Modeling of Substation PACS using IEC61850 and IEC61499

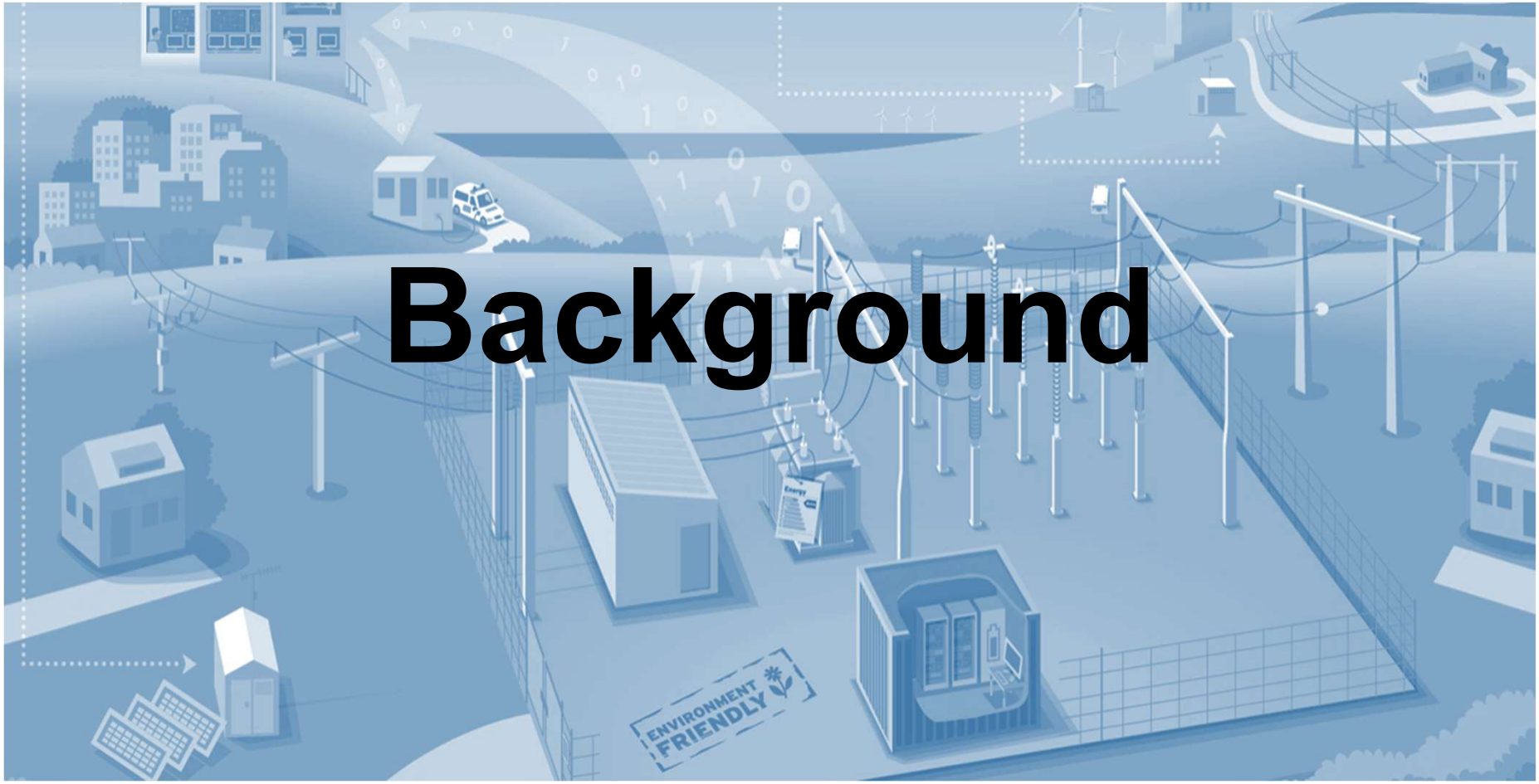
Layout, Protection and control functions, and Communications

Francisco de Lima, Yiming Wu

Master thesis project, 2019-05-20, Stockholm

Agenda

1. Background
2. Challenges and Potentials
3. Integration of IEC61850 and IEC61499
4. Case studies
5. Result and conclusion

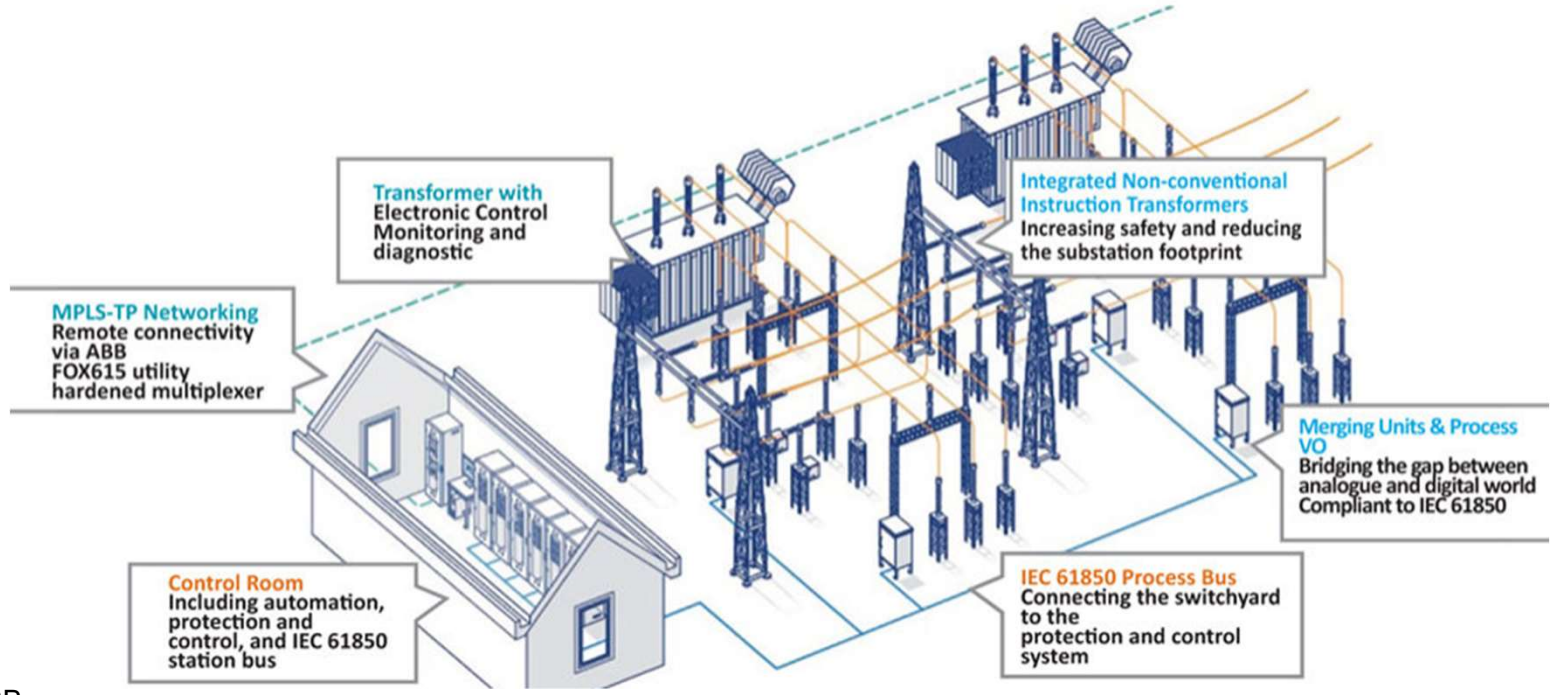


Background

Digital Substation PACS

- **IED (protection relay)**
  - Protection functions
  - Selectivity
- **MU**
  - Sampling rate
  - SAMU
  - NCIT
- **RTU**
- **Process bus**
  - SV/GOOSE
- **Station bus**
  - MMS/GOOSE

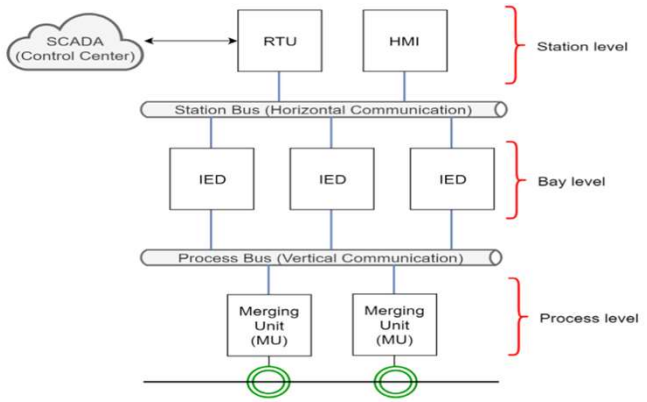
Digital Substation PACS

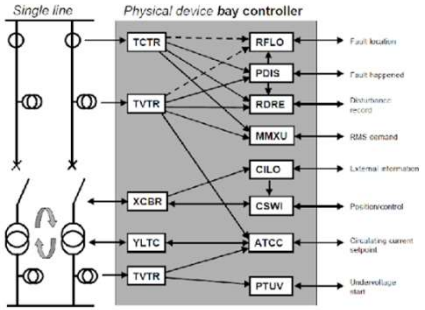


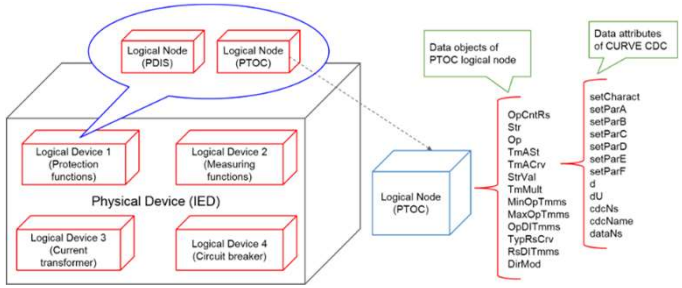
Source: ABB
https://new.abb.com/substation-automation/systems/digital-substation/bridging-the-gap

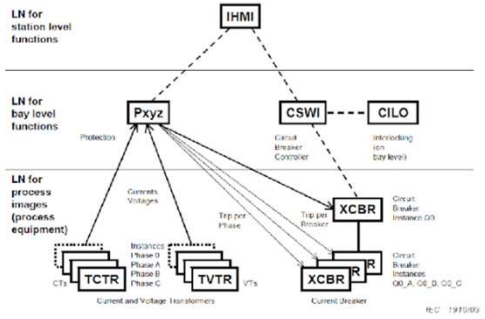
Modeling of Substation PACS

- **IEC61850**
  - Architecture of the substation
  - Information modelling
  - Communication protocols
- **What is missing?**
  - Description of functionality
  - Information flows between LNs



