OPC DAY
FINLAND 2021
16.11.2021

OPC UA PubSub Explained

Jouni Aro, CTO
Prosys OPC

FINNISH SOCIETY OF AUTOMATION
SUOMEN AUTOMAATIOSEURA RY

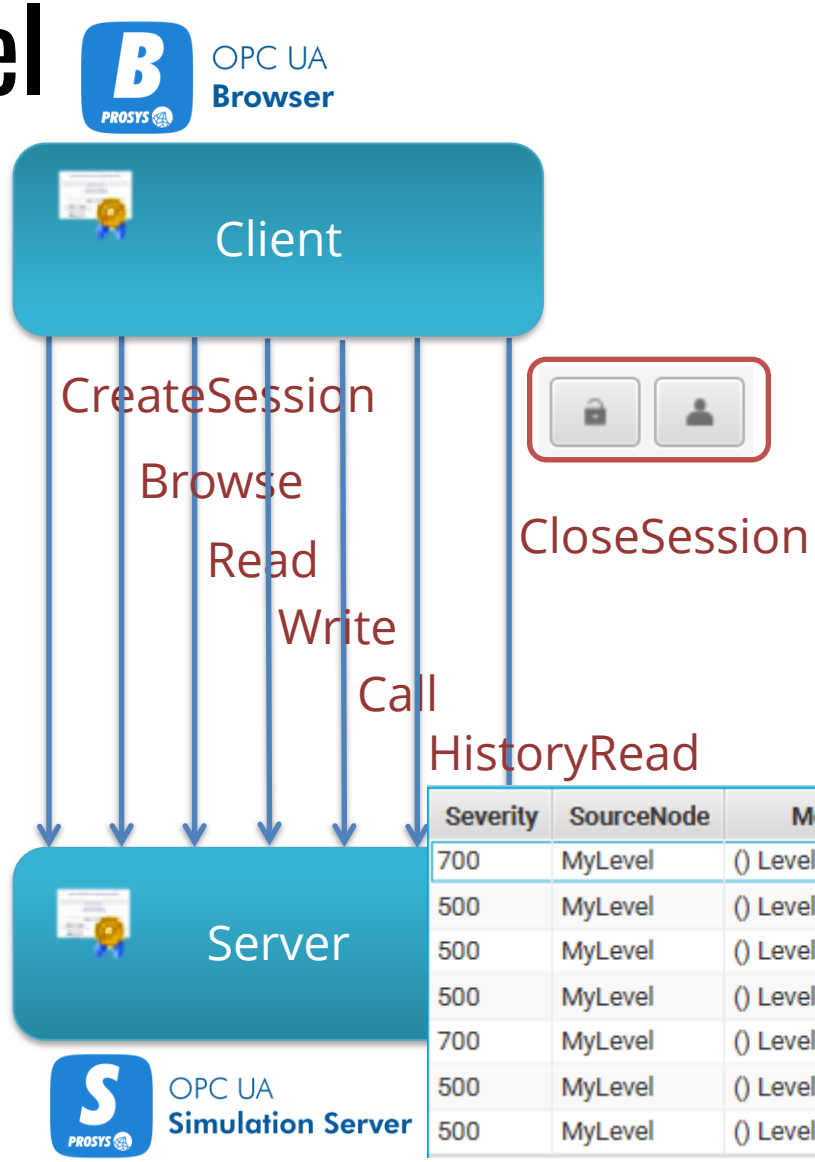
SPONSORS:

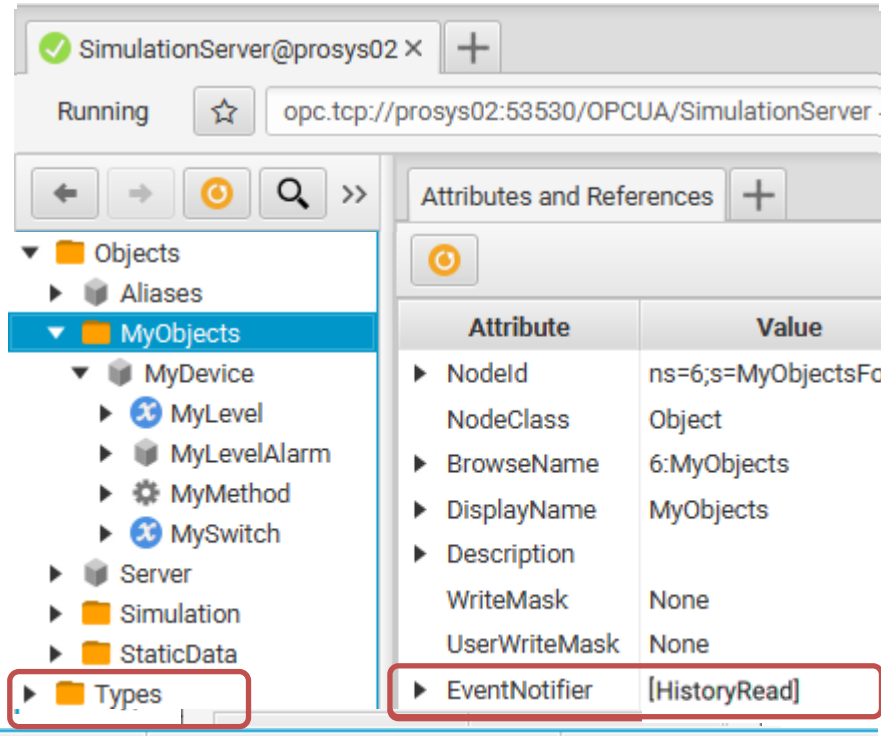
BECKHOFF NAPCON Nortal OPC FOUNDATION PROSYS OPC Life Is On Schneider Electric Semantum Valmet FORWARD Wapice

Contents

Client/Server Model

1. Connect a client to a server
   - Create a Session
   - ➢ Security Context
2. Browse AddressSpace
   - To find what is available
3. Read
   a. Variable Values
   b. Meta data
4. Write
   a. Variable Values
   b. (Meta data)
5. Call Methods
6. Read History
   a. Variables
   b. Events
7. Disconnect when done
   - Close the Session



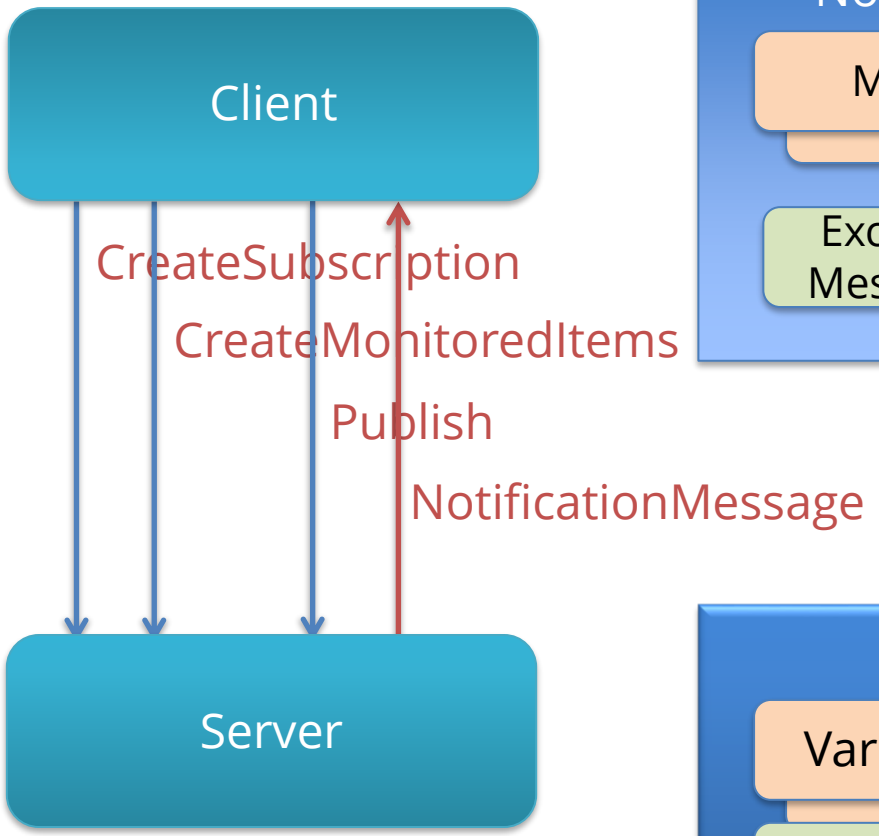


Severity	SourceNode	Message	Time	EventType
700	MyLevel	() Level exceeded	11/03/21 15:20:27.5260000 GMT	ExclusiveLevelAlarmType
500	MyLevel	() Level exceeded	11/03/21 15:20:46.5250000 GMT	ExclusiveLevelAlarmType
500	MyLevel	() Level exceeded	11/03/21 15:21:06.5380000 GMT	ExclusiveLevelAlarmType
500	MyLevel	() Level exceeded	11/03/21 15:21:47.5250000 GMT	ExclusiveLevelAlarmType
700	MyLevel	() Level exceeded	11/03/21 15:22:07.5250000 GMT	ExclusiveLevelAlarmType
500	MyLevel	() Level exceeded	11/03/21 15:22:26.5250000 GMT	ExclusiveLevelAlarmType
500	MyLevel	() Level exceeded	11/03/21 15:22:46.5260000 GMT	ExclusiveLevelAlarmType

