

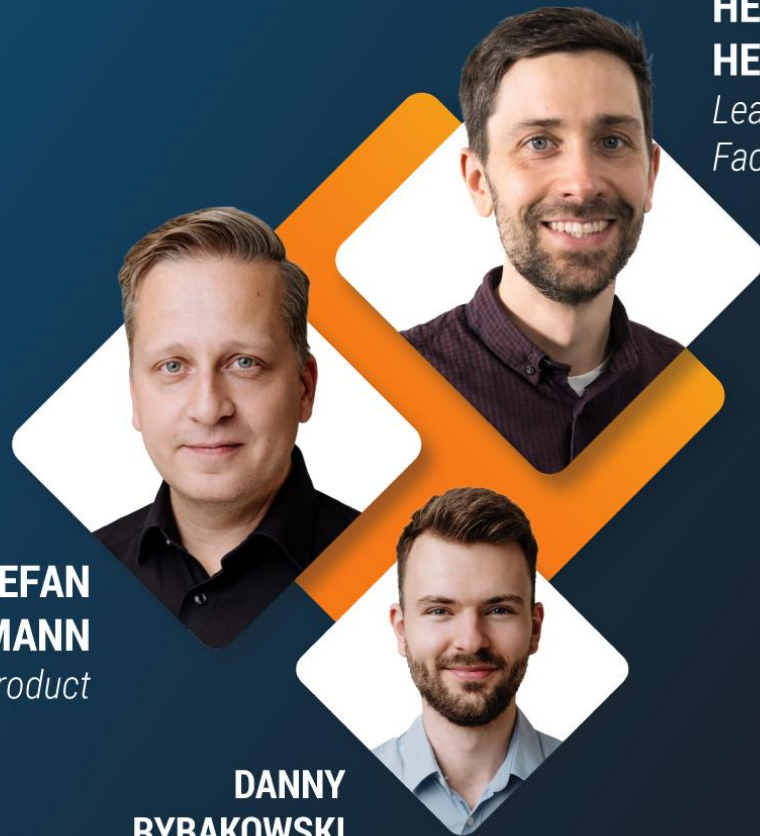
Define Your Unified Namespace

Insights and Learnings
from the Field

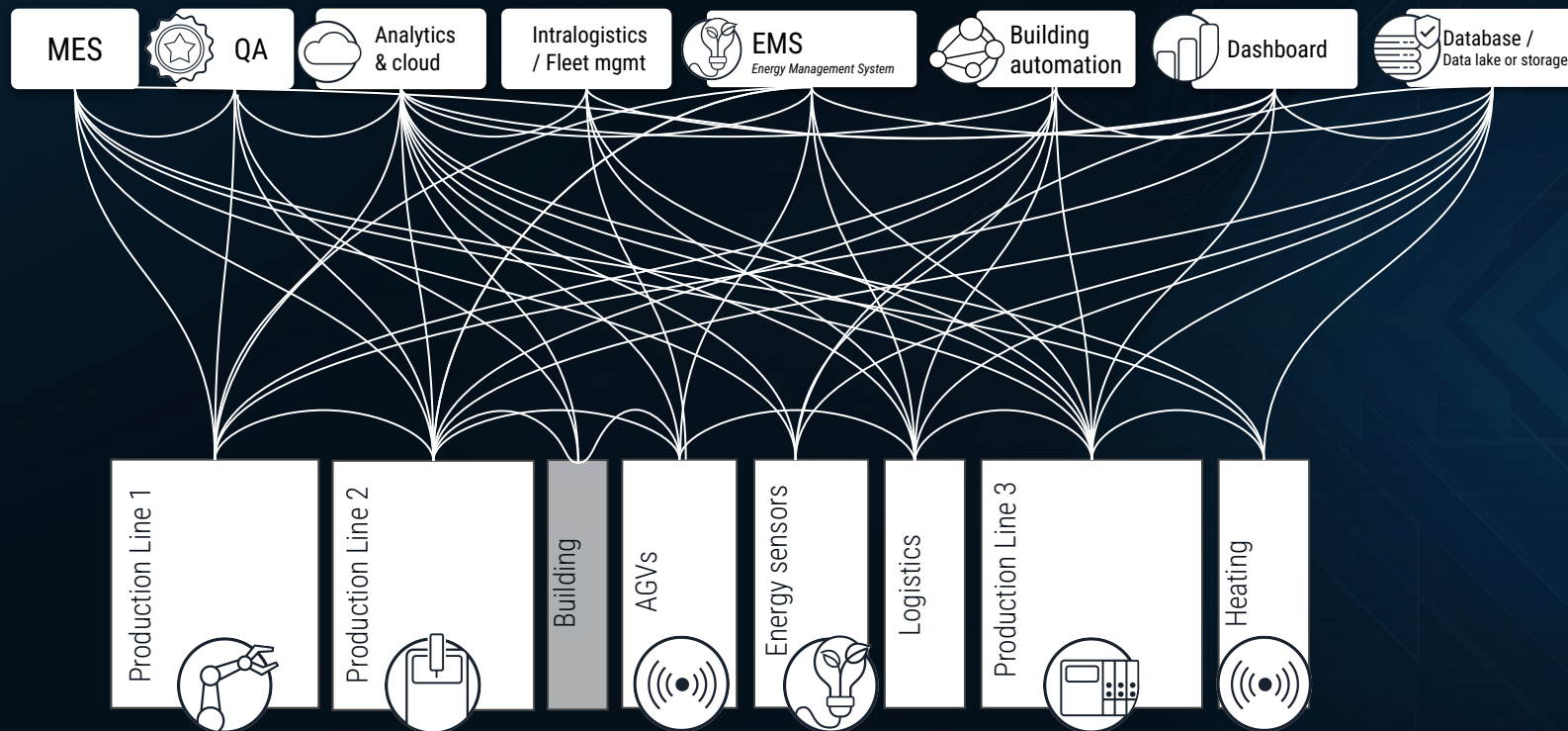
