OpenMTC - oneM2M middleware for prototyping IoT services

Ronald Steinke

Business Unit Software-based Networks (NGNI)
Fraunhofer FOKUS Institute for Open Communication Systems
OPENMTC
PROTOTYPING M2M APPLICATIONS

Ronald Steinke
November 23, 2017
AGENDA

1. Motivation and Challenges
2. oneM2M
3. OpenMTC: Open Machine-Type Communication
4. Summary and Future Outlook
MACHINE-TO-MACHINE VS INTERNET OF THE THINGS

CPS + Connectivity

- Related to **connected devices and objects**
- Focusing on **connectivity and device management**
- Heterogeneous **access technologies**
  - Cellular: 2G, 3G, 4G and 5G
  - Low power local area networks: Bluetooth, WLAN, Zigbee, etc.
  - Low power wide area networks: Ultra Narrow Band, ultra-long range spread spectrum, etc.
- **Various transport protocols**: HTTP, CoAP, MQTT, etc.
- Module and devices with sensing, actuating and connectivity capabilities
- Roles of various stakeholders in the value chain

Data

- Focusing on the **data integration**, data **processing**, analytics and service **mash-up**
- Based on **connected objects** (M2M or data)
- **Heterogeneous data streams** from different sources
  - Dedicated M2M platforms
  - Public Internet
- **Various data models**
- Virtual devices and objects
- Real-time and historical **data analytics**
- Simple and open API for enabling application **mash-up**
TECHNOLOGICAL ENABLERS FOR IOT-BASED SOLUTION

Data

Connectivity

CPS
BUSINESS PROPOSITION

- Improving operational efficiency
- Enabling connectivity and visibility
- Increasing revenue stream
- Improving customer experience
- Enabling new business models
HOLISTIC OVERVIEW OF KEY CHALLENGES

Market fragmentation

Interoperability and portability

Costs reduction in M2M/IoT Technologies

Technological challenges: BW, connections, & signaling
OTHER CHALLENGES

- Legacy Market, installed base with communications already in place
- Coverage
- Accessibility of Ethernet, Wi-Fi or DSL connections
- Global Deployment
- Wireless interference issues in certain markets
- Security
- Installation and Easy to use
- Cost
AGENDA

1. Motivation and Challenges

2. oneM2M

3. OpenMTC: Open Machine-Type Communication

4. Summary and Future Outlook
M2M STANDARDS LANDSCAPE

- M2M standards address in general only one part of the M2M communication
- ETSI, OMA & 3GPP standards together develop a complete network oriented M2M communication architecture
- oneM2M
  - Release 1 (published Jan 2015, updated Mar 2016)
  - Release 2 (published Aug 2016)
ONEM2M AS GLOBAL PARTNERS OF INTERNATIONAL STANDARDS BODIES

- **TIA (Telecommunications Industry Association)** of the U.S.
- **ETSI (European Telecommunications Standards Institute)**
- **ARIB (Association of Radio Industries and Businesses of Japan)**
- **ATIS (Alliance for Telecommunications Industries Solutions)** of the U.S.
- **CCSA (China Communications Standards Association)**
- **TSDSI (Telecommunications Standard Development Society India)**
- **TIC (Telecommunications Industry Association of Japan)**
- **TTA (Telecommunications Technology Association of Korea)**
- **Fraunhofer FOKUS**
- **Continua Health Alliance**
- **Home Gateway Initiative (HGI)**
- **Open Mobile Alliance (OMA)**
Application Layer (AE):
- Comprises oneM2M Applications and related business and operational logic.

Common Services Entity (CSE):
- A set of service functions common to the M2M environment.
- Common Services Entity can utilize Underlying Network capabilities and can interact with each.