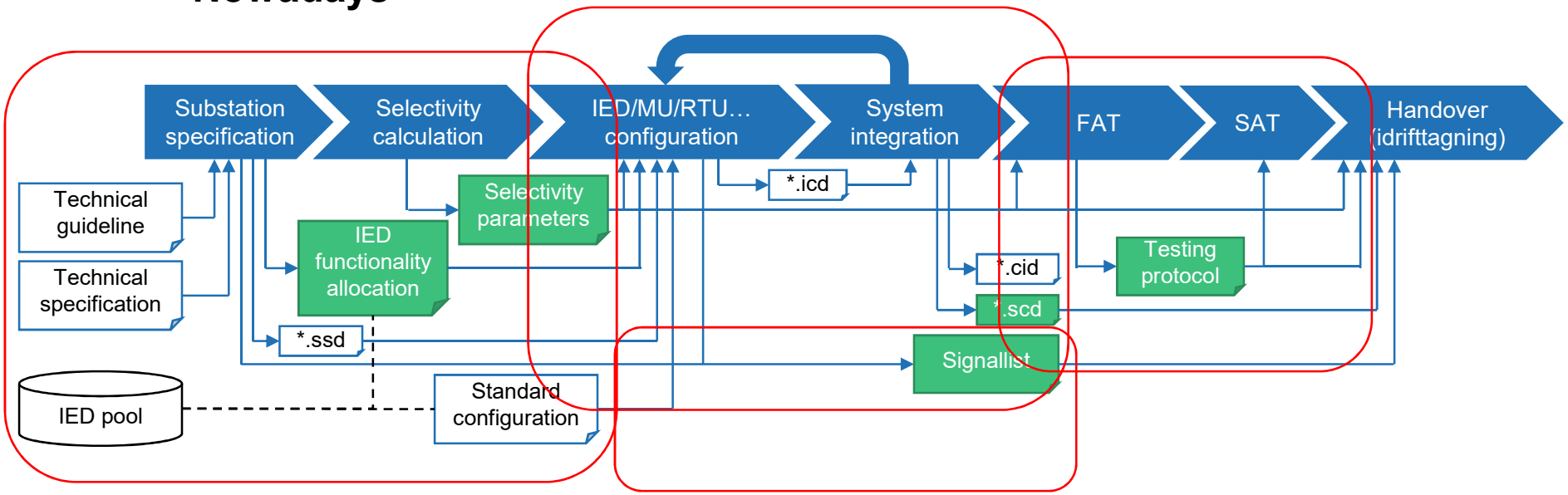






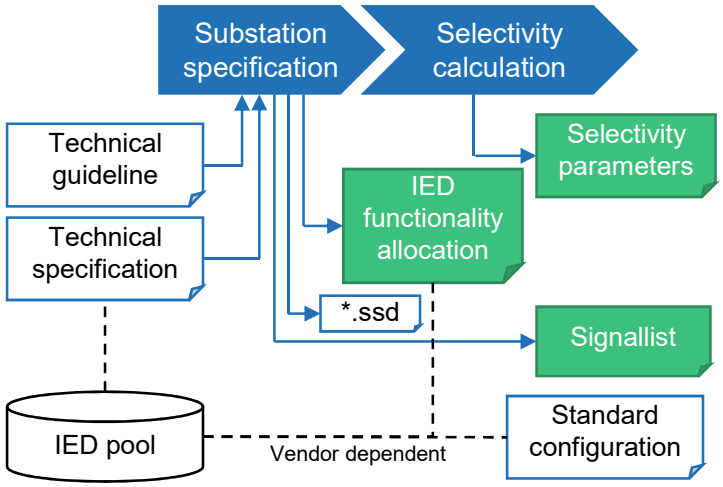
Substation Design and Engineering Process

- **Nowadays**



Substation Design and Engineering Process

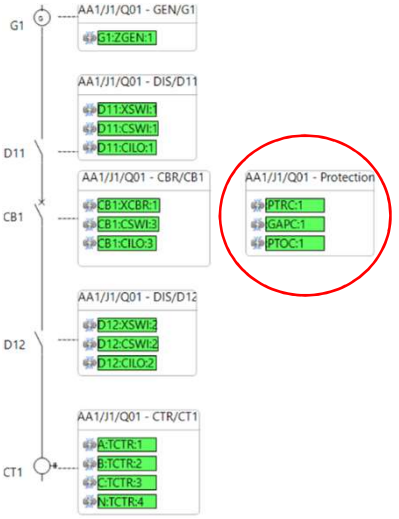
- **Nowadays**

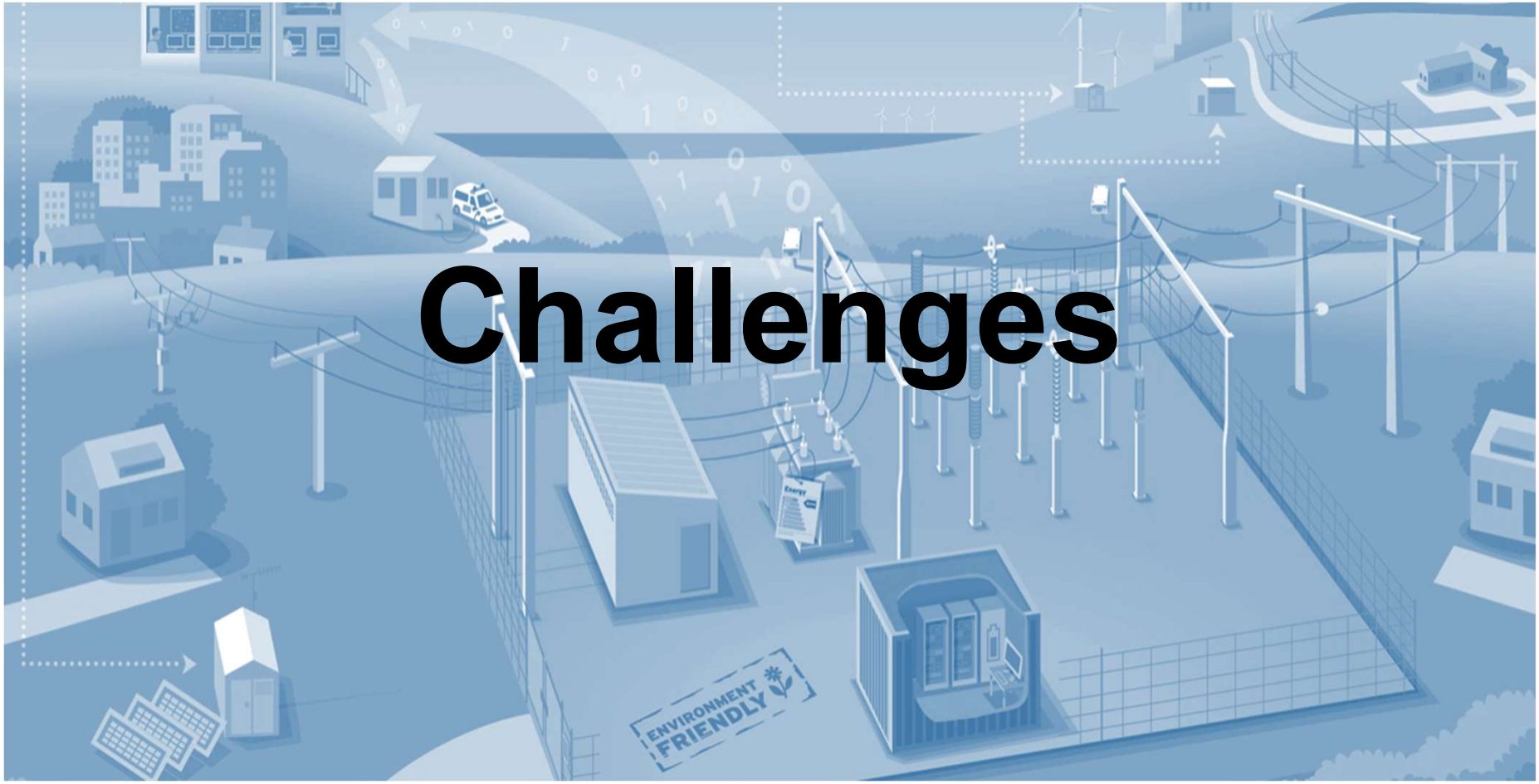


VTR

6	**Ledningsskydd 50-6 kV**
6.1	Oriktat 3-fas Överströmsskydd (3I>)
~~6.2~~	~~Riktat 3-fas Överströmsskydd (3I> →)~~
~~6.3~~	~~Reservskydd (parallellskydd)~~
~~6.4~~	~~Obalans/Fasavbrottsskydd (I2/I1>)~~
6.5	Riktat Jordströmsskydd (I0> →)
~~6.6~~	~~Transientmätande Jordfelsskydd (I0> → IEF) (Wischer)~~
~~6.7~~	~~Nollföljdsspänningsskydd (Uo>steg1)~~
~~6.8~~	~~Uo> Undertidsfunktion (UTT)~~
6.9	Distansskydd (Z>)
~~6.10~~	~~Längsdifferentialskydd (3Id>)~~

Technical specification

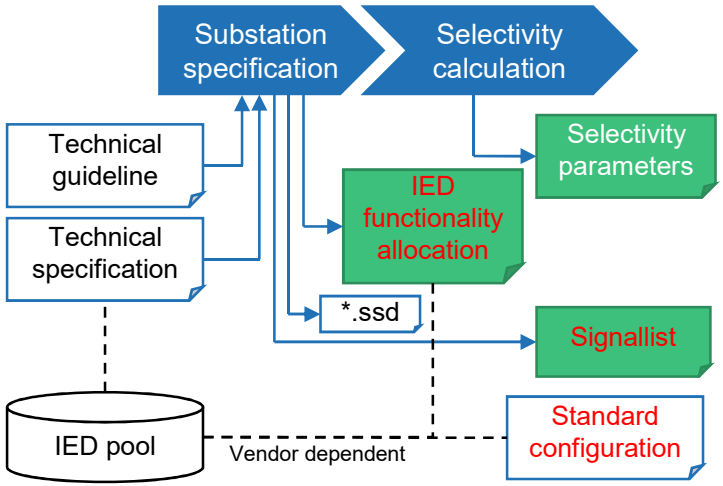




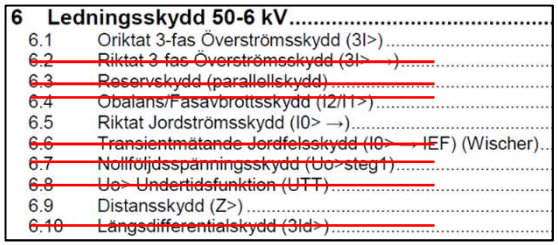
Challenges

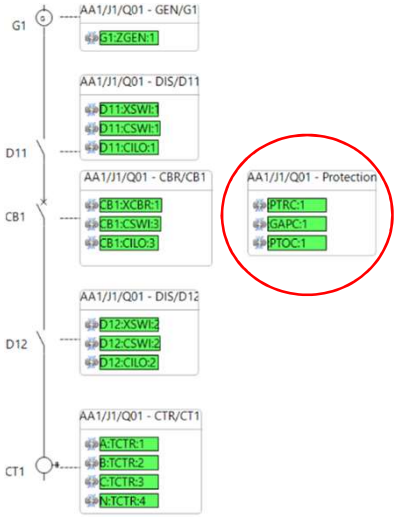
Identified challenges

- **Restrictions due to vendor dependent**



VTR





Technical specification

Function	SIEMENS	ABB
JSR	DNSR_PTOC	DEFLPTOC
JSR Trans.	DINT_PTOC	INTRPTEF
I>	ID_PTOC	DHLPTOC

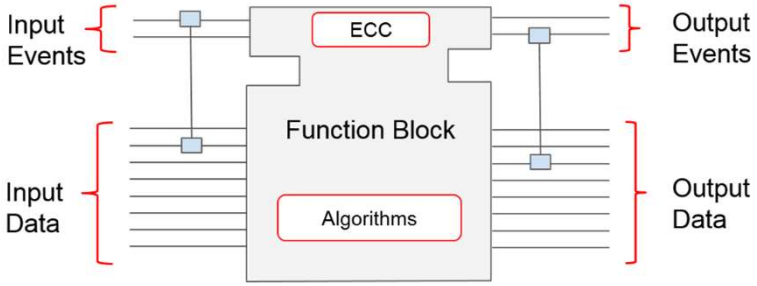
Identified challenges

- **IEC61850**
  - Lack of functionality description of LN
  - Lack of information flow within IED

- **Vendor dependent**
  - Functionality
  - Interchangeability (Functional level)
  - Hardware dependent
  - Naming convention
  - Require comprehensive knowledge of vendor’s product

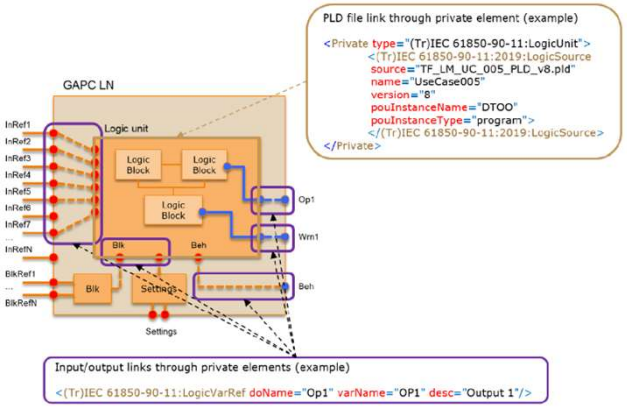
Identified challenges

- **IEC61499**
  - Functionality description
    - Further developed based on IEC61131-3 (RTU560 PLC)
  - Integrate multiple IEC61131-3 algorithms
  - Simple communication protocol
  - XML based
- **What is missing?**
  - Communication protocol is less capable
  - No information modeling
- **IEC61499, a possible solution??**

