Wizarding Addon

The Wizarding Addon is responsible for displaying wizards in OpenSCD. Wizarding could be made in many ways. Libraries allow plug-in editors to choose their own experiences. Plug-in authors can use the OSCD wizard component to make life easier.

Principles:

- Plug-in author is responsible for how to display dialog's/wizards
- Allow multiple types of wizards
- Wizards should be framework independent
- The wizard initiator decides what kind of wizards are displayed

Scope of wizarding

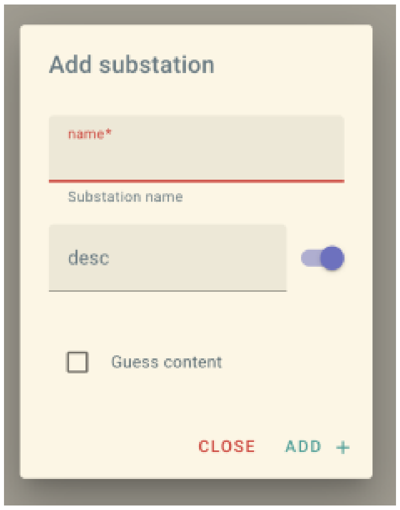
Easy wizarding allows to display things and edit SCL files. The idea is/was to re-use existing SCL edits

Open Question: What is a wizard?

Some wizards that are could be frequently used:

Form wizard

Webforms to enter/modify information.

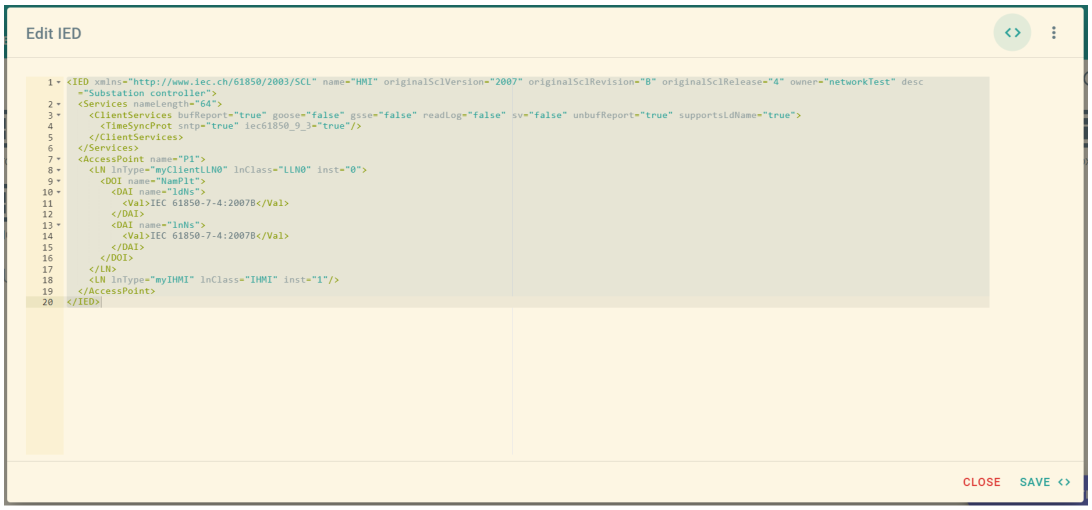


SCL wizard

Can generate wizards based on the SCL (XSD) schema (not build yet). This allow to change the wizard more easily once the 61850 XSD changes.