Underlying Network Services Entity (NSE):
- Provides services to the CSEs.
- Examples of such services include device management, location services and device triggering.
ONEM2M COMMON SERVICE FUNCTIONS

ASLM – Application and Service Layer Management CSF
CMDH – Communication Management and Delivery Handling CSF
DMR – Data Management and Repository CSF
DMG – Device Management CSF
DIS – Discovery CSF
GMG – Group Management CSF
LOC – Location CSF
NSE – Network Service Exposure, Service Execution and Triggering CSF
REG – Registration CSF
SEC – Security CSF
SCA – Service Charging and Accounting CSF
SUB – Subscription and Notification CSF
COMMUNICATION MANAGEMENT AND DELIVERY HANDLING

CMDH

• Supporting communication with other CSEs, AEs and NSEs
  o Need to access to provisioned delivery handling policies

• Allow to store requests

• Processing based on provisioned policies and delivery handling parameters

• Transparent data delivery

• Management through role based permissions
## COMMUNICATION PROTOCOLS

<table>
<thead>
<tr>
<th>IP based protocols</th>
<th>HTTP (TS-0008)</th>
<th>CoAP (TS-0009)</th>
<th>MQTT (TS-0010)</th>
</tr>
</thead>
<tbody>
<tr>
<td>Method is mapped to the oneM2M <em>operation</em> parameter</td>
<td>Request-URI is derived from the oneM2M <em>to</em> parameter</td>
<td>Support of blockwise transfers</td>
<td>Two scenarios for MQTT server location:</td>
</tr>
<tr>
<td>Status-Code and Reason-Phrase are derived from the oneM2M <em>responseStatusCode</em> parameter of the response primitive</td>
<td></td>
<td>Caching is supported using freshness and validity information carried with CoAP responses</td>
<td>MQTT server co-located within a node (MN, IN), and MQTT server located independently from nodes</td>
</tr>
<tr>
<td>This specification supports binding to HTTP 1.1</td>
<td>Due to limited code field in CoAP, some oneM2M response code shall be carried via CoAP payload field</td>
<td>The received resource representation shall be in plain XML, binary XML or JSON</td>
<td></td>
</tr>
</tbody>
</table>
DATA MANAGEMENT AND REPOSITORY

DMR

• Data storage and mediation

• Data aggregation
  o From one device or more.

• Data conversion into specific format
  o For analytics and semantic process.

• Data can be raw or processed by intermediate node
  o Application data, subscriber information, location information, device information, semantic information, communication status, access permission, etc.

• Support transfer of data to/from the AEs and other CSEs
  o Operation, create, read, update and delete.
  o Access based on defined policies.

• Provide a means to perform analytics on large amount of data

• Provide semantic information and enable functions for annotation
  o Exposing M2M resources based on semantic information.
  o Enable application to discover, interpret and M2M data from different sources.
DMG

• To manage the device capabilities on
  o Middle Nodes (M2M Gateways),
  o Application Service Nodes
  o Application Dedicated Nodes reside within an M2M Area network

• Utilize existing device management technologies
  o TR-069, OMA-DM, and LWM2M

• Management Adapter performs translation and adaptation between the DMG and the Management technology
  o Interacting with the management server over ms or la interfaces

• Device Management Resource Lifecycle
  o Includes resources management information to
    ◆ identify and access the Device Management Server
    ◆ Invoke the security mechanism of the Device Management Server
  o Define procedures to Create, Update and Delete resources
Re-using device management capabilities in oneM2M

<table>
<thead>
<tr>
<th>OMA (Open Mobile alliance)</th>
<th>creating standards for device management widely used in mobile networks.</th>
</tr>
</thead>
<tbody>
<tr>
<td>oneM2M-TS-0005</td>
<td>specifies the usage of OMA DM and OMA LWM2M resources</td>
</tr>
<tr>
<td></td>
<td></td>
</tr>
<tr>
<td>BBF (Broadband Forum)</td>
<td>creating standards for device management (TR-069) mainly used in fixed networks (home gateways, home devices).</td>
</tr>
<tr>
<td>TS-0006-Management_Enablement_(BBF)</td>
<td>specifies the usage of the BBF TR-069 protocol</td>
</tr>
</tbody>
</table>
DISCOVERY

