OSLC Update

INCOSE IW 2022

Axel Reichwein
January 30, 2022

Example Relationships in Engineering

Benefits of Linking Engineering Data

- Traceability
- Automation of workflows
- Impact Analysis/Root cause analysis
- Change Management
- Improved holistic system overview
- Configuration management
- Full-text search
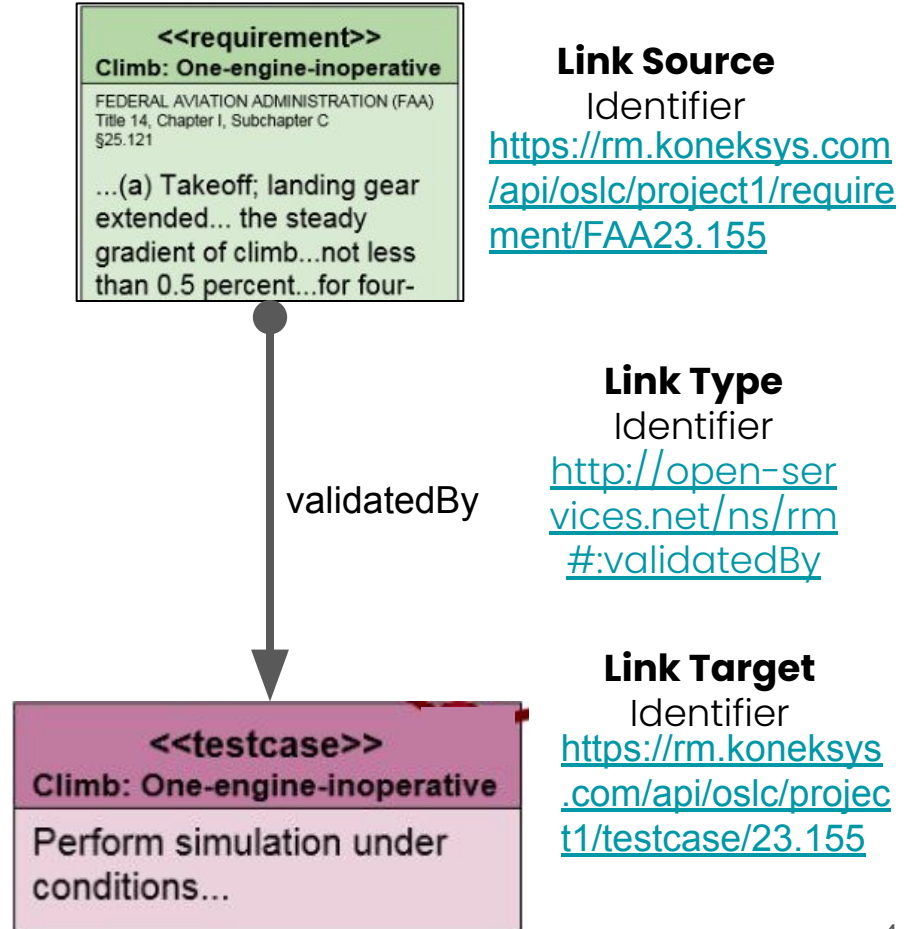- Discovering new knowledge
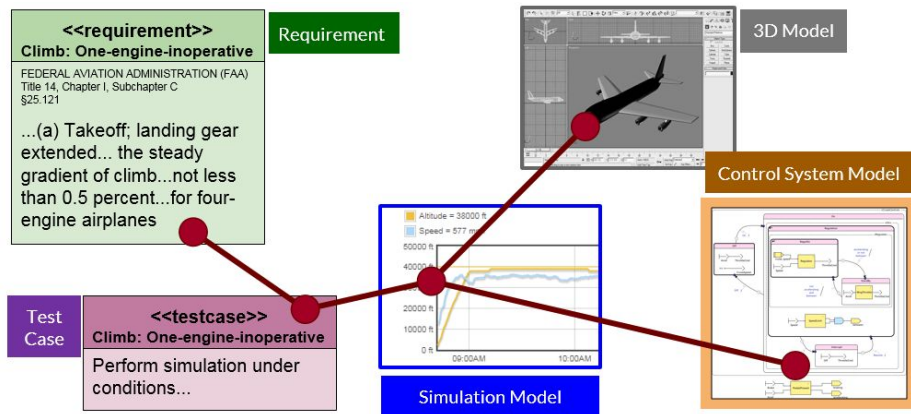
Linked Engineering Data is more meaningful than siloed Engineering Data

What is a Link?

