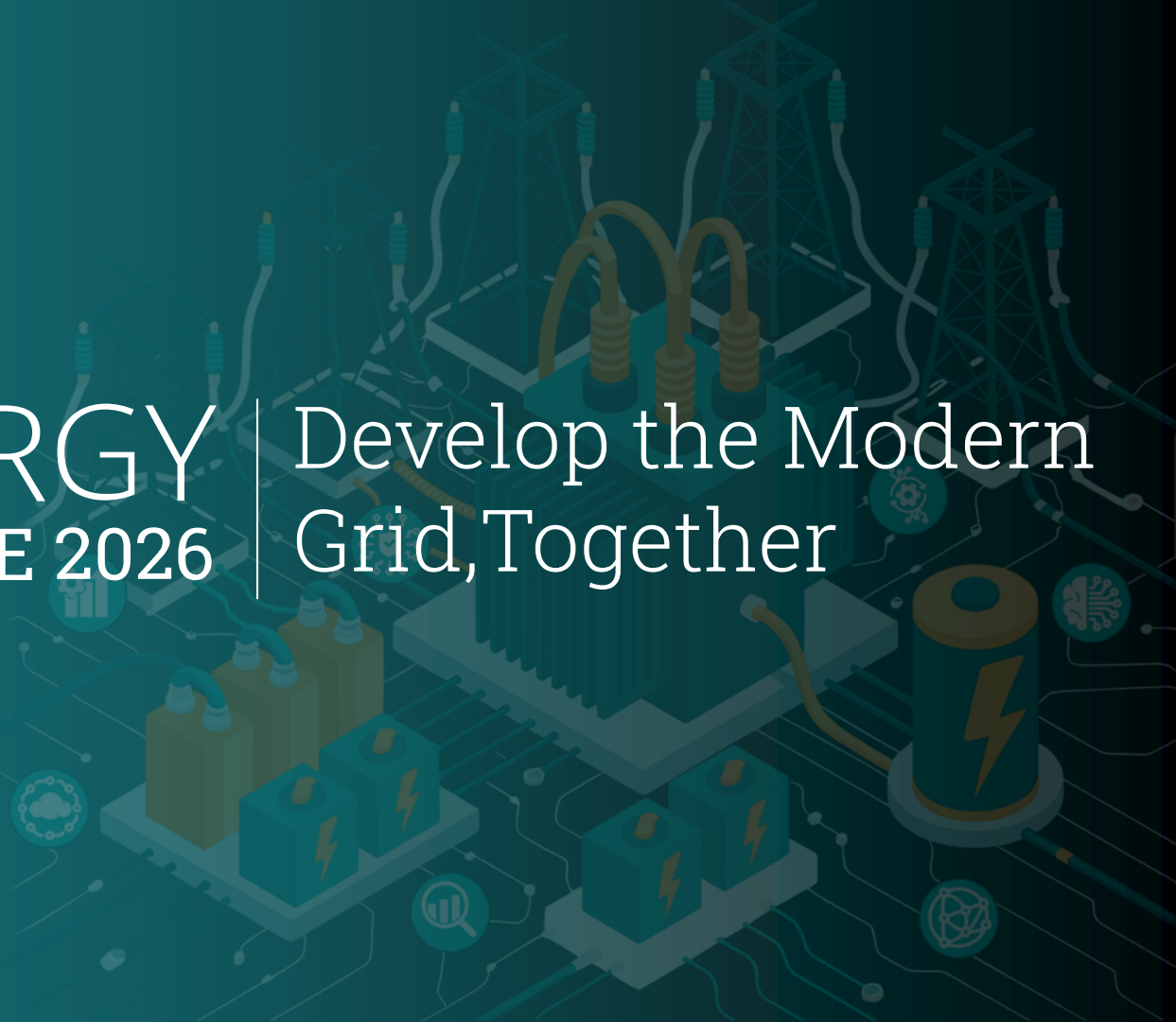
LF ENERGY SUMMIT EUROPE 2026

Develop the Modern Grid, Together



Agenda

1	Building a substation specification with CoMPAS	Sander | 30 min
2	Designed to Extend: How Developers shape CoMPAS	Lorenz/Stefan | 20 min
3	Driving CoMPAS Adoption with Better UX	Philomena | 10 min
4	Extend CoMPAS in Minutes: Build Your First Plugin (Live Coding)	Lorenz | 10 min
5	Quality You Can Trust: Raising the Bar for SCL Engineering	Stefan | 10 min
6	FAQ - Questions from the audience	All presenters

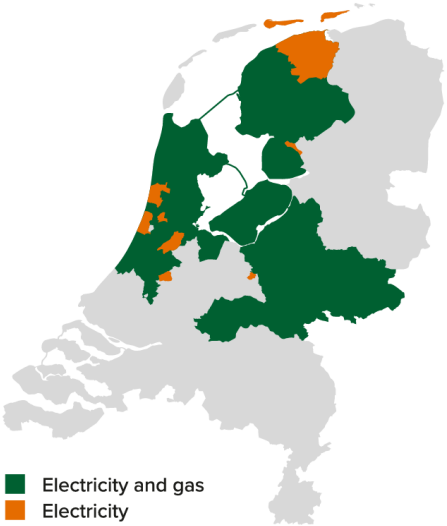
Building an substation specification with CoMPAS



LF ENERGY SUMMIT EUROPE 2026 | Develop the Modern Grid, Together

About Alliander & myself

DSO in our service area





Product owner virtual stations
CoMPAS TSC chair

Current state of Substation automation @Alliander

- First generation centralized control and protection considered legacy
- No process bus
- IED's per bay approach
- Multi-year contracts with vendors
- In-house engineering and maintenance (**Qirion**)
- Modulair building blocks (primary/secondary)
- (mostly) button up engineering



Growing complexity in substation automation

Added in the last years:

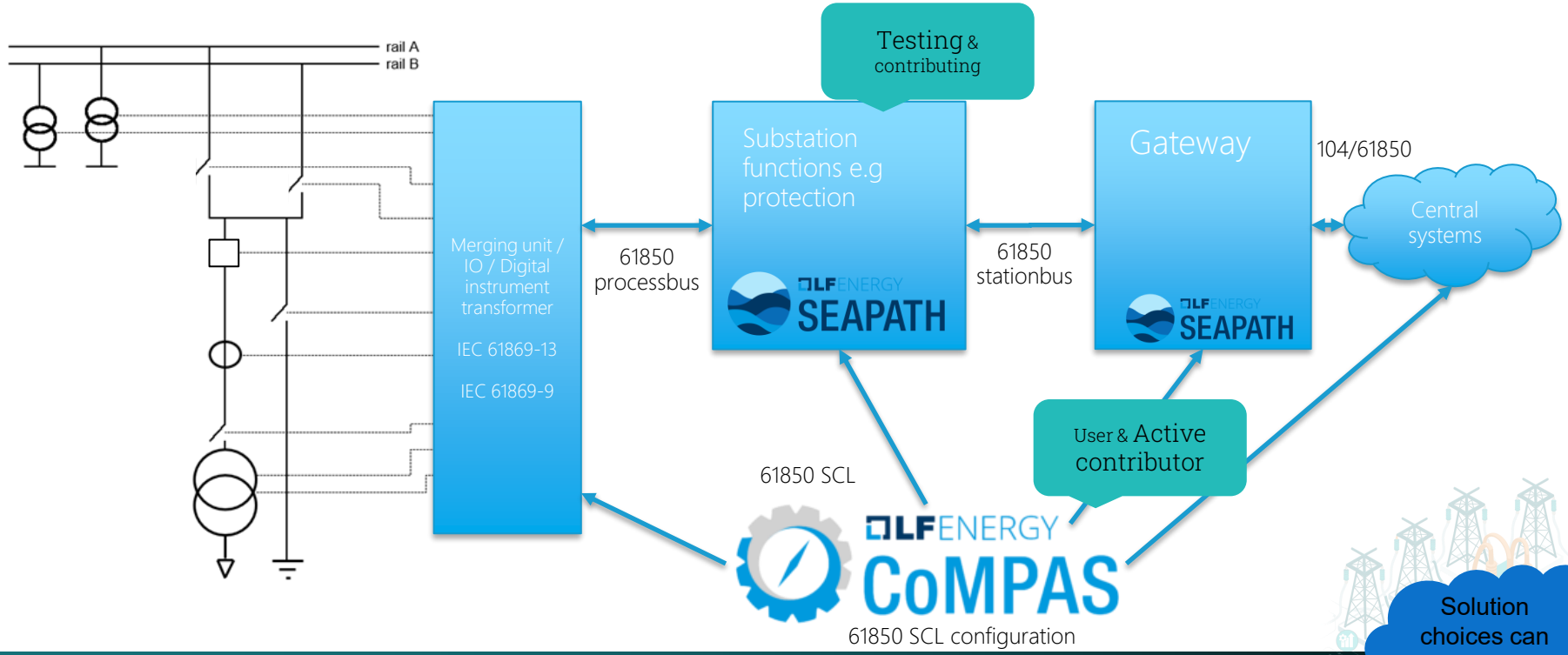
- Compound based control for transformers
- Congestion protection (Load and generation)
- Phase angle measurements
- Frequency protection

Possible future needs:

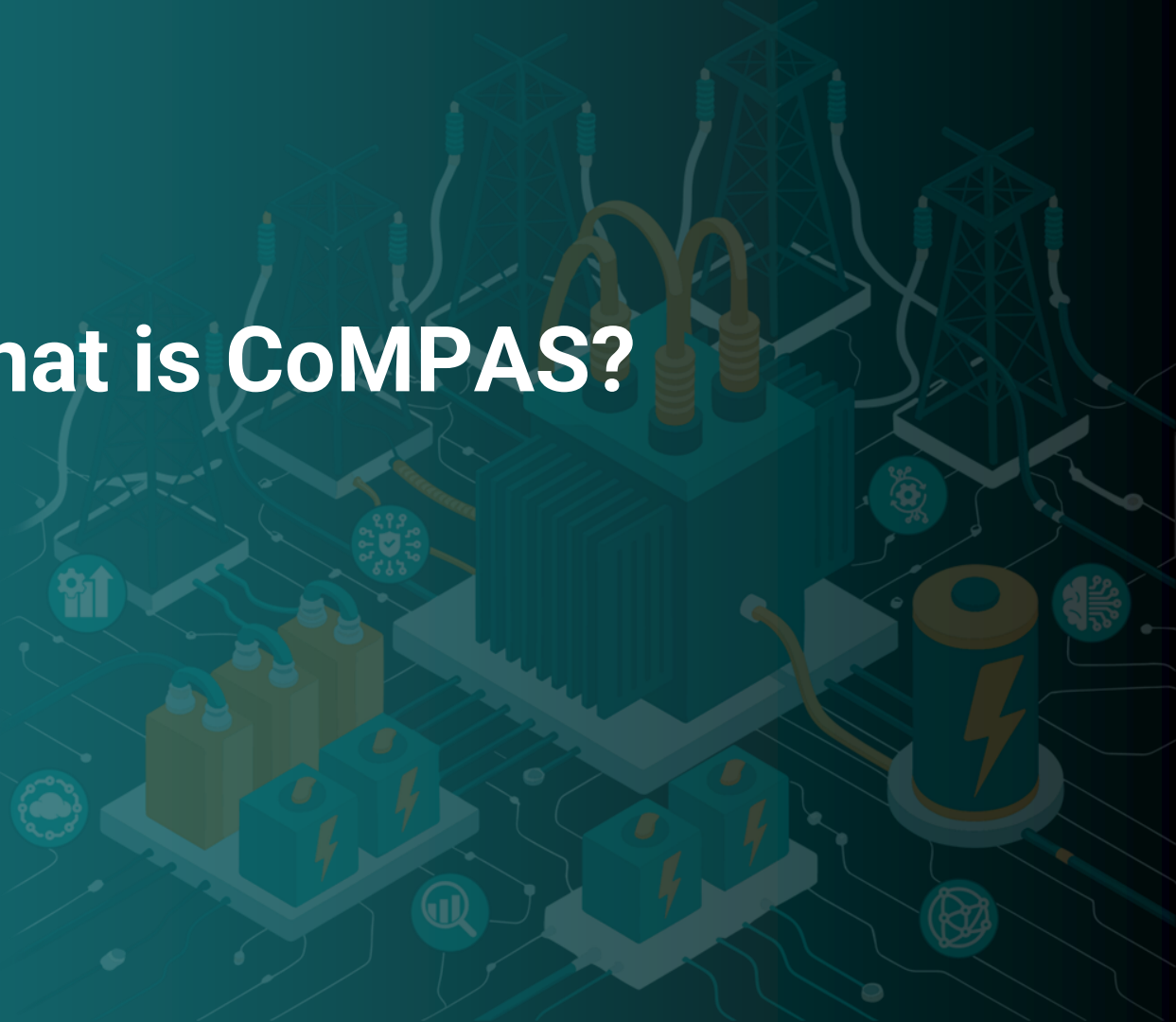
- Traveling wave
- Centralized (backup) diff
- PD online as protection
- Line differential protections
- Thermal ‘overcurrent’ protection / Dynamic protection settings
- Rate of change protections



Concept architecture & activities



What is CoMPAS?



(Co)nfiguration (M)odules for (P)ower industry (A)utomation (S)ystems

Brief Description:

Profile management and configuration for IEC 61850 protection, automation and control systems

Primary use cases:

- The planning process for grid infrastructure components facilitates the design of power system adaptations and the engineering of substation PACS to be installed or upgraded
- Specifying requirements and performing multi-vendor system integration and commissioning during the PACS procurement and delivery phase
- Managing assets in order to perform maintenance actions and PACS upgrades
- Improving the interlinkage between grid planning and PACS engineering processes
- Improving the interlinkage between the configuration of PACS and the configuration of central systems, including SCADA and asset management

Key links

Website: compas.energy

GitHub: https://github.com/com-pas/

Video: https://www.youtube.com/watch?v=CAMYMdUwrVg

OpenSSF Badge URL: https://www.bestpractices.dev/en/projects/5925/silver

CoMPAS demo: demo.compas.energy

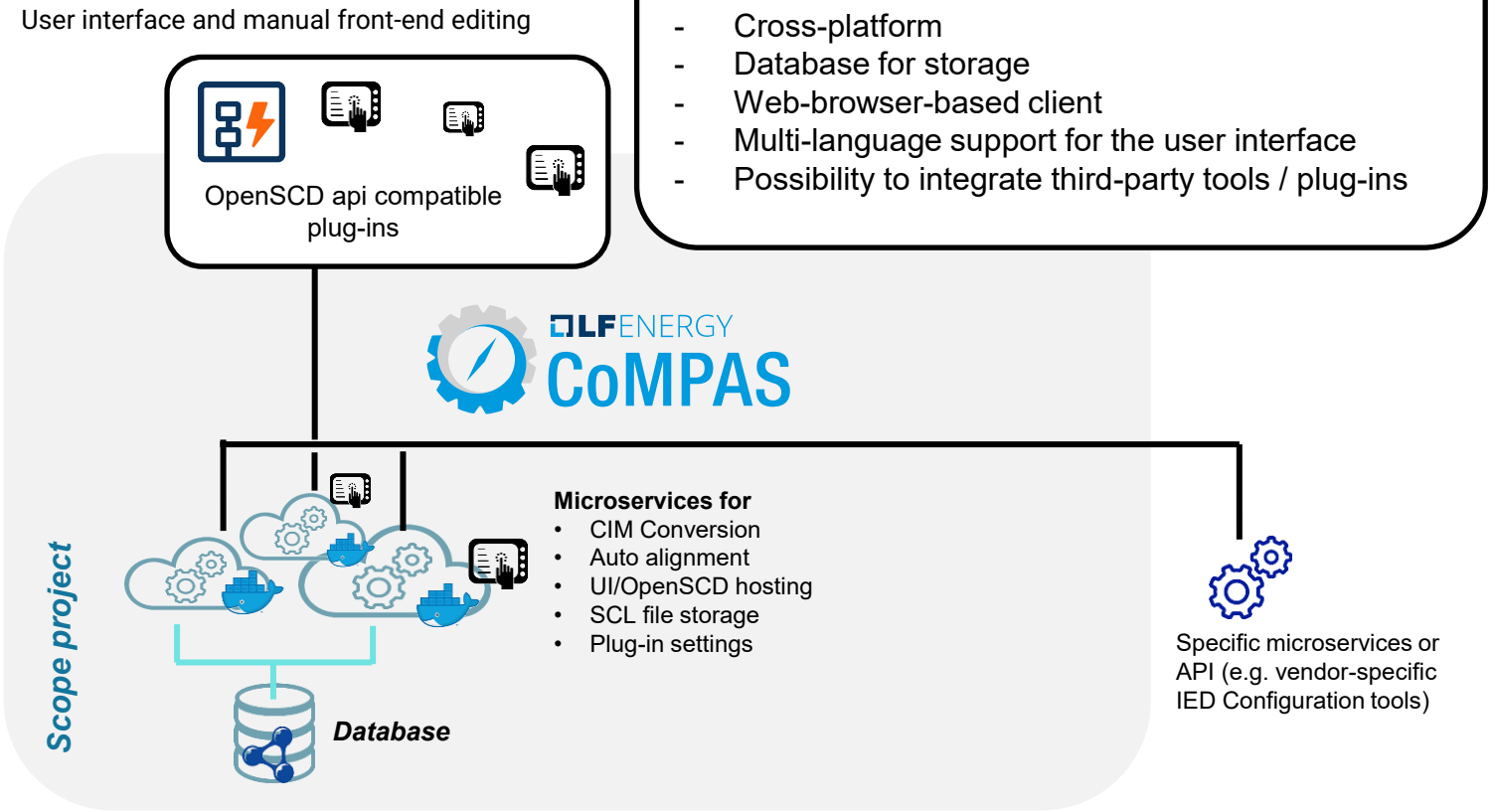
Reasons for staring the CoMPAS project

- Engineering worklfows varies: flexiblity needed
- Need to integrate with other services (E.g. CIM import)
- Lower the barrier to SCL editing and top-down engineering
- Configuration are more imported in full digital substations
- Centrally management of the (web) application