**DIS**

- Provide discovery of information/resources residing in its own CSE
  - supported attributes and resources
  - Permission required

- Support various discovery methods through the use of "filter criteria"
  - keyword, identifiers, location, semantic information

- Response includes actual discovered information or resource address(es)

- Discovery is based on the capabilities defined through the device configuration function
GROUP MANAGEMENT

GMG

- Responsible for handling Group related requests.
- Bulk operations includes read, write, subscribe, notify, device management, etc.
- Access control is facilitated by grouping.
- May rely on broadcast and multicast network capabilities.

- Support subscriptions to individual Groups.
- Management of a Group and its membership
  - Allow to create, query, update, and delete a Group
  - Allow to retrieve the information (e.g. URI, metadata, etc.) of a Group and its associated members
  - Allow to add or remove members to and from a Group's member list
- Only M2M Applications or CSEs with common role shall be included in the same Group
  - role based access control
SECURITY

SEC

• **Sensitive Data Handling functionality**
  o Protects the local credentials on which security relies during storage and manipulation.
  o Several cryptographically separated security environments.

• **Security Administration functionality**
  o Creation and administration of dedicated security environment
  o Post-provisioning of a root credential
  o Provisioning and administration of subscriptions related to M2M services

• **Security Association Establishment functionality**
  o Between corresponding M2M nodes

• **Authorization and Access Control functionality**
  o According to provisioned security policies and assigned roles

• **Identity Protection Functionality**
  o Unique identifier
  o Support pseudonym identifier without linking to true identity
SUBSCRIPTION AND NOTIFICATION

SUB

• Manage subscriptions to resources, subject to access rights, and send corresponding notifications to the watcher address(s).

• Watcher address(s) represents the resource subscribers want to receive the notification.

• An AE subscribes to resources on a local or remote CSE.

• A CSE subscribes to resources on multiple CSEs.

• Subscription request includes
  • Subscription ID
  • Hosting CSE-ID
  • Subscribed-to resource addresses
  • Criteria (optional)
  • The address(s) for sending the notification

• Supporting single subscription for multiple resources
• **All entities in the oneM2M System are represented as resources**
  • Such as Applications, CSEs, data, etc.,

• **A resource structure is specified as a representation of the "resources"**
  • Resources are uniquely addressable.
  • Procedures for accessing such resources are also specified.

• Each resource is identified by a unique identifier (**URI**)
• Each resource has a number of defined attributes. Attributes can be of two types:
  • **Attribute**: meta-data that provides properties associated with a resource representation.
  • **Sub-Resource**: A resource that has a containment relationship with the addressed (parent) resource. The parent resource representation contains references to the children. The lifetime of the sub-resource is linked to the parent's resource lifetime.

• **Attributes can be:**
  • **RW**: read/write by client
  • **RO**: Read-Only by client, set by the server
  • **WO**: Write-once, can be provided at creation, but cannot be changed later
All entities in the oneM2M System, such as AEs, CSEs, data, etc. are represented as resources.

RESTful access

oneM2M identifies three categories of resources:

- **Normal resources**
  include the complete set of representations of data which constitutes the base of the information to be managed.

- **Virtual resources**
  used to trigger processing and/or retrieve results, but they do not have a permanent representation in a CSE.

- **Announced resources**
  a resource at a remote CSE that is linked to the original resource that has been announced, and it maintains some of the characteristics of the original resource.

One can now have:
/cse01/app01/container02
/container_63eb90a2
1. Reference points expose the M2M resource tree in a RESTful way
   • Representational State Transfer

2. Transfer of representations of uniquely addressable resources.

3. Not a protocol in itself – more of an access pattern
   1. Though easily mapped to existing protocols, e.g. HTTP, CoAP, …

4. Classic REST verbs:
   • CREATE: Create child resources.
   • RETRIEVE: Read the content of the resource.
   • UPDATE: Write the content of the resource.
   • DELETE: Delete the resource.