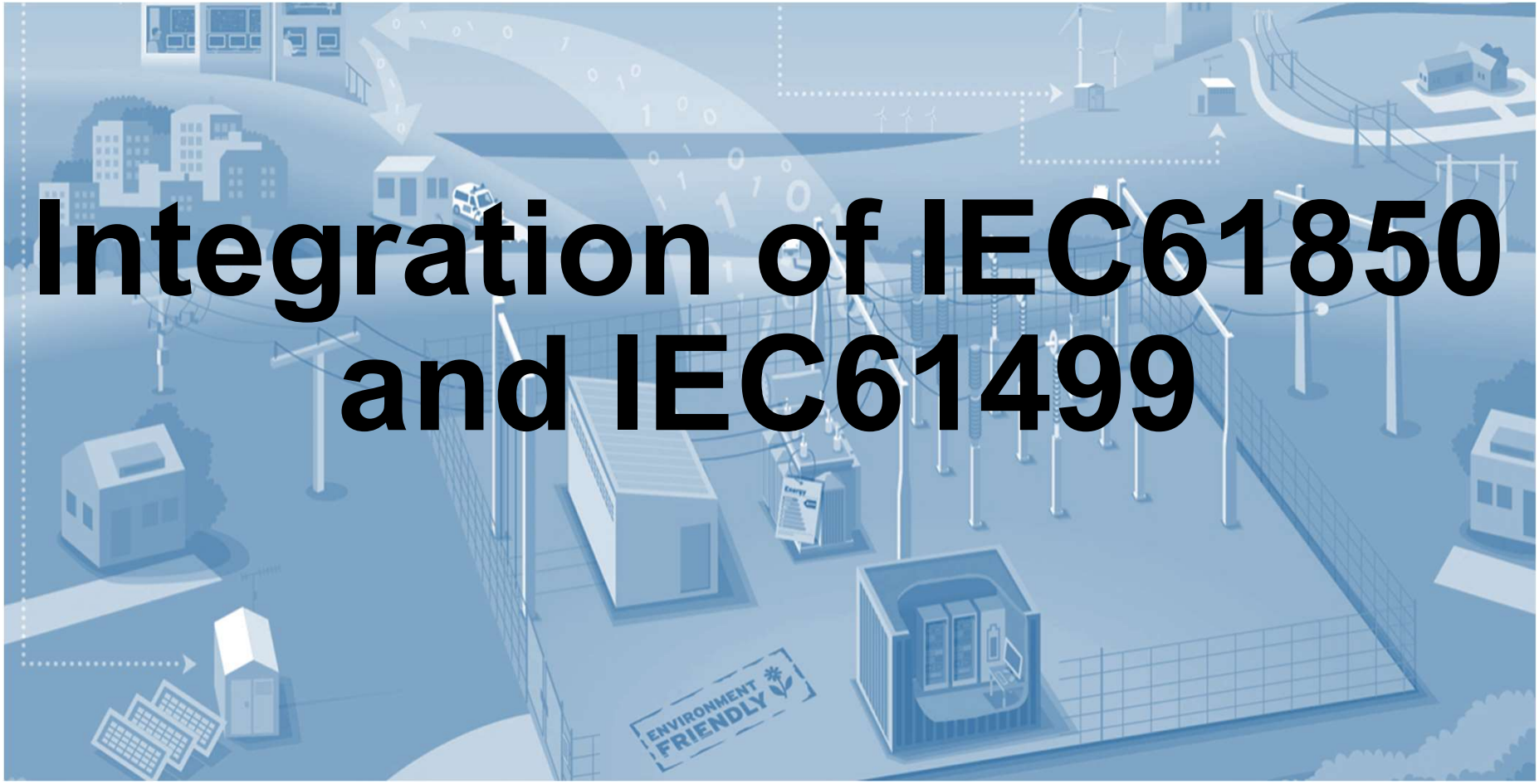


Modeling of Substation PACS

- **Integration of IEC61850 and IEC61499**
  - By using functionality description from IEC61499 to fill the gap identified from IEC61850
- **IEC61850-90-11**
  - Recommendation of integrate IEC61131-3 into IEC61850



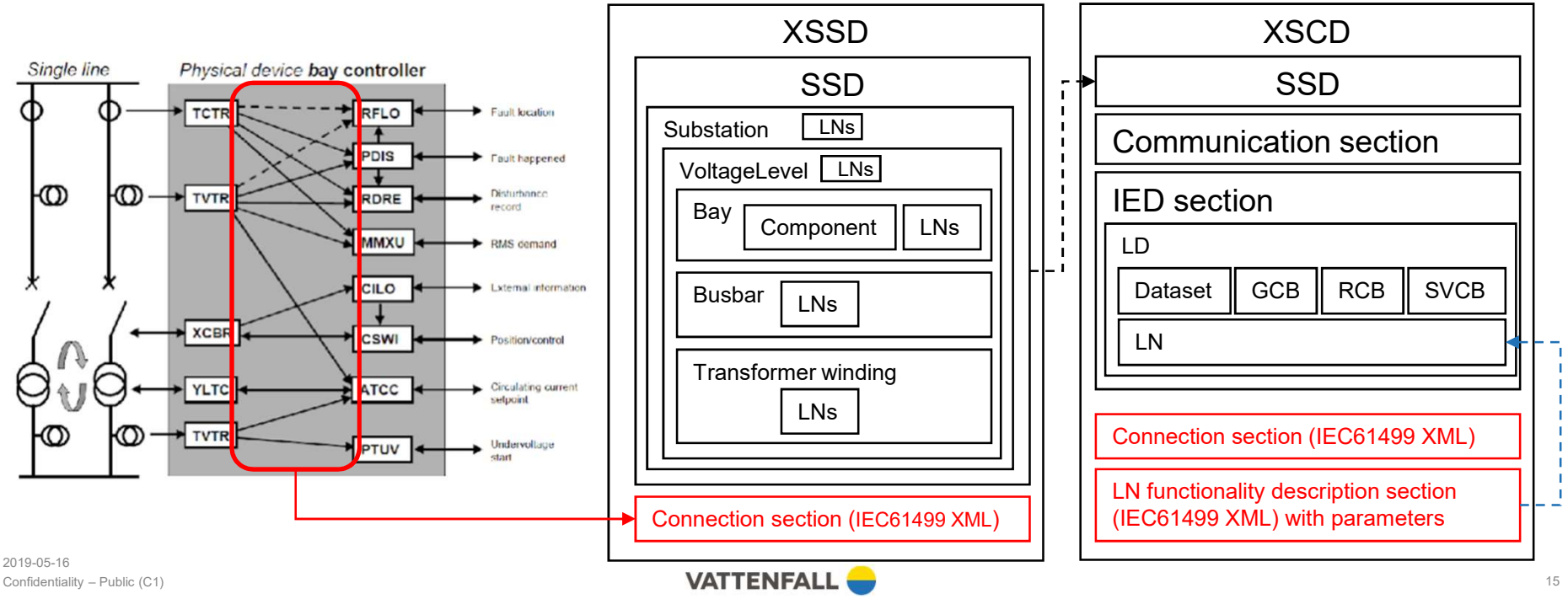

```
<LN desc="" lnType="GAPC_TDV" lnClass="GAPC" inst="1" prefix="TDV">
...
the full LN instance information
...
</LN>

<LNodeType desc="" id="GAPC_TDV" lnClass="GAPC">
        <Private type="(Tr)IEC 61850-90-11:2019:LogicUnit" >
        <(Tr)IEC 61850-90-11:2019:LogicSource source="TF_LM_UC_005_PLD_v8.pld" name="UseCase005"
        version="8" pouInstanceName="DTOO" pouInstanceType="program">
                <(Tr)IEC 61850-90-11:2019:LogicVarRef doName="InRef1" varName="POSA1" desc=""/>
                ...
                <(Tr)IEC 61850-90-11:2019:LogicVarRef doName="Op1" varName="OP1" desc=""/>
        </(Tr)IEC 61850-90-11:2019:LogicSource>
        </Private>
...
the full LNodeType information
...
</LNodeType>
```



Integration of IEC61850 and IEC61499

Integrate LN function block in SCL file

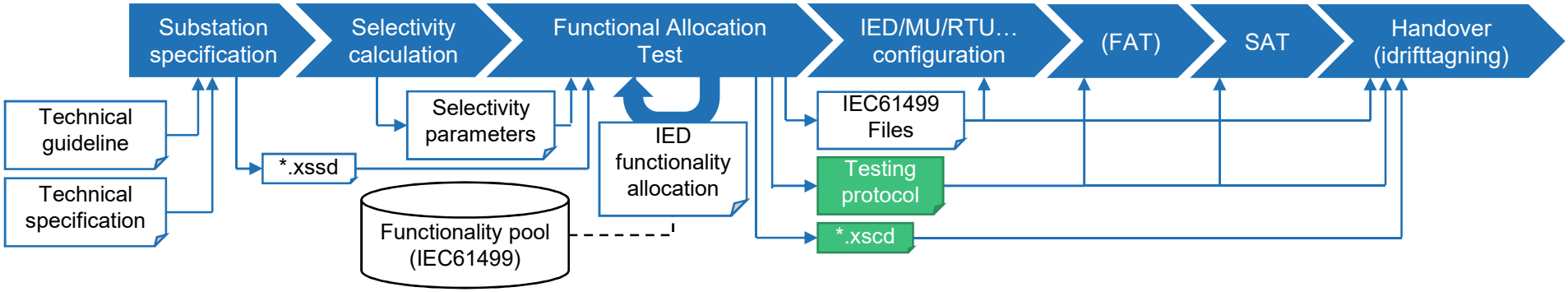
- **Introduce of eXtended SSD (XSSD) and eXtended SCD (XSCD)**



Comparison?

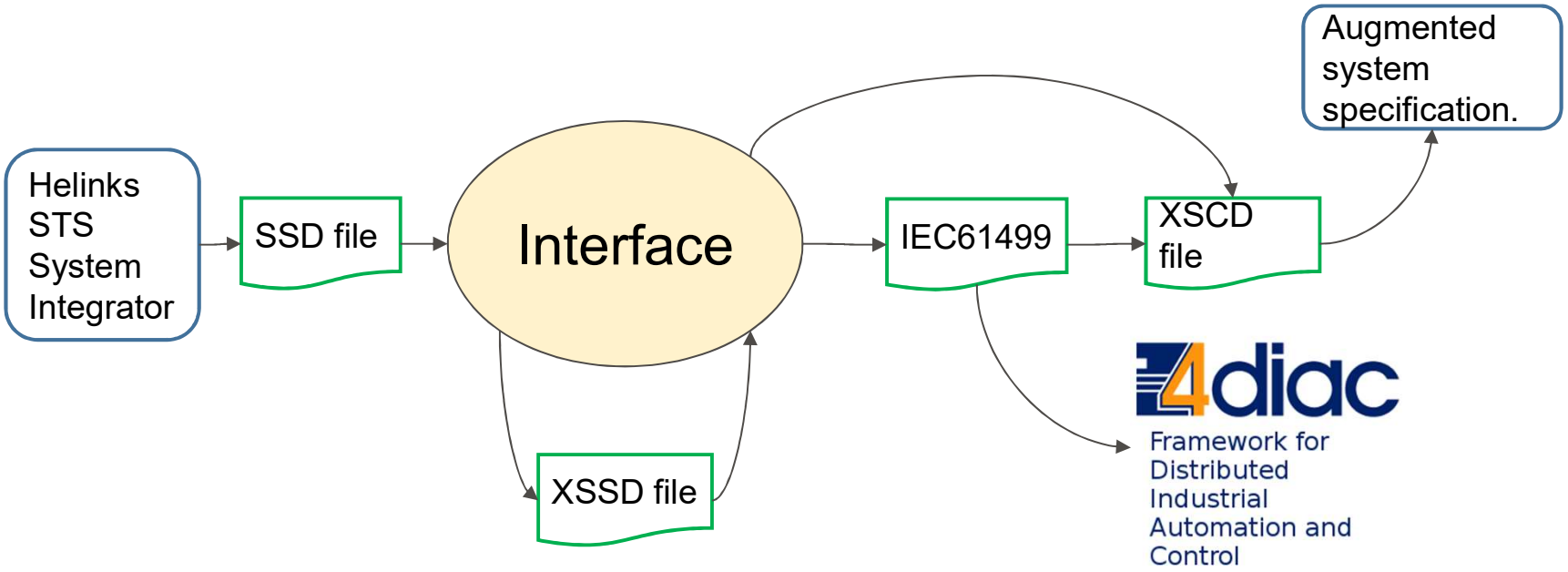
A new top-down substation PACS design and engineering process

