Aalto University
School of Electrical
Engineering

OPC Unified Architecture Client-Server and PubSub

Information systems in industry ELEC-E8113

Start at 12.15!

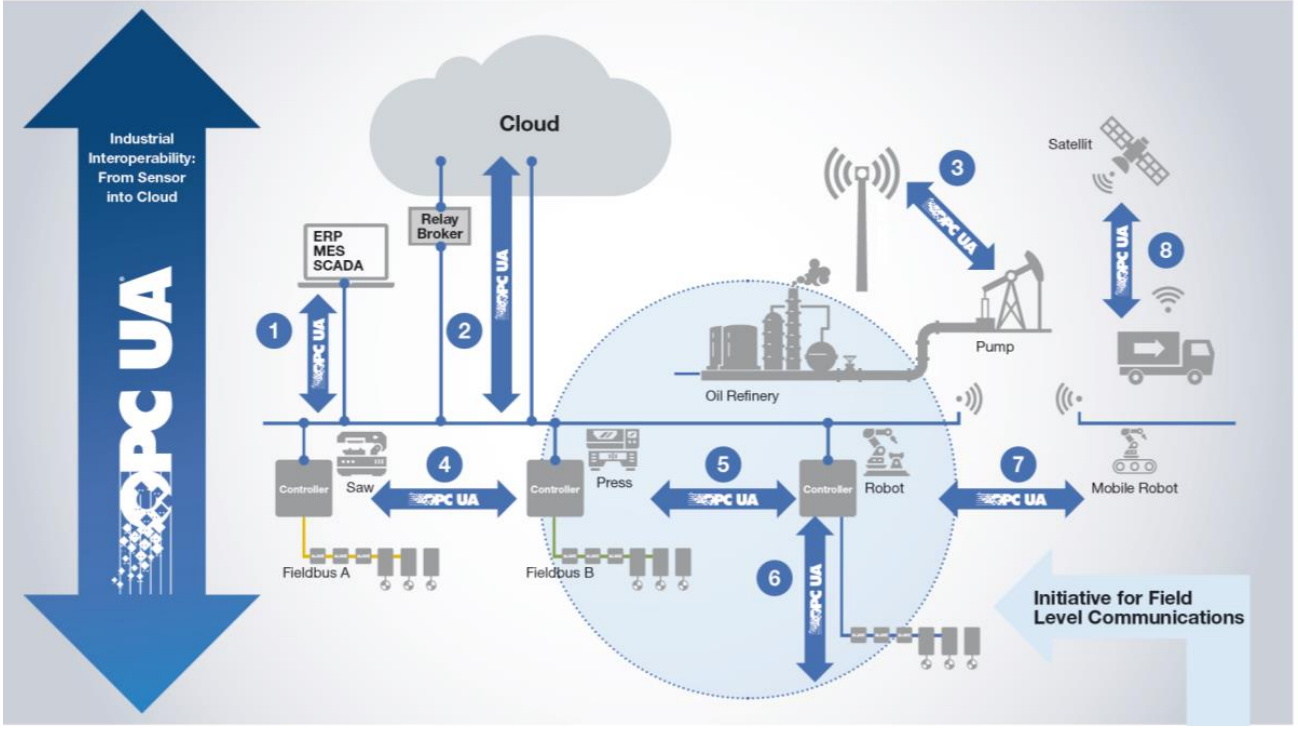
Contents

- **Services and other basic features**
- **Application development**
- **PubSub**

Rationale of the lecture: OPC UA is a good example of a developing communication technology. OPC UA has changed due to developments in requirements and other communication technologies.