Architecture overview

Principles
- Microservices front-end/backend
- Cross-platform
- Database for storage
- Web-browser-based client
- Multi-language support for the user interface
- Possibility to integrate third-party tools / plug-ins



© 2019-2026 Omicron, Bearingpoint, Alliander, GE, National Grid, OSIsoft, RTE, Schneider Electric, TenneT. Licensed under Creative Commons Attribution 4.0 International (CC BY 4.0)

CoMPAS community



CoMPAS meetup participant companies

aedilis

Weidmüller

TRANSNET BW

BearingPoint

OMICRON

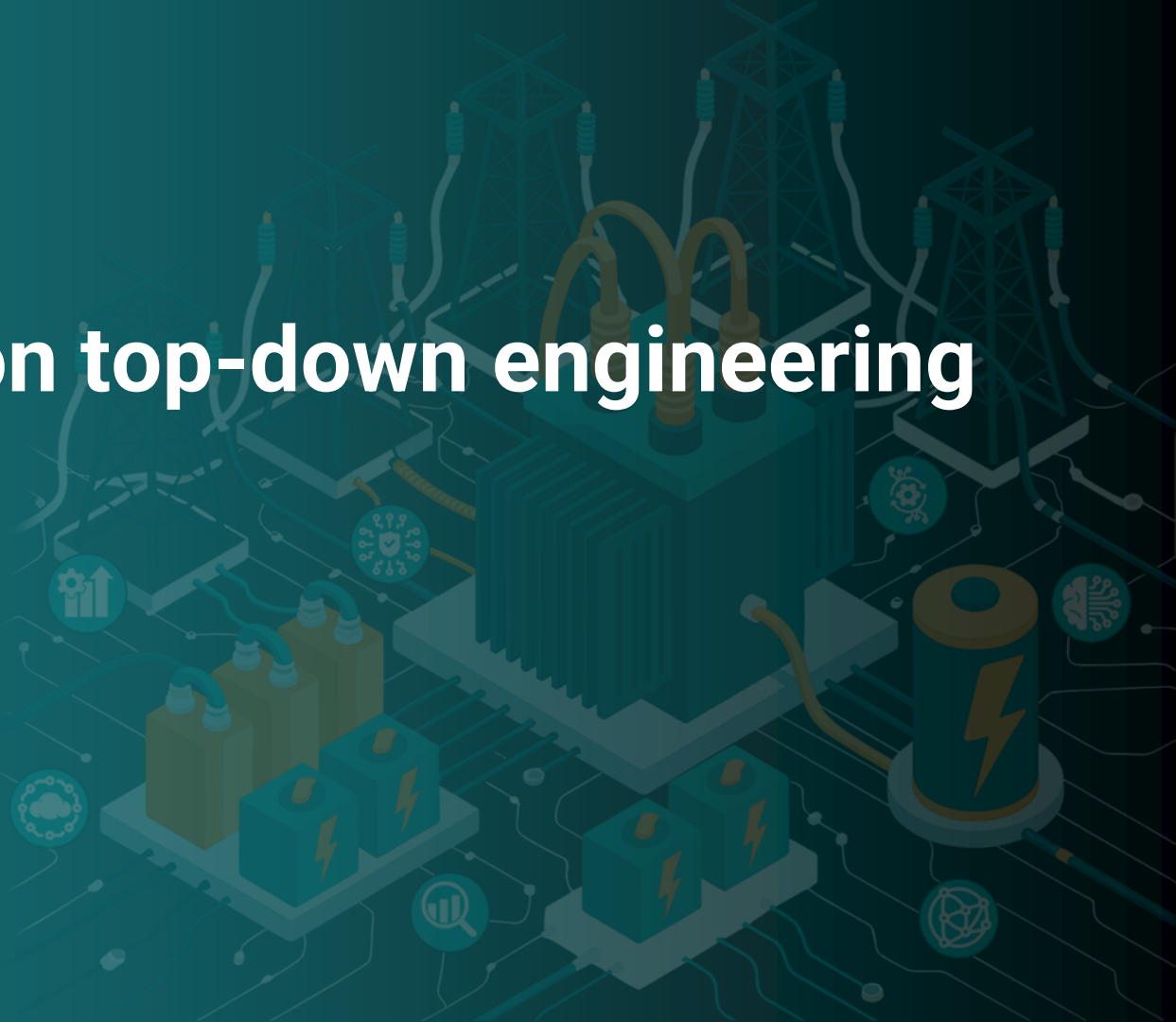
Belden

alliander

50hertz | Elia Group

SPRINTEINS

Introduction top-down engineering



What is 61850 SCL?

SCL : system configuration language

XML way to express substation/system functions



What and how

1. We want to have PAC functions in model to realize the vision of software defined substations.
2. We want to use IEC 61850 to model PAC functions.

- IEC 61850 is the de-facto standard to model PAC functions
- Standardized
- Vendor agnostic
- Widely applied and well tested
- A large basis for modelling PAC functions that is still growing
- Goes well with IEC CIM



What is top down engineering?

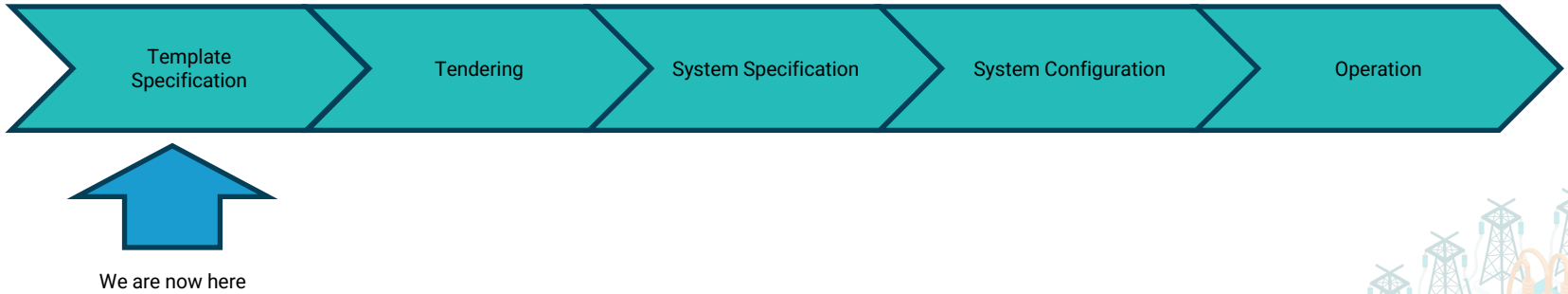
- Top-down generic: Make decision as early in a process as possible and reuse these in later stages of the process.

- Top-down in IEC 61850: Make decision as early in a process as possible, save those in SCL files and reuse these in later stages of the process.

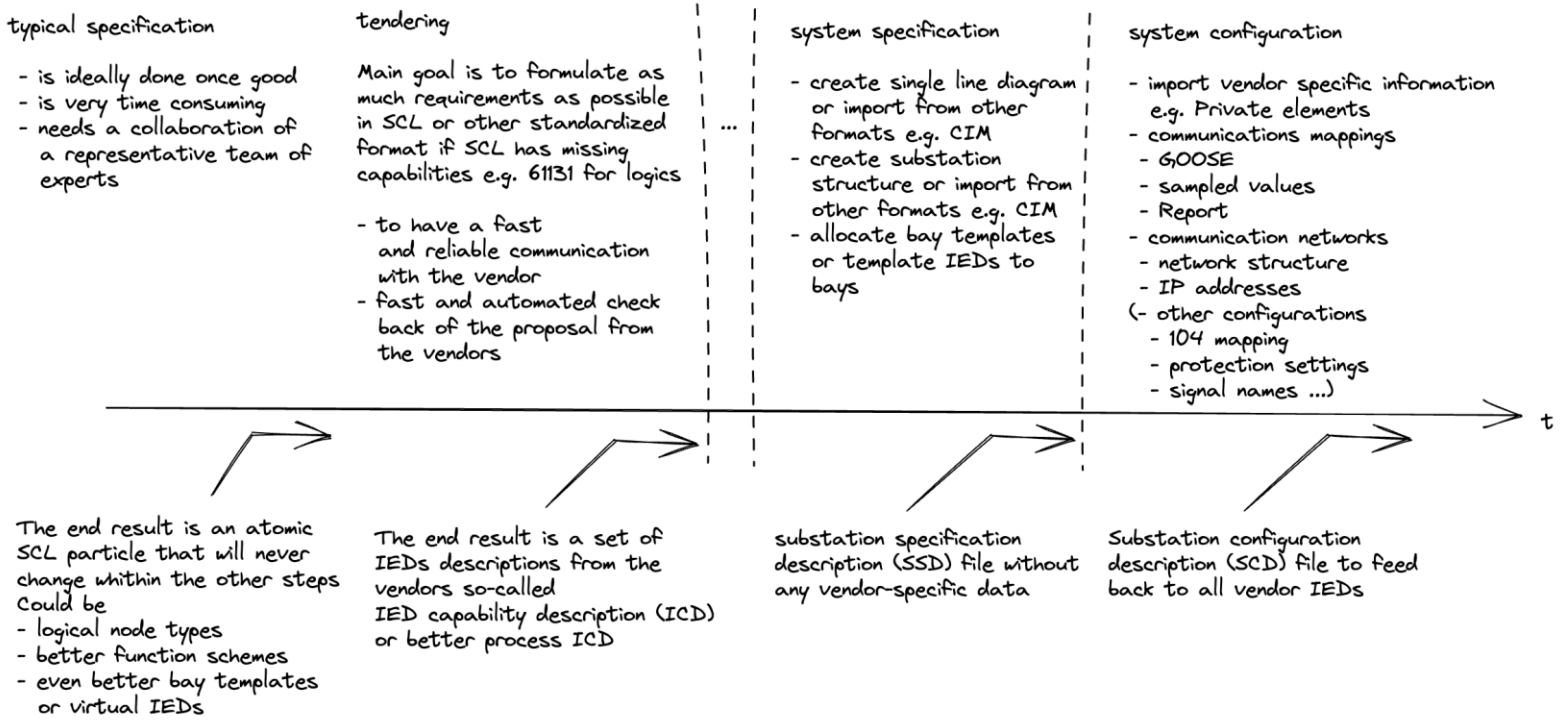


Our SCL workflow in development

A process – SCL workflow - that defines when and which information is saved to SCL files



Our SCL workflow in development in detail



Approach template specification

- Word document with all the LN, mandatory and optional
- Get requirements from:
  - Previous generation substation automation
  - Departments e.g. system operation, maintenance and support etc
- Review LN's selection
- Add the LN's to the LNtypedb in CoMPAS
- Test against vendor solutions
- Build bay typicals



LN selection

- What functions do we need as a company run operate the grid?
  - Measurement function
  - Protection functions
  - Etc

- Requires input from all data and function users e.g. protection policy, grid operations, system operations, maintaince and service department.

- If LN is selected: What parameters do you need for your function? DO/DA/Enum

LN document example

Document to discuss the choices

Mandatory or optional field

Non-Qirion usage

CSWI (Switch Controller)

CSWI					
Data object name	**Common data class**	**T**	**Explanation**	**PresConditions/ds**	**Qirion M/O**	**Liander M/O**
Descriptions						
NamPlt	LPL		(DomainLN) Name plate of the logical node	MONamPlt / MONamPlt		
Status information						
OpOpn	ACT		If its 'general' is true, the function decided to open circuit breaker (XCBR) or switch (XSWI). This data object is used only if no control services are available between CSWI and XCBR/XSWI and GOOSE is used instead. Note that the state must last long enough that the subscriber (XCBR/XSWI) can detect it.	O / F		
SelOpn	SPS		If true, a circuit breaker or switch has been selected for open operation.	O / F		
OpCls	ACT		If its 'general' is true, the function decided to close the circuit breaker (XCBR) or switch (XSWI).This data object is used only if no control services are available between CSWI and XCBR/XSWI and GOOSE is used instead. Note that the state must last long enough that the subscriber (XCBR/XSWI) can detect it.	O / F		
SelCls	SPS		If true, a circuit breaker or switch has been selected for close operation.	O / F		
Loc	SPS		If its 'general' is true, the function decided to open circuit breaker (XCBR) or switch (XSWI). This data object is used only if no control services are available between CSWI and XCBR/XSWI and GOOSE is used instead. Note that the state must last long enough that the subscriber (XCBR/XSWI) can detect it.	O / F		

Add LNs to CoMPAS

Multiple way to do depends on the way engineering approach

- E.g. TransnetBW LN editor
- Template generator / updater
- Choices are saved in an SSD file for later usage



Demo time!

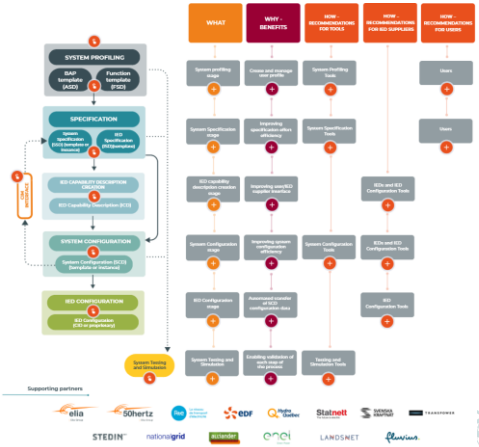
- How to select LNs?
  - Template generator
  - Template updater

- How to test in virtual IEDs?
  - Current limitation: services section need to be added manually



Next steps (to be started)

- Our approach: Bay typicals for tendering
- Full blown top-down based on the Elia vision is overkill for our usage
  - Functions are not that complex and common
  - Already used to bay typical engineering
- Some parts of the common vision are simplified



IEC61850 Engineering Process Common Vision

Approach

- Draw the single line
- Add functions to the single line
  - Add LNs to the functions
  - LNs are stored in the LNode database
- Connect functions/LN's
  - Current limitation: not all functions are present yet

61850-90-30 likely concepts to be used:

- ProcessResource
- LNodeSpecNaming
- LNodeInputs
- SourceRef

Demo time!

- Draw a single line
- Add functions



Next steps

- What is missing:
  - Placeholder for data coming from outside the bay
  - Add/remove LNs
  - Detailed editing

- Tender based on the SCL specification
  - (virtual) IED vendor should be flexible enough to map our specification on their internal functions
- Create bay typicals with IED's included

- Full engineering approach: from specification to validated SCD file





Questions? Reach out!

Community channels
or
sander.jansen@alliander.com
linkedin.com/in/sjjansen/

We are BearingPoint!



Our BearingPoint-Team contributing to CoMPAS

Eight people, every discipline covered - design, front-end, back-end, architecture, testing and hands-on substation know-how.



UX/UI-Designer



Front-End Developer



Front-End Developer



Solution Architect



Fullstack Developer



Back-End Developer



UX/UI-Tester



Power-User
CoMPAS TSC vice chair

Global presence with a European focus: BearingPoint has a strong focus on Europe and Germany



6.200+
Employees

> 1.900
Employees in DE, A & CH

50
Countries in which BearingPoint delivers projects

> 90
Partner in DE, A & CH

47
Offices in 24 Countries

12
Offices in DE, AT & CH

© BEARINGPOINT

We have extensive expertise across the entire energy value chain as well as in adjacent cross-functional areas

Our KRITIS expertise includes:

Transmission, Transformation & Distribution

- Implementation of next-generation SCADA solutions (modules and full systems)
- Introduction of event-streaming platforms
- Integration of secondary/OT systems with IT
- Implementation of OT/IT engineering processes supported by open-source tools

Trading

- Front- and back-office process optimization
- Optimization of ETRM systems, STP, and trading platforms
- Market transparency and transactions

Sales & Customer Service

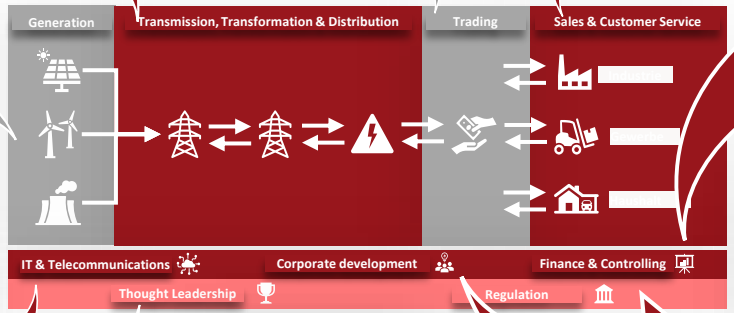
- Sales strategies and product management
- CRM implementations
- Billing systems

Finance & Controlling

- Optimization for incentive-based regulation
- Implementation of control and steering systems
- Regulatory and transparency reporting
- Market communication
- Digitalization of financial processes

Generation

- Operational excellence
- Benchmarking of conventional generation
- Renewable-energy funding studies and financing concepts
- Nearshore/offshore tendering



IT & Telecommunications

- AI / IT strategy, IT/OT convergence
- IT architecture, governance, ITSM
- Data strategies and concepts
- Cybersecurity, NIS 2, safety analyses and concepts
- Sourcing optimization, IT due diligence
- Specialized, functional and IT concepts, innovation

Thought Leadership

- Commissioned studies
- Market analyses
- Analytics services
- Research network
- Internal R&D

Corporate Development

- Business development and competitive analysis
- Strategy assessment, development, and implementation
- Organizational assessment and optimization
- Governance optimization and implementation
- Performance management design and implementation
- Profitability improvement (measured through financial KPIs)
- Competitor analysis
- Product management
- Product innovation
- Go-to-market strategy

© BEARINGPOINT

Designed to Extend: How Developers Shape CoMPAS



LF ENERGY SUMMIT EUROPE 2026 | Develop the Modern Grid, Together

OpenSCD / CoMPAS / Omicron / Meinberg / ...