- **Process**

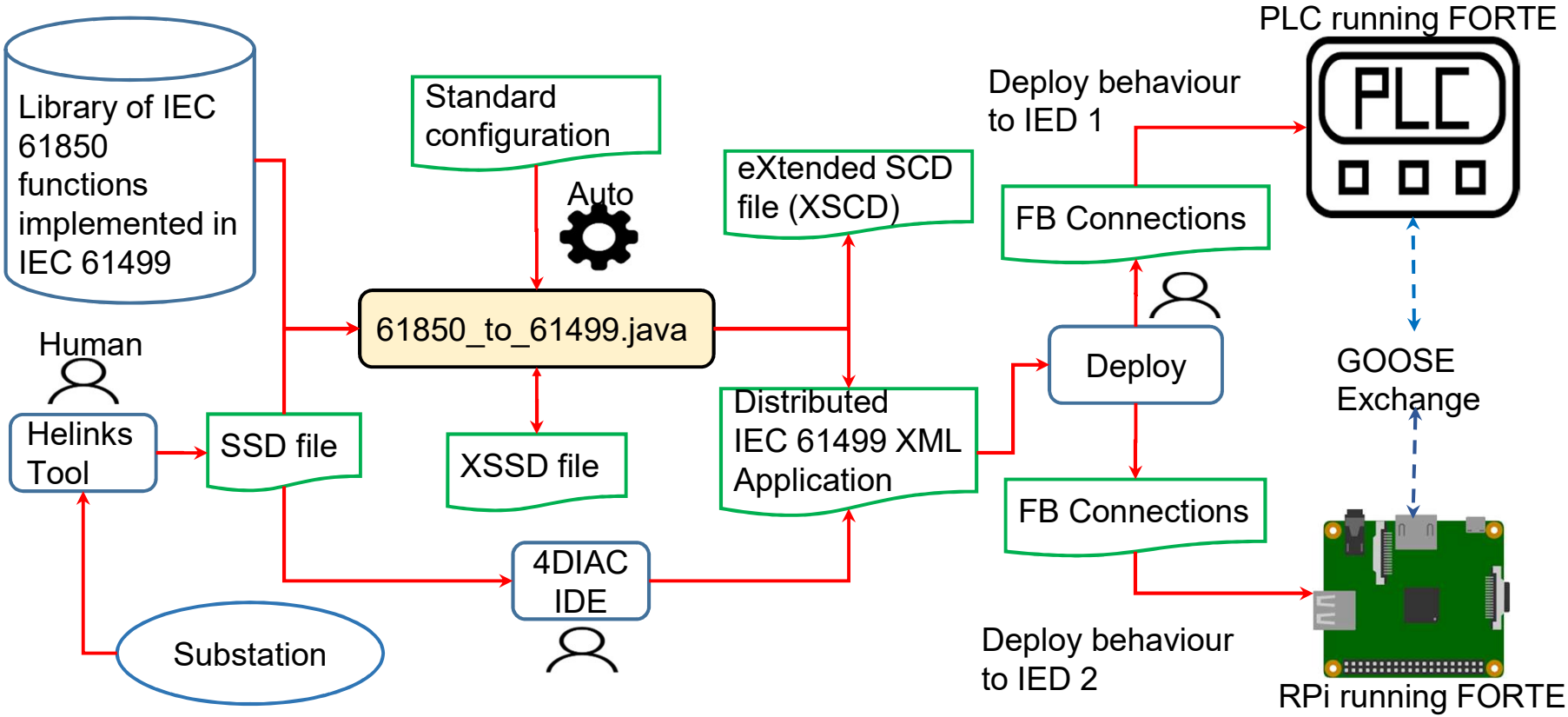


- **Benefits**
  - Interchangeability at functional level
  - Vendor independent hardware platform
  - Comprehensive documentation: layout, function description, information flow, parameters
  - Cost saving at the design phase and maintenance phase