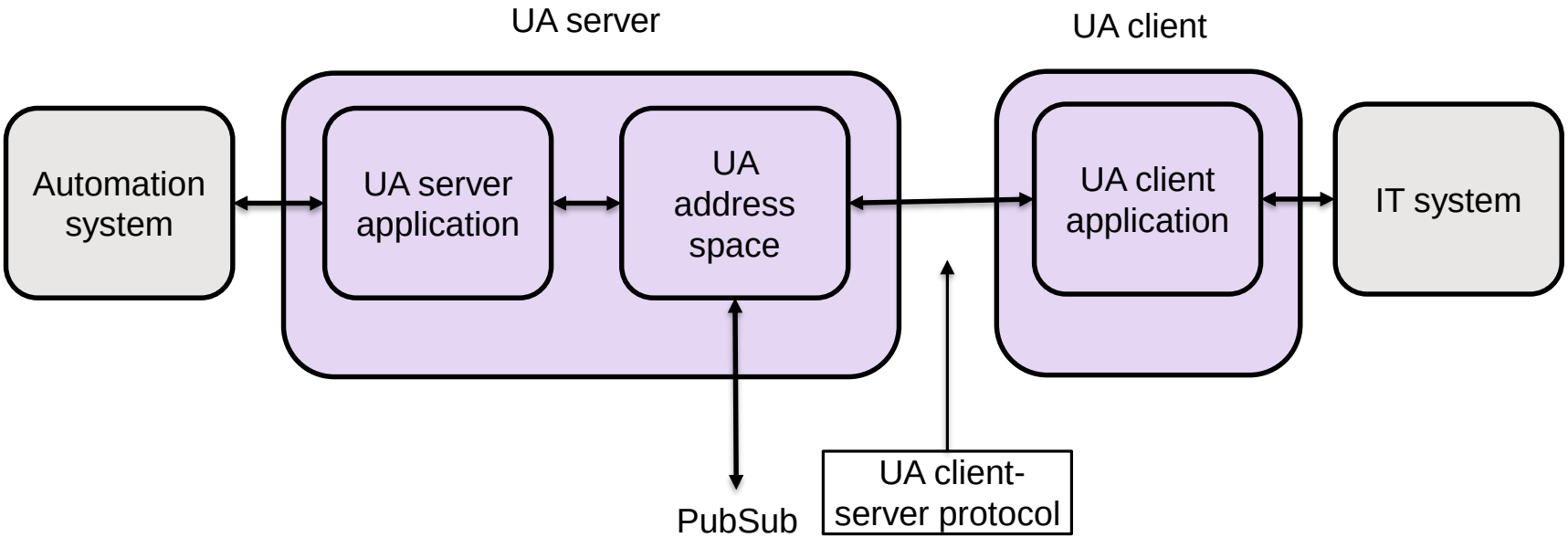
.

Situation



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

Focused view

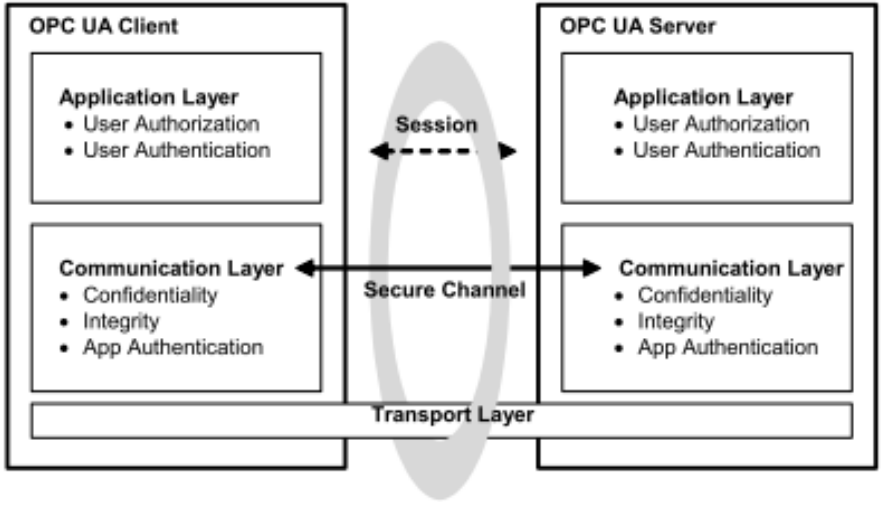


Service sets

Service set	Functionality
Discovery	Finding servers
Secure channel	Opening and closing a secure communication channel
Session	Creating and closing a communication session
Node management	Adding and deleting Nodes and References
View	Browsing the address space
Query	Querying the address space
Attribute	Reading and writing Attributes of Nodes
Method	Calling Methods
Monitored item	Creating, modifying and deleting Monitored Items
Subscription	Creating, modifying and deleting Subscriptions