Status: Success

Client/Server Subscriptions

1. Create a **Subscription**
2. Create **MonitoredItems**
   a. Variables
   b. EventNotifiers
3. Call **Publish** to receive
4. **NotificationMessages** with
   a. Data Changes
   b. Events



NotificationMessage

MyLevel = 90.7

ExclusiveLevelAlarmType:
Message="Level exceeded"

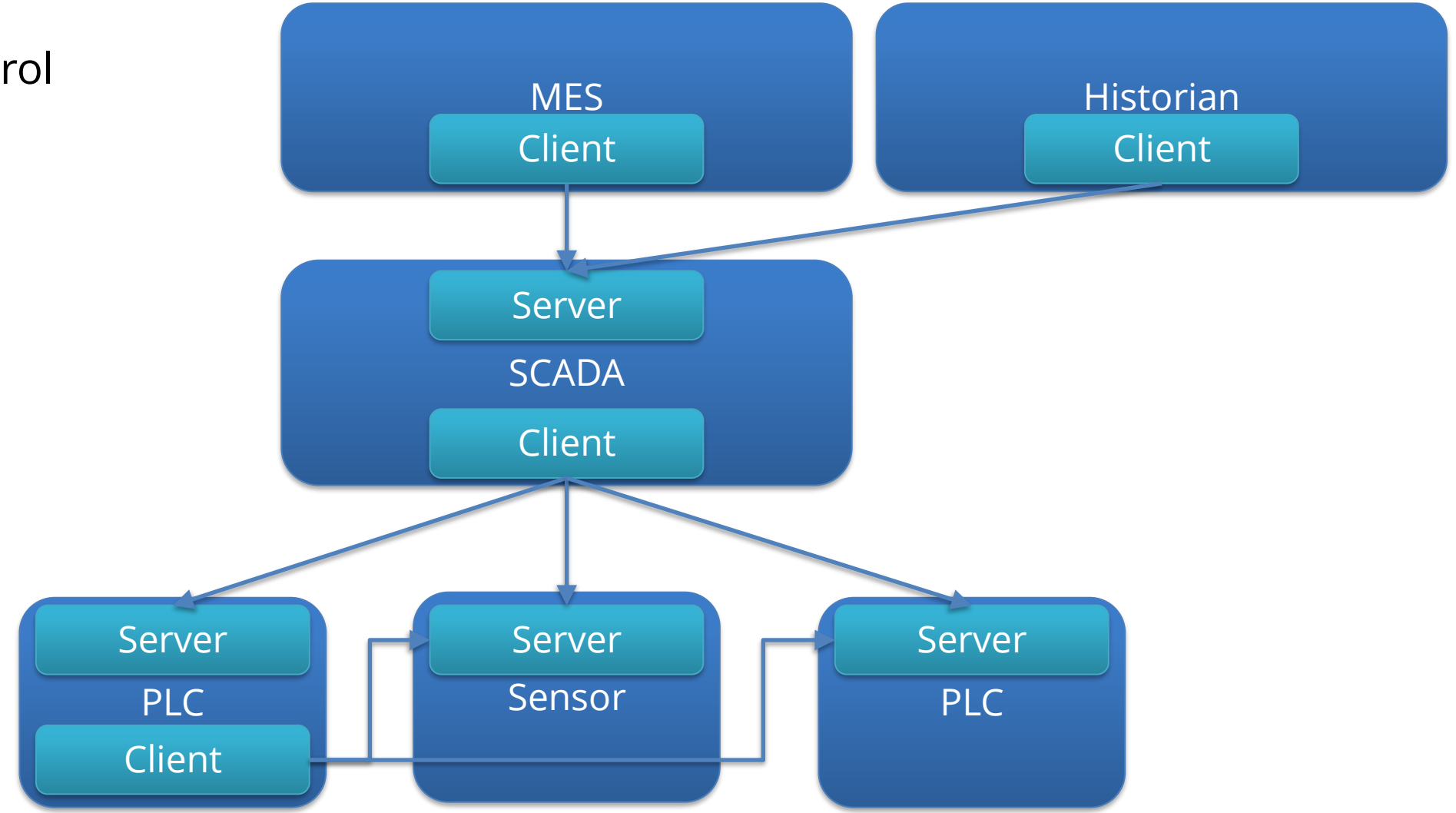
Subscription

Variable: MyLevel

EventNotifier: MyDevice

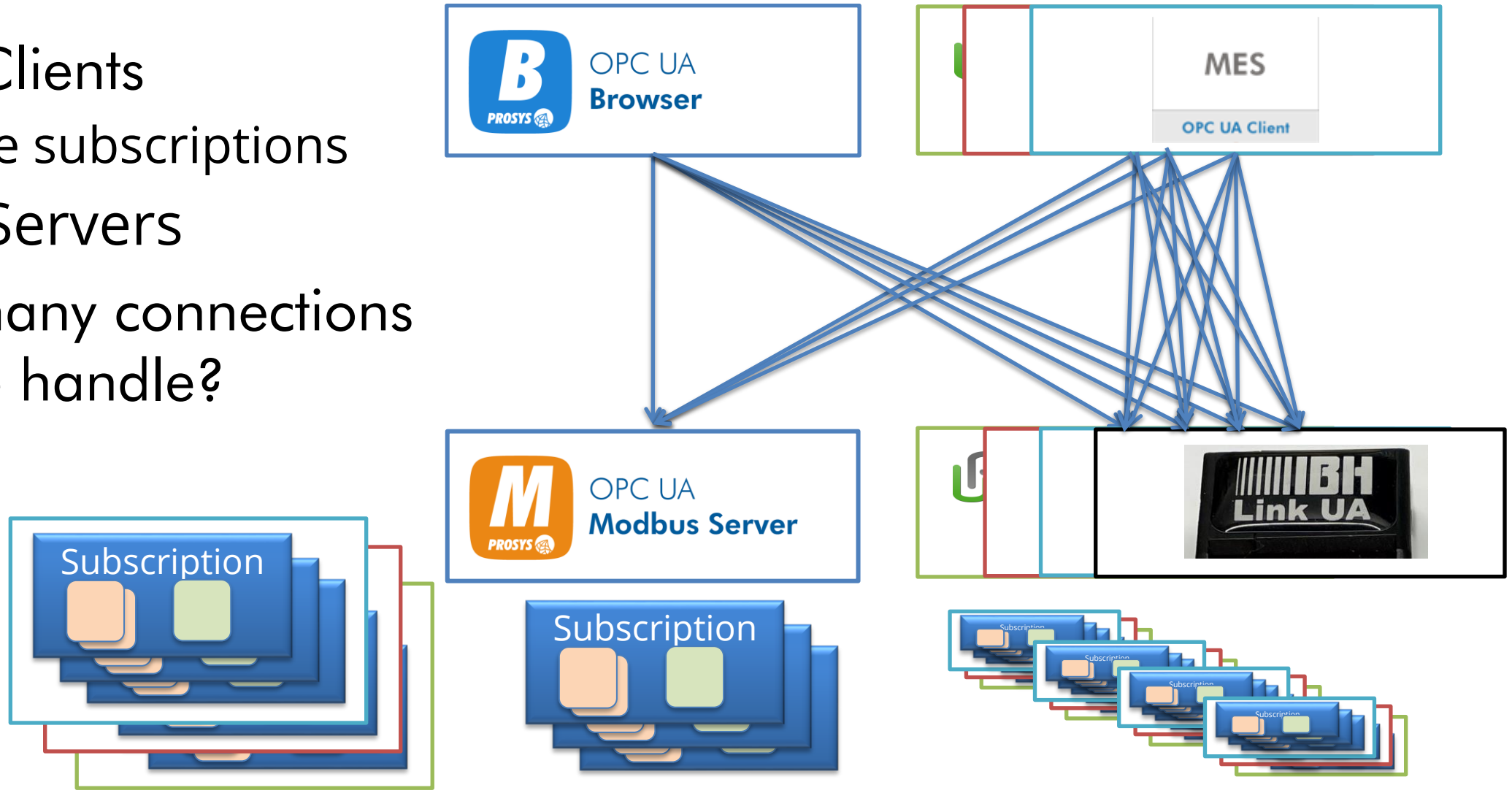
Client/Server Main Scenario

- Supervisory Control
- Data Acquisition
- \+ MES
- \+ Historian
- \+ Even PLC-to-PLC!