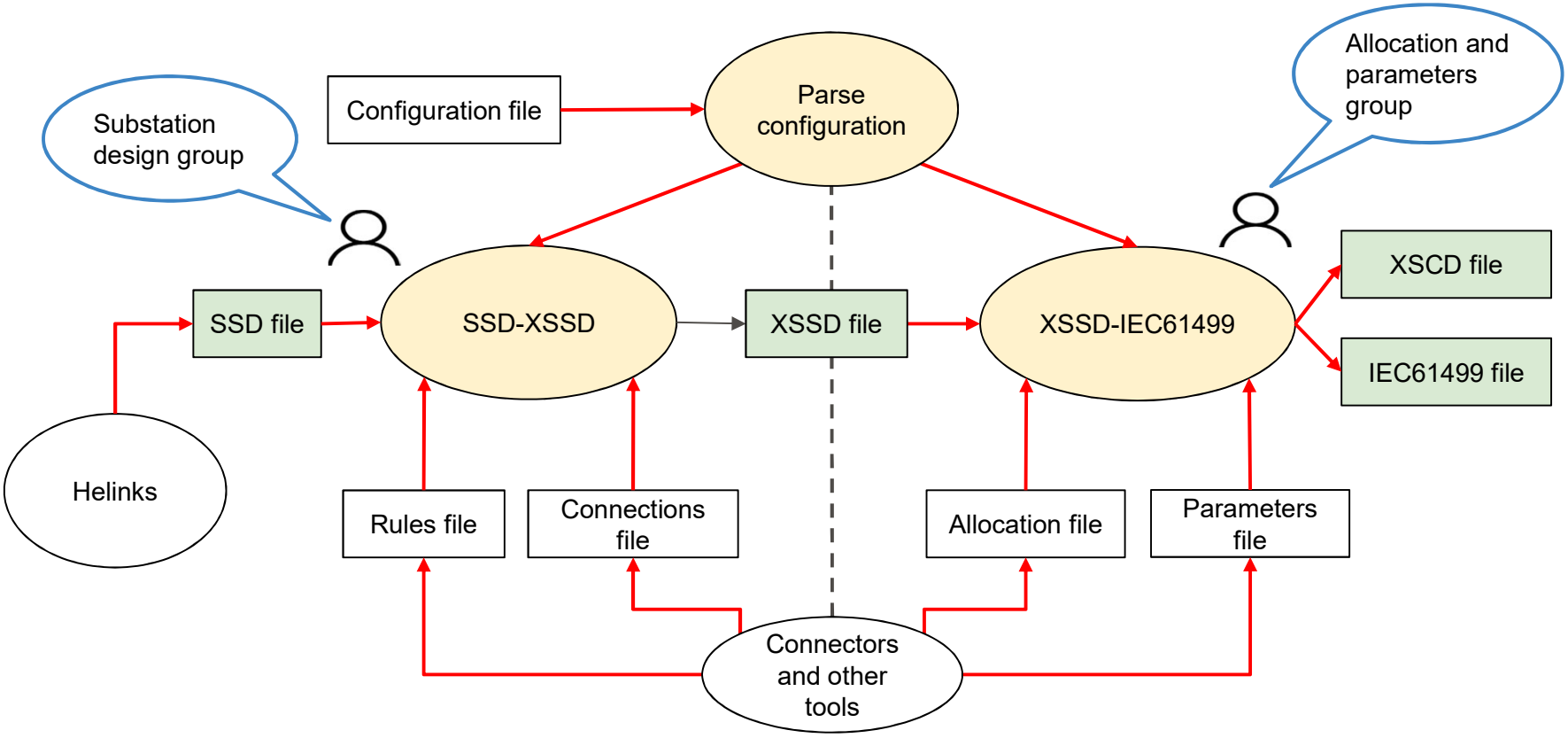
The Interface - General Overview



The Interface - Process Diagram



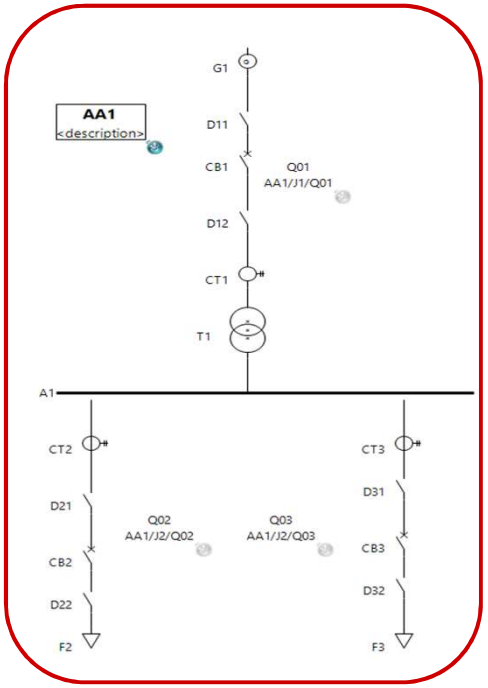
The Interface - General Workflow

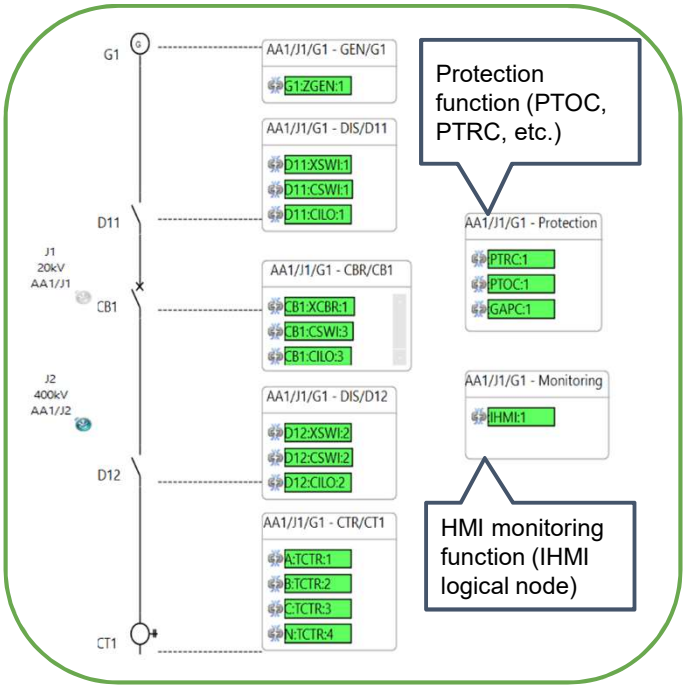


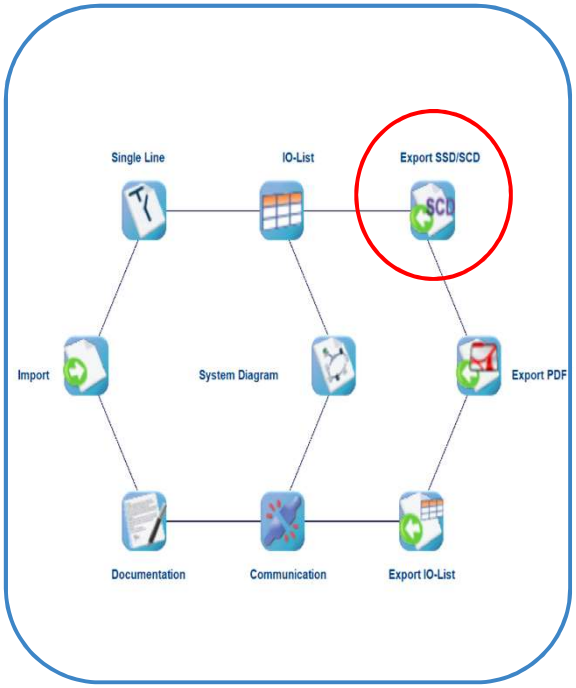
VATTENFALL

The Interface - Substation Specification

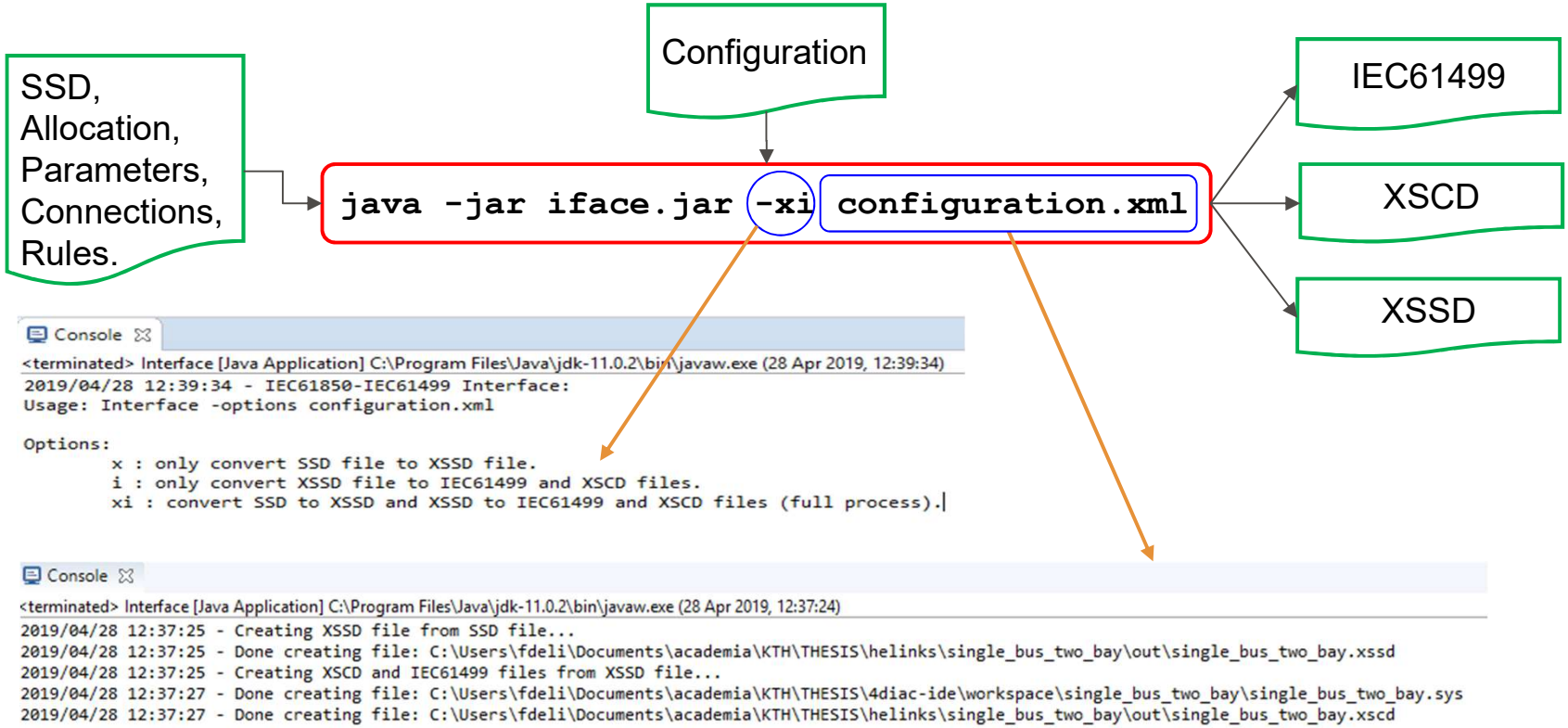
(1) Create substation → (2) Specify new functions and logical nodes → (3) Export SSD file



