# Getting Started

# Getting Started

- **Overview of OPC Foundation, OPC UA and IJT**
  - https://opcfoundation.org/about/opc-technologies/opc-ua/
  - https://opcfoundation.org/markets-collaboration/IJT/
- **OPC UA IJT Group Presentation:**
  - Refer to the **OPC UA IJT Group Presentation.pdf/pptx.**
  - https://github.com/umati/UA-for-Industrial-Joining-Technologies/tree/main/IJT_Documents
- **Specifications/Online Reference**
  - **Joining:** https://reference.opcfoundation.org/nodesets?u=http://opcfoundation.org/UA/IJT/Base/
  - **Tightening:** https://reference.opcfoundation.org/nodesets?u=http://opcfoundation.org/UA/IJT/Tightening/
- **OPC UA IJT Prototypes/Reference Implementations**
  - https://github.com/umati/UA-for-Industrial-Joining-Technologies

# OPC UA for Industrial Joining Technologies (IJT)

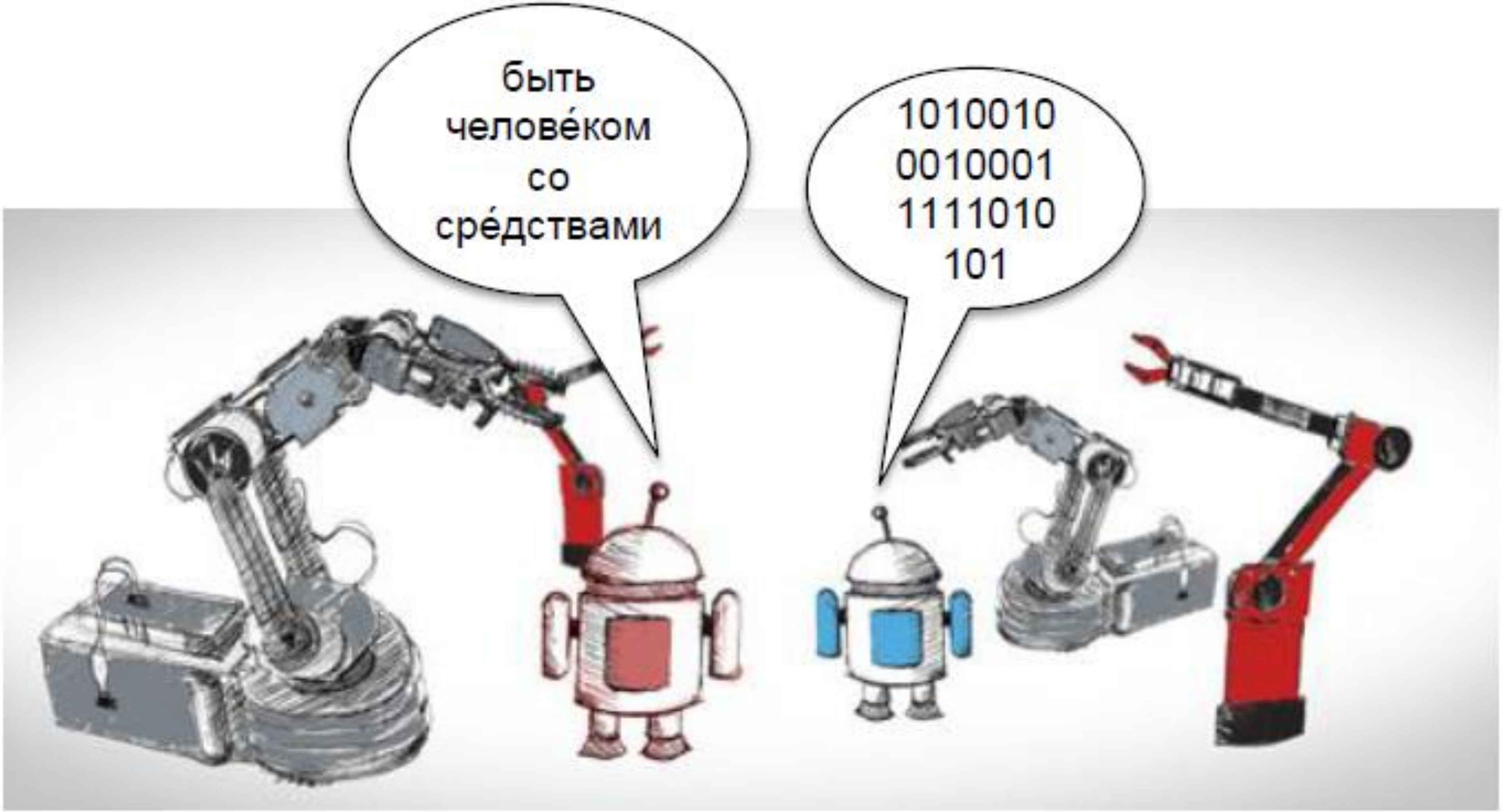
Joining and Tightening System Overview

# Agenda

- » OPC UA Overview
- » Industrial Joining Technologies (IJT) Overview
- » IJT Use Cases
- » Technical Overview of Models
- » Demonstration
- » Questions

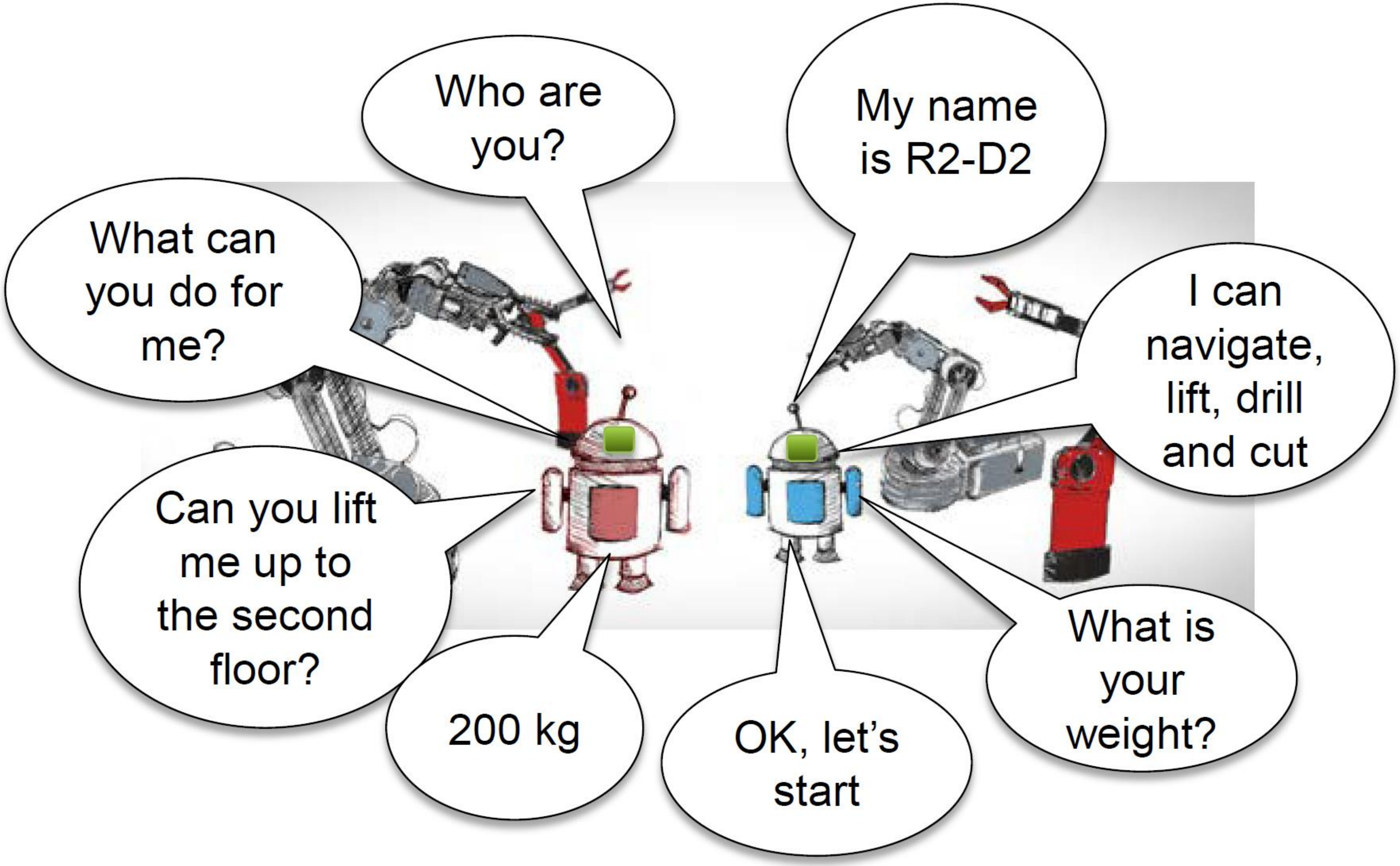
# Quick OPC UA and IJT Overview

# Interoperability Issues



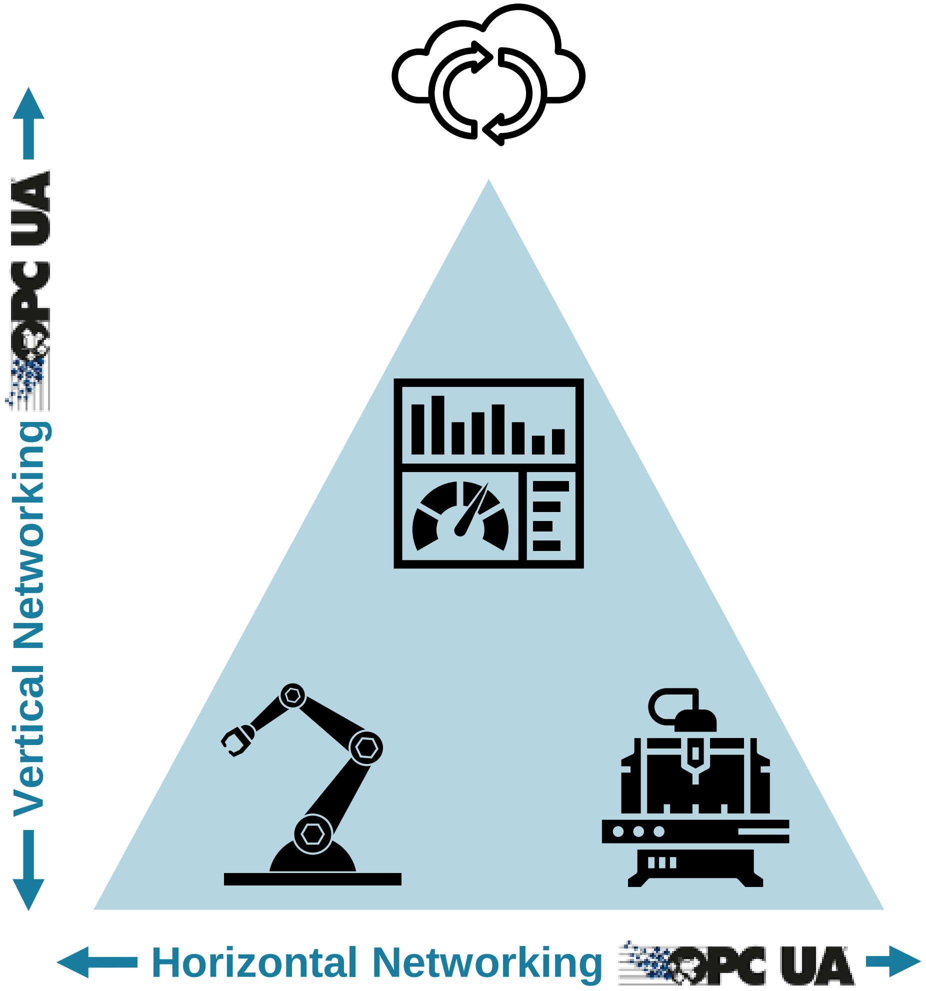
Source: OPC Foundation

# Interoperability Solution



Source: OPC Foundation

# OPC UA – Field to Cloud

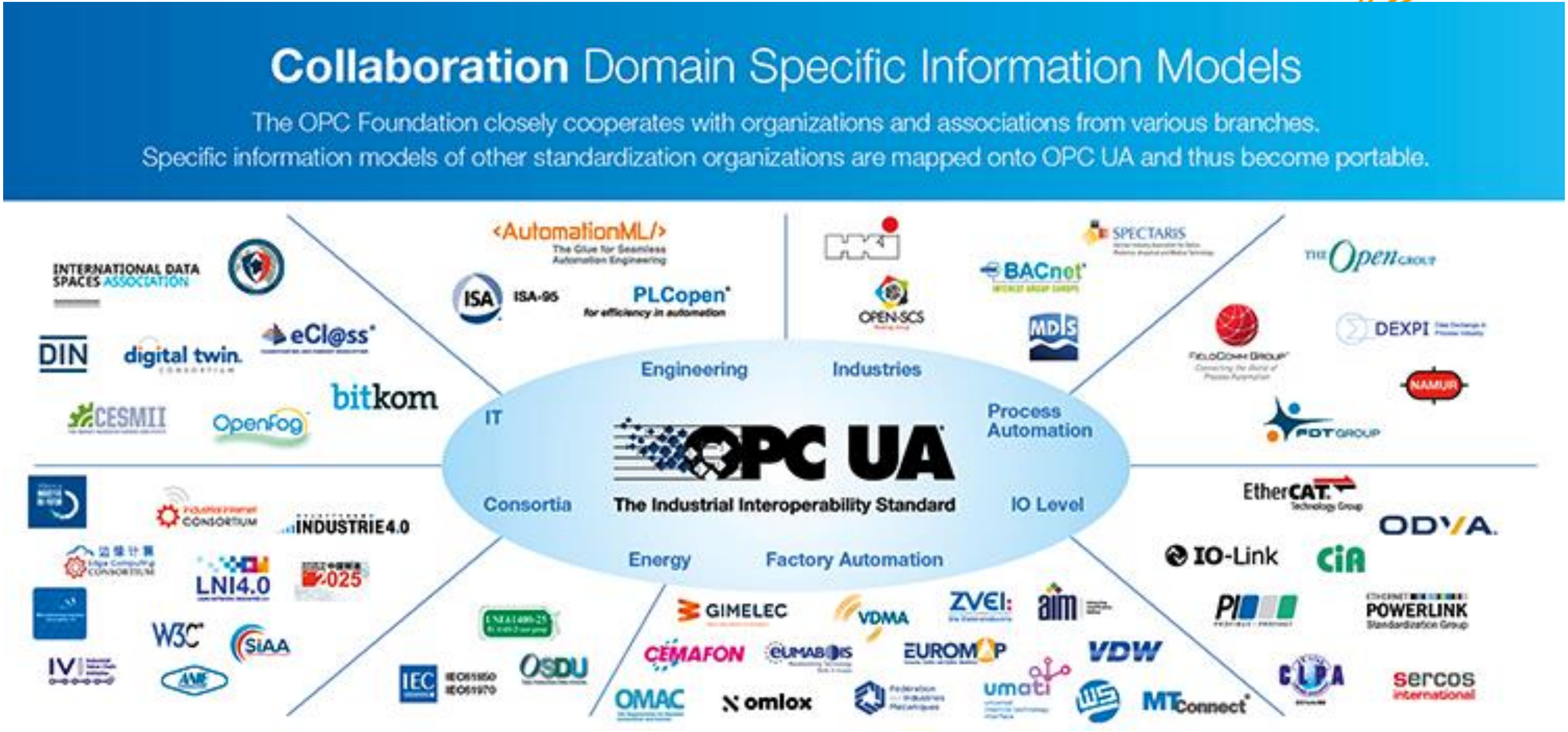


- Open source
- Security
- Two transport mechanisms with various protocols
- Scalable
- Global acceptance
- Semantic information models

# OPC UA Overview Summary

- **ONE – Information Model**
  - Object-oriented, flexible, and extendable.
  - Domain-specific models.
- **TWO – Communications**
  - Client/Server – service-oriented, request/response, on demand.
  - Publish/Subscribe – multicast, unidirectional, cyclic.
- **THREE – Protocols**
  - UA-TCP – TCP/IP based, HTTP/HTTPS, UA Binary, Port 4840 (LDS).
  - UADP – UDP based, UA Binary, TSN deterministic.
  - UADP – MQTT/AMQP based, JSON, Cloud, optional broker.
- **PLUS – Discovery and Security**

# Collaboration Domain Specific Information Models



## Collaboration Domain Specific Information Models

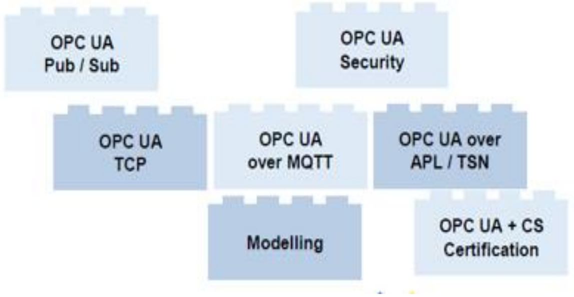
The OPC Foundation closely cooperates with organizations and associations from various branches.
Specific information models of other standardization organizations are mapped onto OPC UA and thus become portable.

Source: OPC Foundation

# Interoperability Solution Summary

## ( OPC UA + Companion Specs ) = Promise for Interoperability

- **OPC UA: Collection of technology bricks**
  - Connectivity, different protocols
  - Security
  - …



OPC UA Pub / Sub

OPC UA Security

OPC UA TCP

OPC UA over MQTT

OPC UA over APL / TSN

Modelling

OPC UA + CS Certification

**HOW to communicate**

“speak the same language”

+

- **Companion Specifications: Collection of bricks for different markets**
  - Information modelling to describe the specific market
  - …

\=

**WHAT to communicate**

“use the same dictionary”

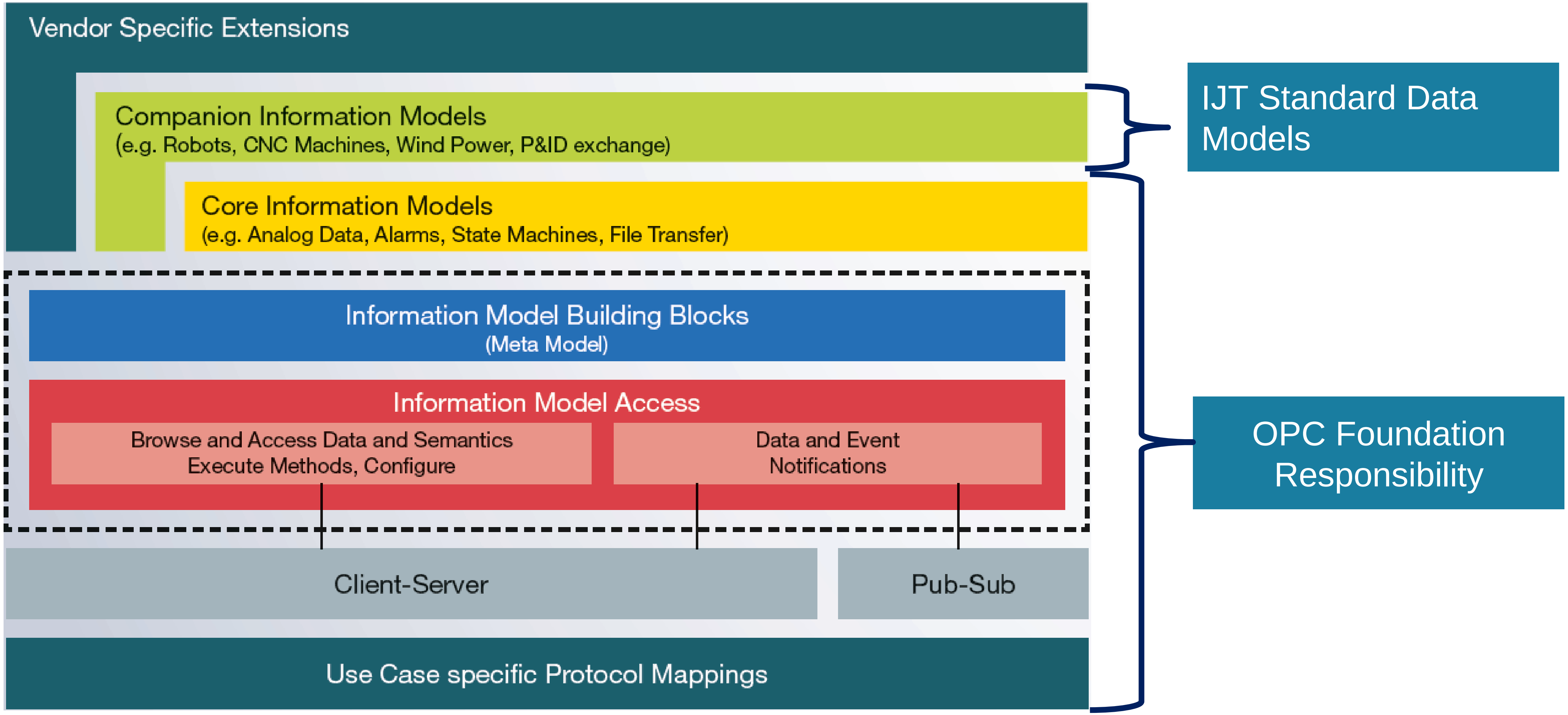
- **OPC UA + Companion Specification drives towards Interoperability**
  - Mandatory bricks guarantee interoperability
  - Optional bricks allow flexibility
  - …

semantic interoperability

**“understand each other”**

Source: OPC Foundation

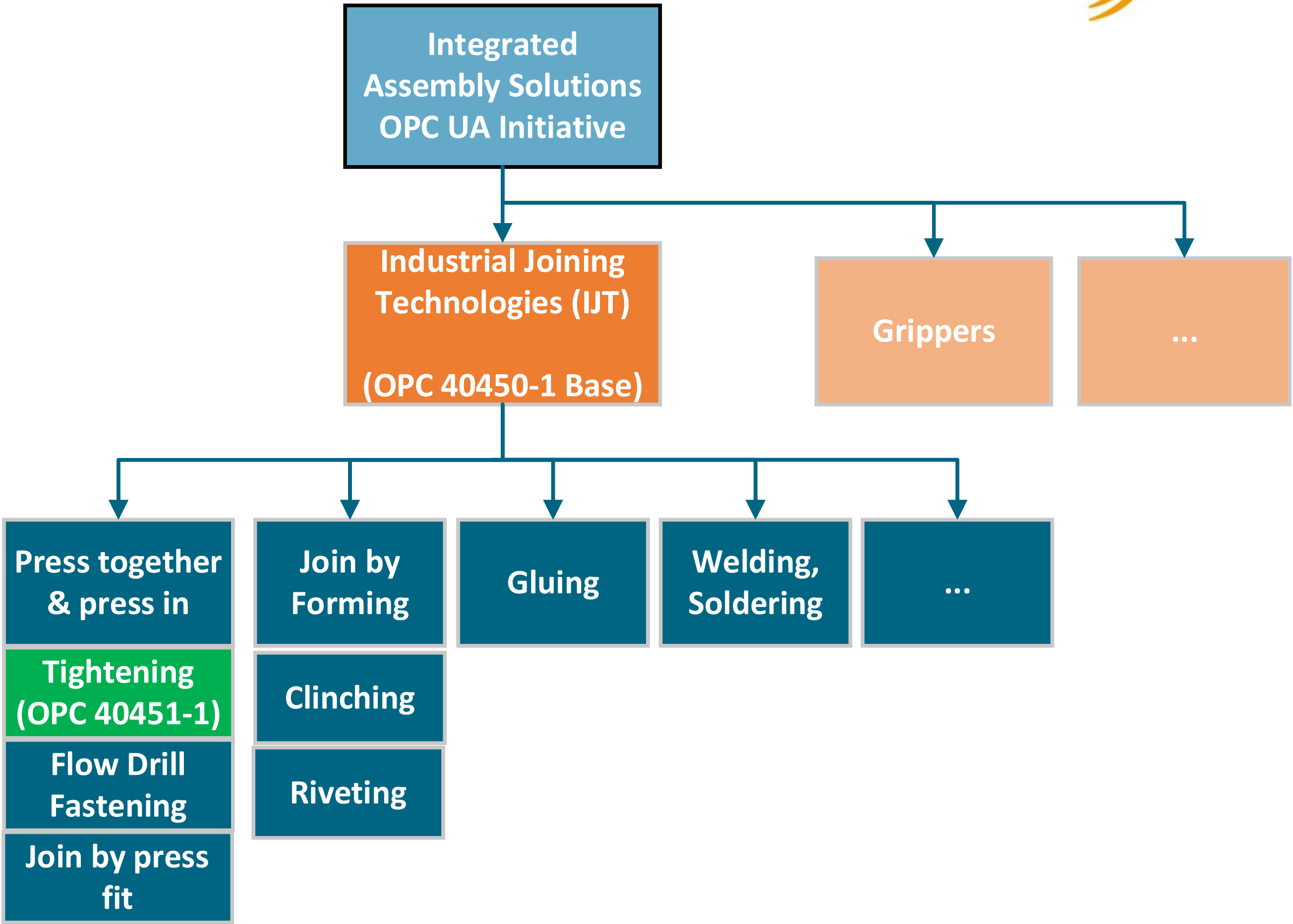
# OPC UA Layered Architecture


Vendor Specific Extensions
Companion Information Models
(e.g. Robots, CNC Machines, Wind Power, P&ID exchange)
Core Information Models
(e.g. Analog Data, Alarms, State Machines, File Transfer)
Information Model Building Blocks
(Meta Model)
Information Model Access
Browse and Access Data and Semantics
Execute Methods, Configure
Data and Event
Notifications
Client-Server
Pub-Sub
Use Case specific Protocol Mappings
IJT Standard Data Models
OPC Foundation Responsibility

Source: OPC Foundation

# Industrial Joining Technologies (IJT) Overview

- History
- Taxonomy
  - https://opcfoundation.org/markets-collaboration/IJT/
- Long term vision
  - Initially started with Tightening but was interesting to see the base elements useful for other joining systems.
- Status
- **OPC 40450-1 UA** for Joining Systems
  - **OPC 40451-1 UA** for Tightening Systems
  - …



# IJT Working Group Members

VDMA

Atlas Copco

rexroth
A Bosch Company

WEBER

OPC
FOUNDATION

Cleco®
Production Tools

Desoutter

DEPRAG
machines unlimited

STÖGER
AUTOMATION

CSP
EFFICIENCY COMES TRUE

SCS
Concept Group

PLARAD®
Torque & Tension Systems

EST
TECHNOLOGIE

XITASO

VISION
DEVICES

MAFU
AUTOMATION

AMT
Product of EST

# Reuse of Harmonized Specifications

| NamespaceUri | Description | Use | Namespace Index | Example |
|---|---|---|---|---|
| http://opcfoundation.org/UA/ | OPC UA Base | Mandatory | 0 | 0:EngineeringUnits |
| http://opcfoundation.org/UA/DI/ | OPC UA for Devices (OPC 10000-100). | Mandatory | 2 | 2:DeviceRevision |
| http://opcfoundation.org/UA/AMB/ | OPC UA for Asset Management Basics (OPC 10000-110). | Mandatory | 3 | 3:IRootCauseIndicationType |
| http://opcfoundation.org/UA/Machinery/ | OPC UA for Machinery Basic Building Blocks (OPC 40001-1). | Mandatory | 4 | 4:MachineIdentificationType |
| http://opcfoundation.org/UA/Machinery/Result/ | OPC UA for Machinery Result Transfer (OPC 40001-101). | Mandatory | 5 | 5:ResultManagementType |

# Use Cases Overview

### Asset Management

Overview and Identification of physical assets in the given system.

