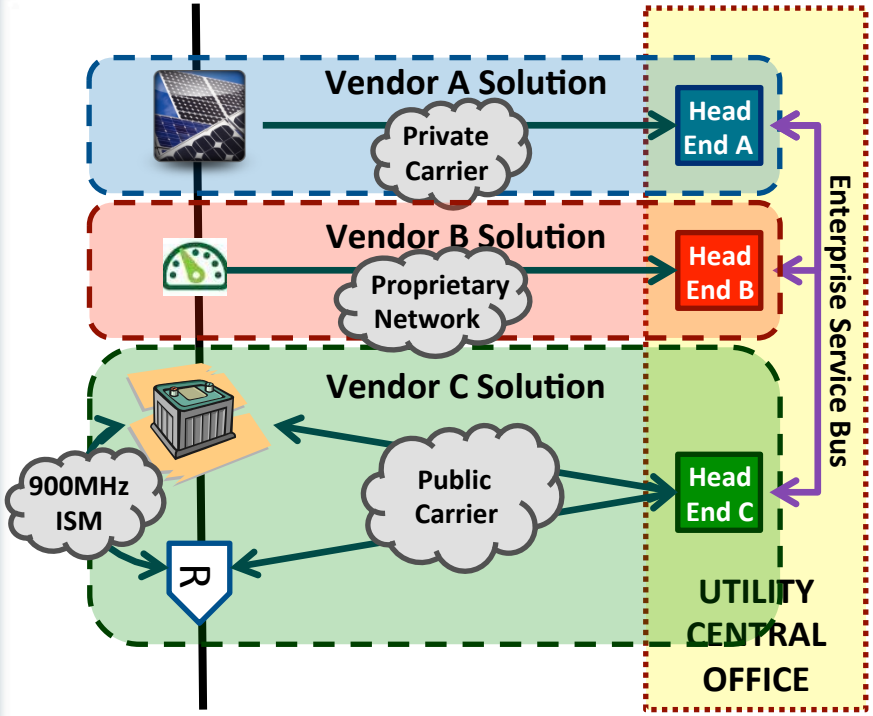
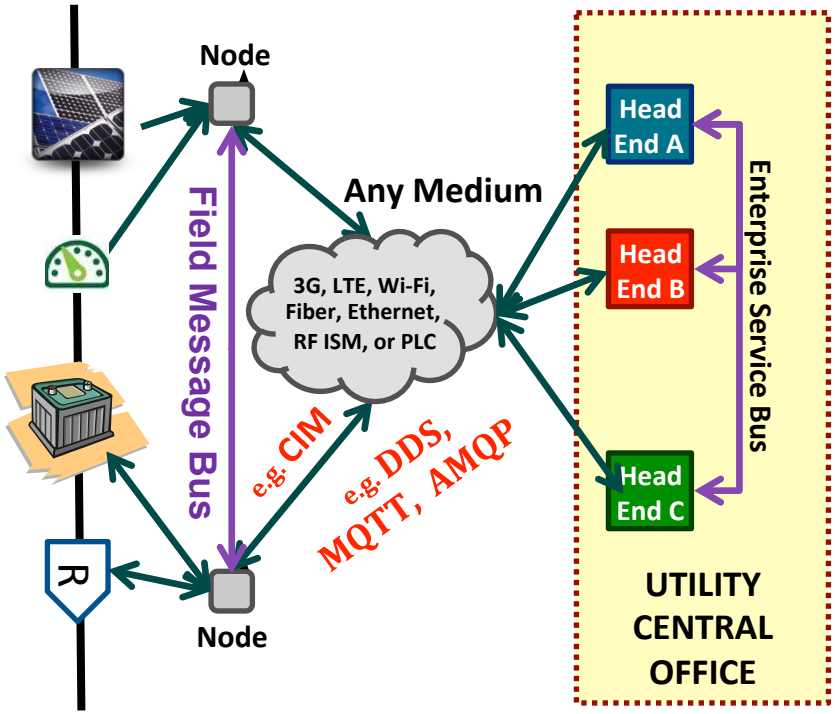
Use of IEC CIM Model for the Open Field Message Bus (OpenFMB)

Duke Energy & Xtensible Solutions

CIM users group

Enhancing DER Integration with OpenFMB





Key Observations:

1. Single-Purpose Functions
2. Proprietary & Silo'ed systems
3. Latent , Error-prone Data
4. OT/IT/Telecom Disconnected
5. No Field Interoperability!

Key Observations:

1. Multi-Purpose Functions
2. Modular & Scalable HW&SW
3. End-to-End Situational Awareness
4. OT/IT/Telecom Convergence
5. True Field Interoperabi

OpenFMB Operation: Federated Deterministic Exchanges

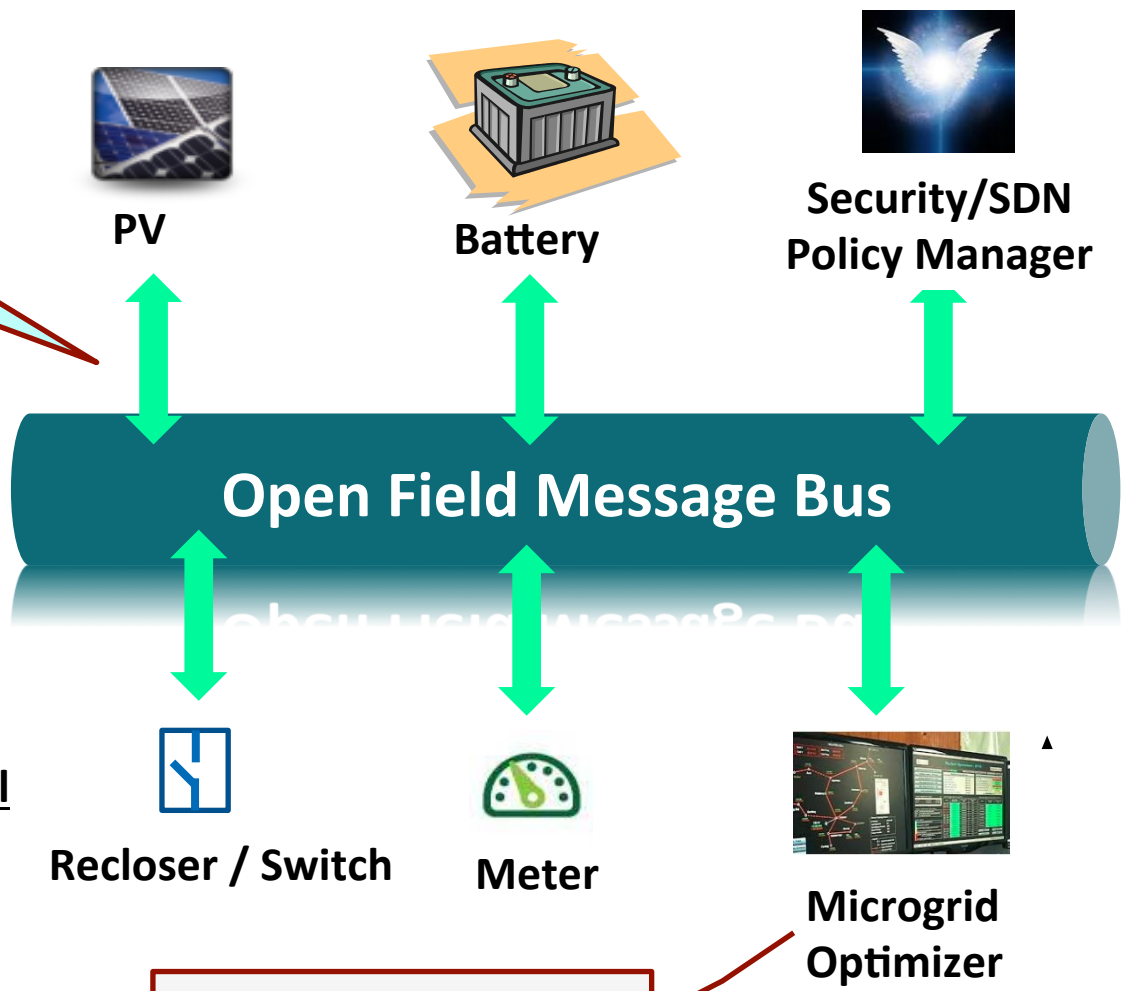
- Periodic Readings - Pub every few secs or near-real-time
- Data-Driven Events – on status change in near-real-time