**STEFAN
HERMANN**
VP Product

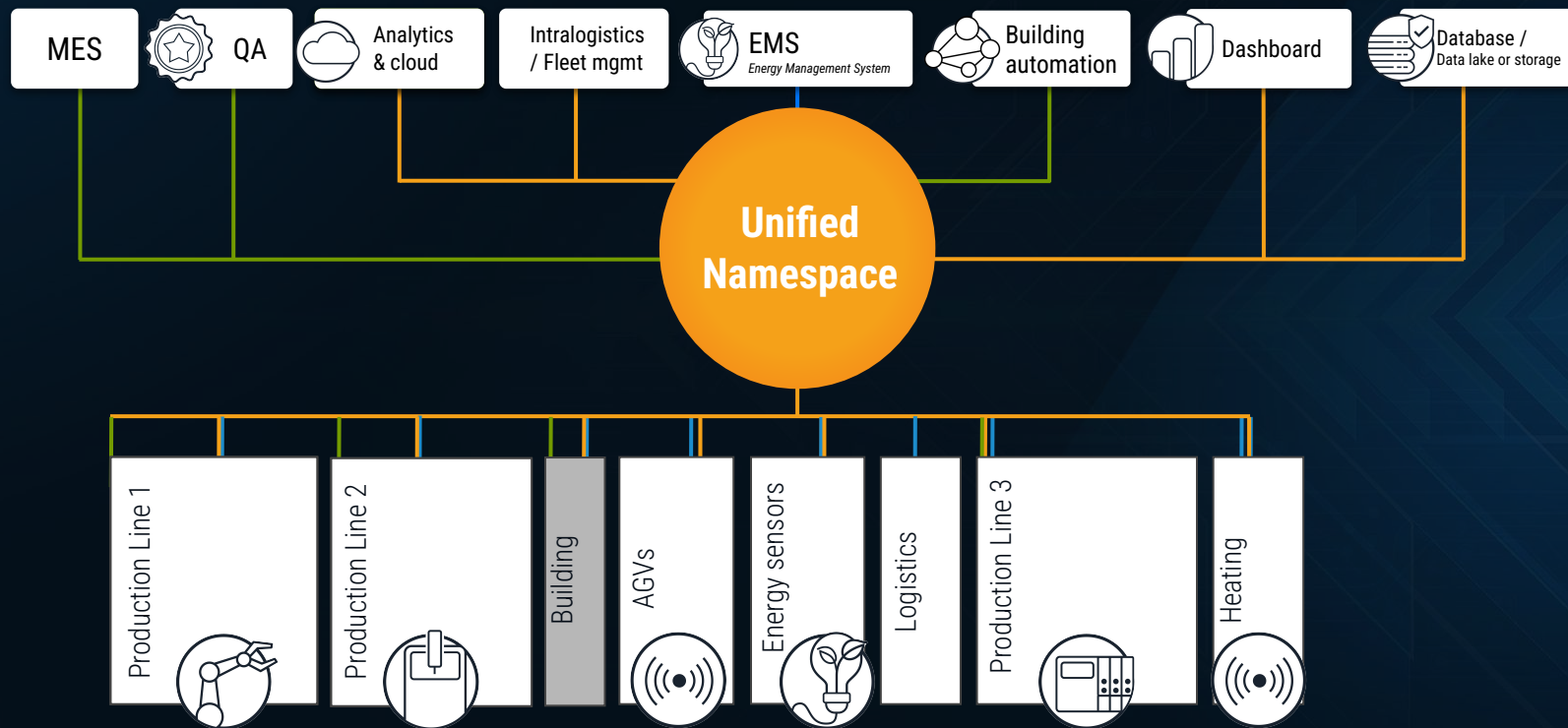
**DANNY
RYBAKOWSKI**
Head of Partnerships