newcomers

OpenSCD

OMICRON

OpenSCD

Fork#1

LFENERGY CoMPAS

OpenSCD

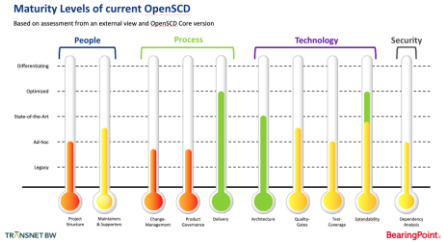
Fork#3

OpenSCD

Fork#2

MEINBERG

Code Due Diligence



*ecosystem before CoMPAS meetup in Aachen 2025

OpenSCD vendor-agnostic API

https://openscd.org/

API Spec

OpenSCD

OpenSCD API: https://github.com/openscd/oscd-api

Distributions

LF ENERGY CoMPAS

OpenSCD OMICRON

MEINBERG

… and many more

Public OpenSCD Distributions: https://openscd.org/get.html

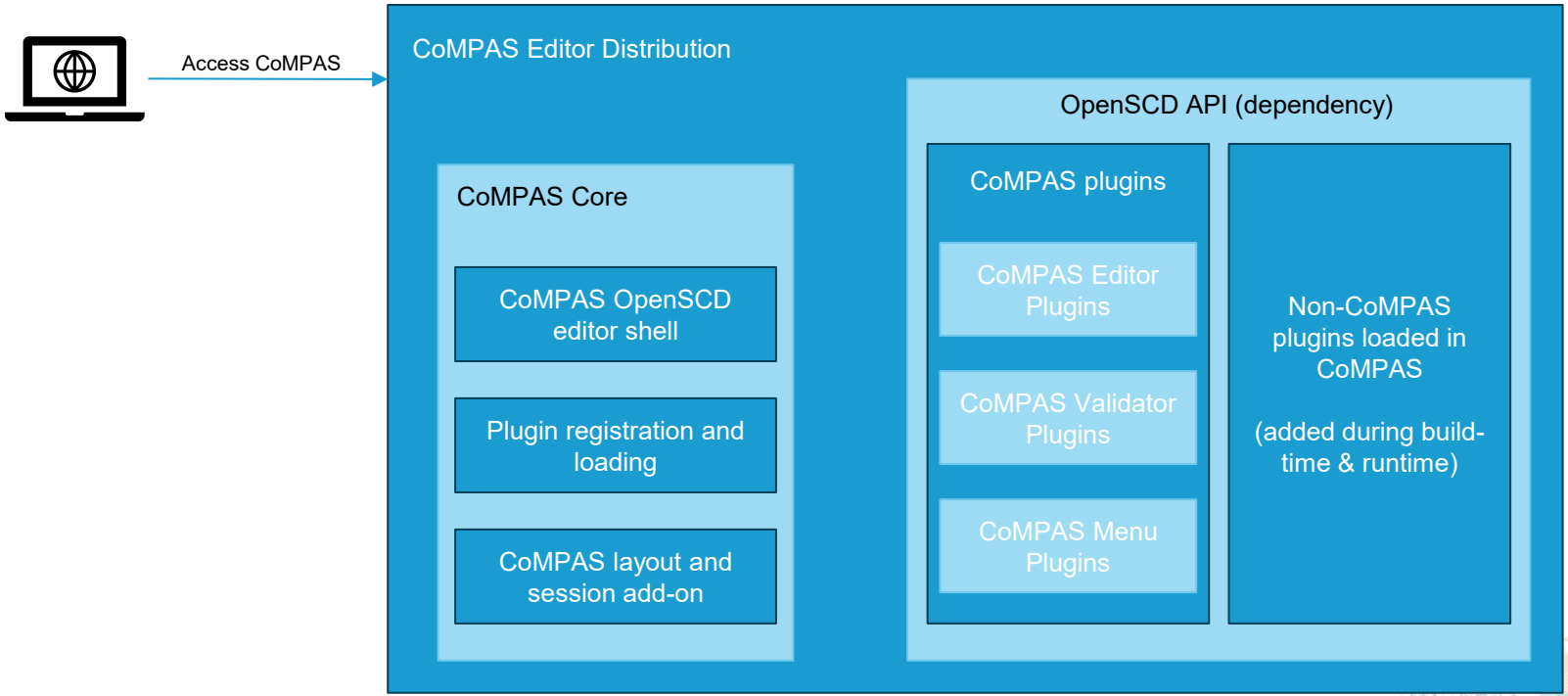
Contributor

alliander

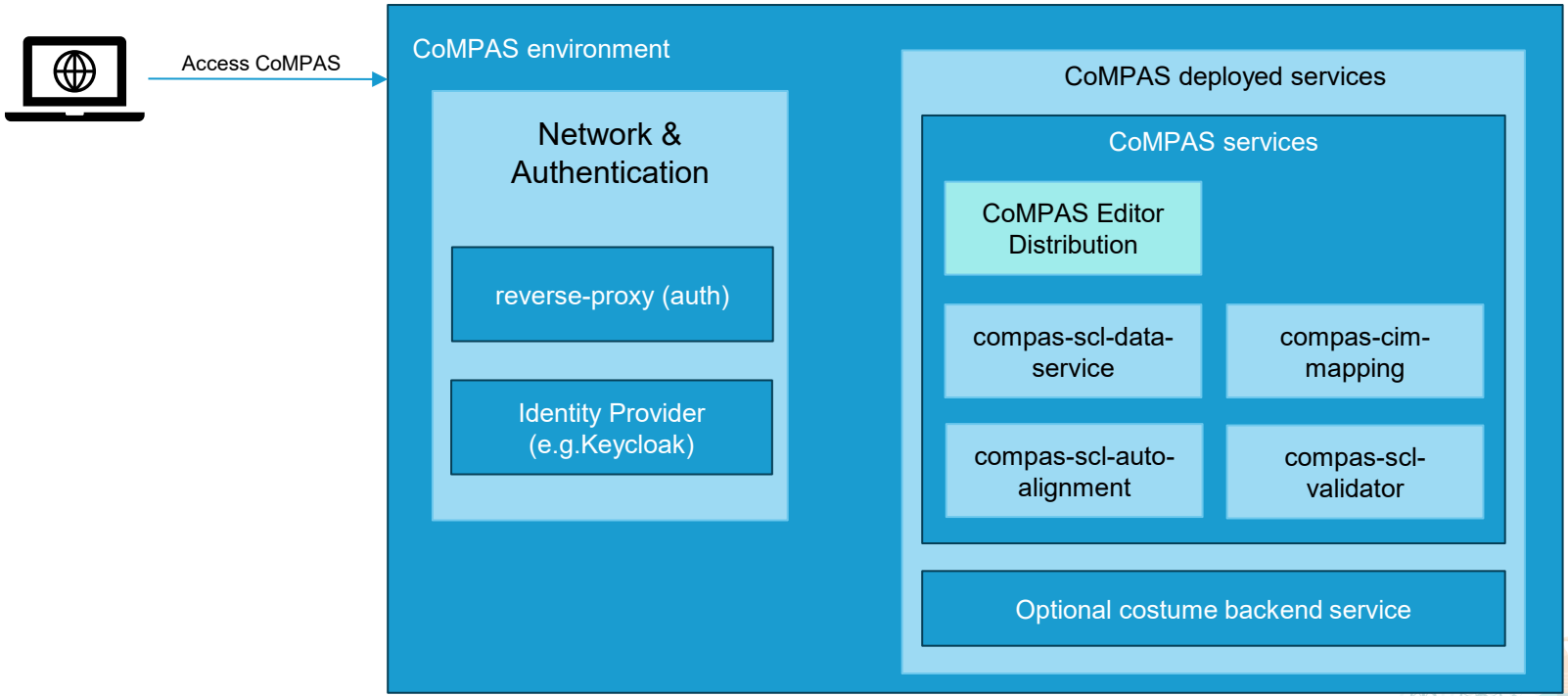
BearingPoint

SPRINTEINS

CoMPAS ecosystem: Editor Distribution Architecture



CoMPAS ecosystem: Provided Components



CoMPAS: Why adding functionality in OpenSCD/CoMPAS?

- Starting from a strong, well established and powerful IEC 61850 application
- Reuse of existing document, editor, issue, menu, wizard, and plugin mechanisms and code base
- Multiple functionality for IEC 61850 already available in the distributions
- Missing functionality can be added easily by provided mechanisms
- **Strong exchange with the community and learning from each other**
- Customization can be provided by multiple providers with IEC 61850 background
- Easy integration of already existing services
- OpenSCD/CoMPAS as OSS allows to share the organizational and monetary effort along multiple companies with a **better outcome**