5. Extended REST verb:
   • NOTIFY: Indicate a change of a resource for a subscription. Variant of RETRIEVE/UPDATE
• Resources and associated attributes are uniquely addressable via Universal Resource Identifiers (URI)
• Three methods for addressing the oneM2M resource structure
  • Hierarchical URI Method
    using the actual path portion of the URI defining the entire relationship for the target resource within the resource structure.
  • Non-Hierarchical URI Method
    using the resource identifier given by its hosting CSE during the resource creation procedure.
  • ID Base Method
    using the CSE-ID of the host where the resource is located, and/or the AE-ID of the interacting/addressed AE
Content is stored in OneM2M CSEs resource tree as content instances
Data that is pushed by sensors can be subscribed by other applications
The content of the data may not known by these applications

In release 1 there are limited possibilities to indicate the proper meaning:
• Labels can be used to mark containers or AEs
• Subscribing AEs can guess the meaning of the provided data
• Also the content is not specified, using proper data format can help to interpret the data - > example SenML
• <AE>, <container> and <contentInstance> can be linked via ontologyRef, e.g. http://purl.oclc.org/NET/ssnx/ssn#SensingDevice

In release 2 the new resource <semanticDescriptor> was added
• Allows to add a complete semantic subscription as child resource
• Allows to provide SPARQL interface to discover resources based on complex queries
• OneM2M Base Ontology for annotating data
REL 2 - ENHANCED SECURITY

- Release 1 defined Authentication and Authorization in a static manner
  - Authentication via Certificates or PSKs
  - Needed to be distributed and configured before
  - Authorization via `<AccessControlPolicy>`
    - Other resources can link to ACP
    - Stores rules for allowed combinations of user and operation for the linked-to resource
  - ACPs needed to be created before and updated when needed

- Release 2 addressed two issues
  - Enhanced management of Certificates
  - Introduction of Dynamic Authorization Server
    - External Component that can be integrated into existing policy engines
    - If no proper rule for access granting is available, the DAS can be asked
    - Result will be stored in ACP -> caching the result
  - Eases the access management
In release 1 requests/responses and the contained data is only secured between AEs and CSEs
  • In a multi-hop scenario where data is transferred from one AE to another all CSE in between have the requests/responses unprotected
  • AEs have to trust CSEs

Release 2 adds a Trust Enabler Function
  • External component
  • Sending AE can encrypt request and contained data
  • Receiving AE can receive request and decrypt it
  • No need to trust transit CSEs
• **<flexContainer> resources**

1. Specific `<container>` that allow to store specific content directly in `<container>` instead of `<contentInstance>`

• **Time Series Data**
  - AEs can send data in periodic manner
  - Other AEs can subscribe to data
  - CSE will notify subscribed AE if data is missing in a specific interval

• **WebSocket Transport Binding**
  - Fourth transport protocol binding
  - Bidirectional, client don’t need extra server for receiving notifications

• **oneM2M App-ID registry**

• **New shortcut “.” to <CSEBase>**
• Several enhancements to
  • Request and Response Handling
  • Interworking for AllJoyn, LWM2M and others
  • Underlying network and Device Triggering
  • Resource Discovery and Retrieve Operations
  • Notification Handling
  • Identifier Formatting

• oneM2M Testing Framework
  • Interop tests
  • Conformance Testing

• App Developer Guide
AGENDA

1. Motivation and Challenges
2. oneM2M
3. OpenMTC: Open Machine-Type Communication
4. Summary and Future Outlook
OPENMTC – HORIZONTAL M2M/IOT PLATFORM

- Reference implementation of oneM2M standard
- Rich history in international R&D projects
- Interconnects various sensors and actuators from multiple application domains
- Independent of underlying hardware or network infrastructure
- Licensed under Eclipse Public License v1.0 (EPL1.0)

www.openmtc.org
OSS RELEASE FEATURES

Resources
- CSEBase
- remoteCSE
- AE
- container
- contentInstance
- subscription
- AccessControlPolicy