Readings

KW A/B/C
KVAR A/B/C
V A/B/C
I A/B/C
Phase Angle A/B/C
KWh
TimeStamp
State of Charge

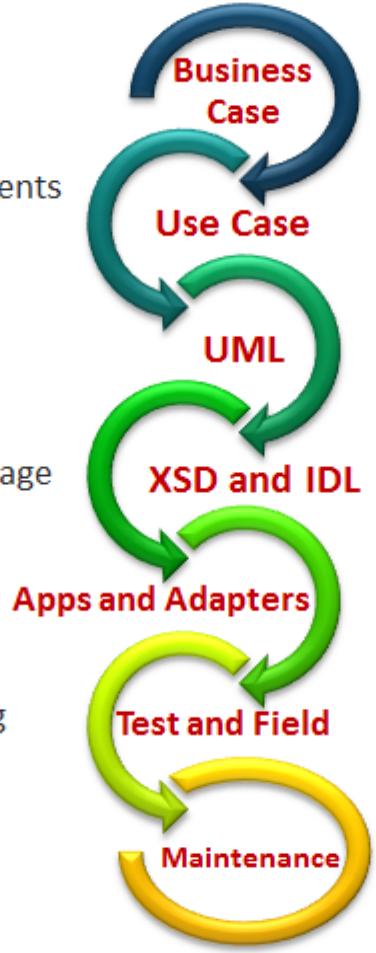
Status, Events, Alarms, & Control

Trip / Close
TimeStamp



Grid Edge Analytics

OpenFMB Framework Life Cycle



- Business-driven solutions

- Functional and non-functional requirements
- Interaction and sequencing

- Profile of applicable, existing data model

- Common software definitions and language

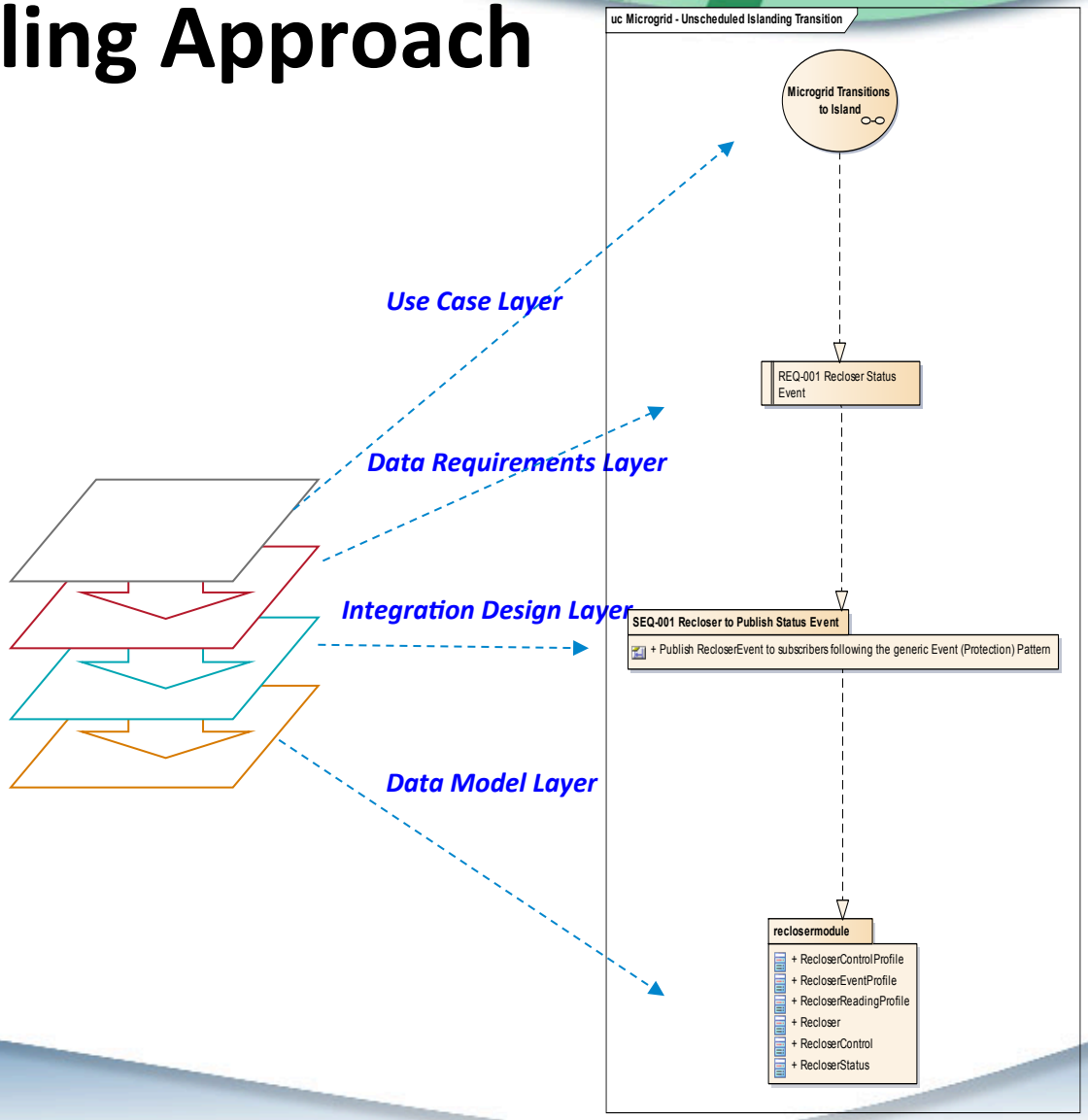
- Software tools to allow actors to interoperate

