

# MindConnect API Training

## Agent Development with MindConnect API

# MindSphere Architecture & API

# Architecture & API

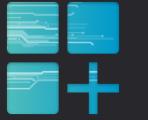
## Learning Goals

**How is MindSphere designed  
&  
how shall we develop  
applications for  
MindSphere?**

- MindSphere Architecture
- MindSphere APIs
- MindAccess Developer



# MindSphere Portfolio Overview



## MindApps

Powerful industry applications and digital services for asset transparency and analytical insights



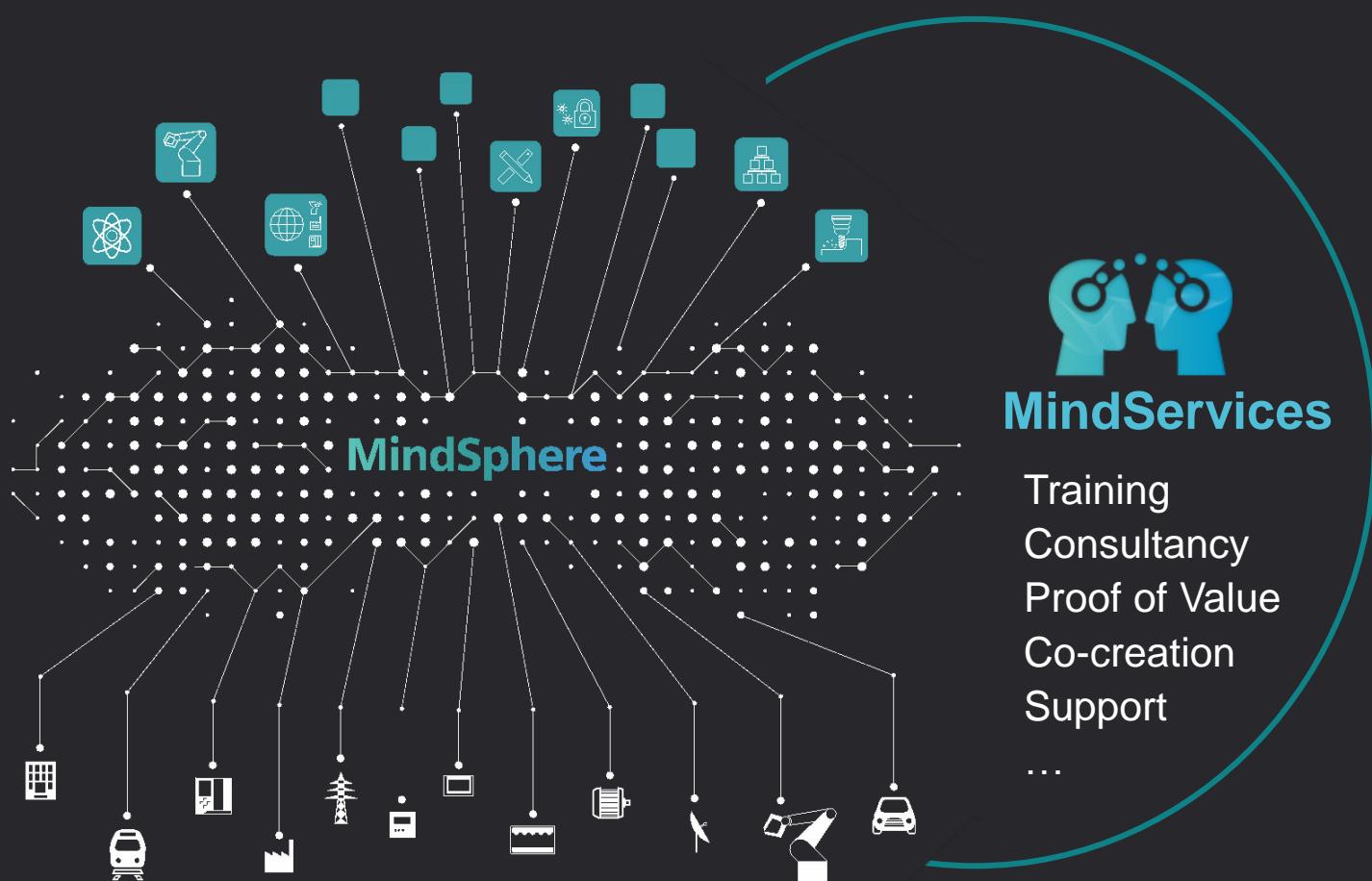
## MindSphere & MindAccess

Open Platform as a Service (PaaS) for scalable, global IoT connectivity and application development



