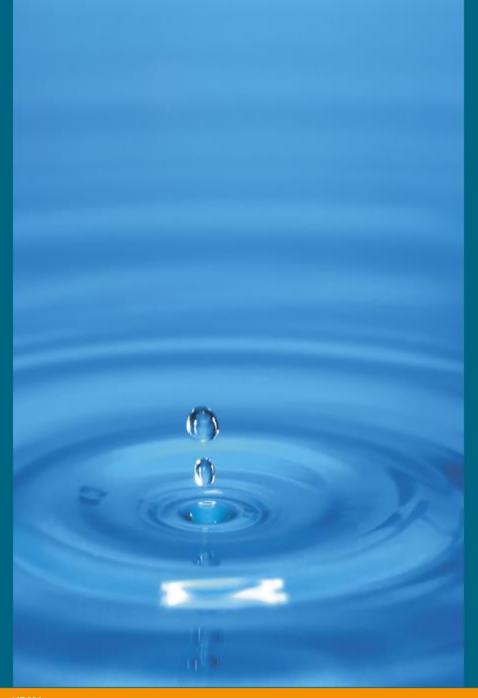


# **Getting Started**

# **Getting Started**

- Overview of OPC Foundation and OPC UA
  - https://opcfoundation.org/about/opc-technologies/opc-ua/
- OPC UA and IJT Overview
  - 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: <a href="https://reference.opcfoundation.org/IJT/Base/v100/docs/">https://reference.opcfoundation.org/IJT/Base/v100/docs/</a>
  - Tightening: <a href="https://reference.opcfoundation.org/lJT/Tightening/v200/docs/">https://reference.opcfoundation.org/lJT/Tightening/v200/docs/</a>
- 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



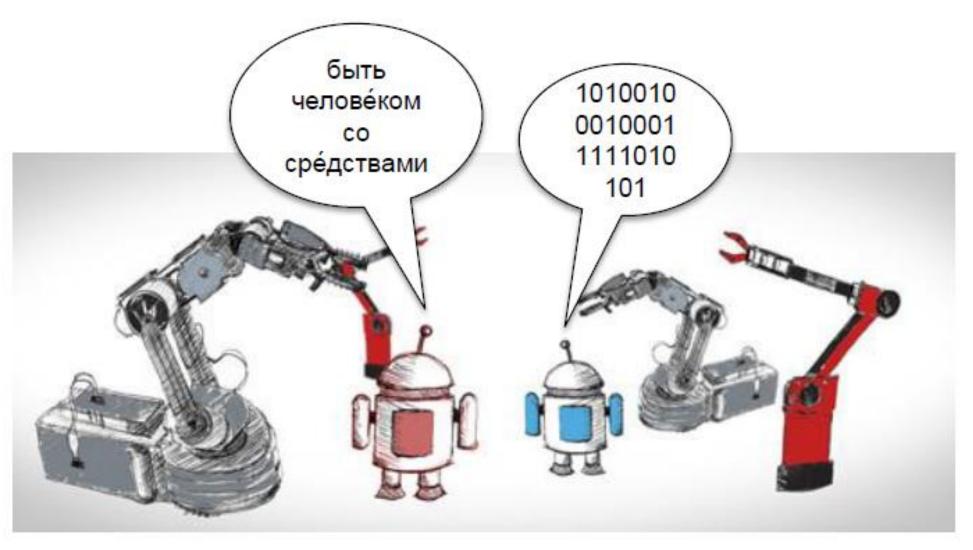
- » OPC UA Overview
- » Industrial Joining Technologies (IJT) Overview
- » IJT Use Cases
- » Technical Overview of Models
- » Demonstration
- » Questions



# **Quick OPC UA and IJT Overview**

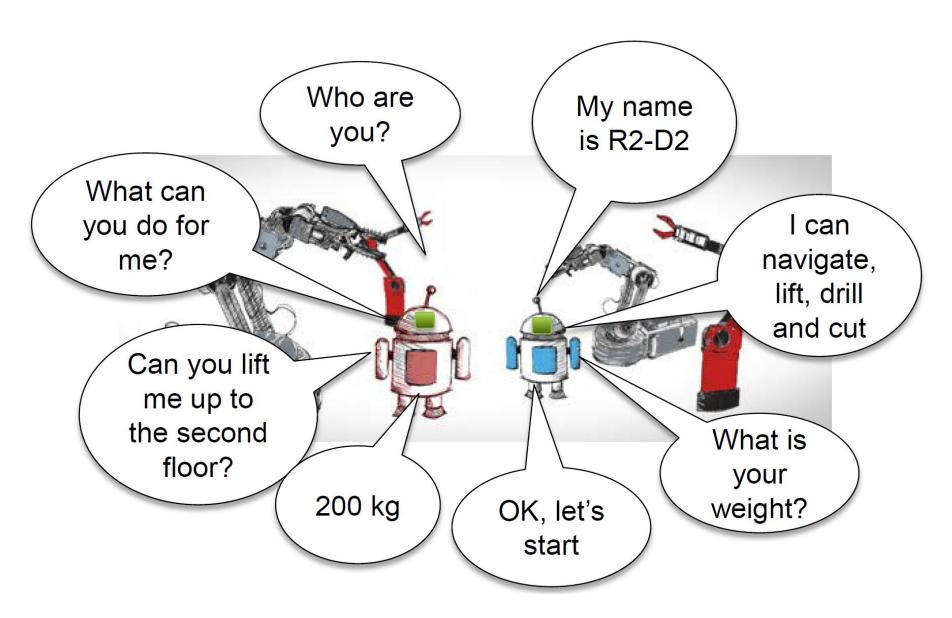
# **Today's Interoperability Issues**





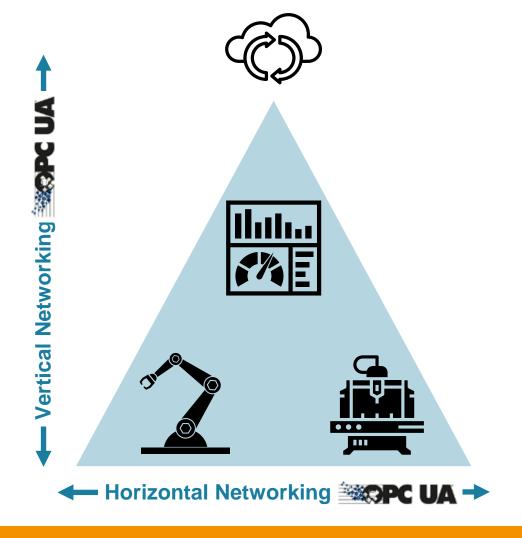
# **Today's Interoperability Solution**





#### **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

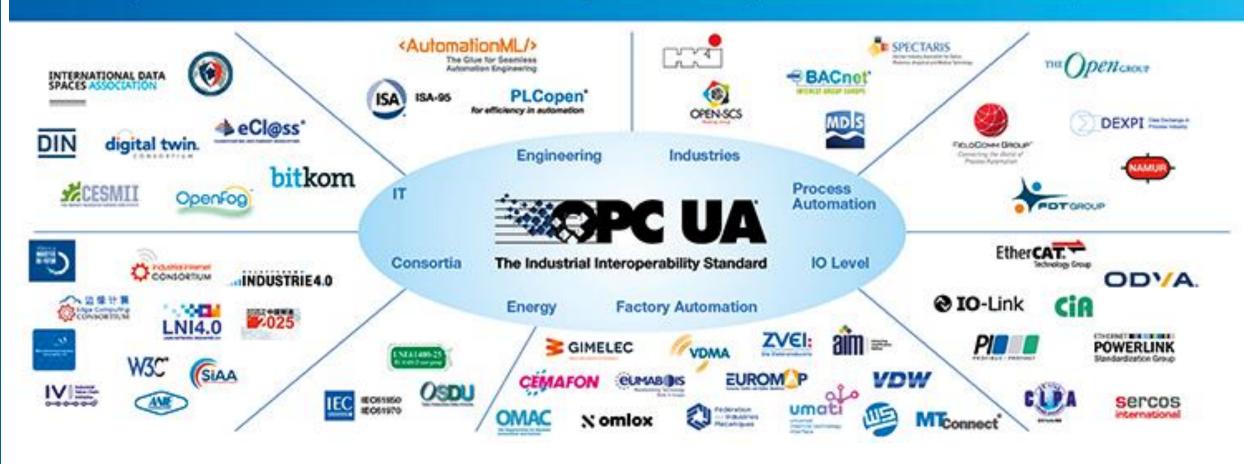
# **Collaboration Domain Specific Information Models**



# 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.



# **Interoperability Solution Summary**

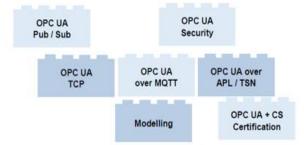
# (OPC UA + Companion Specs) = Promise for Interoperability