- System integration and validation testing

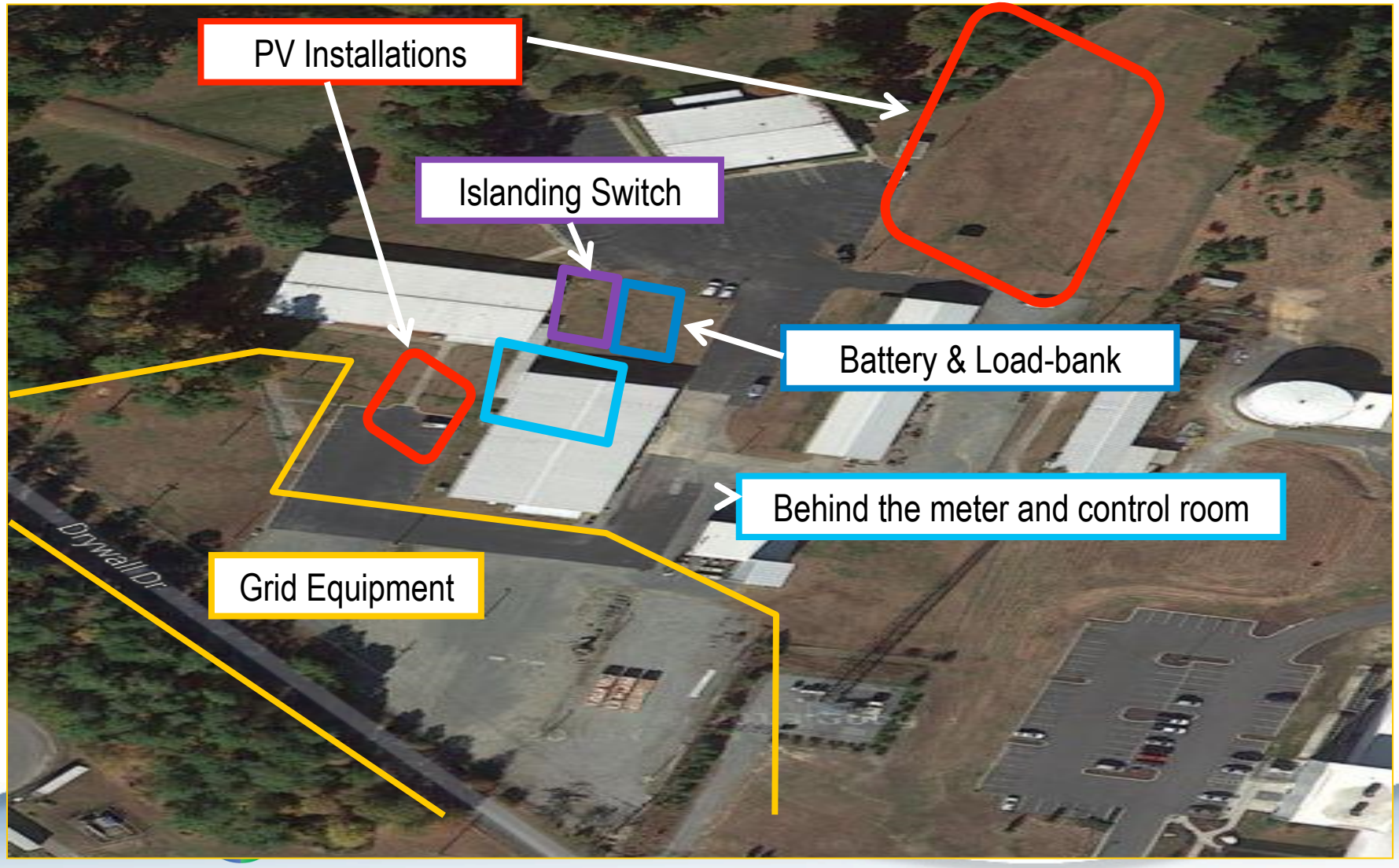
- Updates and versioning

OpenFMB Modeling Approach

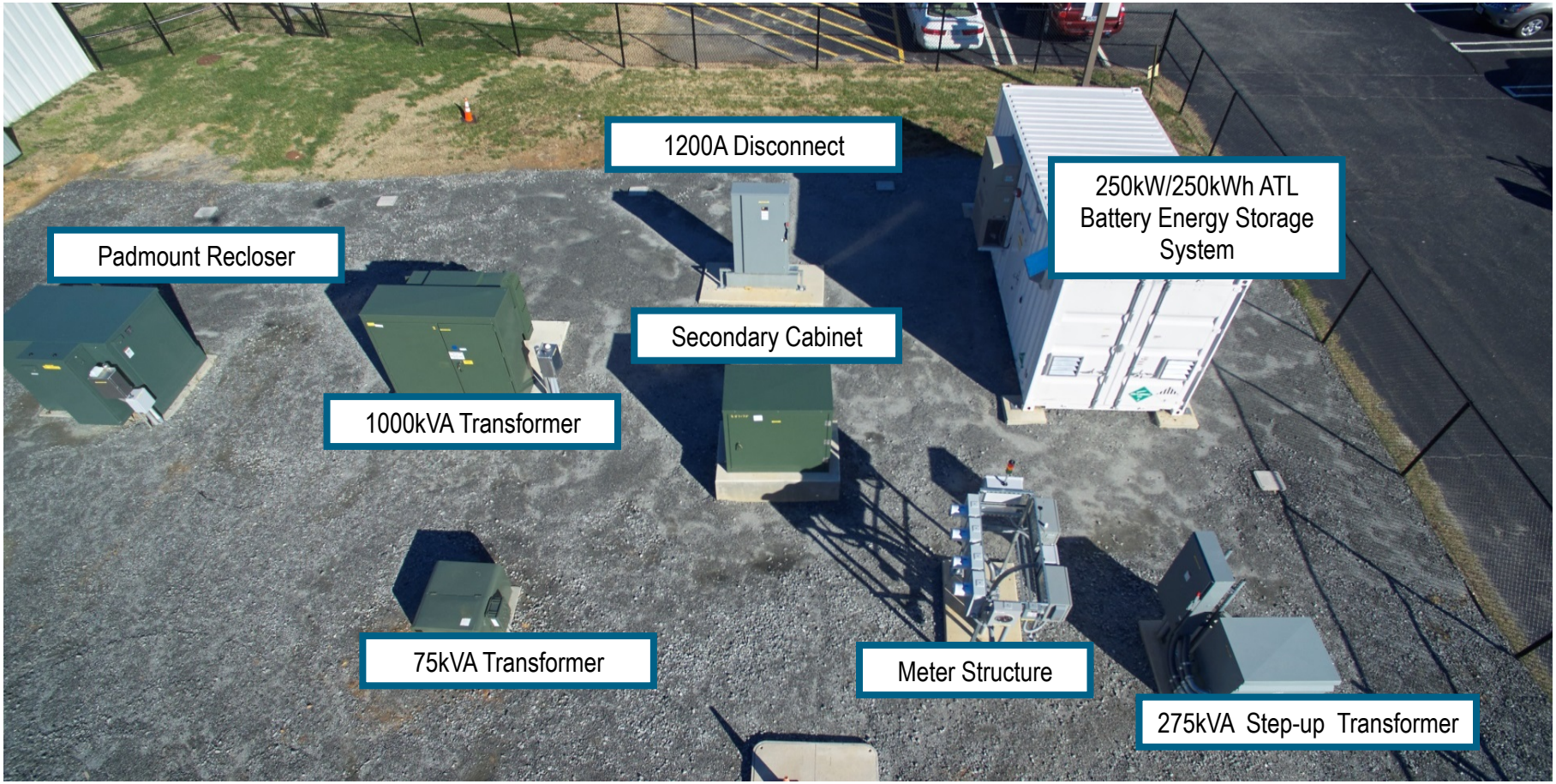
- Top-down business driven
- Layered architecture
  - Start with use cases and requirements
  - Structured in a single UML model
    - Using Sparx EA as modeling tool
  - Traceability among the layers
- Model driven artifacts generation