**HENNING
HEINE**
*Lead Smart
Factory Expert*









The 3 layer approach to digital transformation



APPLICATION LAYER

User facing applications. Focus on good user experience and use cases that support the user to do their work more efficiently.



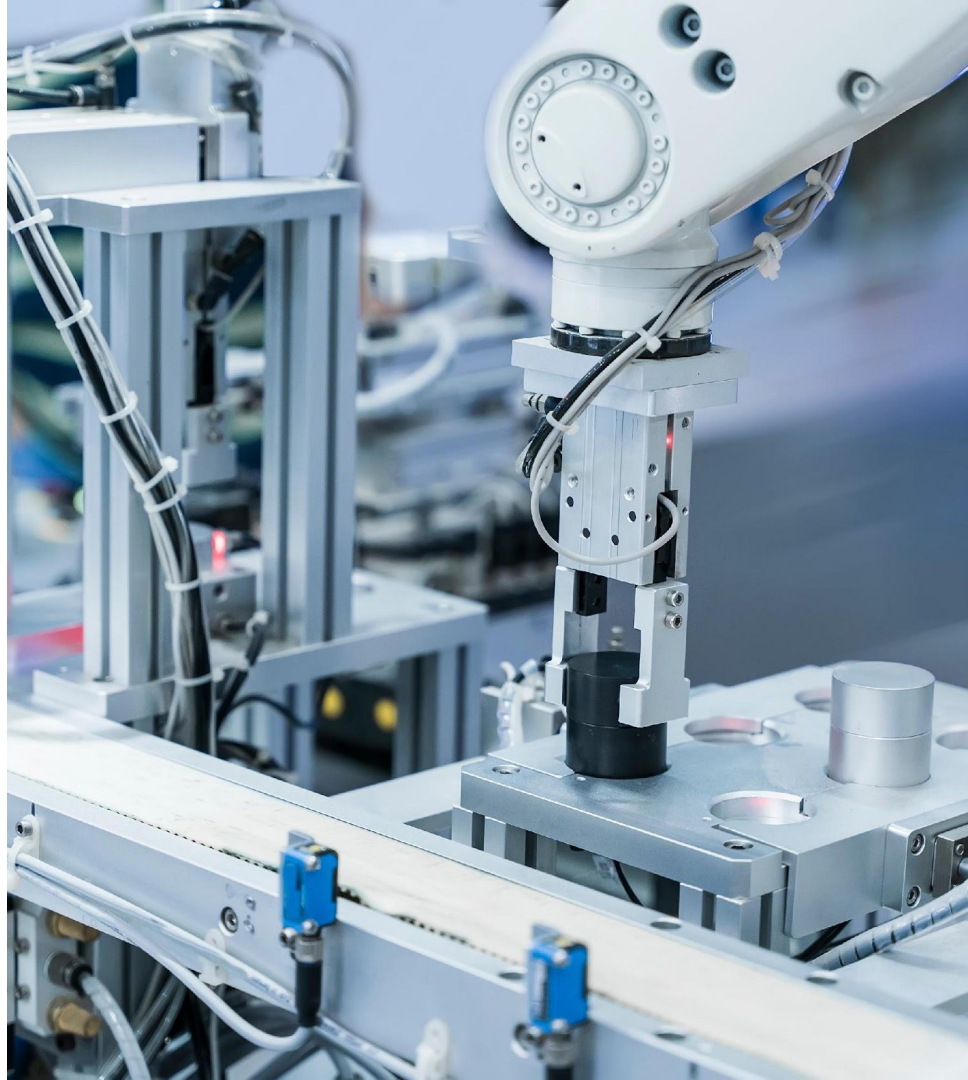
DATA LAYER

Simple machine connectivity, real-time view of the entire production via Unified Namespace, data normalization, data analytics