Transports
- HTTP(S)
- MQTT

Serialization
- JSON

Database
- In-memory

AE SDK
- to develop apps
- examples for AE and IPE

Development and Deployment
- instances can be run from:
  - IDE
  - Python distribution
  - Docker

Protocol Adapter
- FS20 incl. simulation mode
- NGSI-9/NGSI-10 (FIWARE)
M2M INTERWORKING PROXY (IPE)

Functionality (schema)

- IPE registers to the gateway/backend
- Initiates structure (e.g. devices container)

- Handles devices autonomous or preconfigured
- Generates structure per device
  - “sensor_data” container for storing measurements
  - “device_data” container for storing things like battery
  - “commands” container to receive commands from applications

- Creates virtual entities of the devices
ZigBee example (sketch)

- Handles devices connected via ZigBee
- Two different device types supported
  - Power plug that measures power characteristics and can be switched ON and OFF
  - Multisensor that measures environmental values
- Power plugs using command container
- Applications can push data to this container in order to initiate a change of the state
- IPE will parse and send the proper command to the plug
ZigBee example (structure)

- Discovery of container structure
  - "onem2m/ZigBeeIPE/devices",
  - "onem2m/ZigBeeIPE/devices/ZBS121005544",
  - "onem2m/ZigBeeIPE/devices/ZBS121005544/device_data",
  - "onem2m/ZigBeeIPE/devices/ZBS121005544/device_data/battery_voltage",
  - "onem2m/ZigBeeIPE/devices/ZBS121005544/device_data/battery_status",
  - "onem2m/ZigBeeIPE/devices/ZBS121005544/sensor_data",
  - "onem2m/ZigBeeIPE/devices/ZBS121005544/sensor_data/brightness",
  - "onem2m/ZigBeeIPE/devices/ZBS121005544/sensor_data/temperature",
  - "onem2m/ZigBeeIPE/devices/ZBS121005544/sensor_data/humidity",
  - "onem2m/ZigBeeIPE/devices/ZBS121005544/sensor_data/pressure",
  - "onem2m/ZigBeeIPE/devices/ZBS121005544/sensor_data/movement"

- ZigBeeIPE is the IPE itself
- ZBS121005544 is a Multisensor
  - “device_data” contains battery voltage and status
  - “sensor_data” contains all the different types of measurements
ZigBee example (data format) – based on SenML

- Example latest brightness
  ```json
  [{"bn": "urn:dev:xbee:ZBS122S000001", "v": 923.0, "u": "lx",
   "t": "1494417208.508", "n": "brightness"}]
  ```

- Example of latest temperature
  ```json
  [{"bn": "urn:dev:xbee:ZBS122S000001", "v": 21.7, "u": "Cel",
   "t": "1494417113.730", "n": "temperature"}]
  ```

- Example of latest battery voltage
  ```json
  [{"bn": "urn:dev:xbee:ZBS122S000001", "v": 3.6, "u": "V",
   "t": "1494417259.219", "n": "battery_voltage"}]
  ```

- “bn” names the device that was measuring the observation
- “t” is Unix timestamp
- “n” is container name and type of the phenomenon
- “u” indicates the given unit
- “v” is the value given as floating-point number
**M2M INTERWORKING PROXY (IPE) – EXAMPLE USE CASE**

1. Registers itself
2. Discovers resources
3. Subscribes to data
4. Visualizes data
5. Supports actuating

---

**Dashboard IN-AE-1**

**IN-CSE-1**

**MN-CSE-1**

**MN-CSE-2**

**MN-CSE-2**

---

**ZigBee IPE MN-AE-1**

1. Registers
2. Creates structure
3. Pushes sensor data
4. Subscribes to actuator data

**ZigBee IPE MN-AE-2**

**FS20 IPE MN-AE-3**

---

**Multisensor**

**Plugs**

**Multisensor**

**Door/Window**

**Motion**

---

**Fraunhofer FOKUS**
STAKEHOLDERS - WHO CAN USE OpenMTC?

Research Institutions & Universities
- R&D on real M2M network
- Innovating new concept and algorithms