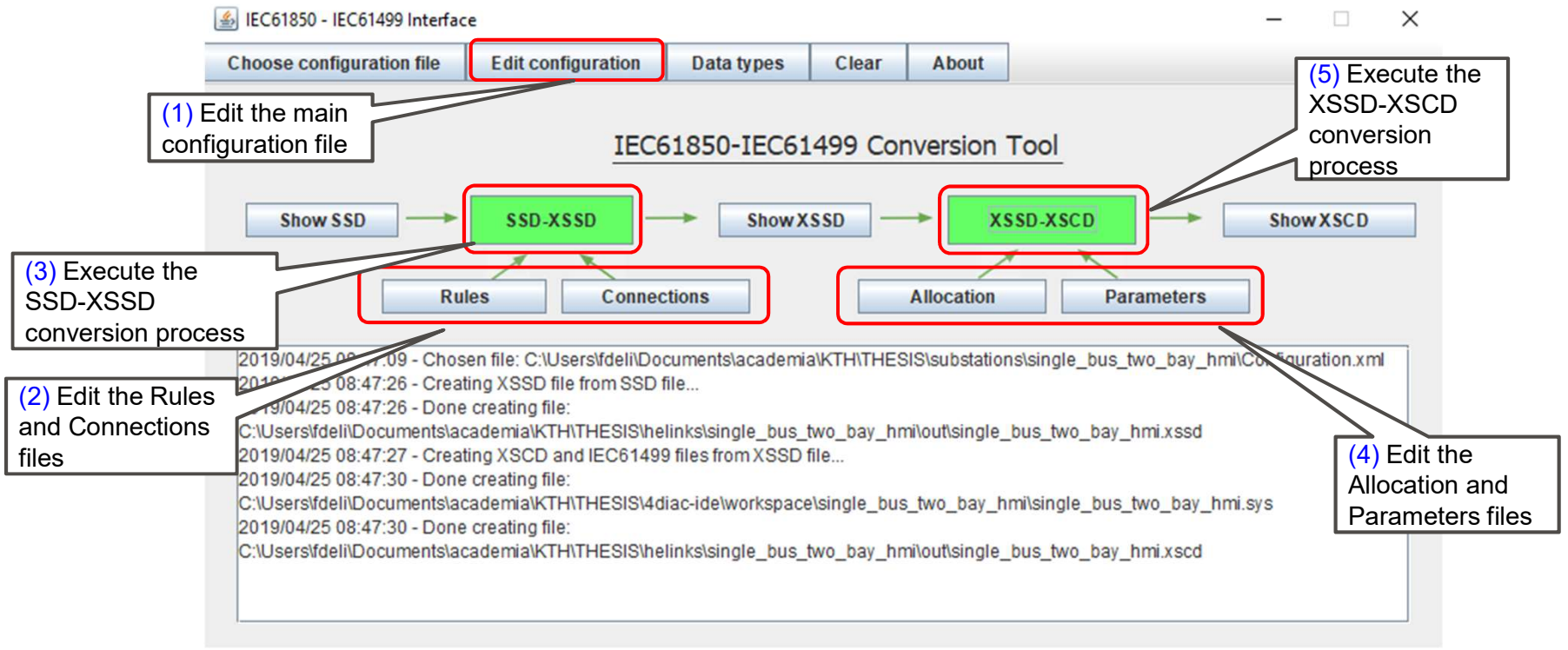




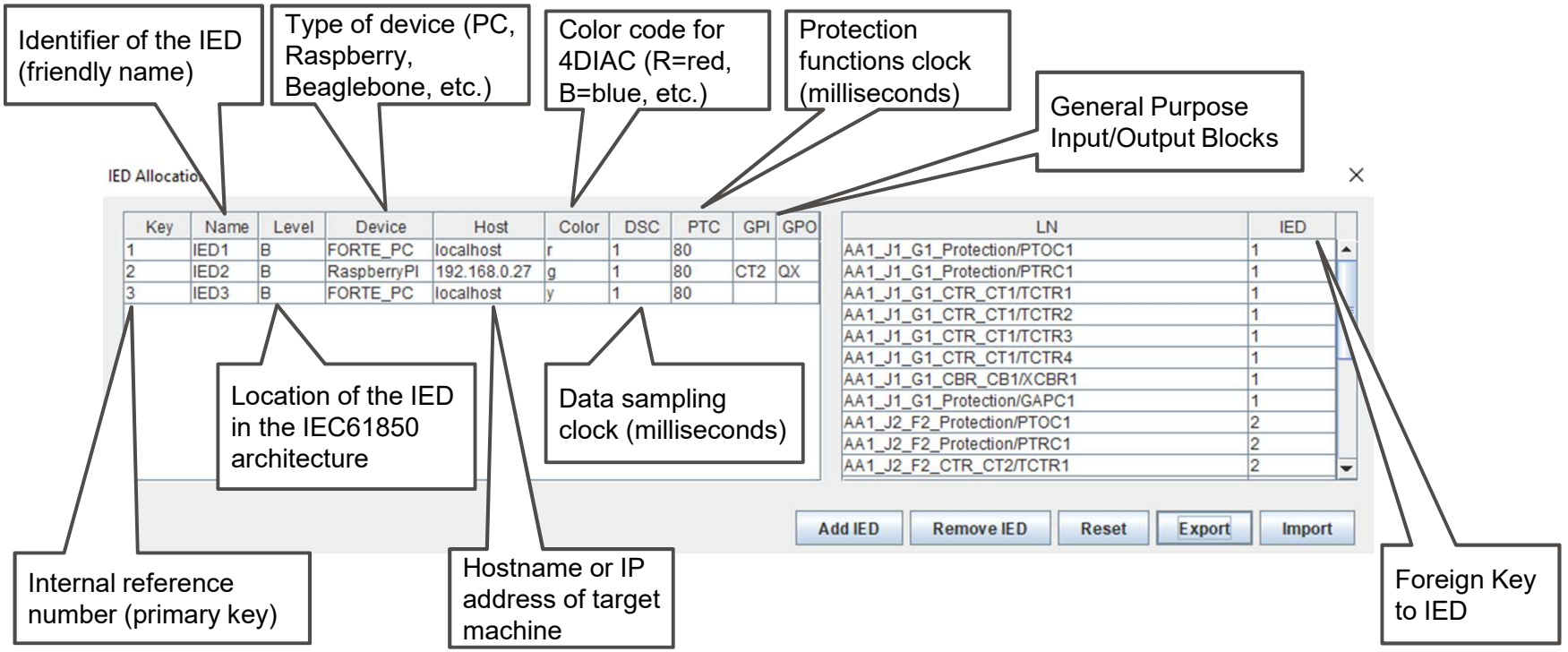
The Interface - Console Execution



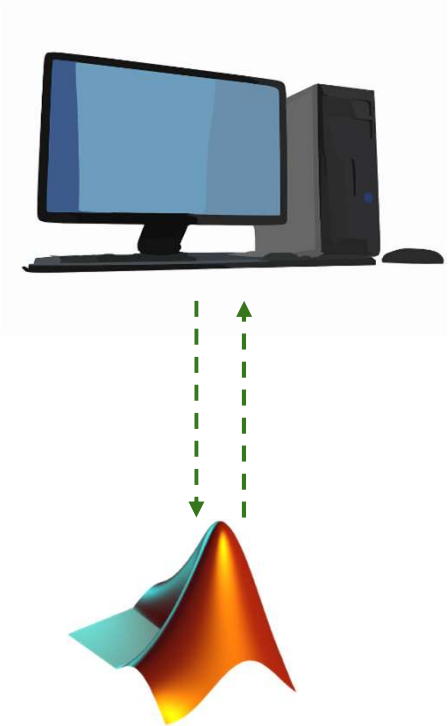
The Interface - Graphical Execution



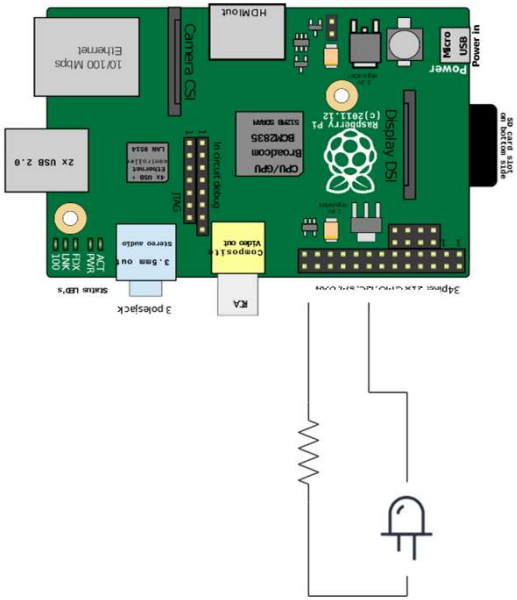
The Interface - Graphical Execution



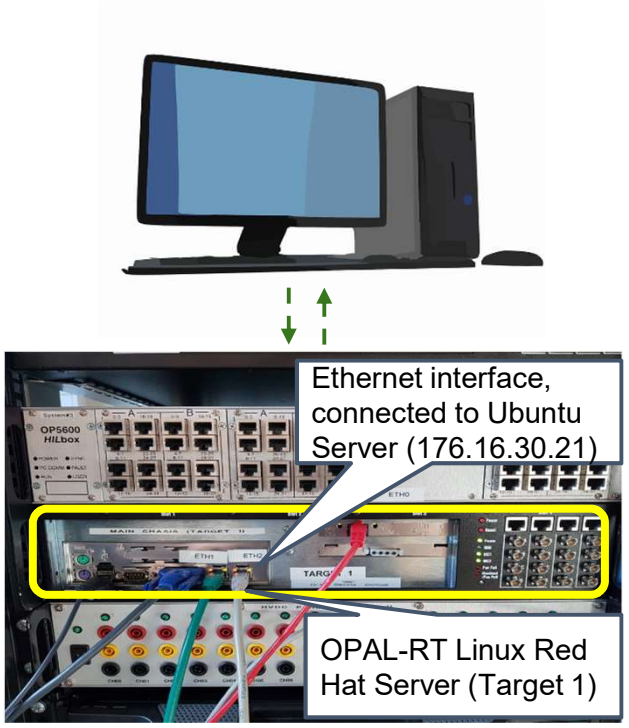
The Interface - Hardware Interfacing



Matlab TCP/IP Client- Server (CT2)



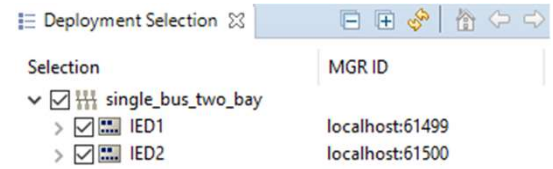
Raspberry GPIO Ports (QX)



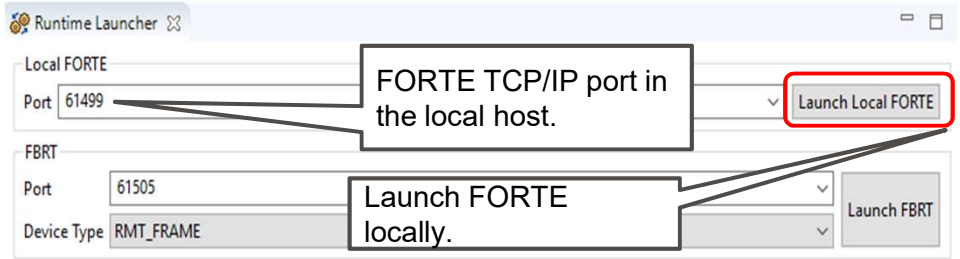
OPAL-RT UDP/IP Publish-Subscribe (CT3)

The Interface - Deployment

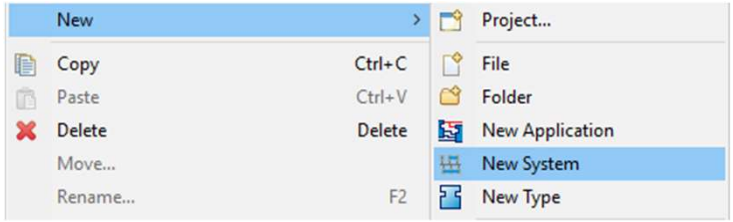
(2) Specify IEDs to be deployed and tested



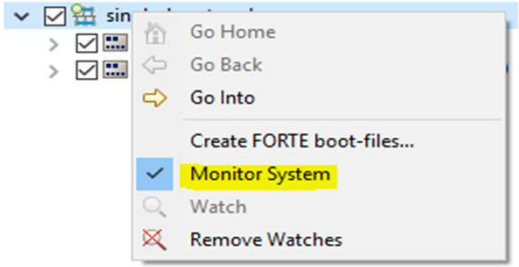
(3) Launch FORTE in the required ports



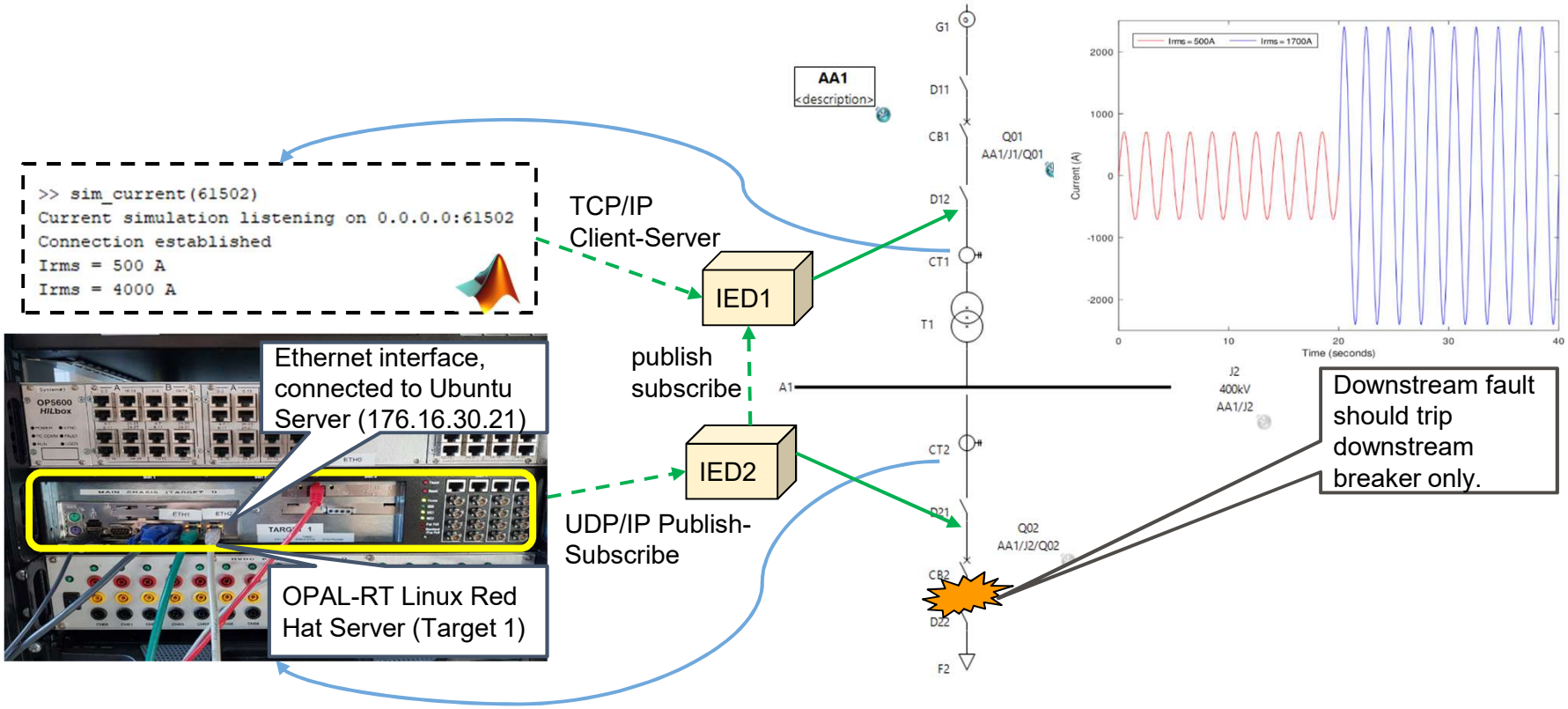
(1) Create new system in 4DIAC



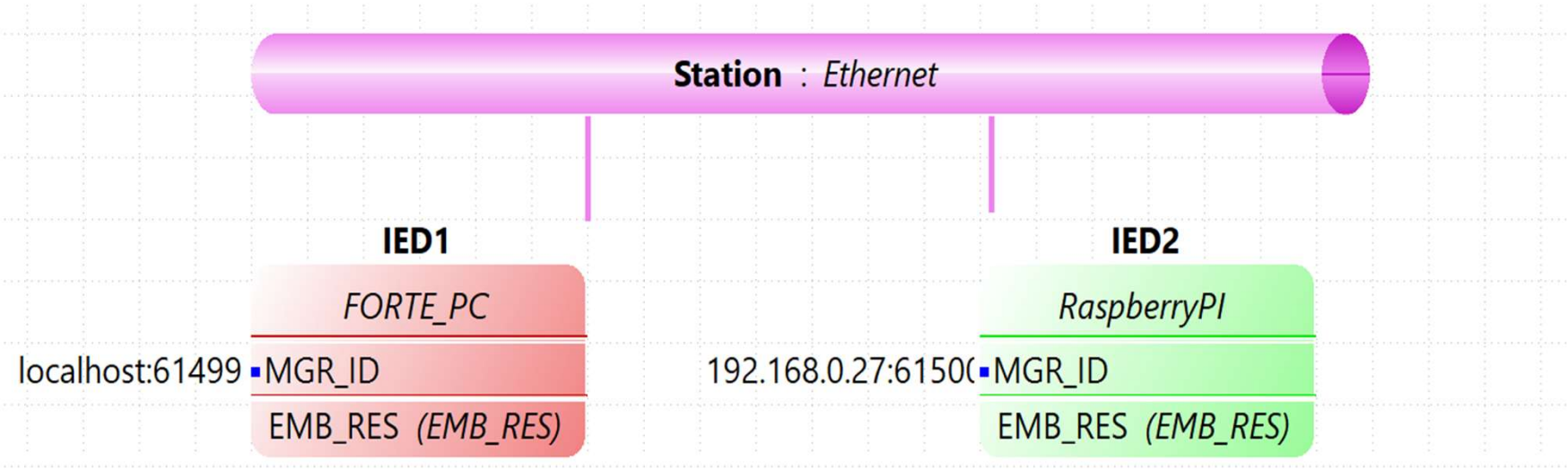
(4) Execute system monitoring (watch variables)



