OPC DAY
FINLAND 2022
29.11.2022

Open Industry 4.0
Open Edge Computing

Konrad Heidrich, Hilscher

FINNISH SOCIETY OF AUTOMATION
SUOMEN AUTOMAATIOSEURA RY

SPONSORS:

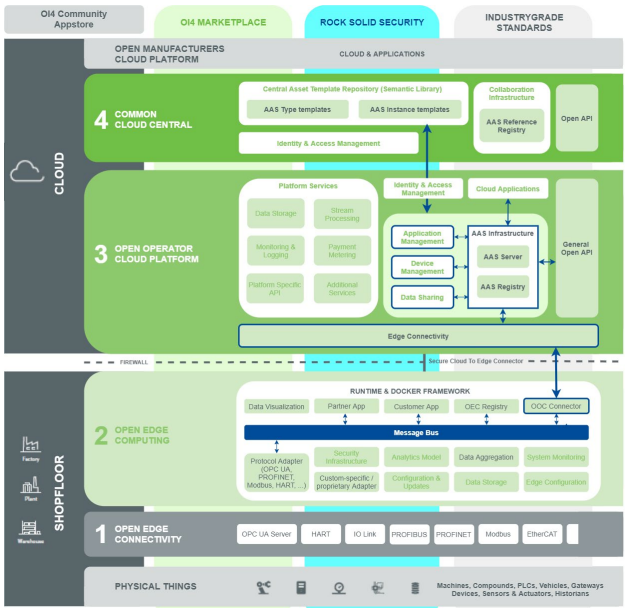
aws BECKHOFF brightly inray NAPCON NOKIA Novotek OPC FOUNDATION OPEN INDUSTRY 4.0 ALLIANCE PROSYS OPC Unified Automation Valmet Wapice

Open Industry 4.0 Alliance

DETAILS TO LAYER 2 OF THE REFERENCE ARCHITECTURE

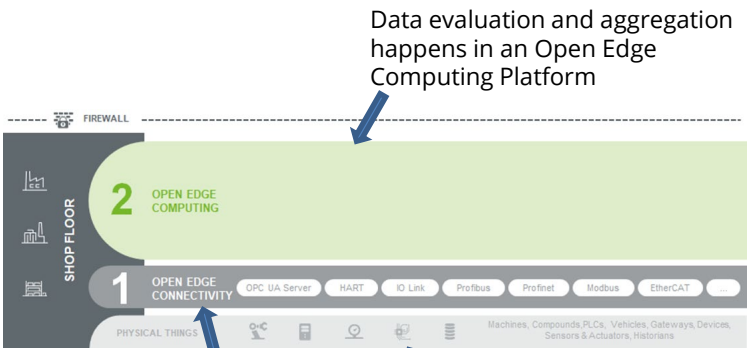
OI4 Reference Architecture

- ...as shown in the previous presentation
- We have now a closer look into Layer 2 and how OPC UA technology helps us with our challenges in interoperability
- Layer 2 of the OI4-RA
  - is responsible for Open Edge Computing
  - is a scalable platform
  - connects to IT and/or cloud on northbound and to shopfloor infrastructure on southbound



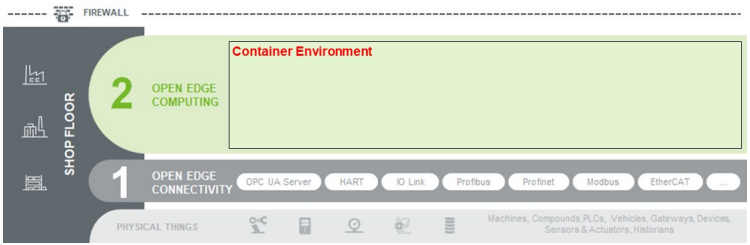
Open Edge Computing Platform

- ARM32, ARM64 or x86-64 based computing platform
- Number of cores, amount of RAM/ROM, HW support for security are up to the use case
- At least an ETH interface is neccessary
- Driven by a Linux operating system