**Example:**

Manufacturer, Serial number, Software Revision…

### Condition Monitoring

Acquisition and processing of information that indicate the state of an asset over time.

**Example:**

Health status, temperature…

### Result Management

Primary process output of the tightening operation.

**Example:**

Single Result, Batch Result, Job Result, Multi-spindle Result, etc.

### Event Management

Various types of events with standard payload and filter criteria.

**Example:**

Tool Connected, Maintenance Events, etc.

### Commands

**Asset** Management control mechanisms.

**Joining Process** Management.

**Example:**

Select Program, Send Program, Enable Tool, etc.

### Joint Management

Provides joint data.

**Example:**

Joint with associated Programs, etc.

# The Key to Reducing Costs: Data Interoperability

We need…

1. A common **Interface** (Analogy: A Book)

2. A common **Data Format** (Analogy: A Latin Alphabet)

3. A common **Data Model** (Analogy: Then English Language)

4. Common **Semantics** (Analogy: The Plot and Characters of the Novel)

Only when all 4 things are present, we can truly understand each other!

# Interoperability Example for OPC UA-enabled Assets

1. **Interface:** OPC UA Client/Server

2. **Data Format:** OPC UA Binary or JSON

3. **Data Model:** OPC UA Information Model

4. **Semantics:** OPC UA IJT Companion Specification

# Summary

- **Overview of OPC Foundation, OPC UA and IJT**
  - https://opcfoundation.org/about/opc-technologies/opc-ua/
  - https://opcfoundation.org/markets-collaboration/IJT/
- **OPC UA IJT Group Presentation:**
  - Refer to the **OPC UA IJT Group Presentation.pdf/pptx.**
  - https://github.com/umati/UA-for-Industrial-Joining-Technologies/tree/main/IJT_Documents
- **Specifications/Online Reference**
  - **Joining:** https://reference.opcfoundation.org/nodesets?u=http://opcfoundation.org/UA/IJT/Base/
  - **Tightening:** https://reference.opcfoundation.org/nodesets?u=http://opcfoundation.org/UA/IJT/Tightening/
- **OPC UA IJT Prototypes/Reference Implementations**
  - https://github.com/umati/UA-for-Industrial-Joining-Technologies

# OPC UA Overview
# (IEC 62541)

# Problem Statement

- **I4.0 Challenges for Communication**
  - Interoperability
  - Security
  - …

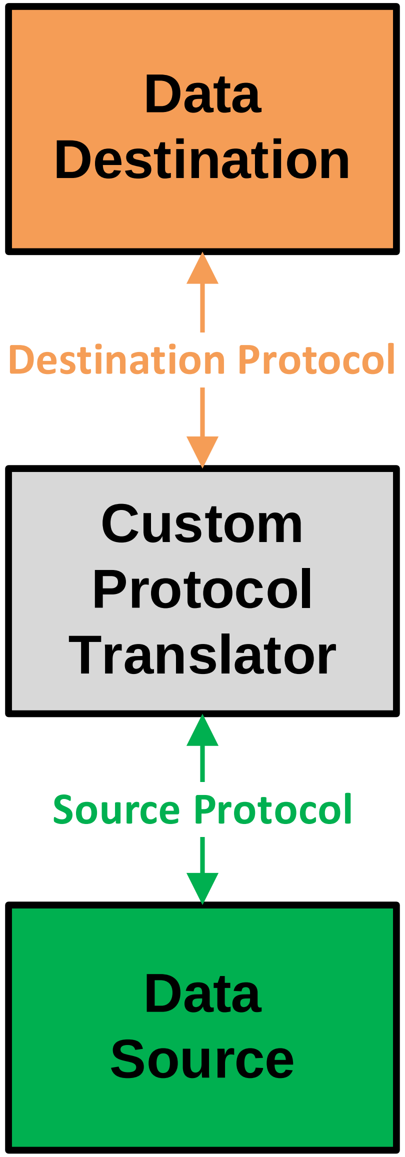
# Traditional Solution – Custom Protocol Converter – Before OPC

**Custom software is built through an extensive collaboration of both system’s vendors.**

- Enabled tightly controlled communication between the source and destination regardless of complexity.

**But it has the following limitations:**

- Vendor Specific
- Costly to produce
- Costly to manage
- Proprietary Data Paths
- Little Integration
- Each application had custom drivers

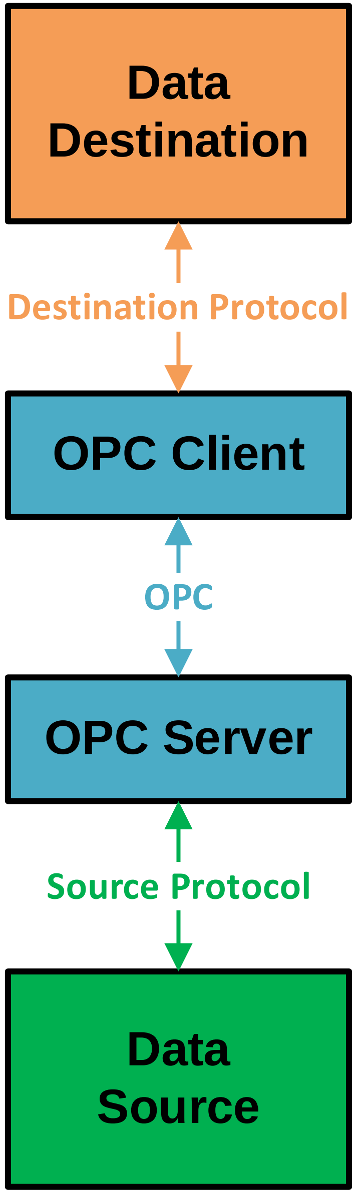


# OPC Connectivity – OPC Classic Data Path

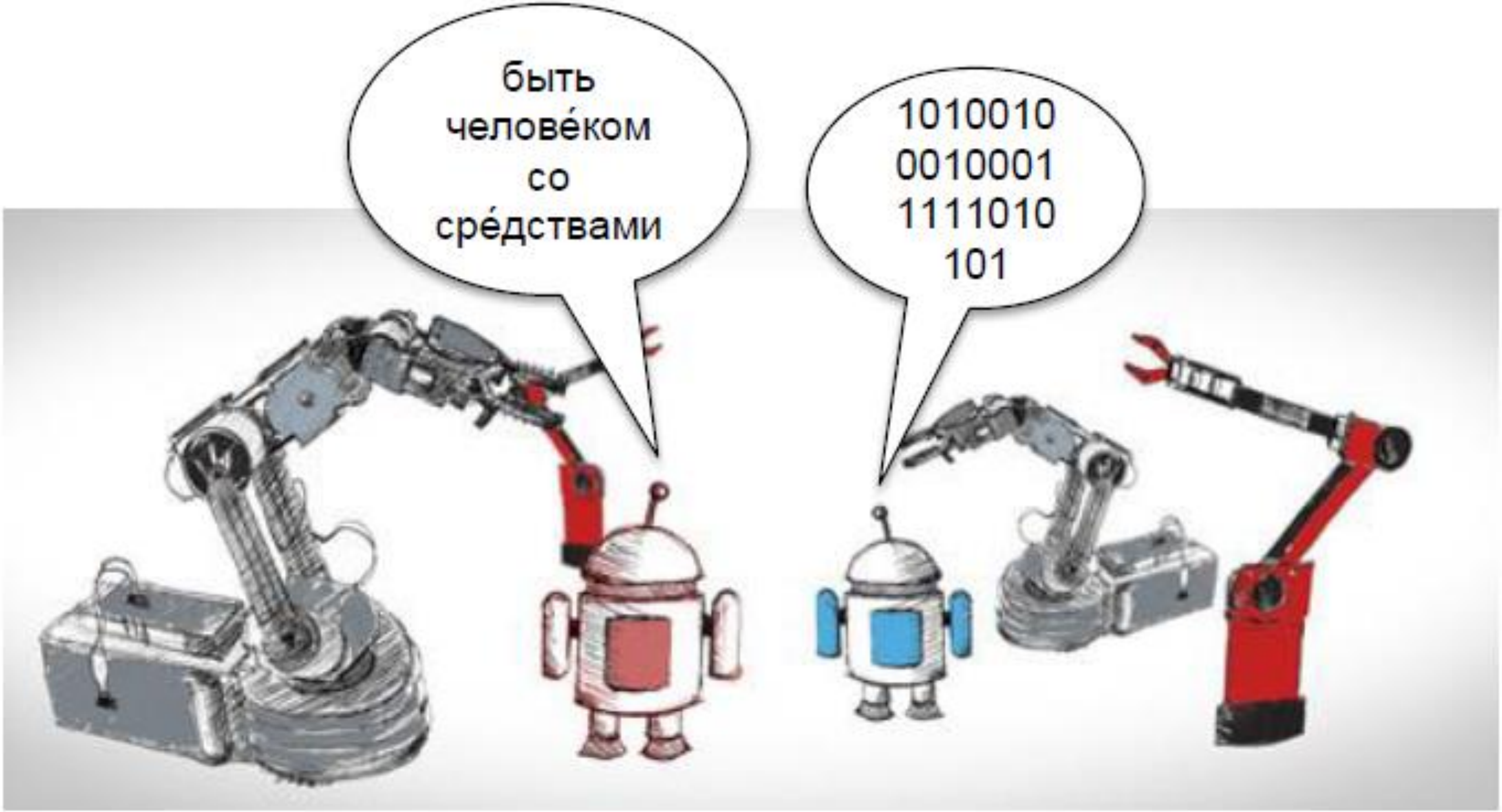
- All OPC Data Paths will consist of these four components.

- More complex architectures will consist of many OPC Data Paths.

- Understanding the OPC Data Path will enable understanding all OPC Solutions
  - Architecting
  - Implementing
  - Troubleshooting

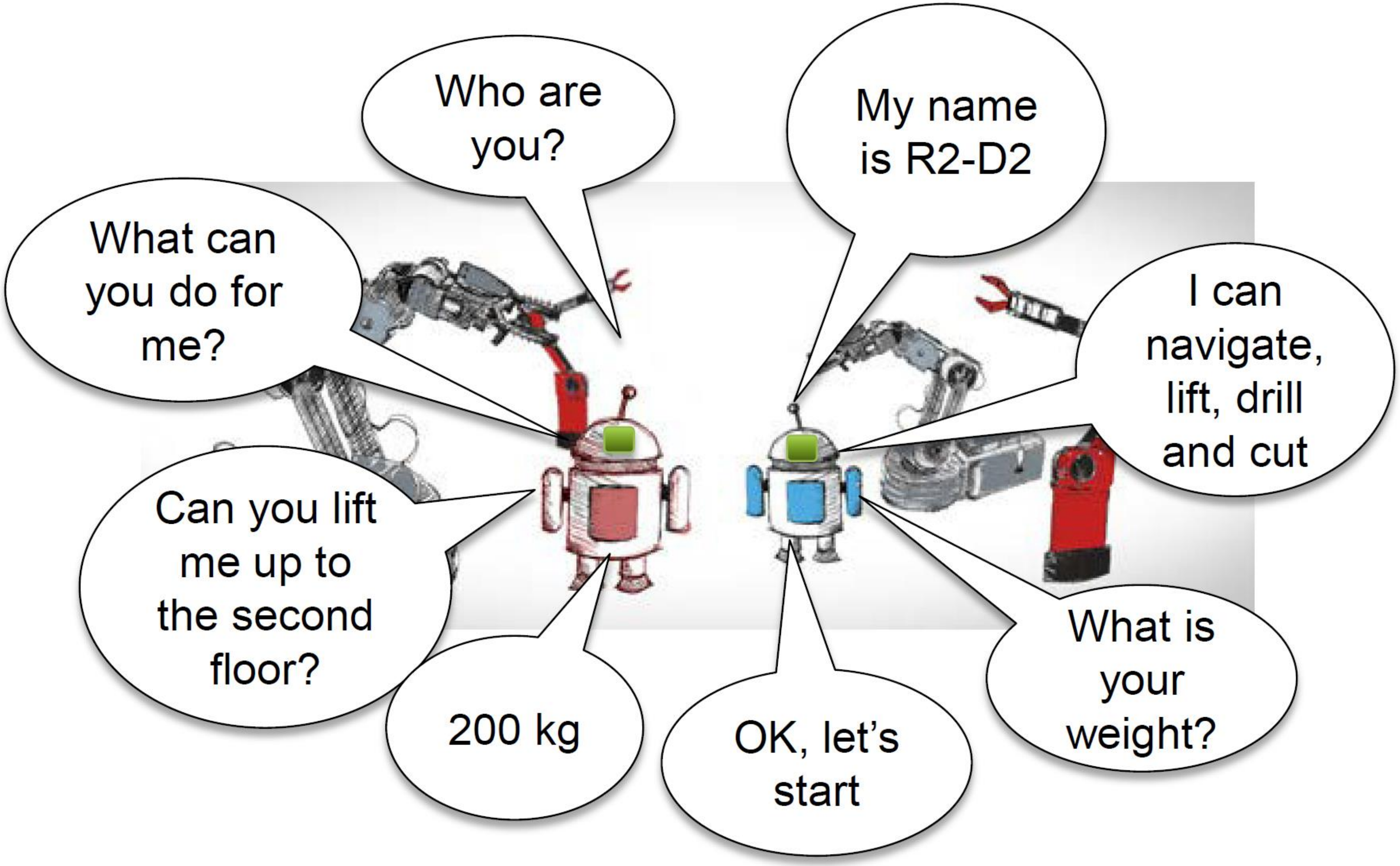


# Today’s Interoperability Issues



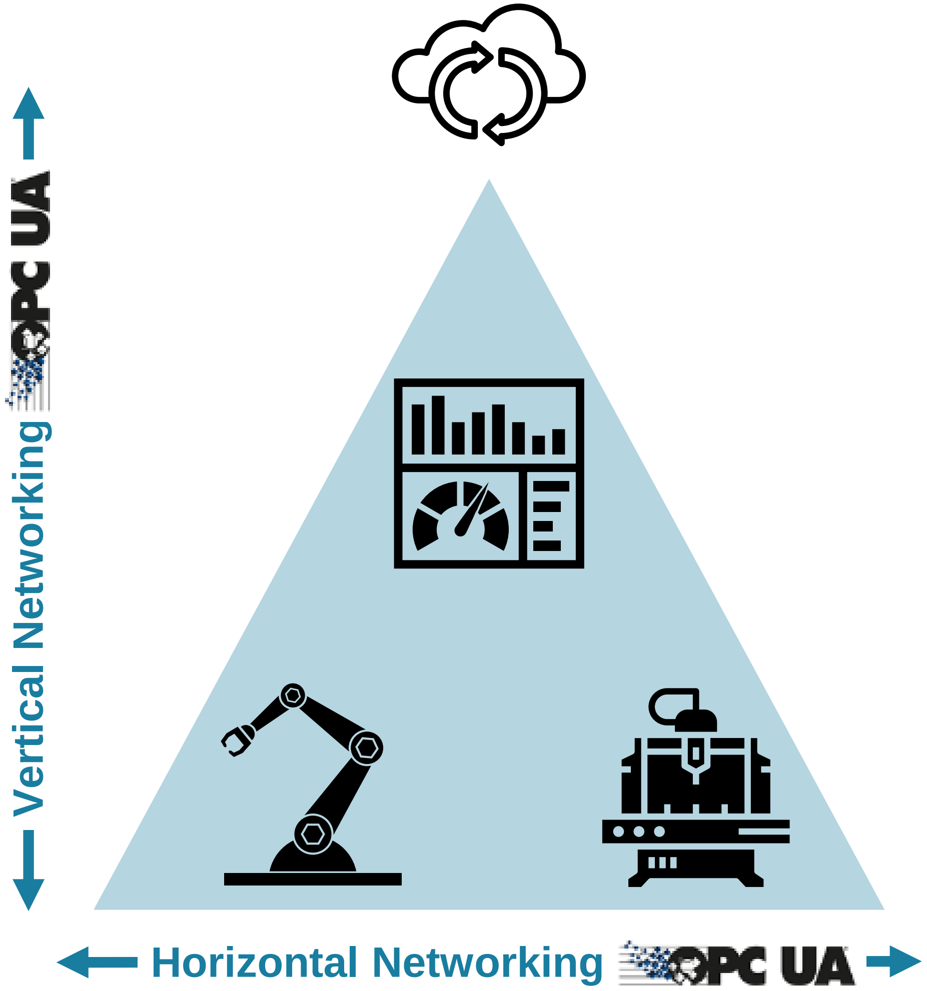
Source: OPC Foundation

# Today’s Interoperability Solution



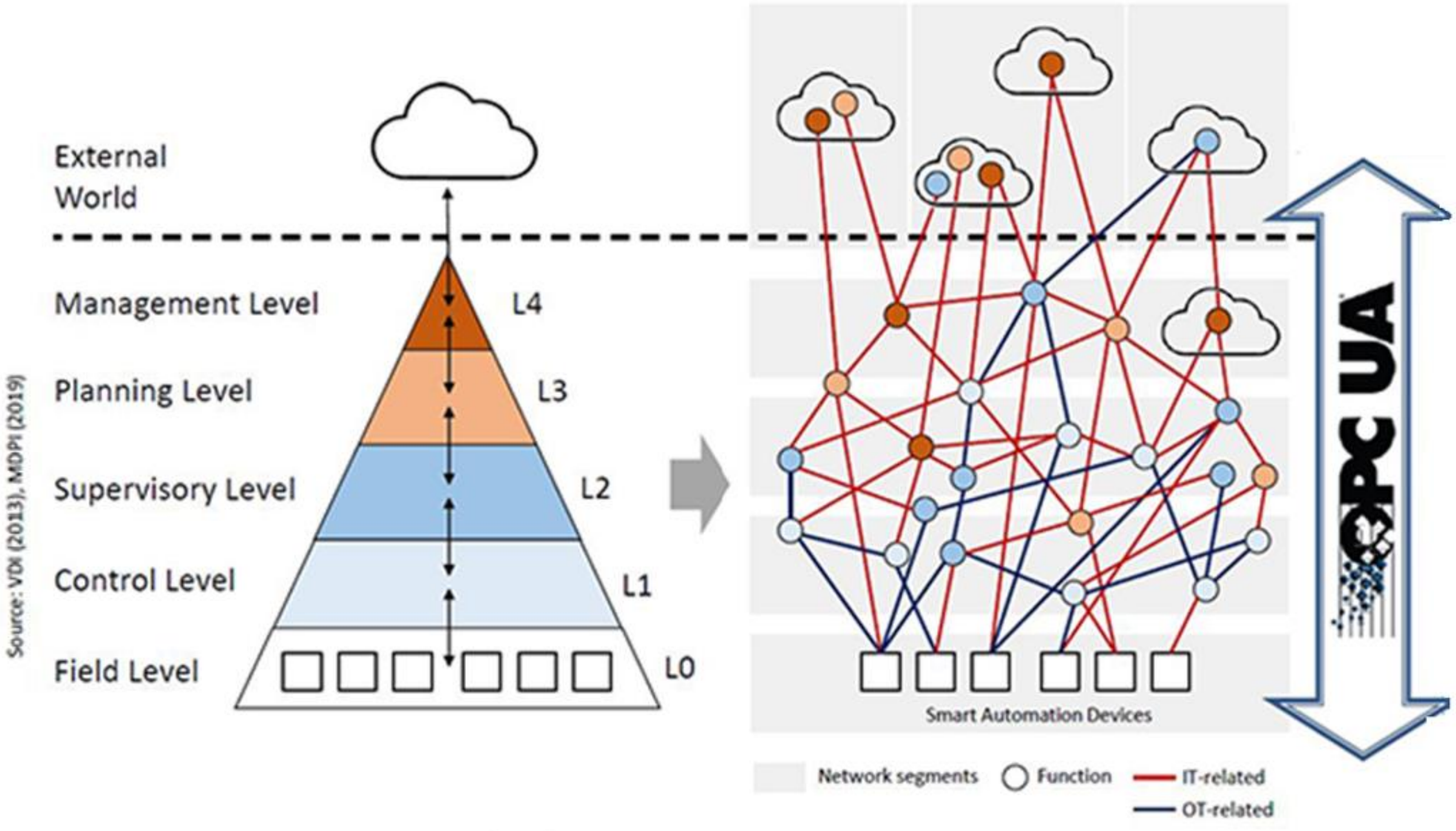
Source: OPC Foundation

# OPC UA – Field to Cloud



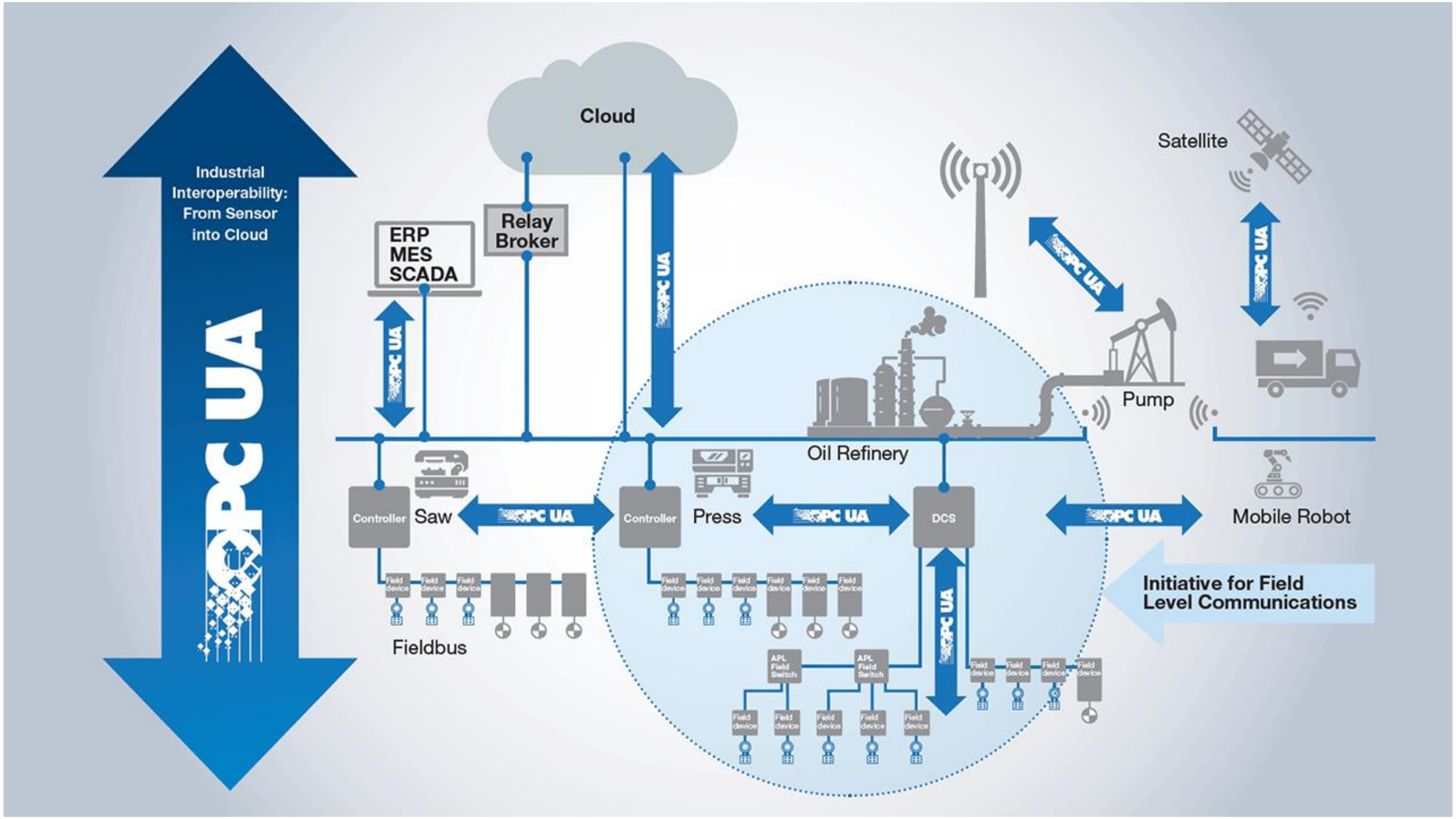
- Open source
- Security
- Two transport mechanism with various protocols
- Scalable
- Global acceptance
- Semantic information models