- 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

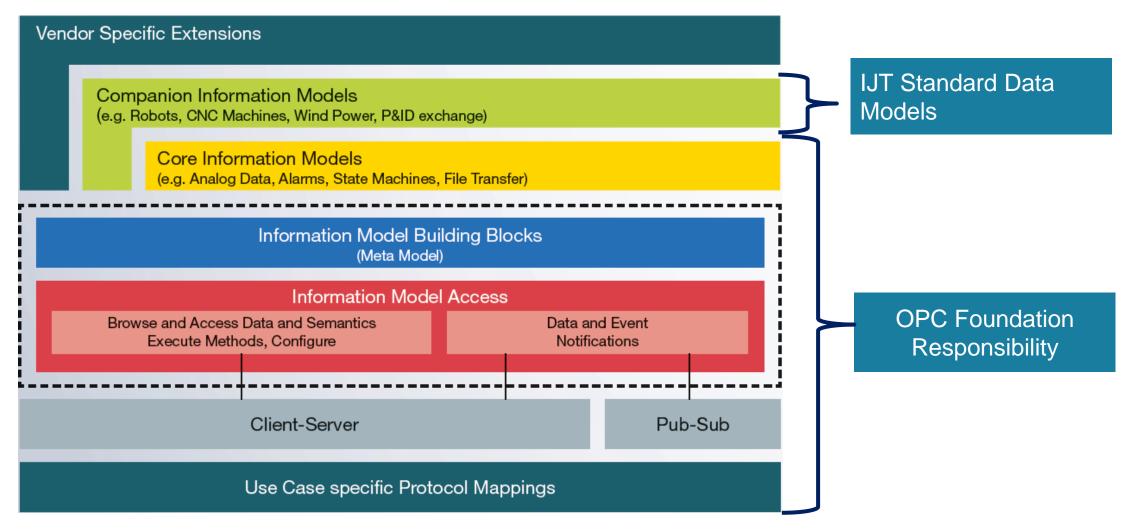
"understand each other"



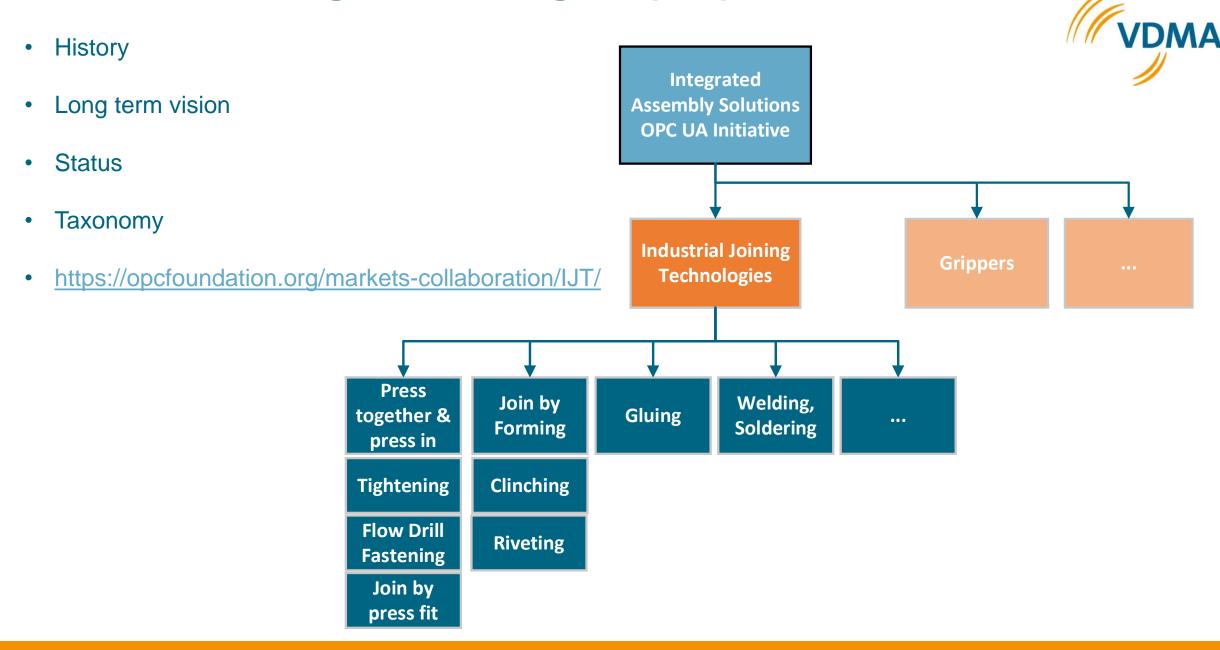




# **OPC UA Layered Architecture**



# Industrial Joining Technologies (IJT) Overview



# **IJT Working Group Members**























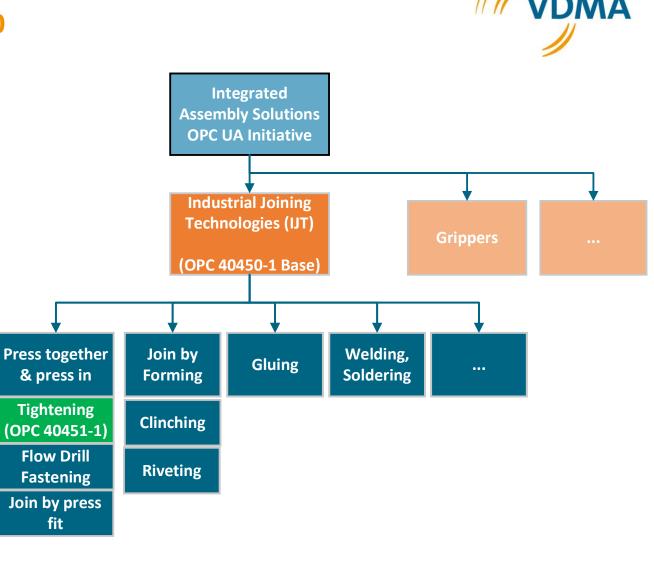






# IJT Specification(s) Overview

- OPC 40450-1 for Joining Systems version 1.0
  - Common elements of various joining technologies.
  - http://opcfoundation.org/UA/IJT/Base/
  - Opc.Ua.ljt.Base.NodeSet2.xml
- OPC 40451-1 for Tightening Systems version
   2.0
  - Version 1.0 was published in October 2021.
  - Moved common models to the base specification.
  - http://opcfoundation.org/UA/IJT/Tightening/
  - Opc.Ua.ljt.Tightening.NodeSet2.xml



# **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, Multispindle 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.



VDMA Seite 17 | 19/02/25

# The Key to Reducing Costs: Data Interoperability

VDMA

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 and OPC UA
  - https://opcfoundation.org/about/opc-technologies/opc-ua/
- OPC UA and IJT Overview
  - Refer to the OPC UA IJT Group Presentation.pdf/pptx.
  - https://opcfoundation.org/markets-collaboration/IJT/
- Specifications
  - https://opcfoundation.org/developer-tools/documents/view/343
  - https://opcfoundation.org/developer-tools/documents/view/341
- OPC UA IJT Prototypes/Reference Implementations
  - https://github.com/umati/UA-for-Industrial-Joining-Technologies
- OPC UA Online Reference
  - https://reference.opcfoundation.org/





# OPC UA Overview (IEC 62541)

# **Problem Statement**





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

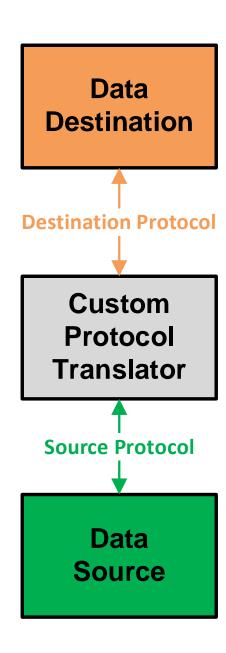
#### Traditional Solution – Custom Protocol Converter – Before OPC

# 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.

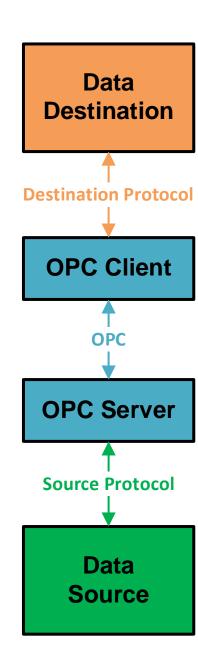
#### But it has the following limitations:

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



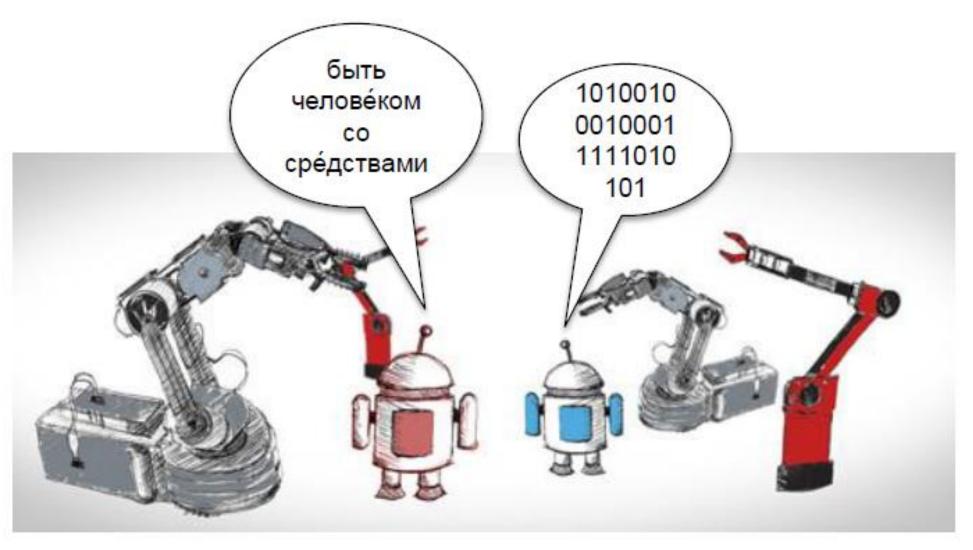
#### **OPC Connectivity – OPC Classic Data Path**