## MindConnect

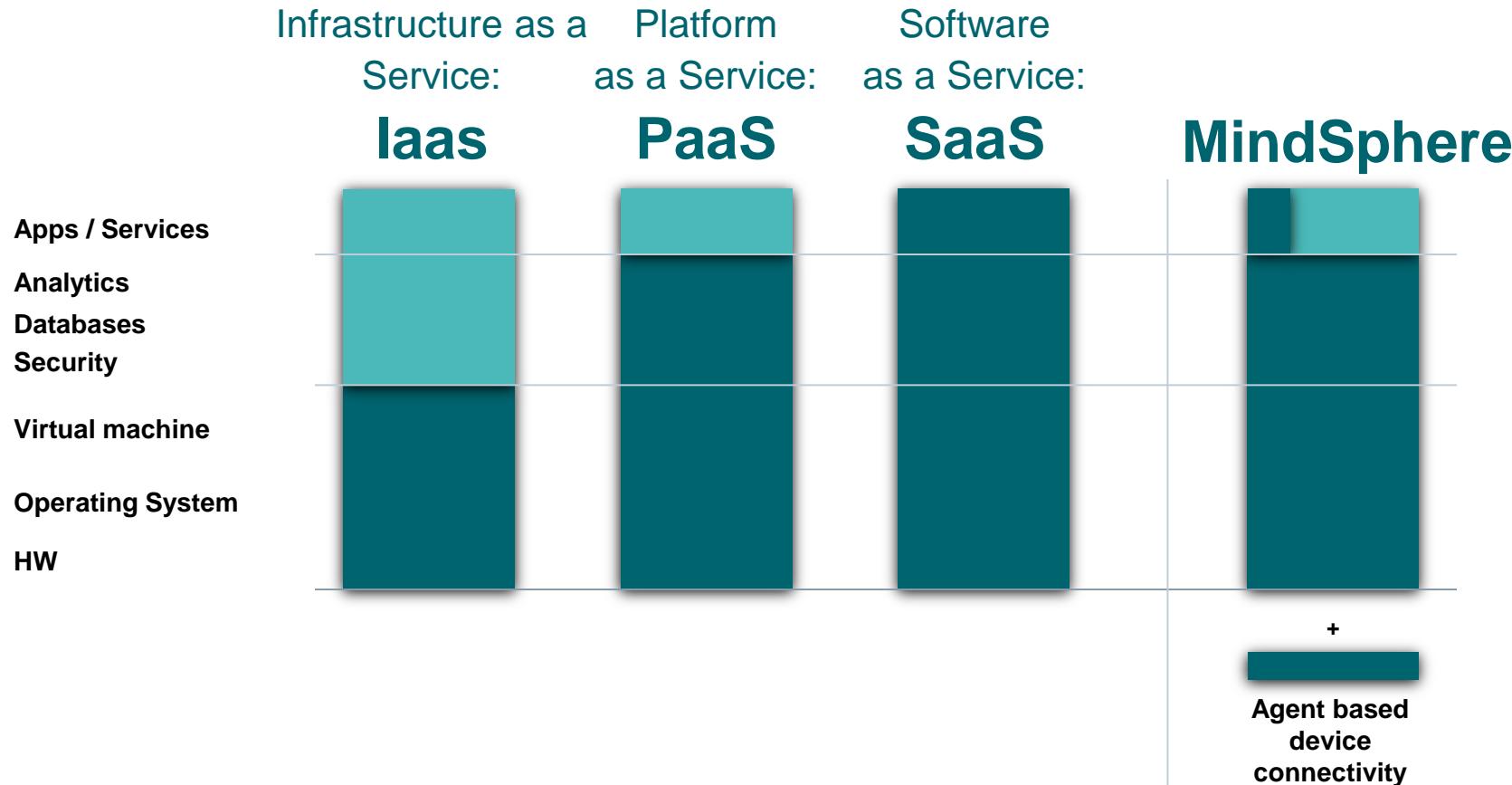
Secure plug and play connection of Siemens and third-party products and equipment