Each linked artifact needs to have a unique global/universal identifier (e.g. URL)

Each link has a type so that we understand the meaning of the relationship

Each link has a direction from source to target

<<requirement>>
Climb: One-engine-inoperative

FEDERAL AVIATION ADMINISTRATION (FAA)
Title 14, Chapter I, Subchapter C
§25.121

...(a) Takeoff; landing gear extended... the steady gradient of climb...not less than 0.5 percent...for four-

Link Source
Identifier
https://rm.koneksys.com/api/oslc/project1/requirement/FAA23.155

validatedBy

Link Type
Identifier
http://open-services.net/ns/rm#:validatedBy

<<testcase>>
Climb: One-engine-inoperative

Perform simulation under conditions...

Link Target
Identifier
https://rm.koneksys.com/api/oslc/project1/testcase/23.155

Koneksys

Traditional Centralized Link Creation Strategy

Certain applications describe many different aspects of a system, so they provide support for data integration (PLM, ALM, MBSE)

Many useful capabilities provided by integration apps: visualization, analysis, etc.

**Should ALL the links be created in such a central application?**

"Central" Data Store (e.g. PLM/ALM/MBSE)
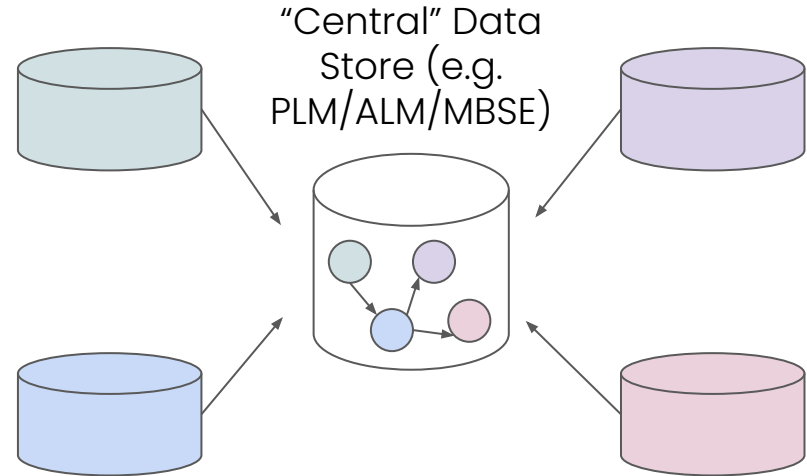
Until now, links are created within a "central" application

Koneksys

Traditional Centralized Link Creation Strategy

Problems:

- **Not user-friendly** for engineers to have to switch to a different application
- **Importing data not always possible** due to vendor lock-in (lack of integrations)
- **Schema often not flexible enough** to describe new link types
- **No single database suitable for all purposes!** No one size fits all! Example: data lake for IoT data vs data warehouse for "traditional" PLM data

"Central" Data Store (e.g. PLM/ALM/MBSE)
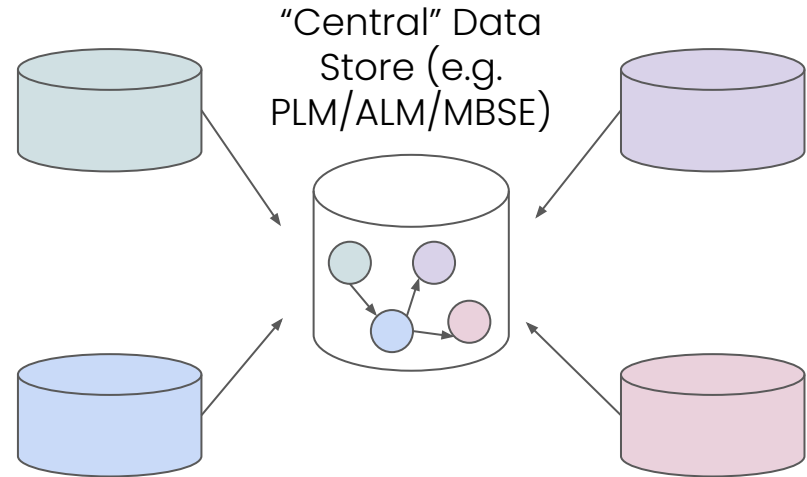
Centralized Link Creation does NOT SCALE