- 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



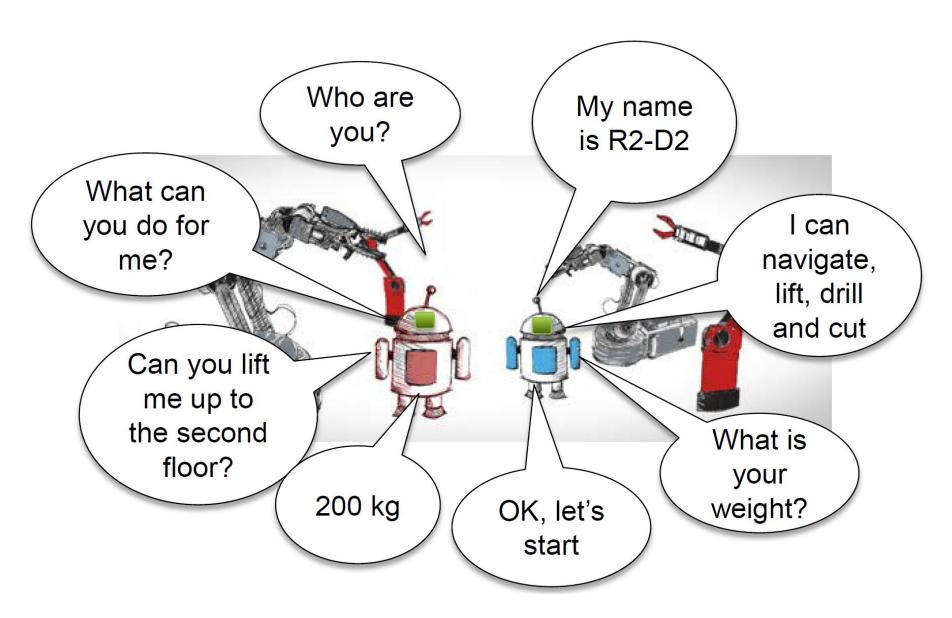
# **Today's Interoperability Issues**





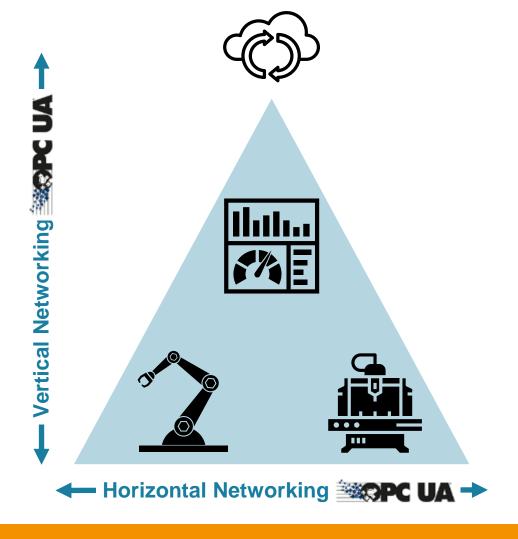
# **Today's Interoperability Solution**





#### **OPC UA - Field to Cloud**







Open source



Security



Two transport mechanism with various protocols



Scalable



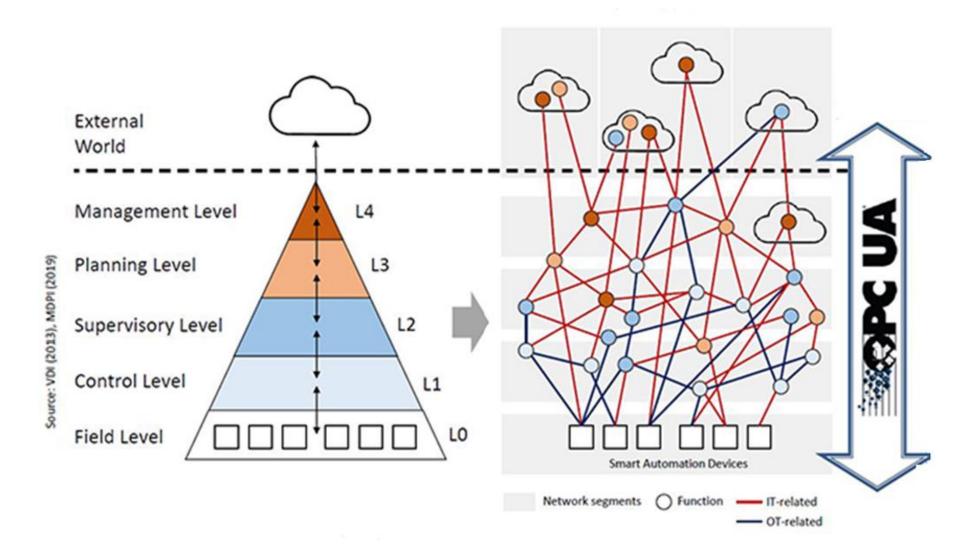
Global acceptance



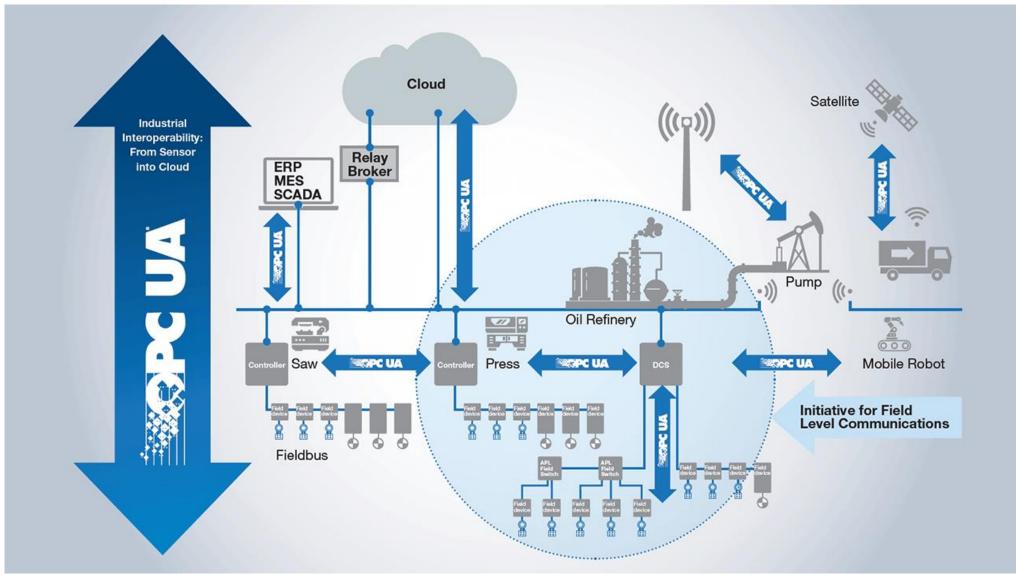
Semantic information models

# From Automation Pyramid to Information Network



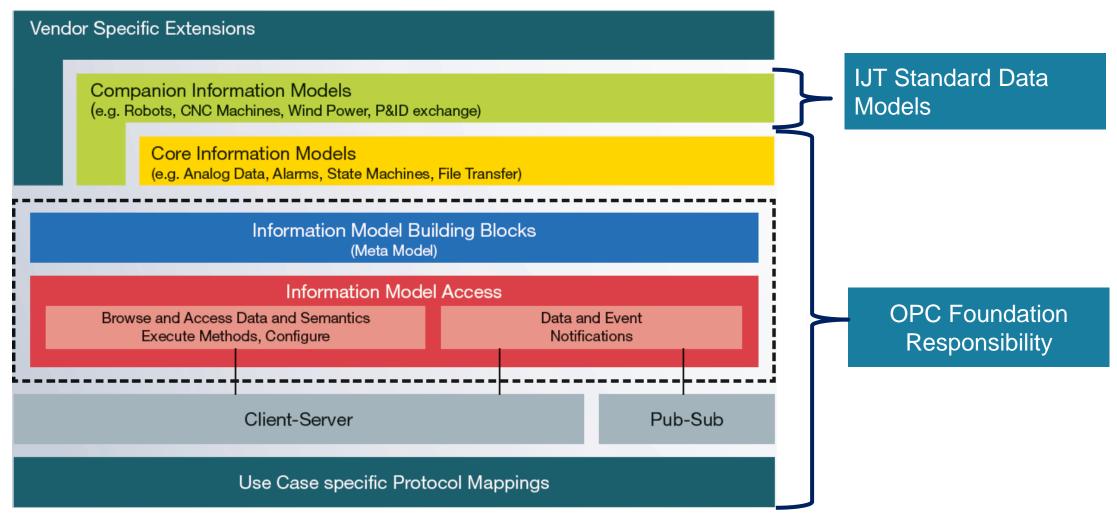


# **OPC UA Factory Network**





# **OPC UA Layered Architecture**





# **OPC UA Deployment Scenarios**

# **Platform Independence**

### Hardware

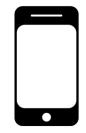














# **Operating Systems**





Real Time
Operating
System

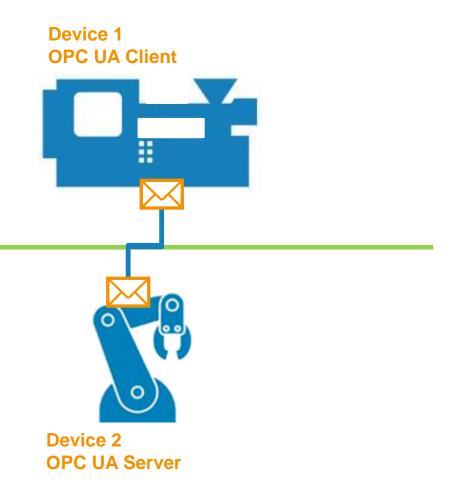






#### **OPC UA Client/Server Architecture**





#### **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**

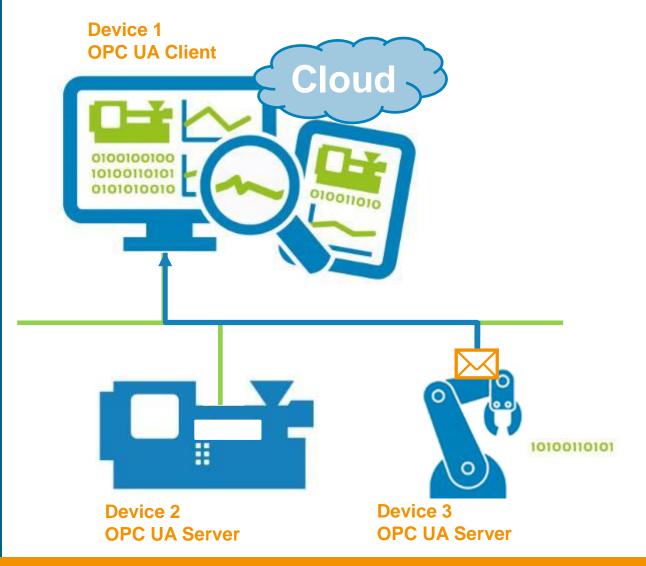
» Letter service with registered mail and advice of receipt.

# 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

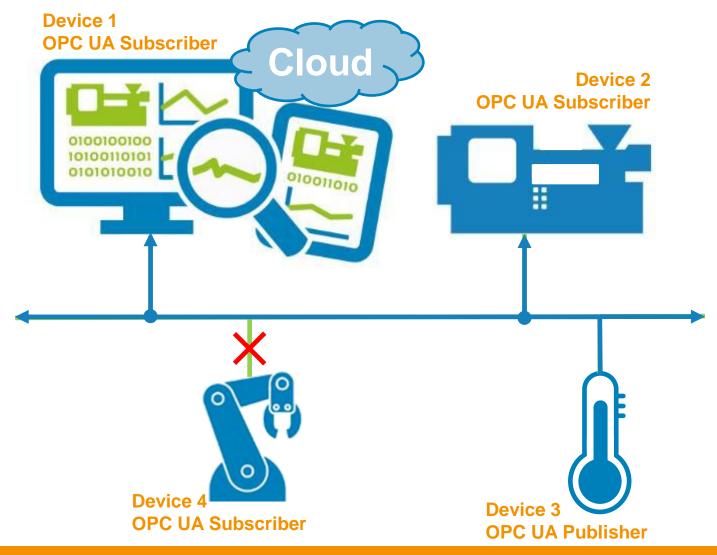
#### **Analogy**

» 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

### **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 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 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: <a href="https://opcfoundation.org/joint-working-groups/">https://opcfoundation.org/joint-working-groups/</a>
  - **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

VDMA

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**

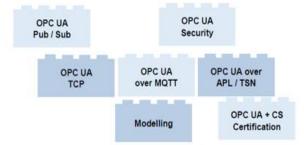
### (OPC UA + Companion Specs) = Promise for Interoperability



- OPC UA: 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

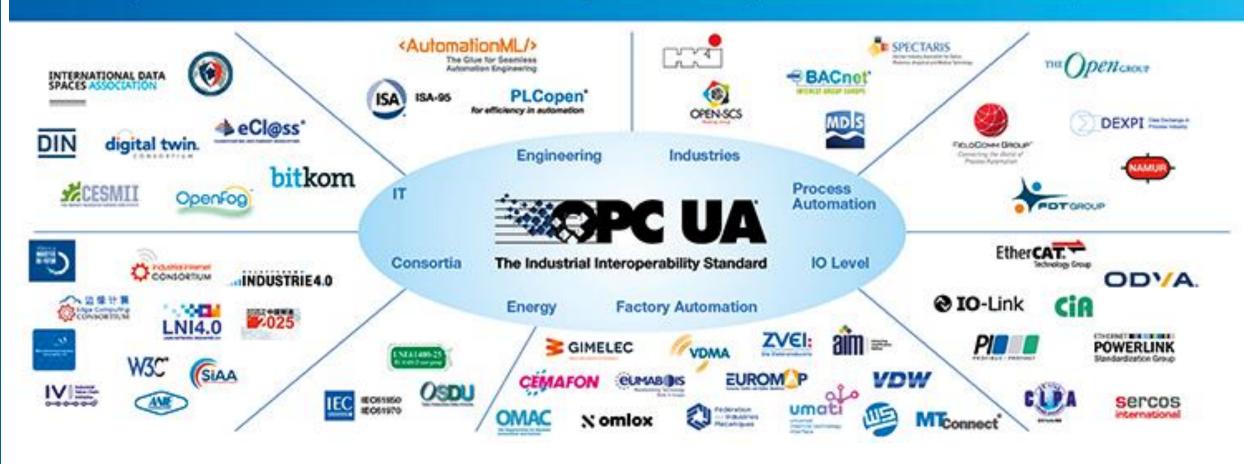
## **Collaboration Domain Specific Information Models**



### 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.

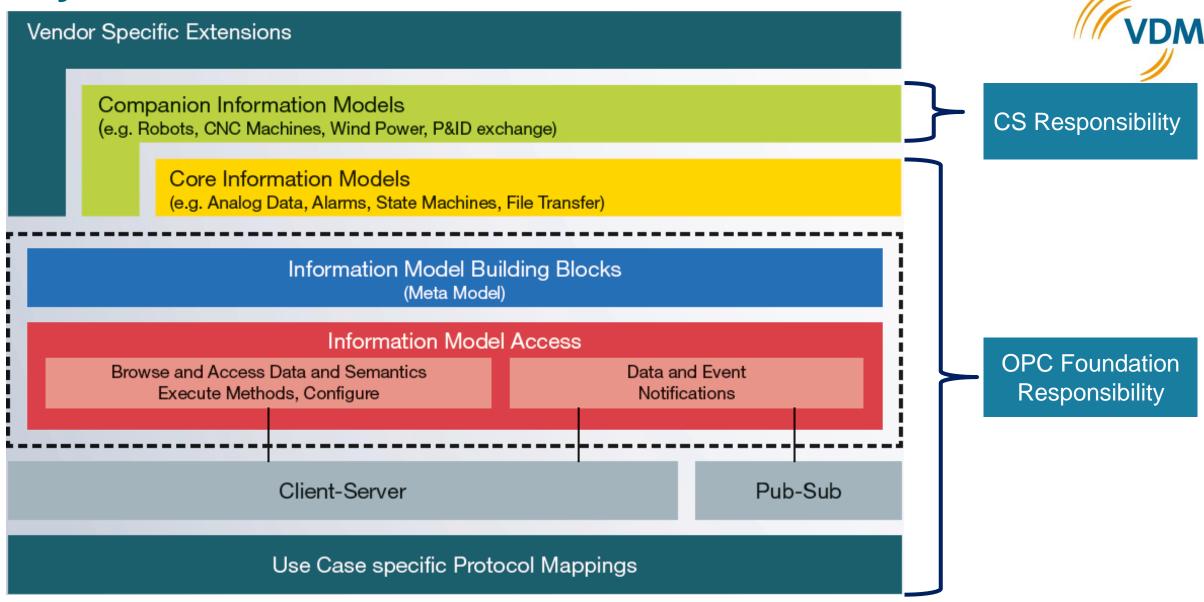


**Source: OPC Foundation** 



### **OPC UA Stack/SDK**

### **Layered Architecture**



**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: <a href="https://reference.opcfoundation.org/Core/Part7/v105/docs/3.1">https://reference.opcfoundation.org/Core/Part7/v105/docs/3.1</a>
  - Profile reporting tool: <a href="https://profiles.opcfoundation.org/">https://profiles.opcfoundation.org/</a>