CoMPAS: How to navigate trough existing functionality

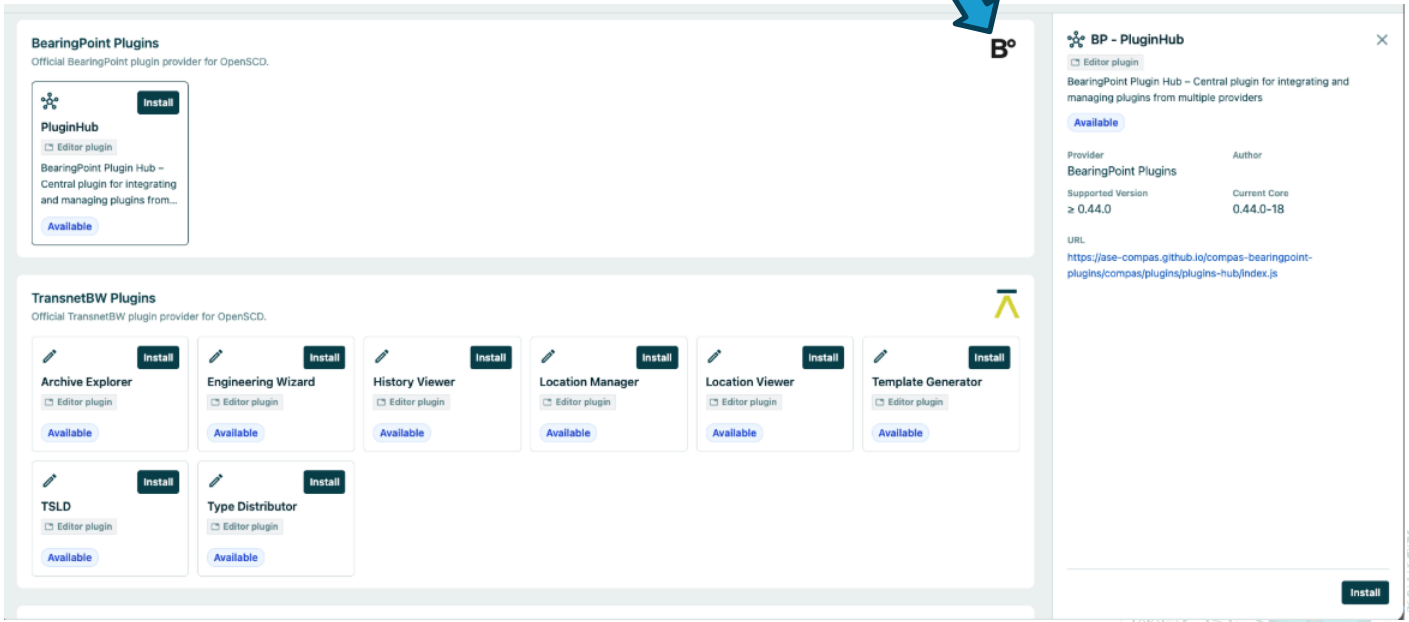
Documentation

Plugins-Hub plugin

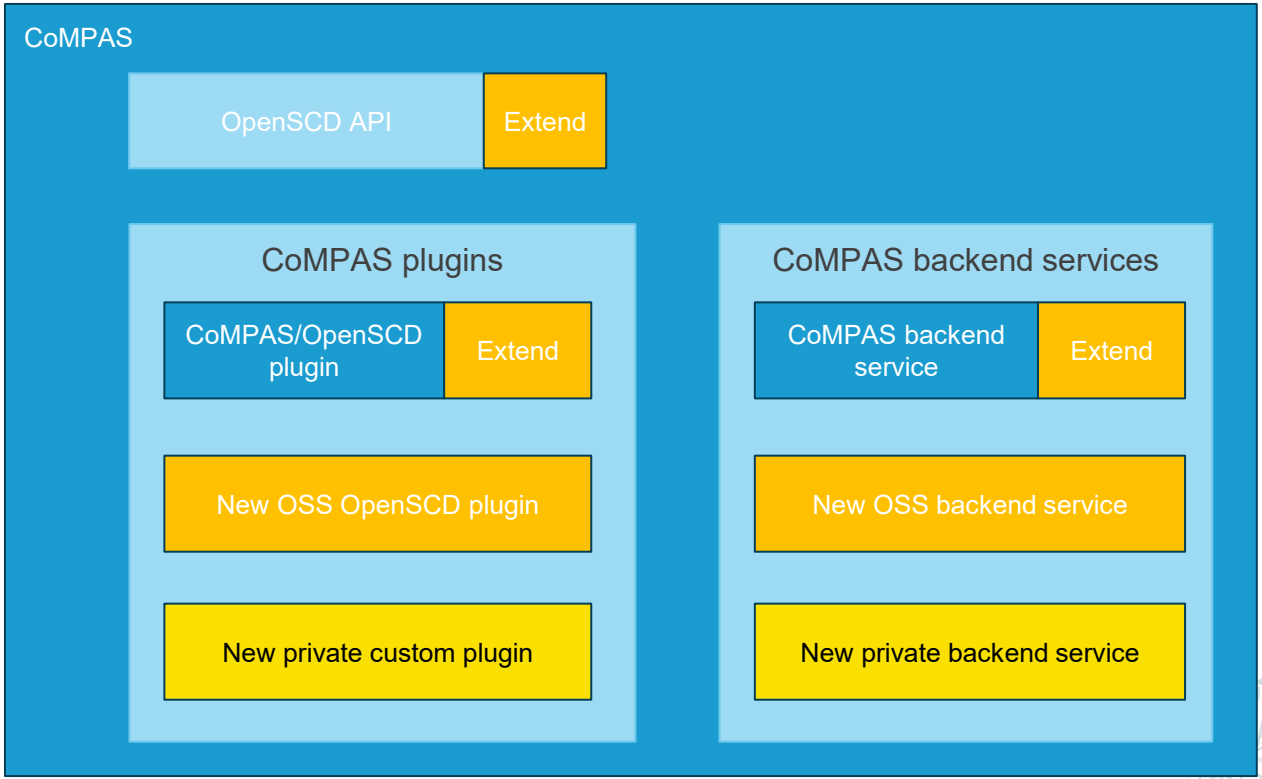
Plugins Provider

Plugins Provider Name

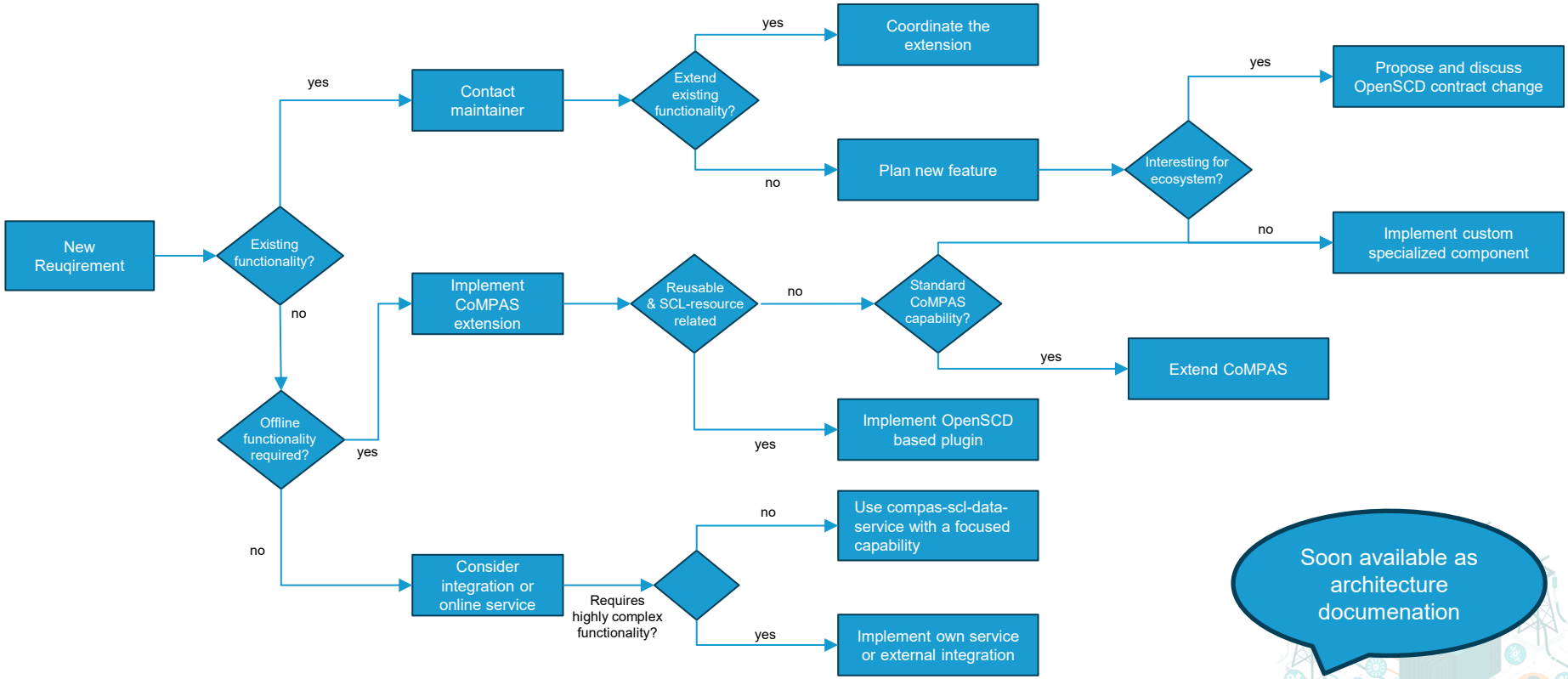
Provided Plugins



Various ways to extend CoMPAS



CoMPAS: Where to implement a feature - Decision path



CoMPAS Feature Quick Decision Matrix

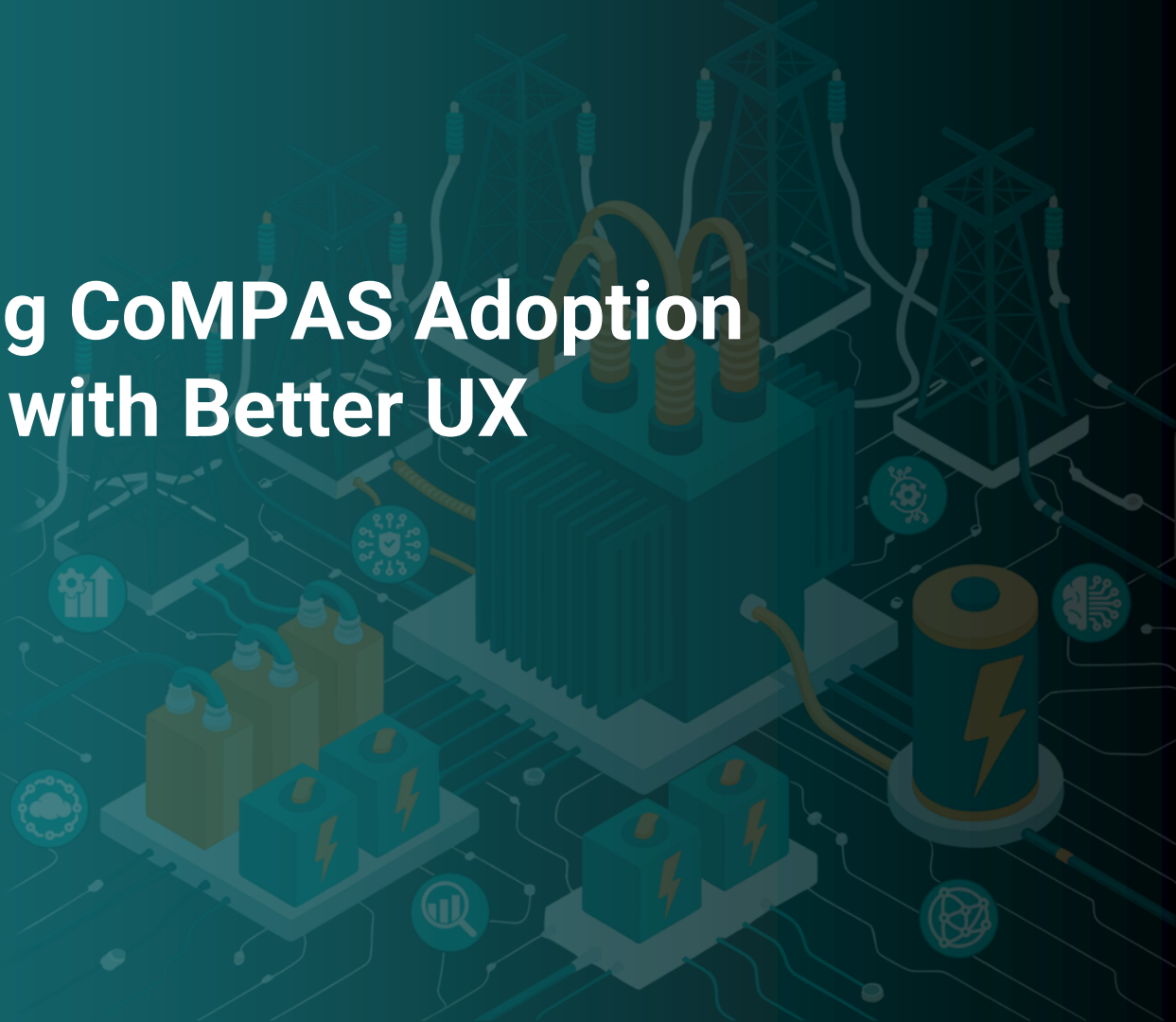
Primary need	Likely placement	Why
Existing functionality is close but does not fully meet the need	Discuss with the maintainer, creator, or community before implementation	An agreed extension or configuration may avoid duplicate functionality and preserve interoperability.
Offline, generic & specialized SCL view & editing	Reusable OpenSCD/CoMPAS plugin	It extends the editor without making the core carry niche behavior.
Reusable CoMPAS feature with optional versioned SCL- & non-SCL resources	OpenSCD plugin plus compas-scl-data-service in CoMPAS	The core feature remains backend-independent while CoMPAS can persist generic, versioned non-SCL resources.
Reusable need blocked by a missing OpenSCD extension point	Propose an OpenSCD contract change	A small governed contract is preferable to plugins relying on internals or a CoMPAS-specific backend.
Broad, standard CoMPAS workflow	CoMPAS base capability	It is part of the common product experience and needs long-term product ownership.
Company-specific functionality	CoMPAS plugin	It benefits CoMPAS users without becoming a dependency for every deployment.
Company-specific functionality shared across organization	CoMPAS plugin + CoMPAS backend services	Shared state, permissions, and auditability need explicit backend ownership.
Company-specific functionality shared across organization with highly complex functionality	CoMPAS plugin + CoMPAS backend services + custom backend service	The service provides generic resource persistence and versioning; the feature owns its domain behavior and user experience; specialized behavior resides in custom backend

Responsibilities & Governance

- OpenSCD community
  - https://openscd.org/resources.html
- CoMPAS community
  - https://lf-energy.atlassian.net/wiki/spaces/SHP/overview#Regular-meetings
- Plugins-Provider via Plugins-Hub Plugin
- BearingPoint GmbH



Driving CoMPAS Adoption with Better UX



LF ENERGY SUMMIT EUROPE 2026 | Develop the Modern Grid, Together

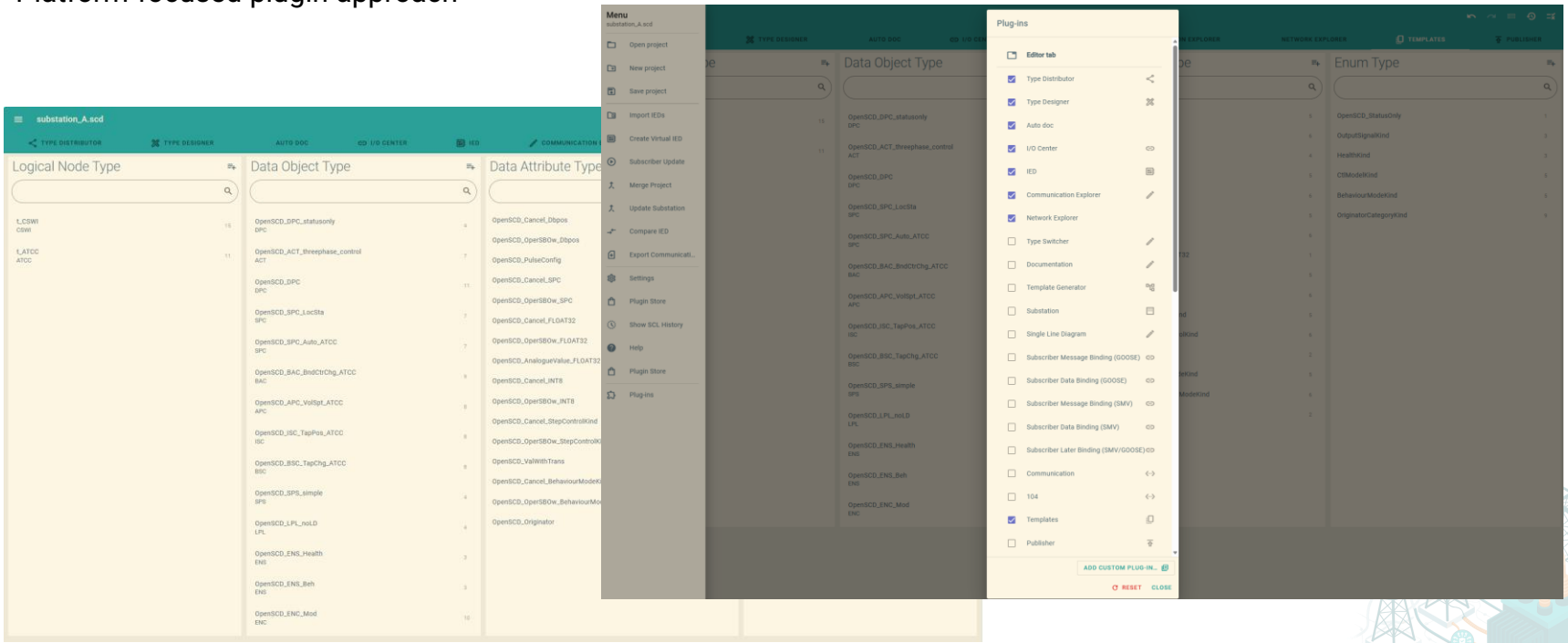
Agenda

- Where we started
- UX/UI Optimisation potential in CoMPAS
- UI-Deep-Dive
  - Plugin-Hub
  - Engineering Wizard
  - Template Generator
  - Future topics



Where we started

Platform-focused plugin approach



UX/UI Optimisation potential in CoMPAS

Current challenges

- **Lack of scalable UI governance**
- **Inconsistent user experience across plugins**
- **Difficult information discovery**
- **Poor visibility of relevant information**
- **Inefficient and complex user-workflows**
- **Preventable user errors and rework**



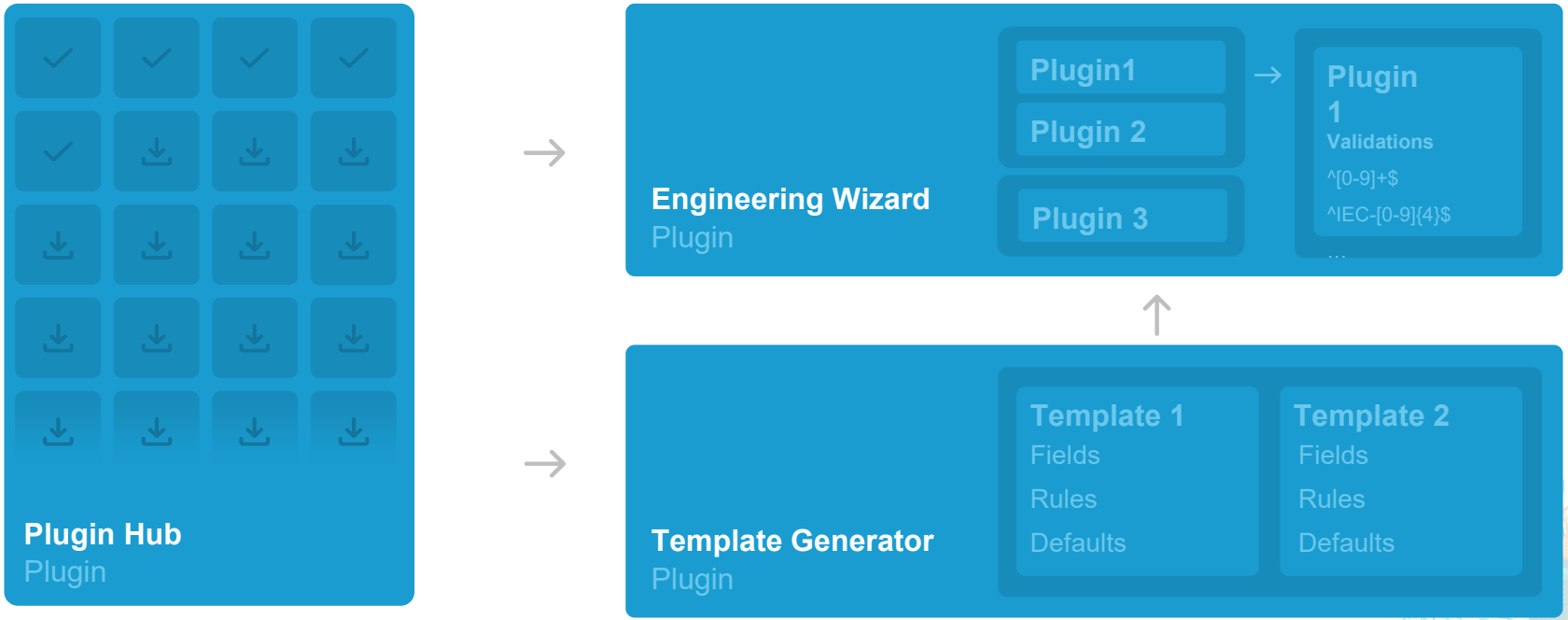
UX/UI Optimisation potential in CoMPAS

Possible solutions

Lack of scalable UI governance	Inconsistent user experience across plugins	Difficult information discovery	Poor visibility of relevant information	Inefficient and complex user-workflows	Preventable user errors and rework
↓	↓	↓	↓	↓	↓
Design System	**Consistent UX Patterns**	**Improved Information Architecture**	**Visual Hierarchy**	**User-Centered Design**	**User Validation**

General approach

How to work with CoMPAS

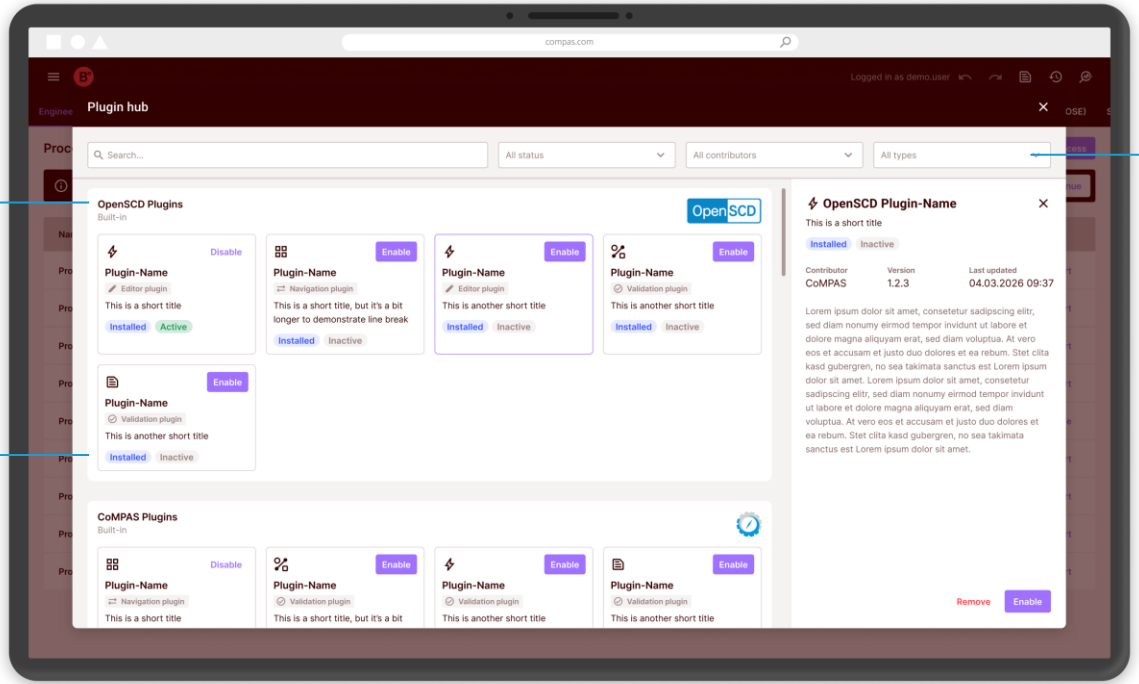


Plugin hub

Central app-store for all plugins

Simple plugin management

Always show the current state



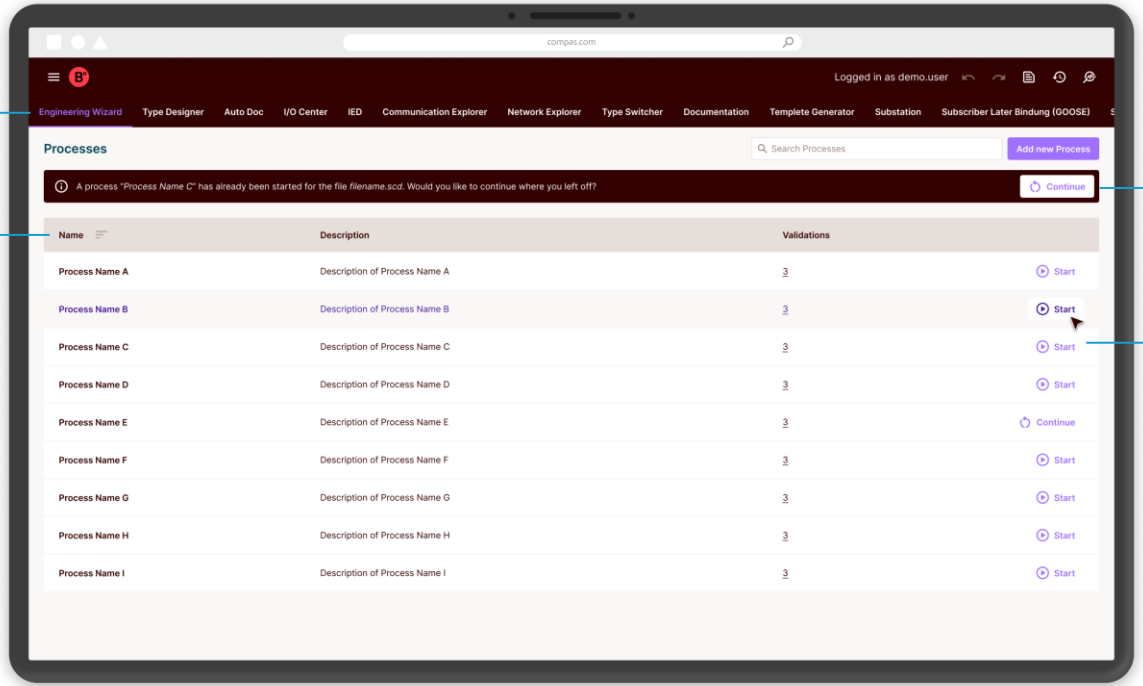
Plugins are easy to find through improved discoverability