Koneksys

Distributed Link Creation Strategy

Links are created from **within ANY application.** Advantages:

- **User-friendly**: Links created only as needed by engineers within their familiar applications
- **Central registry for linking rules** to ensure that only meaningful links of a certain type are used between artifacts of a certain type
- **No bottleneck**

Central Registry for Linking Rules

All applications are equal from a linking perspective

Koneksys
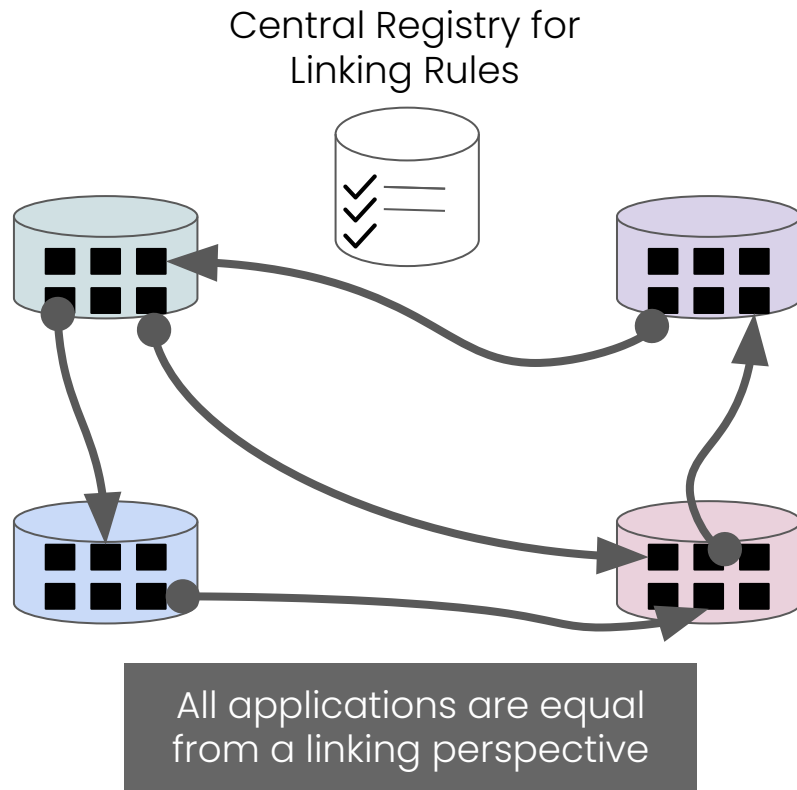
Link Creation Decoupled from Link Management

Links can be persisted in multiple storage solutions at time of creation

Data and Links are accessible to API clients and can be collected and saved in another storage solution for
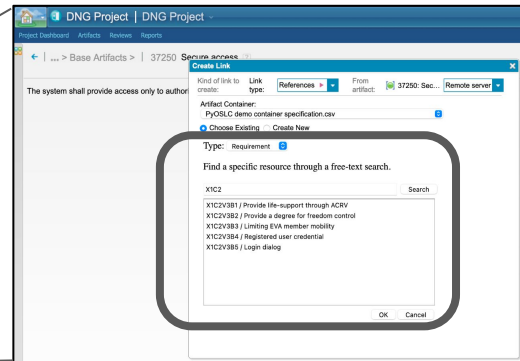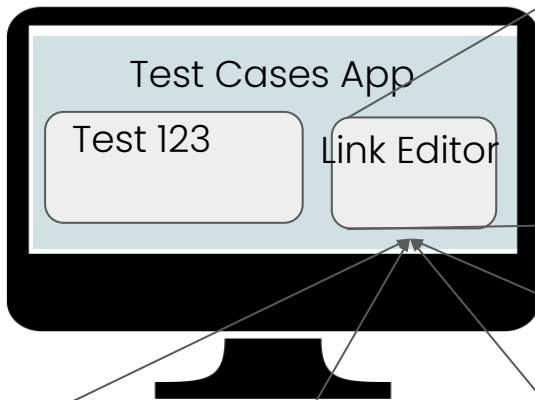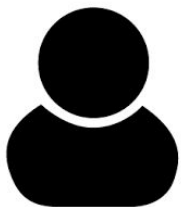
- Query purposes
- Visualization
- Analysis
- Triggering automated workflows
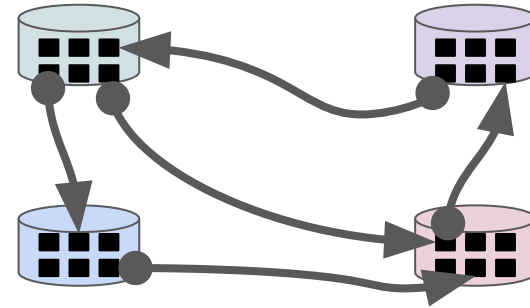
Link Management not constrained to the capabilities of a single application
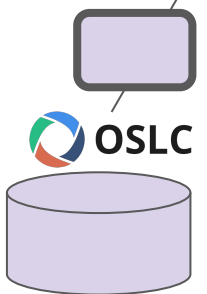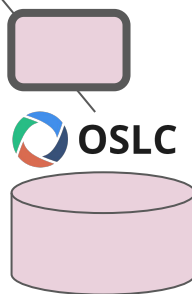
Data Lake   PLM   SysML   Graph database