# From Automation Pyramid to Information Network



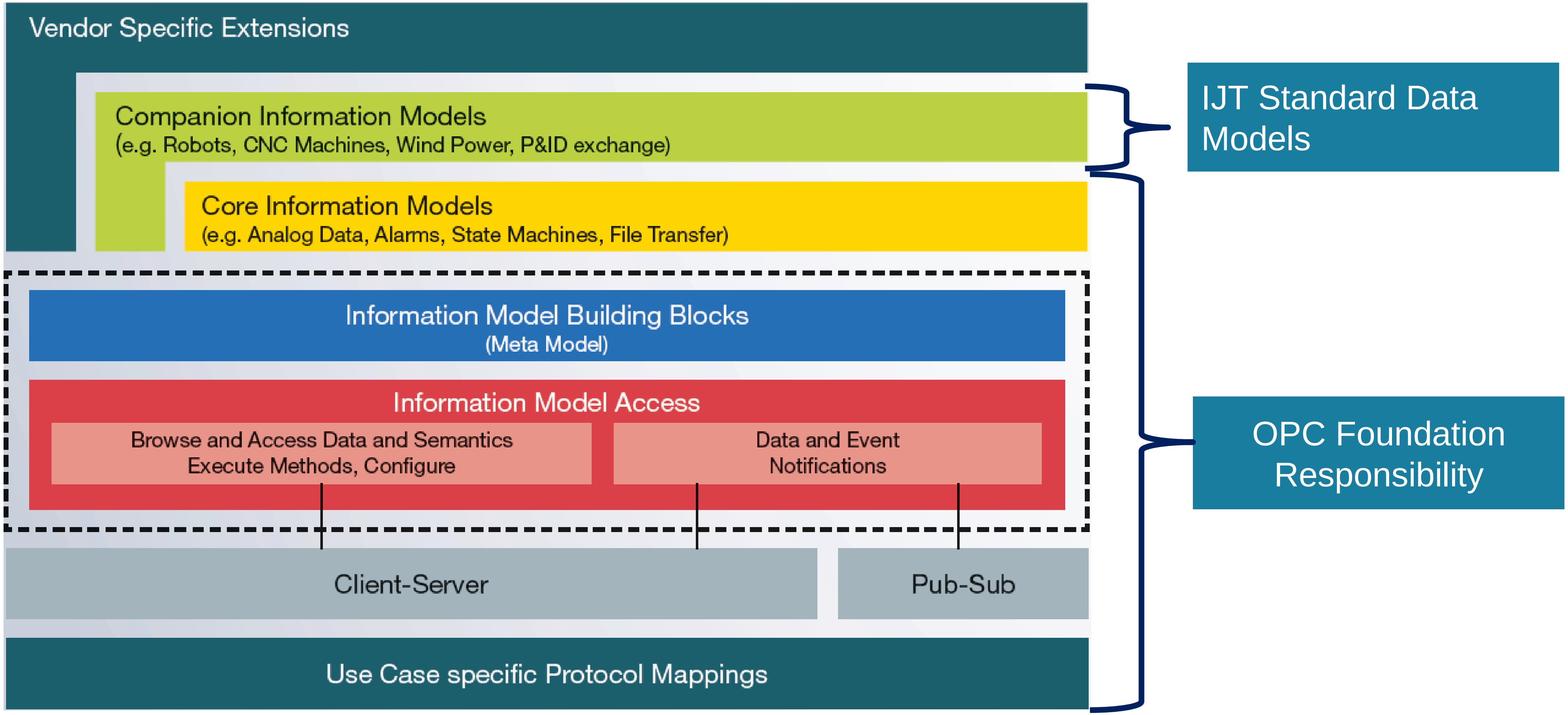
Source: OPC Foundation

# OPC UA Factory Network



Source: OPC Foundation

# OPC UA Layered Architecture


Vendor Specific Extensions
Companion Information Models
(e.g. Robots, CNC Machines, Wind Power, P&ID exchange)
Core Information Models
(e.g. Analog Data, Alarms, State Machines, File Transfer)
Information Model Building Blocks
(Meta Model)
Information Model Access
Browse and Access Data and Semantics
Execute Methods, Configure
Data and Event
Notifications
Client-Server
Pub-Sub
Use Case specific Protocol Mappings
IJT Standard Data Models
OPC Foundation Responsibility

Source: OPC Foundation

# OPC UA Deployment Scenarios

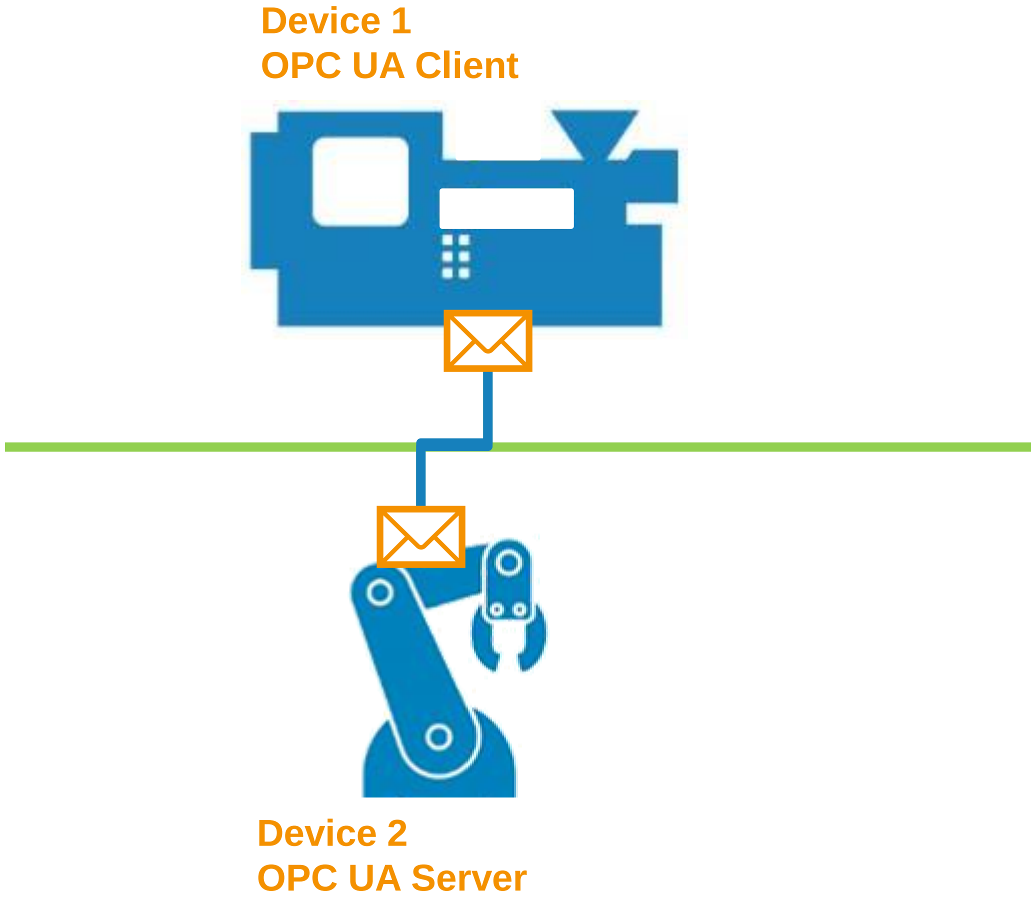
# Platform Independence

## Hardware

## Operating Systems

**R**eal **T**ime **O**perating **S**ystem

# OPC UA Client/Server Architecture



## Client/Server

- » Client accesses information from the server via a permanently configured connection

## Example: Request → Response

- » Device 1 requests information from Device 2
- » Device 2 replies to Device 1

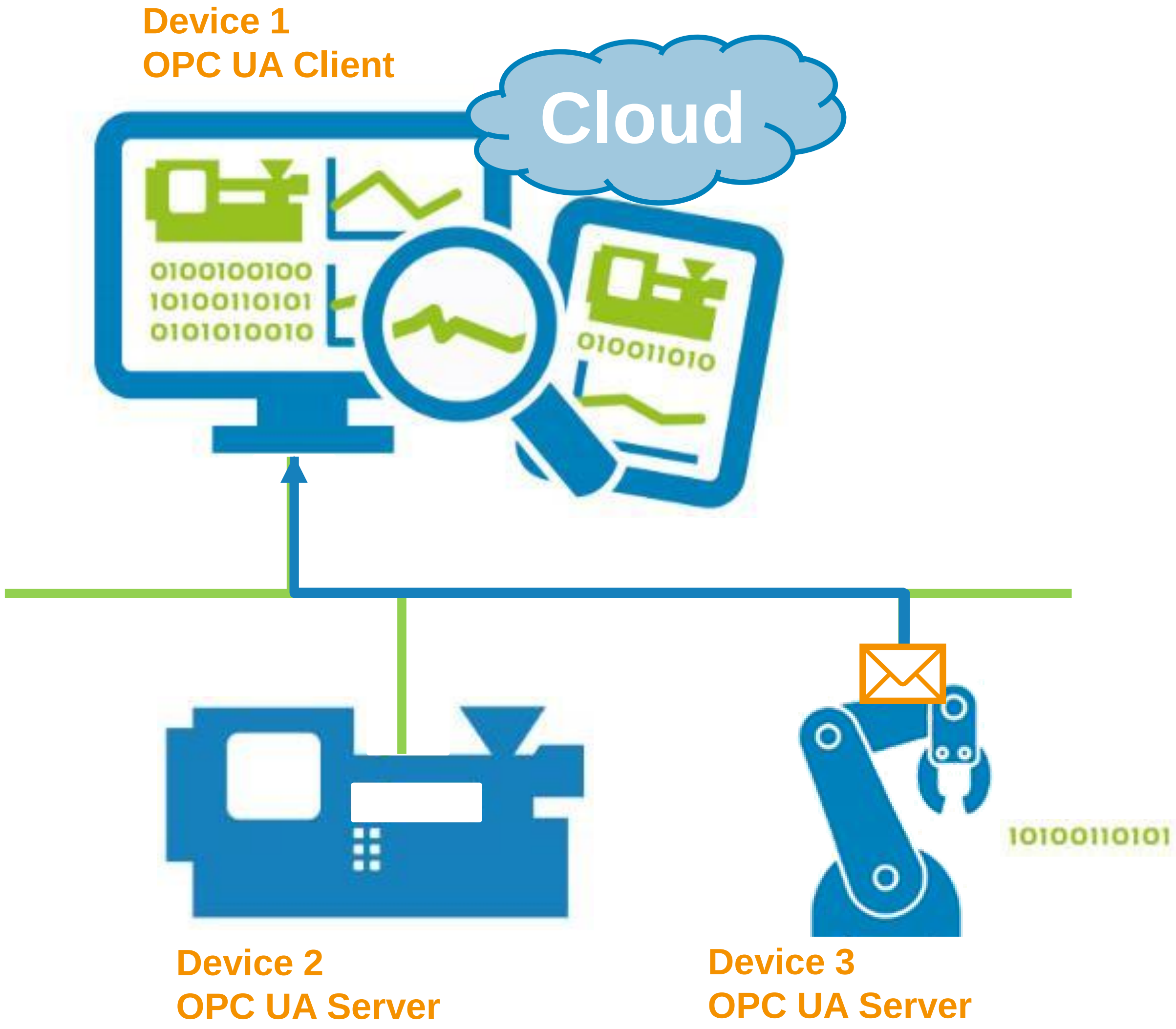
## Analogy

- » Letter service with registered mail and advice of receipt.

Quelle: VDMA Leitfaden Industrie 4.0 Kommunikation mit OPC UA

# OPC UA – Client/Server Architecture

## *Communication via Subscription Notify*



### Client/Server

- » Client accesses information from the server via a permanently configured connection

### Example: Subscription → Notify

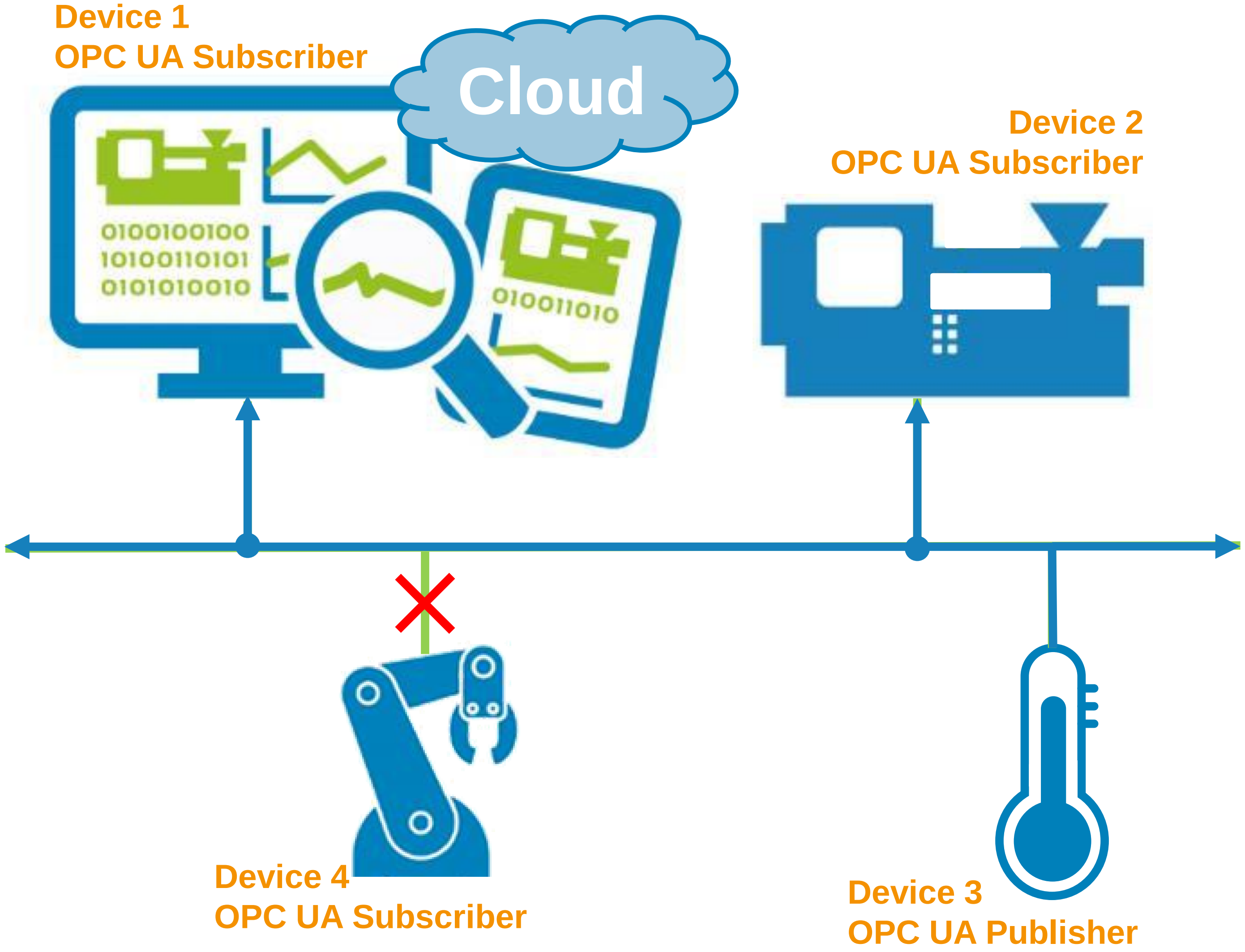
- » Device 1 subscribes to the information from Device 3
- » Device 3 informs Device 1 as soon as information changes

### Analogy

- » Registration to football live ticker with feedback on special events via SMS

Quelle: VDMA Leitfaden Industrie 4.0 Kommunikation mit OPC UA

# OPC UA Pub/Sub Architecture



## Publisher/Subscriber

- » Publisher sends to unknown subscribers without a fixed connection
- » Application example:
  - – 1:n - Sensor publishes data used by different systems

## Example: Pub/Sub

- » Device 3 continuously sends information to the network
- » Device 1 and 2 have subscribed to the information
- » Device 4 has not subscribed to the information

## Analogy

Setting a radio frequency

Quelle: VDMA Leitfaden Industrie 4.0 Kommunikation mit OPC UA

# OPC UA Overview Summary

- **ONE – Information Model**
  - Object-oriented, flexible, and extendable.
  - Domain-specific models.

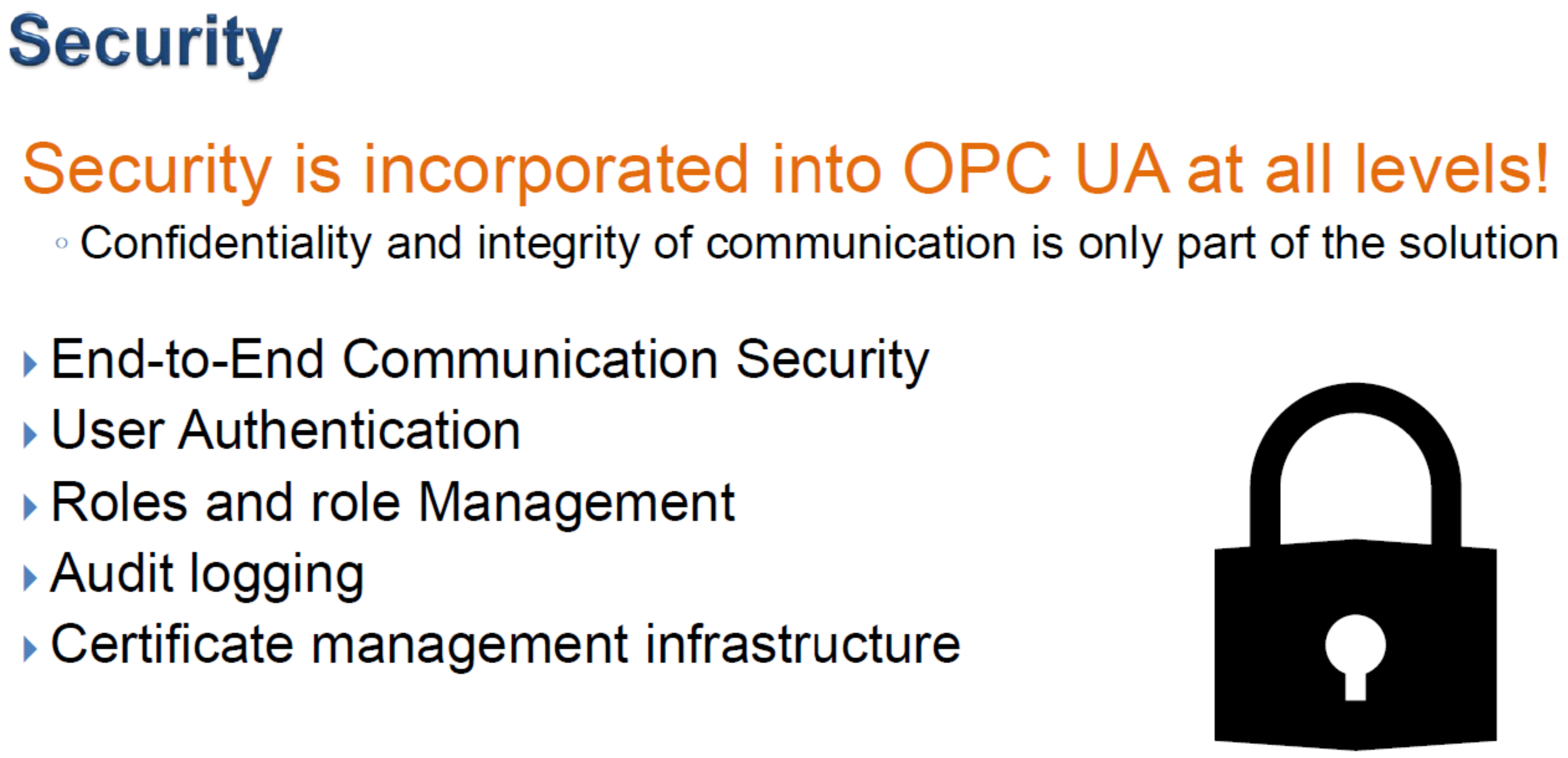
- **TWO – Communications**
  - Client/Server – service-oriented, request/response, on demand.
  - Publish/Subscribe – multicast, unidirectional, cyclic.

- **THREE – Protocols**
  - UA-TCP – TCP/IP based, HTTP/HTTPS, UA Binary, Port 4840 (LDS).
  - UADP – UDP based, UA Binary, TSN deterministic.
  - UADP – MQTT/AMQP based, JSON, Cloud, optional broker.

- **PLUS – Discovery and Security**

# OPC UA Security

# OPC UA Security



## Security

Security is incorporated into OPC UA at all levels!

- Confidentiality and integrity of communication is only part of the solution

- End-to-End Communication Security
- User Authentication
- Roles and role Management
- Audit logging
- Certificate management infrastructure

# OPC UA Security Objectives – Trusted Information (CIA Triad)

## Confidentiality

- Keep private data private.

## Integrity

- Data remains unchanged from sender to receiver.

## Availability

- Availability, by restricting the message size and returning no security related codes

# OPC UA Security Objectives – Access Control (AAA Framework)

## Authentication

- Proof of Identity based on something the entity is, has or knows.

