
Prediktor
OPC UA
Asset heavy industries
Information Models
Johan Sverdrup example
Pledge


OPC UA
THE FUTURE OF INTEROPERABILITY
Information models
An example
Takeaways

Information model

From Wikipedia, the free encyclopedia
(Redirected from Information models)

An **information model** in software engineering is a representation of concepts and the relationships, constraints, rules, and operations to specify data semantics for a chosen domain of discourse. Typically it specifies relations between kinds of things, but may also include relations with individual things. It can provide sharable, stable, and organized structure of information requirements or knowledge for the domain context.[1]

An IDEF1X Diagram, an example of an Integration Definition for Information Modeling.

Overview [edit]

The term *information model* in general is used for models of individual things, such as facilities, buildings, process plants, etc. In those cases the concept is specialised to facility information model, building information model, plant information model, etc. Such an information model is an integration of a model of the facility with the data and documents about the facility.

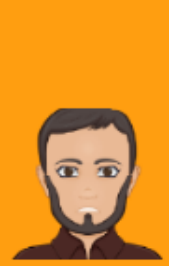
Within the field of software engineering and data modeling an information model is usually an abstract, formal representation of entity types that may include their properties, relationships and the operations that can be performed on them. The entity types in the model may be kinds of real-world objects, such as devices in a network, or occurrences, or they may themselves be abstract, such as for the entities used in a billing system. Typically, they are used to model a constrained domain that can be described by a closed set of entity types, properties, relationships and operations.

An information model provides formalism to the description of a problem domain without constraining how that description is mapped to an actual implementation in software. There may be many mappings of the information model. Such mappings are called data models, irrespective of whether they are object models (e.g. using UML), entity relationship models or XML schemas.

An **information model** in software engineering is a representation of concepts and the relationships, constraints, rules, and operations to specify data semantics for a chosen domain of discourse. Typically it specifies relations between kinds of things, but may also include relations with individual things. It can provide sharable, stable, and organized structure of information requirements or knowledge for the domain context.[1]

An example

A set of different living human beings, flesh and blood instances of human entities