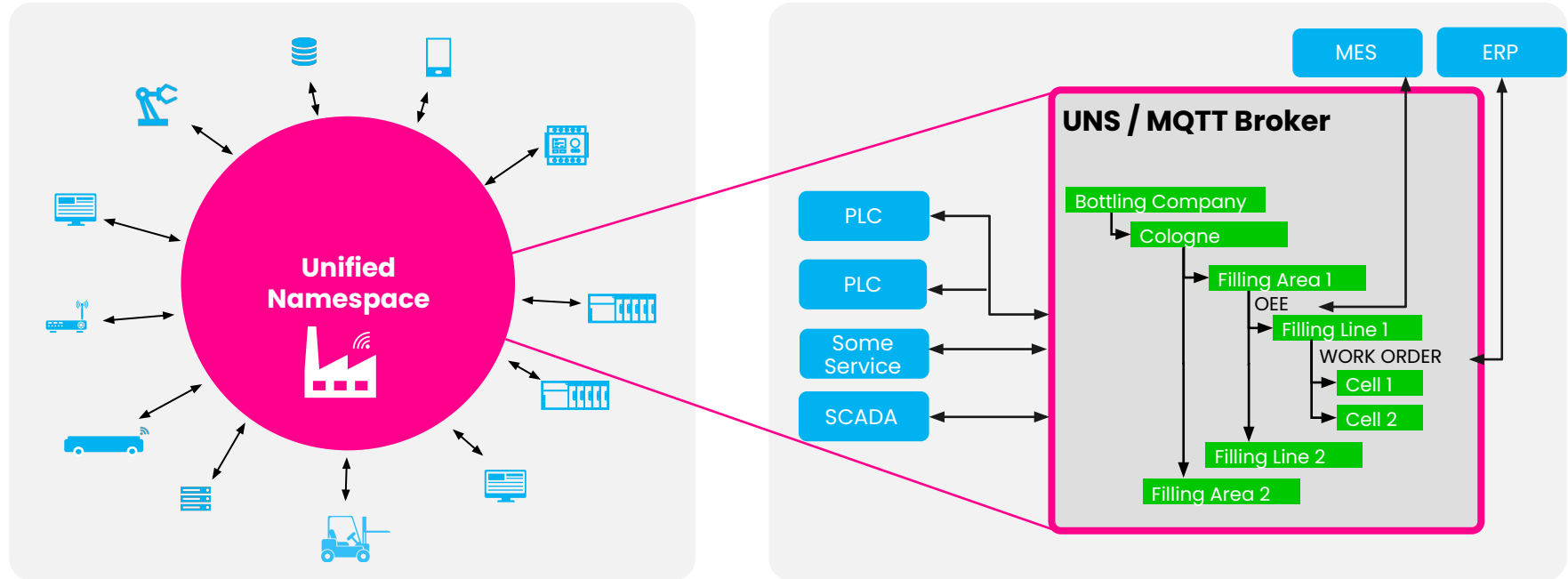


INFRASTRUCTURE LAYER

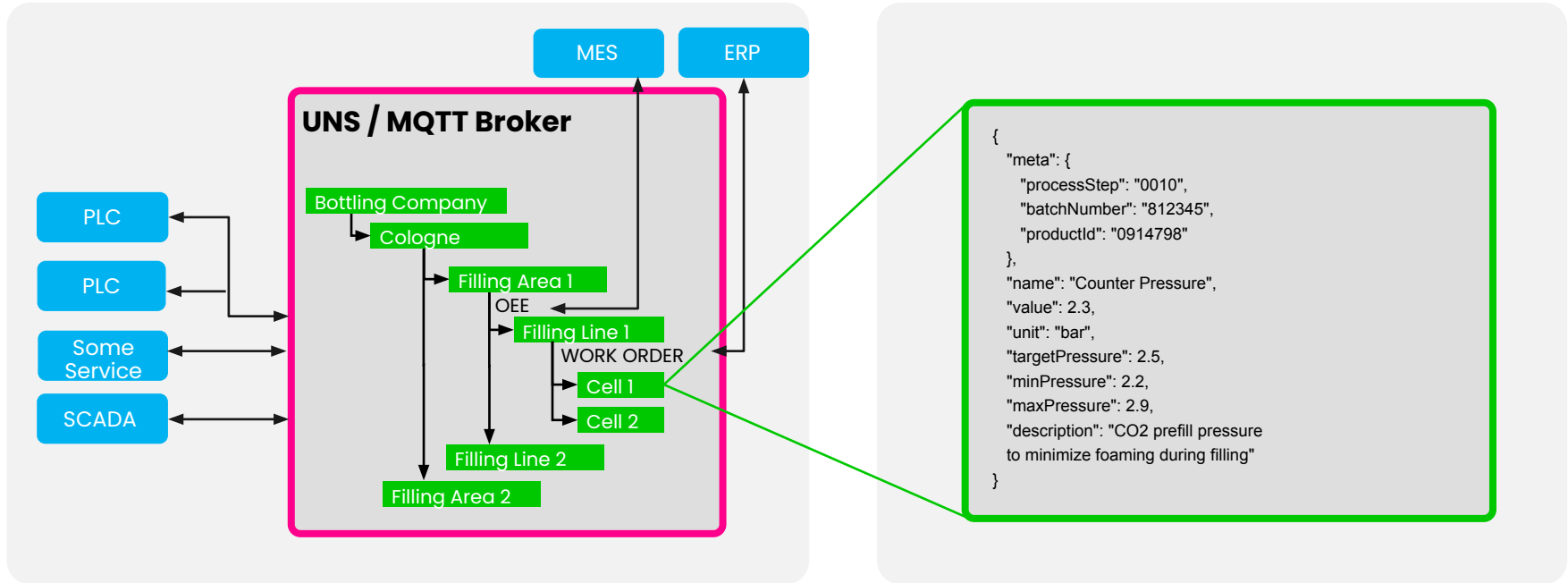
Support for Cloud & Edge, support for all relevant workloads from services to ML, scalable, highly available, centralized management