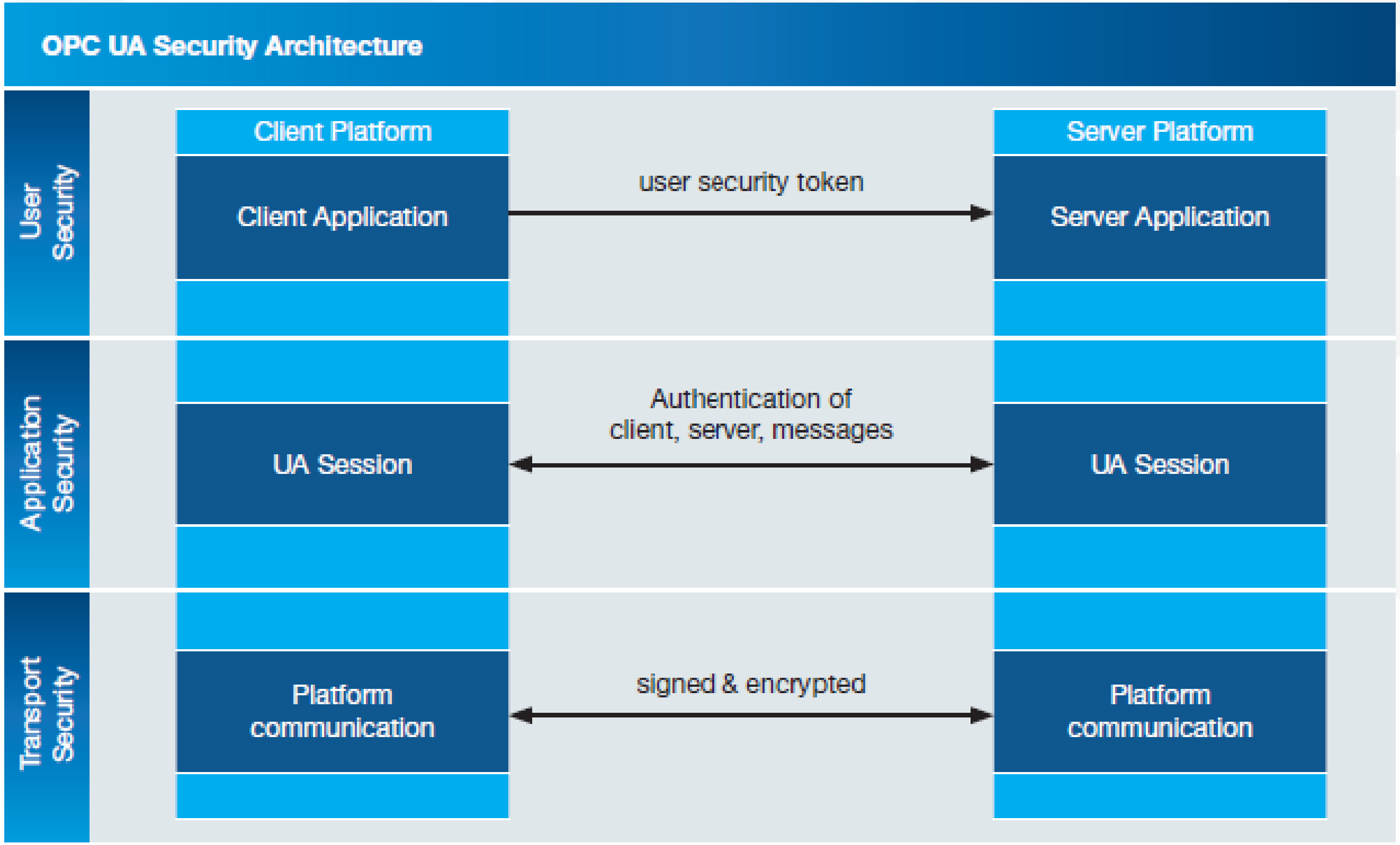
## Authorization

- Not specifically in the OPC UA specification but is vendors choice to implement.

## Auditability

- Provide evidence that the system functions as specified and identify initiators of certain actions.

# Security Architecture



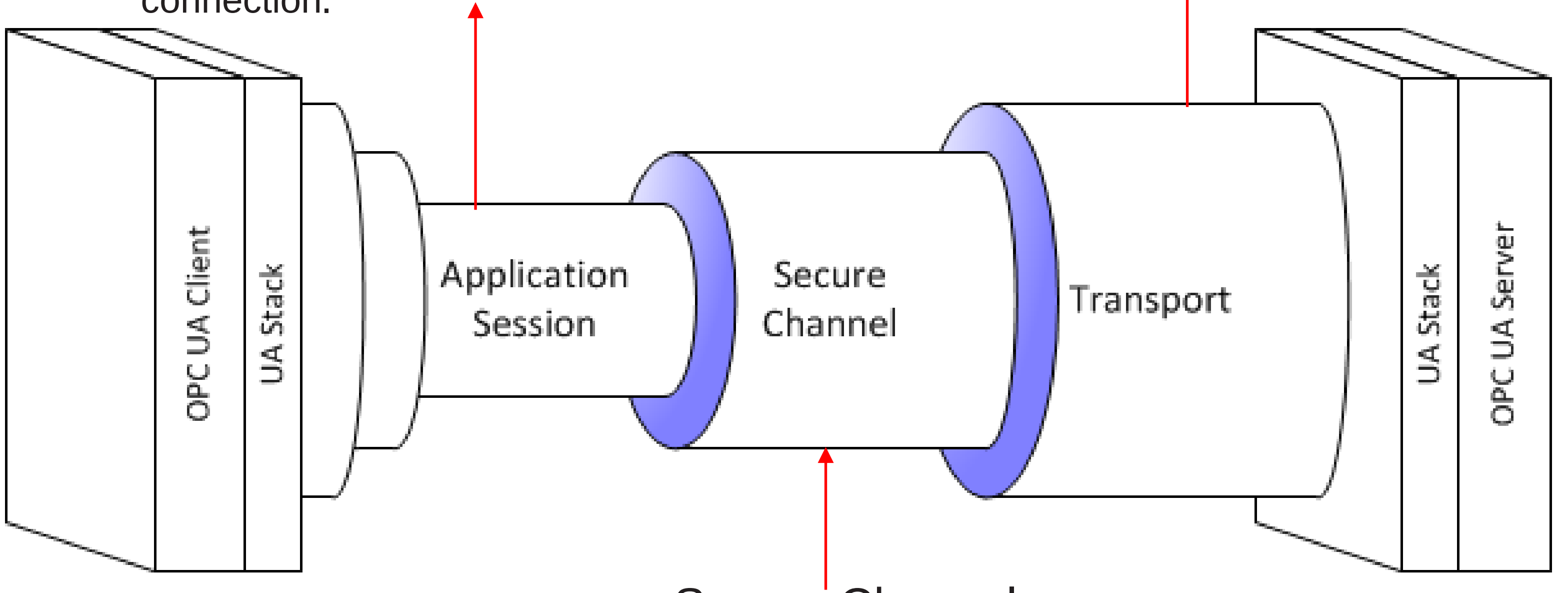
# The Connection Process

## Secure Session

- Application layer connection.
- The session uses the algorithms defined for the channel and assigns an identity to the connection.

## Transport Layer

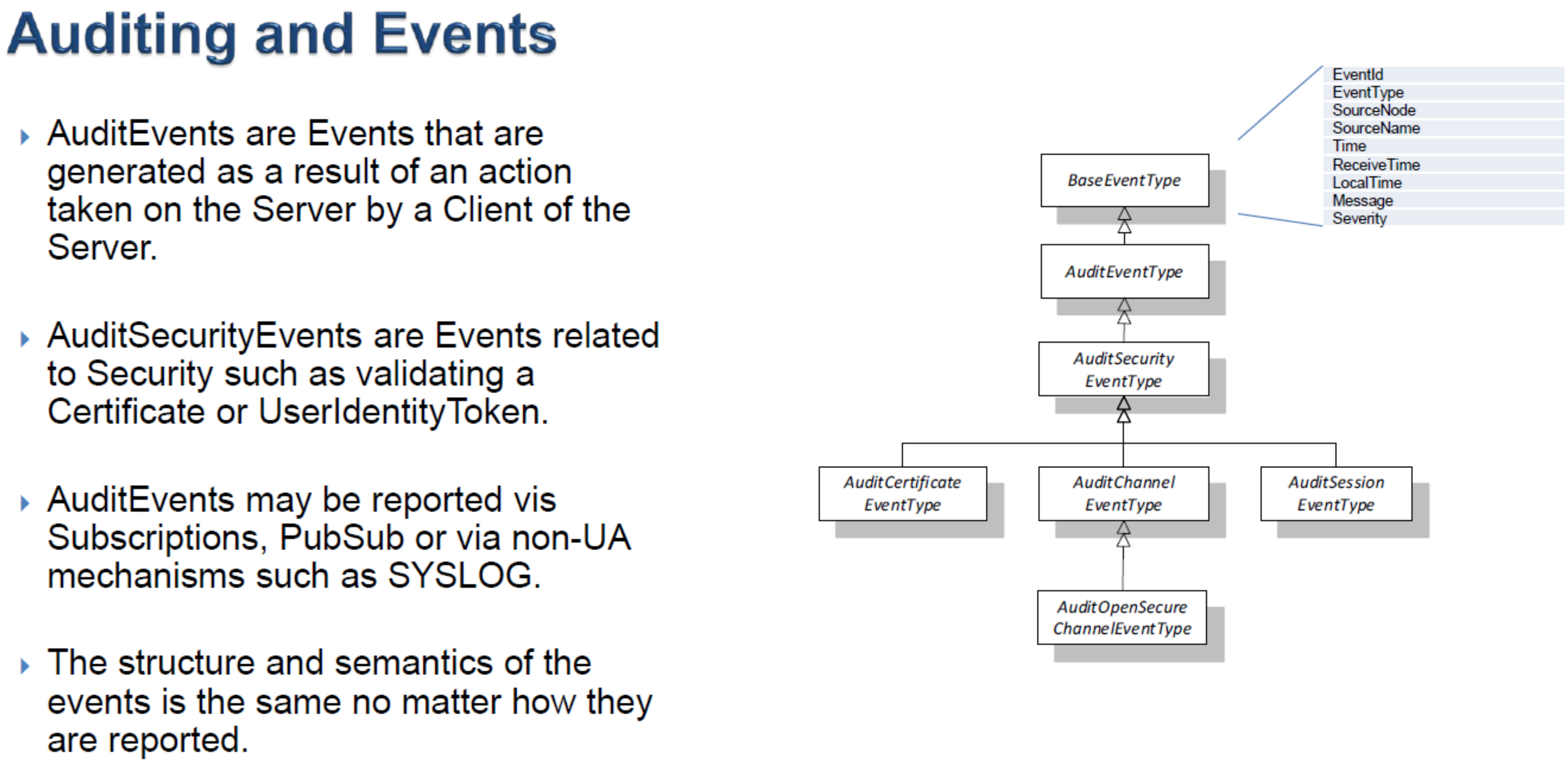
- Protocol determined by the UA Stack.
- Security not required, but may be implemented by the protocol, i.e. HTTPS.



## Secure Channel

- Long-running logical connection between single client and server.
- Ensures *Confidentiality* and *Integrity* of the messages exchanged.
- Supports three *SecurityModes*: None, Sign, and SignAndEncrypt.
- Latest security algorithms.

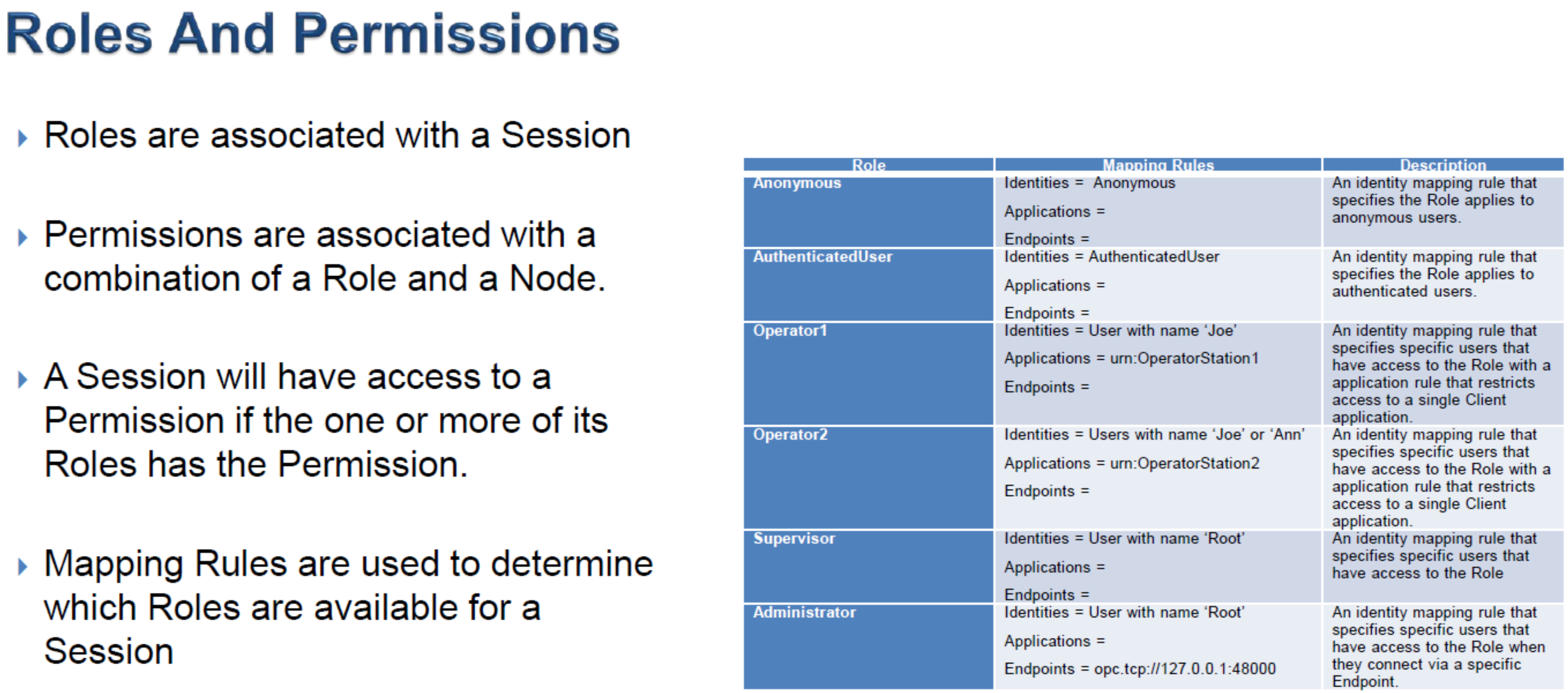
# Auditing and Events



## Auditing and Events

- AuditEvents are Events that are generated as a result of an action taken on the Server by a Client of the Server.
- AuditSecurityEvents are Events related to Security such as validating a Certificate or UserIdentityToken.
- AuditEvents may be reported vis Subscriptions, PubSub or via non-UA mechanisms such as SYSLOG.
- The structure and semantics of the events is the same no matter how they are reported.

# Roles and Permissions



## Roles And Permissions

- Roles are associated with a Session
- Permissions are associated with a combination of a Role and a Node.
- A Session will have access to a Permission if the one or more of its Roles has the Permission.
- Mapping Rules are used to determine which Roles are available for a Session

| Role | Mapping Rules | Description |
|---|---|---|
| Anonymous | Identities = Anonymous<br>Applications =<br>Endpoints = | An identity mapping rule that specifies the Role applies to anonymous users. |
| AuthenticatedUser | Identities = AuthenticatedUser<br>Applications =<br>Endpoints = | An identity mapping rule that specifies the Role applies to authenticated users. |
| Operator1 | Identities = User with name 'Joe'<br>Applications = urn:OperatorStation1<br>Endpoints = | An identity mapping rule that specifies specific users that have access to the Role with a application rule that restricts access to a single Client application. |
| Operator2 | Identities = Users with name 'Joe' or 'Ann'<br>Applications = urn:OperatorStation2<br>Endpoints = | An identity mapping rule that specifies specific users that have access to the Role with a application rule that restricts access to a single Client application. |
| Supervisor | Identities = User with name 'Root'<br>Applications =<br>Endpoints = | An identity mapping rule that specifies specific users that have access to the Role |
| Administrator | Identities = User with name 'Root'<br>Applications =<br>Endpoints = opc.tcp://127.0.0.1:48000 | An identity mapping rule that specifies specific users that have access to the Role when they connect via a specific Endpoint. |

# OPC UA Global Discovery Server (GDS)

## Problem: How do we ensure certificate management in a factory?

- OPC UA specifies a Global Discovery Server

## OPC UA Global Discovery Server (GDS)

- provides the necessary infrastructure to provide enterprise-wide administration of OPC UA Servers.
- handles discovery of registered OPC UA servers.
- **…but also handles certificate management.**

## Major organizations have implementations of OPC UA GDS.

- Example: OPC Vault from Microsoft

# OPC UA Companion Specifications

# Purpose of Companion Specification

- OPC UA Companion Specifications are developed for various reasons:
  - To publish specific information models (e.g., for specific industries, specific devices, specific use cases).
  - To specify how to use OPC UA in specific environments.
  - **Examples:**
    - OPC UA standard for Robotics (OPC 40010), Vision Systems (OPC 40100), Joining Systems (OPC 40450 and 40451), etc.

- **New Information Models** can be created based on the OPC UA Data Model and eventually derived from OPC UA **Base** Information Models.
  - Companion specifications of such Information Models are often called "**Industry standard models**" because they typically address a dedicated industry problem.
  - The synergy of the OPC UA infrastructure to exchange such industry information models enables interoperability at the semantic level.

# Harmonization and Reuse

- The purpose of harmonization is to ensure that several companion standards reuse common building blocks. Several use cases could be common for various domains.
    - **Examples:** Asset Identification, Asset States, etc.

- OPC Foundation and partner organizations such as VDMA enable the reuse of common elements across domains.

- There are several OPC Harmonization groups where representatives of different companion specifications participate and harmonize the information models.
    - **Examples:** Device Integration, Machinery Building Blocks, Machinery Result Transfer, Asset Management Basics, Relative Spatial Location, and many more...

- Any new companion specification planned to be developed is recommended to be reviewed in the OPC Harmonization Group.

- This approach helps in avoiding reinvention of the wheel.

# Ways to produce companion specifications

OPC Foundation differentiates **three** ways of producing companion specifications:

- INTERNAL: These are specifications created by OPC-internal working groups.

- JOINT: These are specifications that are created in a joint working group between the OPC Foundation and another organization.
    - These joint specifications represent the majority. The JOINT working group program is defined here: https://opcfoundation.org/joint-working-groups/
    - **Example:** IJT Working Groping is a Joint Working Group of OPC Foundation, VDMA and various vendors. VDMA hosts the IJT Working Group.

- EXTERNAL: Companion specifications can also be created independent of the OPC Foundation.

# The Key to Reducing Costs: Data Interoperability

We need…

1. A common **Interface** (Analogy: A Book)

2. A common **Data Format** (Analogy: A Latin Alphabet)

3. A common **Data Model** (Analogy: Then English Language)

4. Common **Semantics** (Analogy: The Plot and Characters of the Novel)

Only when all 4 things are present, we can truly understand each other!

# Interoperability Example for OPC UA-enabled Assets

1. **Interface:** OPC UA Client/Server

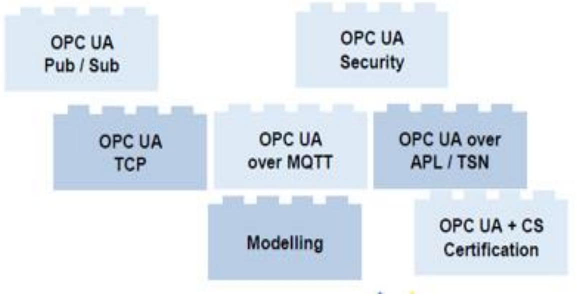
2. **Data Format:** OPC UA Binary or JSON

3. **Data Model:** OPC UA Information Model

4. **Semantics:** OPC UA IJT Companion Specification

# Interoperability Solution Summary

## ( OPC UA + Companion Specs ) = Promise for Interoperability



OPC UA Pub / Sub

OPC UA Security

OPC UA TCP

OPC UA over MQTT

OPC UA over APL / TSN

Modelling

OPC UA + CS Certification

- **OPC UA: Collection of technology bricks**
  - Connectivity, different protocols
  - Security
  - …

**HOW to communicate**

“speak the same language”

+

- **Companion Specifications: Collection of bricks for different markets**
  - Information modelling to describe the specific market
  - …

=

**WHAT to communicate**

“use the same dictionary”

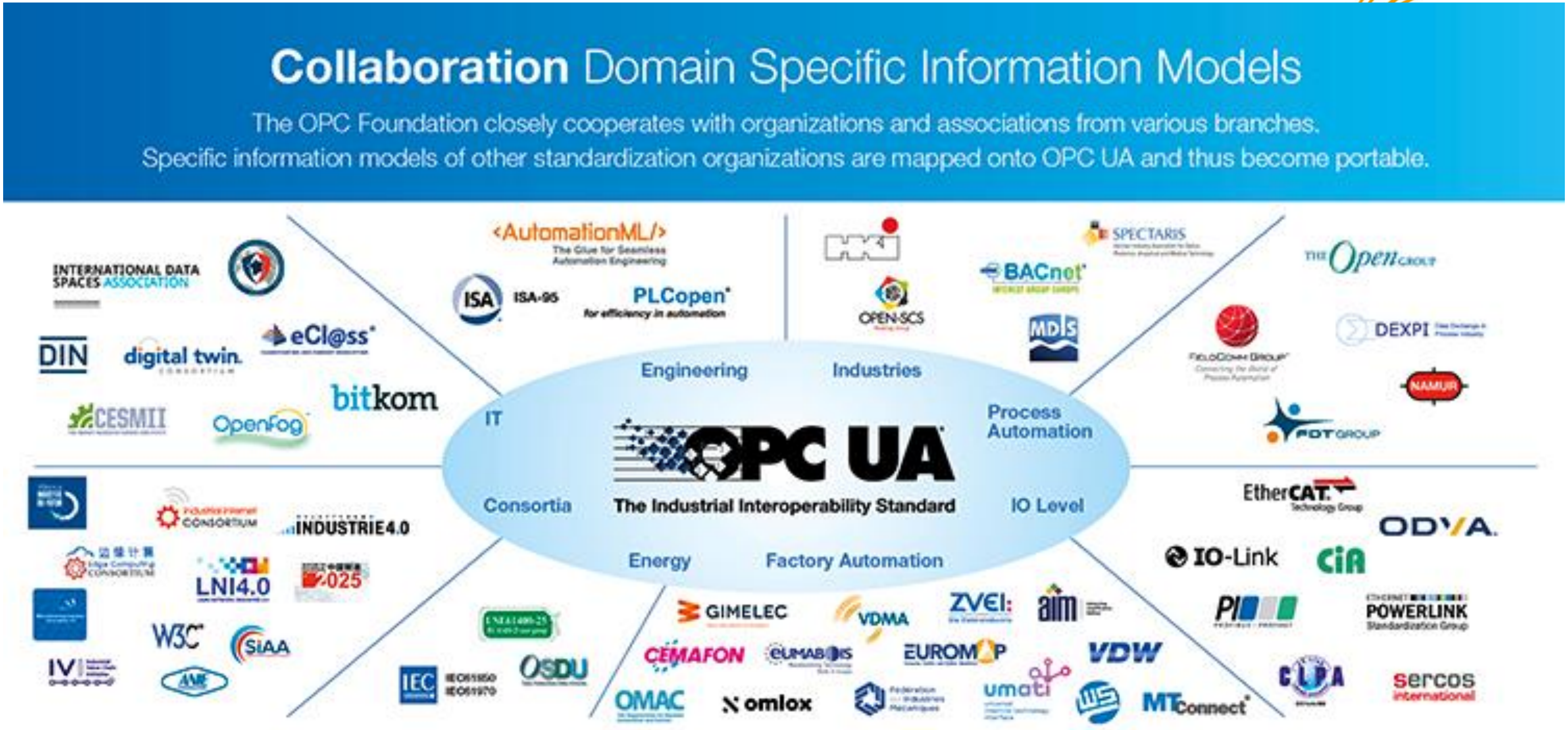
- **OPC UA + Companion Specification drives towards Interoperability**
  - Mandatory bricks guarantee interoperability
  - Optional bricks allow flexibility
  - …

semantic interoperability

**“understand each other”**

Source: OPC Foundation

# Collaboration Domain Specific Information Models



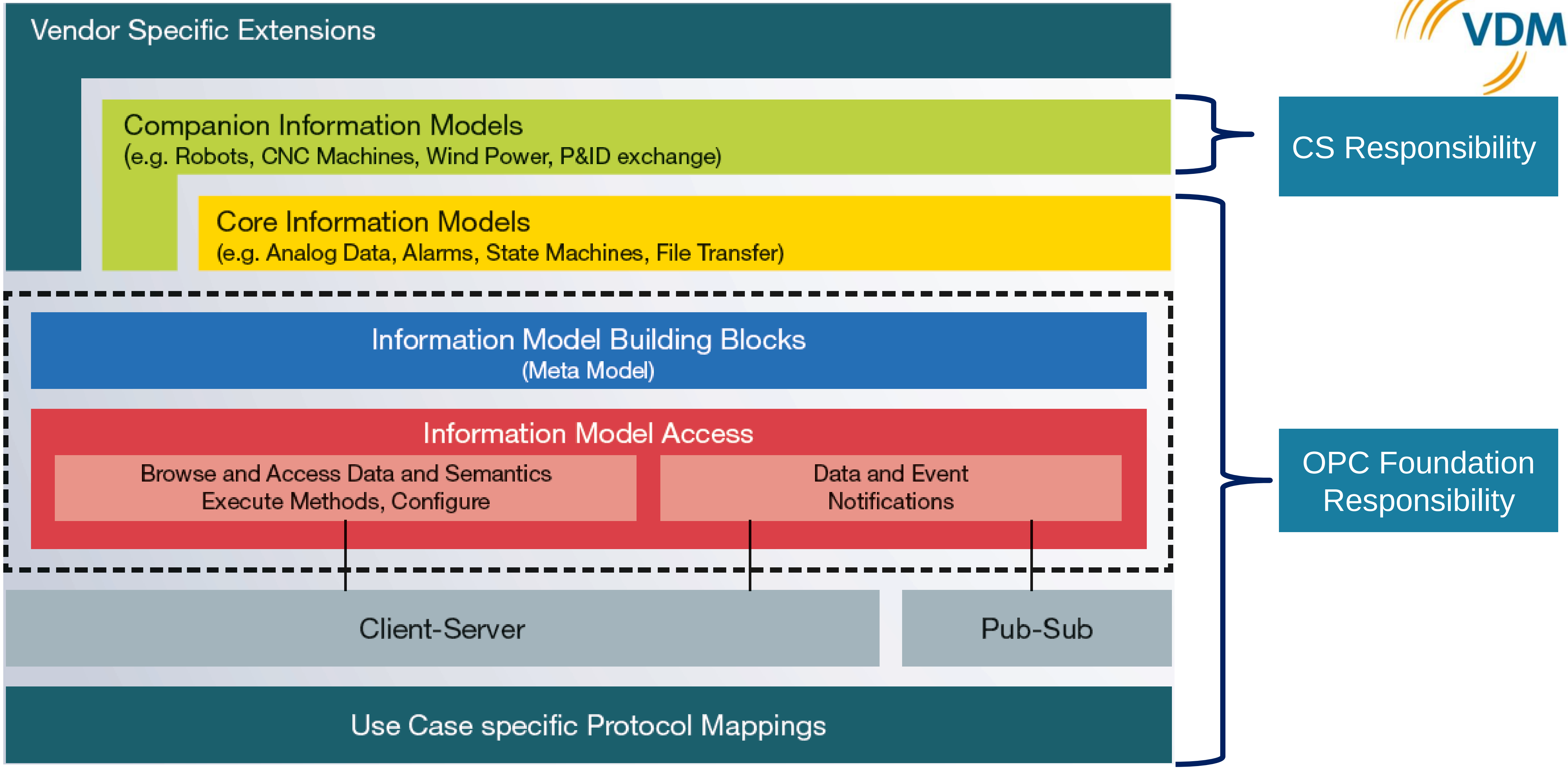
## Collaboration Domain Specific Information Models

The OPC Foundation closely cooperates with organizations and associations from various branches.
Specific information models of other standardization organizations are mapped onto OPC UA and thus become portable.