Paul


Anna


Jeff


Nick


Peter


Donald


Steve


Kevin


Theresa


Tom

Seen through local sports-court caretaker, Leopold: A junior basketball match

Match

Team B

Team A


Paul


Anna


Jeff


Nick


Peter


Donald

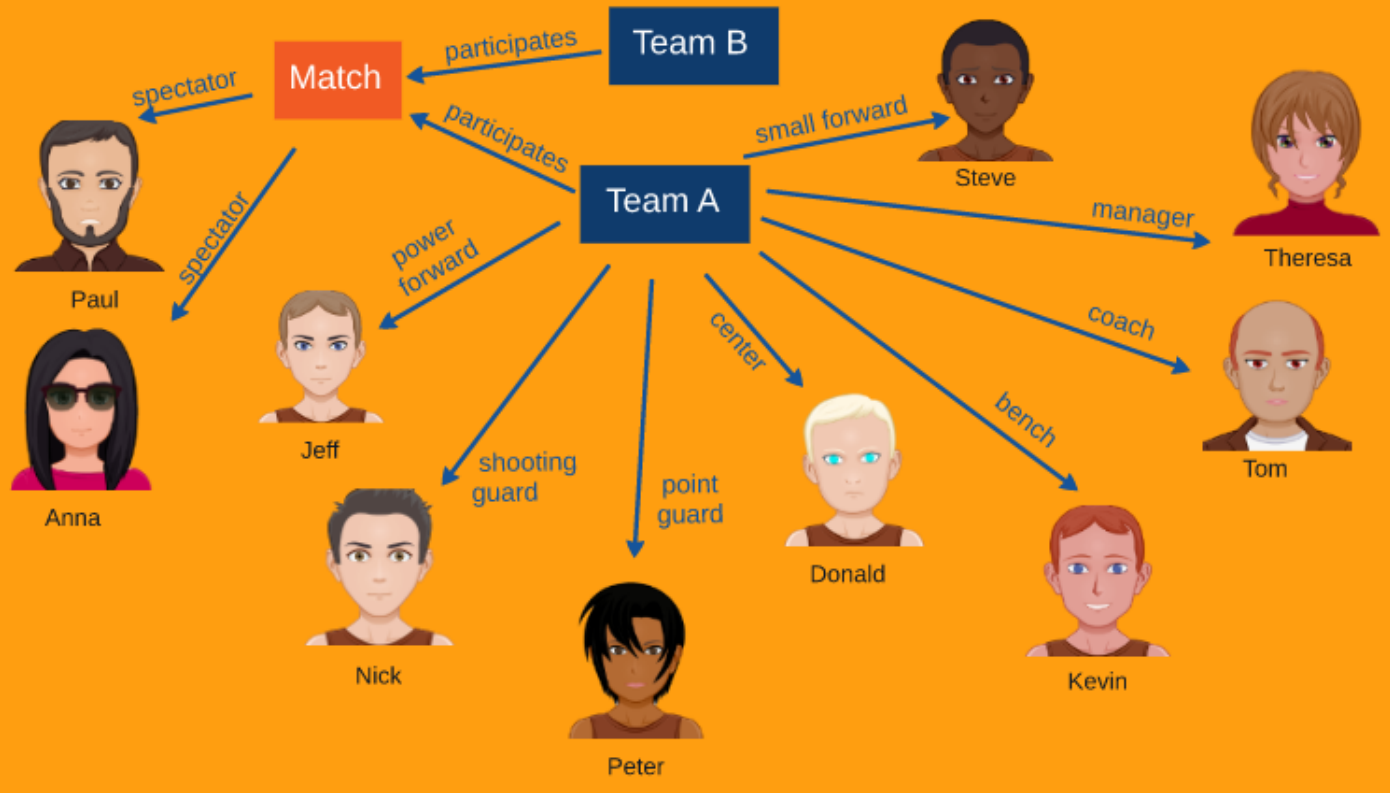

Steve


Kevin

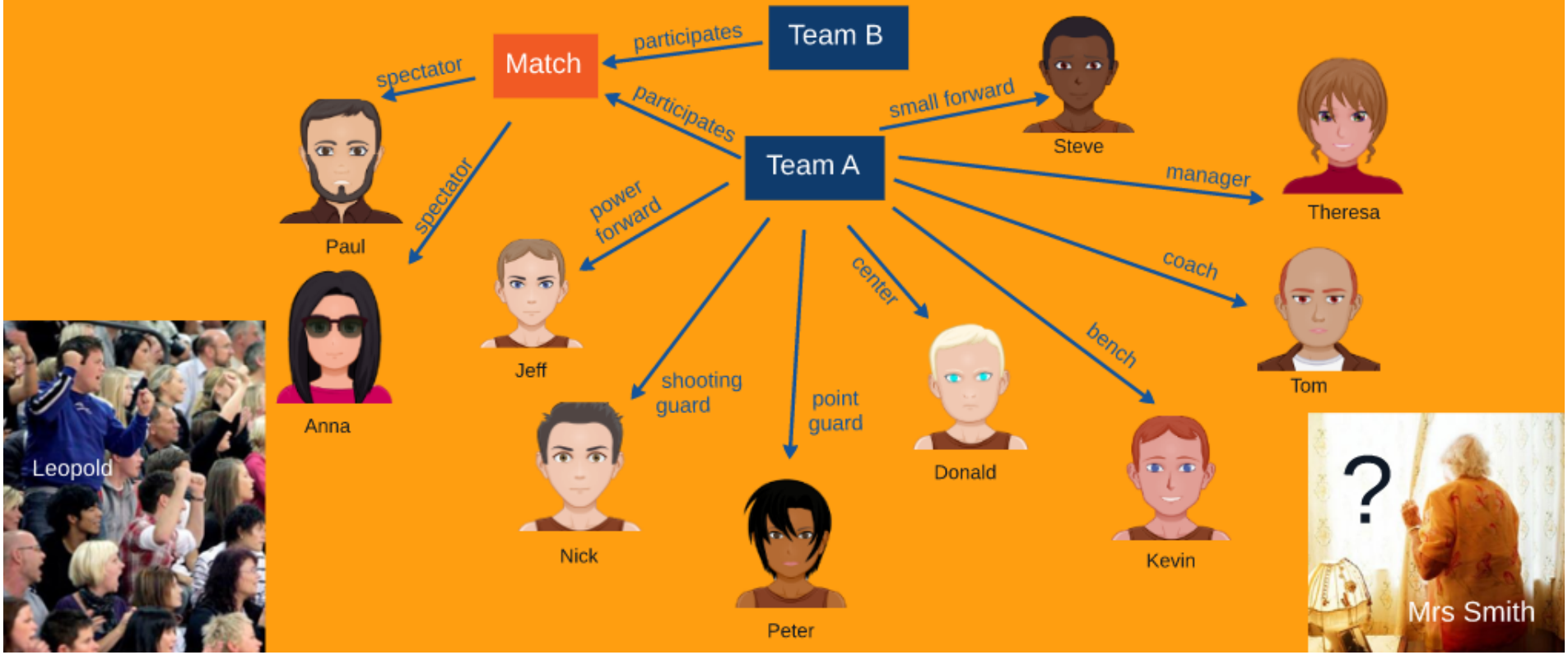

Theresa


Tom

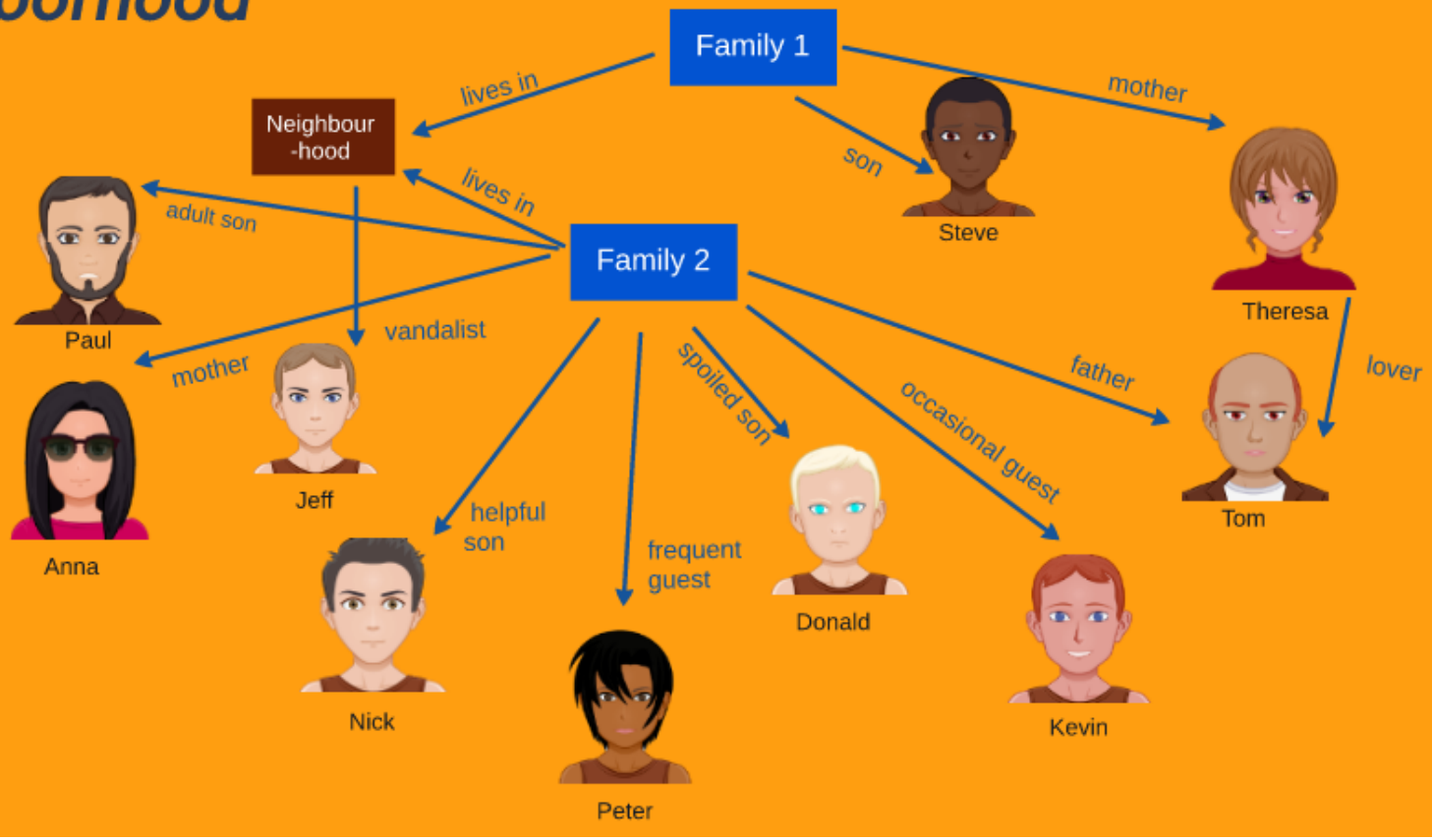
Seen through local sports-court caretaker, Leopold: A junior basketball match



Seen through local sports-court caretaker, Leopold: A junior basketball match



Seen through Mrs Smiths' window: A neighborhood



Takeaways

Same real entities take place in different information models

The models are built using real entities, abstract entities and typed relations

No model is the right one, they are used for different purposes

Different models coexist, simultaneously

Humans use information models to be able to interoperate properly