Applications Developers
- Validating M2M applications
- End-to-End M2M connectivity

Manufacturers
- Validate their products end-to-end
- Looking for the missing pieces

Operators
- Be prepared for M2M/IoT mass fixed & devices
- Validate and evaluate M2M services
AGENDA

1. Motivation and Challenges

2. oneM2M

3. OpenMTC: Open Machine-Type Communication

4. Summary and Future Outlook
1. **Open Source release** of the oneM2M platform **OpenMTC**!

2. Deterministic **Time-sensitive Networking** (TSN) – new standard factory connectivity!

3. **Narrowband IoT** – low-cost, high-penetration / wide area IoT connectivity is ready!

4. **Edge Intelligence / Fog & Edge Computing** for Industry 4.0 and Smart Cities!
   - Fog-/Edge-based **Public Surveillance** (Smart City)
   - Fog-/Edge-based **Condition Monitoring & Predictive Maintenance** (Industry 4.0)

5. Platform Industrie 4.0 Asset Administration Shell (AAS) in action!

6. Fog-/Edge-based Retrofitting and Condition Monitoring (OPC UA, MEMS)
OPENMTC PAST PRESENCE AND FUTURE

<table>
<thead>
<tr>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>OpenMTC Rel.1 (4/12)</td>
<td>OpenMTC Rel.2 (8/13)</td>
<td>OpenMTC Rel.3 (4/14)</td>
<td>OpenMTC Rel.4 (4/15)</td>
<td>OpenIoTFog Rel.1 (6/16)</td>
<td>OpenIoTFog Rel.2 (1/18)</td>
</tr>
<tr>
<td>- Generic Communication</td>
<td>- Integration with core network</td>
<td>- Cloud-based Platform, horizontal Scalability</td>
<td>- OneM2M alignment</td>
<td>- Container-based Framework</td>
<td>- TSN, NB-IoT, LoRa Support</td>
</tr>
<tr>
<td>- Remote Management</td>
<td>- Reachability extensions</td>
<td>- CoAP, Diameter and WebSocket support</td>
<td>- CoAP-DTLS support</td>
<td>- I4.0 Asset Administration</td>
<td>- OPC UA, AAS communication</td>
</tr>
<tr>
<td>- Application Enablement</td>
<td>- Applications Extensions</td>
<td>- Polyglot: Java, Python, Ruby, JavaScript</td>
<td>- Android gateway</td>
<td>- Edge Analytics Enablement</td>
<td>- Full Orchestration support</td>
</tr>
<tr>
<td>- Service Subscription Profiles</td>
<td>- MQTT Support</td>
<td>- Cloud-based Platform, horizontal Scalability</td>
<td>- CoAP-DTLS support</td>
<td>- Fog-to-Fog Communication</td>
<td>- Condition Monitoring, Predictive M</td>
</tr>
</tbody>
</table>

Upcoming OpenMTC Releases
- FlexContainer Resource
- Group and Node Resources
- CoAP(s) & MQTT
- Pub/Sub improvements
- Service Subscription Profiles

New R&D projects
Industrial Fog Nodes bring Intelligence to the Device/Edge for

- Plug & Play Sensor Data Aggregation
- PLC Interworking (OPC UA)
- Stream & Video Analytics, Machine Learning
- Rule-based Actuator Control
- Condition Monitoring, Optimization of Overall Equipment Effectiveness (OEE)
- Predictive Maintenance

Benefits:

- Ultra-Low Latency, Configurable Security, Robustness, Autonomy and Interoperability
- Retrofitting with various Sensor Technologies & Networks
- Semantic Sensor Data Enrichment

www.openiotfog.org
• Standard-based **Asset Administration Shell** for seamless integration and M2M communication of Shop Floor Assets