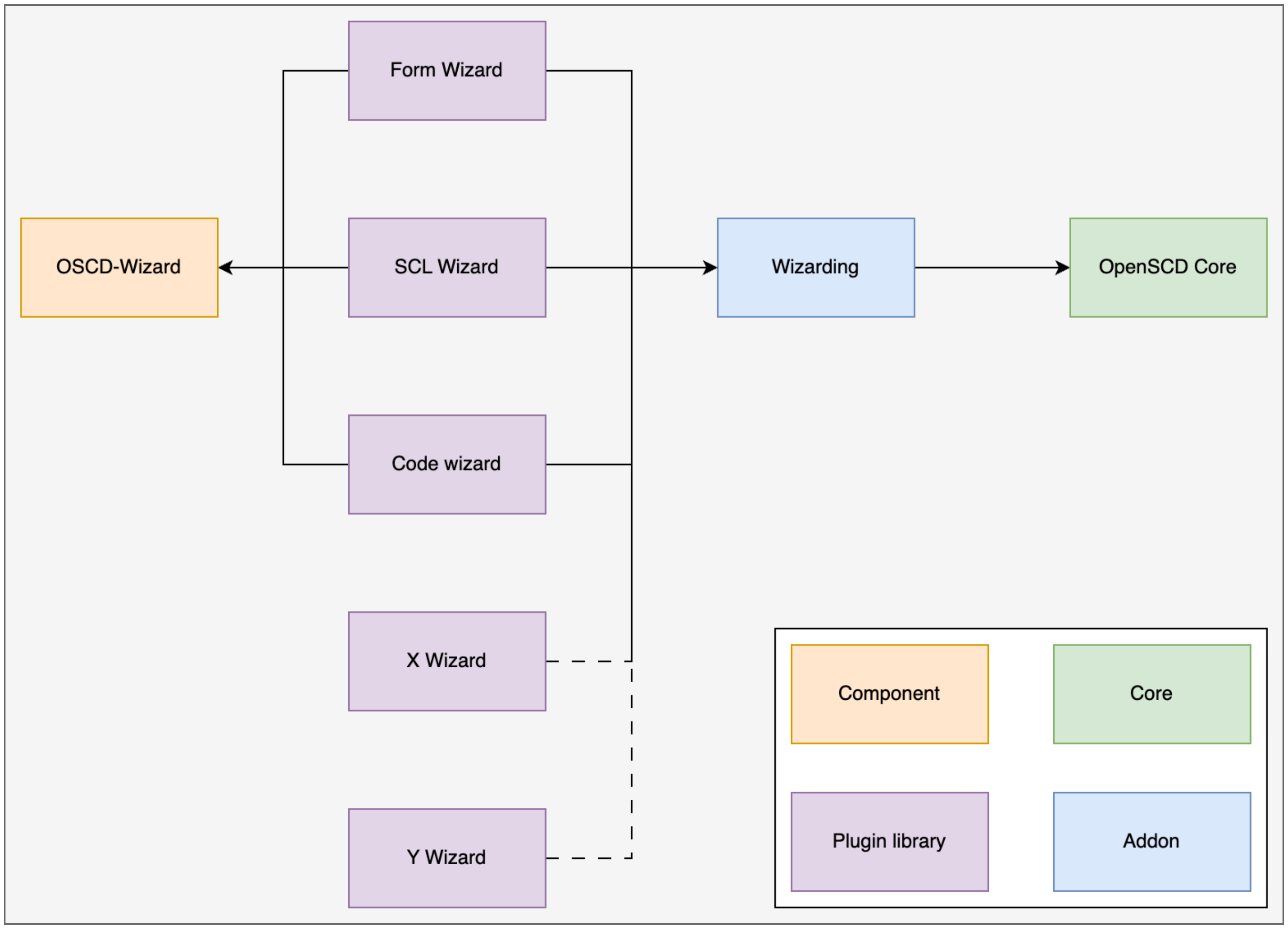
Code wizard

Could be used to display plain SCL files



```xml
<IED xmlns="http://www.iec.ch/61850/2003/SCL" name="HMI" originalSclVersion="2007" originalSclRevision="B" originalSclRelease="4" owner="networkTest" desc
  ="Substation controller">
  <Services nameLength="64">
    <ClientServices bufReport="true" goose="false" gsse="false" readLog="false" sv="false" unbufReport="true" supportsLdName="true">
      <TimeSyncProt sntp="true" iec61850_9_3="true"/>
    </ClientServices>
  </Services>
  <AccessPoint name="P1">
    <LN lnType="myClientLLN0" lnClass="LLN0" inst="0">
      <DOI name="NamPlt">
        <DAI name="ldNs">
          <Val>IEC 61850-7-4:2007B</Val>
        </DAI>
        <DAI name="lnNs">
          <Val>IEC 61850-7-4:2007B</Val>
        </DAI>
      </DOI>
    </LN>
    <LN lnType="myIHMI" lnClass="IHMI" inst="1"/>
  </AccessPoint>
</IED>
```

Overview of the different aspects



Conserquences

Freedom comes with responsibilities (e.g. you can add pacman as plug-in).

Technical working

The Wizarding Addon listens to the `oscd-wizard` CustomEvent.

The `oscd-wizard` events contains the `OscdWizard` reference.

oscd-wizard

```
export interface OscdWizardEventDetail {
    wizard: OscdWizard;
}

export interface OscdWizardEvent extends Event {
    detail: OscdWizardEventDetail;
}
```

The `OscdWizard` must be an HTMLElement that contains the `open() and close() functions.`