Information Models

...used by computers...

For advanced operability, we need information models

Examples of information models from the industry:

S-95: Describes general structures in manufacturing companies, like process topologies, material structures and activity structures

PLCOpen: Description of control logic in sequence controlled systems

WITSML: Descriptions of data structures connected to wells from drilling operations

Capability of information modelling in OPC Classic vs OPC UA

140.3

Engineering unit: bara

Name: PT-18-1637A:X

Description: production control valve

123.7

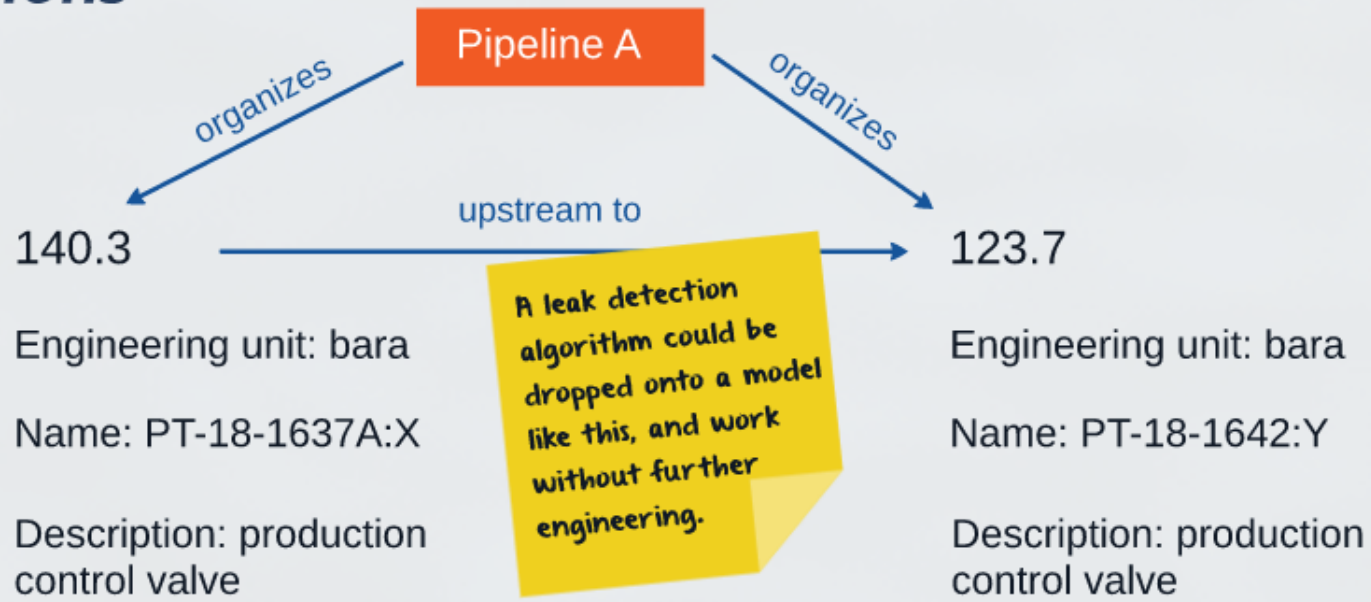
Engineering unit: bara

Name: PT-18-1642:Y

Description: production control valve

OPC classic - numbers with properties

In OPC UA we can add abstract objects and typed relations



Data becomes meaningful with contextual information, targeting the specific purpose of us observing the data