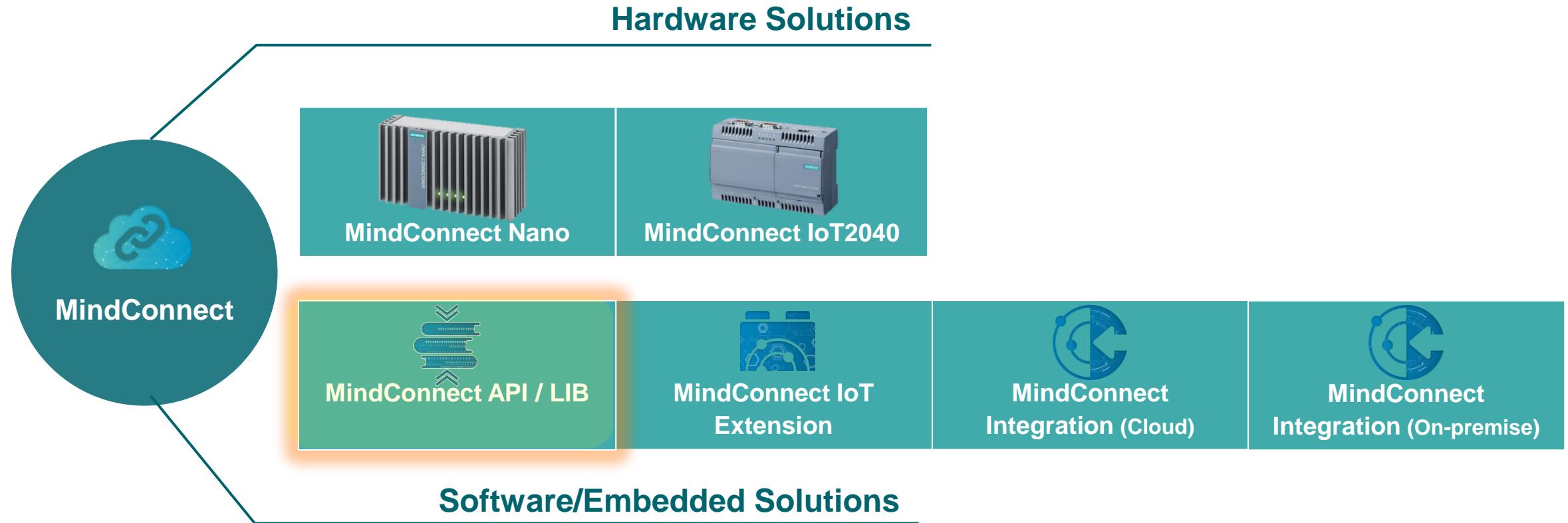


## MindServices

Training  
Consultancy  
Proof of Value  
Co-creation  
Support  
...