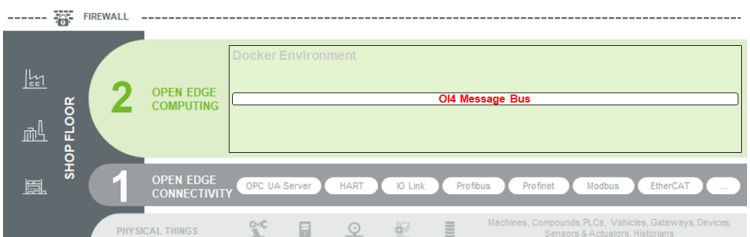
Containerized Applications

- Container environment as industrial standard to run containerized Open Industry 4.0 Alliance applications
- Applications have defined interfaces for interoperability in the OI4 ecosystem
- Containerized applications provide a scalable, cascadable and lightweight solution



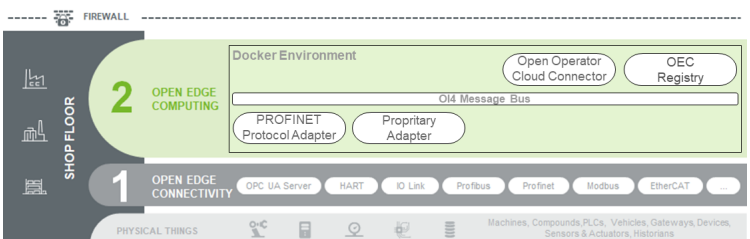
Standardized Message Bus

- MQTT technology as widely used standard for industrial data exchange
- Well defined topic structure for better interoperability
- Standardized communication and semantics via payload, based on OPC UA PubSub JSON



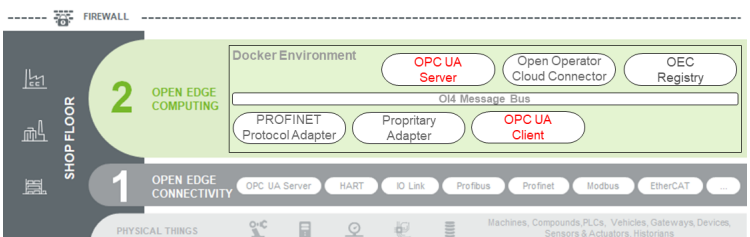
Connectors and Applications

- Connect to OT networks to detect assets and provide read/write access to them
  Handle standardized or proprietary communication and provide domain agnostic information to the Message Bus
- Connect to IT and Cloud to exchange information
  Encapsulate secure communication, handle domain agnostic information from the Message Bus
- Aggregate and compute information



OPC UA Client/Server Integration

- A Protocol Adapter with OPC UA client functionality reaches OPC UA servers on the shop floor
- An OPC UA server represents the content of the Open Edge Computing Platform to any connected OPC UA client with the right credentials



Open Industry 4.0 Alliance

THE MESSAGE BUS

MQTT as communication protocol

- Several Edge platforms are using MQTT for interoperability between (micro) services
  - Lightweight protocol, easy to integrate and to understand
  - Easy and fast data exchange
  - Lots of engineering is necessary afterwards because of missing definitions for topic structure and payload

- OI4 messaging over MQTT follows a topic schema
  - Evaluable (full schema validation possible)
    `Oi4/<ServiceType>/<AppId>/<Method>/<Resource>[[/<Source>]/<Filter>]`
  - Basic information is reachable without knowing details about other applications
  - OPC UA PubSub JSON as payload