Client/Server Scaling

- More Clients
  - More subscriptions
- More Servers
- How many connections can we handle?



Client/Server Pros & Cons

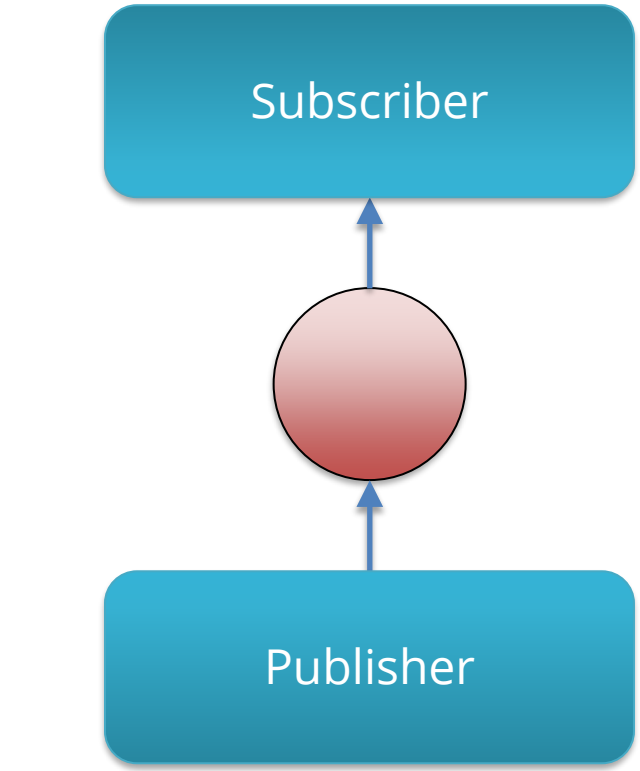
- Session-based
  - \+ Security context
  - – Requires a steady connection –> Suffers from network interruptions
  - – Requires resources per connection -> Does not scale
- Client-specific Subscriptions
  - \+ Fully customized per client
  - – Requires resources per connection -> Does not scale
- Synchronous services
  - \+ Read, HistoryRead
  - \+ Write, Method Call
- Information Models
  - \+ Data and Metadata (Types) Discovery via Address Space
- Security
  - \+ Flexible for fine-grained ruling per application and user

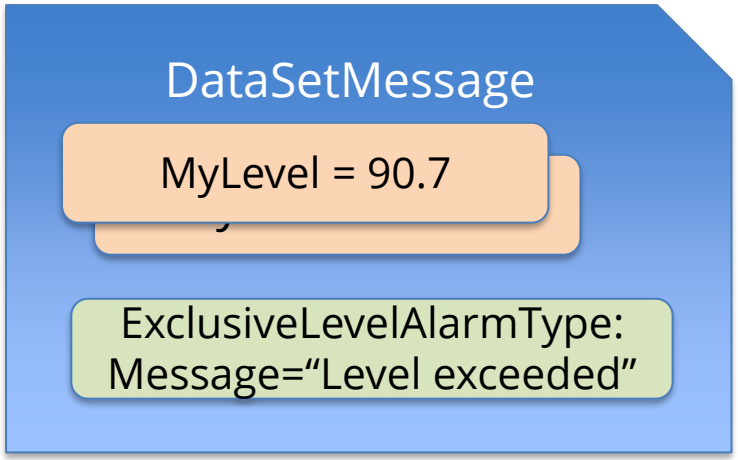
Contents

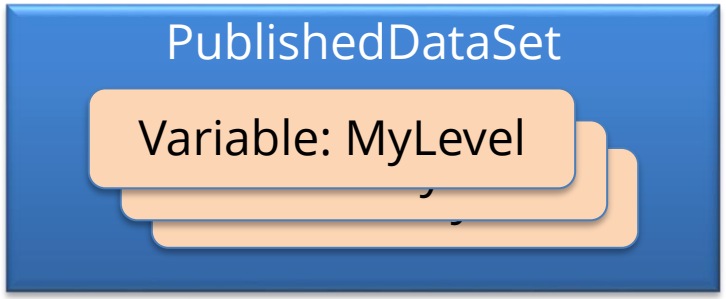
- OPC UA Client/Server
  - Model
  - Subscriptions
  - Scenarios
  - Scaling
  - Pros & Cons
- OPC UA Publisher/Subscriber
  - Model
  - Scaling
  - Networks
  - Pros & Cons
- PubSub Scenarios
  - Smart Manufacturing
  - Improved Client/Server Applications
  - Synchronized Servers
  - Edge-to-Cloud
- PubSub Demos
  - #1 Azure
  - #2 Amazon AWS
  - #3 “Global Production Line”
- Conclusions

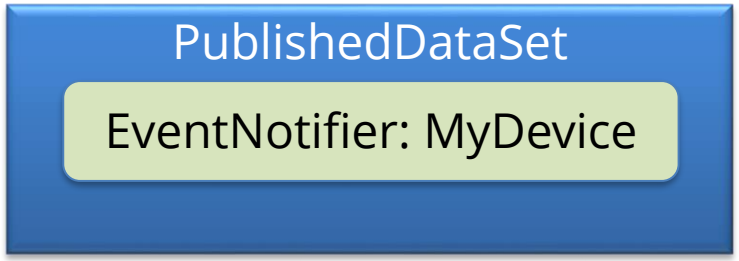
Publisher/Subscriber Model

1. Create **DataSets** with
   a. Variables
   b. EventNotifiers
2. **Publish DataSetMessages** to Network with
   a. Data Changes
   b. Events
3. **Subscriber** can filter what it needs