Source: OPC Foundation

# OPC UA Stack/SDK

# Layered Architecture



Vendor Specific Extensions

Companion Information Models
(e.g. Robots, CNC Machines, Wind Power, P&ID exchange)

Core Information Models
(e.g. Analog Data, Alarms, State Machines, File Transfer)

Information Model Building Blocks
(Meta Model)

Information Model Access

Browse and Access Data and Semantics
Execute Methods, Configure

Data and Event
Notifications

Client-Server

Pub-Sub

Use Case specific Protocol Mappings

CS Responsibility

OPC Foundation Responsibility

Source: OPC Foundation

# OPC UA Stack and SDKs

- OPC UA SDKs are available in various programming languages.
  - Ansi C, C++, C#, Java, Python, JavaScript, etc.

- The core functionality of OPC UA is handled as part of OPC UA SDK.

- OPC Foundation and other organizations provide various frameworks.
  - Open Source
  - Commercial

# OPC UA Profiles and Certification

# OPC UA Profiles and Certification

- **What is OPC Certification?**
  - https://opcfoundation.org/certification/overview-benefits/
  - https://opcfoundation.org/products/?certified=yes
- **Compliance Test Tools (CTT)**
  - OPC Foundation provides tools which can validate OPC UA Core functionality.
  - https://opcfoundation.org/developer-tools/certification-test-tools
- **Companion Specification Certification**
  - OPC Foundation provides support in testing CS manually or also helps in defining the automated test cases based on collaboration.
  - **Working Group Responsibility:** Definition of Profiles, CUs, Facets, and Test Cases.
- **OPC UA Profiles**
  - Terms and Definitions: https://reference.opcfoundation.org/Core/Part7/v105/docs/3.1
  - Profile reporting tool: https://profiles.opcfoundation.org/

# Industrial Joining Technologies (IJT) Overview

# Industrial Joining Technologies (IJT) Overview

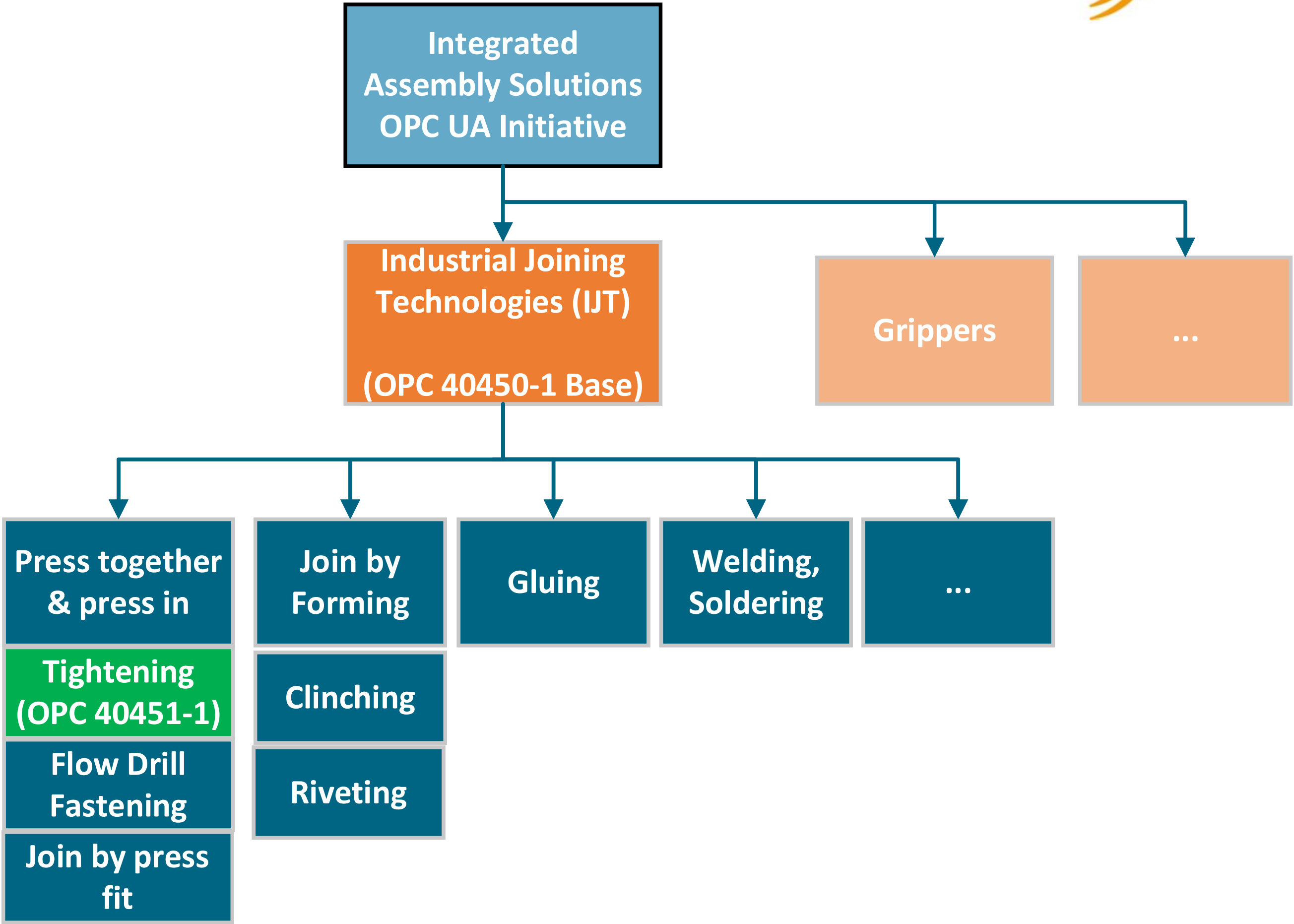
- History

- Taxonomy
  - https://opcfoundation.org/markets-collaboration/IJT/

- Long term vision
  - Initially started with Tightening but was interesting to see the base elements useful for other joining systems.

- Status

- **OPC 40450-1 UA** for Joining Systems
  - **OPC 40451-1 UA** for Tightening Systems
  - …



# IJT Working Group Members

VDMA

Atlas Copco

rexroth
A Bosch Company

WEBER

OPC
FOUNDATION

Cleco
Production Tools

Desoutter

DEPRAG
machines unlimited

STÖGER
AUTOMATION

CSP
EFFICIENCY COMES TRUE

SCS
Concept Group

PLARAD
Torque & Tension Systems

VISION DEVICES

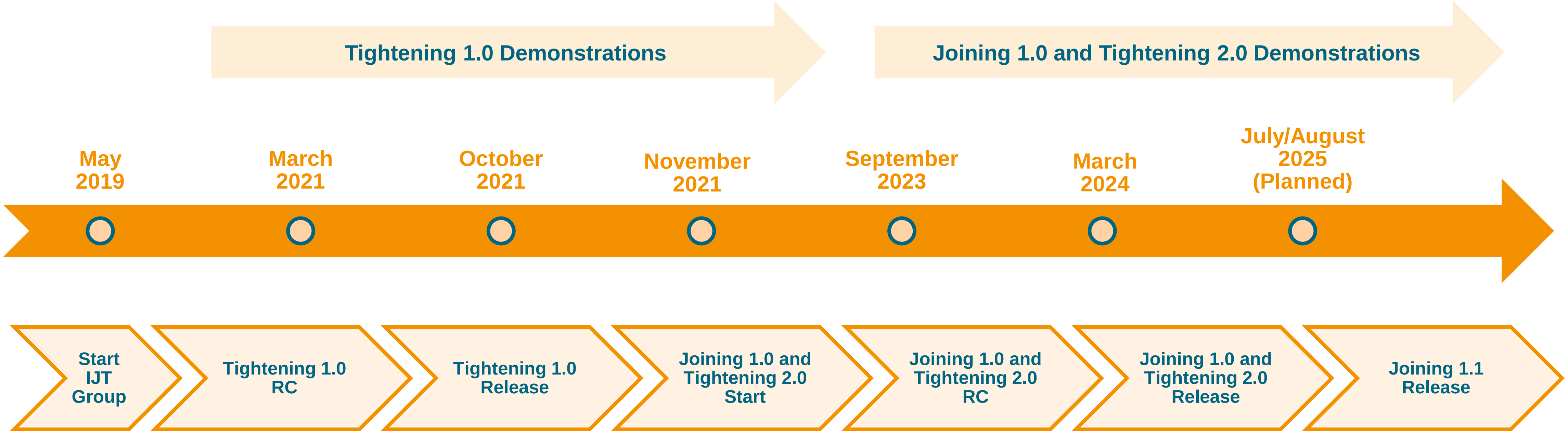
MAFU
AUTOMATION

EST
TECHNOLOGIE

AMT
Product of EST

XITASO

# History and Milestones



Tightening 1.0 Demonstrations

Joining 1.0 and Tightening 2.0 Demonstrations

May 2019 – Start IJT Group

March 2021 – Tightening 1.0 RC

October 2021 – Tightening 1.0 Release

November 2021 – Joining 1.0 and Tightening 2.0 Start

September 2023 – Joining 1.0 and Tightening 2.0 RC

March 2024 – Joining 1.0 and Tightening 2.0 Release

July/August 2025 (Planned) – Joining 1.1 Release

# Reuse of Harmonized Specifications

| NamespaceUri | Description | Use | Namespace Index | Example |
|---|---|---|---|---|
| http://opcfoundation.org/UA/ | OPC UA Base | Mandatory | 0 | 0:EngineeringUnits |
| http://opcfoundation.org/UA/DI/ | OPC UA for Devices (OPC 10000-100). | Mandatory | 2 | 2:DeviceRevision |
| http://opcfoundation.org/UA/AMB/ | OPC UA for Asset Management Basics (OPC 10000-110). | Mandatory | 3 | 3:IRootCauseIndicationType |
| http://opcfoundation.org/UA/Machinery/ | OPC UA for Machinery Basic Building Blocks (OPC 40001-1). | Mandatory | 4 | 4:MachineIdentificationType |
| http://opcfoundation.org/UA/Machinery/Result/ | OPC UA for Machinery Result Transfer (OPC 40001-101). | Mandatory | 5 | 5:ResultManagementType |

# IJT Use Cases

# Use Cases Overview

## Asset Management

Overview and Identification of physical assets in the given system.

**Example:**

Manufacturer, Serial number, Software Revision…

## Condition Monitoring

Acquisition and processing of information that indicate the state of an asset over time.

**Example:**

Health status, temperature…

## Result Management

Primary process output of the tightening operation.

**Example:**

Single Result, Batch Result, Job Result, Multi-spindle Result, etc.

## Event Management

Various types of events with standard payload and filter criteria.

**Example:**

Tool Connected, Maintenance Events, etc.

## Commands

**Asset** Management control mechanisms.

**Joining Process** Management.

**Example:**

Select Program, Send Program, Enable Tool, etc.

## Joint Management

Provides joint data.

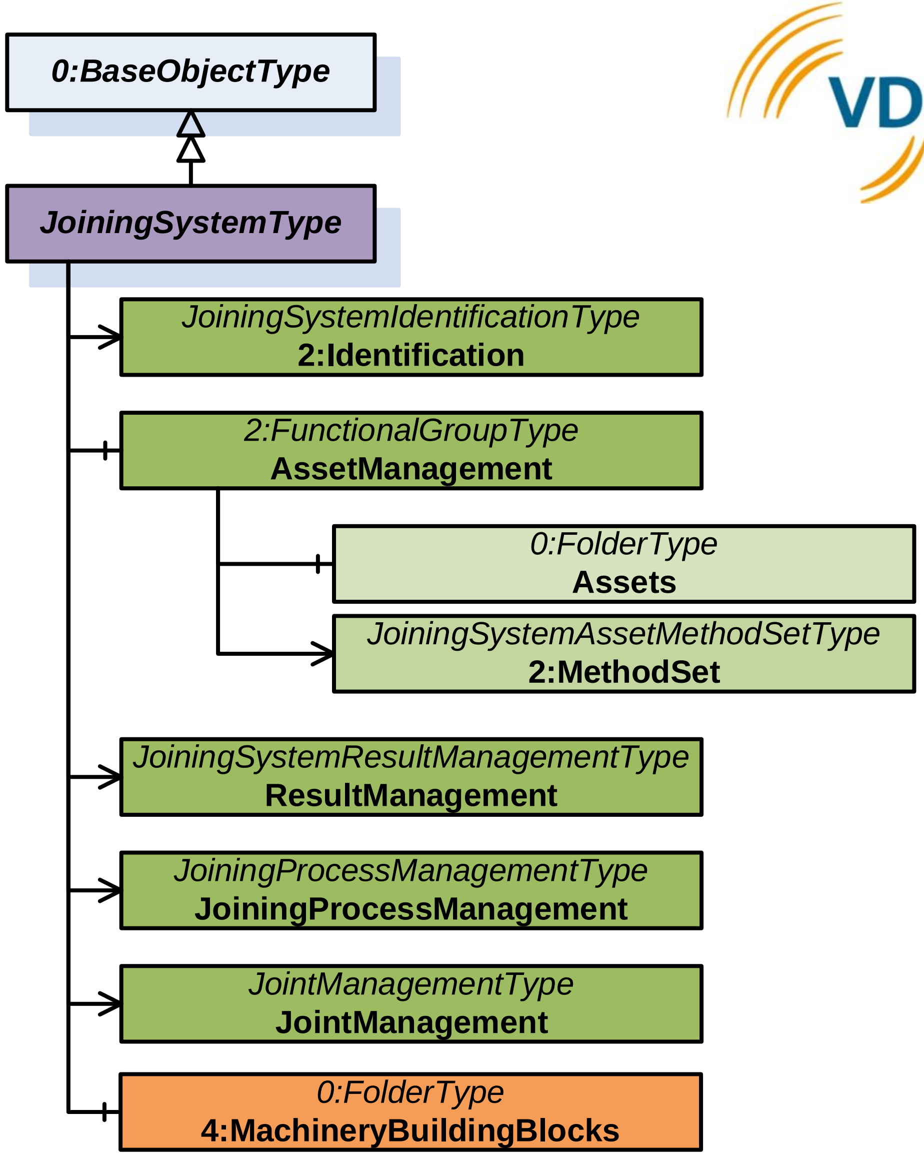
**Example:**

Joint with associated Programs, etc.

# Technical Overview of Models

# Joining System Overview

- Standard entry point for a joining system.
- Common interface for any joining system.
- Top-level structure with building blocks for the **use cases** discussed such as:
  - Asset Management
  - Result Management
  - Joining Process Management
  - Joint Management
  - …



# Asset Management

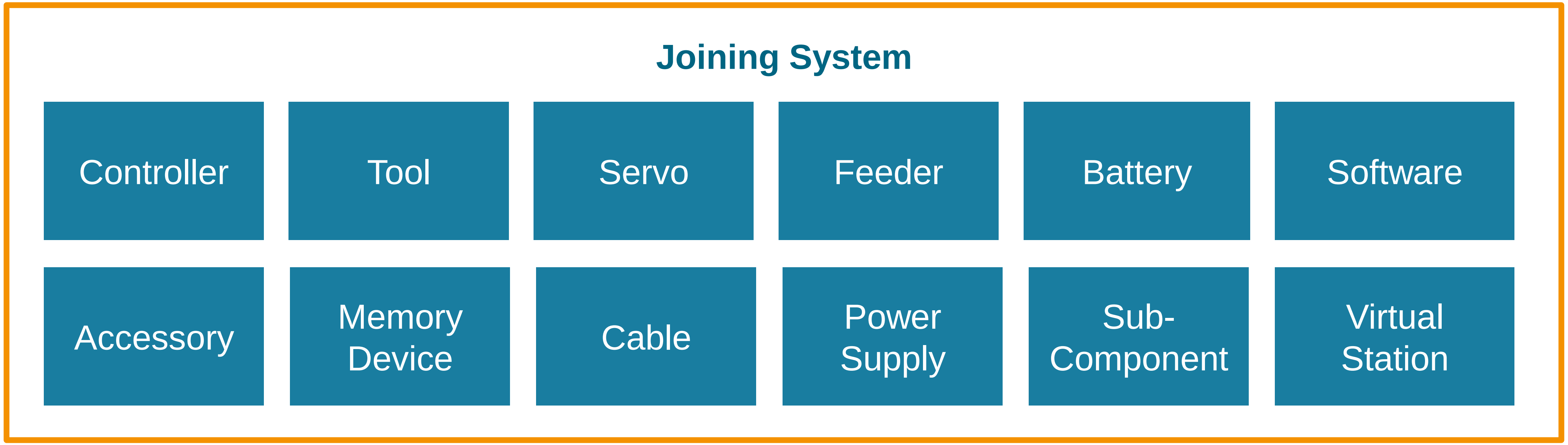
# Asset Management Overview

- Definition of assets building blocks.

- Diverse systems – Flexible asset management model to build different systems.
  - Fixtured, Handheld, Pneumatic, Multi-Channel, Single Channel

- Reuse of Machinery Building Blocks and Asset Management Basics.

- Future Extensibility with the usage of  Interfaces and Add-Ins instead of concrete types.