Duke Energy Microgrid Test Site: Mount Holly, NC



Mount Holly Microgrid Components

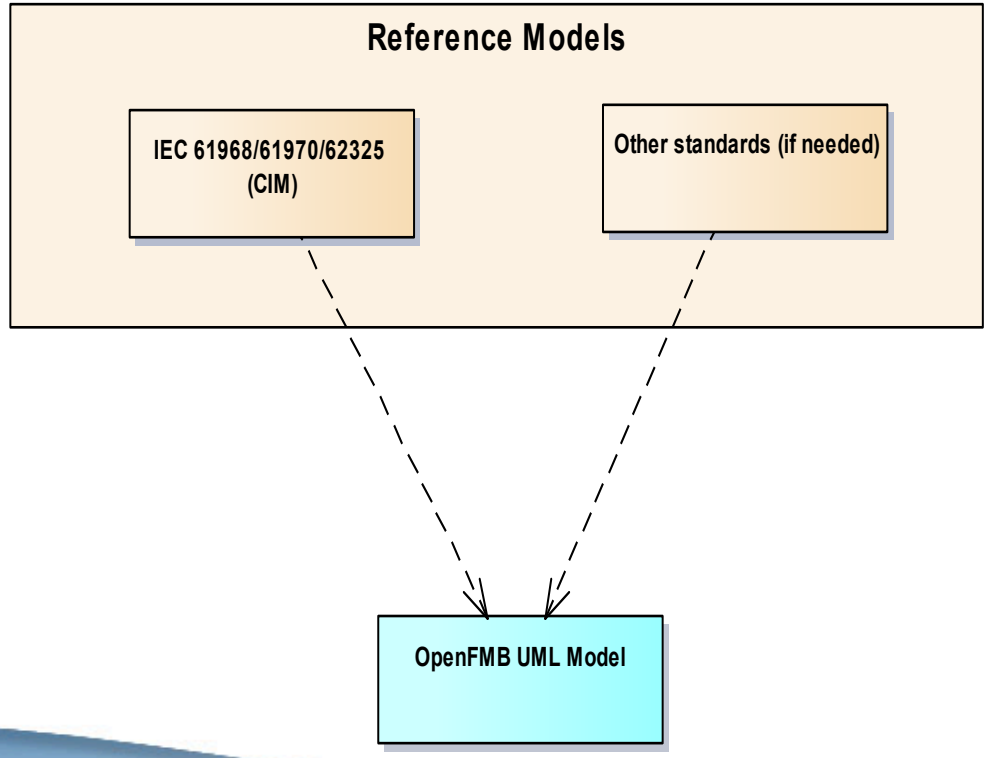


Not Pictured

100kW Output	Hanwha model 305 HSL72	Parker 100kVA Inverter	380 Polycrystalline Panels
10kW Output	310Watt HSL72	ABB PowerOne Uno 8.6 kW	30 Polycrystalline Panels
500kW	Avtron Load Bank		

Data Model Layer

- The data exchanged between the devices and systems are modeled in UML Class Diagrams based on standards.



Data Modeling

Reference Models

Standard UML

Reference Model

- Standards such as IEC CIM & IEC 61850
- Provide objects and relationships for OpenFMB requirements
- Application independent, but defines all concepts needed for any application

Context (Profile)

OpenFMB Profiles

Contextual layer restricts information model and extends as needed

- Cherry-picking reference model for given profile
- Restrictions and extensions
- Mandatory and optional
- Propose extension to the standards / reference models

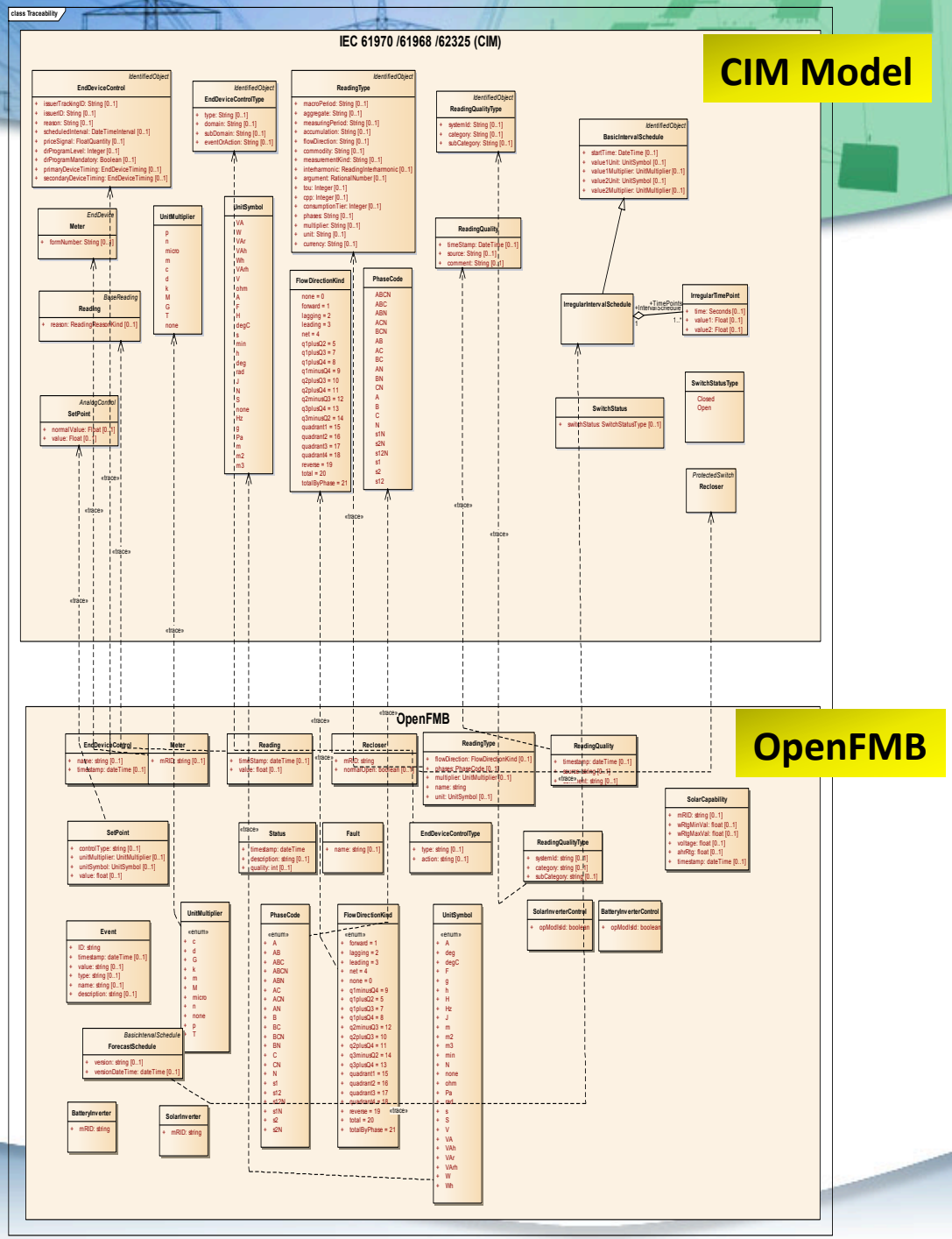
Message Syntax

Message/File Format (XSD, IDL and etc.)