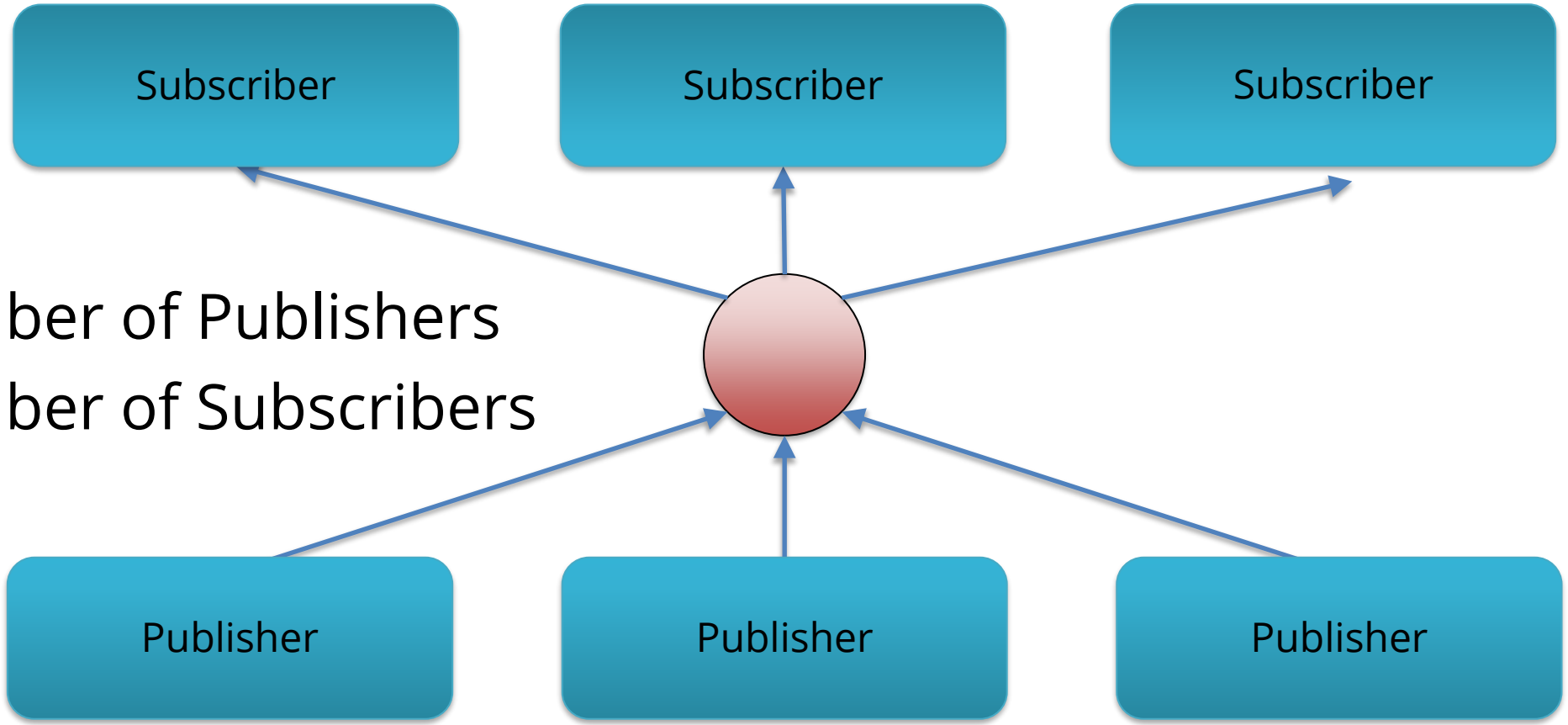








Publisher/Subscriber Scaling

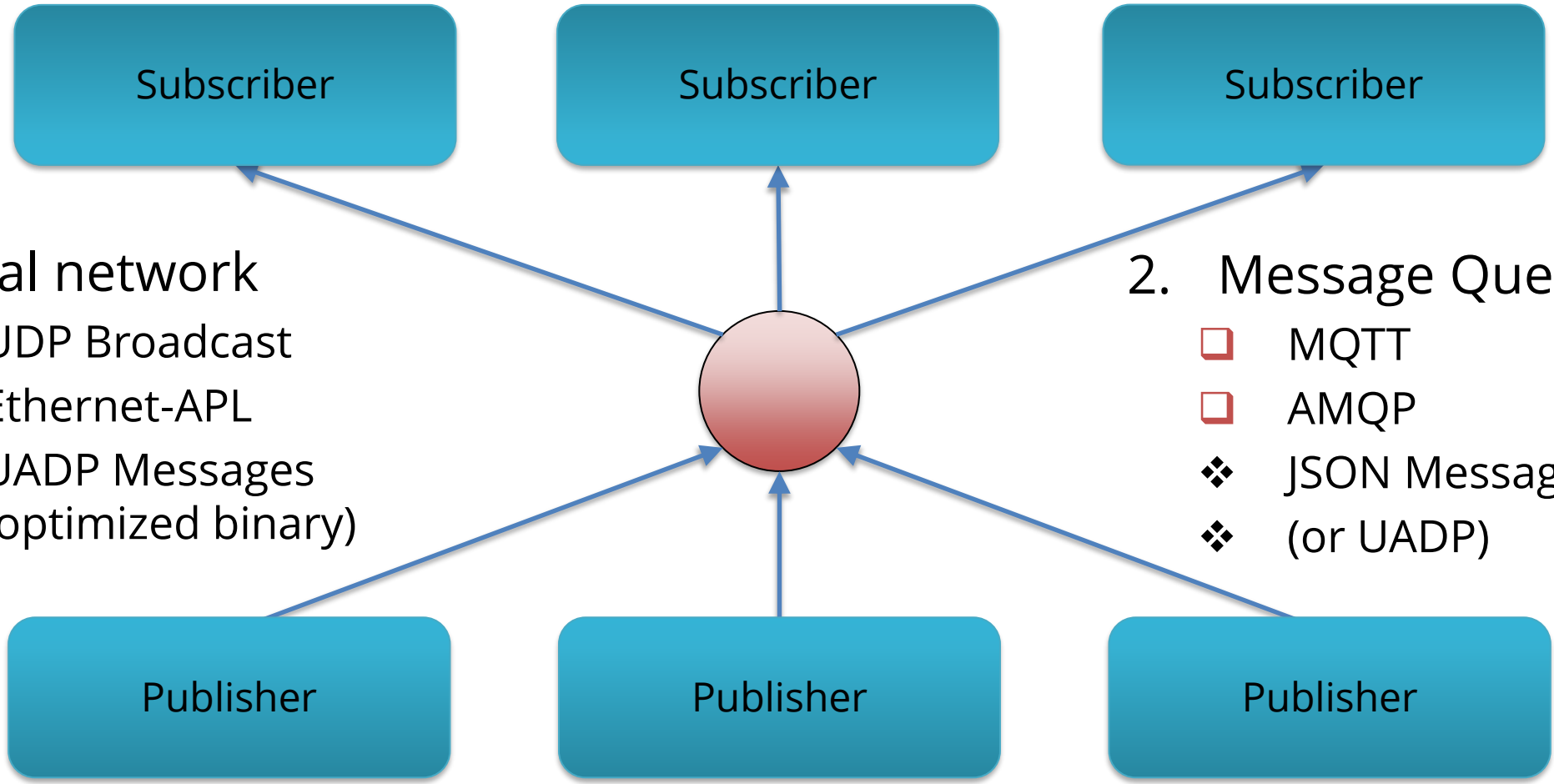


1. Any number of Publishers
2. Any number of Subscribers

➢ Highly Scalable

Verified: 2 Publishers, 31 Subscribers,
40000 variables = 3.5 MB/s (by both publishers)