Koneksys
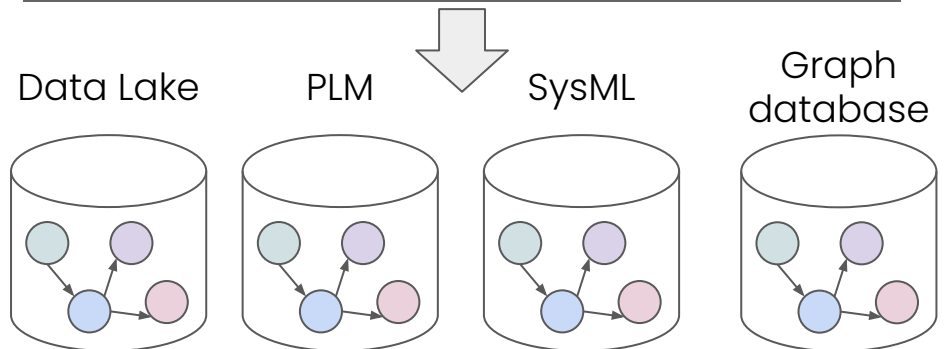
Application-specific Search Dialogs

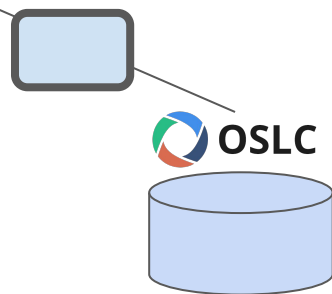Application-specific search dialogs exposed by OSLC APIs

Test Cases App

Test 123

Link Editor

Requirements

Simulation Models

3D Models/PLM

SysML Models

Koneksys

9

Discovery of the Search Dialog URL

Test Cases App

Test 123

Link Editor
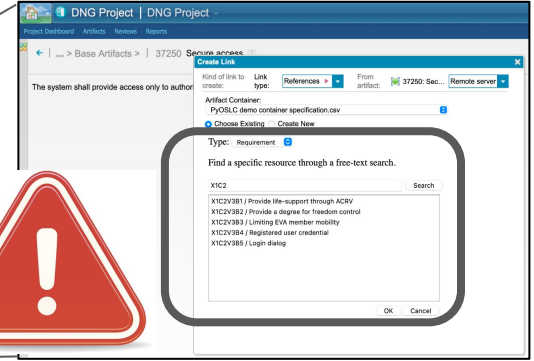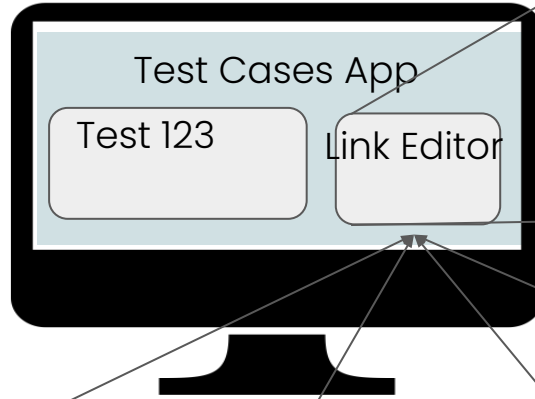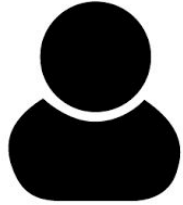
Each software application exposes its OSLC search dialog at a different URL