Message syntax describes format for instance data

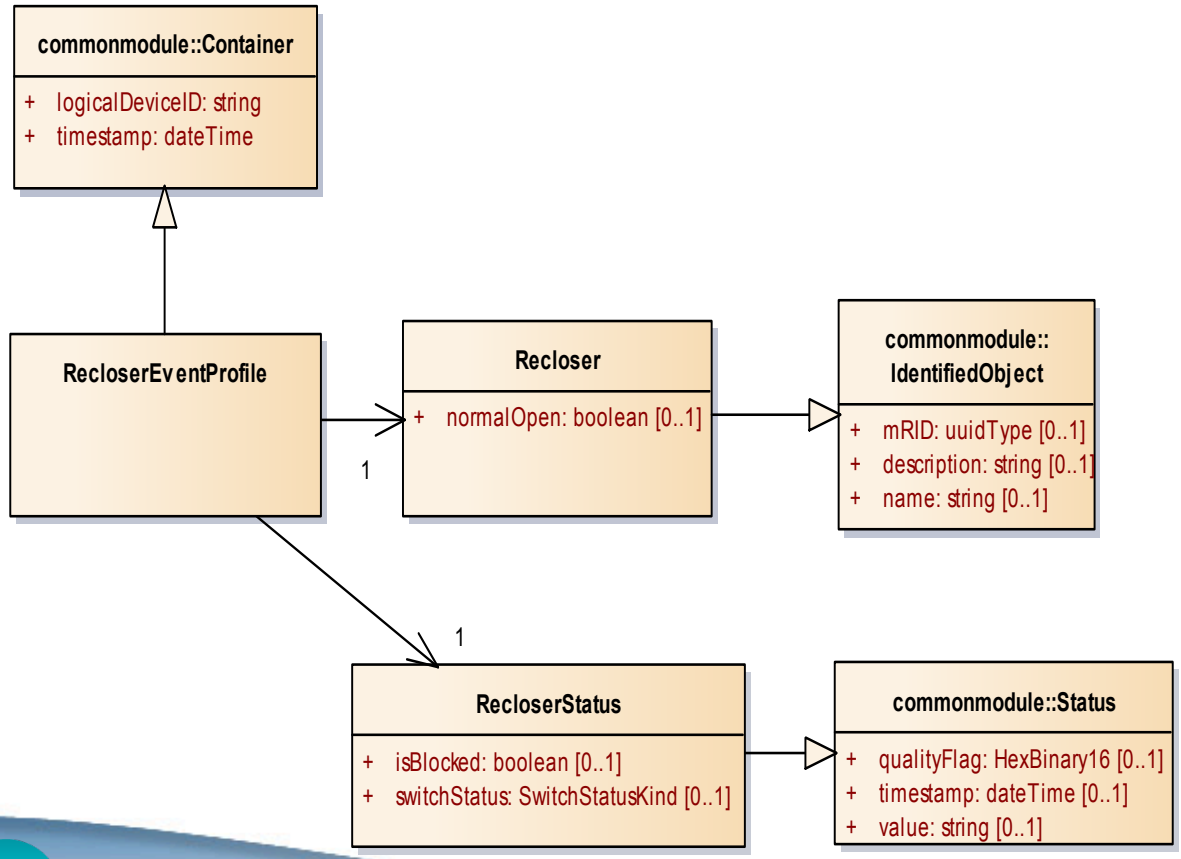
- Model driven artifacts generation
- Serialization of instance data
- May modify container or associations for message payloads
- Mappings to various technologies can be defined