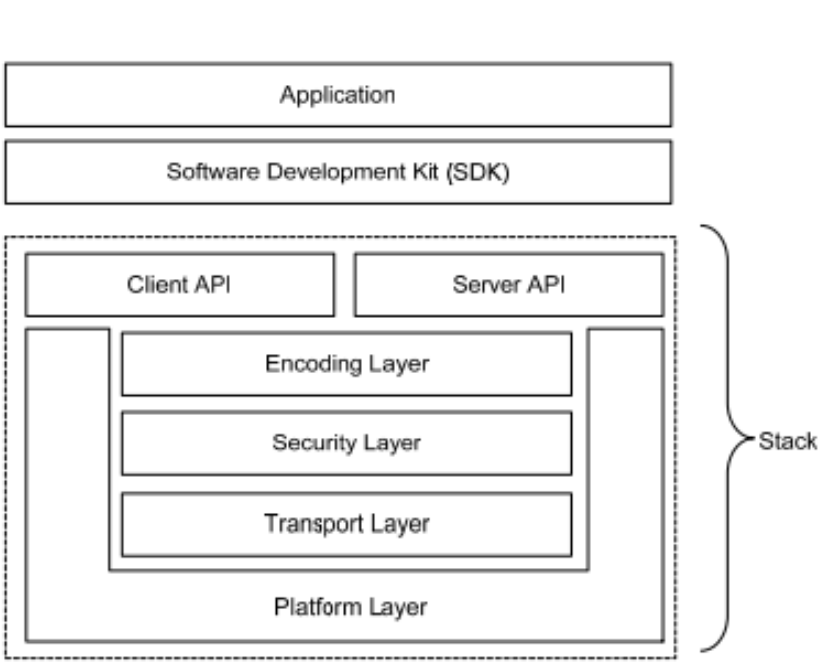
Security

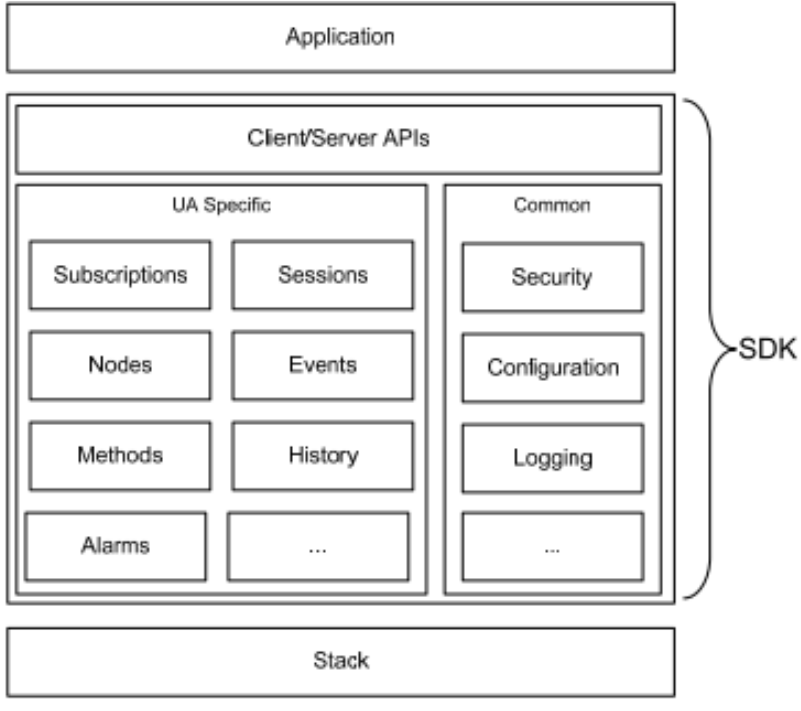
- Application authentication when a client start communication with a server
- User authentication and authorization when a client start a session with a server
- Message confidentiality and integrity when communicating between a client and a server
- There can be a firewall between them