Engineering Wizard – Process application

Manage your processes

New plugin appears in tabbar

Visual hierarchy & natural reading order



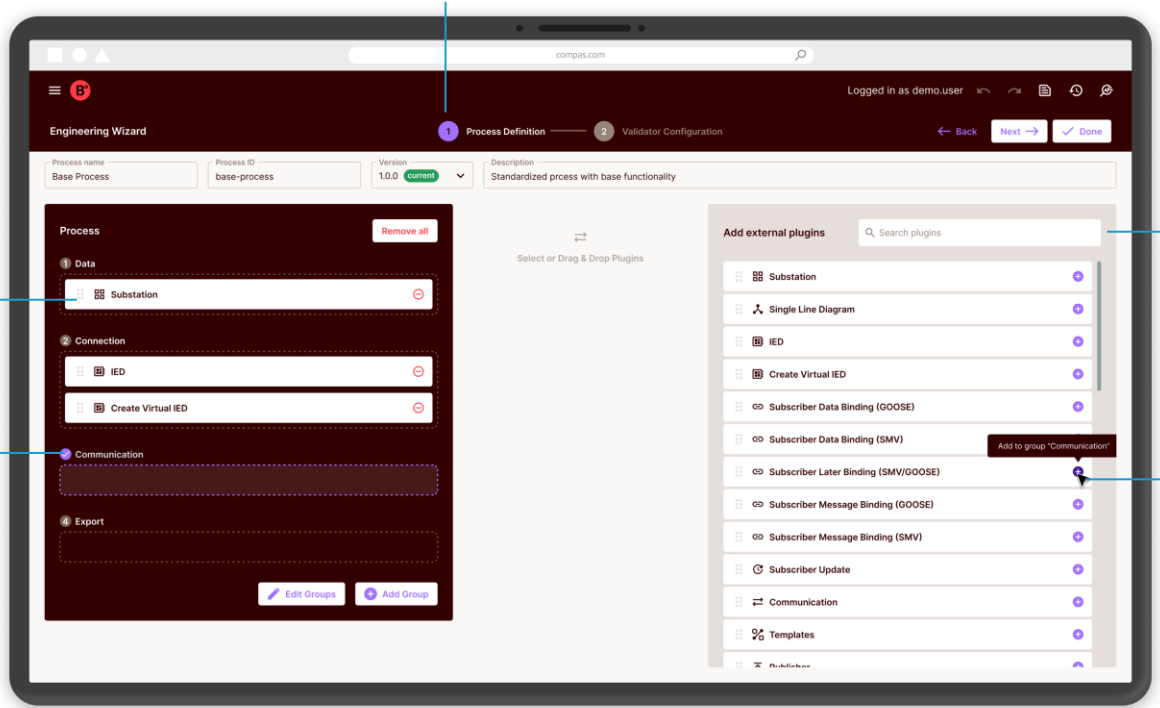
Guide users to their next relevant actions

„Progressive disclosure“ – only show relevant information in overview

Engineering Wizard – Process definition

Build your own process

Guided workflow



Drag & drop for intuitive structuring

Clear selection state for better orientation

Quickly find plugins through search functionality

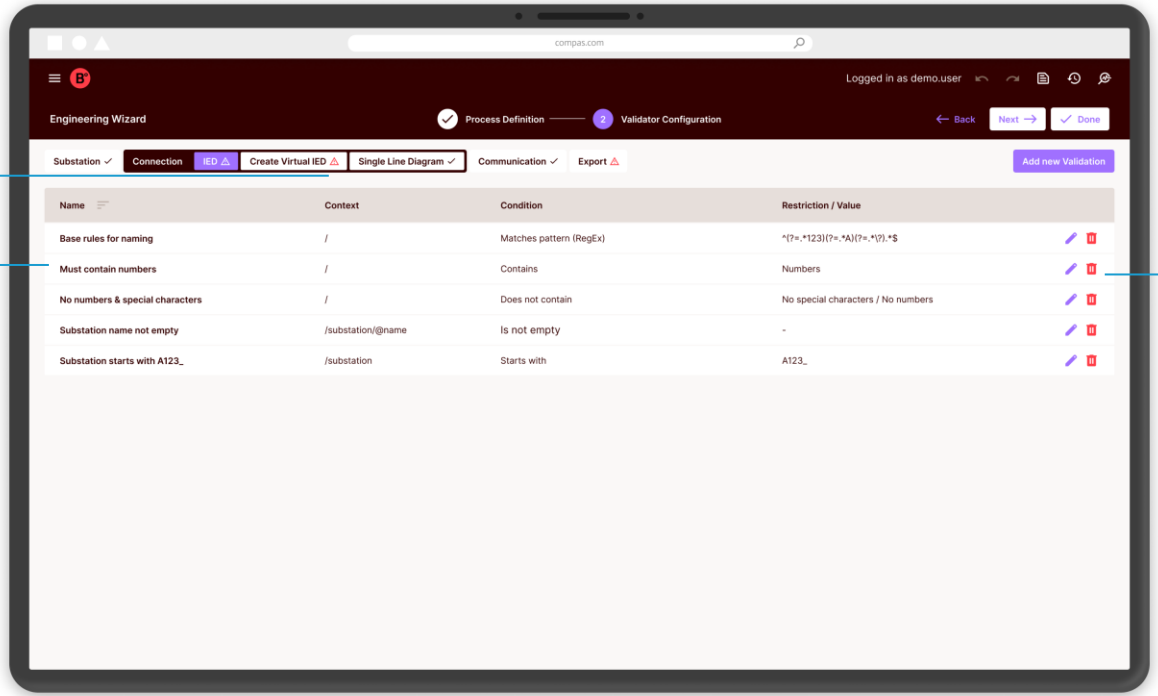
One-click assignment for fast & efficient plugin organization