Traceability



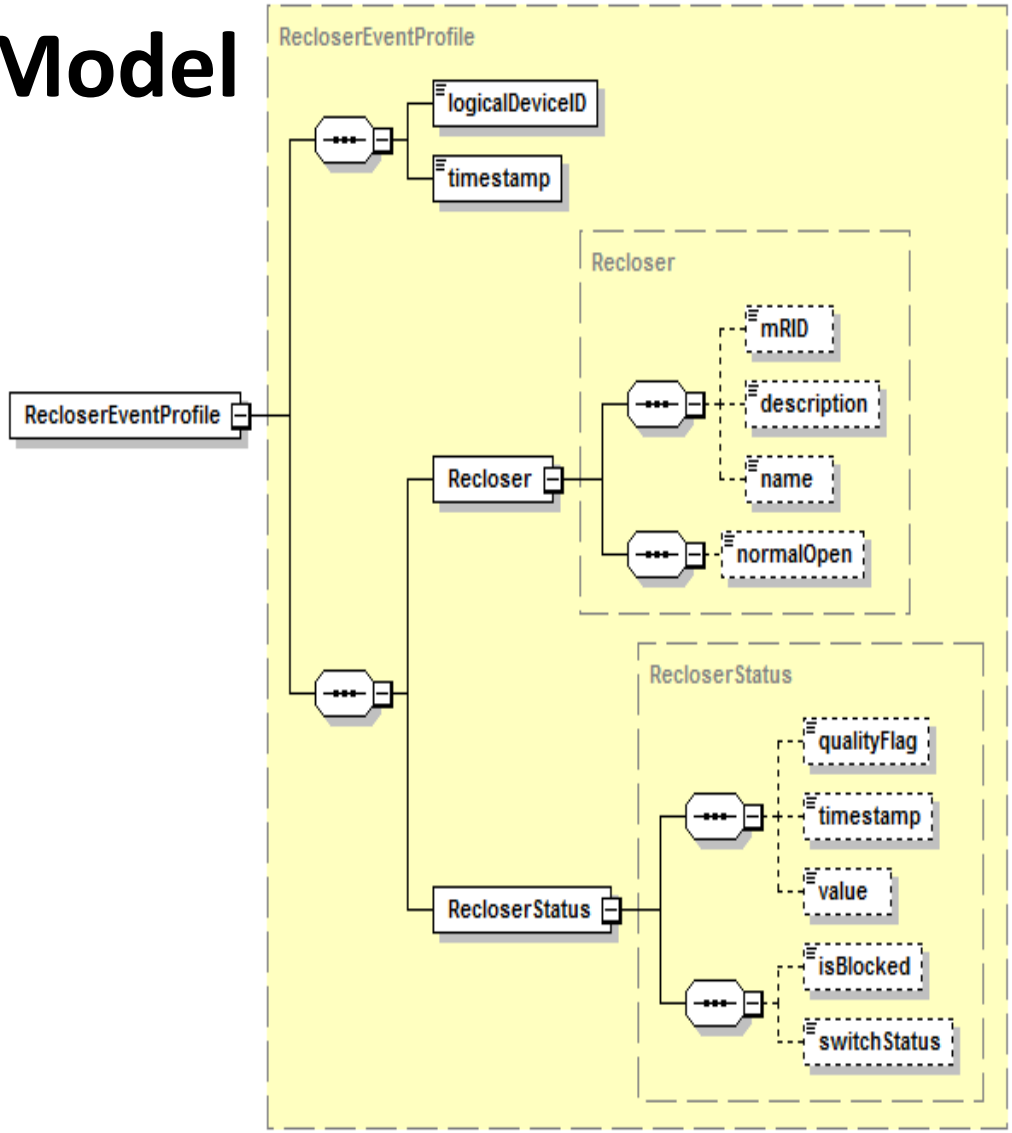
Platform Independent Model

- Logical model (Profile) built based on the mapping



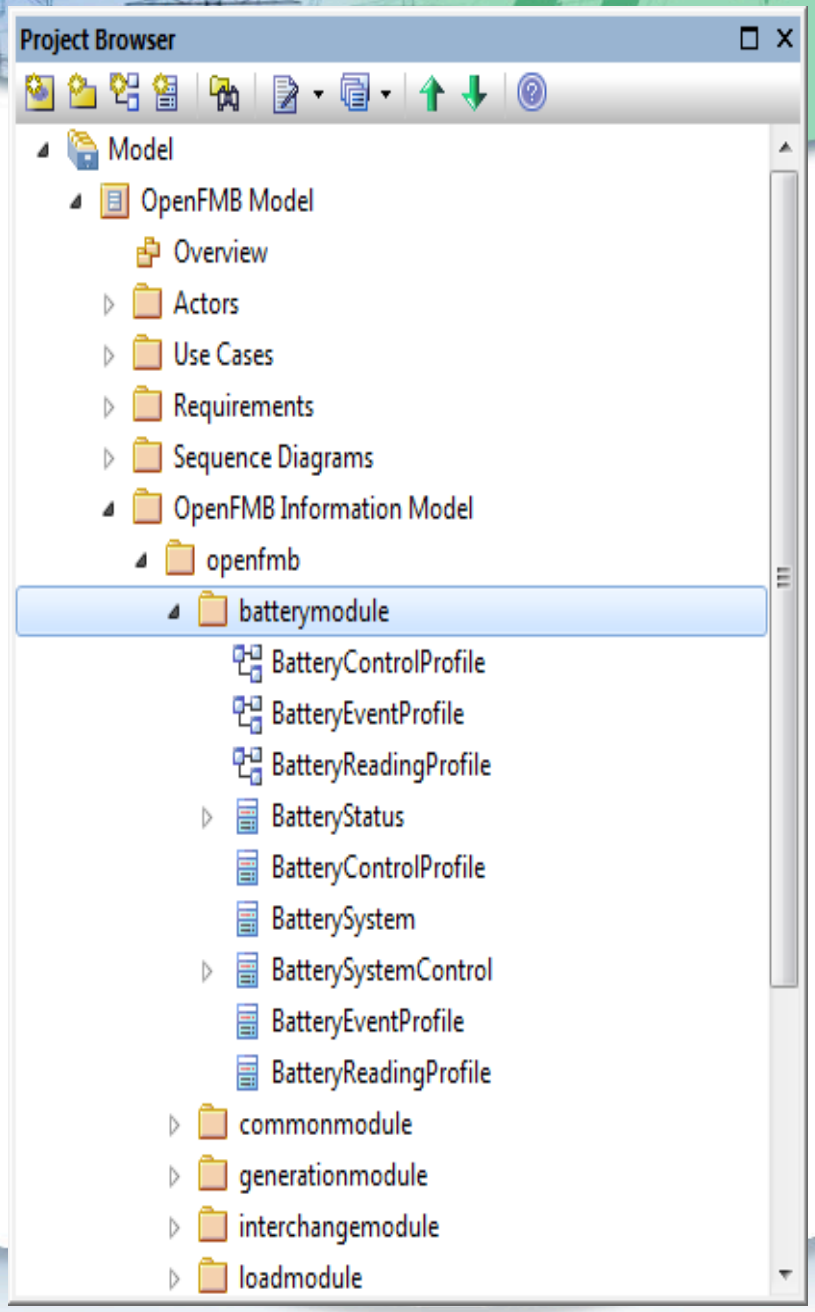
Platform Specific Model

- Physical implementation artifacts such as XSDs & IDLs are generated from the logical model