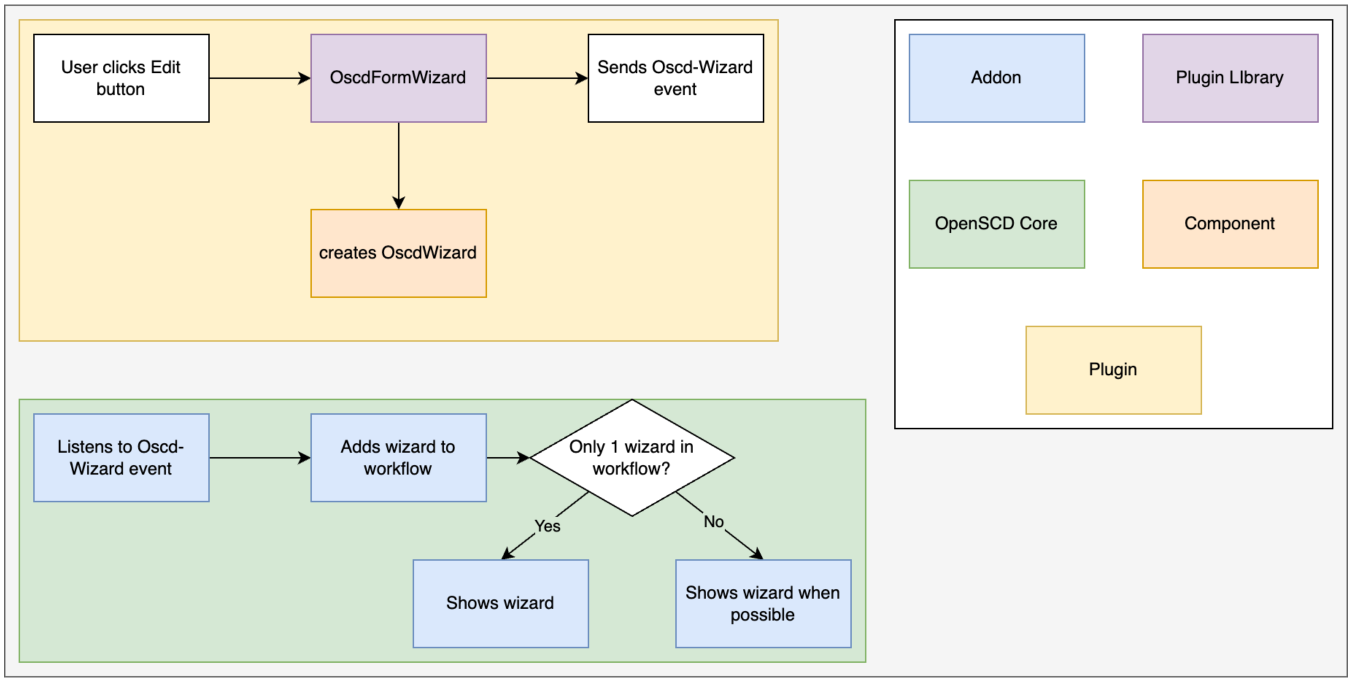
`The OscdWizard can be viewed below`

OscdWizard

```
export type OscdWizard = HTMLElement & { open(): Promise<void>, close(): Promise<void> }
```

If you want to create your own wizard, you ***must*** adhere to this API.

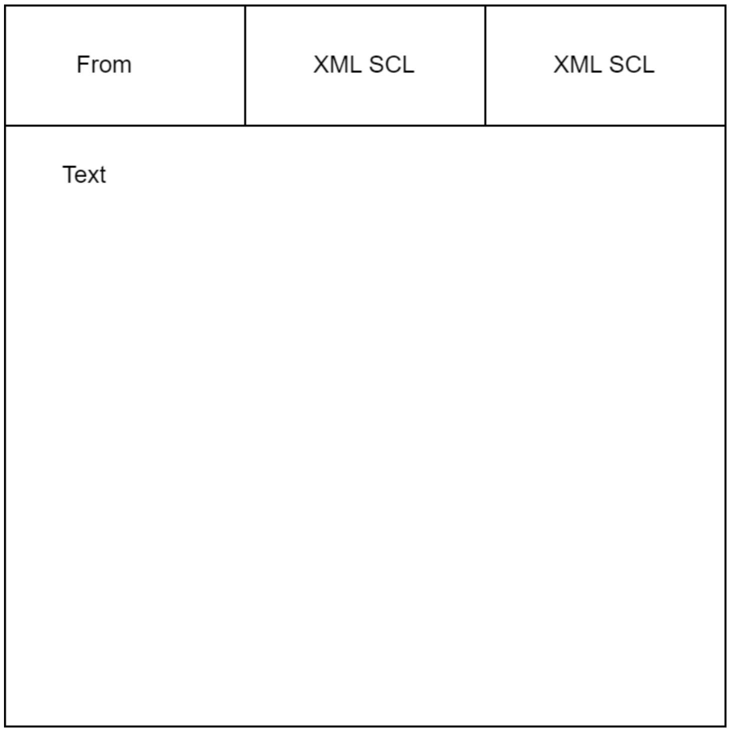
Wizarding flows from OpenSCD-Core plug-in en OpenSCD-Core plug.



Alternatives

To reduce the complexity, it is also an option to leaving the displaying/editing of the SCL part fully up to the wizard library. This allows users to switch between wizards (e.g. plain XML vs form style) to edit a specific SCL element.

The idea: the wizard use an SCL element to render the form elements. The wizard API is responsible for the generation the wizard (thus look and feel).



Pro's:

One single API for all wizards

Might reduce complexity

Give distributors and end-users the freedom to chose the right wizard-options/styles

Con's:

End-user could be overwhelmed by choices

The wizard API has a lot of functions to handle all situations

Limited freedom for the wizard initiator (no edits outside SCL elements)

Might be limited to editing

Questions:

Is it possible to add other namespace attributes?

Can you edit multiple SCL elements together?

Other alternative: No standardized Wizarding

Every plug-in authors will/can write it's own solution

Pro's: Complete freedom

Con's: Every plug-in author needs to reinvent the wheel in every plug-in.