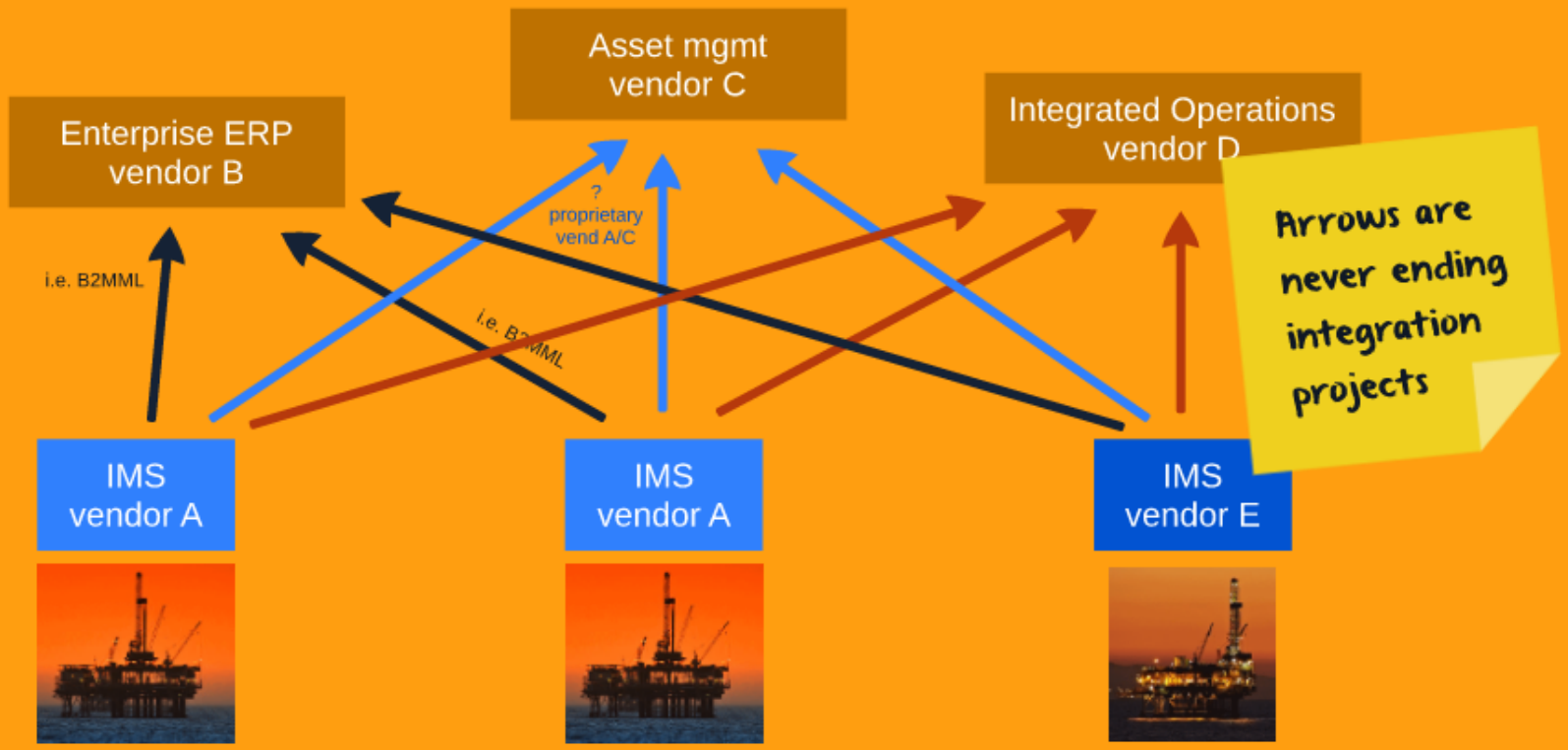
Real-time protocol designed to host foreign information models