https://urlX

https://urlY

https://urlZ
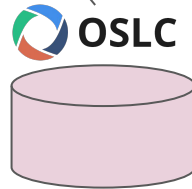
OSLC

OSLC

OSLC

OSLC

Requirements

Simulation Models

3D Models/PLM

SysML Models

Koneksys

OSLC API Adopting OWA decoupled from Linking Rules adopting CWA

Before HTTP Request

HTTP PUT Request

API Client

New links to "unexpected" artifacts

OSLC API

Data Source

Central Registry for **Linking Rules**

New links valid?

After HTTP Request

200 OK

API Client

OSLC API

Data Source

OSLC API and Linking rules are decoupled to support the reuse of an OSLC API
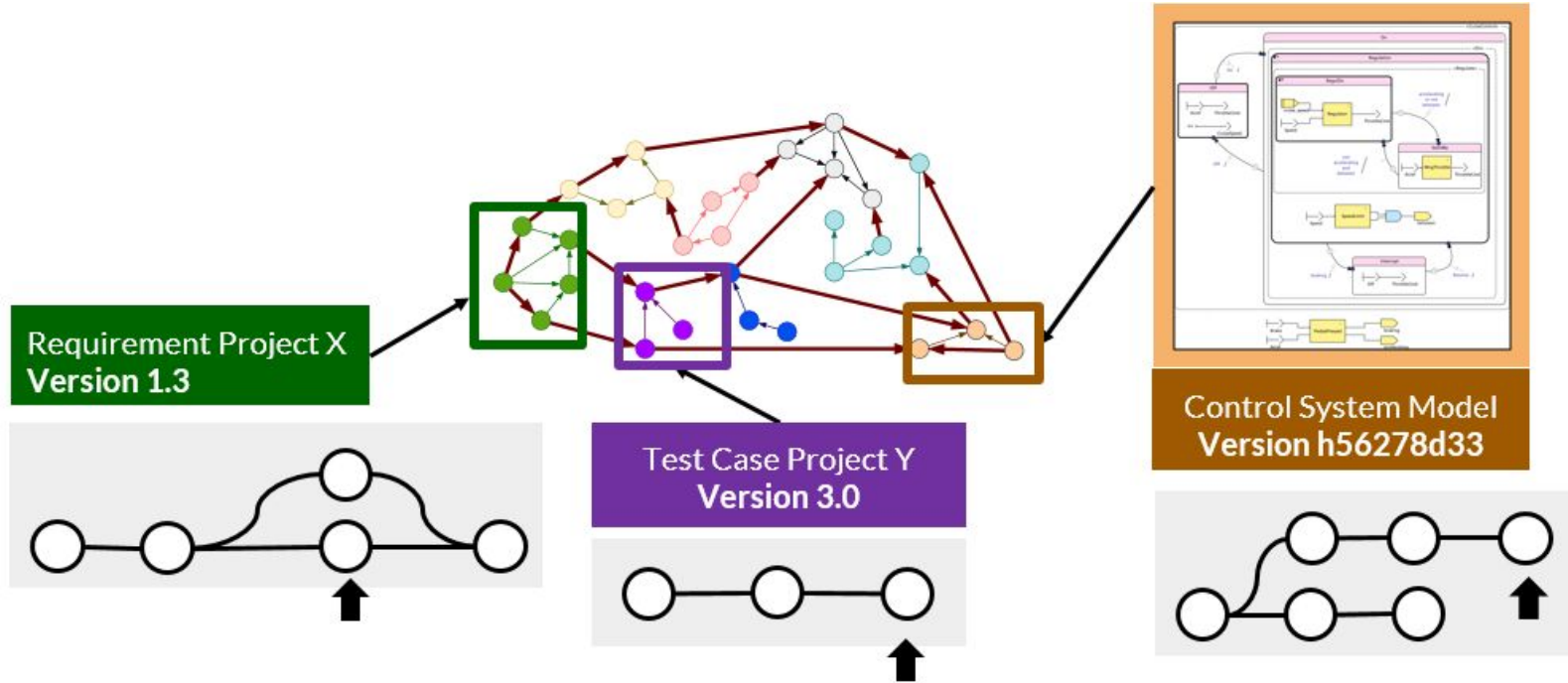
Koneksys
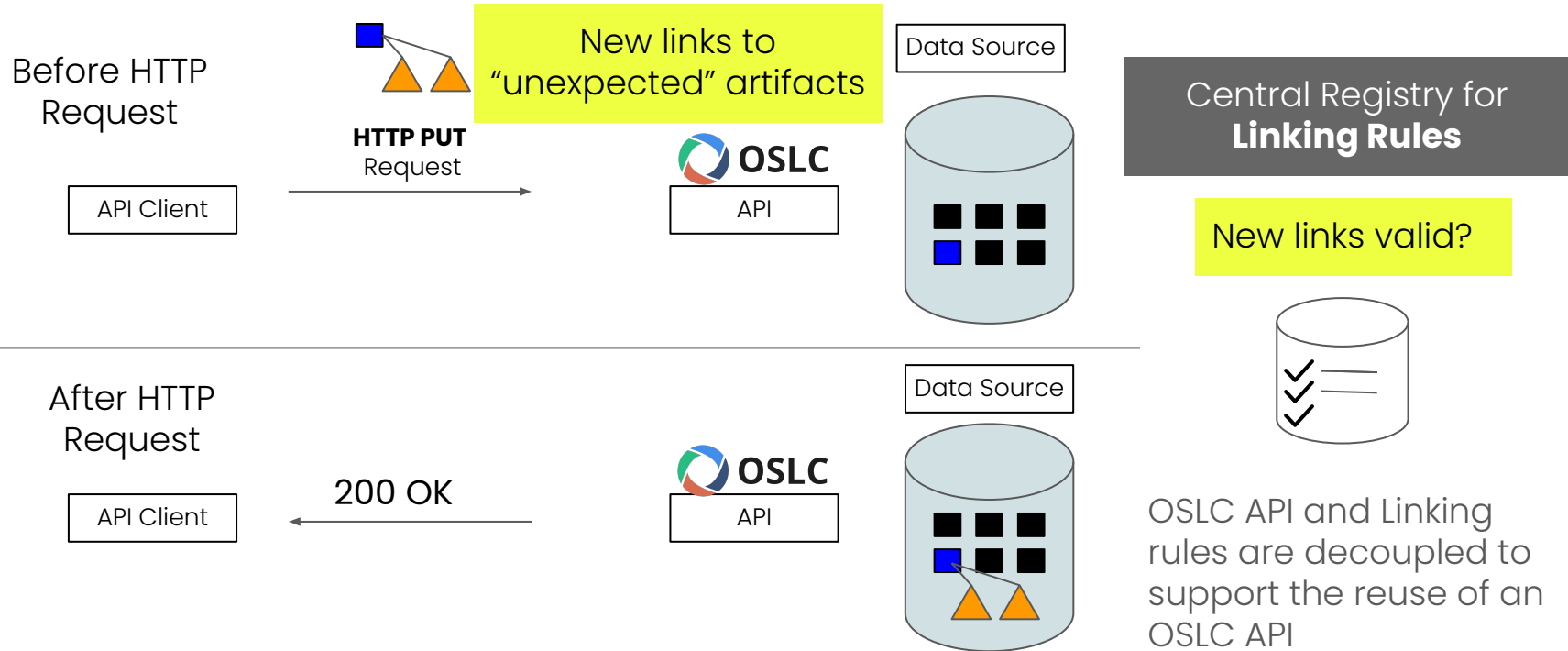
Configurations Exposed by OSLC APIs