## **Industrial Joining Technologies (IJT) Overview**

#### Industrial Joining Technologies (IJT) Overview History **Integrated Assembly Solutions** Long term vision **OPC UA Initiative** Status **Industrial Joining** Taxonomy **Grippers Technologies** https://opcfoundation.org/markets-collaboration/IJT/ **Press** Join by Welding, Gluing together & ••• **Soldering Forming** press in **Tightening** Clinching Flow Drill **Riveting Fastening** Join by press fit

### **IJT Working Group Members**























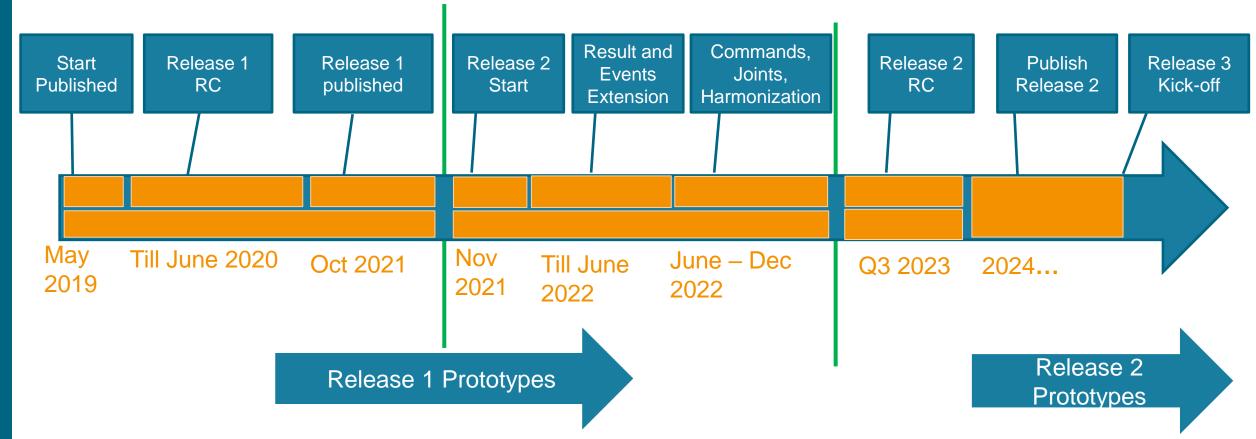






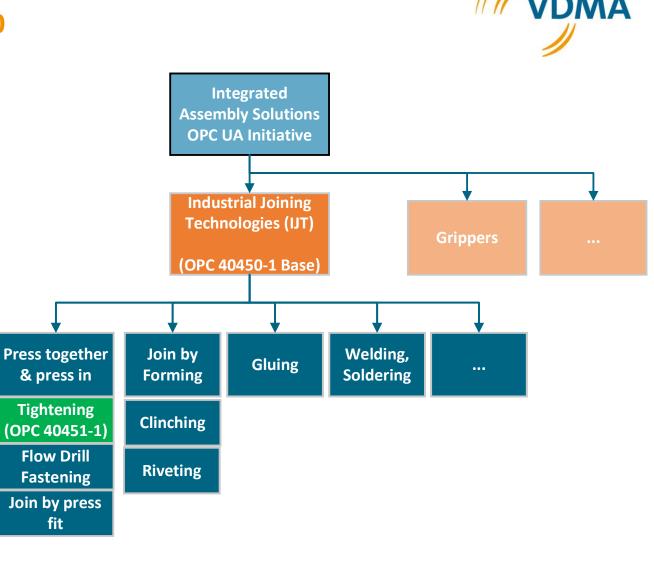
### **History and Milestones**





## IJT Specification(s) Overview

- OPC 40450-1 for Joining Systems version 1.0
  - Common elements of various joining technologies.
  - http://opcfoundation.org/UA/IJT/Base/
  - Opc.Ua.ljt.Base.NodeSet2.xml
- OPC 40451-1 for Tightening Systems version
   2.0
  - Version 1.0 was published in October 2021.
  - Moved common models to the base specification.
  - http://opcfoundation.org/UA/IJT/Tightening/
  - Opc.Ua.ljt.Tightening.NodeSet2.xml



# **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, Multispindle 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.



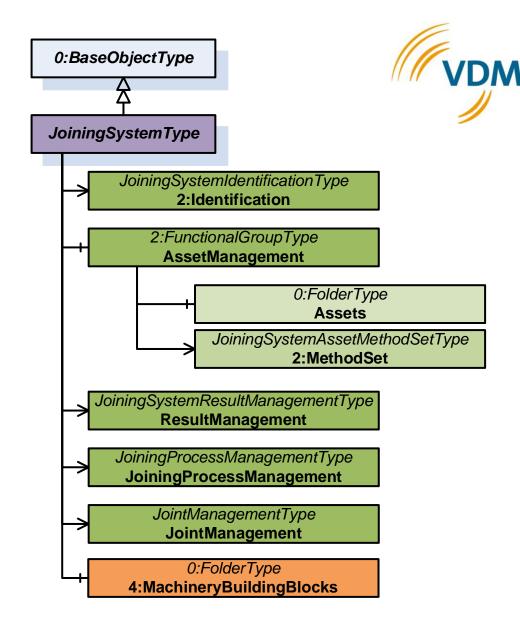
VDMA Seite 59 | 19/02/25



### **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
  - •





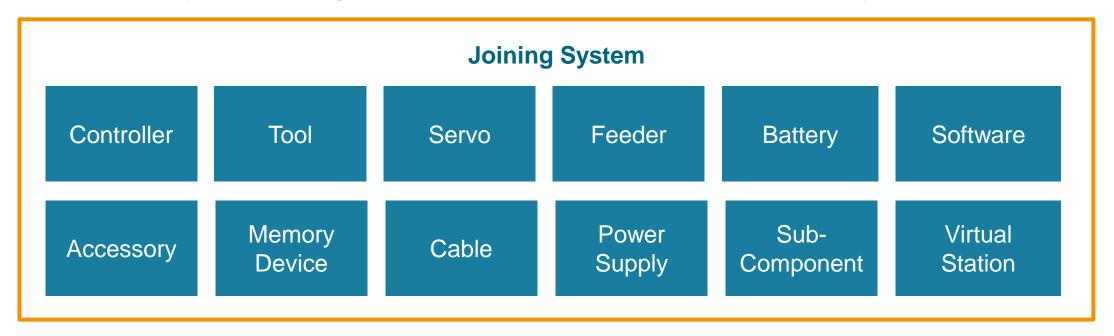
# **Asset Management**

### **Asset Management Overview**

Definition of assets building blocks.