Semantic modelling capabilities independent of the protocol stack

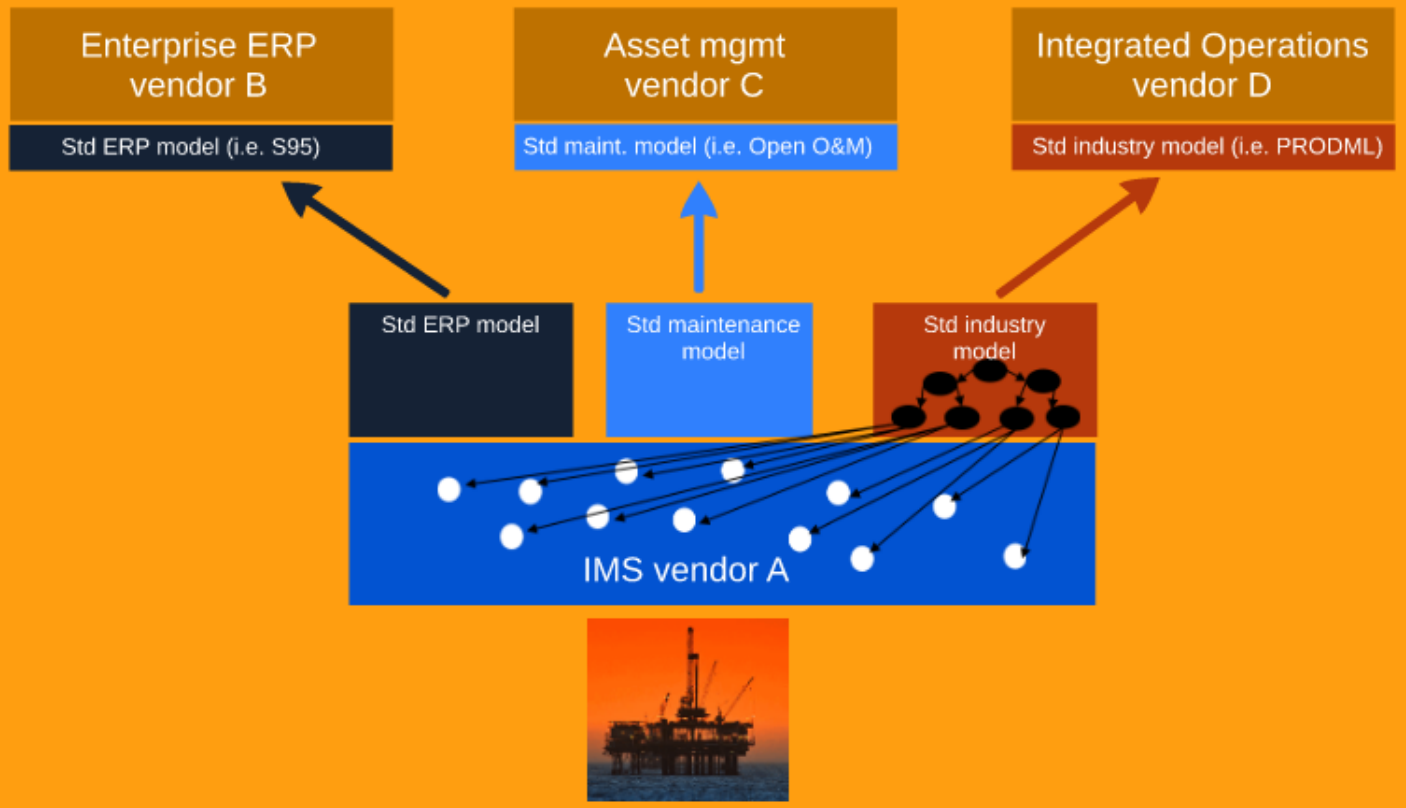
Supports multiple simultaneous models on the same system


Asset heavy industries
...and their interoperability challenges...
Enterprise IT strategies
Interop through standard information models
OPC UA is a solution for Enterprise IT strategies