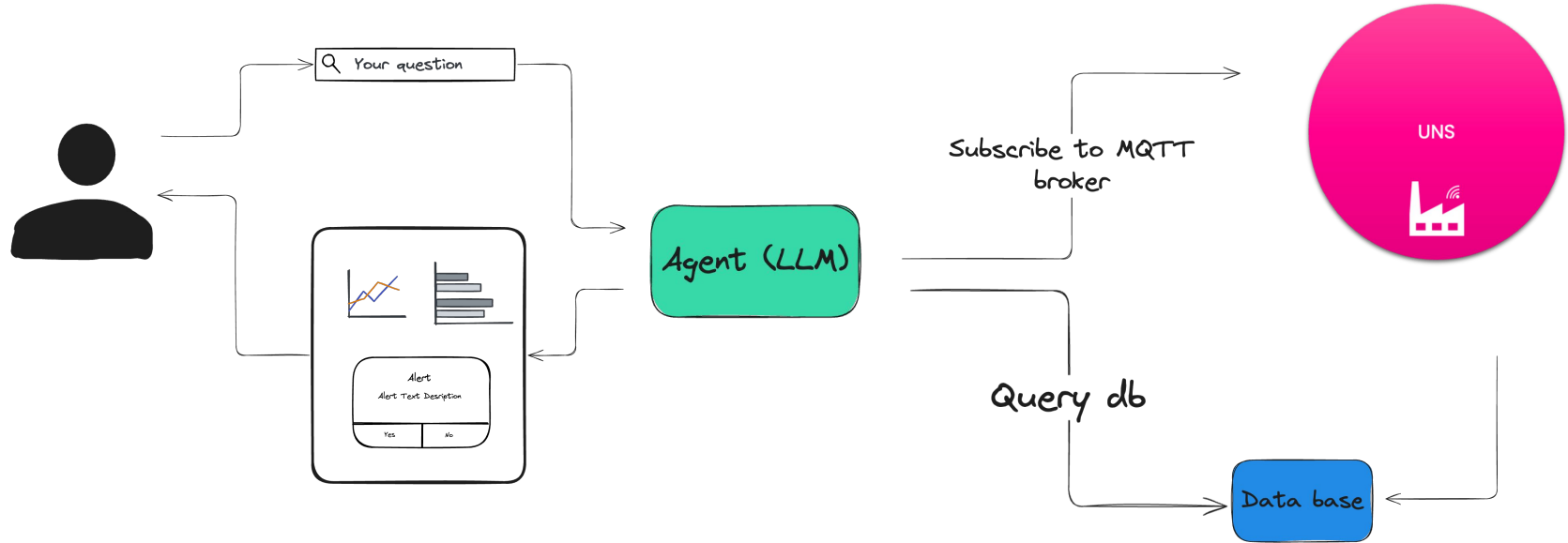
A Unified namespace is a paradigm shift – the goal is to create one source for all relevant information



A Unified namespace is a paradigm shift – the goal is to create one source for all relevant information



The Factory UNS Agent



Speed of a punching machine

Machine/system: punching_machine (Machine01)

Protocol: Modbus-TCP

Query (Modbus-Address): 40001

Query

40001

Data

Register 40001 → 450

MQTT-Topic according to ISA-95

carfactory/plant1/pressShop/machine01/speed

JSON output

```
{
  "timestamp": "2024-02-13T12:00:00Z",
  "value": 450,
  "unit": "strokes_per_min",
  "quality": "good"
}
```

Temperature of a punching machine

Machine/system: Temperatursensor (Machine02)

Protocol: OPC UA

Query (OPC UA Node ID): ns=2;s=Machine02.Temperature

Query

```
ns=2;s=Machine02.Temperature
```

Data

```
<DataValue>
  <Value>
    <Float>75.3</Float>
  </Value>
  <SourceTimestamp>2024-02-13T12:01:00Z</SourceTimestamp>
</DataValue>
```

MQTT-Topic according to ISA-95

```
carfactory/plant1/pressShop/machine02/temperature
```

JSON output

```
{
  "timestamp": "2024-02-13T12:01:00Z",
  "value": 75.3,
  "unit": "Celsius",
  "quality": "good"
}
```

Status of a painting robot

Machine/system: Painting robot (Robot05)