• Semantic **Harmonization of Shop Floor Data**

• **Time Sensitive Networking** (TSN), fieldbus systems and robust SDN-based industrial wireless supported

• **Soft-PLC Readiness**

---

1. **Plug & Play** (existing devices): PLCs, Sensors, Actuators, Robots, Machinery, Drives, etc.
2. **Collect Data** (wired/wirelessly): Wi-Fi, BLE, FS20, ZigBee, LoRa, Industrial Wireless PROFINET, Modbus/TCP, TSN
3. **Enhance Data** (semantically): OWL, RDF, XML, JSON-LD, SPARQL, etc.
4. **Analyze Data** (in the fog node): Powerful CEP for Safety, Predictive Maintenance, Event-driven Actuation
5. **Secure Data** (on different layers): Preprocessing, Anonymization, VPN Tunneling, Filtering, etc.
6. **Provide Data** (standard conform): oneM2M, OPC UA, HTTP, MQTT, CoAP

---

www.openiotfog.org
• **Asset Administration Shell** for
  - Plug & Produce integration of Shop-Floor Assets
  - Peer-to-Peer negotiation of manufacturing tasks

• **OneM2M-Architecture**
  - container-based application injection of microservices
  - vertical and horizontal (Peer-to-Peer) communication modes
  - TSN Network Provisioning via Mcn Interface (Prioritization, Multi-Path, etc.)
  - integrated health monitoring

www.openiotfog.org
OpenMTC Open Source Release

- Backend + Gateway
- SDK
- CRUD operations
- Protocols
  - HTTP(S)
  - MQTT (3rd party broker required)
- Protocol Adapter
  - Cul868(FS20) for Smart Building
  - NGSI-9/NGSI-10 (FIWARE)
- Security
  - TLS (Certificate-based)
- JSON serialization
- Storage (in-memory)
- Docker support
- Basic Dashboard

Addons, Plugins

- Protocol Adapters (IPE)
  - ZigBee for Smart Building - Sensors and Actuators (Power, Temp, Motion, Brightness, Humidity, Air Pressure, Switches)
  - Industrial Fieldbus Support (Profinet and ModBus) for Industry 4.0
  - Robot Operating System Support (ROS)
- Semantic Annotation
- Historical Data Handling
- IoT Fog-/Edge Analytics
  - Complex Event Processing (CEP) including classification and regression models, decision trees, clustering, statistical analysis
  - Machine Learning providing means and services for model training,
  - Video Analytics including object, motion detection, tracking, counting & diagnosis, 3D video analysis
- Load Generator for creating virtual resources
- Dynamic Authorization Server (DAS)
- Advanced Dashboard
- Asset Administration Shell Support
SUPPORT, TRAINING, DEVELOPMENT AND TESTING SERVICES

- Want to connect your own / a special device? We can help!
- Want to develop a complex IoT application? We can help!
- Want to setup a fully fledged IoT / I4.0 testbed? Talk to us!
- Training/Tutorials for M2M/IoT, OpenMTC and oneM2M
  - IoT Security – ½ day
  - IoT Testing – ½ day
  - IoT Analytics – ½ day
  - M2M standards, the oneM2M standard, Architecture, Application Entities, Common Service Entities, Network Service Entities, Interworking Proxy Entities, etc. – ½ day
  - Installation, administration, and configuration of the OpenMTC – ½ day
  - OpenMTC Working with Network, Data and Device APIs and Java and Python SDKs – ½ day
  - OpenMTC Resource Trees, Communication Establishment, Device Management, Security – ½ day
  - OpenMTC Hands-on Exercises: Application development, Device integration, etc. – full day

© Fraunhofer FOKUS
Dipl.-Ing. Ronald Steinke
Software Developer (IIoT Center)
Tel.: +49 30 3463-7157
Email: ronald.steinke@fokus.fraunhofer.de

Fraunhofer Institute for Open Communication Systems FOKUS
Kaiserin-Augusta-Allee 31
10589 Berlin, Germany