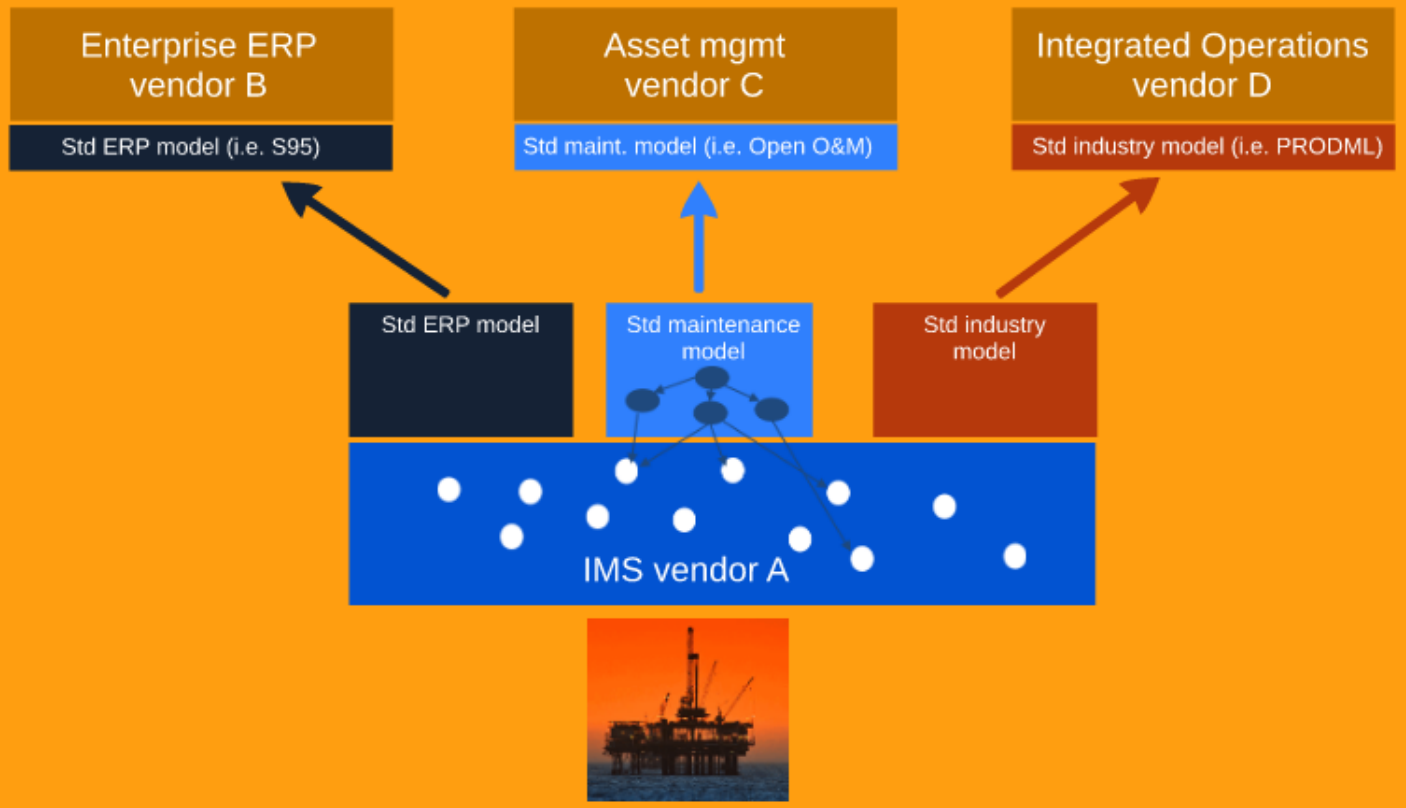
Enterprise IT strategies



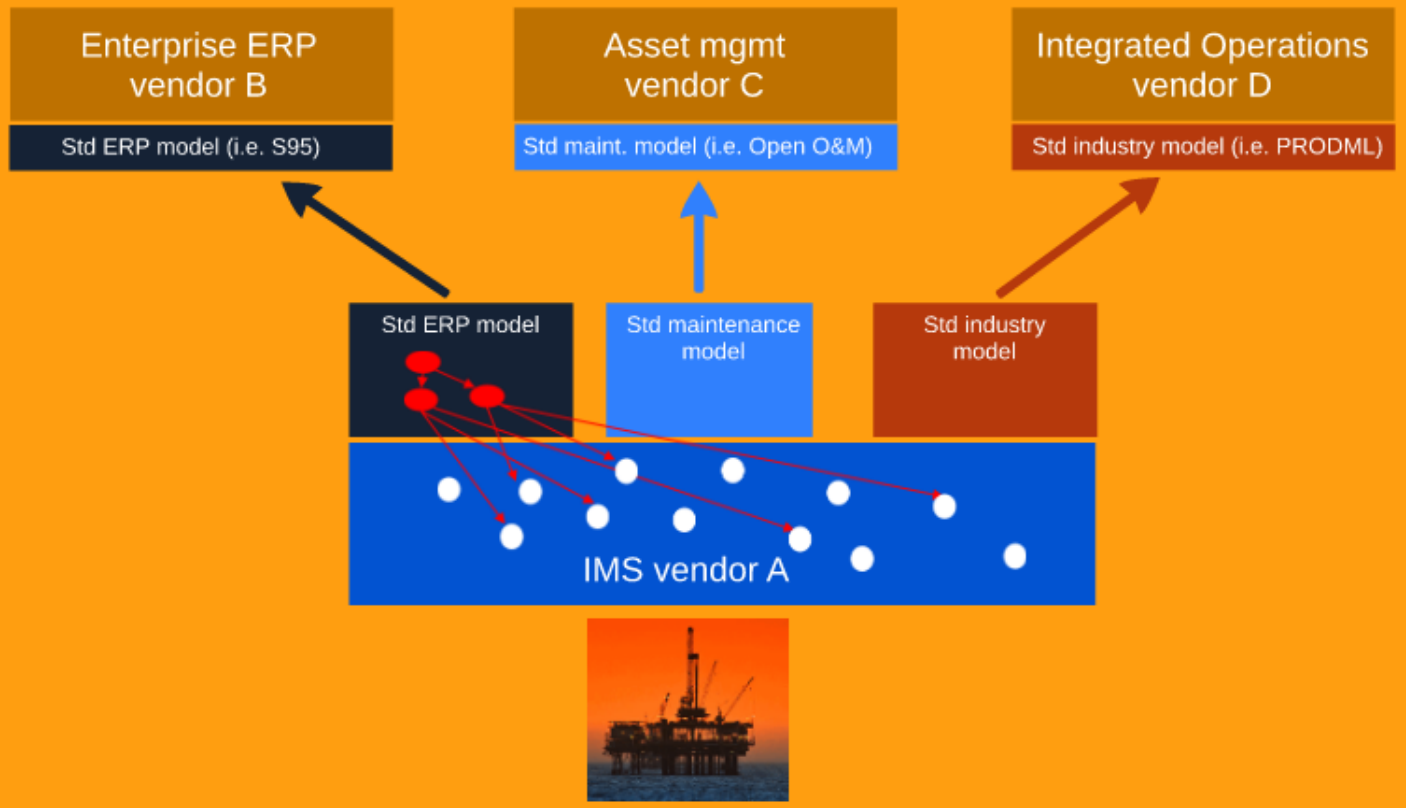
Interop through standard information models