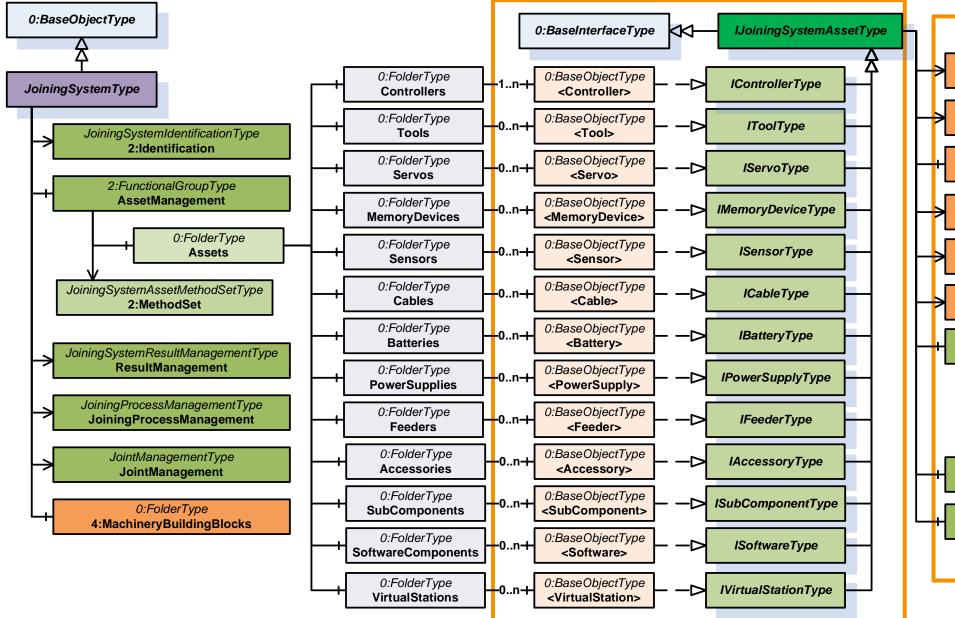
- Diverse systems Flexible asset management model to build different systems.
  - Fixtured, Handheld, Pneumatic, Multi-Channel, Single Channel
- Reuse of Machinery Building Blocks and Asset Management Basics.
- Future Extensibility with the usage of Interfaces and Add-Ins instead of concrete types.



**Asset Management** 

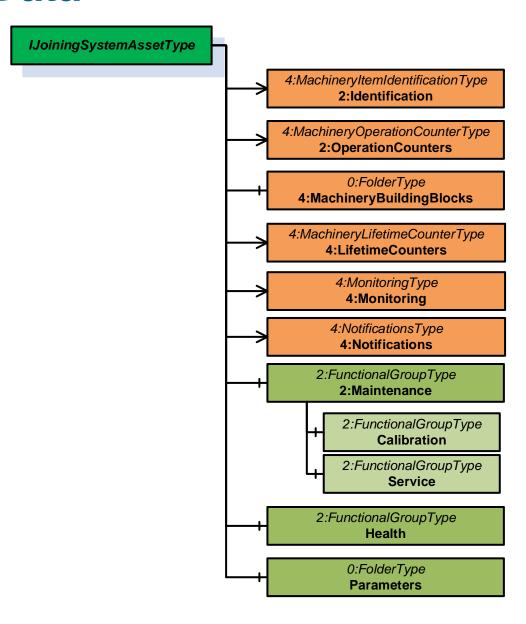
Flexible Asset Model...

Common Asset Data...



4:MachineryItemIdentificationType 2:Identification 4:MachineryOperationCounterType 2:OperationCounters 0:FolderType 4:MachineryBuildingBlocks 4:MachineryLifetimeCounterType 4:LifetimeCounters 4:MonitoringType 4:Monitoring 4:NotificationsType 4:Notifications 2:FunctionalGroupType 2:Maintenance 2:FunctionalGroupType Calibration 2:FunctionalGroupType Service 2:FunctionalGroupType Health 0:FolderType **Parameters** 