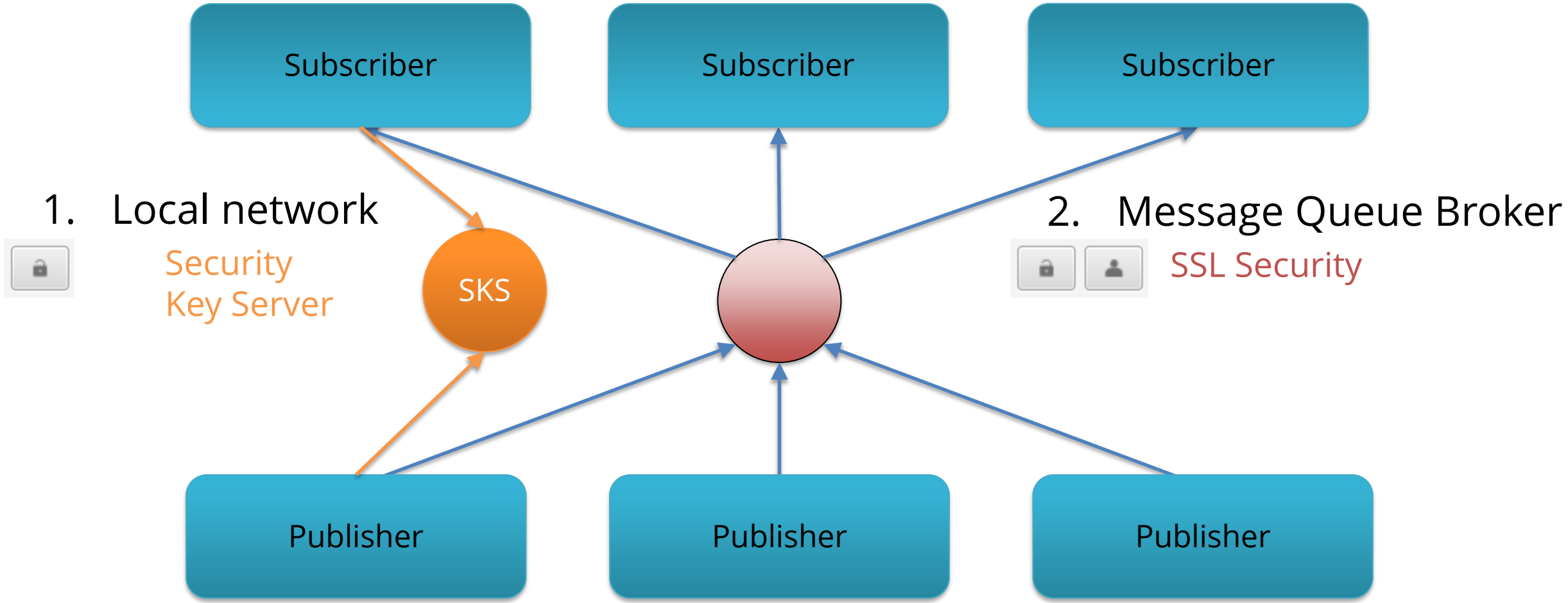
Publisher/Subscriber Networks



1. Local network
   - ❑ UDP Broadcast
   - ❑ Ethernet-APL
   - ❖ UADP Messages (optimized binary)

2. Message Queue Broker
   - ❑ MQTT
   - ❑ AMQP
   - ❖ JSON Messages
   - ❖ (or UADP)

Publisher/Subscriber Security



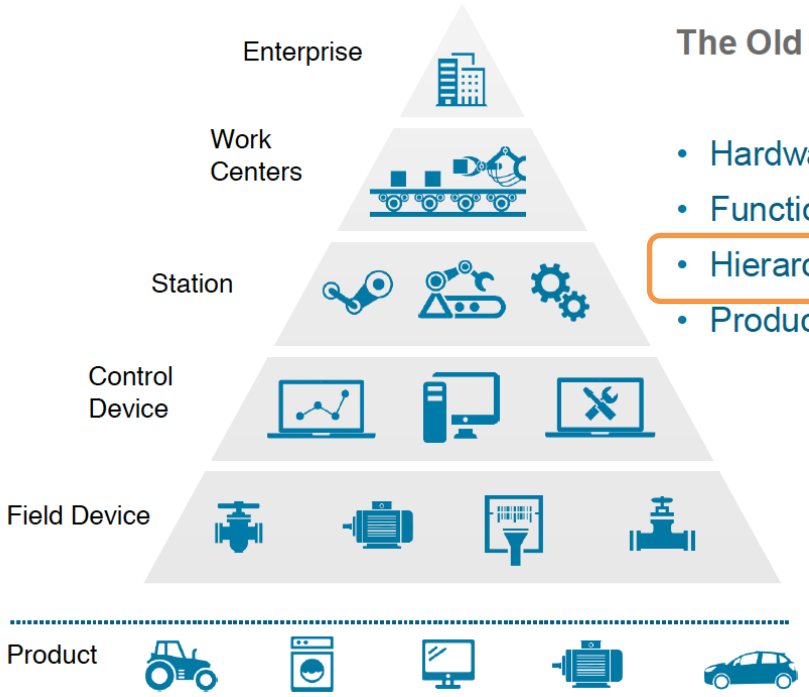
Publisher/Subscriber Pros & Cons

- No Sessions
  - \+ No connection management
  - \+ Flexible at unreliable networks
  - – Requires an external shared Security Key Server with UDP/Ethernet APL
- Multicast
  - \+ Deliver big amounts of data to many receivers
- Fixed Datasets
  - \+ Fixed resource management -> Scales to large networks
  - – Static configuration
- No Synchronous services
  - – No Write
  - – No Method calls
  - – No HistoryRead
- Configuration
  - – Especially in large networks

Contents

- OPC UA Client/Server
  - Model
  - Subscriptions
  - Scenarios
  - Scaling
  - Pros & Cons
- OPC UA Publisher/Subscriber
  - Model
  - Scaling
  - Networks
  - Pros & Cons
- PubSub Scenarios
  - Smart Manufacturing
  - Improved Client/Server Applications
  - Synchronized Servers
  - Edge-to-Cloud
- PubSub Demos
  - #1 Azure
  - #2 Amazon AWS
  - #3 “Global Production Line”
- Conclusions

PubSub Scenario: Smart Manufacturing



Client / Server

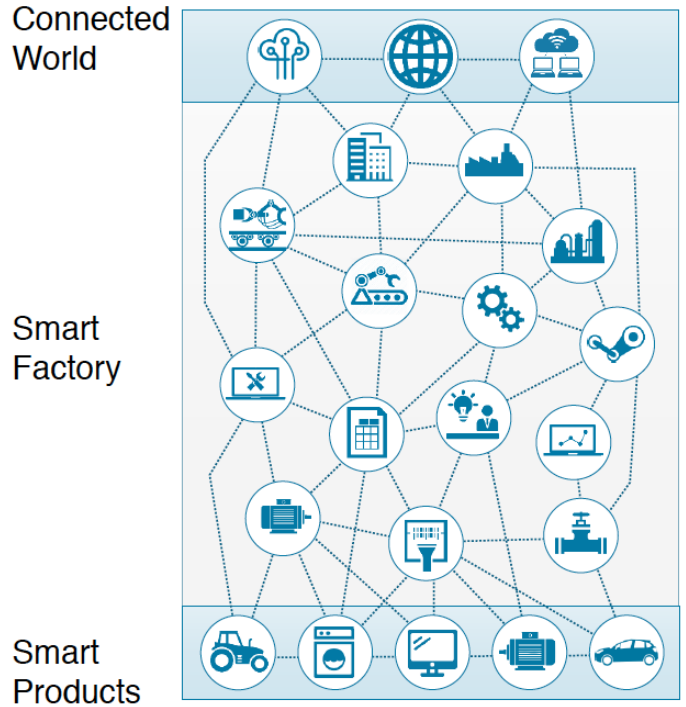
The Old World: Industrie 3.0

- Hardware-based structure
- Functions are bound to hardware
- Hierarchy-based communication
- Product is isolated

The New World: Industrie 4.0

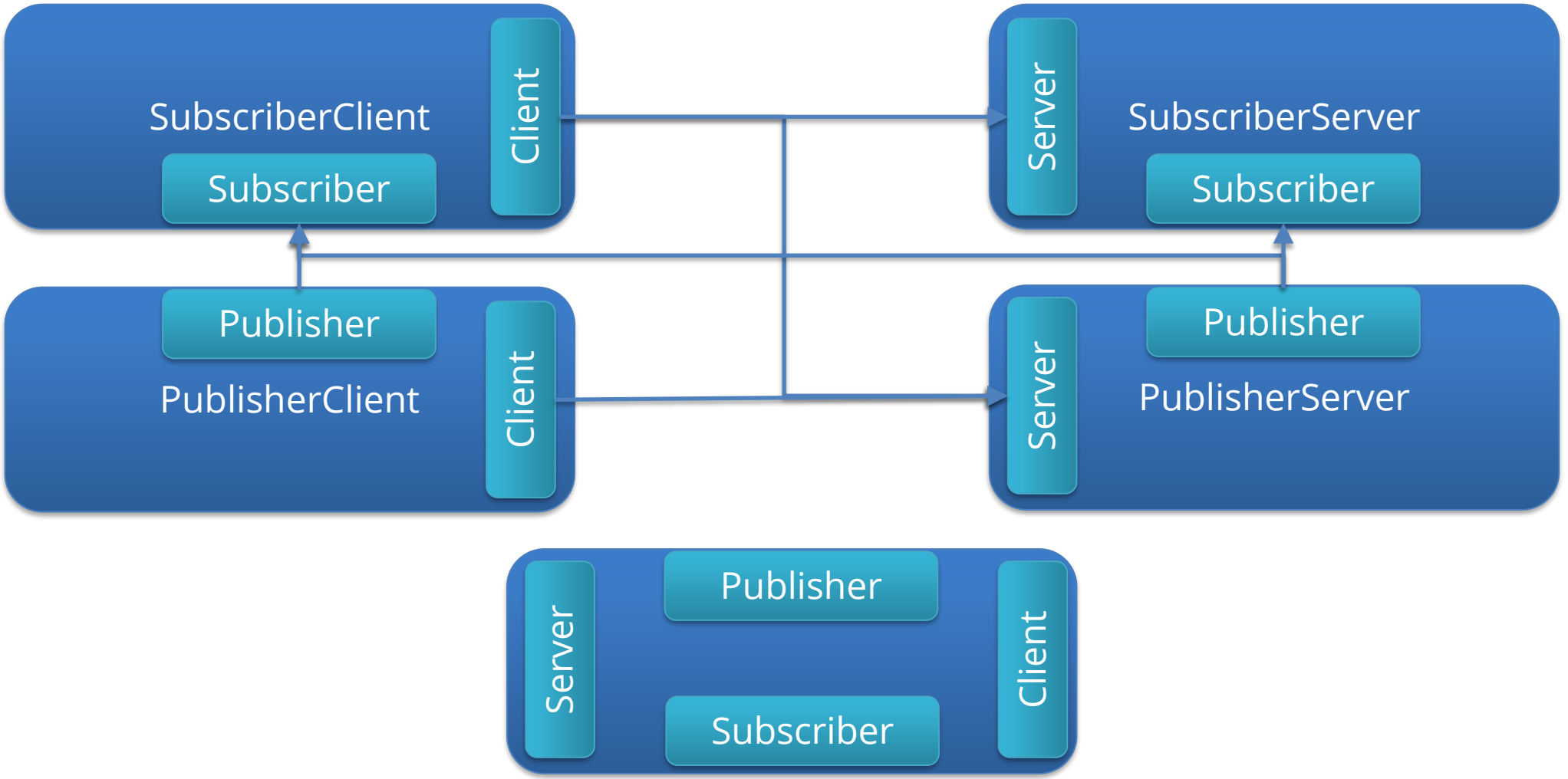
- Flexible systems and machines
- Functions are distributed throughout the network
- Participants interact across hierarchy levels
- Communication among all participants
- Product is part of the network