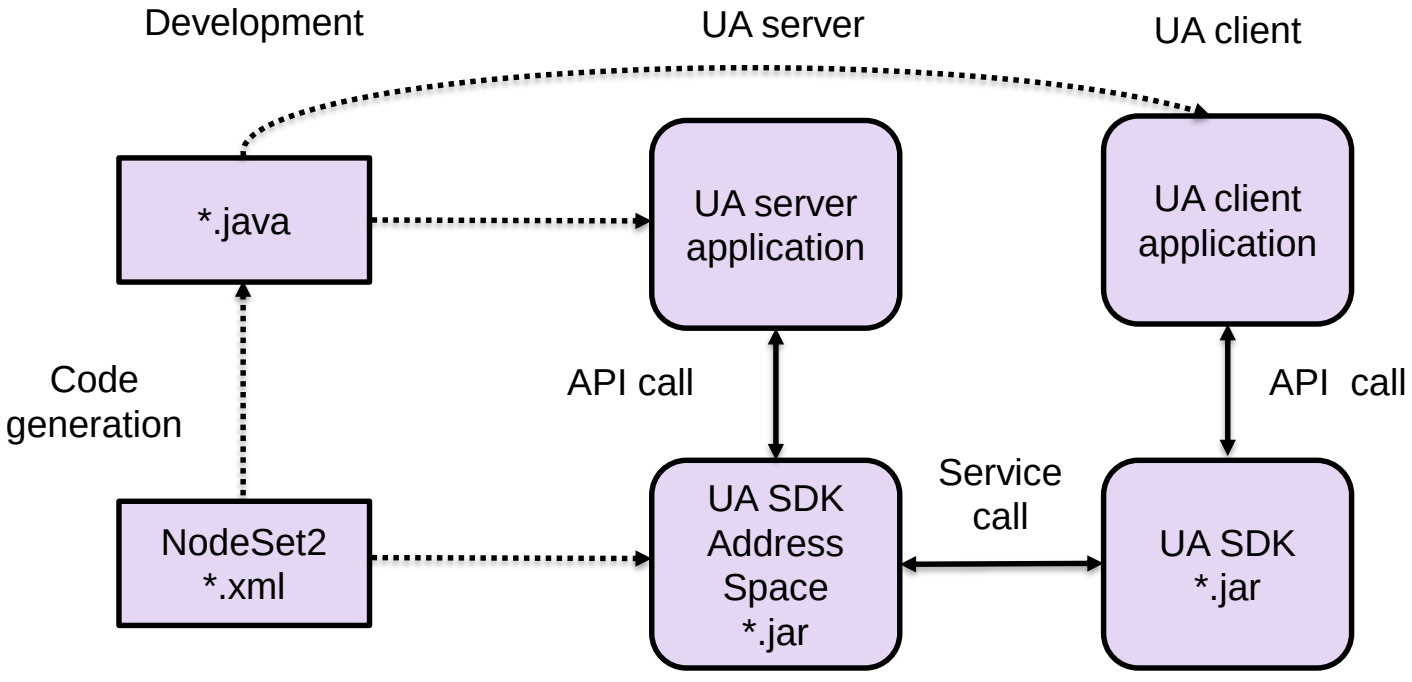
OPC UA security architecture

Application architecture: Stack and SDK



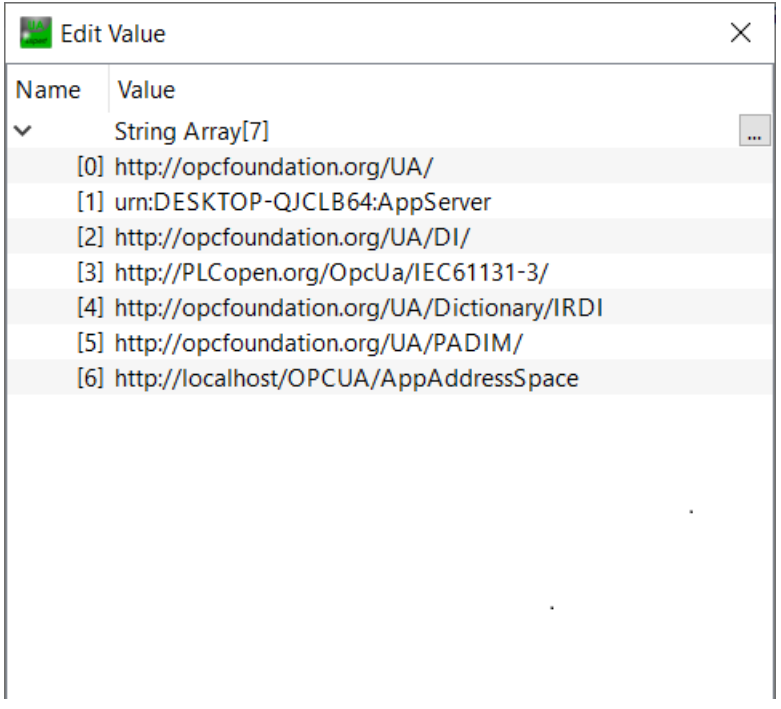


Development with Information Models



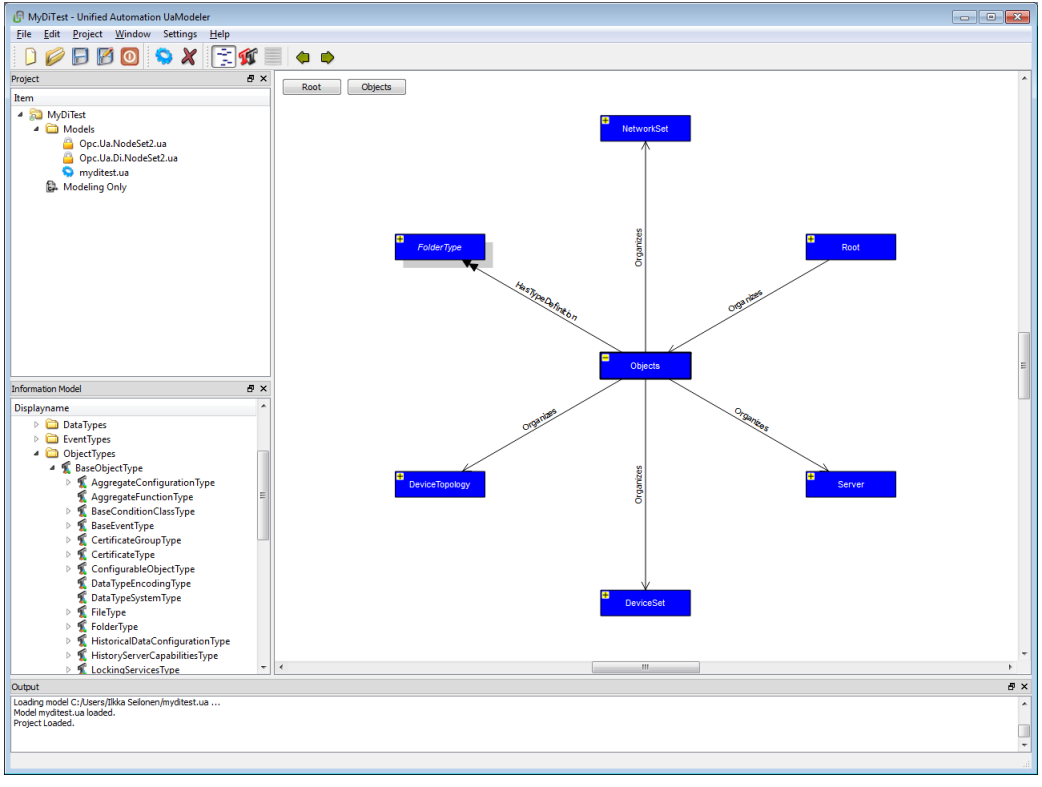
Namespaces

- **Namespaces organize nodes at an UA server**
- **NodeId specify (1) namespace and (2) node**
- **You have separate namespaces for**
  - Build-in information models
  - Other information models
  - Your application



UaModeler and code generation

- Edit graphically your own information models
- Utilize standard and companion information models as a basis
- Export to NodeSet2.xml format
- Generate source code for ANSI C, C++, C#
- Code is generated for types
- Example: For C++ 7 *.h and 5 *.cpp files are generated for one very simple ObjectType