### **Common Asset Data**



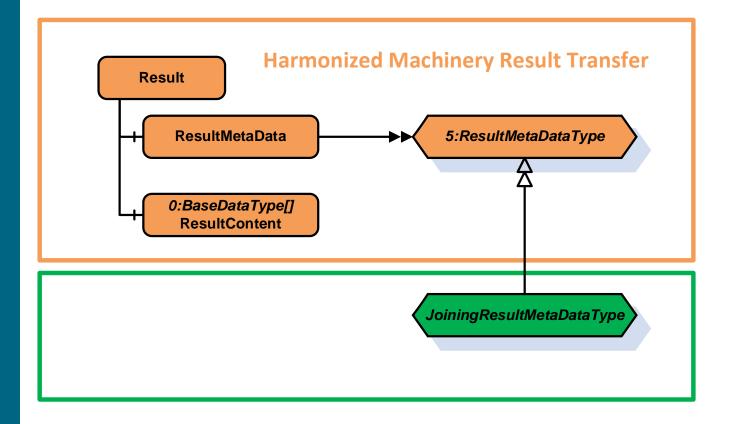




# **Result Management**

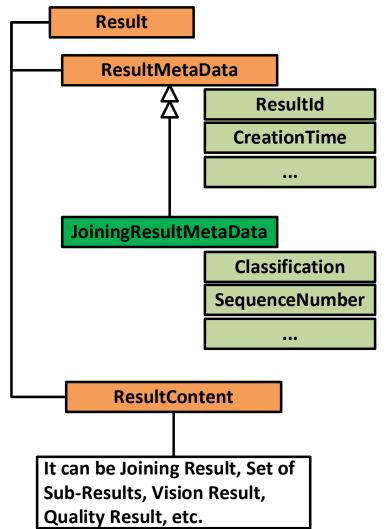
#### **Result Overview**

#### Structure



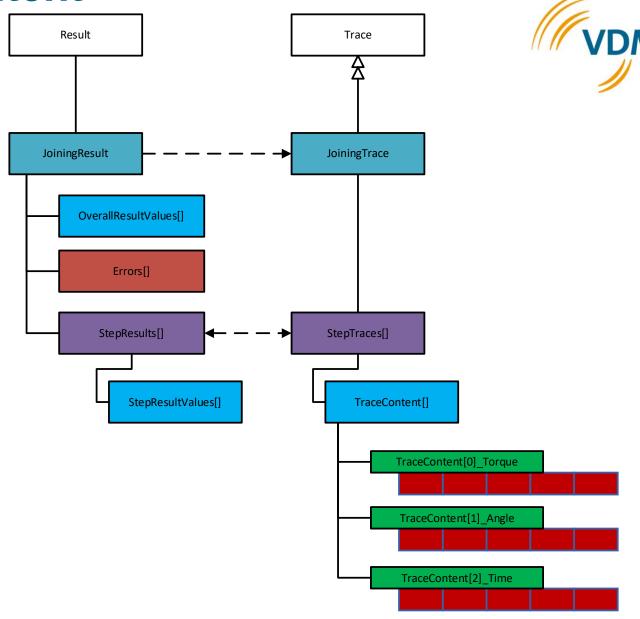






# **Single Joining Result Content**

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

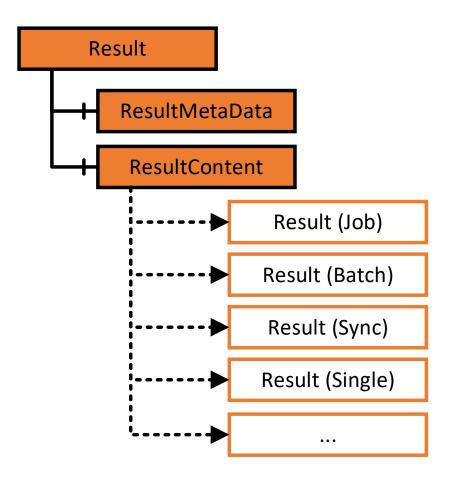


## Result Classification – Multiple Use Cases

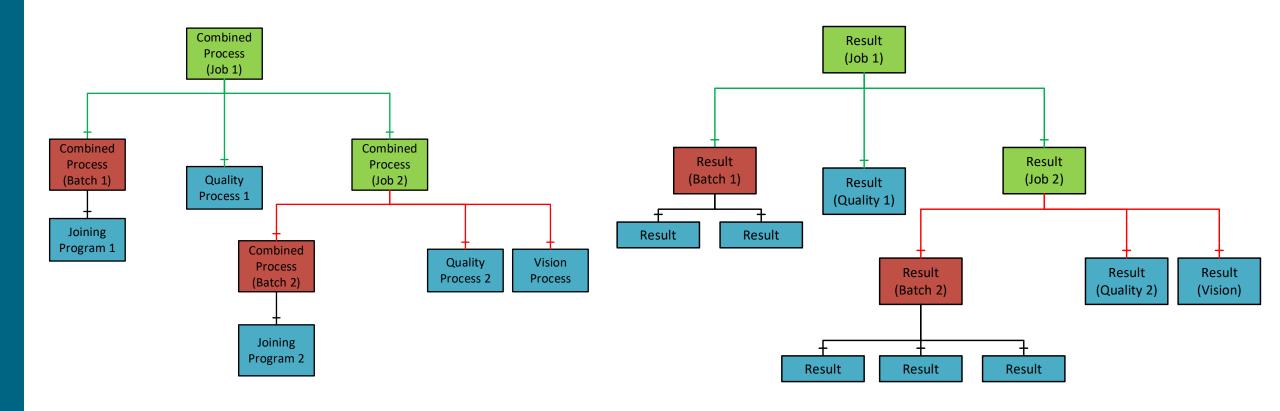
VDMA

- 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**

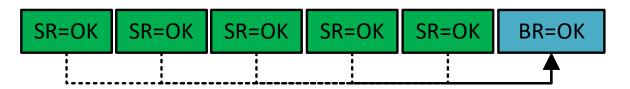
VDMA

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

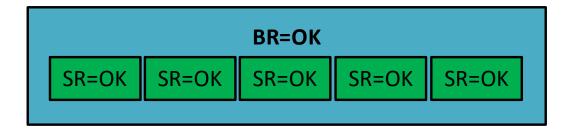
#### **Batch Result Example**

**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**

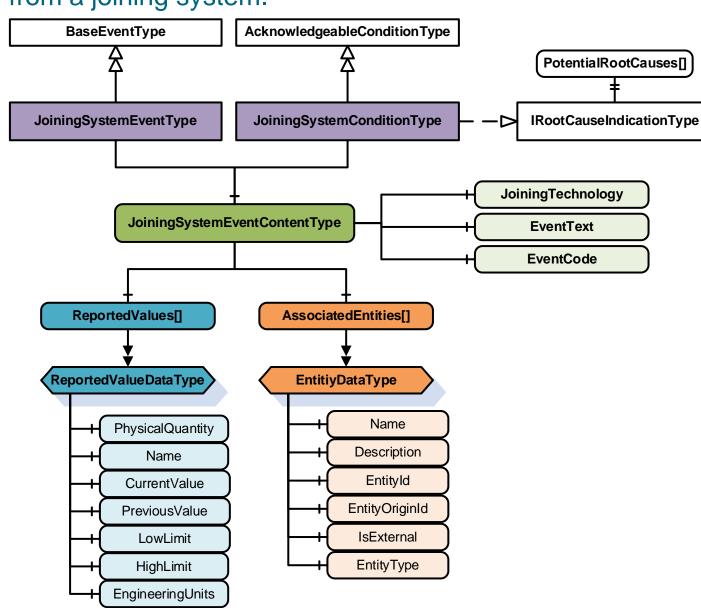
Event and Condition with standard payload from a joining system.

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

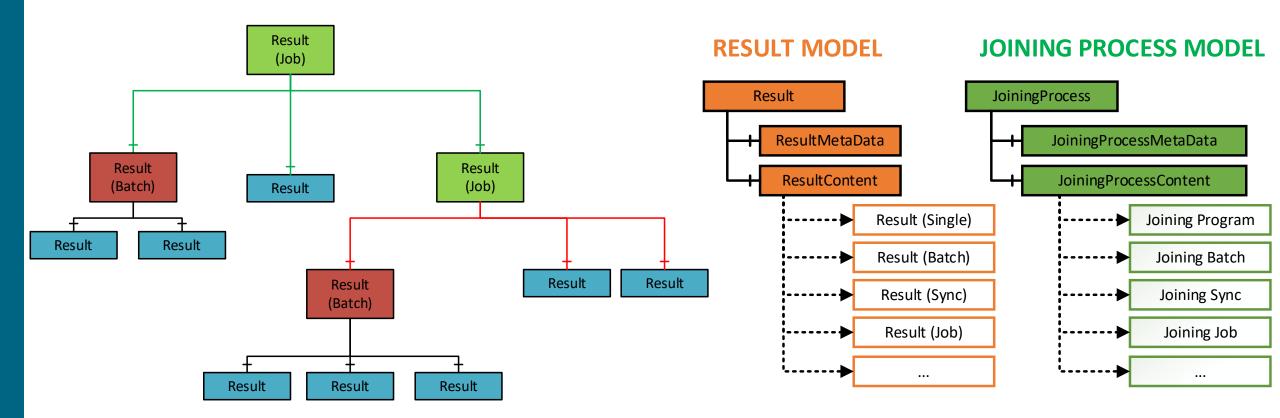
Condition Classes in	Condition Classes in		
Base Specifications	IJT Specification(s)		
OPC UA Base			
BaseConditionClass	AssetConnectedConditionClass		
ProcessConditionClass	Asset Disconnected Condition Class		
MaintenanceConditionClass	Asset Enabled Condition Class		
SystemConditionClass	Asset Disabled Condition Class		
SafetyConditionClass	Threshold Violation Condition Class		
Highly Managed Alarm Condition Class	Threshold Violation Resolved Condition Class		
TrainingConditionClass	Joining System User Logged In Condition Class		
Statistical Condition Class	Joining System User Logged Out Condition Class		
TestingConditionClass	LocationInRangeConditionClass		
Asset Management Basics	LocationOutOfRangeConditionClass		
ConnectionFailureConditionClass	AssetLocationConditionClass		
OverTemperatureConditionClass	DataValidationFailureConditionClass		
CalibrationDueConditionClass	InputValidationFailureConditonClass		
SelfTestFailureConditionClass	ConfigurationChangeConditionClass		
${\it Flash Update In Progress Condition Class}$	ErrorConditionClass		
Flash Updated Failed Condition Class	Software Condition Class		
BadConfigurationConditionClass	Hardware Condition Class		
OutOfResourcesConditionClass	CertificateConditionClass		
OutOfMemoryConditionClass	LicenseConditionClass		
Inspection Condition Class	MissingEntityConditionClass		
External Check Condition Class	Expired Entity Condition Class		
ServicingConditionClass	InvalidEntityConditionClass		
Improvement Condition Class	Incompatible Entity Condition Class		
Repair Condition Class	Accepted Entity Condition Class		
	Rejected Entity Condition Class		
	AddedEntityConditionClass		
	UpdatedEntityConditionClass		
	RemovedEntityConditionClass		
	Received Entity Condition Class		



# **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.

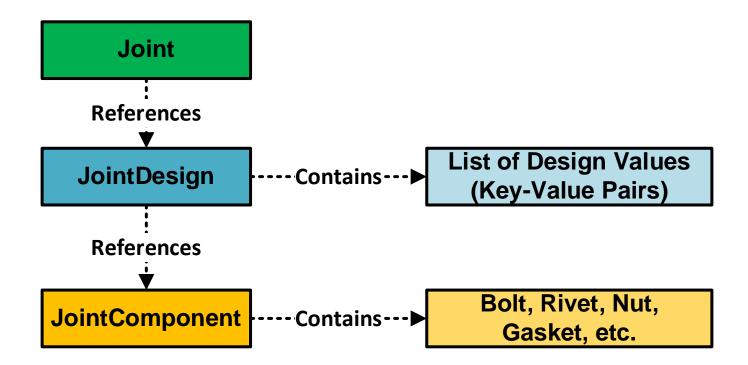




# **Joint Management**

## **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.





# Joining System Method List (Commands)

## **Joining System Method List**

Note: Result Management Methods are reused from Machinery - Result Transfer.

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		SetJoiningProcessMapping	GetJointRevisionList
GetFeedbackFileList		SelectJoiningProcess	GetJointDesignList
SetOfflineTimer		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	



## **IJT Profiles Overview**

#### **IJT Certification 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		
Result Server Facet	Basic Joining System Facet	
Asset Management Assets Server Facet		
Basic Joining System Facet		
Joining Result Server Facet	General Joining System 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
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**

- VDMA
- 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 <a href="https://umati.org/">https://umati.org/</a> for more information.
- Demonstration: <a href="https://umati.app/">https://umati.app/</a>
  - Done in automatica 2022 and 2023.



#### **Release 1 and Release 2 Differences**

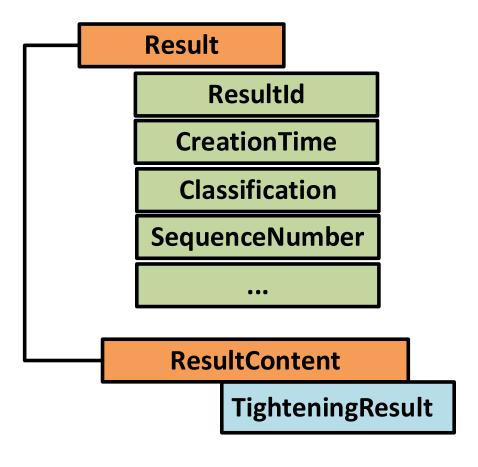
#### **Overview**



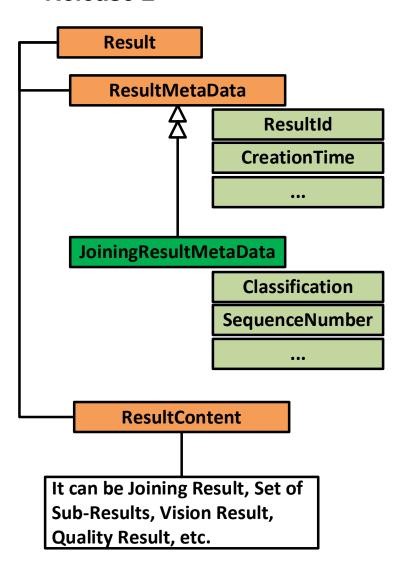
- Release 1 was only for Tightening Domain.
  - OPC 40451-1 UA CS for Tightening Systems 1.00.0.
- During Release 2 work, most of the use cases are generalized for various joining systems.
   Hence, the models were extended and moved to a new base specification.
  - OPC 40450-1 UA CS for Joining Systems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