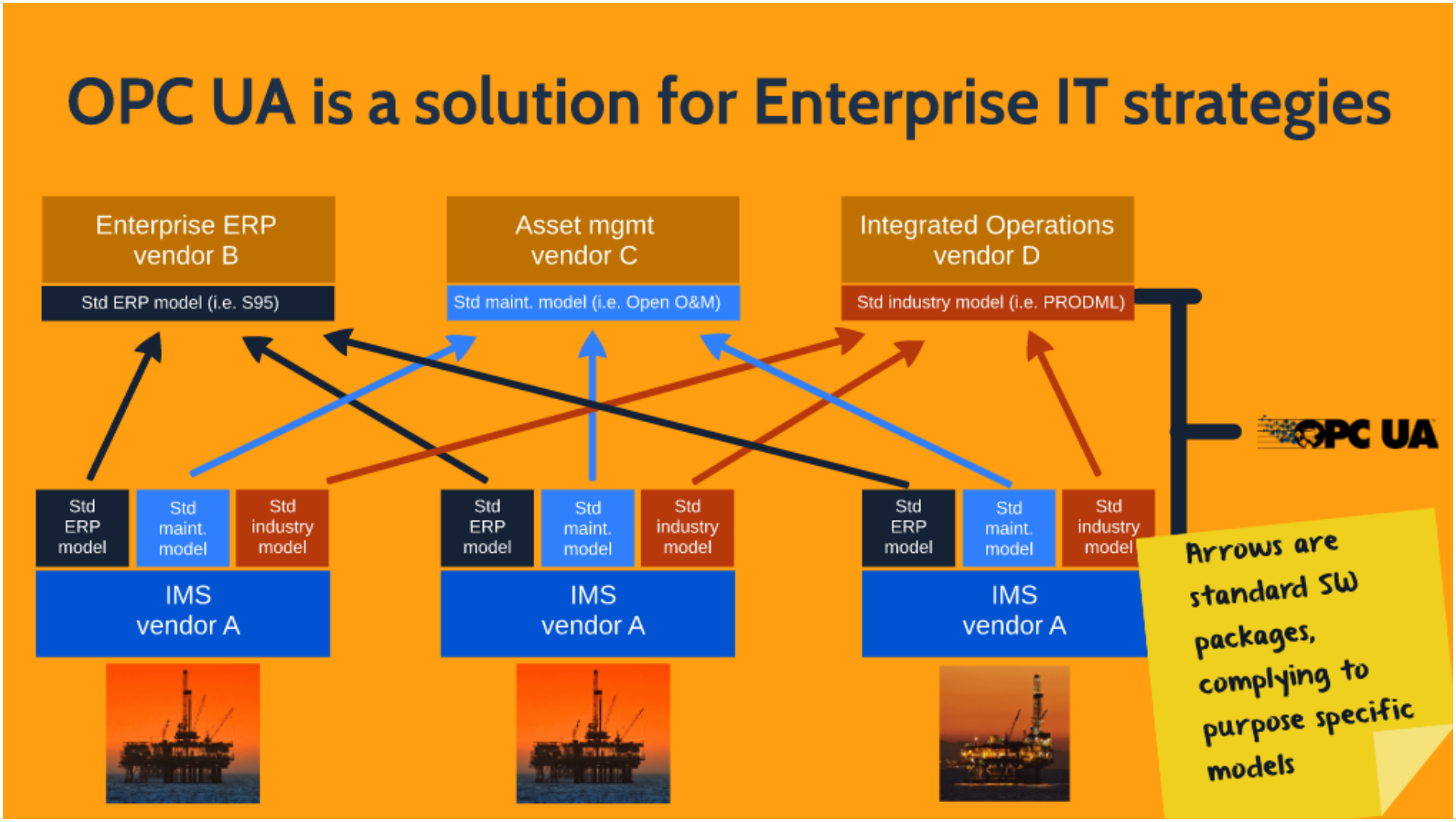
Interop through standard information models



Interop through standard information models



OPC UA is a solution for Enterprise IT strategies



Johan Sverdrup example

Statoil Operates 39 fields built over 40 years, in 37 countries
Johan Sverdrup production to start in 2019
- Phase 1 Investment: 99 BNOK
- Total Investment: 150 BNOK
- Break Even Oil Price: 25 USD
- Estimate 2-3 Billion Barrels Oil (3rd largest on NCS)

OPC UA is underpinning operational technologies in the Johan Sverdrup Project

Use Cases:

- Reducing IT integration cost
- Asset Management
- Automation Management
- Production Optimization

Why OPC UA?

- Member of OPC Foundation
- Standardized models vs company specific models
- OPC UA's support for combination of semantic modelling with realtime values and events
- Security

Prediktor delivery to Johan Sverdrup

APIS OPC UA Gateway, maps productions system to standard OPC UA information models