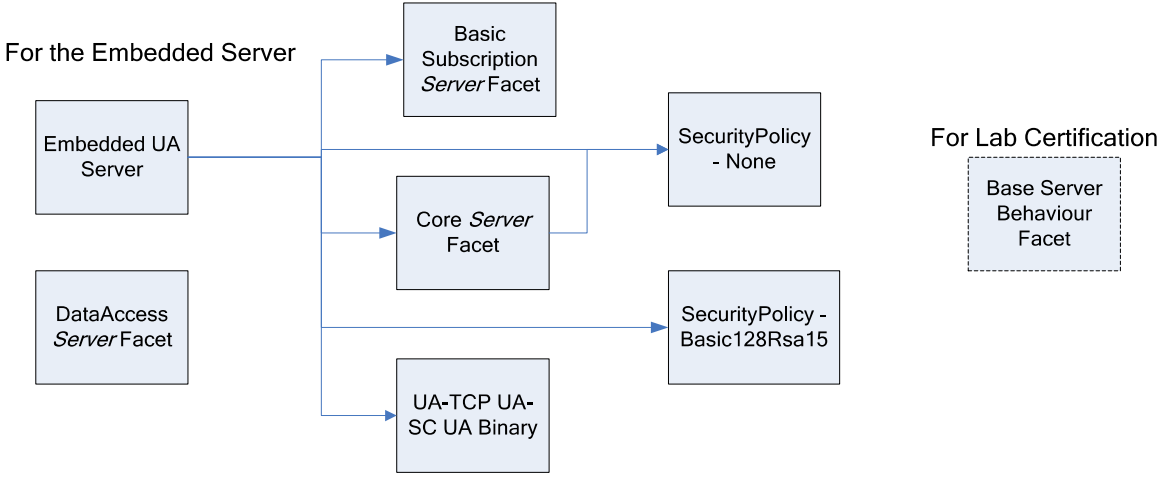


Development tools

- Stacks enable OPC UA communication between clients and servers
- SDKs extends stacks with functionality for developing client and server applications
- Both stacks and SDKs are available for several programming languages and operating systems (e.g. C/C++, C#, Java)
- Test clients are general purpose clients to access any OPC UA server. Test servers are intended for testing OPC UA client applications
- Gateways are wrappers for OPC Classic servers
- Modelers enable editing information models and generating code from them.

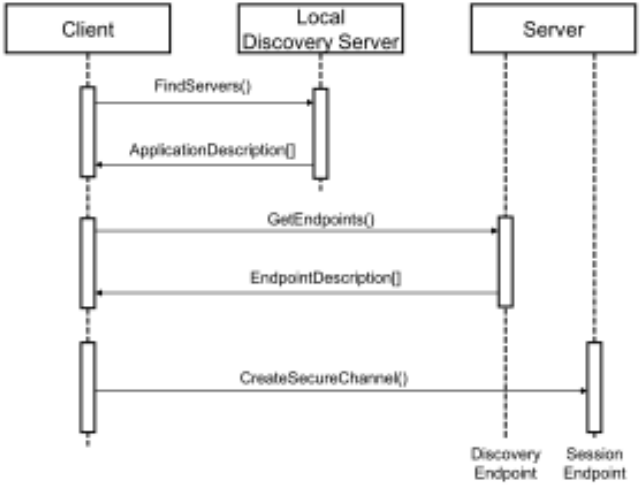
Profiles

- **Server profiles define which OPC UA features the server has. All servers do not need to has all features**
- **Profiles are defined with a concept of Facets, which are tested features**