**Joining System**

Controller

Tool

Servo

Feeder

Battery

Software

Accessory

Memory Device

Cable

Power Supply

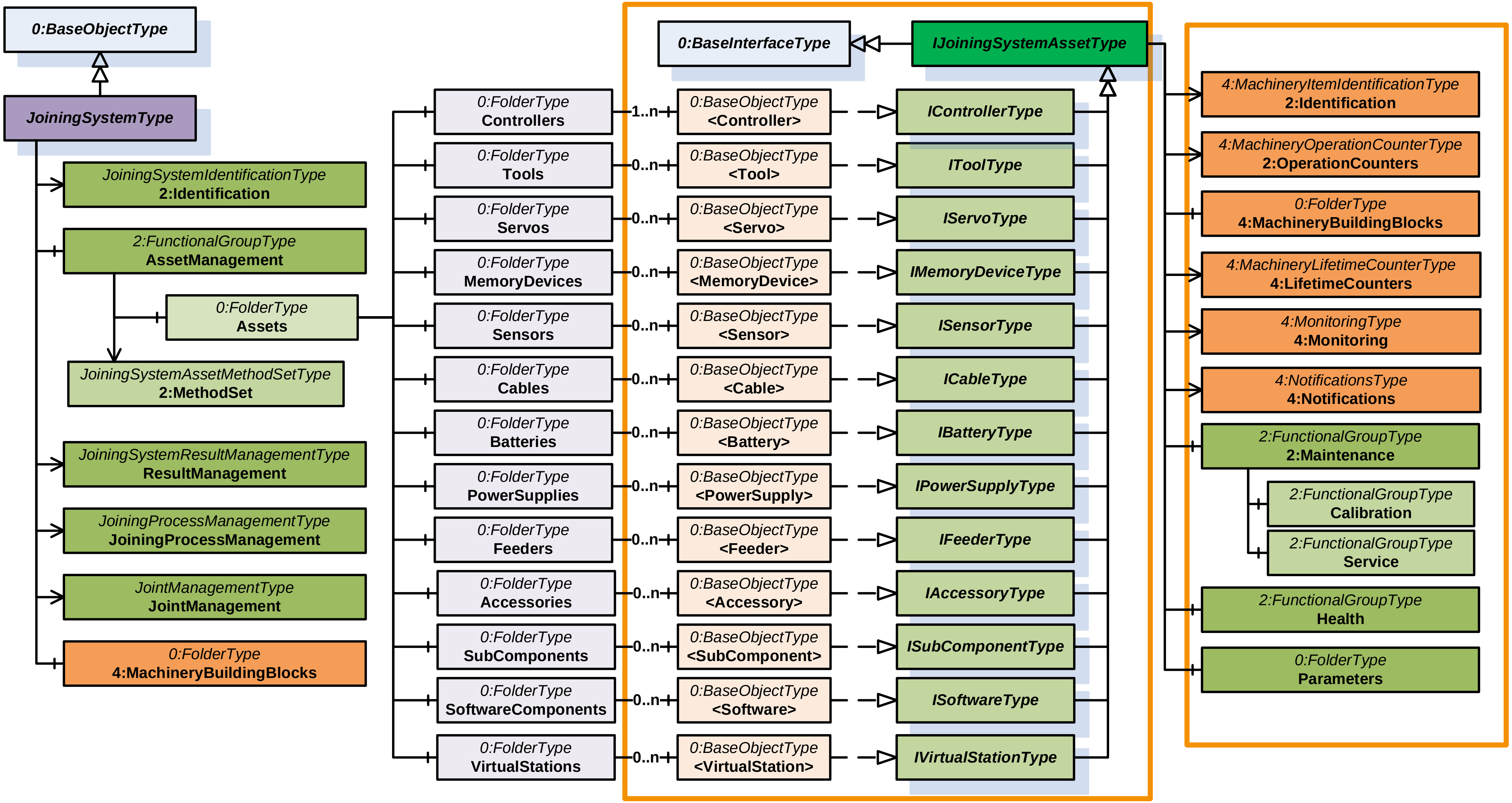
Sub-Component

Virtual Station

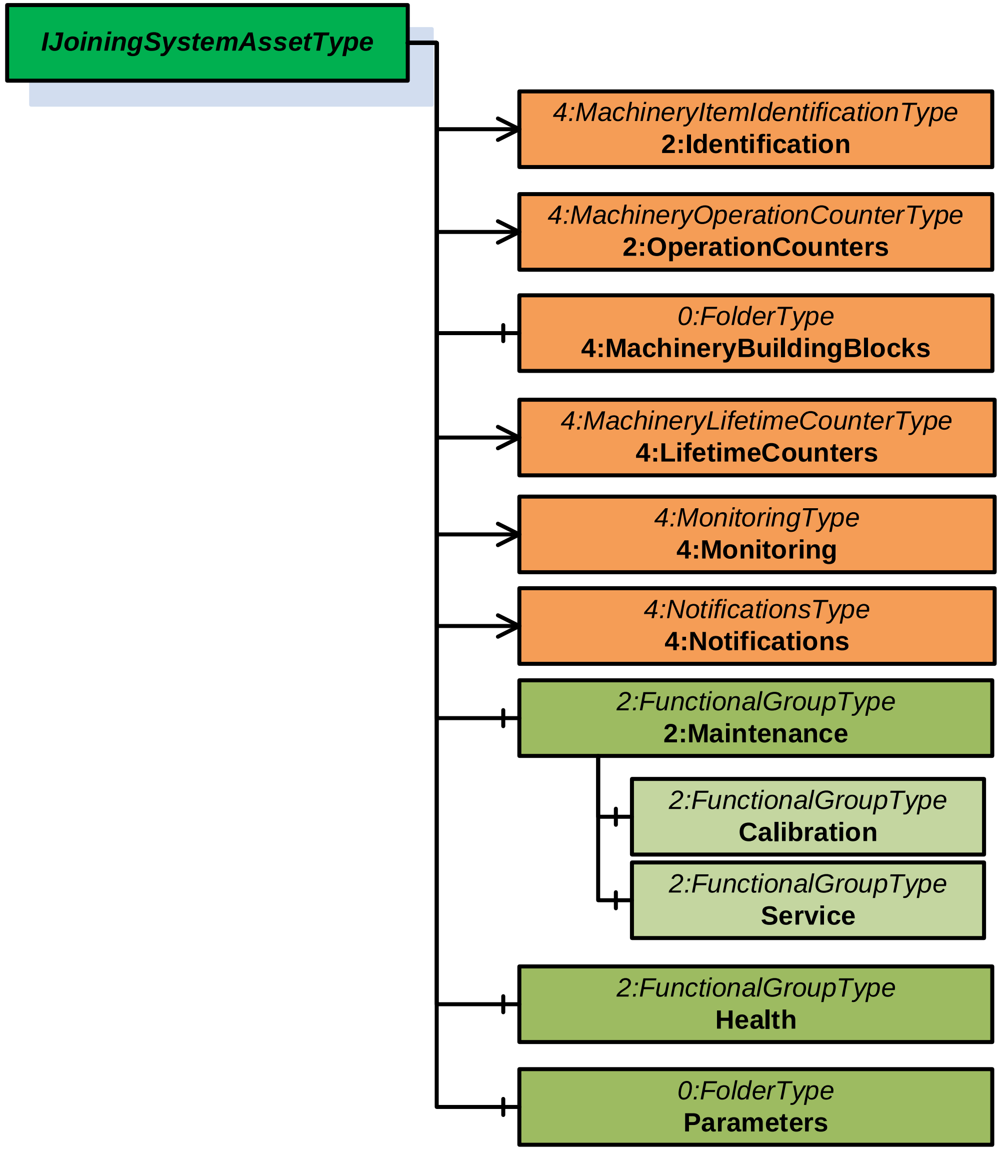
# Asset Management

Flexible Asset Model…

Common Asset Data…



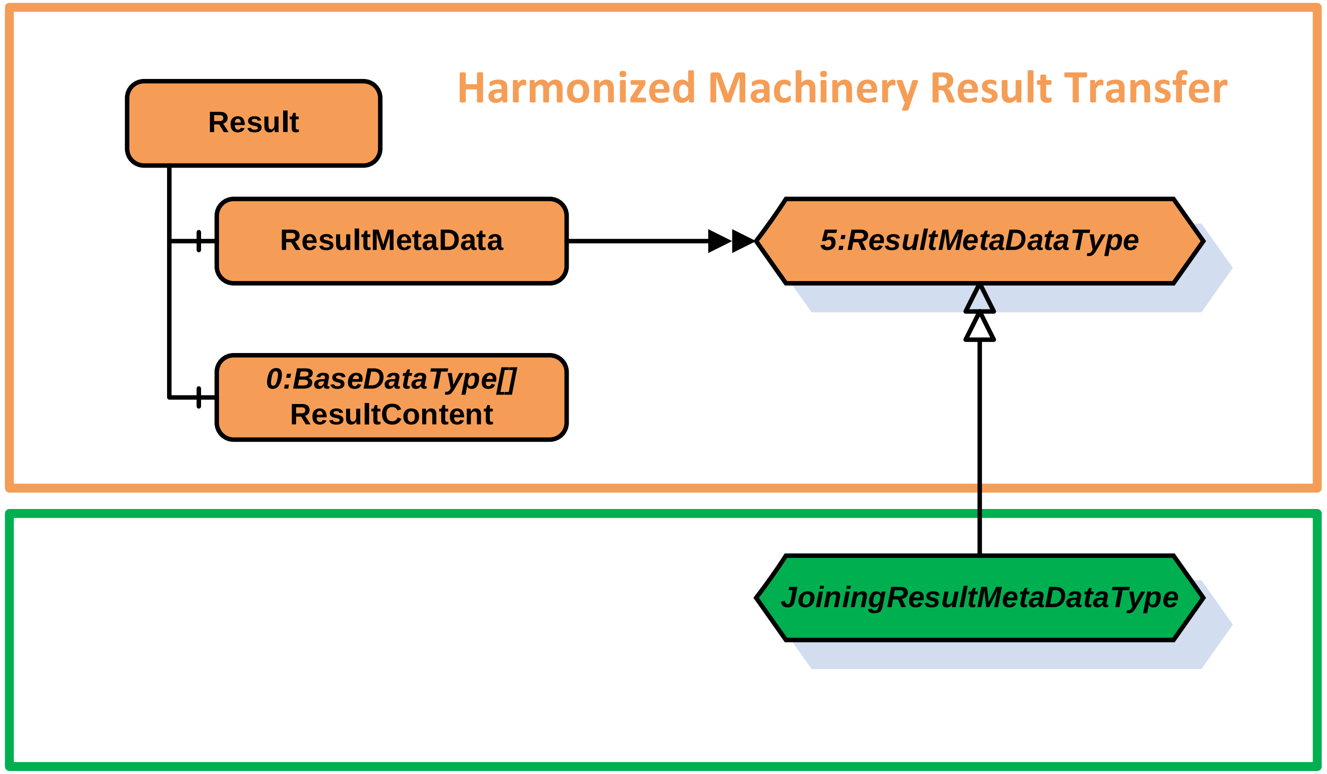
# Common Asset Data



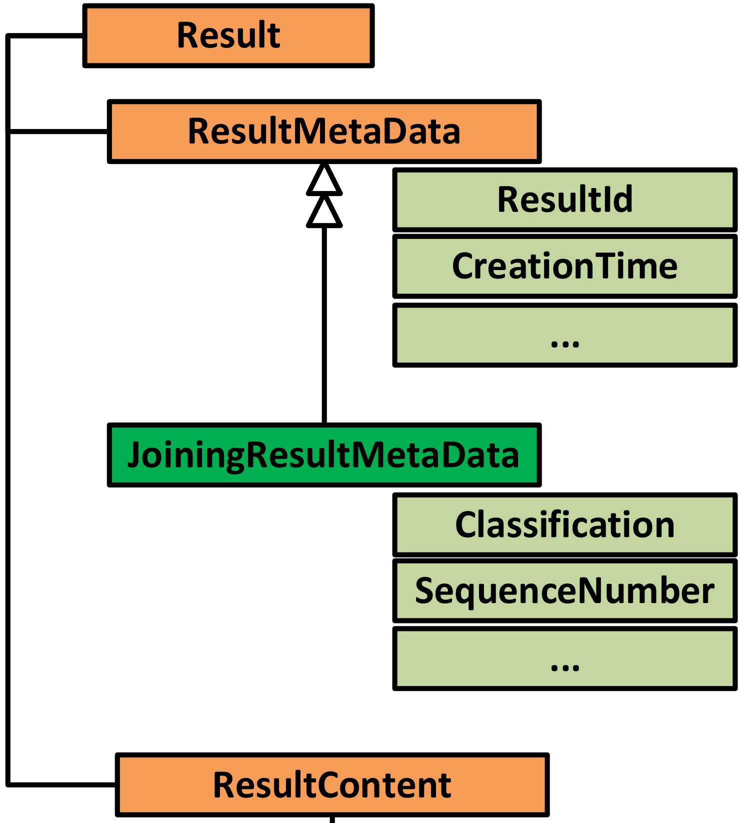
# Result Management

# Result Overview

## Structure



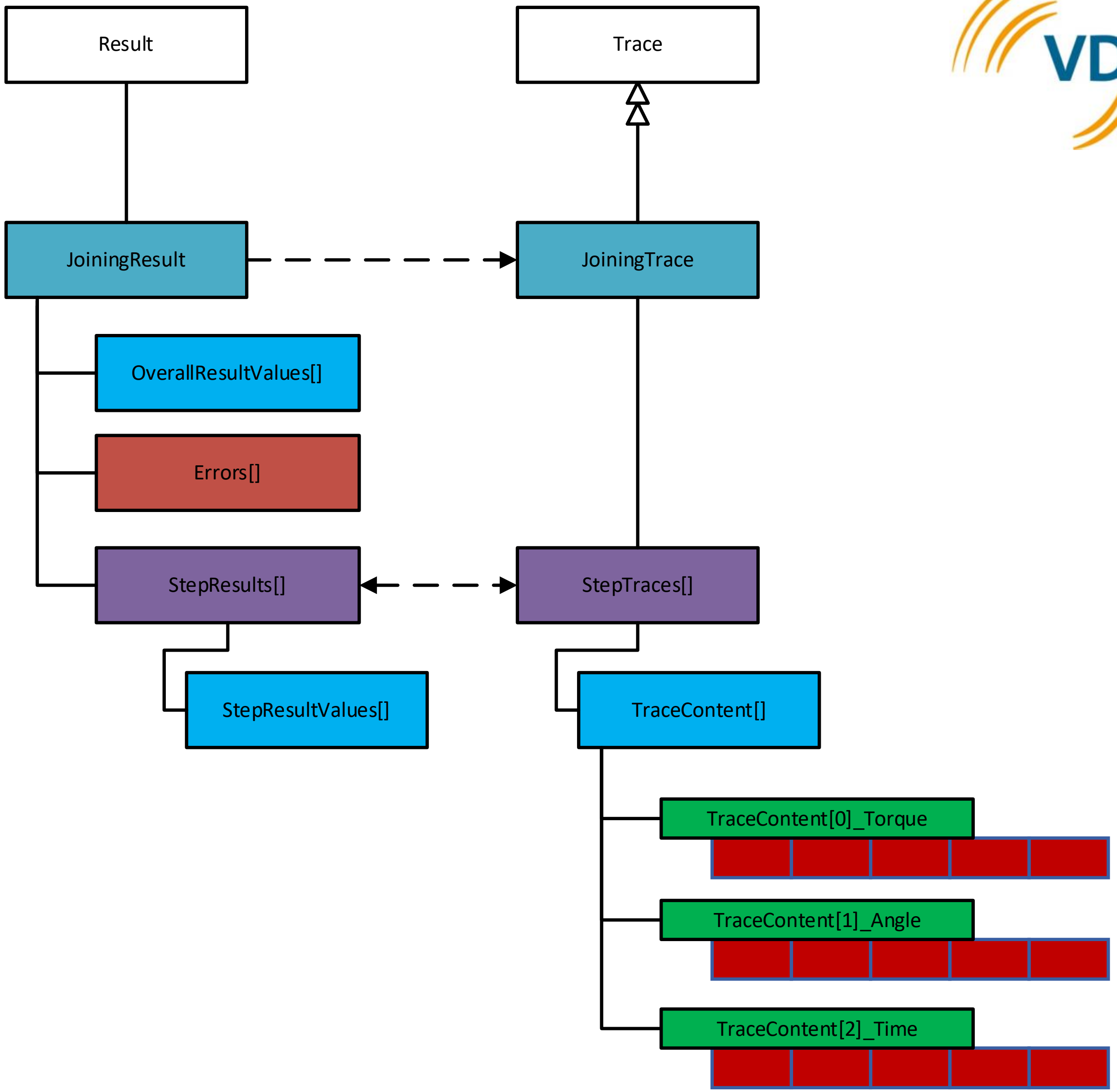
## Example



It can be Joining Result, Set of Sub-Results, Vision Result, Quality Result, etc.

# Single Joining Result Content

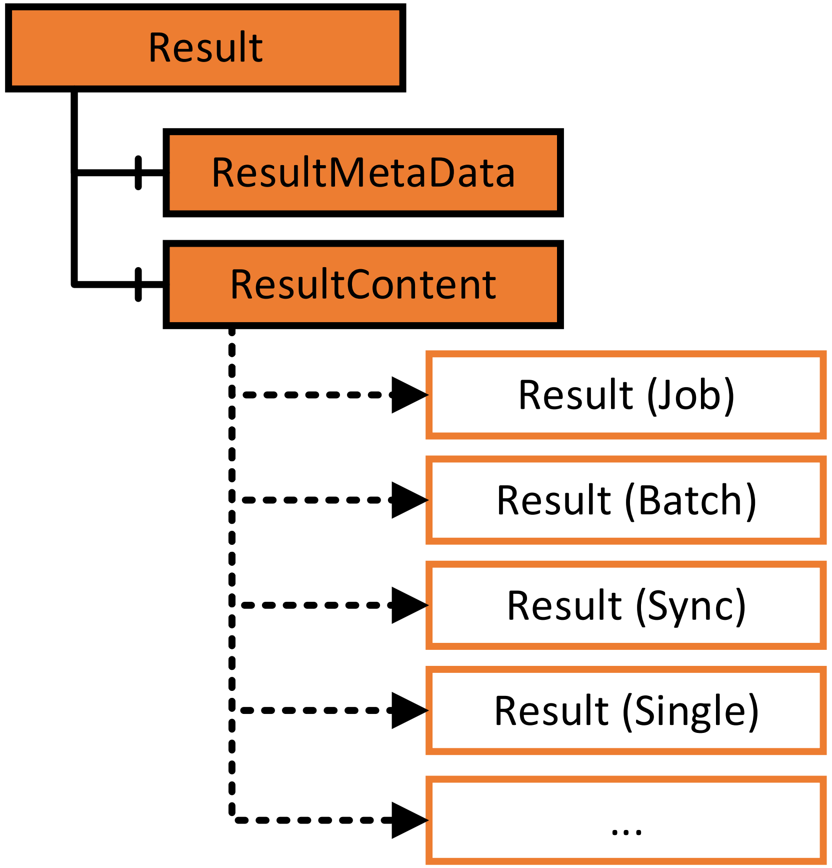
- Joining Result
  - Global values
  - Step Results
  - Errors
  - Traces



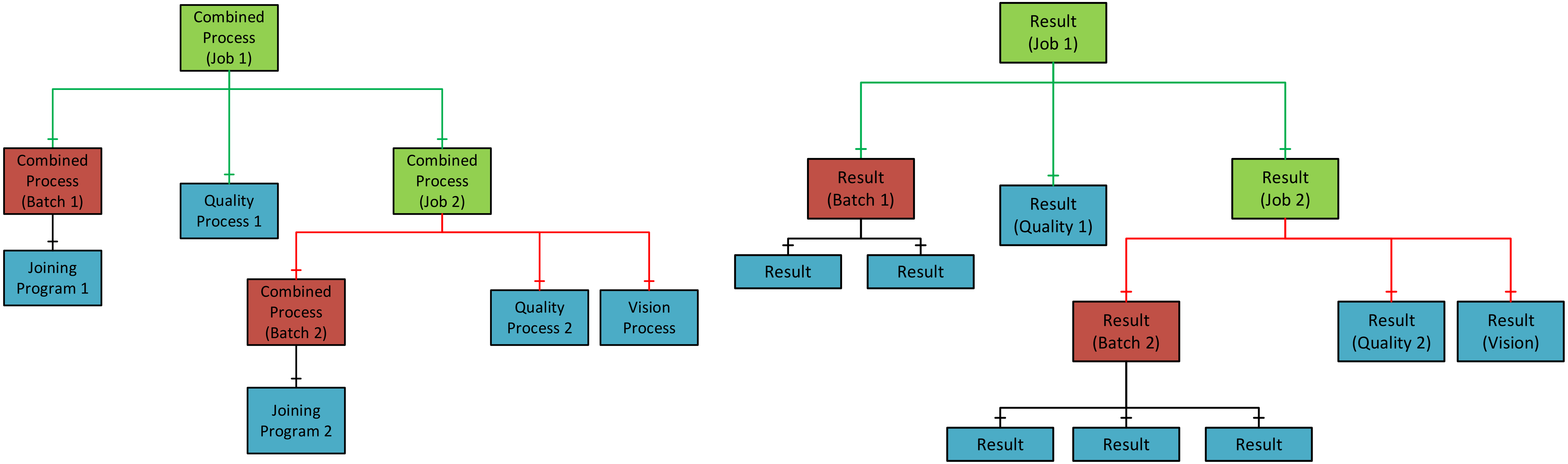
# Result Classification – Multiple Use Cases

- Result
  - Common Joining Meta Data
- Result Content
  - Single Result
  - Batch Result
  - Job Result
  - Sync (Multi-spindle) Result
  - Stitching Result

RESULT MODEL



# Combined Results – Job Result Example



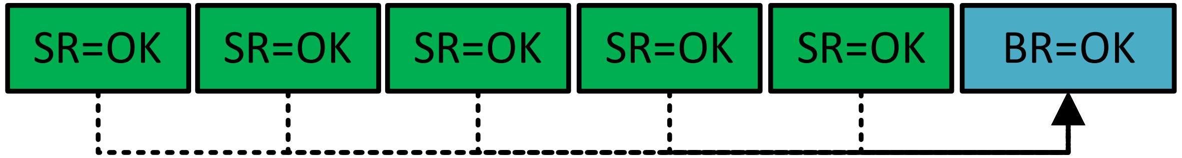
# Parent and Child Results - Options

- Results can be sent in different ways:
- **Case 1:** Parent and Child results separately
- **Case 2:** Parent result with payload of child results

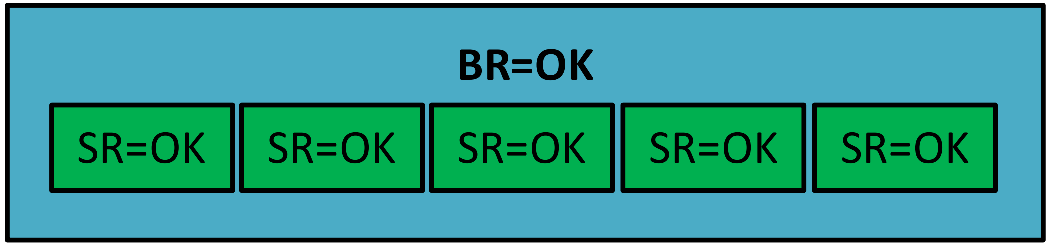
## Batch Result Example

**SR** - SINGLE RESULT
**BR** - BATCH RESULT

**Case 1: Batch Result and Single Results separately.**
**Single Result with reference to parent Batch Result**



**Case 2: Batch Result and Single Results together**

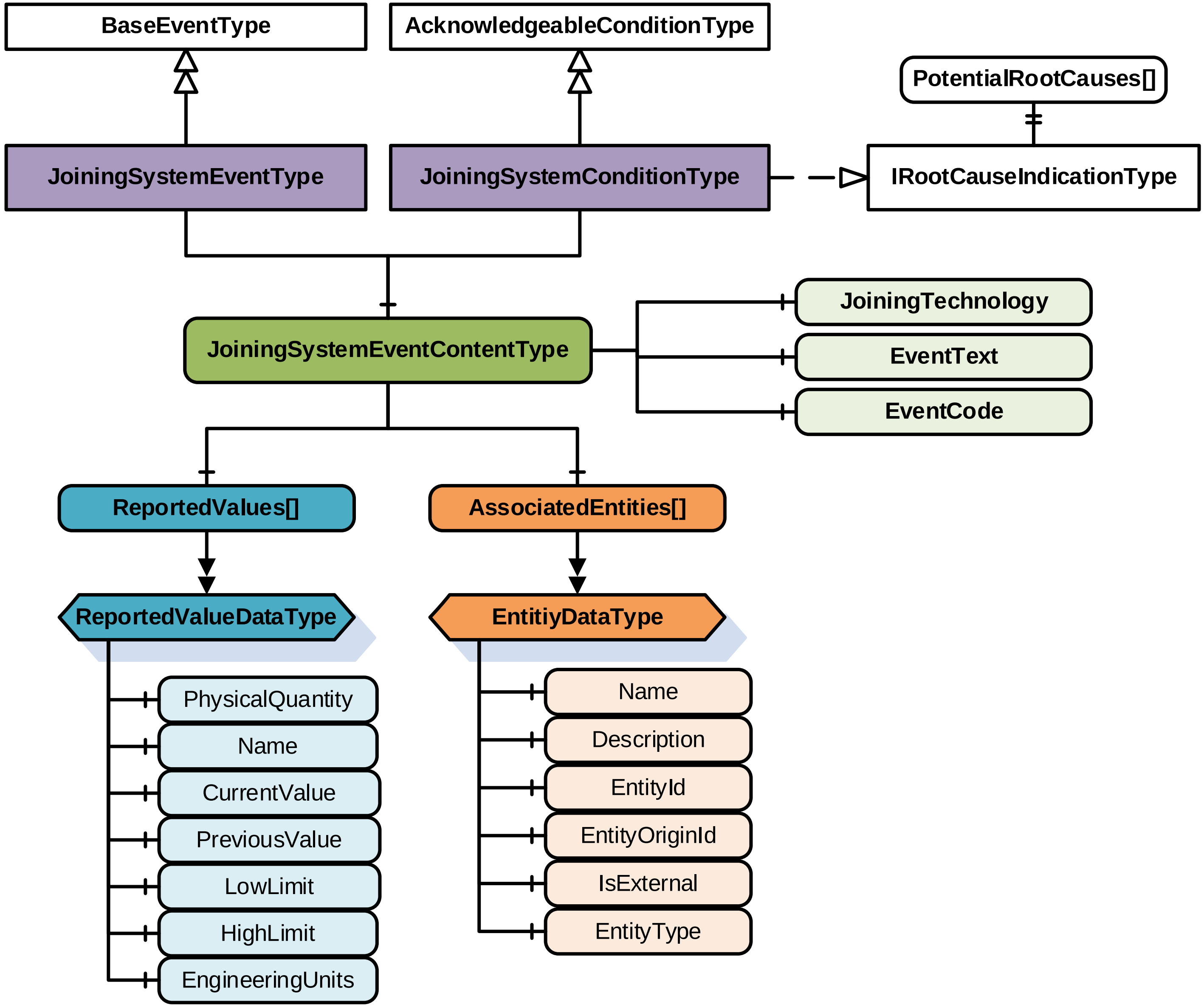


# Event Management

# Event Management

Event and Condition with **standard payload** from a joining system.

- **Event**
  - Simple transient information sent from the underlying system.
  - Fire and forget from the sender's perspective.
- **Condition**
  - Have a State associated.
  - Can be acknowledged by the client.



# Condition Classes

- **Condition Classes**
  - Reuse from base specifications.
- **Condition Sub Classes**
  - Defined in the IJT Working Group.

- **Example 1: Tool Disconnected**
  - **ConditionClass** = SystemConditionClass
  - **ConditionSubClass** = AssetDisconnectedConditionClass

- **Example 2: The tool is out of range from the station.**
  - **ConditionClass** = SystemConditionClass
  - **ConditionSubClass** = LocationOutOfZoneConditionClass

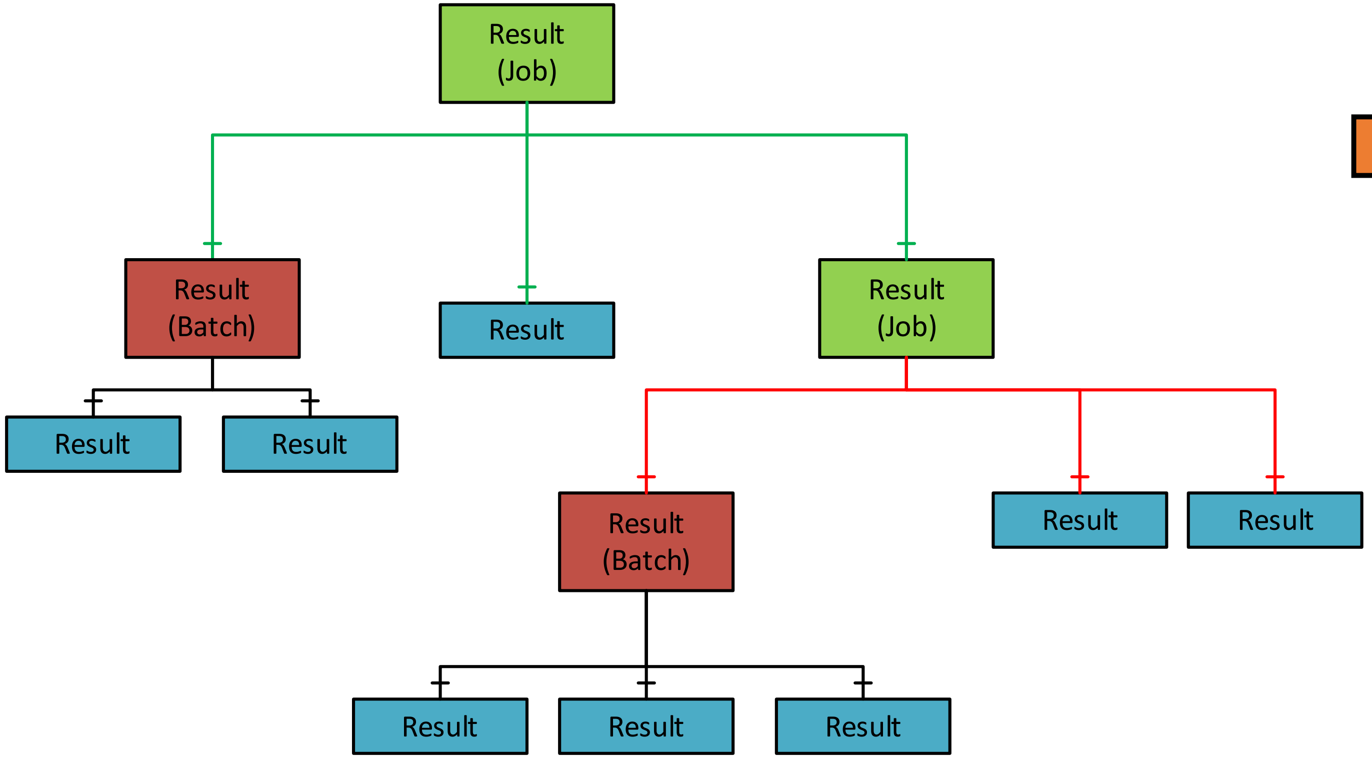
- **Example 3: Software Expired**
  - **ConditionClass** = SystemConditionClass
  - **ConditionSubClass[0]** = SoftwareConditionClass
  - **ConditionSubClass[1]** = ExpiredConditionClass

| Condition Classes in Base Specifications | Condition Classes in IJT Specification(s) |
|---|---|
| OPC UA Base | AssetConnectedConditionClass |
| BaseConditionClass | AssetDisconnectedConditionClass |
| ProcessConditionClass | AssetEnabledConditionClass |
| MaintenanceConditionClass | AssetDisabledConditionClass |
| SystemConditionClass | ThresholdViolationConditionClass |
| SafetyConditionClass | ThresholdViolationResolvedConditionClass |
| HighlyManagedAlarmConditionClass | JoiningSystemUserLoggedInConditionClass |
| TrainingConditionClass | JoiningSystemUserLoggedOutConditionClass |
| StatisticalConditionClass | LocationInRangeConditionClass |
| TestingConditionClass | LocationOutOfRangeConditionClass |
| Asset Management Basics | AssetLocationConditionClass |
| ConnectionFailureConditionClass | DataValidationFailureConditionClass |
| OverTemperatureConditionClass | InputValidationFailureConditionClass |
| CalibrationDueConditionClass | ConfigurationChangeConditionClass |
| SelfTestFailureConditionClass | ErrorConditionClass |
| FlashUpdateInProgressConditionClass | SoftwareConditionClass |
| FlashUpdatedFailedConditionClass | HardwareConditionClass |
| BadConfigurationConditionClass | CertificateConditionClass |
| OutOfResourcesConditionClass | LicenseConditionClass |
| OutOfMemoryConditionClass | MissingEntityConditionClass |
| InspectionConditionClass | ExpiredEntityConditionClass |
| ExternalCheckConditionClass | InvalidEntityConditionClass |
| ServicingConditionClass | IncompatibleEntityConditionClass |
| ImprovementConditionClass | AcceptedEntityConditionClass |
| RepairConditionClass | RejectedEntityConditionClass |
|  | AddedEntityConditionClass |
|  | UpdatedEntityConditionClass |
|  | RemovedEntityConditionClass |
|  | ReceivedEntityConditionClass |
|  | SelectedProcessConditionClassType |
|  | SelectedEntityConditionClass |
|  | UnacknowledgedResultsConditionClass |
|  | EntityExpiryWarningConditionClass |
|  | StartedEntityConditionClass |
|  | StoppedEntityConditionClass |
|  | NotAvailableEntityConditionClass |
|  | NotSupportedEntityConditionClass |

# Joining Process Management

# Joining Process

- Generic container for any type of Joining Process.
  - **Examples:** Joining Program, Joining Batch, Joining Job, etc.
- A concrete definition of the process is vendor-specific.
- The specification defines the interface to access the required processes.