Prediktor Level 3 - Data Interface GW

Purpose

- Reduce datasource load
- Adding security
- Harmonize protocols

Design capacity

- 10M Nodes
- 50k variables (1/s)
- 500 events/s
- < 10 clients

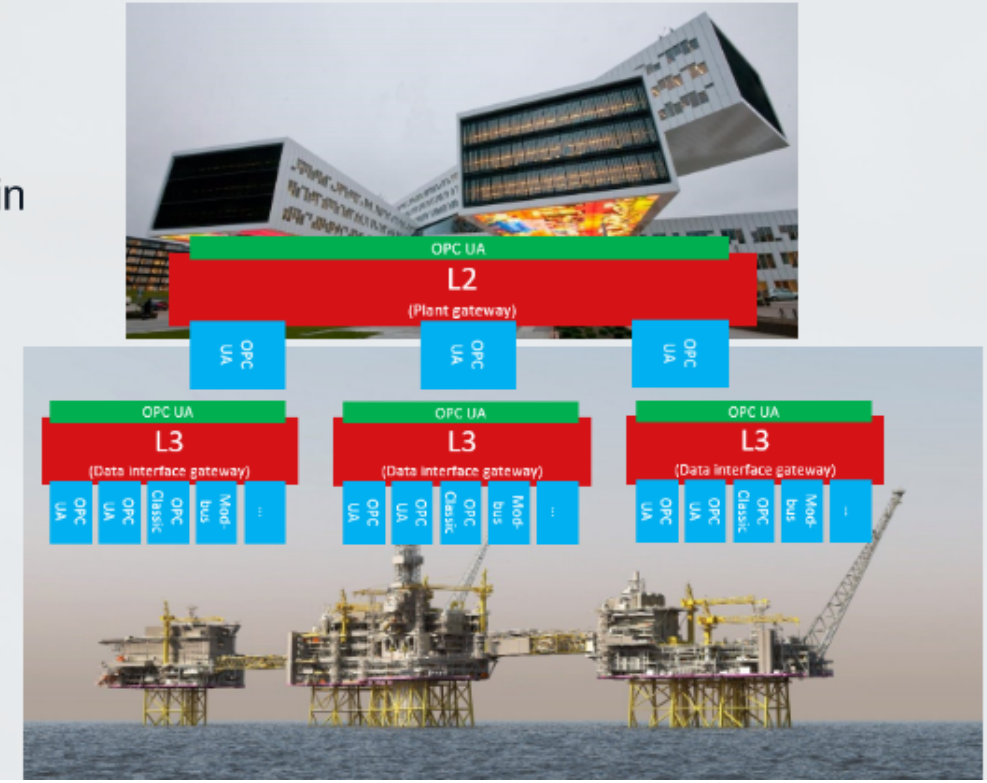
Prediktor Level 2 - Plant GW

Purpose

- Provide access from office domain
- Aggregator for L3 gateways
- Harmonize information models

Design capacity

- 50M Nodes
- 250k variables (1/s)
- 5k events/s
- 20 clients



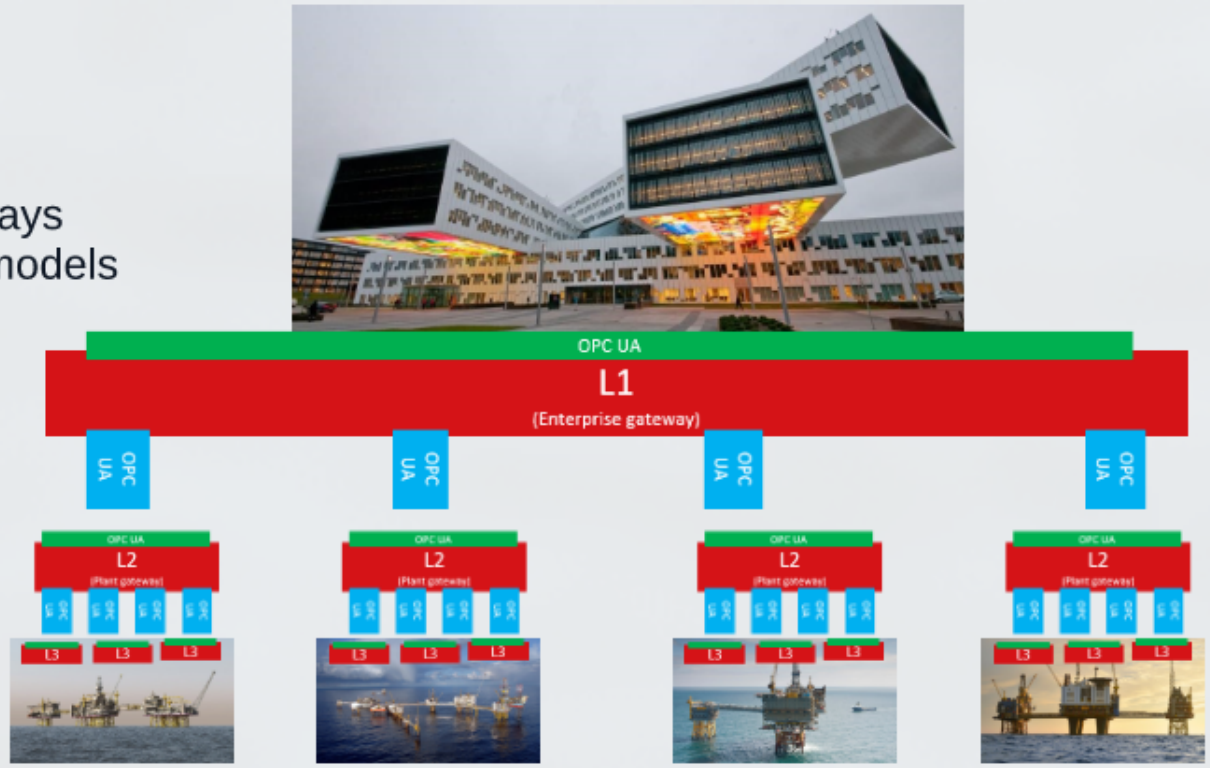
Prediktor Level 1 - Enterprise GW

Purpose

- Aggregator for L2 gateways
- Harmonize information models
- Single point of access

Design capacity

- 2.5B Nodes
- >12.5M variables (1/s)
- 125k events/s
- 500 clients



Pledge

We seek interoperability

- across companies
- across disciplines
- across hierarchies

...for more and more complex purposes...

...with rapidly increasing amounts of data...

Our common focus on establishing purpose specific information models, is a (the only?) sensible strategy to cope with the challenge

Espen Krogh
Prediktor BOD Chairman

espen@prediktor.no
www.prediktor.no

OPC UA

Brings semantic models to real-time interoperability

Prediktor

Brings OPC UA tools for enterprise interoperability