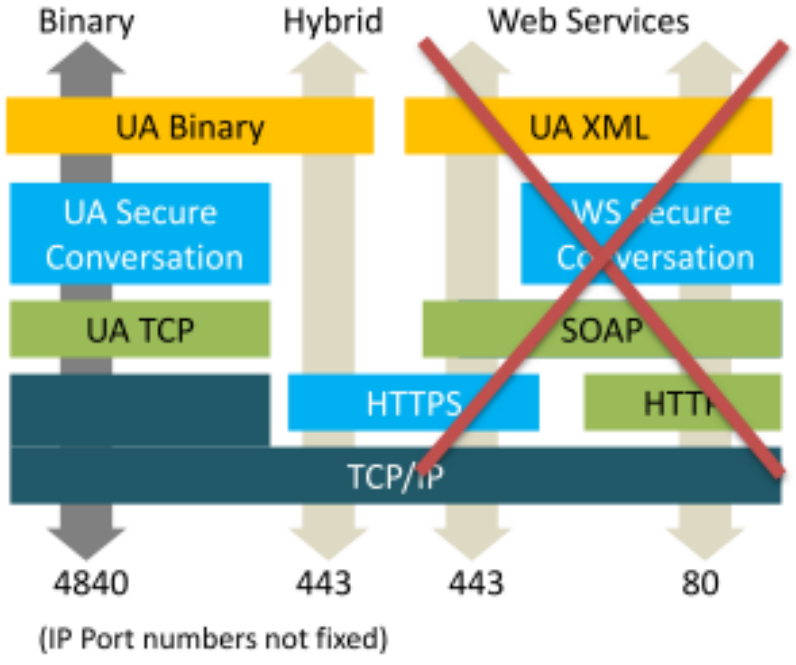
Discovery

- OPC UA clients need Endpoints (address and security) to create a secure channel to a server
- Clients can find servers through either local or global Discovery Servers
- OPC UA servers can register to discovery servers



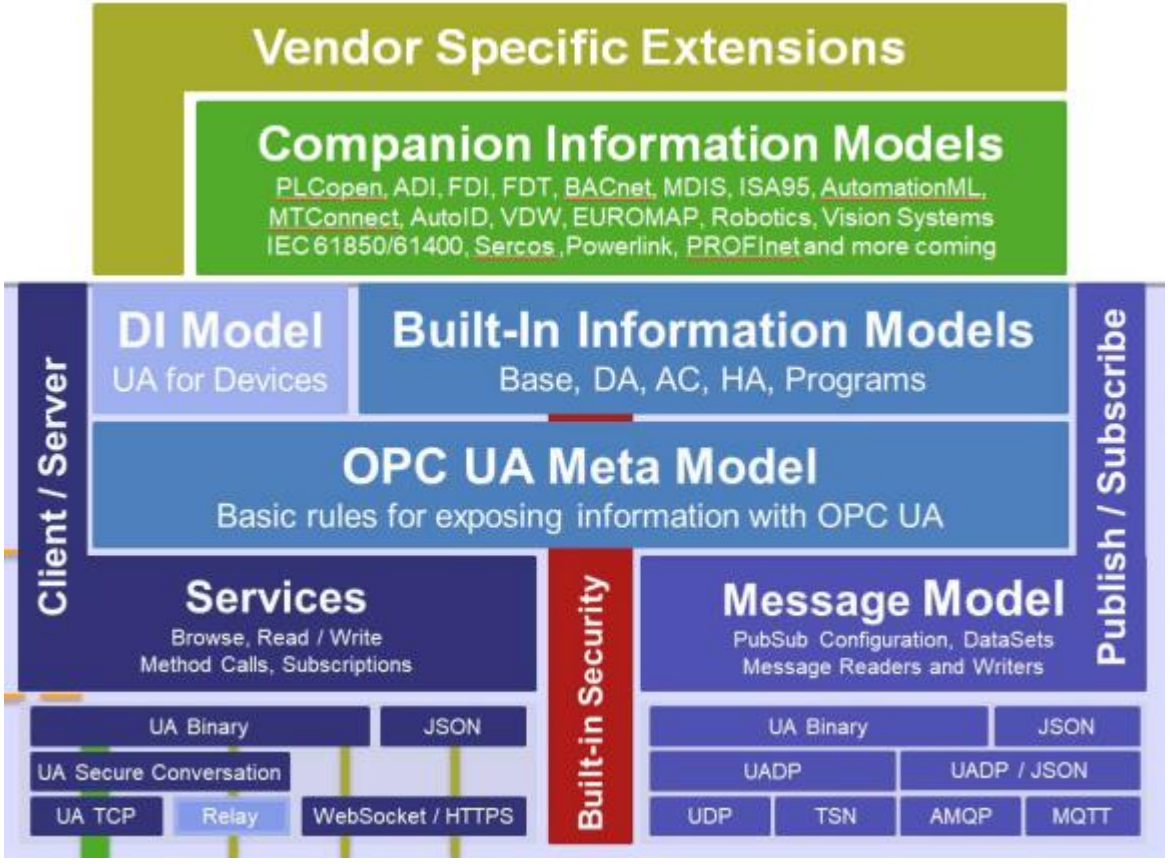
Mappings

- Mappings define message encodings, security and transport protocols to be used in service requests and responses
- Initially two alternatives were provided
- UA * mappings are specific to OPC UA
- WS Secure Conversation and SOAP have been removed