# Service Models

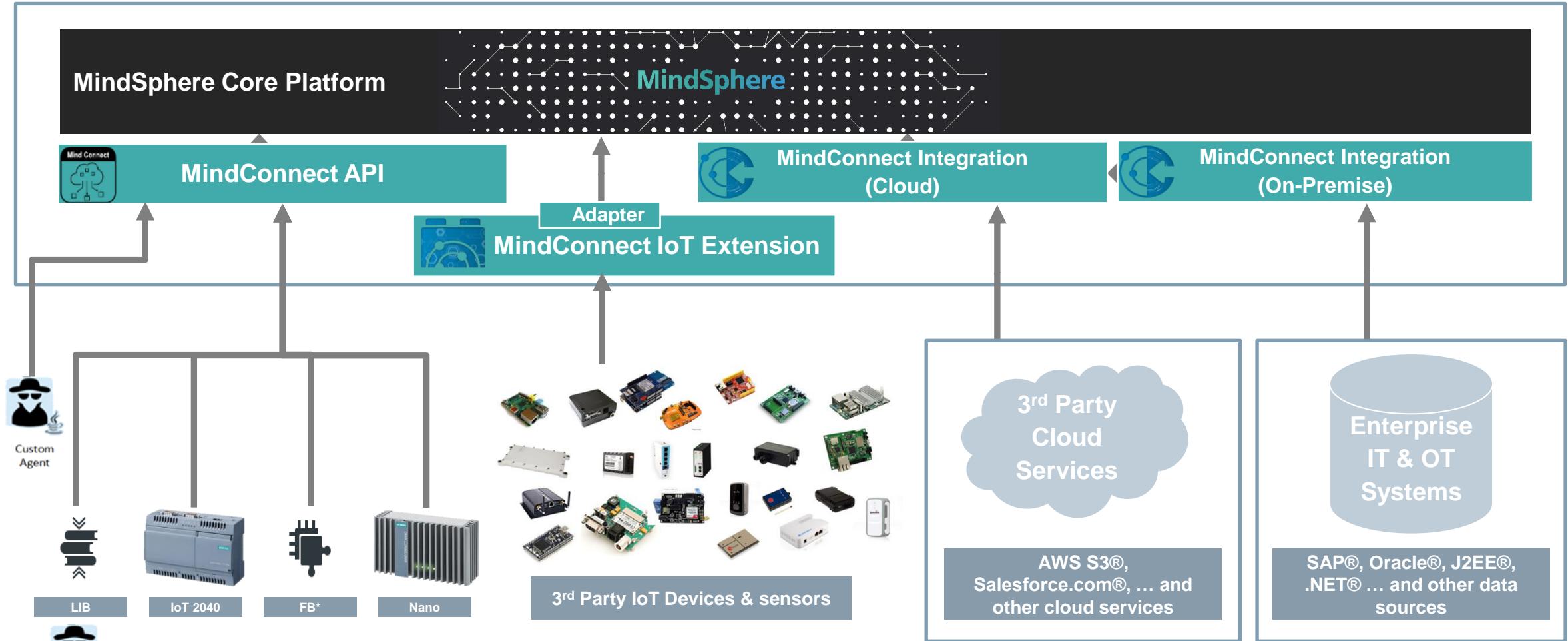




# MindSphere Architecture and Connectivity Portfolio

## *Big Picture Connectivity and MindConnect Integration*

**SIEMENS**  
Ingenuity for life



# Structure of MindSphere Connectivity

MindConnect Nano/ IoT2040

**SIEMENS**  
Ingenuity for life

MindSphere



To use OPC UA together with MindSphere, you need a S7-1500 V2.x or an OPC UA server.