Engineering Wizard – Process definition

Manage your user defined validation rules

Visual indicators for problems

Scannable data presentation



Inline actions

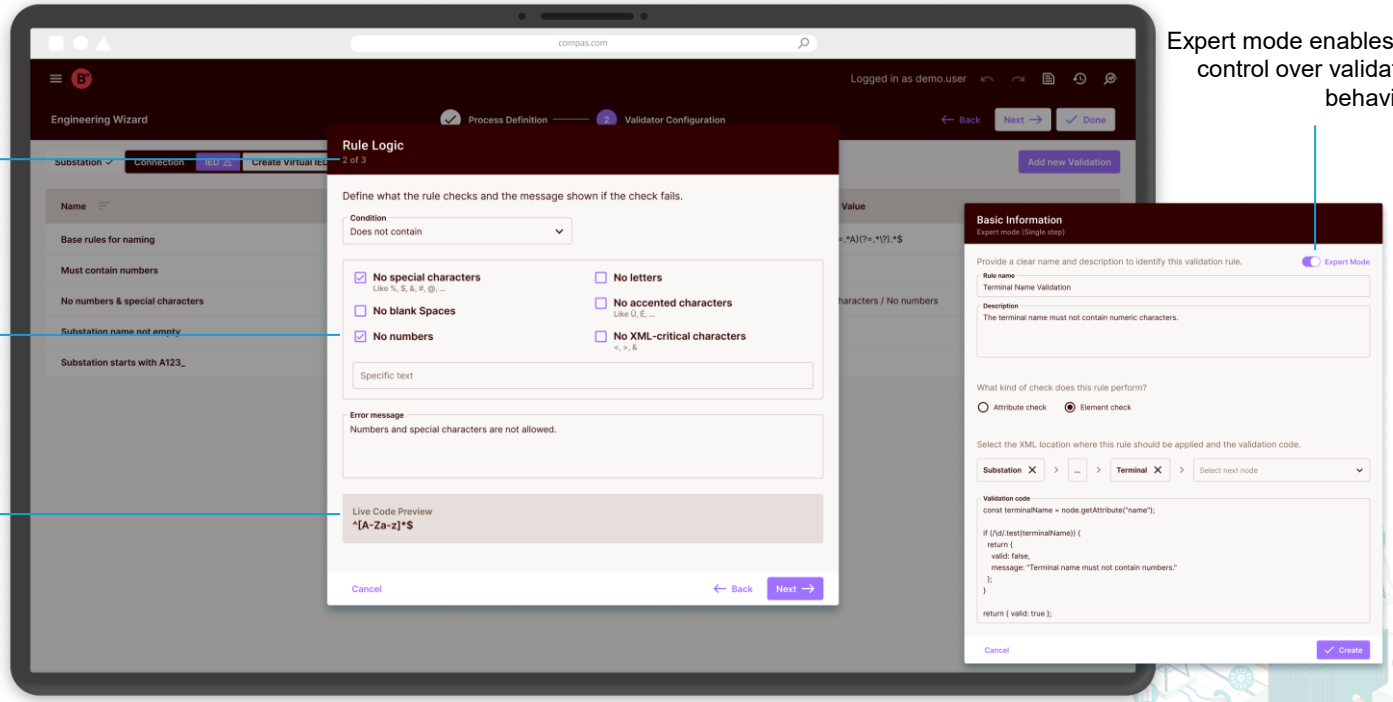
Engineering Wizard – Validation rules

Check SCL quality with your validation rules

Breaks complex rule creation into manageable steps

Guided validation creation without regex expertise

Live preview increases transparency and user confidence

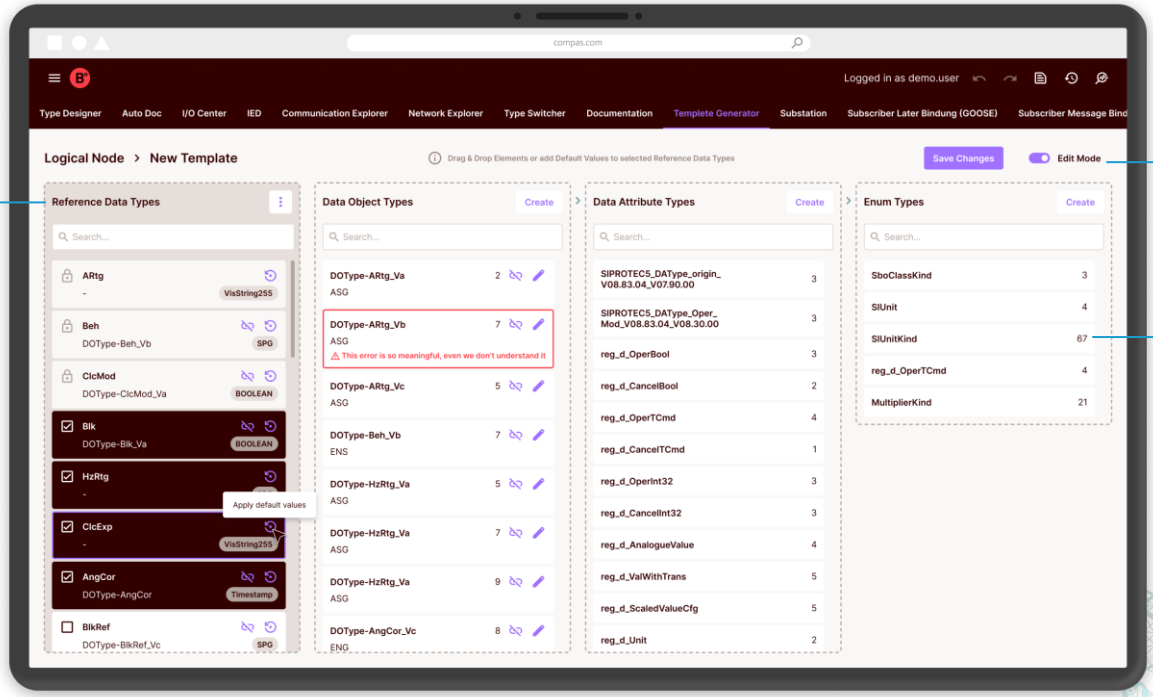


Expert mode enables full control over validation behaviour

Template Generator – Edit/View Mode

Reusable data types for standardized data models

Visualizes a complex data structure & supports natural left-to-right workflow



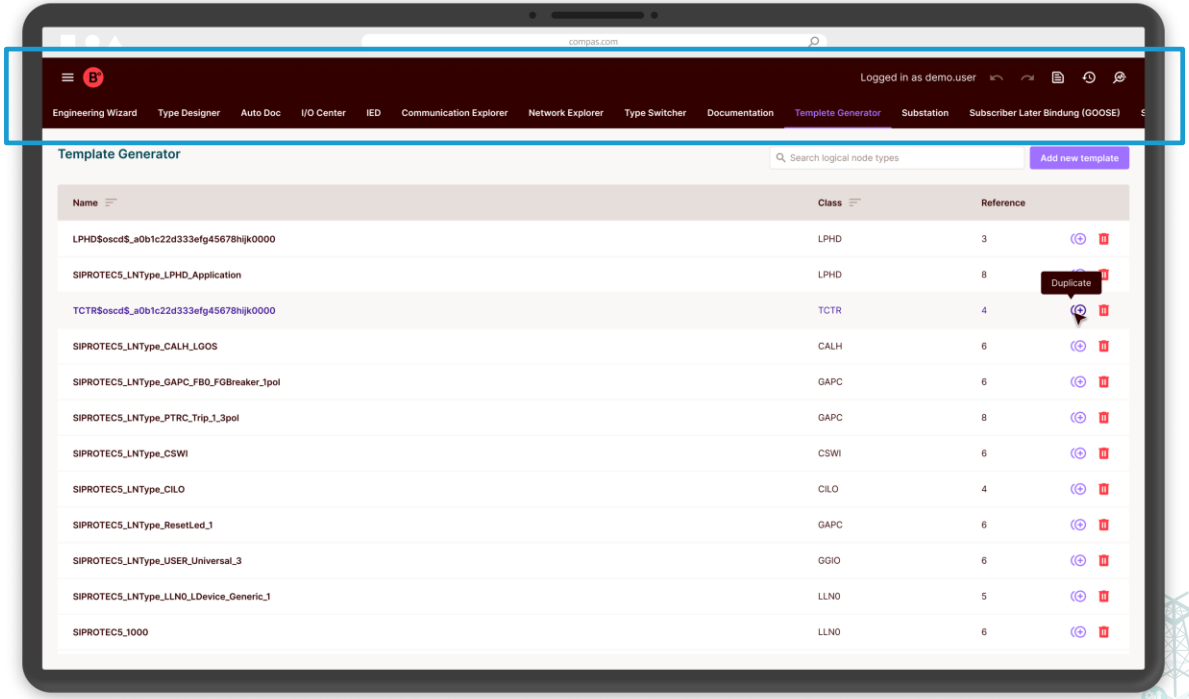
Toggle for edit/view mode

Provides a clear overview of template composition

Current Plugin Bar

We don‘t like horizontal lists

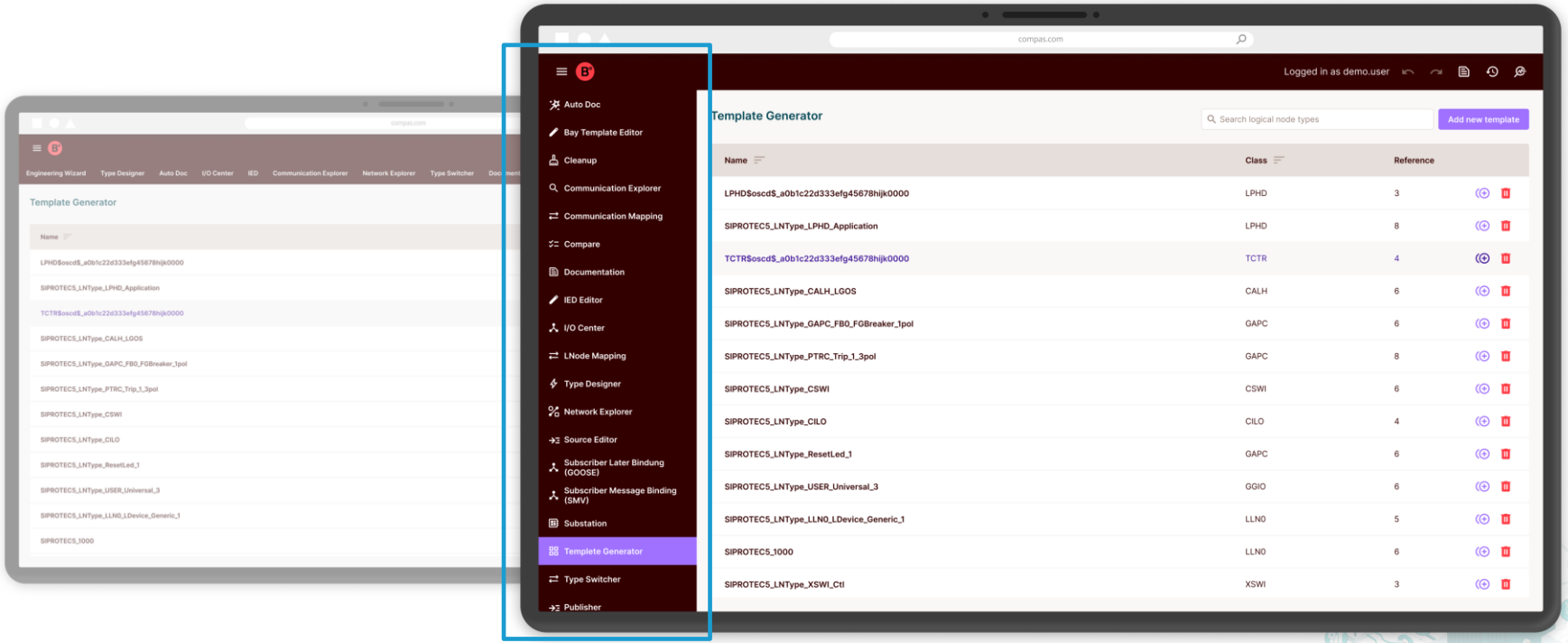
Current Plugin Bar



Quelle: https://omicronenergyoss.github.io/oscd-academy/

Future topic: New Plugin Bar

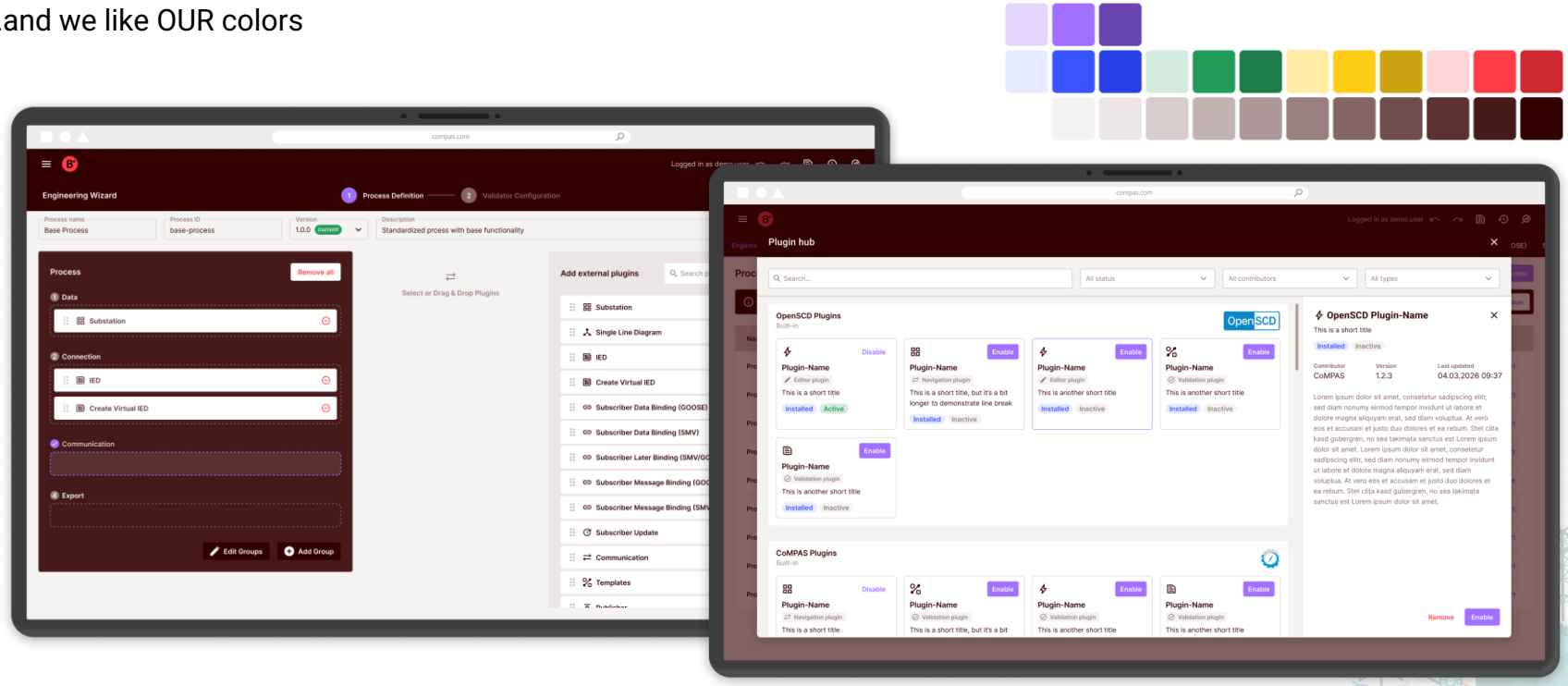
We (the community) DO like sidebars



Quelle: https://omicronenergyoss.github.io/oscd-academy/

Future topic: Themes

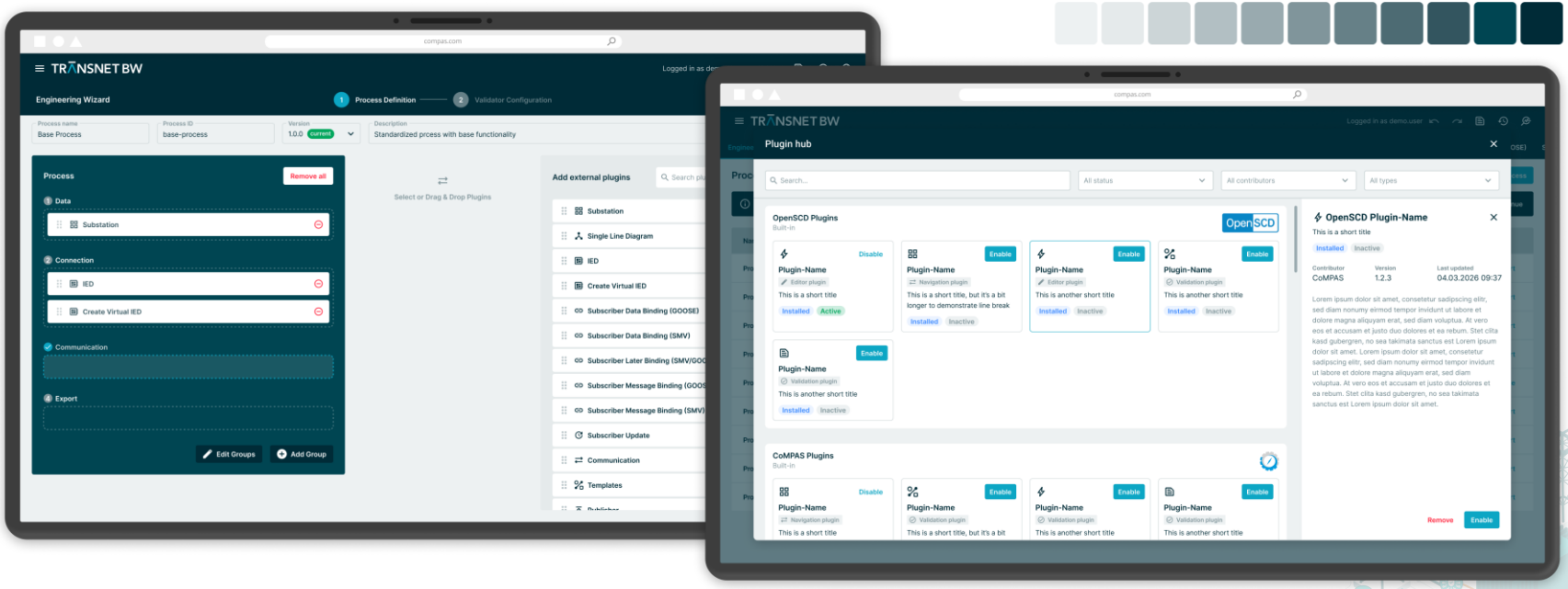
…and we like OUR colors



Future topic: Themes

…and we like OUR colors

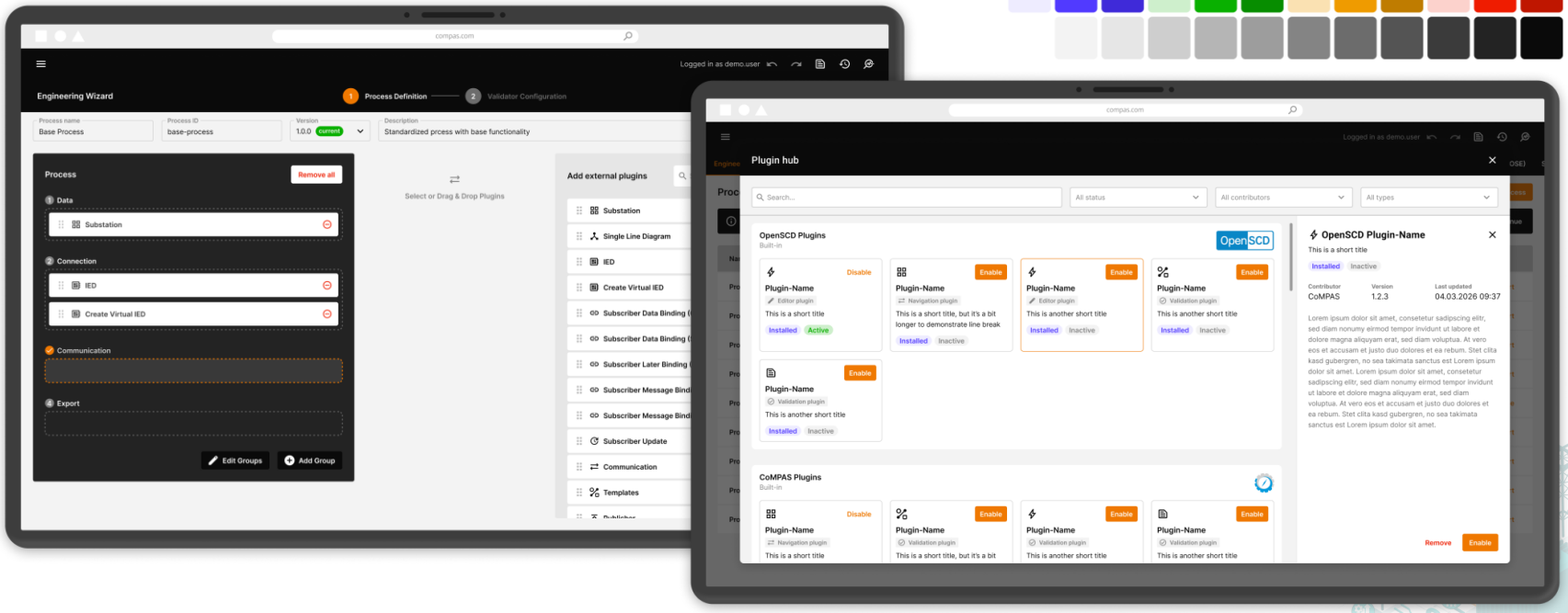




Future topic: Themes

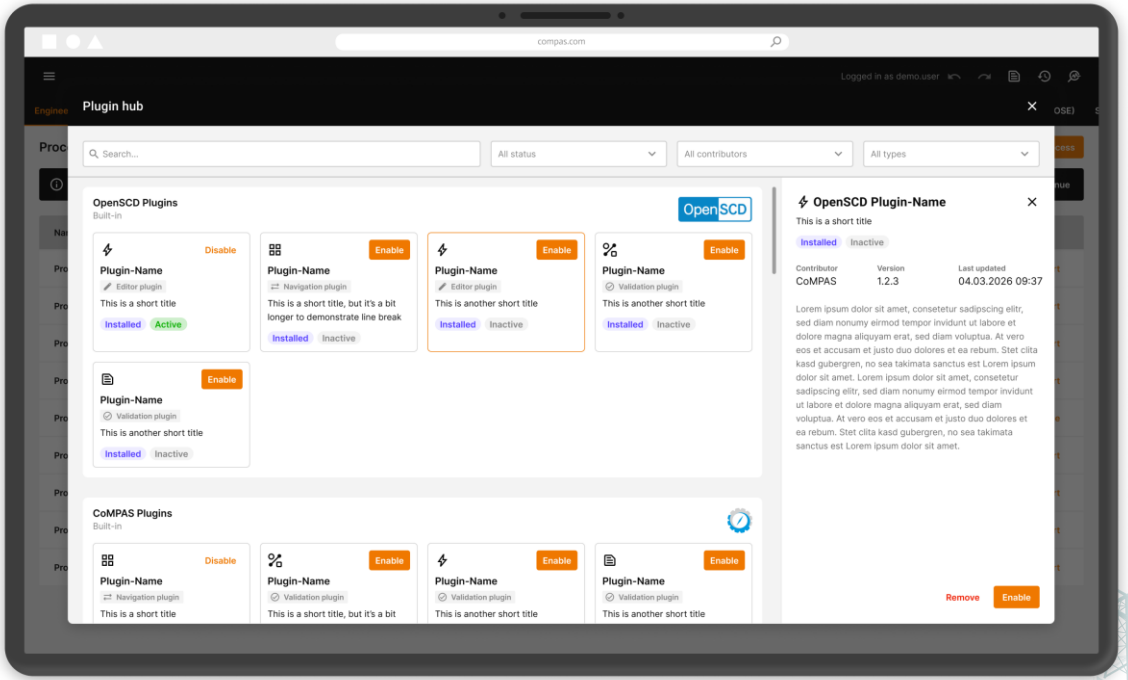
…and we like OUR colors





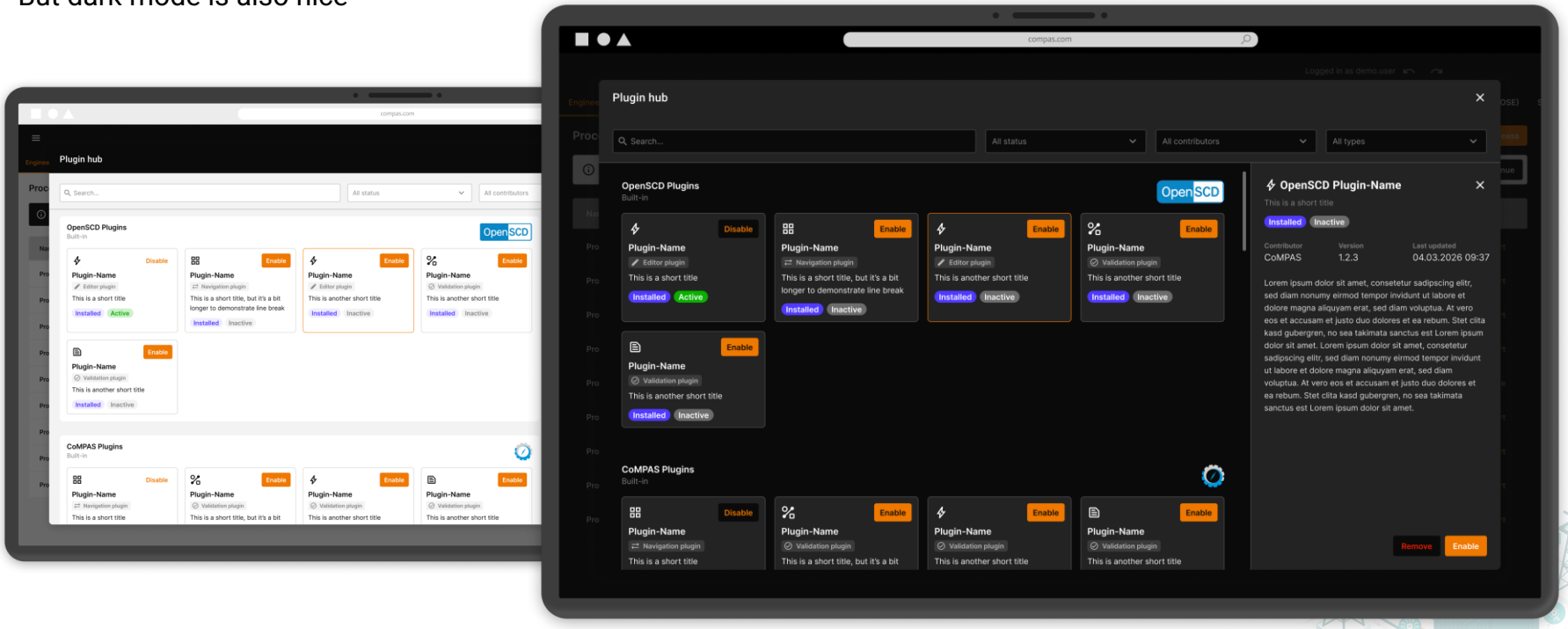
Current Light Mode

And light mode is nice



Future topic: Dark Mode

But dark mode is also nice



Extend CoMPAS in Minutes: Build Your First Plugin



What is a Plugin?

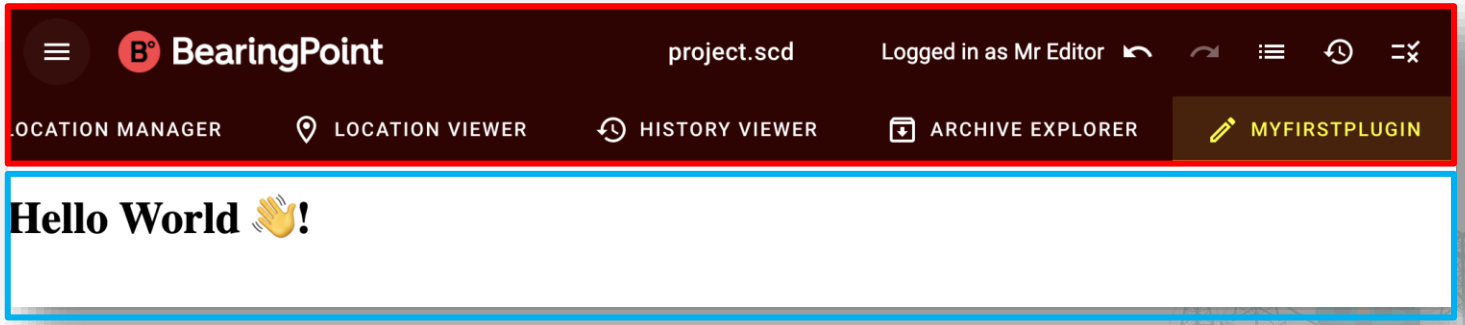
- A **Custom Element**
- **Default export** of the plugin
- Registered as **custom element**
- **Rendered** by CoMPAS Core

```
export default class MyFirstPlugin extends HTMLElement {

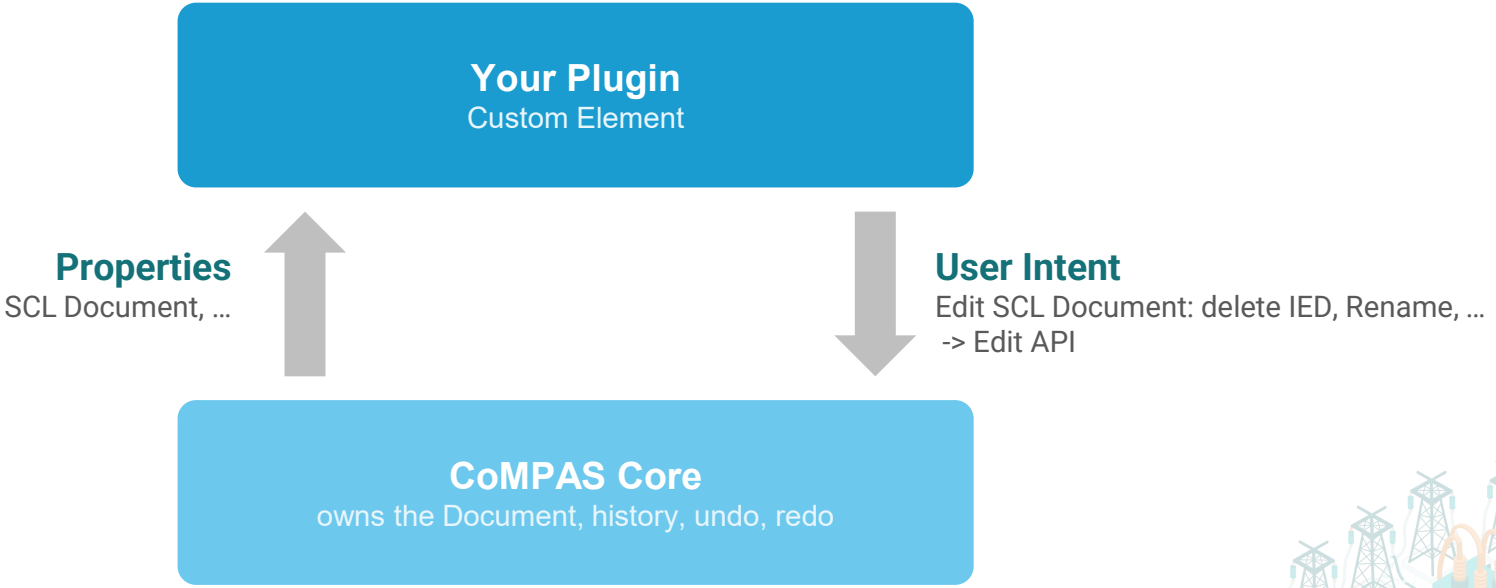
  connectedCallback() {
      this.innerHTML = '<h1>Hello World 👋!</h1>'
  }
}
```