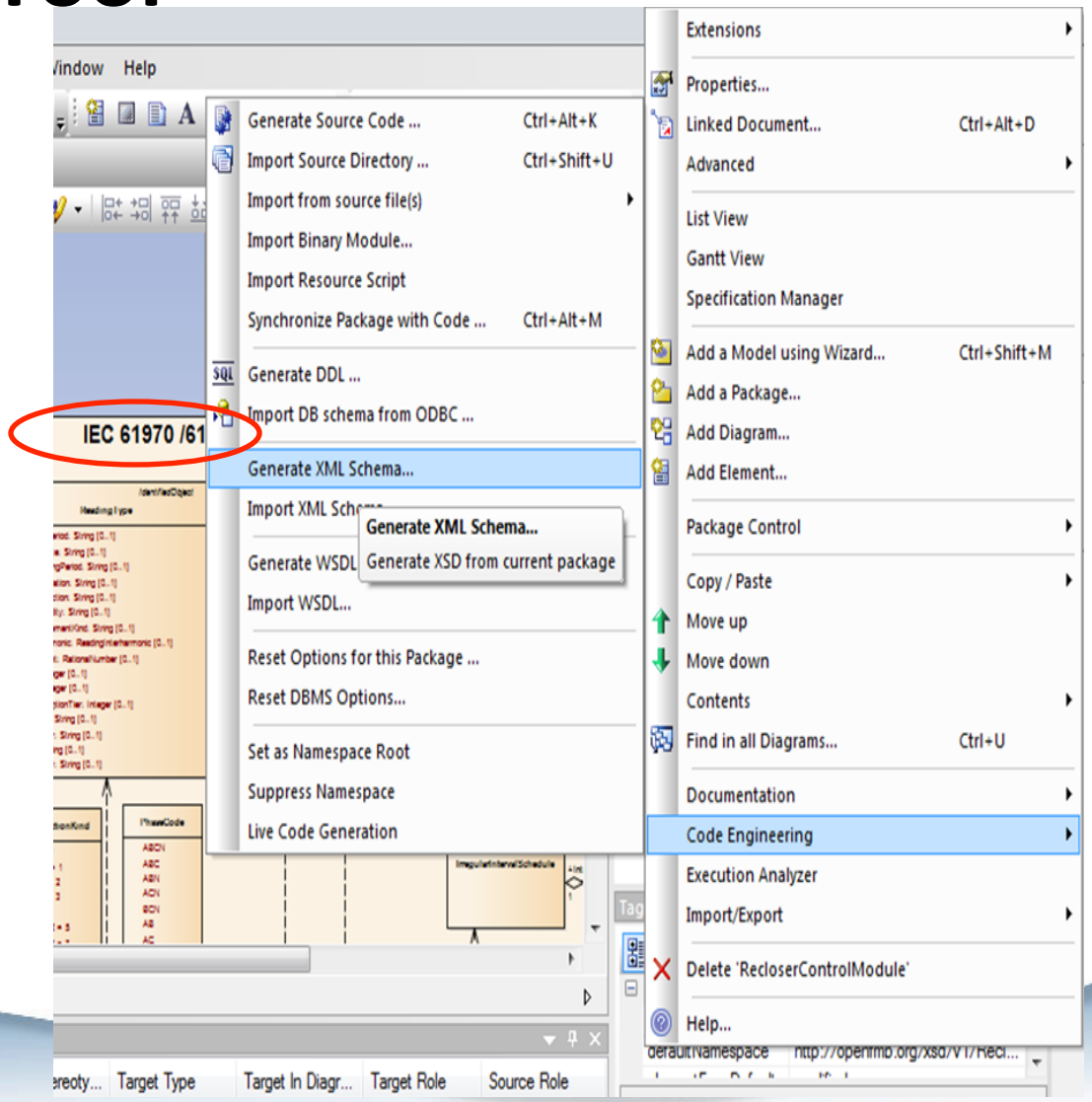
Module Structure

- Overall model structure



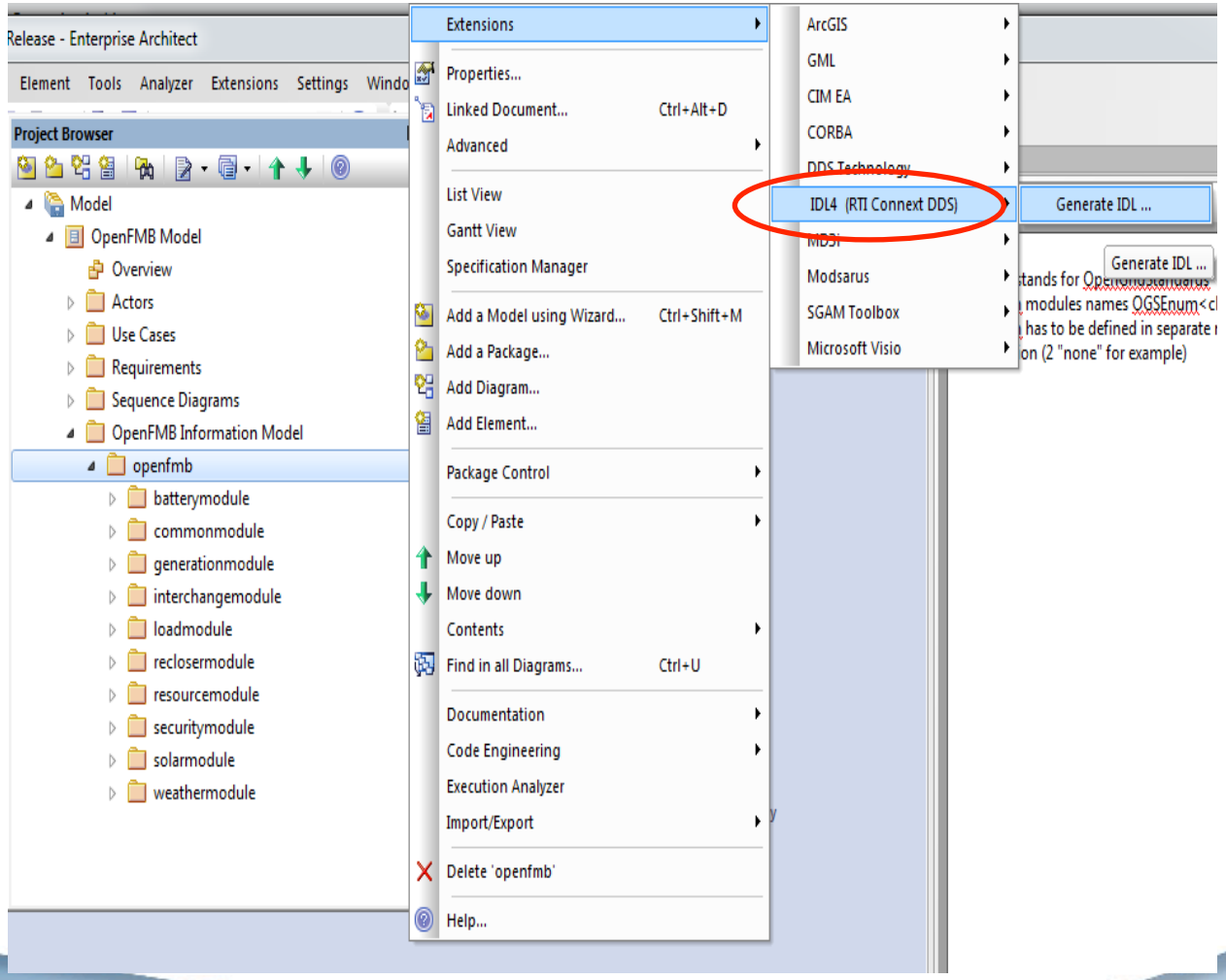
XSD Generation Tool

- Native Sparx EA tool used for XSD generation



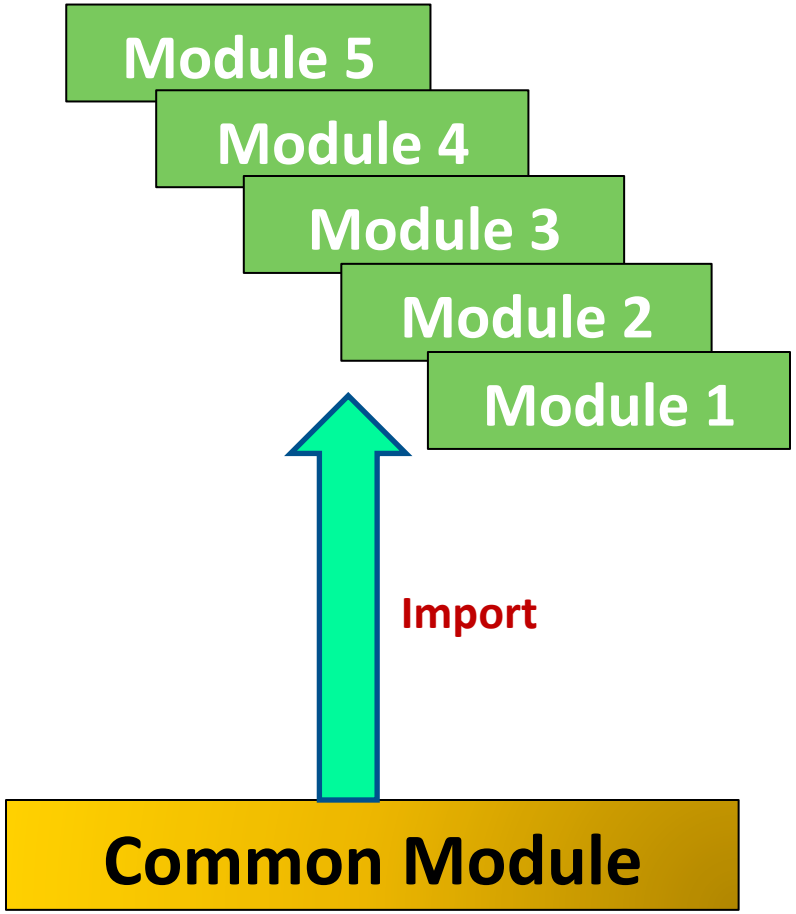
IDL Generation Tool

- RTI IDL4 for IDL generation



Common Module

- Common Module contains reusable classes shared (imported) across other modules
- Each module may contain multiple profiles



Namespace

- Namespace for all individual module
  - http://openfmb.org/<version #>/openfmb/<Module Name>
    - e.g. http://openfmb.org/2016/11/openfmb/reclosermodule