Result (Job)

Result (Batch)

Result

Result (Job)

Result

Result

Result (Batch)

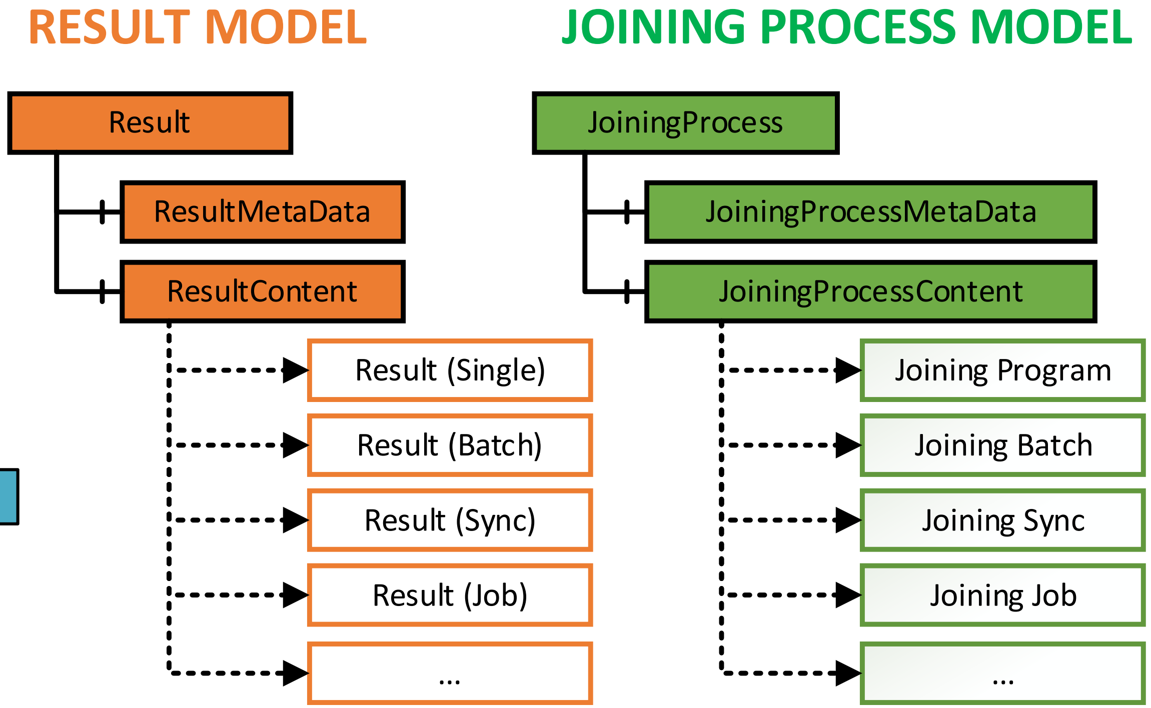
Result

Result

Result

Result

Result



RESULT MODEL

Result

ResultMetaData

ResultContent

Result (Single)

Result (Batch)

Result (Sync)

Result (Job)

…

JOINING PROCESS MODEL

JoiningProcess

JoiningProcessMetaData

JoiningProcessContent

Joining Program

Joining Batch

Joining Sync

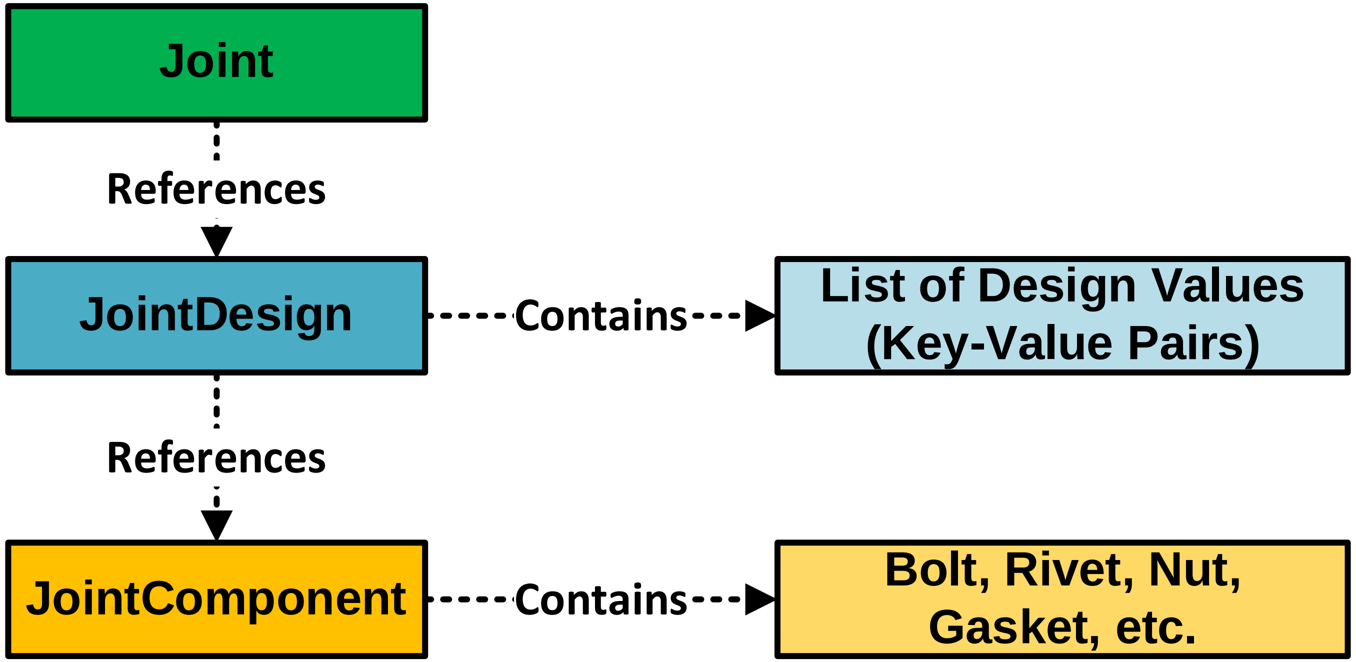
Joining Job

…

# Joint Management

# Joint Management

- **Use Case:** The user could send information on **what** type of joint needs to be done instead of **how** the joining operation should be executed.
- Definition of Bolt, Rivet is vendor-specific.


Joint
References
JointDesign
Contains
List of Design Values (Key-Value Pairs)
References
JointComponent
Contains
Bolt, Rivet, Nut, Gasket, etc.

# Joining System Method List (Commands)

# Joining System Method List

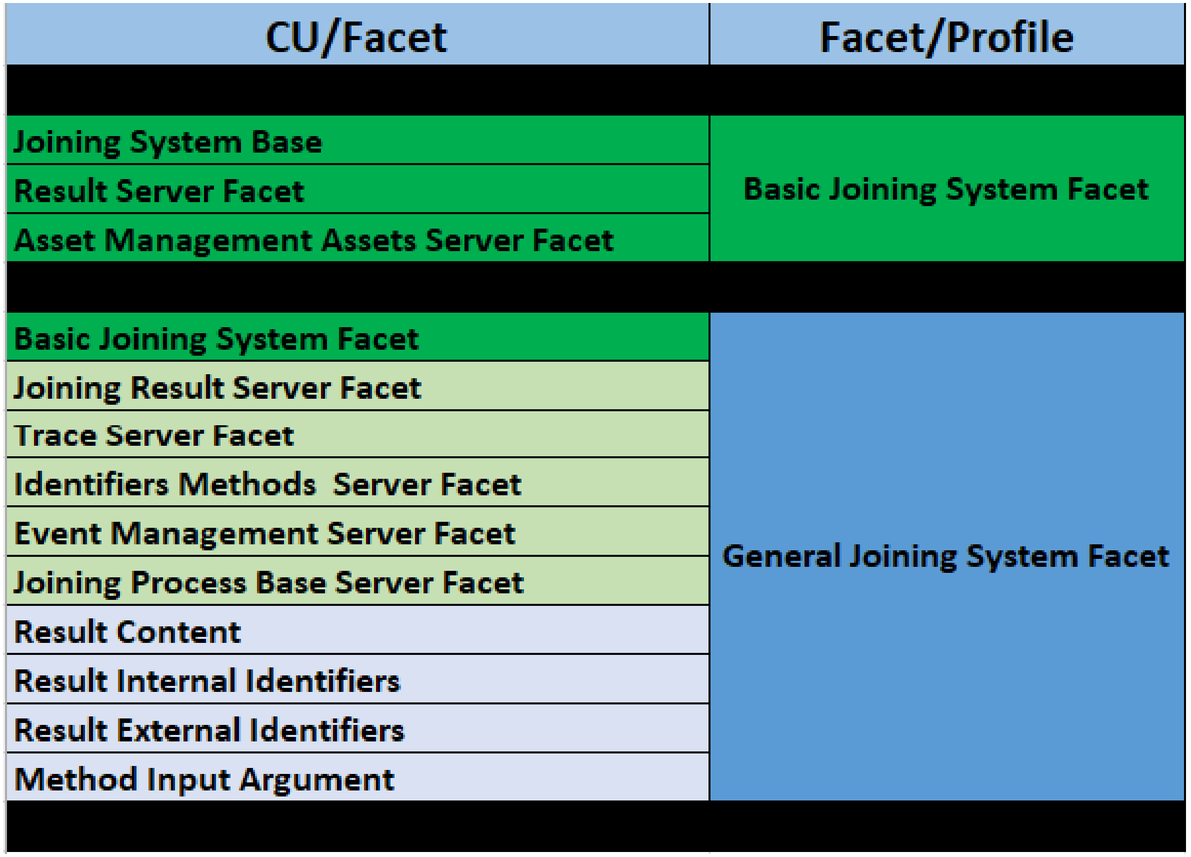
| Asset Management | Result Management | Joining Process Management | Joint Management |
|---|---|---|---|
| SetCalibration | GetLatestResult | SendJoiningProcess | SendJoint |
| EnableAsset | GetResultById | GetJoiningProcess | SendJointDesign |
| DisconnectAsset | GetResultIdListFiltered | GetJoiningProcessList | SendJointComponent |
| RebootAsset | ReleaseResultHandle | GetJoiningProcessRevisionList | GetJointList |
| SendFeedback | AcknowledgeResults | SetJoiningProcessMapping | GetJointRevisionList |
| GetFeedbackFileList | RequestRequests | SelectJoiningProcess | GetJointDesignList |
| SetOfflineTimer | RequestUnacknowledgedResults | DeselectJoiningProcess | GetJointComponentList |
| SetTime | | IncrementJoiningProcessCounter | GetJoint |
| SendIOSignals | | DecrementJoiningProcessCounter | GetJointDesign |
| GetIOSignals | | SetJoiningProcessCounter | GetJointComponent |
| SendIdentifiers | | SetJoiningProcessSize | SelectJoint |
| SendTextIdentifiers | | ResetJoiningProcess | DeleteJoint |
| GetIdentifiers | | AbortJoiningProcess | DeleteJointDesign |
| ResetIdentifiers | | StartJoiningProcess | DeleteJointComponent |
| ExecuteOperation | | StartSelectedJoining | |
| GetErrorInformation | | DeleteJoiningProcess | |
| | | GetSelectedJoiningProgram | |
| | | | |

# IJT Profiles Overview

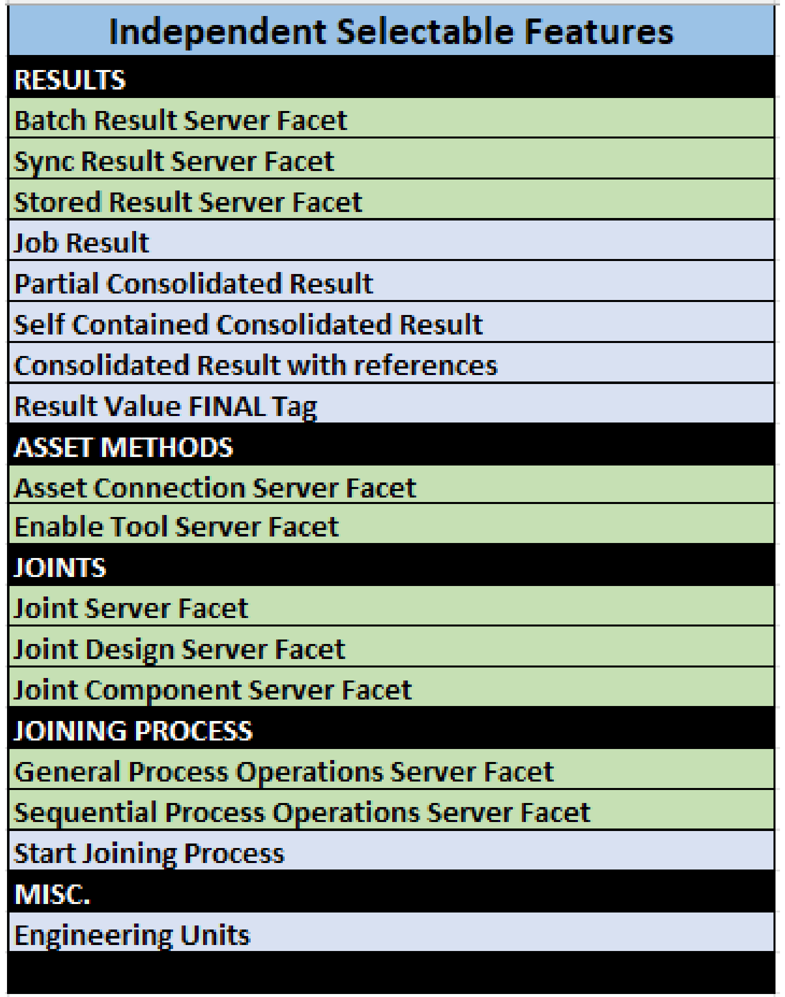
# IJT Profiles Overview

- OPC UA Specifications define a set of rules that are needed for a product to be certified.
  - It is done using the definition of Conformance Units, Facets and Profiles.

- The following image provides an overview of profiles defined in IJT specifications.



| CU/Facet | Facet/Profile |
|---|---|
| Joining System Base | Basic Joining System Facet |
| Result Server Facet | |
| Asset Management Assets Server Facet | |
| Basic Joining System Facet | General Joining System Facet |
| Joining Result Server Facet | |
| Trace Server Facet | |
| Identifiers Methods  Server Facet | |
| Event Management Server Facet | |
| Joining Process Base Server Facet | |
| Result Content | |
| Result Internal Identifiers | |
| Result External Identifiers | |
| Method Input Argument | |



| Independent Selectable Features |
|---|
| RESULTS |
| Batch Result Server Facet |
| Sync Result Server Facet |
| Stored Result Server Facet |
| Job Result |
| Partial Consolidated Result |
| Self Contained Consolidated Result |
| Consolidated Result with references |
| Result Value FINAL Tag |
| ASSET METHODS |
| Asset Connection Server Facet |
| Enable Tool Server Facet |
| JOINTS |
| Joint Server Facet |
| Joint Design Server Facet |
| Joint Component Server Facet |
| JOINING PROCESS |
| General Process Operations Server Facet |
| Sequential Process Operations Server Facet |
| Start Joining Process |
| MISC. |
| Engineering Units |

# Status

- The specifications were formally released and published in March 2024.

- OPC 40450-1 UA CS for Joining Systems **1.00.0**
  - https://reference.opcfoundation.org/nodesets?u=http://opcfoundation.org/UA/IJT/Base/

- OPC 40451-1 UA CS for Tightening Systems **2.00.0**
  - https://reference.opcfoundation.org/nodesets?u=http://opcfoundation.org/UA/IJT/Tightening/
  - Release 1 was published in **October** 2021.

# Prototypes / Simulators and Demonstration

# Prototypes / Simulators / Reference Implementation

- The following portal contains simulators and prototypes for both Clients and Servers.
  - https://github.com/umati/UA-for-Industrial-Joining-Technologies
  - **OPC_UA_Servers/Release1**
    - Available
  - **OPC_UA_Servers/Release2**
    - Available
  - **OPC_UA_Clients/NodeOPCUA_IJT_Client**
    - A client to connect to the OPC UA IJT CS model.
  - The reference implementations are updated regularly.

# Demonstrations

- Demonstrations are done using umati infrastructure.
- **What is umati?**
  - It is a common dashboard connected to several machines from various vendors using a standard information model.
  - Refer to https://umati.org/ for more information.
- Demonstration: https://umati.app/
  - Done in automatica 2022, 2023 and 2025.

# Release 1 and Release 2 Differences

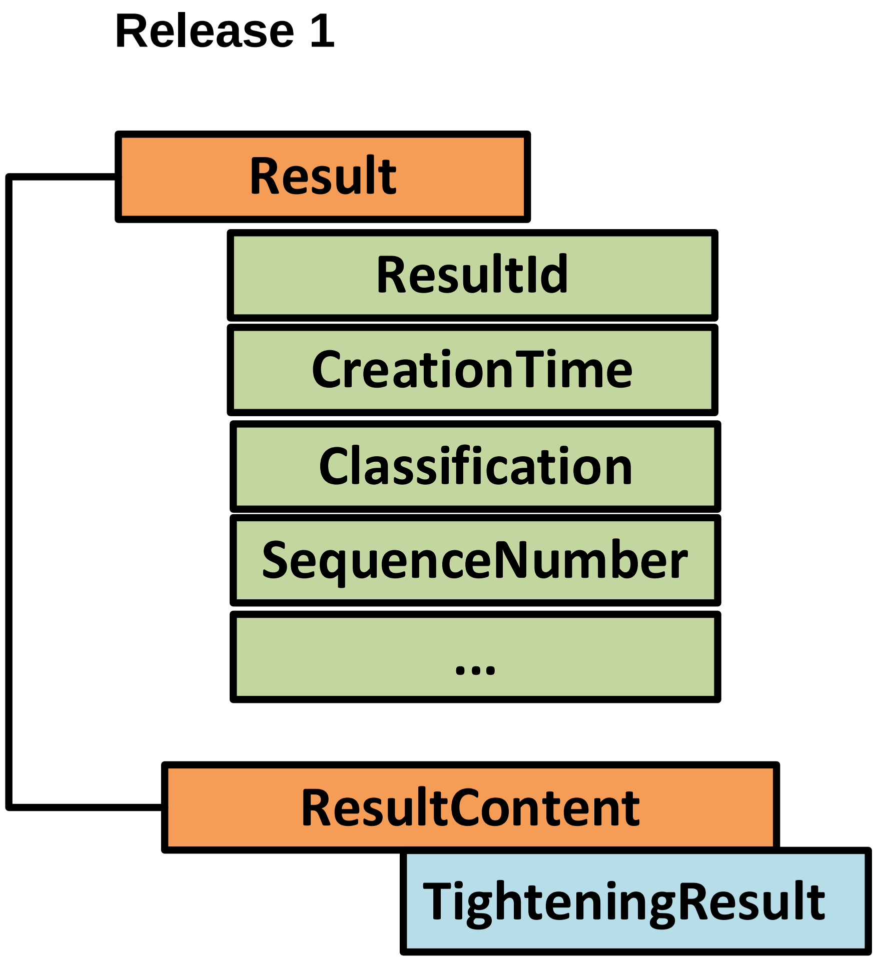
# Overview

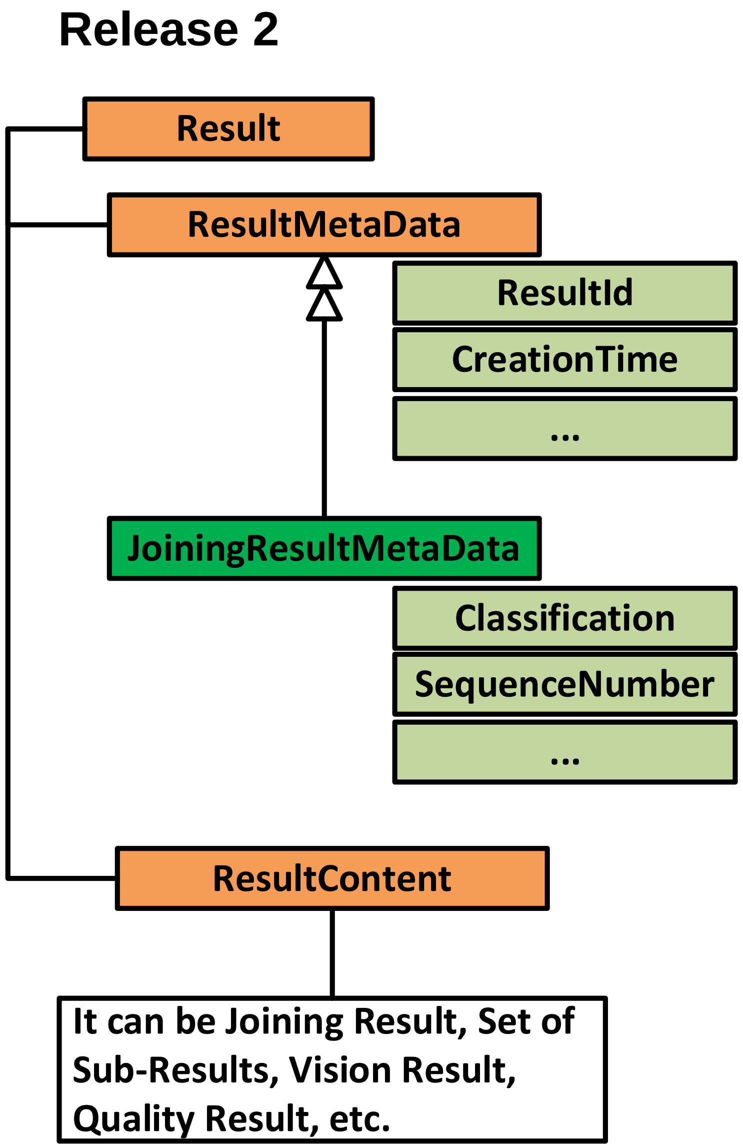
- **Release 1** was **only** for Tightening Domain.
  - OPC 40451-1 UA CS for Tightening Systems **1.00.0.**

- During **Release 2** work, most of the use cases are generalized for various joining systems. Hence, the models were **extended** and **moved** to a new **base** specification.
  - OPC 40450-1 UA CS for Joining Systems **1.00.0.**
  - Tightening Specification is upgraded to OPC 40451-1 UA CS for Tightening Systems **2.00.0.**

- **Why is it a major version release?**
  - Most of the models from the Tightening Specification are generalized and moved to a new base specification.