Publisher/Subscriber



(Martin Hankel, Plattform Industrie 4.0 / Graphics © Anna Salari, designed by freepik)

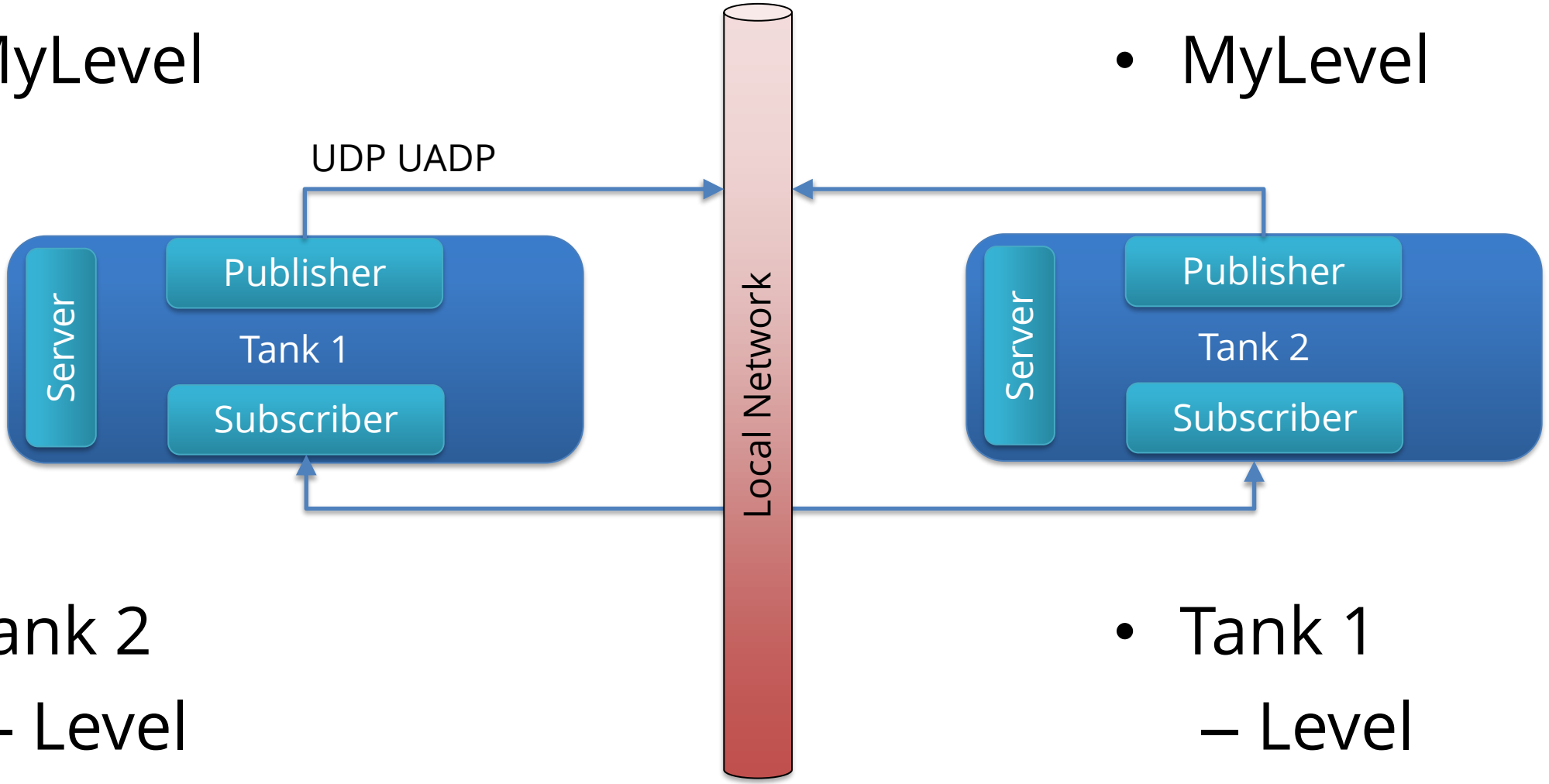
PubSub Scenario: Improved Client/Server Applications