New model of OPC UA

- Publish / subscribe model of communication
- JSON as a new message encoding
- New transport protocols
- OPC UA for Devices (DI) as a part of the core specification
- Larger set of companion specifications



PubSub (Publish / Subscribe)

- Two models: with or without a broker, both following the same model
- Actors: Publisher, Subscriber, MOM (Message-Oriented Middleware)
- Data: DataSet, Messages
- Functions: DataSetReader, DataSetWriter

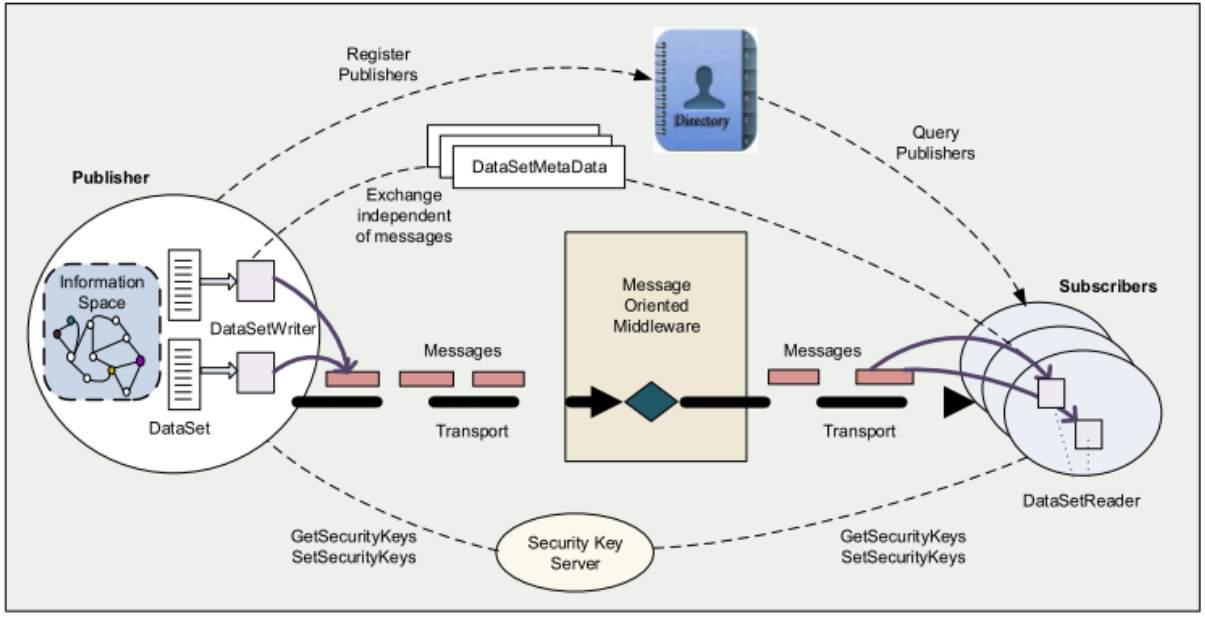


Figure 2 – Publisher and Subscriber entities

Configuration model

- Connection defined a communication channel for sending and receiving messages
- Writer/ReaderGroup for each NetworkMessage to write/read
- DataSetWriter/Reader for each DataSet to write/read

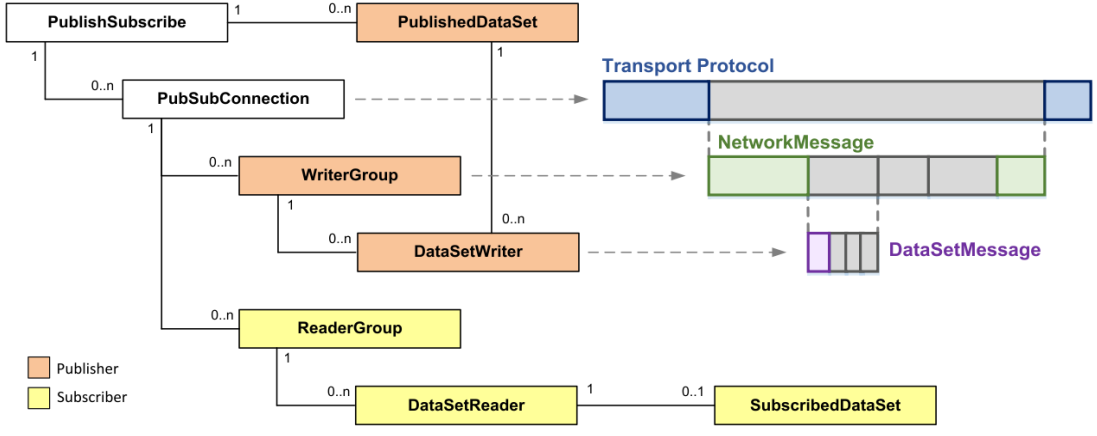


Figure 17 – PubSub Component Overview

Broker-based PubSub

- In Broker-based PubSub the MOM is a broker, i.e. a message queue server
- Both the Publishers and Subscribers are clients of the Broker
- All actors do not need have anything to do with OPC UA
- JSON is the typically intended message encoding
- Bindings to AMQP and MQTT as Brokers
- Data collection to cloud storages is a the typical intended use case