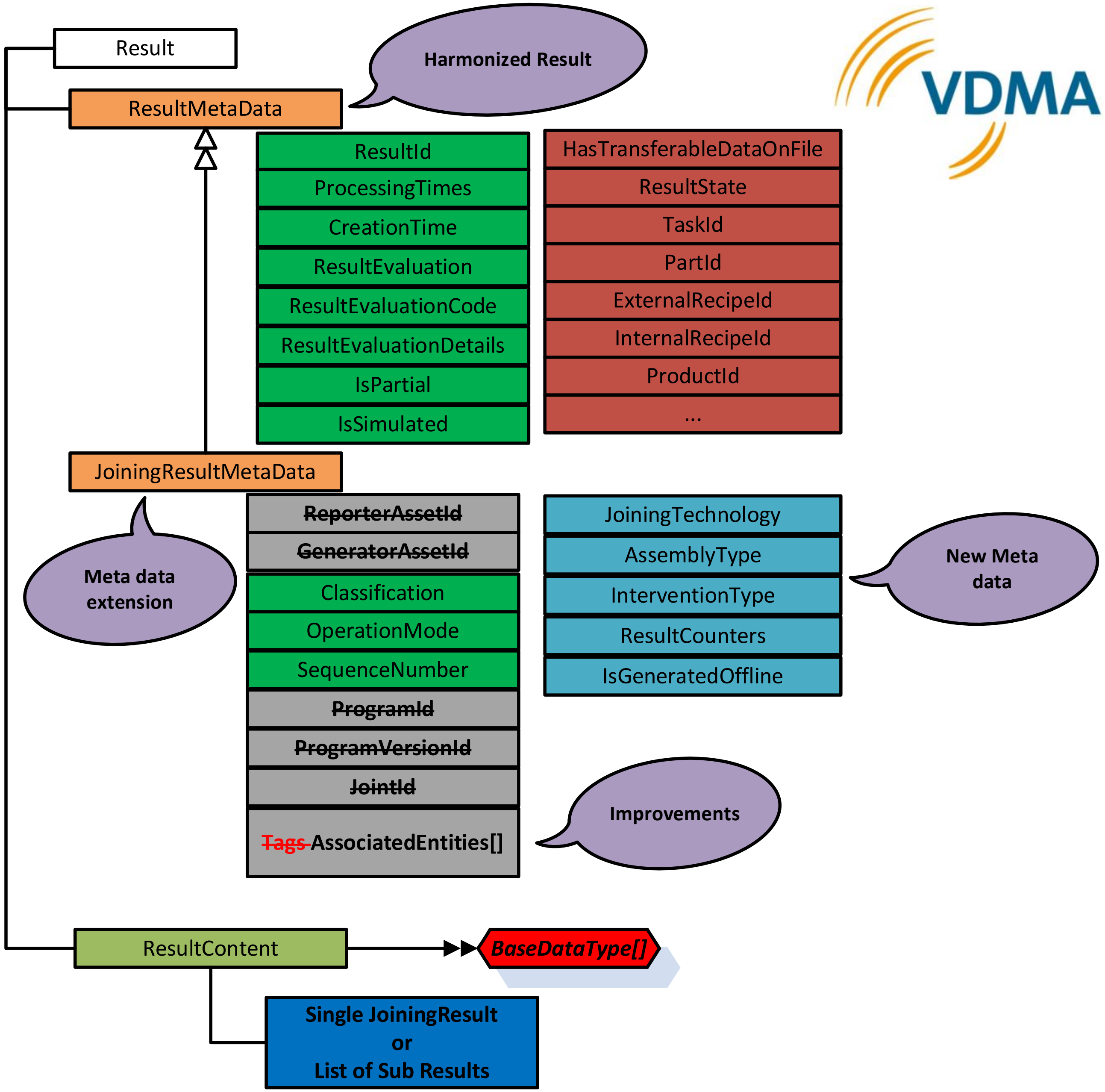
# Result Structure Changes





# Result Data Changes

- **Machinery Result** Usage from Harmonization Group.
  - **Result** properties **grouped** as **ResultMetaData**.
- **Additional** set of properties to cover the use cases for Batch/Job/Sync Result, etc.
- **Generalization** of **Tightening** Result to **Joining** Result.
- **Uses** UA 1.05.02 Structures.



Result

ResultMetaData

**Harmonized Result**

ResultId
ProcessingTimes
CreationTime
ResultEvaluation
ResultEvaluationCode
ResultEvaluationDetails
IsPartial
IsSimulated

HasTransferableDataOnFile
ResultState
TaskId
PartId
ExternalRecipeId
InternalRecipeId
ProductId
...

JoiningResultMetaData

**Meta data extension**

~~ReporterAssetId~~
~~GeneratorAssetId~~
Classification
OperationMode
SequenceNumber
~~ProgramId~~
~~ProgramVersionId~~
~~JointId~~
~~Tags~~ AssociatedEntities[]

JoiningTechnology
AssemblyType
InterventionType
ResultCounters
IsGeneratedOffline

**New Meta data**

**Improvements**

ResultContent

*BaseDataType[]*

**Single JoiningResult**
**or**
**List of Sub Results**

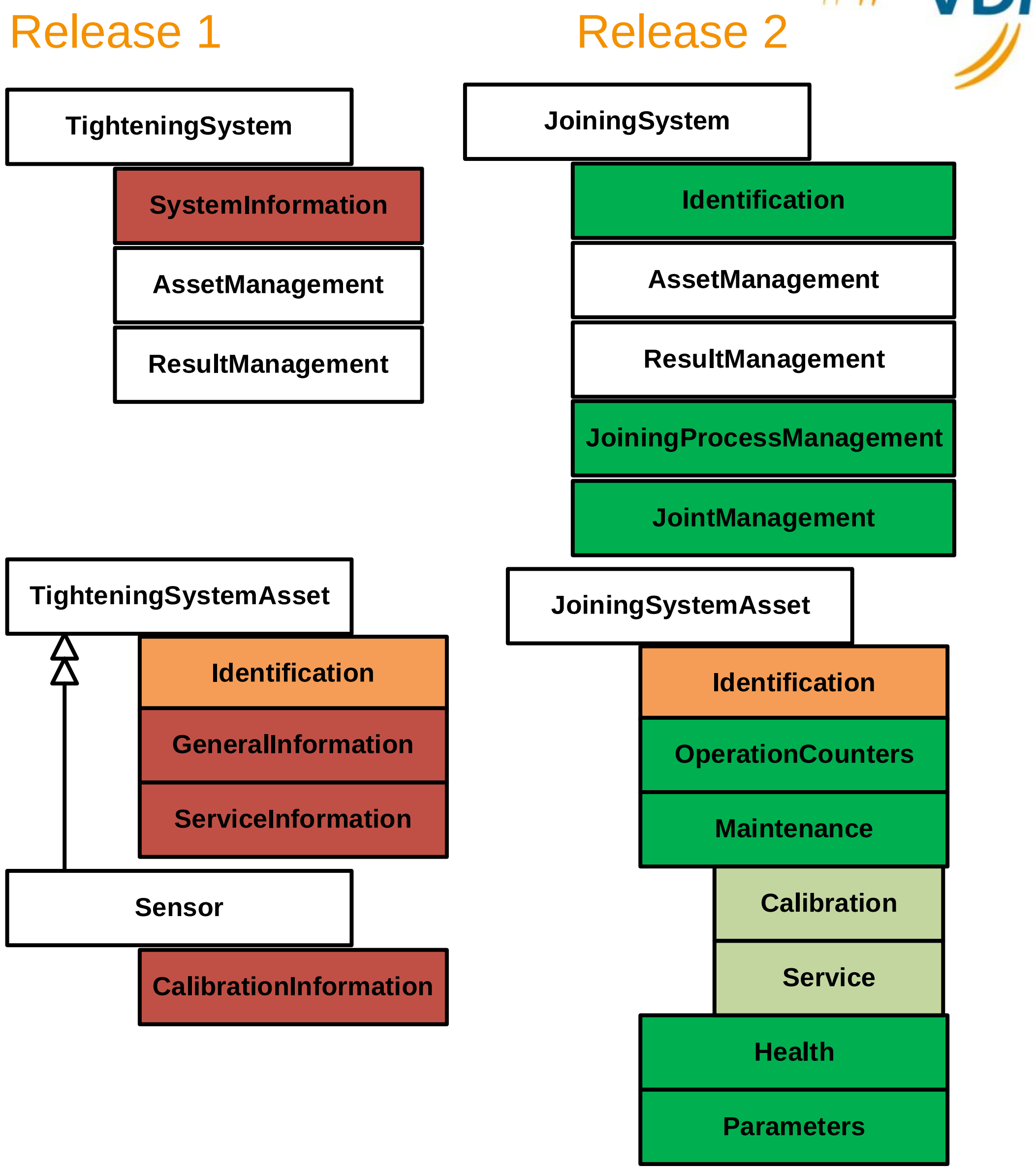
# System, Asset and Event Changes

- **Improvement** in the Joining System for Identification AddIn.

- **Extension** of Joining System to include newer models.

- **Reorganization** and **Extension** of Asset Management Model.
  - Software and Virtual Station as new assets.

- **Extension** of Event model by **common joining payload** and **filtering** mechanism.



# Maintenance, Migration, Extensions

# Extensions, Maintenance and Other Specifications

- The long-term vision to have a common base for various joining technologies is in place.

- In future, if there are very specific use cases needed for Riveting, Gluing, etc. which are not covered in the joining base specification, the specification can be defined.

- **Possible Future Use Cases**
    - Reuse of Network Structure for Assets from other harmonization groups.
    - Reuse of Location models.
    - Definition of Part, Product, etc.
    - Definition of Threaded Fastener/Bolts.
    - Issues in the existing specification.
    - …

# Greenfield vs Brownfield in Joining Domain

- For **brownfields**, a phased migration can be done. It **depends** on the use case and other factors.
    - **Example:** For control mechanisms, existing legacy protocols can be used and OPC UA can be deployed for data acquisitions.

- For **greenfields**, it is recommended to use OPC UA with IJT Companion Standard.

- **OPC UA** and **other** protocols can **coexist** together.

# Reference Links

- **OPC UA IJT Overview:**
  - https://opcfoundation.org/markets-collaboration/IJT/
- **Specifications/Online Reference**
  - https://reference.opcfoundation.org/nodesets?u=http://opcfoundation.org/UA/IJT/Base/
  - https://reference.opcfoundation.org/nodesets?u=http://opcfoundation.org/UA/IJT/Tightening/
- **OPC UA IJT Prototypes/Reference Implementations**
  - https://github.com/umati/UA-for-Industrial-Joining-Technologies
- **Access to Specification Documents:**
  - **Formally** released specifications can be accessed for free from the OPC Foundation Portal. Use Online Reference to view the specifications in the browser without any login/registration.
  - **Release Candidate (DRAFT)** is **only** accessible for an OPC Foundation Member Company. The alternative is to download from the VDMA portal: https://vdma.org/der-verband
    - Search the portal with the specification number. **Examples:** 40450, 40451, etc.

# Additional Slides

# OPC Legacy and OPC UA History

- Refer to the following link which explains the Before and After OPC Scenarios:
- https://www.ia.omron.com/product/special/sysmac/nx1/opcua.html

**Source: https://www.ia.omron.com/product/special/sysmac/nx1/opcua.html**

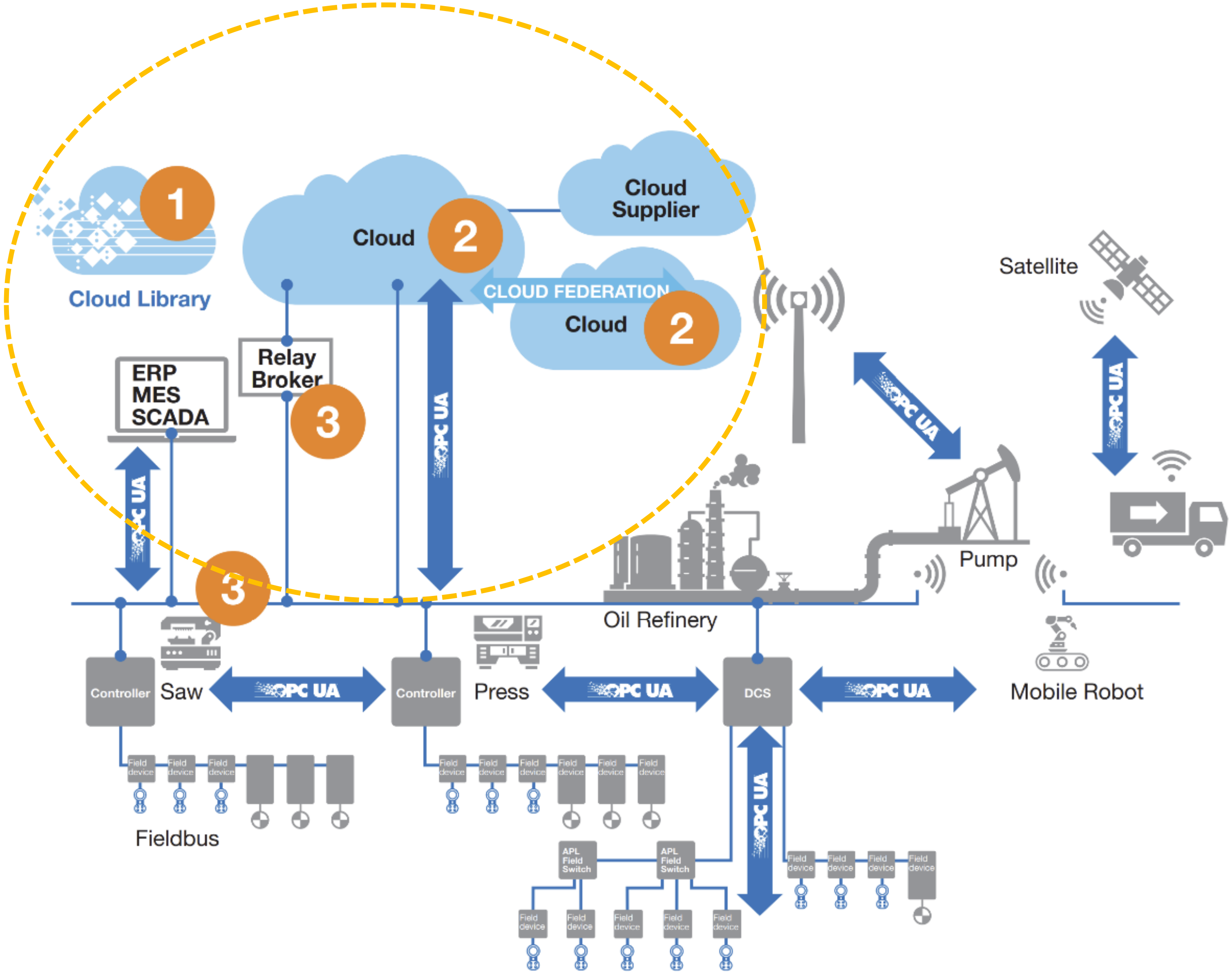
# OPC UA Overview and Benefits

Industry 4.0 main contender for industrial interoperability standard

Core Characteristics:

- Service Oriented Architecture (Subscription, Methods, Events, etc.)
- Platform independent (Any hardware - operating system - programming language)
- Integrated security mechanisms (Message Encryption, Authentication, User Rights, Certificates)
- Standard discovery mechanisms (Discovering devices and management of certificates)
- Comprehensive information model

# UA Cloud Initiative



1. **Cloud Library**
   - Repository for OPC UA based information models (IMs)
   - Upload, store, search, download IMs

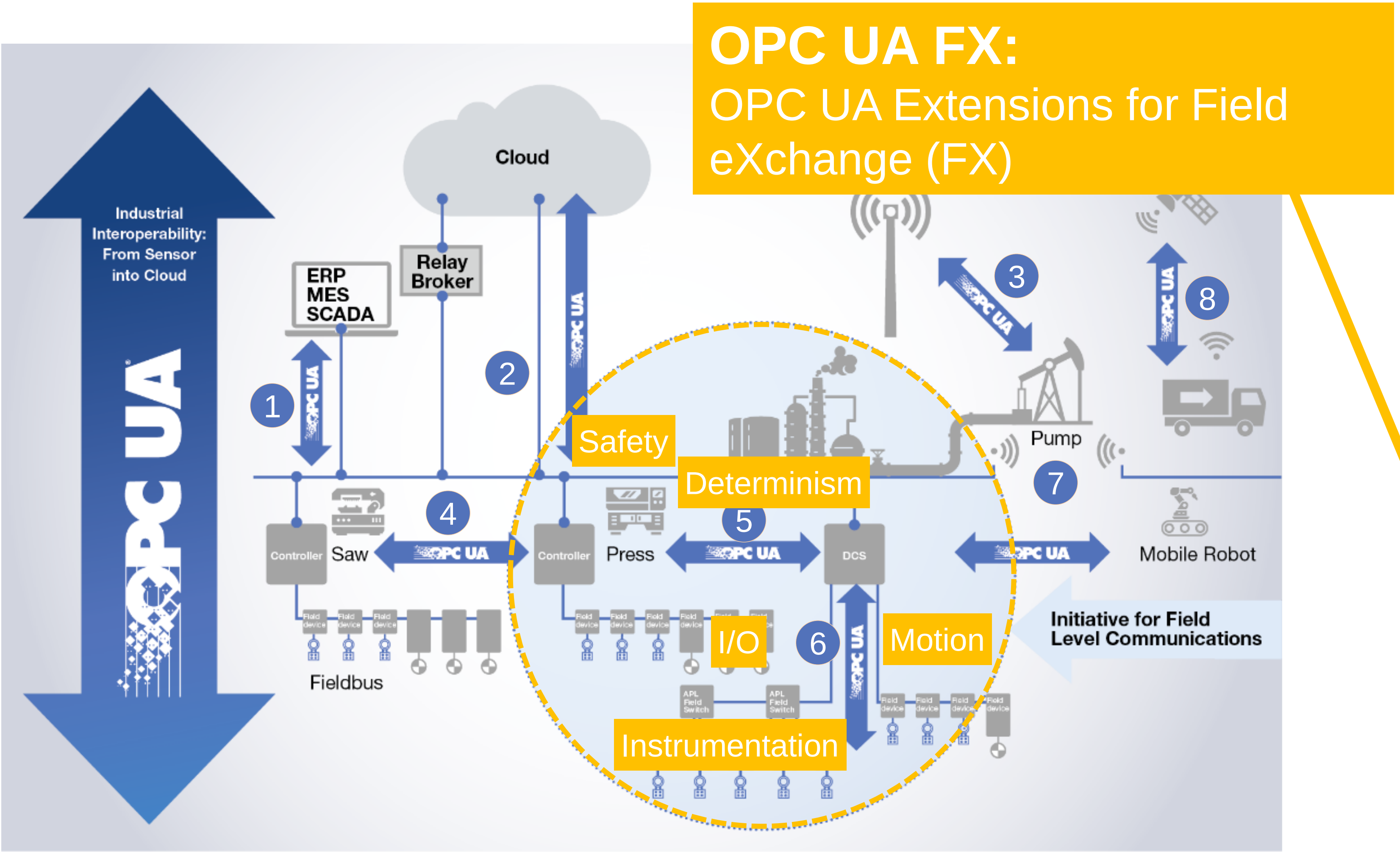
2. **Cloud Federation**
   - Standardized communication
   - Cloud to Cloud

3. **Asset / Edge / Cloud**
   - Standardized communication
   - Field to Cloud
   - Cloud to Field

4. **Education, IIOT Starter Kit**
   - Success stories

Source: OPC Foundation

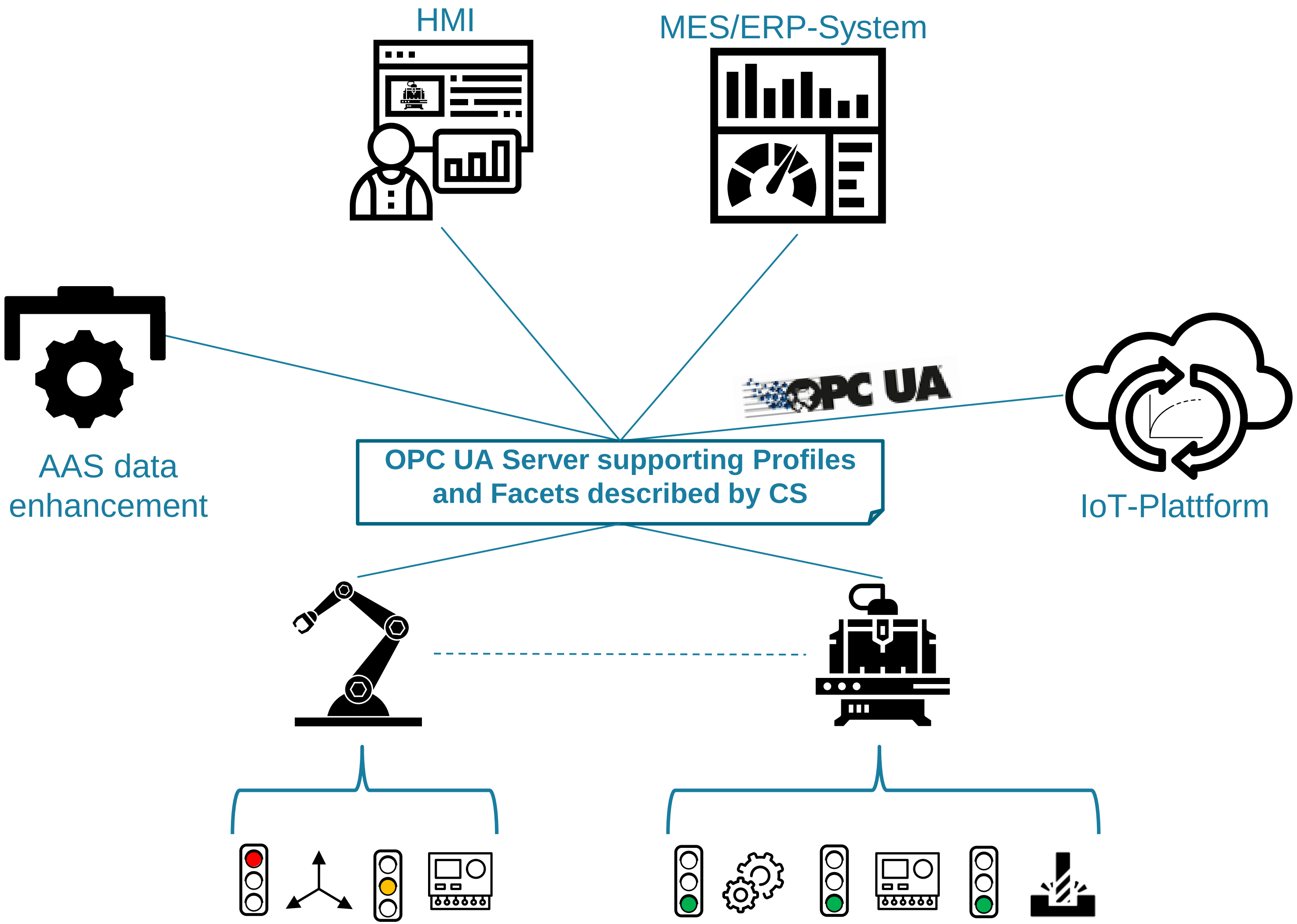
# OPC UA for Field eXchange (FX):
# Extending OPC UA to the field incl. Determinism, Safety & Motion



1. IT / OT Communication
2. Cloud Integration
3. Secure Remote Access
4. Local OT Communication
5. Controller to Controller
6. Controller to Device incl. Device to Device
7. Wireless Integration (5G)
8. Future Ready

Source: OPC Foundation

# OPC UA as the standardized communication interface of machines & components


HMI
MES/ERP-System
OPC UA
AAS data enhancement
OPC UA Server supporting Profiles and Facets described by CS
IoT-Plattform

# Contact

**Bernd Heitzmann**

**Director VDMA Integrated Assembly Solutions (VDMA IAS)**

| | |
|---|---|
| Phone | +49 (0) 69 6603 1530 |
| E-Mail | Bernd.Heitzmann@vdma.org |

**Mohit Agarwal**

**Principal Engineer**

**Atlas Copco Industrial Technique, Sweden**

**Editor of Industrial Joining Technologies Group**

| | |
|---|---|
| Phone | +46 (0) 766 96 19 90 |
| E-Mail | mohit.agarwal@atlascopco.com |

# Questions