PubSub Scenario: Synchronized Servers

- MyLevel

- MyLevel

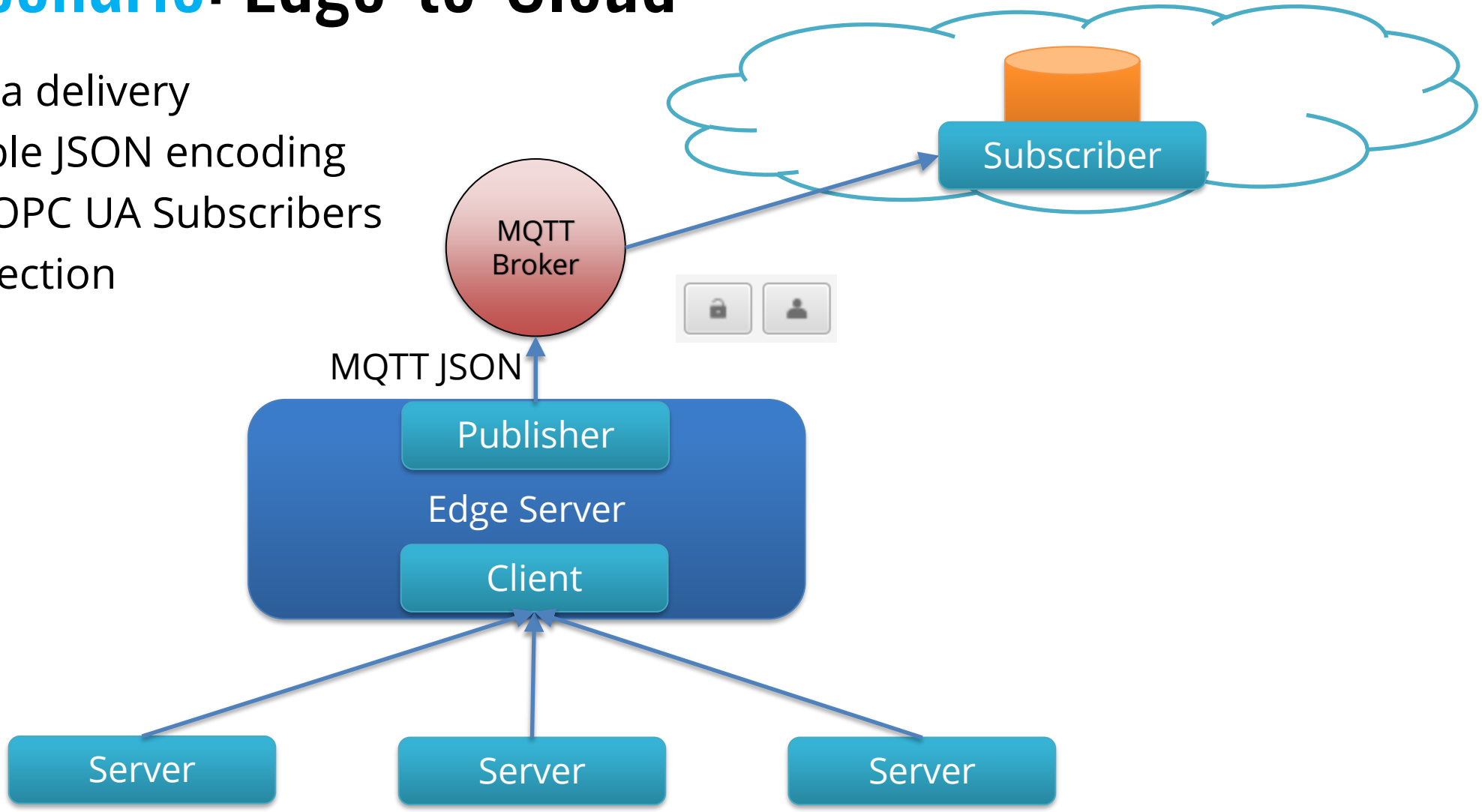


- Tank 2
  - Level

- Tank 1
  - Level

PubSub Scenario: Edge-to-Cloud

- One way data delivery
- Non-reversible JSON encoding
  - for non-OPC UA Subscribers
- Secure connection
  - SSL

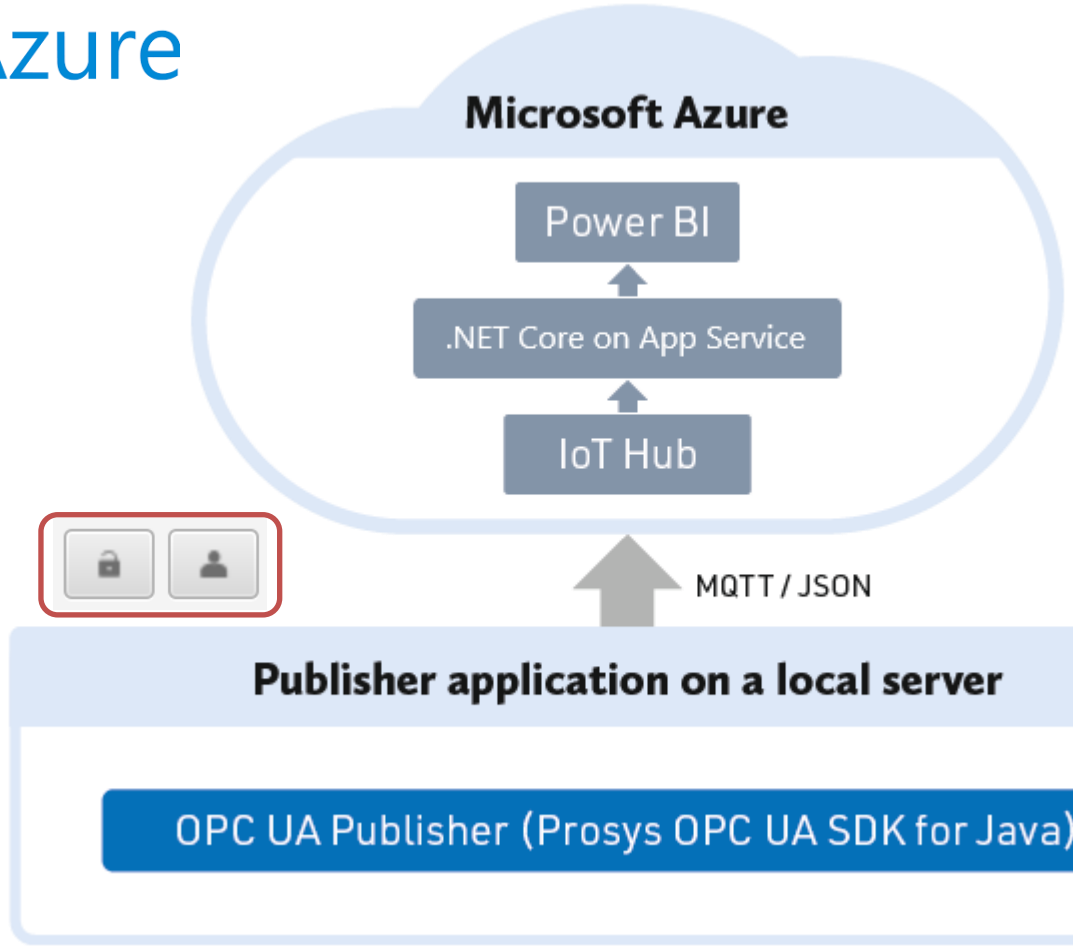


Contents

- OPC UA Client/Server
  - Model
  - Subscriptions
  - Scenarios
  - Scaling
  - Pros & Cons
- OPC UA Publisher/Subscriber
  - Model
  - Scaling
  - Networks
  - Pros & Cons
- PubSub Scenarios
  - Smart Manufacturing
  - Improved Client/Server Applications
  - Synchronized Servers
  - Edge-to-Cloud
- PubSub Demos
  - #1 Azure
  - #2 Amazon AWS
  - #3 “Global Production Line”
- Conclusions

PubSub Demo #1: Azure MQTT

Azure



DataSetMessage

```
[{
  "MyLevel":{
    "Value":73.0,
    "SourceTimestamp":"2021-07-07T17:36:29+0300"
  },
  "MyLevelDisplayName":{
    "Value":"MyLevel"
  }
}]
```

```
>samplepublisherserver.bat
  --address ssl://<IoT Hub hostname>:8883
  --encoding json
  --username <IoT Hub hostname>/<device id>/?api-version=2018-06-30
  --password <device SAS token>
  --client-id <device id>
  --queue-name devices/<device id>/messages/events/
  --metadata-queue-name devices/<device id>/messages/events/
  --non-reversible-json
```

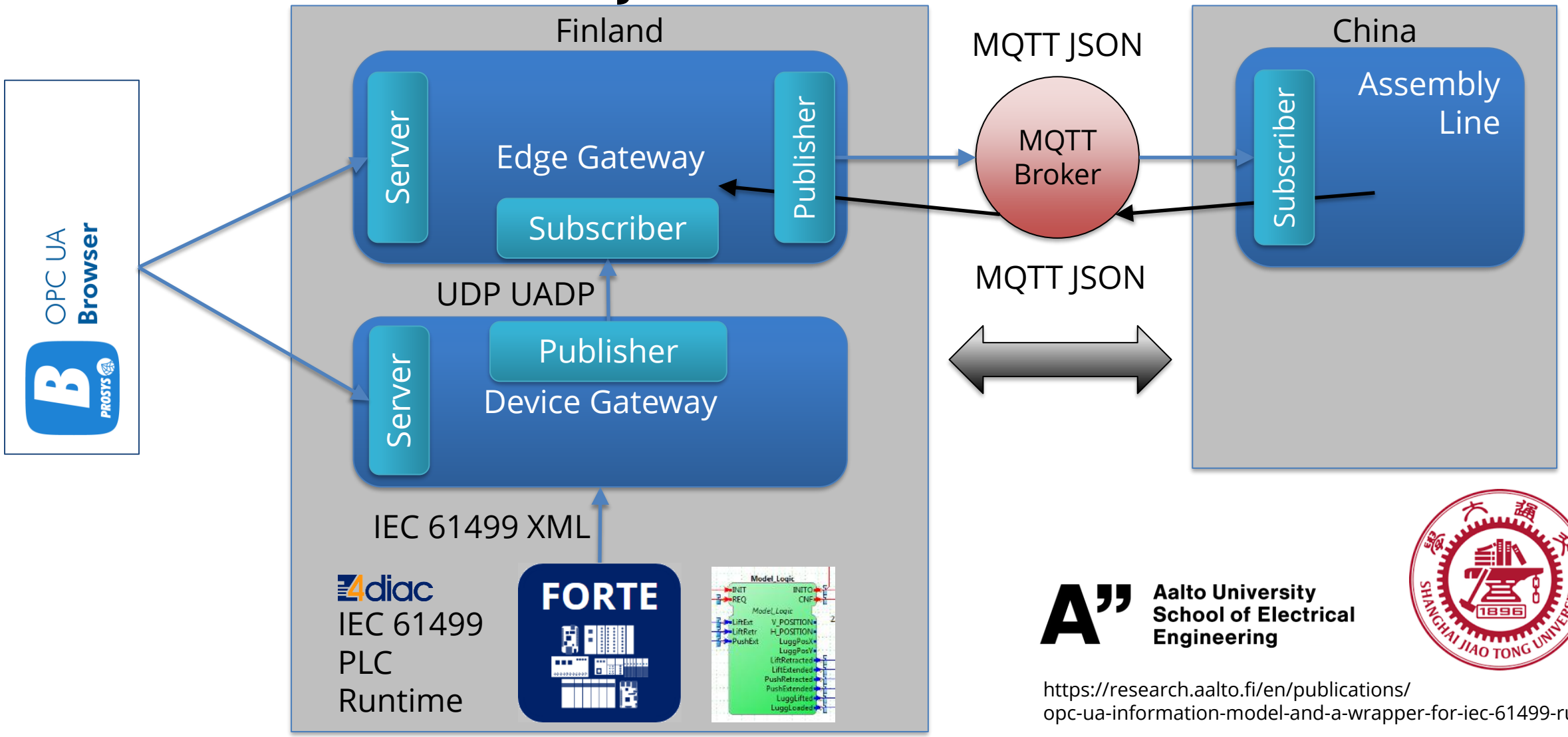
https://prosysopc.com/blog/azure-mqtt-demo/