CoMPAS Shell

Plugin



OpenSCD API – Communication



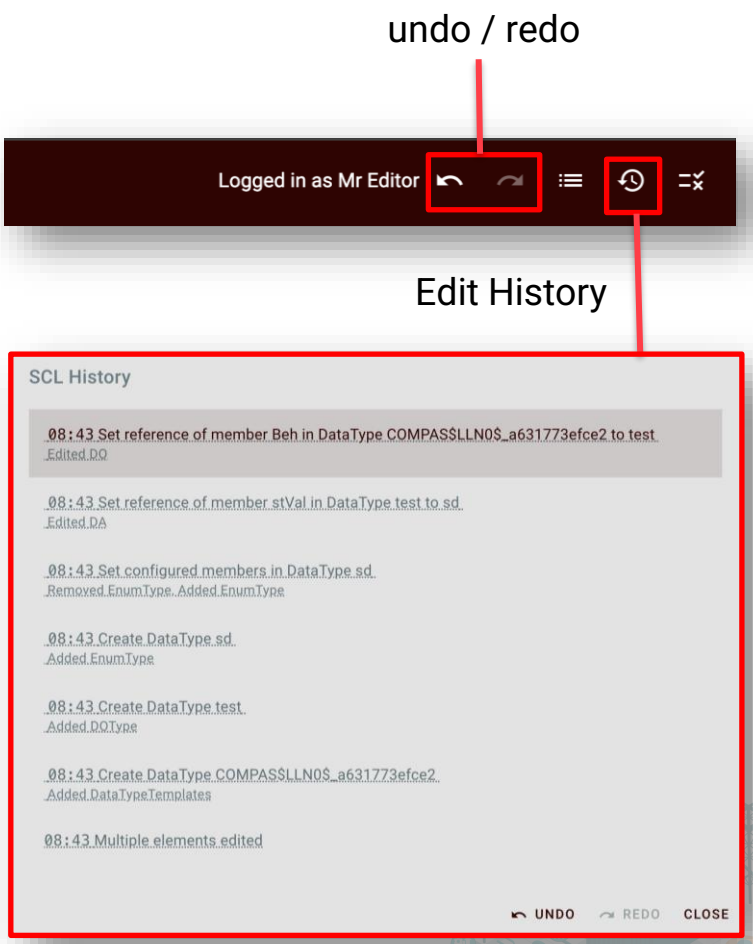
Advantages of the API

Your Plugin Stays Simple

No persistence, no versioning, no storage to implement.
Core own that.

One Edit History

Every plugin's edits land in the same **undo / redo** stack.
Users get on coherent history.



Plugin API – Properties

Property	Type	What it is
doc	XMLDocument	The opened SCL document. Read-only input.
editCount	number	Bumps on every change.
docName	string	Name of the currently open document.
locale	string	Selected UI language (IETF tag), defaults to en.
plugins	PluginSet	The full set of registered plugins.



Example – List IED Names

```
export default class MyFirstPlugin extends HTMLElement {

    #doc?: XMLDocument;

    set doc(doc: XMLDocument) {
        this.#doc = doc;
        this.render()
    }

    set editCount(_: number) {
        this.render();
    }

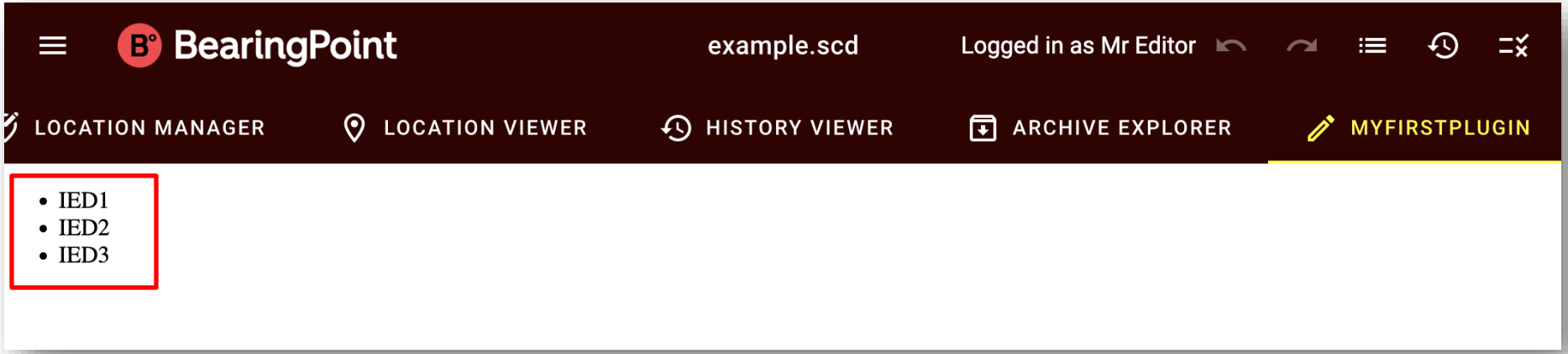
    render() {
        if (!this.#doc) {
            this.innerHTML = `<h1>No document provided.</h1>`;
            return;
        }

        const ieds = Array.from(this.#doc.querySelectorAll('IED'));
        this.innerHTML = `<ul>${ieds.map(ied => `<li>${ied.getAttribute('name')}</li>`).join('')}</ul>`;
    }
}
```

SCL Example with IED's

```
<?xml version="1.0" encoding="UTF-8"?>
<SCL xmlns="http://www.iec.ch/61850/2003/SCL" version="2007" revision="B" release="4">
  <Header id="example" version="1" revision="1" toolID="open-scd-plugins"/>
  <Substation name="Substation A" desc="Demo substation">
    <VoltageLevel name="V1" nomFreq="50" numPhases="3">
      <Bay name="Bay A"/>
    </VoltageLevel>
  </Substation>
  <IED name="IED1" type="Demo" manufacturer="CoMPAS">
    <AccessPoint name="AP1"/>
  </IED>
  <IED name="IED2" type="Demo" manufacturer="CoMPAS">
    <AccessPoint name="AP1"/>
  </IED>
  <IED name="IED3" type="Demo" manufacturer="CoMPAS">
    <AccessPoint name="AP1"/>
  </IED>
</SCL>
```

Example – List IED Names in Plugin



Plugin API – Communicating User Intent

Plugins **dispatch edit events** to tell CoMPAS Core what the user want to do.

```
this.dispatchEvent(newEditEventV2(event))
```

01

Custom Event

Plugin creates and **dispatches** an event.

02

Event Bubbles

The event **bubbles through the DOM**.

03

CoMPAS Core handles it

OpenSCD Core **receives** and **applies** edit event and **increases editCount**.



Plugin API – Edit Event

An **event** is a JavaScript **object**.

Event Types

Insert
Add a node

Set Attributes
Change element attributes

Set Context
Change element text

Remove
Remove node

Complex edits
Combine multiple edits into one operation



Example – Rename IED

```
import { newEditEventV2 } from "@openscd/oscd-api/utils.js";

export default class MyFirstPlugin extends HTMLElement {

    // ...

    renameIed(ied: Element, newName: string) {
        const event: SetAttributes = {
            element: ied, // The IED element to be renamed
            attributes: { name: newName }, // set the attribute name to newName
            attributesNS: {}
        };

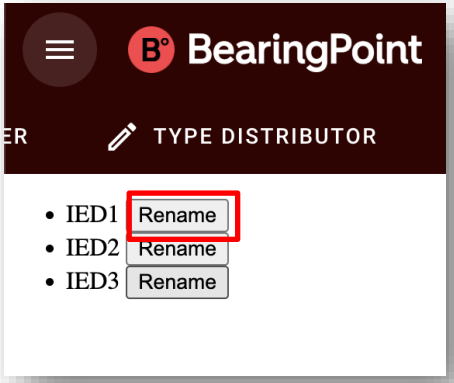
        this.dispatchEvent(
            newEditEventV2(event, { title: `Renaming IED to ${newName}` })
        );
    }

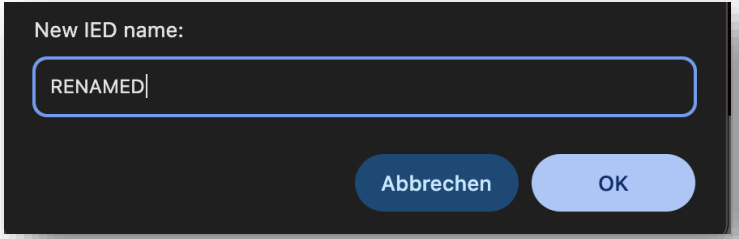
    // ...

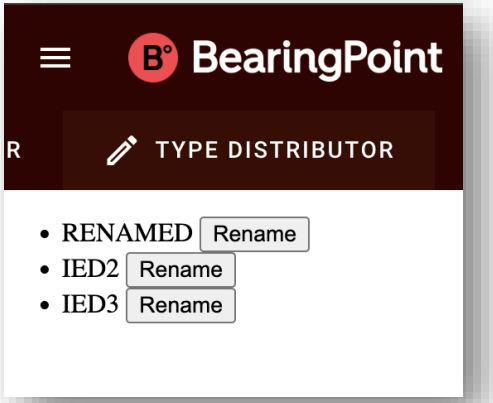
}
```



Example – Rename "IED1" to "RENAMED"







Integrate a Plugin

Build

Bundle plugin into a **single JavaScript** file.
e.g. index.js`.js`

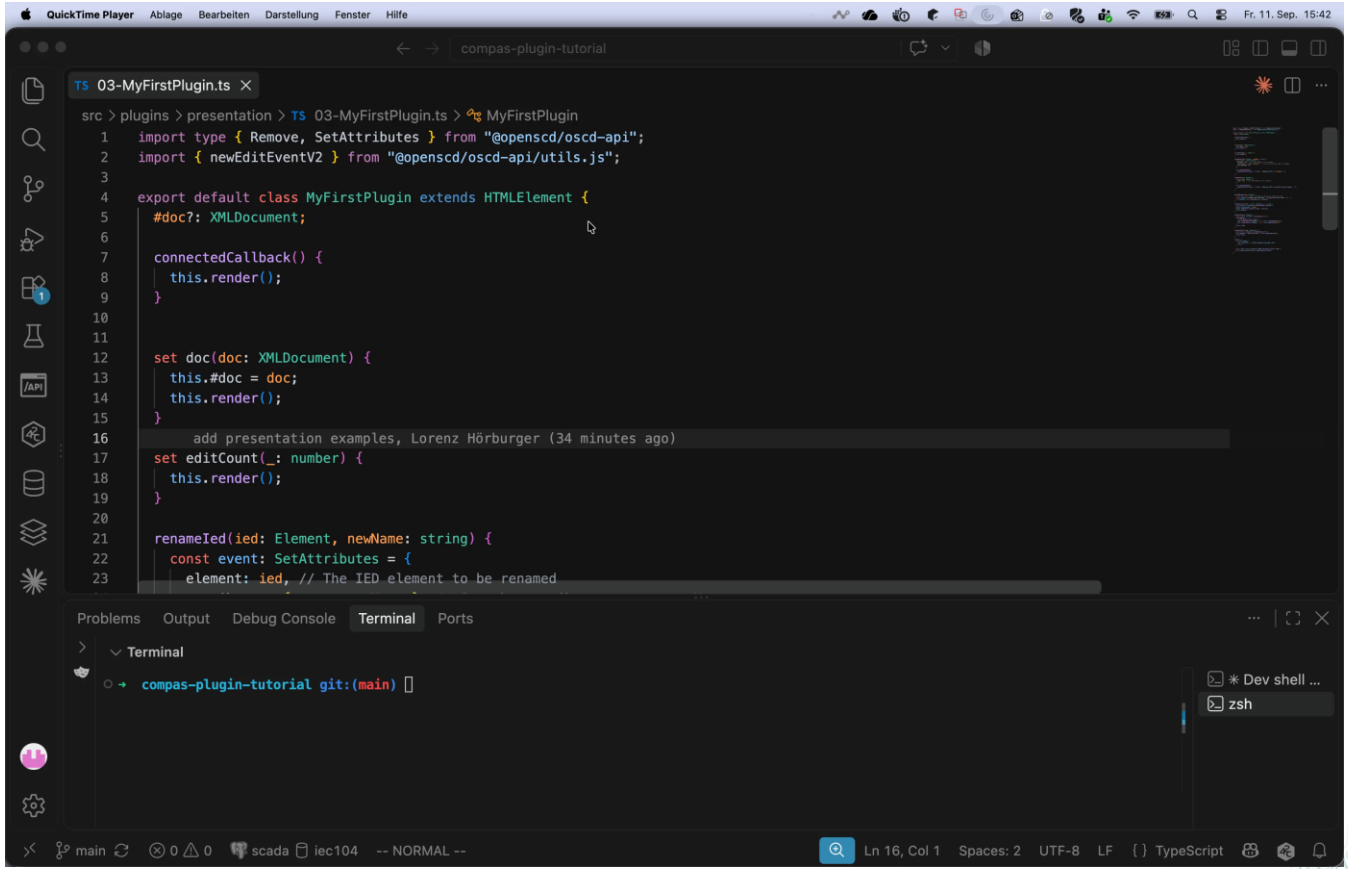
Host

GitHub Pages or other static web server
http://github.com/pages/index.js

Load

Load Plugin from URL




QuickTime Player Ablage Bearbeiten Darstellung Fenster Hilfe
Fr. 11. Sep. 15:42
compas-plugin-tutorial
03-MyFirstPlugin.ts
src > plugins > presentation > 03-MyFirstPlugin.ts > MyFirstPlugin
import type { Remove, SetAttributes } from "@openscd/oscd-api";
import { newEditEventV2 } from "@openscd/oscd-api/utils.js";
export default class MyFirstPlugin extends HTMLElement {
#doc?: XMLDocument;
connectedCallback() {
this.render();
}
set doc(doc: XMLDocument) {
this.#doc = doc;
this.render();
}
add presentation examples, Lorenz Hörburger (34 minutes ago)
set editCount(_: number) {
this.render();
}
renameIed(ied: Element, newName: string) {
const event: SetAttributes = {
element: ied, // The IED element to be renamed
Problems
Output
Debug Console
Terminal
Ports
Terminal
compas-plugin-tutorial git:(main)
Dev shell ...
zsh
main
0 0
scada iec104
-- NORMAL --
Ln 16, Col 1
Spaces: 2
UTF-8
LF
TypeScript

Choose Your Technology



JavaScript



LitElement



Svelte

Custom Element

OpenSCD Plugin

Now it's your turn!

https://github.com/ase-compas/compas-plugin-tutorial

Resources

Plugin API: github.com/openscd/oscd-api
CoMPAS Core: github.com/com-pas/compas-open-scd

Quality You Can Trust: Raising the Bar for SCL Engineering



LF ENERGY SUMMIT EUROPE 2026 | Develop the Modern Grid, Together

Same standard, different quality

Every SCL file claims IEC 61850 - but quality and focus vary widely.

1

Many sources, many quality levels

SCL files come from different vendors, tools and engineers - each focuses on different aspects.

2

Incomplete and inconsistent

Missing data, deviating naming, undocumented conventions - usually discovered late, in test or commissioning.

3

The standard alone is not enough

Schema and IEC 61850 checks pass, yet project- and utility-specific requirements stay unchecked.

With user-centric rules, quality can be established.

Your own conventions become automated checks - on every file, from day one.

Introducing our user defined validation feature

1

No coding required
Define your own validation rules directly in the Engineering Wizard

2

Errors caught early
Find SCL issues before they reach commissioning or the substation

3

Consistent conventions
Enforce project and utility-specific naming, structure and data rules

4

Reusable rule sets
Every engineer works to the same companywide quality standard

5

Actionable feedback
Clear guidance in the wizard speeds up review and rework cycles

6

Flexible and extensible
Adapt and extend rule sets as projects evolve



The Validation rule creation process

From plugin selection to an enforced quality gate - five steps inside the Engineering Wizard.

1

Select plugin in the EW

Open the plugin in the Engineering Wizard and create a new validation rule.

2

Set general parameters

Give the rule a name, description, severity and scope.

3

Define rule condition

Specify the logic the SCL data has to satisfy - no coding required.

4

Test run the validation

Run the rule against a sample file and review the reported findings.

5

Apply during EW process

Activate the rule set so every file is checked automatically.



