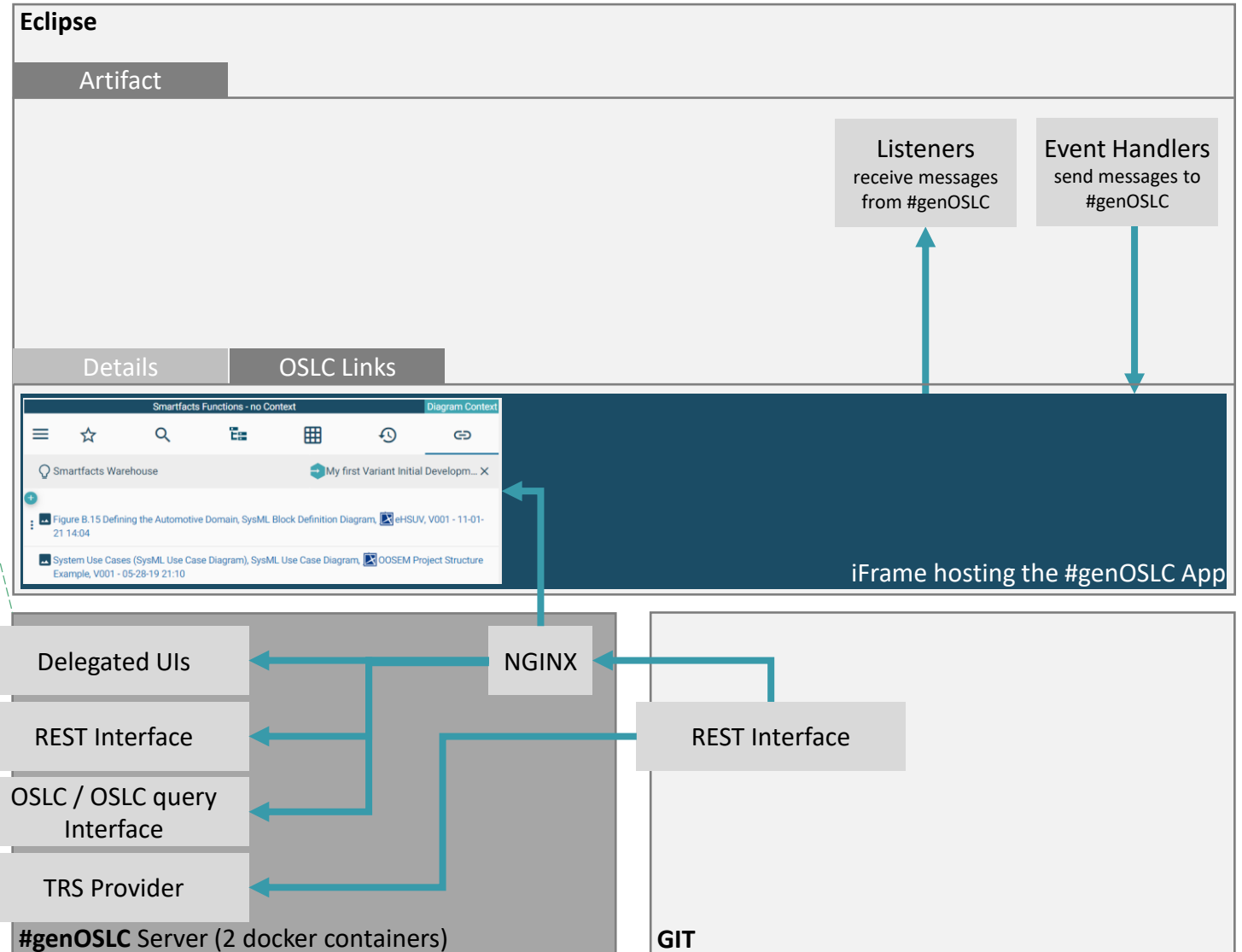- Manage consumer white-listing

Authentication mechanisms
- oAuth1a
- oAuth2
- OIDC
- SSO

Functions available via REST
- Select a project by instance
- Select a „Component" by a project

OSCL Functionality
- List the links on an item
- Set a Jazz GC as global context and filter the links by GC
- Add and Remove a link
- Show a delegated UI of an item
- Allow to select an item via a delegated search & select UI
- Allow to navigate to an item
- Handle Link Validity
- Provide TRS streams for links and items

Eclipse

Artifact

Listeners
receive messages
from #genOSLC

Event Handlers
send messages to
#genOSLC

Details

OSLC Links

Smartfacts Functions - no Context Diagram Context

Smartfacts Warehouse My first Variant Initial Developm... ✕

Figure B.15 Defining the Automotive Domain, SysML Block Definition Diagram, eHSUV, V001 - 11-01-21 14:04

System Use Cases (SysML Use Case Diagram), SysML Use Case Diagram, OOSEM Project Structure Example, V001 - 05-28-19 21:10

iFrame hosting the #genOSLC App

Delegated UIs

REST Interface

OSLC / OSLC query Interface

TRS Provider

NGINX

REST Interface

#genOSLC Server (2 docker containers)

GIT

Next Steps

- Finalize the Link Index for GIT Links

- Add delegated code UIs

- Allow to add links from the link partner (AM, CCM, QM, RM)