Client can discover available configurations from the different OSLC APIs

Link Store

API Client

Configurations

Configurations

API #1

API #2

Data

Data

Koneksys
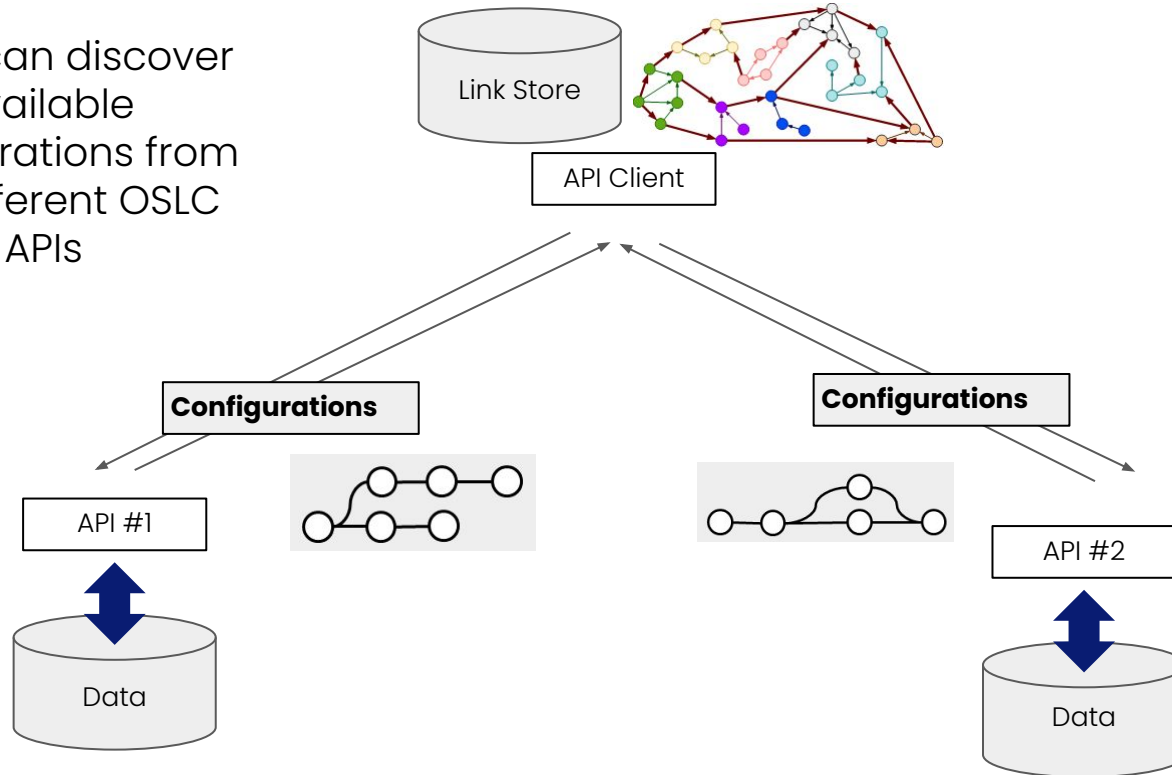
Configuration-Managed Linked Data

Requirement Project X
Version 1.3

Test Case Project Y
Version 3.0

Control System Model
Version h56278d33

Koneksys

Since January 2019

OASIS OSLC Open Project established in June 2019

OSLC standards

Reference Implementations

2 additional OSLCFest virtual conferences

Koneksys

Overview of OSLC Specifications

Image from Andrew Berezowksy, KTH OSLCFest 2021,
https://www.youtube.com/watch?v=IJ7FunqW0rE&list=PLpqpu1CS6Rj4JHr8YkcLbWL6j8lw7w8pr&index=17
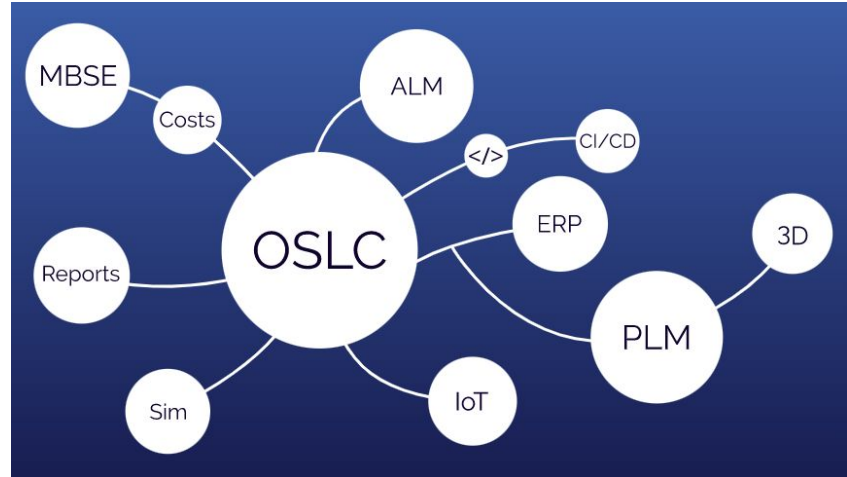
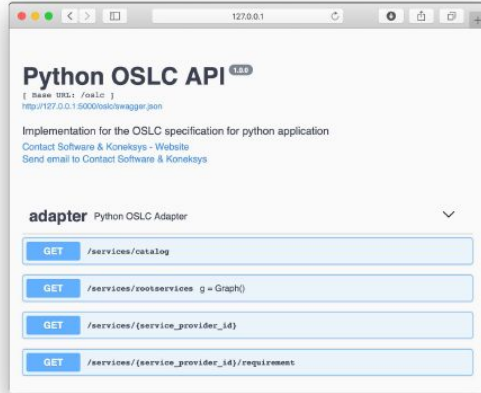Koneksys