Open Industry 4.0 Alliance

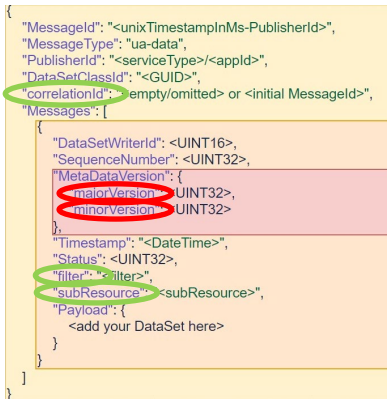
OPC UA PUBSUB JSON

OPC UA PubSub JSON over MQTT

- We wanted a self-explaining payload
- Sparkplug was in discussion as lightweight implementation, but did not solve our needs
- Neither did OPC UA PubSub at this time, but
  - It had a good foundation for our use case needs
  - OPC UA concepts are well accepted in the industry
  - Several members of the OI4 are also heavily involved in OPC UA business
  - We took the bet, that OPC-specs will solve our upcoming challenges (e.g. PubSub Actions)
- OPC UA Client/Server is an integral part of the addressed industries. By using OPC UA PubSub we provide better interoperability to this world.

Is this really OPC UA PubSub?

OI4 related NetworkMessage following V1.0 Guideline



```
{
  "MessageId": "<unixTimestampInMs-PublisherId>",
  "MessageType": "ua-data",
  "PublisherId": "<serviceType>/<appId>",
  "DataSetClassId": "<GUID>",
  "correlationId": "<empty/omitted> or <initial MessageId>",
  "Messages": [
    {
      "DataSetWriterId": <UINT16>,
      "SequenceNumber": <UINT32>,
      "MetaDataVersion": {
        "majorVersion": <UINT32>,
        "minorVersion": <UINT32>
      },
      "Timestamp": "<DateTime>",
      "Status": <UINT32>,
      "filter": "<filter>",
      "subResource": "<subResource>",
      "Payload": {
        <add your DataSet here>
      }
    }
  ]
}
```

- OI4 detected missing functionality, which got solved by adding specific keys to NetworkMessage and DataSetMessage
- OI4 has adopted the notation 1:1 from Part 14

Technicians of the OPCF and the OI4 started a discussion at the Hannover Fair 2022 to solve the problems.

Members of both organizations joined the OPC UA PubSub Prototyping WG.

Is this really OPC UA PubSub?

OI4 related NetworkMessage following V1.1 Guideline

```
{
  "MessageId": "<unixTimestampInMs-PublisherId>",
  "MessageType": "ua-data",
  "PublisherId": "<ServiceType>/<AppId>",
  "DataSetClassId": "<GUID>",
  "CorrelationId": "<empty/omitted> or <initial MessageId>",
  "Messages": [
    {
      "DataSetWriterId": <UINT16>,
      "SequenceNumber": <UINT32>,
      "MetaDataVersion": {
        "MajorVersion": <UINT32>,
        "MinorVersion": <UINT32>
      },
      "Timestamp": "<DateTime>",
      "Status": <UINT32>,
      "Filter": "<Filter>",
      "Source": "<Oi4Identifier>",
      "Payload": {
        <add your DataSet here>
      }
    }
  ]
}
```

- OI4 changed all notations to PascalCase to follow the global OPCF rules.
- OI4 and OPCF evaluated the missing functionality and found out it is basically implemented in binary protocol, but not yet documented for JSON coding.

It looks promising to use **`DataSetWriterName`** and **`WriterGroupName`** instead of **`Filter`** and **`Source`** in upcoming OI4 Guideline.

We are all motivated and committed to OPC UA, so we will find a solution for **`CorrelationId`** too (e.g. ApplicationSpecificHeaders)!

This really is OPC UA PubSub!

With the upcoming OEC Guideline V1.2 (or maybe 2.0) of the OI4 we want to solve the following issues:

- NetworkMessage and DataSetMessage will be OPC UA PubSub JSON compliant!
- If PubSub Actions (method calls over MQTT) are defined by the OPCF, the OI4 will adopt this mechanism.
- Other minor issues can be discussed now in a direct way through the new OPC UA PubSub WG

The Open Industry 4.0 Alliance is fully committed to use OPC UA PubSub JSON as defined in Part 14!

THANK YOU!

Konrad Heidrich
Senior IOT Solution Architect
Telefon: +49 (0) 6190 9907-0
E-Mail: kheidrich@hilscher.com
Web: www.hilscher.com
Rheinstraße 15 I 65795 Hattersheim I Germany