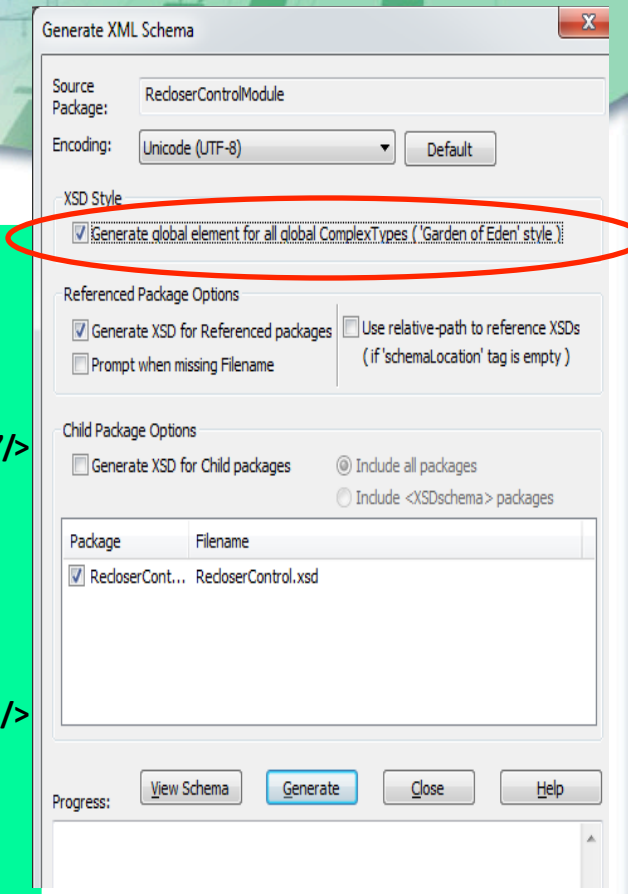
Version Control

- Two types of update in terms of version control:
  - Backward NOT Compatible:
    - Namespace updated with new version #
    - Version # updated in header
  - Backward Compatible:
    - Namespace NOT updated
    - Version # updated in header

XSD Style

- Global level element & Type
  - Garden of Eden

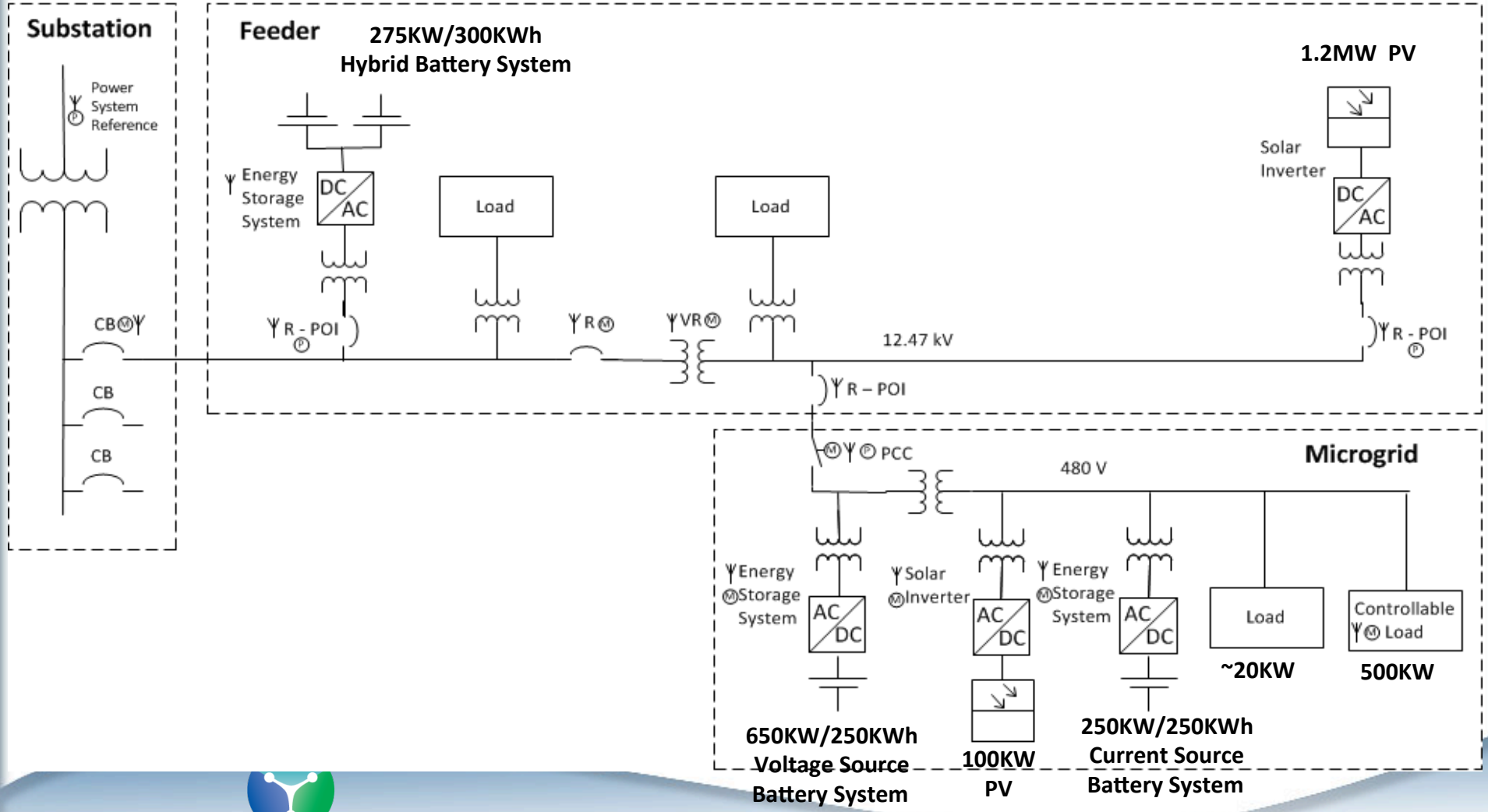
```
<xs:element name="Employee" type="EmployeeType"/>
<xs:element name="ErpPerson" type="ErpPersonType"/>
<xs:element name="ErpAddress" type="ErpAddressType"/>
<xs:complexType name="EmployeeType">
    <xs:sequence>
        <xs:element name="ErpPerson" type="ErpPersonType"/>
        <xs:element name="ErpAddress"
type="ErpAddressType"/>
    </xs:sequence>
</xs:complexType>
<xs:complexType name="ErpPersonType">
    <xs:sequence>
        <xs:element name="lastName" type="xs:string"/>
        <xs:element name="firstName"
type="xs:string"/>
    </xs:sequence>
</xs:complexType>
<xs:complexType name="ErpAddressType">
    <xs:sequence>
        <xs:element name="streetNumber" type="xs:string"/>
        <xs:element name="streetName"
type="xs:string"/>
    </xs:sequence>
</xs:complexType>
```



Message Types

- Reading (both analog & discrete)
- Control & Control Schedule
- Event
  - Alarm
  - Informational
  - Protection
  - Workflow
- Status

Duke Energy Rankin / Mount Holly Microgrid Pilot Circuit



Discussion – Q&A