Release 1

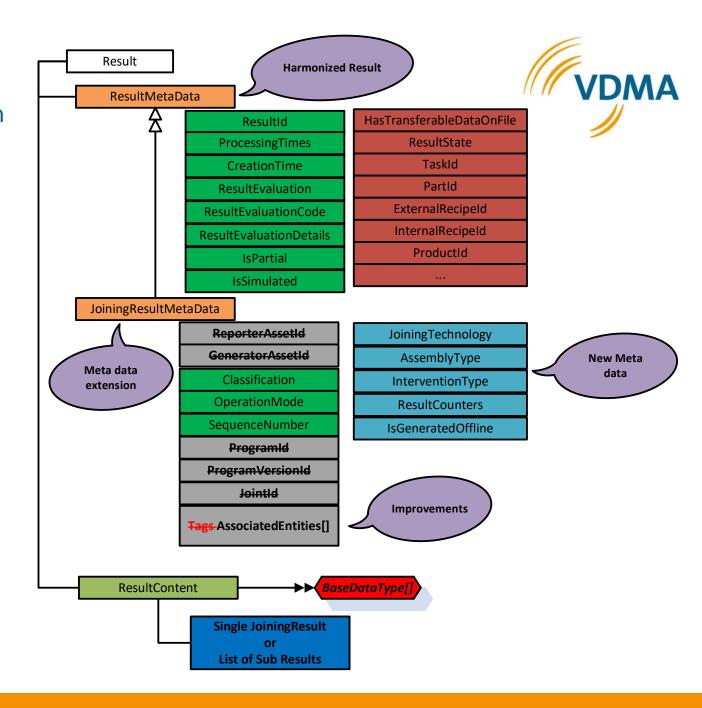


#### Release 2



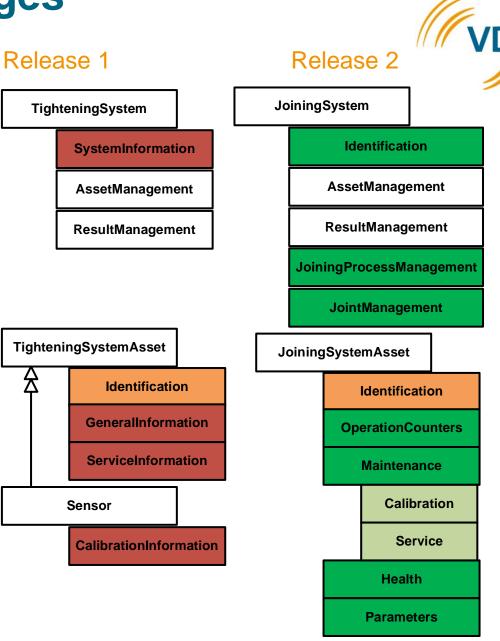
### **Result Data Changes**

- 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.



## System, Asset and Event Changes

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

- VDMA
- 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/IJT/Base/v100/docs/
  - https://reference.opcfoundation.org/IJT/Tightening/v200/docs/
- 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: <a href="https://vdma.org/der-verband">https://vdma.org/der-verband</a>
    - 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

#### **OPC UA Overview and Benefits**

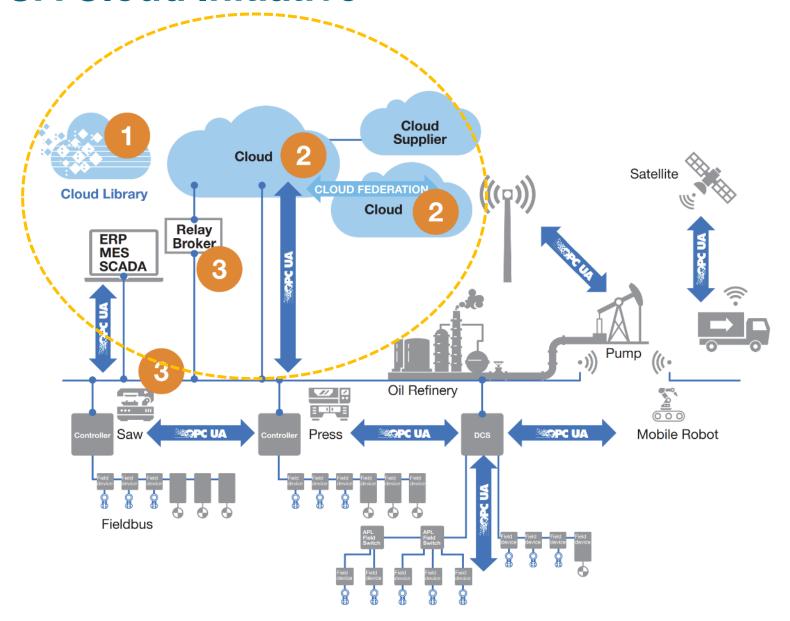


Industry 4.0 main contender for industrial interoperability standard

#### 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

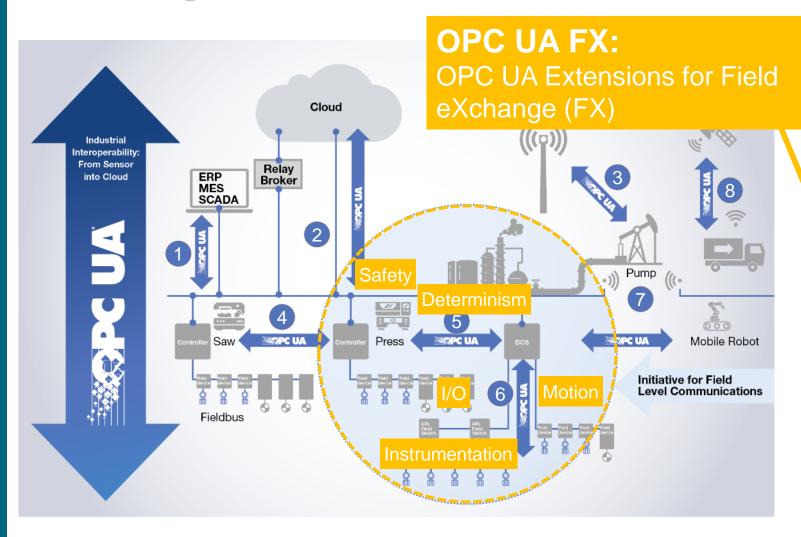
#### **UA Cloud Initiative**



- 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** 

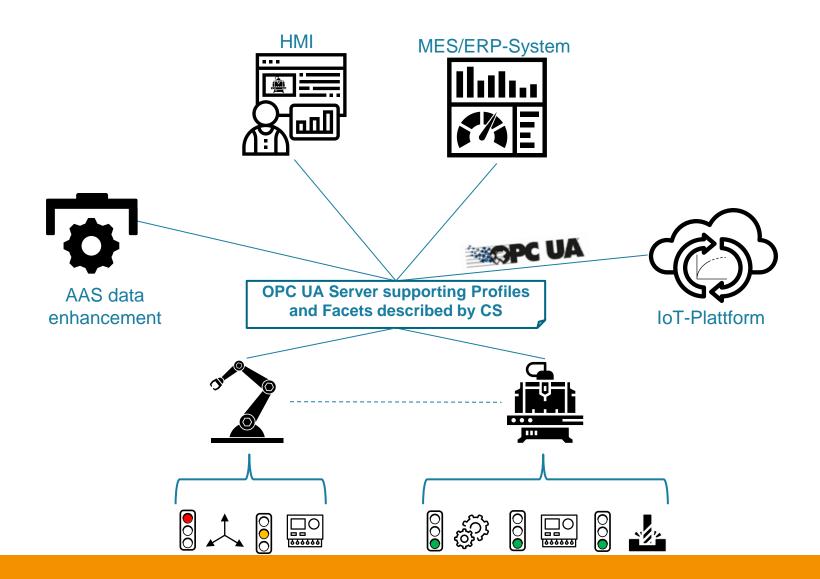
# OPC UA for Field eXchange (FX): Extending OPC UA to the field incl. Determinism, Safety & Motion



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

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





#### **Contact**



**Bernd Heitzmann** 

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

Phone E-Mail

in

+49 (0) 69 6603 1530

Bernd.Heitzmann@vdma.org



**Mohit Agarwal** 

**Principal Engineer** 

**Atlas Copco Industrial Technique, Sweden** 

**Editor of Industrial Joining Technologies Group** 

Phone E-Mail

+46 (0) 766 96 19 90

mohit.agarwal@atlascopco.com



## **Questions**

### Few Example Questions...

- Do we have a reference implementation?
  - Yes, refer to the link to the simulator in the presentation.
- When is the implementation planned for the Devices?
  - The respective device supplier could provide the plan.

•