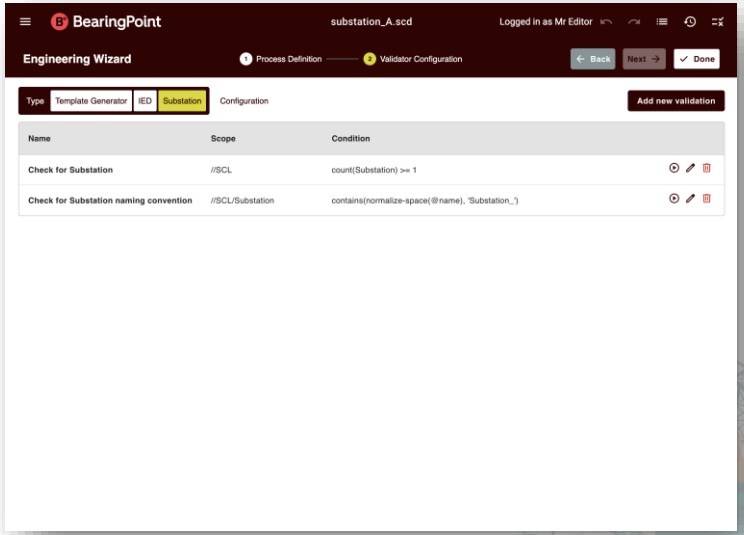
Select the plugin and create a new validation rule

1. **Select plugin**
2. General parameters
3. Rule condition
4. Test run
5. Apply in EW

STEP 1

Where every rule starts

- Open the relevant plugin inside the Engineering Wizard.
- Switch to the validation section and start a new user-defined rule.
- The rule is stored with the plugin, so it travels with the project.



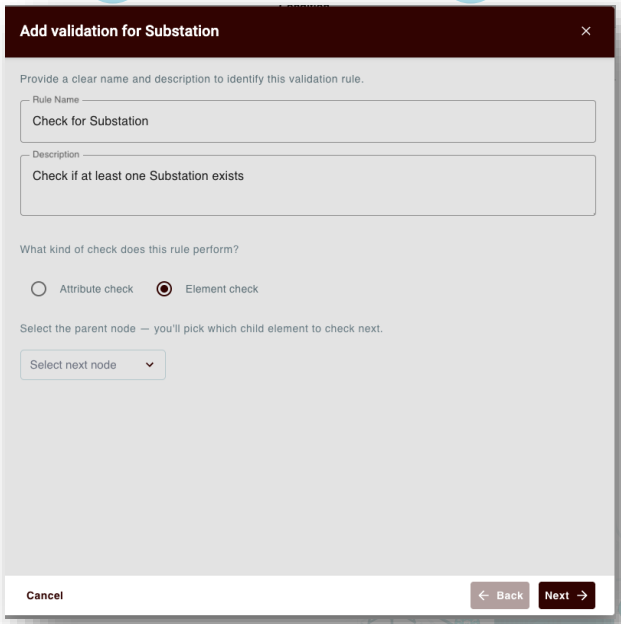
Set the general parameters

1 Select plugin — 2 General parameters — 3 Rule condition — 4 — 5

STEP 2

Make the rule understandable

– Give the rule a clear name and description so reviewers grasp the intent.
– Choose the severity level - error, warning or information.
– Define the scope: which SCL elements the rule applies to.



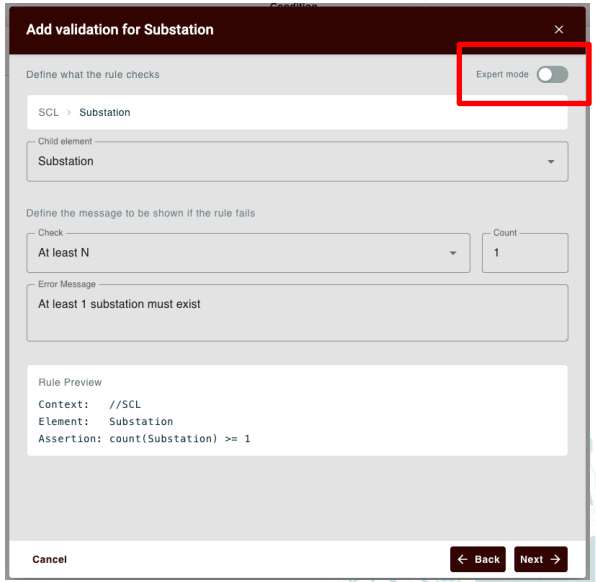
Define the rule condition

1 Select plugin — 2 General parameters — 3 **Rule condition** — 4 Test run — 5 Apply in EW

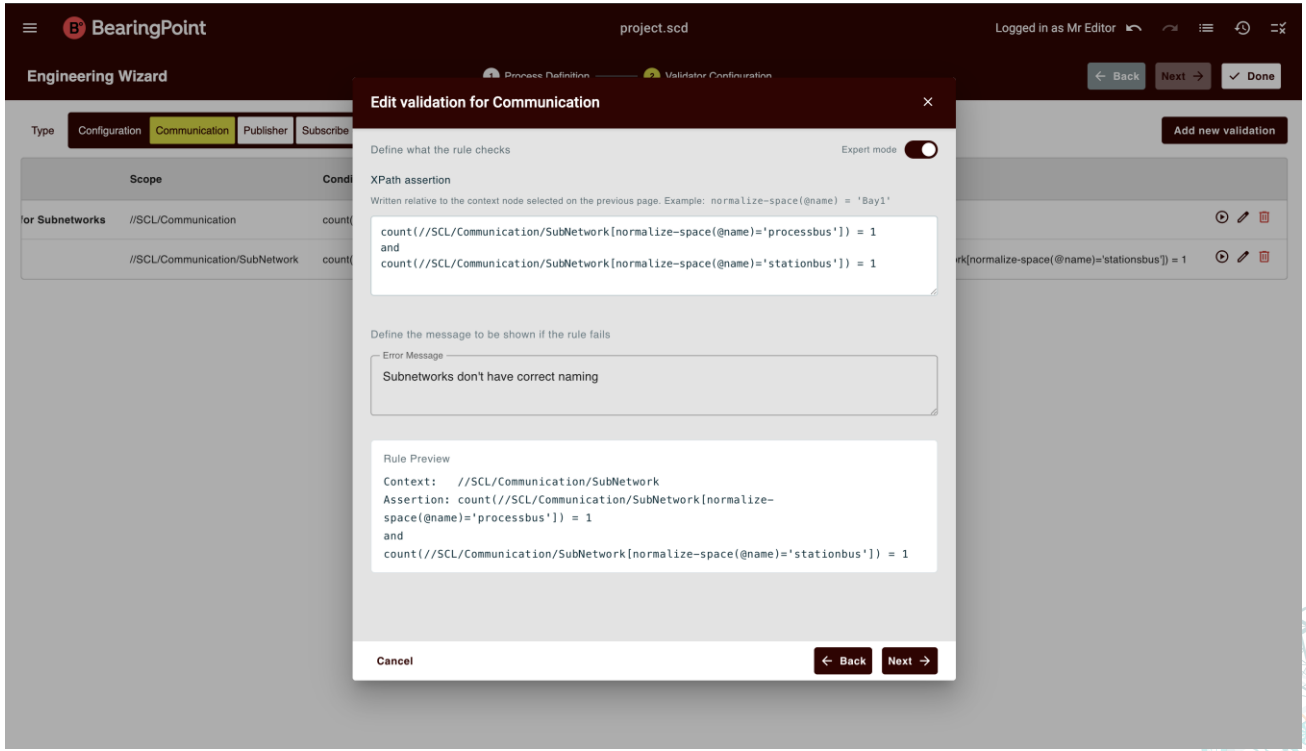
STEP 3

Turn a convention into logic

- Build the check logic directly in the wizard - no coding required.
- Combine attributes, values and patterns to express your naming or structure convention.
- Add a feedback message that tells the engineer exactly what to fix.



Deep Dive: EXPERT MODE!



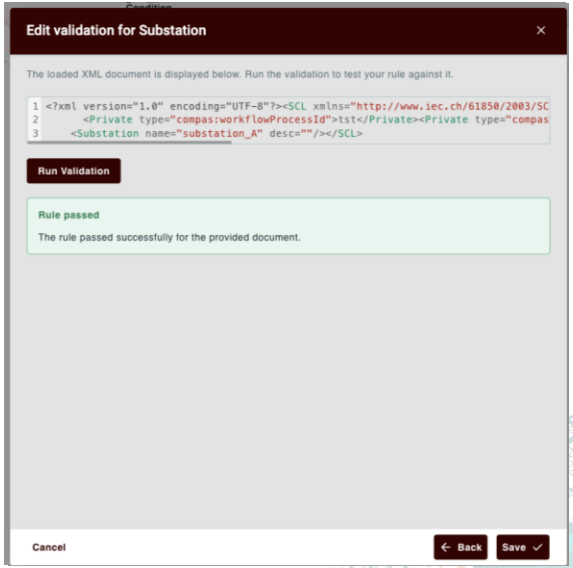
Test run the validation

1 Select plugin — 2 General parameters — 3 Rule condition — 4 Test run — 5 Apply in EW

STEP 4

Prove it works before it counts

- Run the rule against a sample SCL file straight from the wizard.
- Review the findings - failed and passed checks are both listed.
- Refine the condition until the results match your expectation.



Apply the rule during the EW process

1 Select plugin
2 General parameters
3 Rule condition
4 Test run
5 **Apply in your Engineering Process**

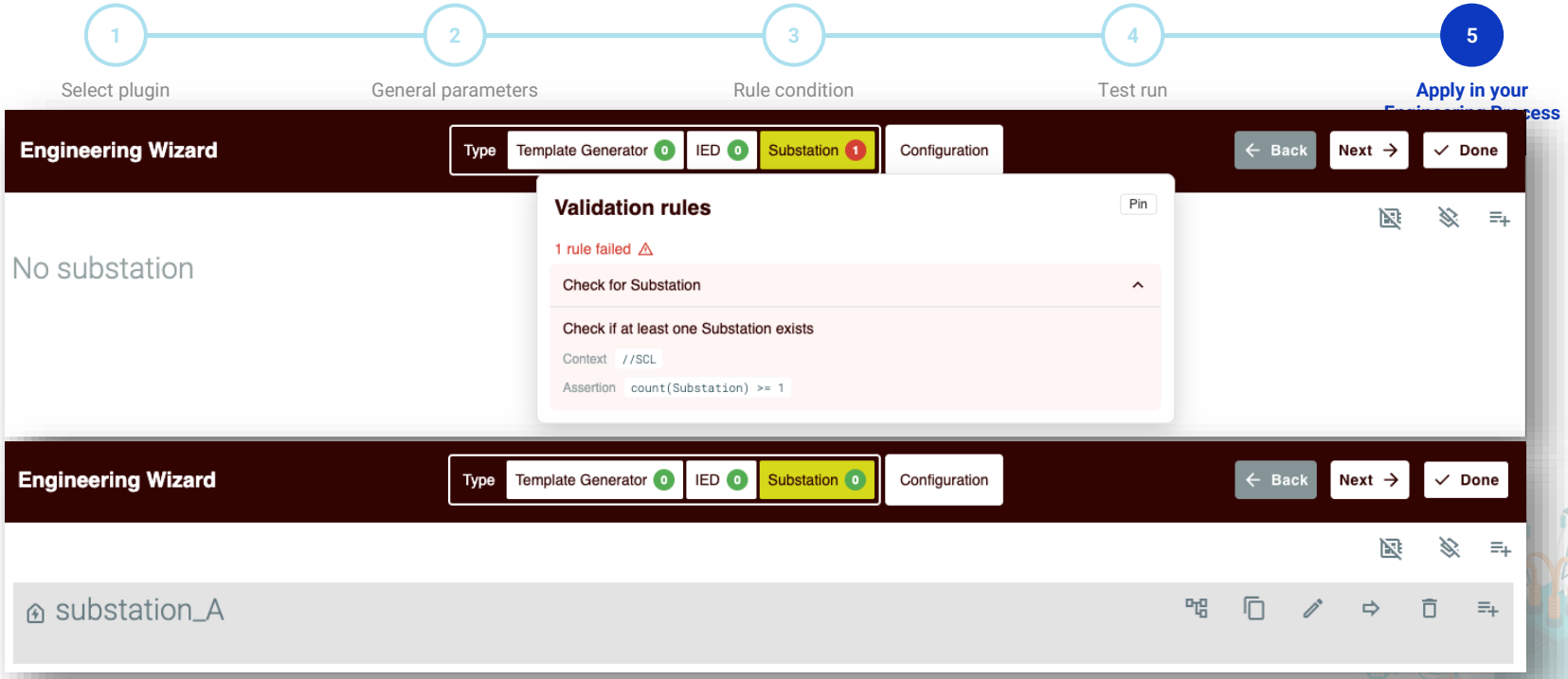
STEP 5

From rule to enforced quality gate

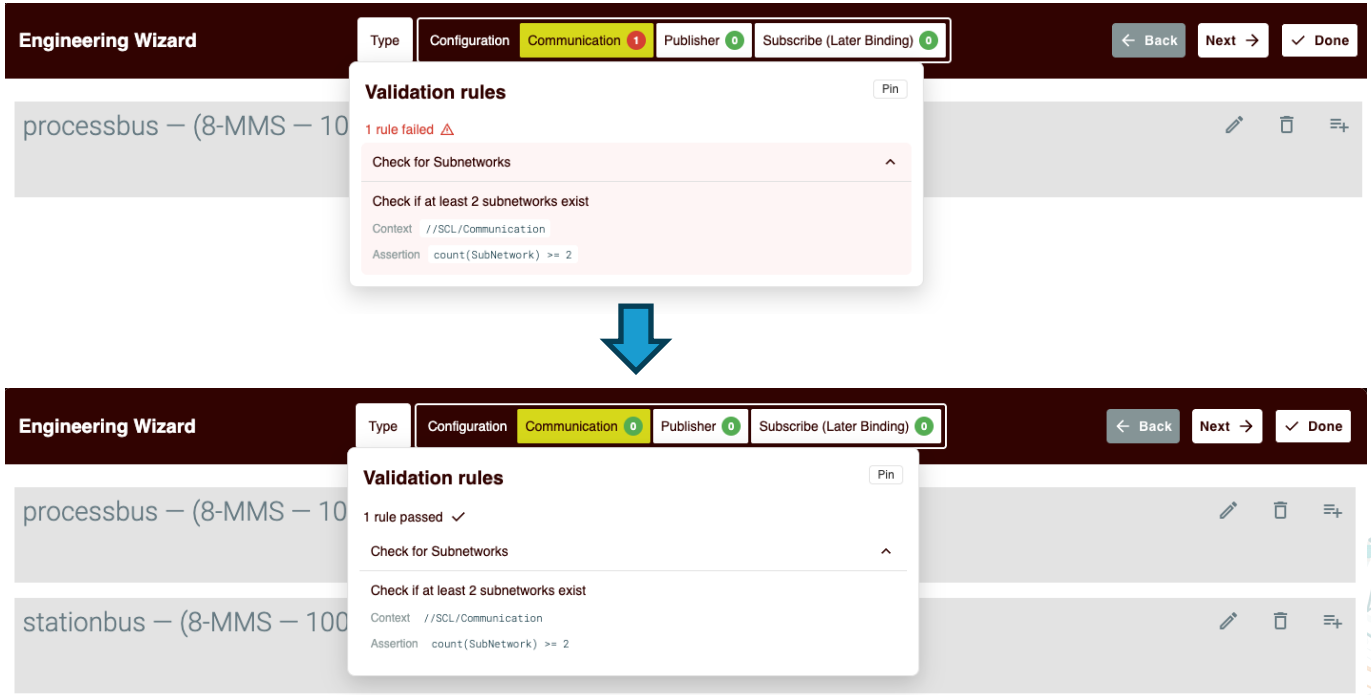
- Activate the rule set so it runs on every file in the project.
- Engineers receive direct, actionable feedback while they work.
- Deviations are caught early - not in test or commissioning.



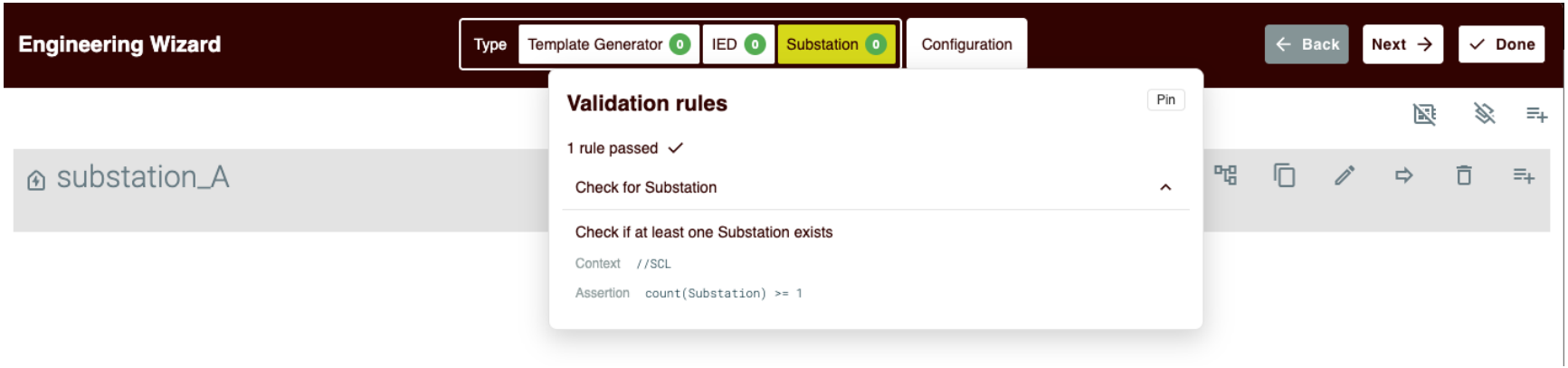
Apply the rule during the EW process (Substation)



Apply the rule during the EW process (SubNetworks)



Bonus Screenshot (Zeigen der passed regeln im popup)



Next Steps Validation Feature

AFTER THE SUMMIT

Update valdaition feature

The user-defined validation feature becomes a major update with expert mode and more.

NEXT

Global validation approach

Driven by community and user demand, reach out and help shape it.

OUTLOOK

OCL Support

Support for OCL-based rules - required to express complex SCL relationships.



Where CoMPAS stands today

A lot of ground covered in a short time – and real momentum behind it

From prototype to platform

Core, plugins and the Engineering Wizard have matured into a toolchain that is used in real projects

A community that keeps growing

Utilities, vendors and developers contribute, review and shape the roadmap together

Short cycles, visible progress

Validation, UX and extensibility move forward release after release

JOIN US

Slack
lfenergy.slack.com – #compas

GitHub
github.com/com-pas

LF Energy
lfenergy.org/projects/compas

Get involved: try the wizard, bring your rules and use cases, open an issue – or simply come and talk to us.

Thank you for your attention

Questions?