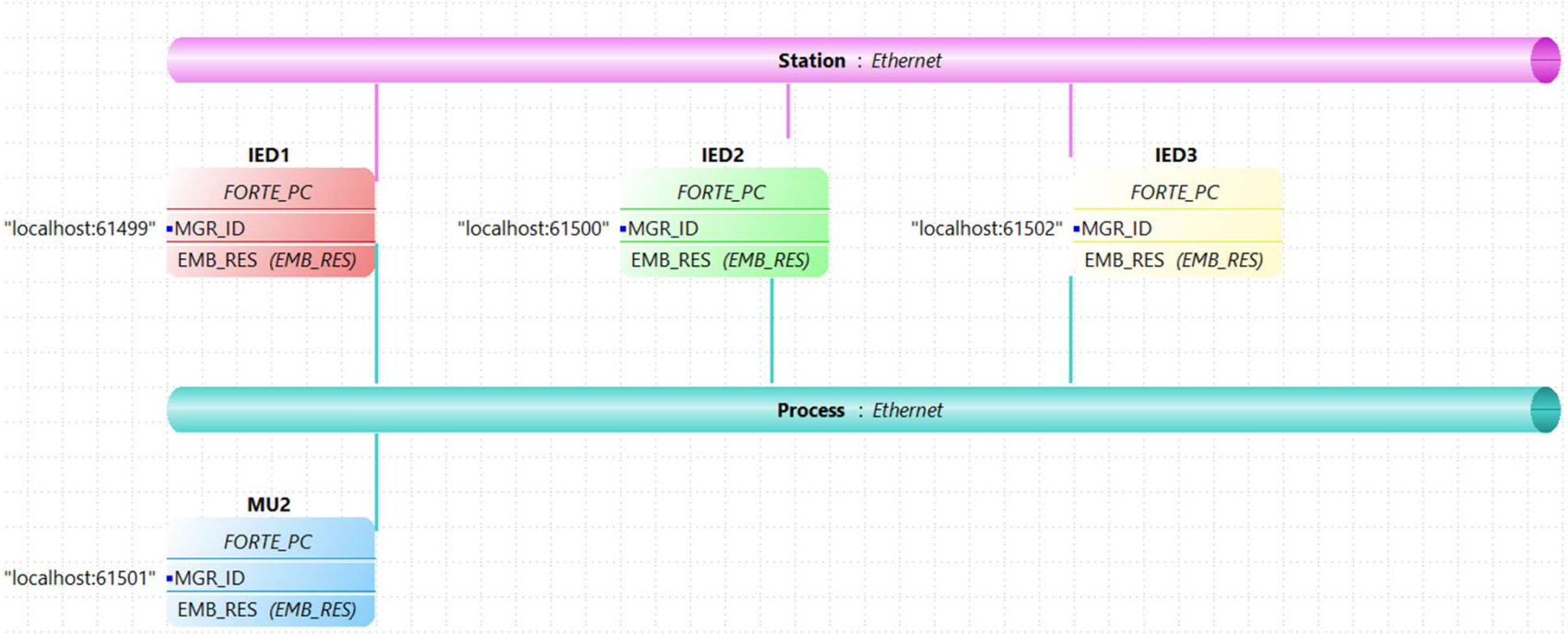
Test Case - Substation Design



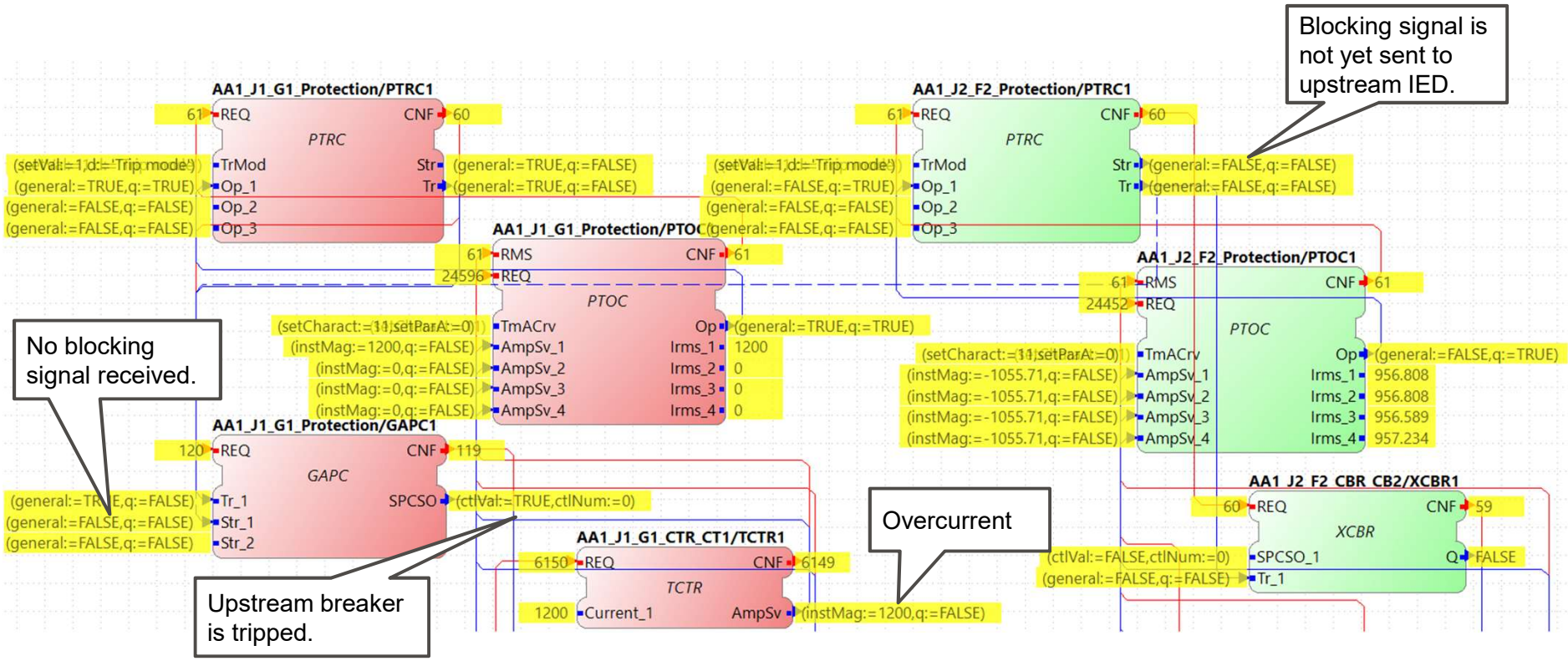
Test Case - System Configuration



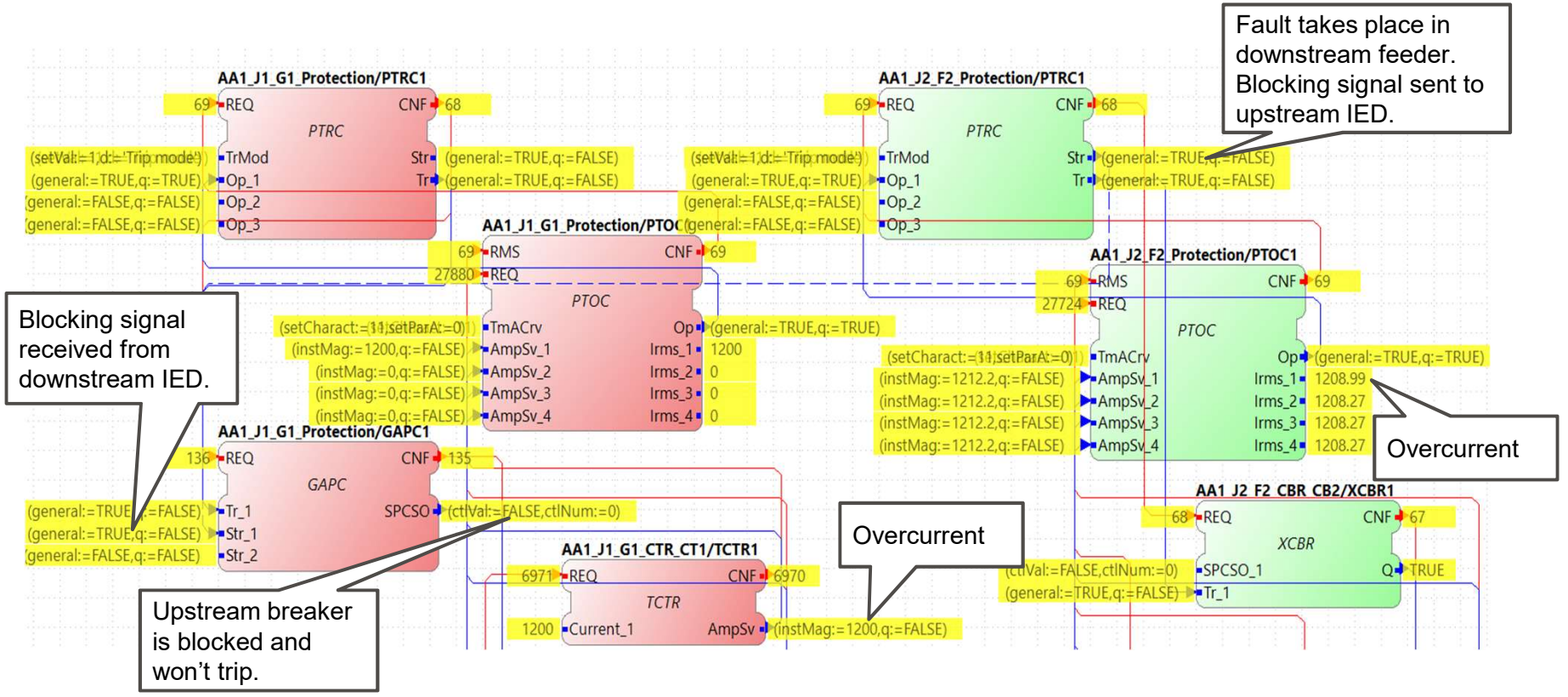
Test Case - System Configuration II



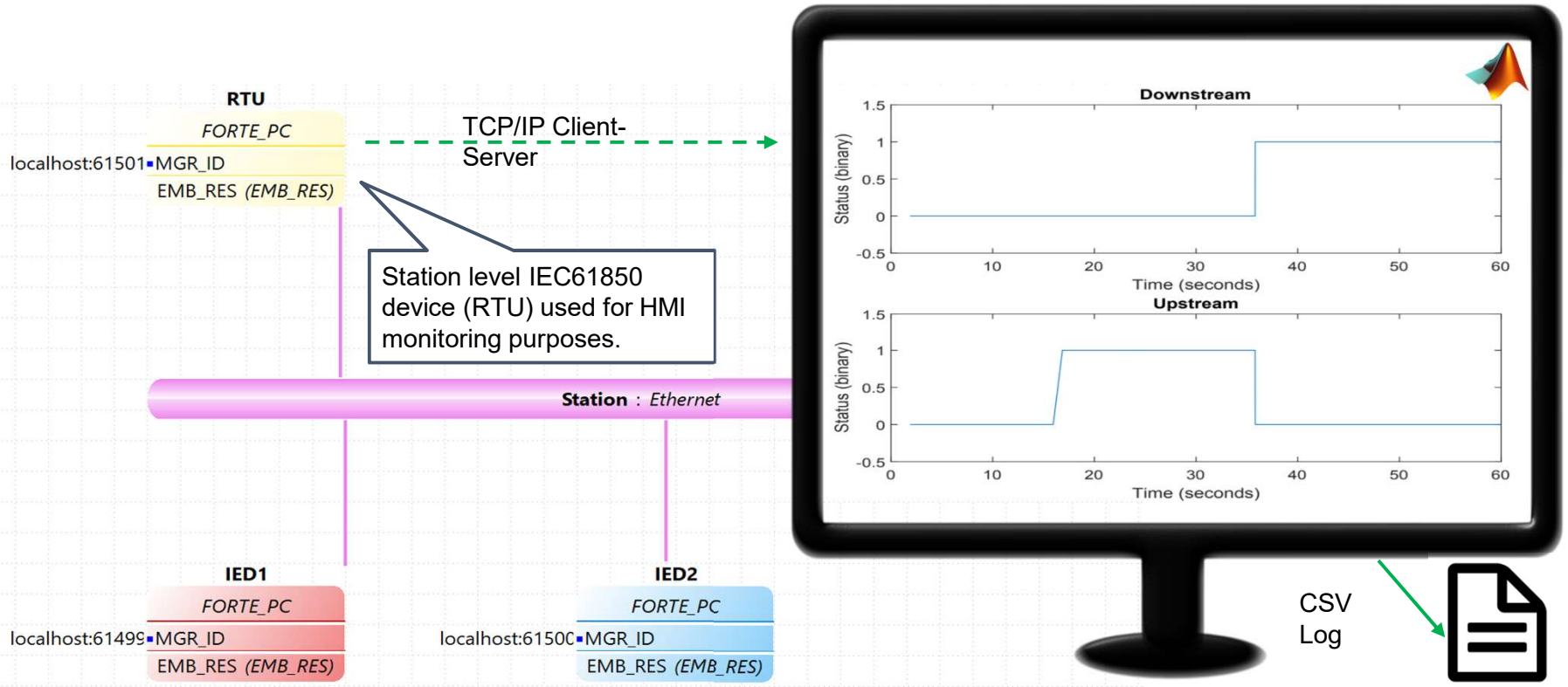
Test Case - Fault in Upstream Feeder



Test Case - Fault in Downstream Feeder



Test Case - System Configuration III

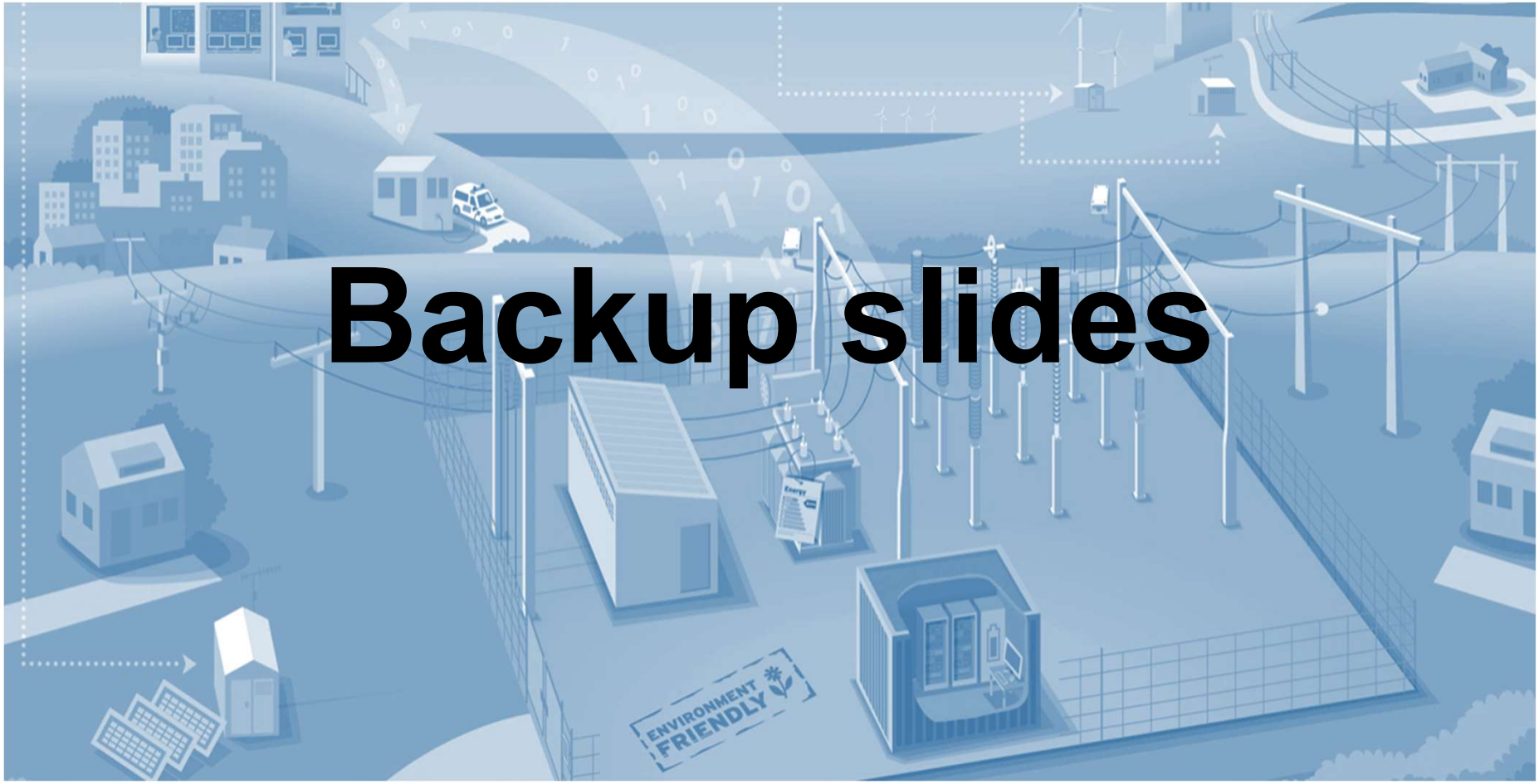


Conclusions

- The proposed methodology can potentially reduce development hours, human errors and hardware costs, as compared to the current approach.
- An automated interface takes an IEC61850 SSD file and outputs a fully fledged IEC61499 system configuration.
- The IEC61499 system configuration can be directly imported into a compliant tool (e.g. 4DIAC) and tested in a network of distributed devices (e.g. Raspberry PI, Beaglebone, etc).
- The interface also generates extended versions of the SSD and SCD documents, with additional connection and topology information presented in the IEC61499 standard.
- This development is completely open-source. Other manufacturers and organizations can implement their own function block algorithms, provided that their input/output ports correspond to standard IEC61850 data objects.

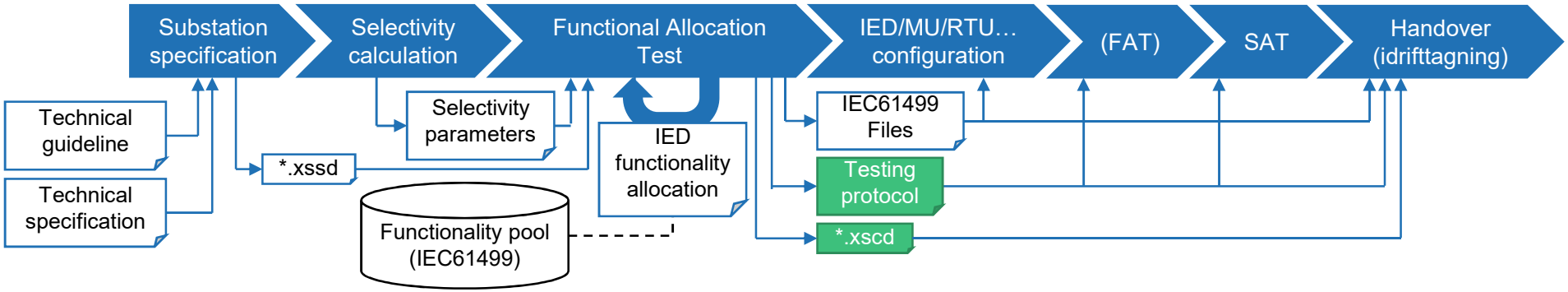
Future Work

- Include more visual aids and graphical tools in the interface, in order to further simplify the creation of the currently unassisted configuration files (especially the parameters).
- Include the full library of IEC61850 logical nodes, implemented in the IEC61499 standard. So far only a small group of simplified IEC61850 blocks have been programmed (and with limited functionality).
- Move the digital signal processing algorithms to a new general purpose function block, which will take care of the RMS calculations.
- Implement the conversion process in a real substation.
- Achieve better execution times in the IEC61499 platform. At this point, the runtime environment (FORTE) has some difficulties working in the microsecond range (possibly related to the size of the queue).

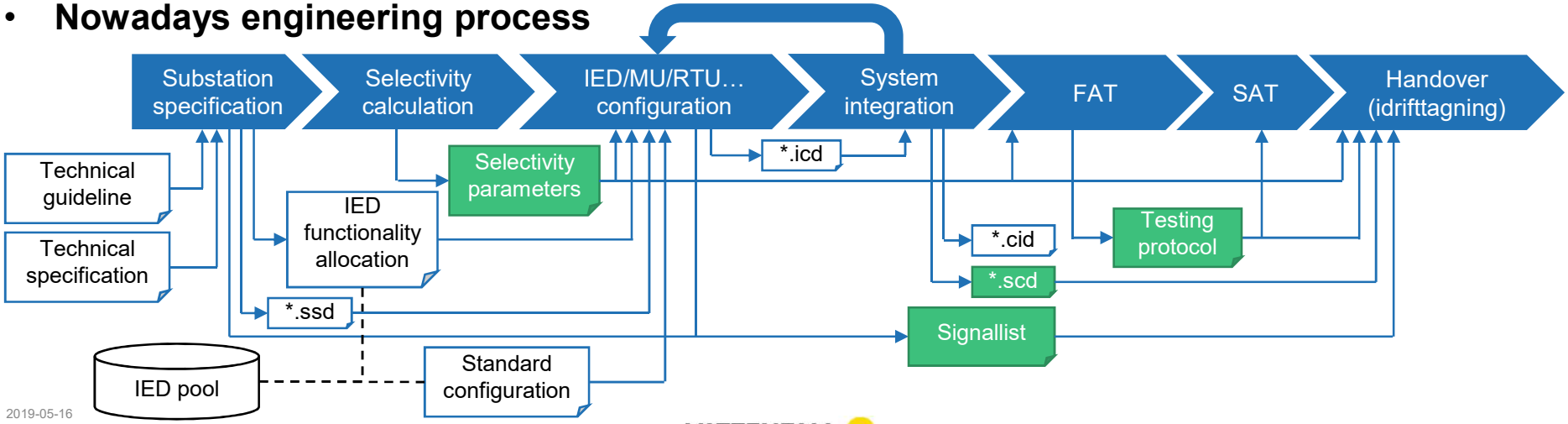


Backup slides

- **New top-down engineering process**



- **Nowadays engineering process**



The Interface - General Configuration

Full path to the different files in the system

Listing 1 General Configuration File

```
<?xml version="1.0" encoding="UTF-8"?>
<Configuration>
  <Files>
    <File type="Allocation" path="path\to\file.xml"></File>
    <File type="Parameters" path="path\to\file.xml"></File>
    <File type="Rules" path="path\to\file.xml"></File>
    <File type="Connections" path="path\to\file.xml"></File>