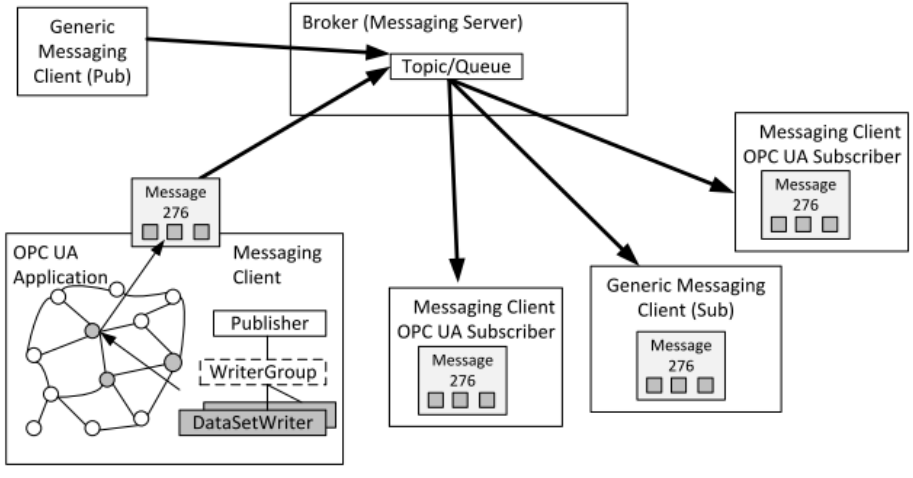


Figure 16 – Broker Overview

Broker-less PubSub

- In Broker-less PubSub the MOM is a network
- Connections are UDP uni/multicast
- All actors do not need to have anything to do with OPC UA
- UADP is the message encoding
- UDP defined as transport protocol, TSN being worked on
- (Fast) communication between OPC UA servers is the typical intended use case
- Deterministic communication delays with TSN

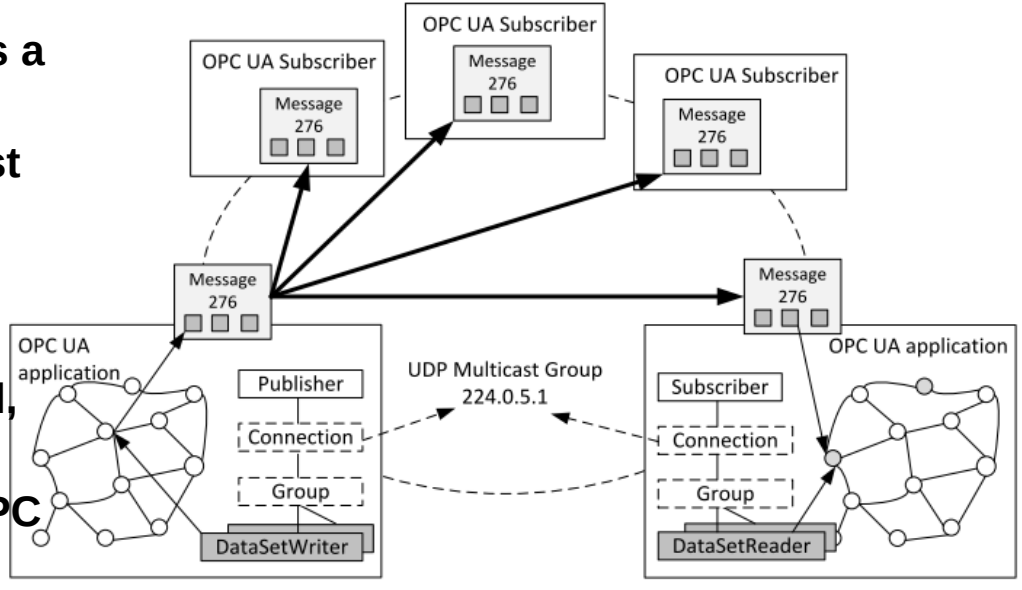


Figure 14 – UDP Multicast Overview

Broker-less PubSub and TSN

- TSN is an IEEE standard being developed for real-time communication over Ethernet
- TSN binding would enable real-time broker-less PubSub
- TSN Profile for Industrial Automation would be needed
- TSN would require more complex configuration of communication than UDP
- Extending PubSub definition is needed
- Currently there are several different Industrial Ethernet technologies with similar capabilities