OSLCFest

End edition in May 2020

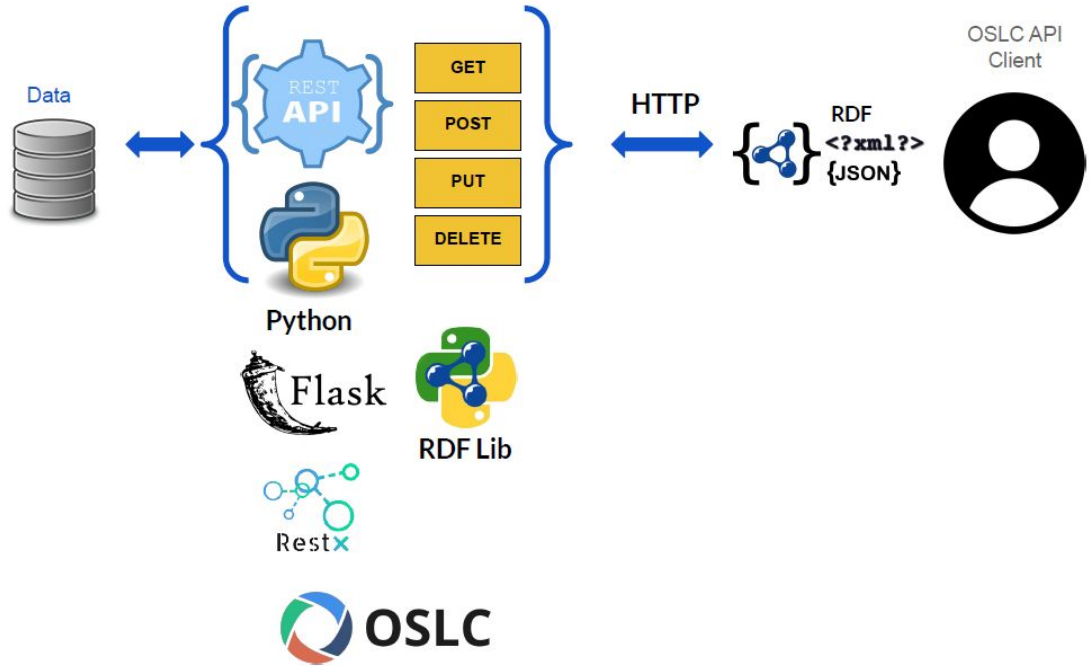3rd edition in November 2021

Demos of many vendors

Covering OSLC and related topics (e.g. DevOPs, IoT, category theory)

OSLC SDK in Python (PyOSLC)

https://github.com/cslab/pyoslc

Koneksys

Future Topics

SysML v2 API and OSLC

Push notifications for TRS (e.g. Kafka based)

Link status

Meetings

Weekly Call on Thursdays at 10AM EST

 Join meeting: https://meet.jit.si/oslc-op#config.startWithVideoMuted=true

The meeting minutes are edited in https://hackmd.io/@driib/oslc-op-minutes/edit. Previous minutes can be found under https://github.com/oslc-op/oslc-admin/tree/master/minutes/2019

Meeting ID: 2979764690#

One tap audio Dial In: +15124022718,,,,2979764690# (US) or +498938038719,,,,2979764690# (Germany) Looking for a different dial in number? Please see: https://meet.jit.si/static/dialInInfo.html?room=oslc-op

Monthly OSLC impact call

Koneksys

Thanks and get in touch!
axel.reichwein@koneksys.com

Koneksys