    <File type="SSD" path="path\to\file.ssd"></File>
    <File type="XSSD" path="path\to\file.xssd"></File>
    <File type="XSCD" path="path\to\file.xscd"></File>
    <File type="IEC61499" path="path\to\file.sys"></File>
  </Files>
  <General>
    <Option name="AppName" value="single_bus_three_bay"></
        Option>
    <Option name="Author" value="Francisco de Lima"></Option>
    <Option name="Organization" value="Vattenfall-KTH"></
        Option>
    <Option name="Version" value="0.1"></Option>
  </General>
</Configuration>
```

Some other general information

The Interface - Rules and Connections

Listing 2 Generic Rules File

```
<?xml version="1.0" encoding="UTF-8"?>
<GenericRules>
  <Substation>
    <VoltageLevel voltage="20kV">
      <Bay type="Generator">
        <RuleRef>1</RuleRef>
        <RuleRef>2</RuleRef>
            ...
      </Bay>
        ...
    </VoltageLevel>
      ...
  </Substation>
  <RuleList>
    <Rule reference="1" name="Rule 1" source="PTOC" destination=
      "PTRC" signal="Op"></Rule>
    <Rule reference="2" name="Rule 2" source="PTRC" destination=
      "GAPC" signal="Tr"></Rule>
      ...
  </RuleList>
</GenericRules>
```

Listing 3 Specific Connections File

```
<?xml version="1.0" encoding="UTF-8"?>
<ConnectionList>
    <Connection name="Connection 1" source="
        AA1_J2_Q02_Protection/PTRC1" destination="
        AA1_J1_Q01_Protection/GAPC1" signal="Str"></Connection
        >
    <Connection name="Connection 2" source="
        AA1_J2_Q03_Protection/PTRC1" destination="
        AA1_J1_Q01_Protection/GAPC1" signal="Str"></Connection
        >
</ConnectionList>
```

The Interface - Allocation and Parameters

Listing 4 Logical Node Allocation File

```
<?xml version="1.0" encoding="UTF-8" standalone="no"?>
<IEDList>
  <IED color="240,128,128" name="IED1" level="B" device="
      FORTE_PC" host="localhost" dsc="100" ptc="10000" gpi=""
      gpo="">
    <LN>AA1_J1_Q01_Protection/PTOC1</LN>
    <LN>AA1_J1_Q01_Protection/PTRC1</LN>
    ...
  </IED>
  <IED color="152,251,152" name="IED2" level="B" device="
      RaspberryPI" host="192.168.0.27" dsc="100" ptc="10000" gpi="
      CT1" gpo="QX">
    <LN>AA1_J2_Q02_Protection/PTOC1</LN>
    <LN>AA1_J2_Q02_Protection/PTRC1</LN>
    ...
  </IED>
  ...
</IEDList>
```

Listing 5 Parameters and Settings File

```
<?xml version="1.0" encoding="UTF-8"?>
<ParameterList>
  <Parameter name="Parameter 1" LN="AA1_J1_Q01_Protection/
      PTRC1" Signal="TrMod">
    <value>(setVal:=1,d:='Trip mode')</value>
  </Parameter>
    ...
  <Parameter name="Parameter 4" LN="AA1_J1_Q01_Protection/
      PTOC1" Signal="TmACrv">
    <value>(setCharact:=11)</value>
  </Parameter>
  ...
</ParameterList>
```