Protocol: OPC UA

Query (OPC UA Node ID): ns=3;s=PaintShop.Robot05.State

Query

```
ns=2;s=Machine02.Temperature
```

Data

```
<DataValue>
  <Value>
    <String>idle</String>
  </Value>
  <SourceTimestamp>2024-02-13T12:02:30Z</SourceTimestamp>
</DataValue>
```

MQTT-Topic according to ISA-95

```
carfactory/plant1/paintShop/robot05/state
```

JSON output

```
{
  "timestamp": "2024-02-13T12:02:30Z",
  "state": "idle",
  "lastJobID": "LKP_987654",
  "error": null
}
```

Quality control – defect detection

Machine/system: Quality control (camera system)

Query: HTTP REST API

Query

```
GET http://quality-control/api/defects
```

Data

```
{
  "timestamp": "2024-02-13T12:02:30Z",
  "vehicleID": "VIN_1234567890",
  "defectType": "paint_scratch",
  "severity": "minor",
  "imageUrl": "http://quality-control/images/defect_12345.jpg"
}
```

MQTT-Topic according to ISA-95

```
carfactory/plant1/qualityControl/station3/defectDetected
```

JSON output

```
{
  "timestamp": "2024-02-13T12:02:30Z",
  "vehicleID": "VIN_1234567890",
  "defectType": "paint_scratch",
  "severity": "minor",
  "imageUrl": "http://quality-control/images/defect_12345.jpg",
  "detectedBy": "Camera_System_02"
}
```

Material stock (ERP)

Machine/system: ERP-System (SAP)

Query: SQL-Datenbank

Query

```
SELECT material_id, quantity, unit, reorder_level, last_updated
FROM inventory
WHERE material_id = 'ALU_2024';
```

Data

```
ALU_2024 | 1500 | kg | 500 | 2024-02-13 12:10:00
```

MQTT-Topic according to ISA-95

```
carfactory/business/erp/supplyChain/materialStock
```

JSON output

```
{
  "timestamp": "2024-02-13T12:10:00Z",
  "materialID": "ALU_2024",
  "quantity": 1500,
  "unit": "kg",
  "reorderLevel": 500
}
```

New customer order

Machine/system: ERP-System (SAP)

Query: Kafka

Query

orders.new

Data

```
{
  "orderId": "ORDER_98765",
  "customer": "FleetCorp GmbH",
  "vehicleModel": "CarX Turbo",
  "quantity": 200,
  "expectedDelivery": "2024-04-01T00:00:00Z"
}
```

MQTT-Topic according to ISA-95

carfactory/business/erp/orders/new

JSON output

```
{
  "timestamp": "2024-02-13T12:15:00Z",
  "orderId": "ORDER_98765",
  "customer": "FleetCorp GmbH",
  "vehicleModel": "CarX Turbo",
  "quantity": 200,
  "expectedDelivery": "2024-04-01"
}
```

Production planning (MES)

Machine/system: MES-System

Protocol: HTTP REST API

Query

```
GET http://mes-system/api/productionSchedule?date=2024-02-14
```

Data

```
{
  "scheduleID": "PLAN_456",
  "start": "2024-02-14T06:00:00Z",
  "end": "2024-02-14T14:00:00Z",
  "plannedVehicles": 120,
  "lineID": "assemblyA"
}
```

MQTT-Topic according to ISA-95

```
carfactory/business/mes/productionSchedule
```

JSON output

```
{
  "timestamp": "2024-02-13T12:15:00Z",
  "scheduleID": "PLAN_456",
  "start": "2024-02-14T06:00:00Z",
  "end": "2024-02-14T14:00:00Z",
  "plannedVehicles": 120,
  "lineID": "assemblyA"
}
```

Questions & Answers



Stefan Hermann

stefan.hermann@cybus.io



Henning Heine

henning.heine@maibornwolff.de



Danny Rybakowski

danny.rybakowski@cybus.io

More resources



[UNS Whitepaper](#)



[UNS Demonstrator @HM](#)



[UNS Workshop @MUC](#)