PubSub Demo #2: Amazon AWS MQTT

- MQTT Client Authentication with SSL Certificates (verified)
- Parallel MQTT connections for extended throughput (to be verified)

(Blog article coming up soon)

PubSub Demo #3: Globally Distributed Production

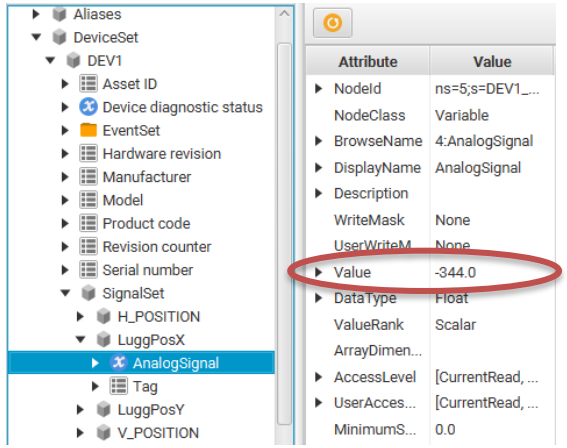


Aalto University
School of Electrical Engineering

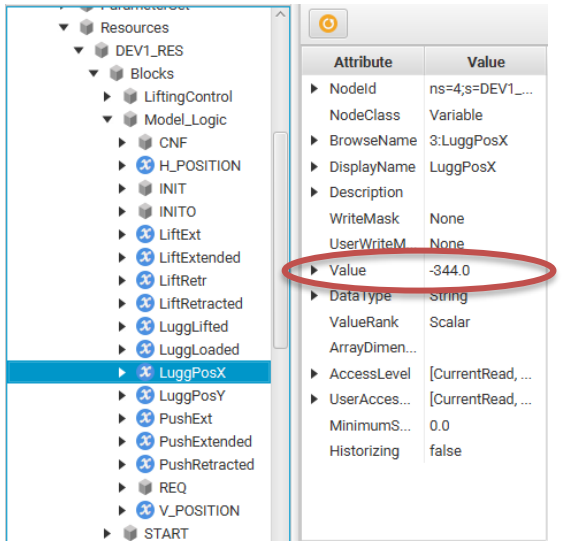
https://research.aalto.fi/en/publications/opc-ua-information-model-and-a-wrapper-for-iec-61499-runtimes

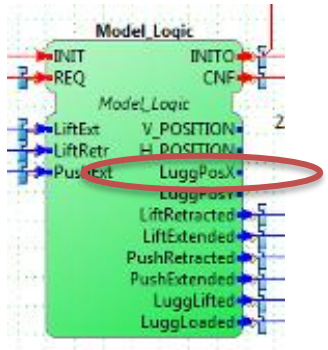
PubSub Demo #3: Information Models

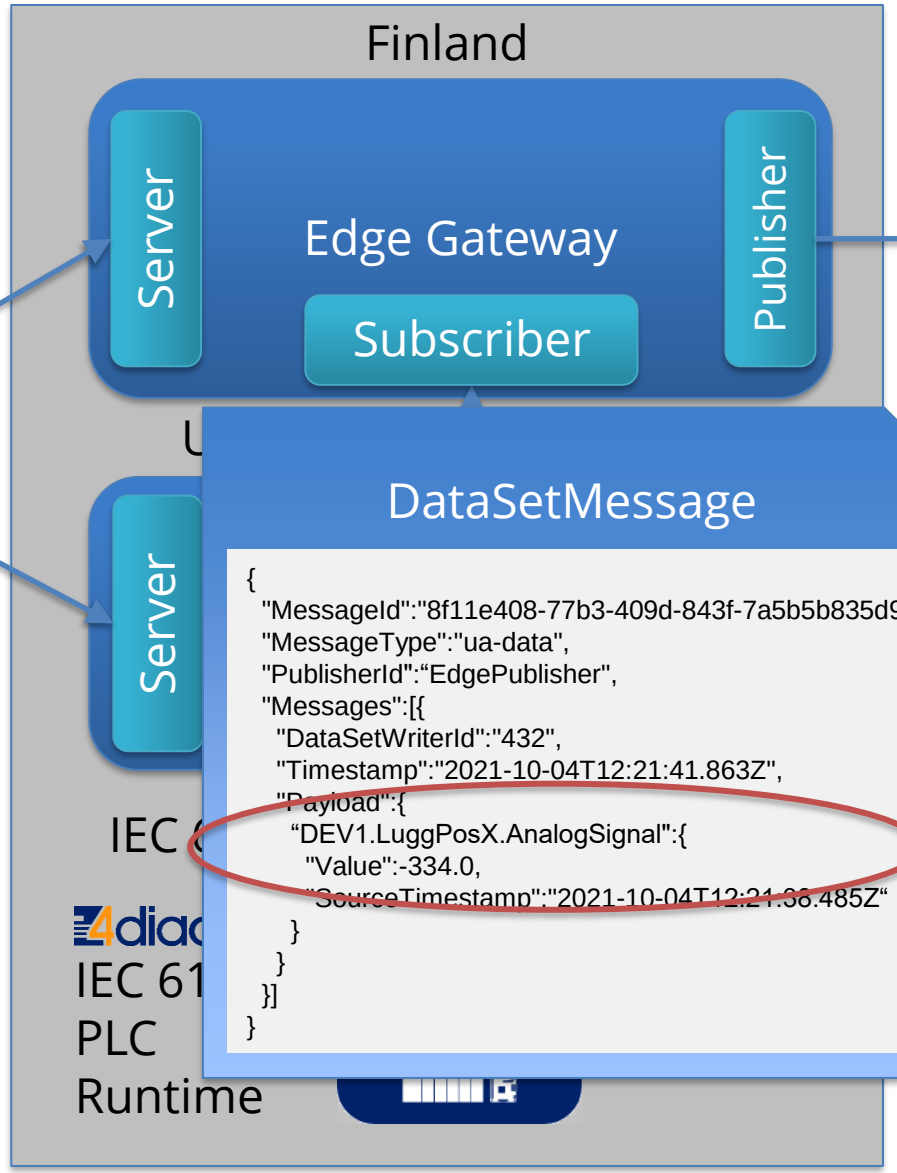
OPC 30081
Process Automation
Device
Information Model
(PA-DIM)



IEC 61499
Information Model
(Proposal)







Conclusions

- OPC UA Client/Server is still useful and required for
  - Synchronous operations (Read, Write, Method call, etc.)
  - Information models
- OPC UA Publisher/Subscriber model will enable new application scenarios
  - Improved scalability towards smart manufacturing networks
  - Improved performance in local networks -> Field Exchange
  - Connectionless data delivery: Edge-to-Cloud
- Tricky issues with PubSub
  - Configuration
  - Security (UDP/Ethernet APL)
  - ➢ Centralized Management Tools Required