MindConnect  
Elements

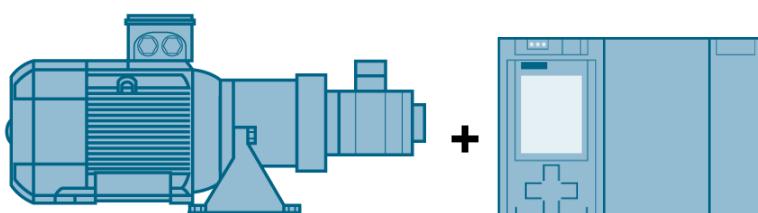
MindConnect Nano

OR

MindConnect IoT2040



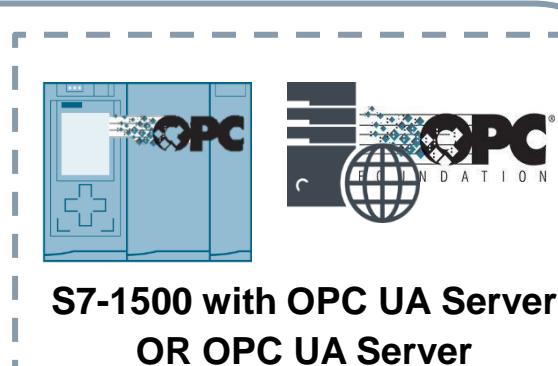
Asset



Asset

S7

OR

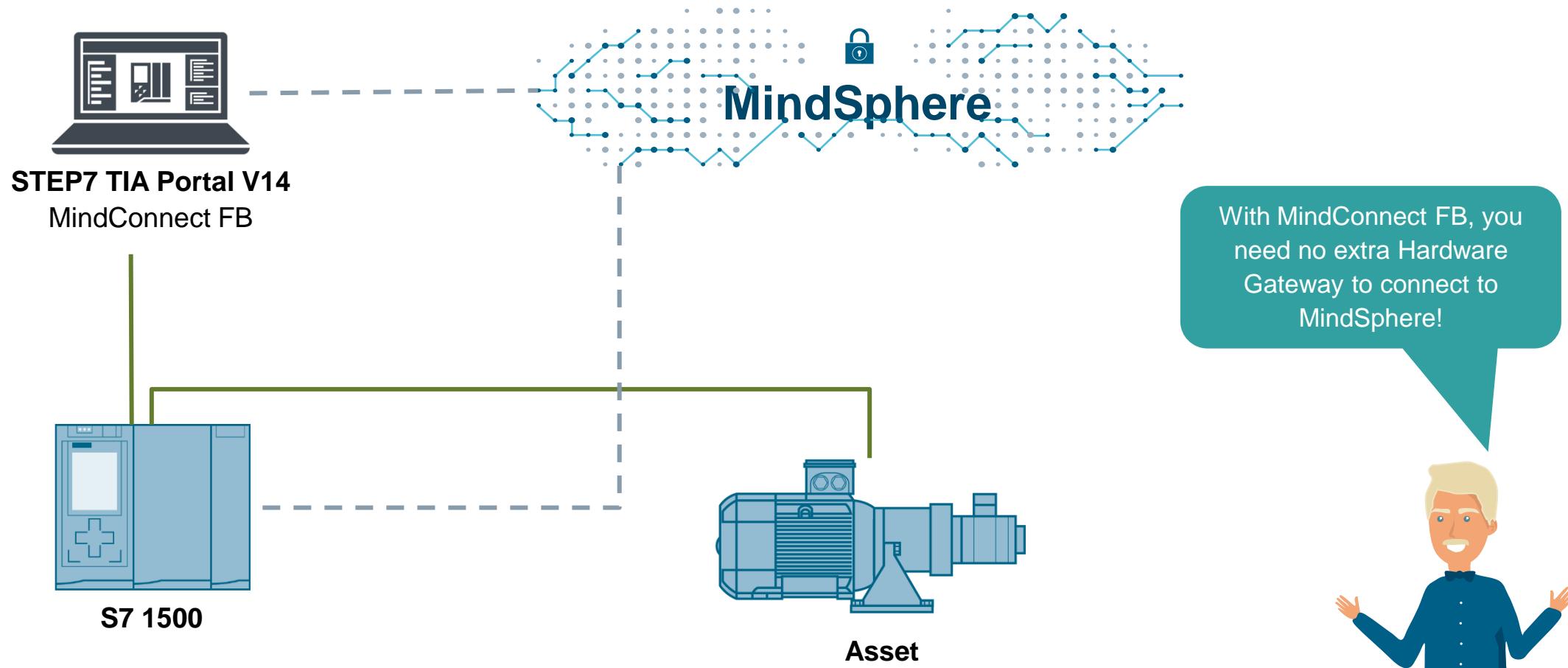


With TIA Portal V14

# Structure of MindSphere Connectivity

MindConnect FB 1500

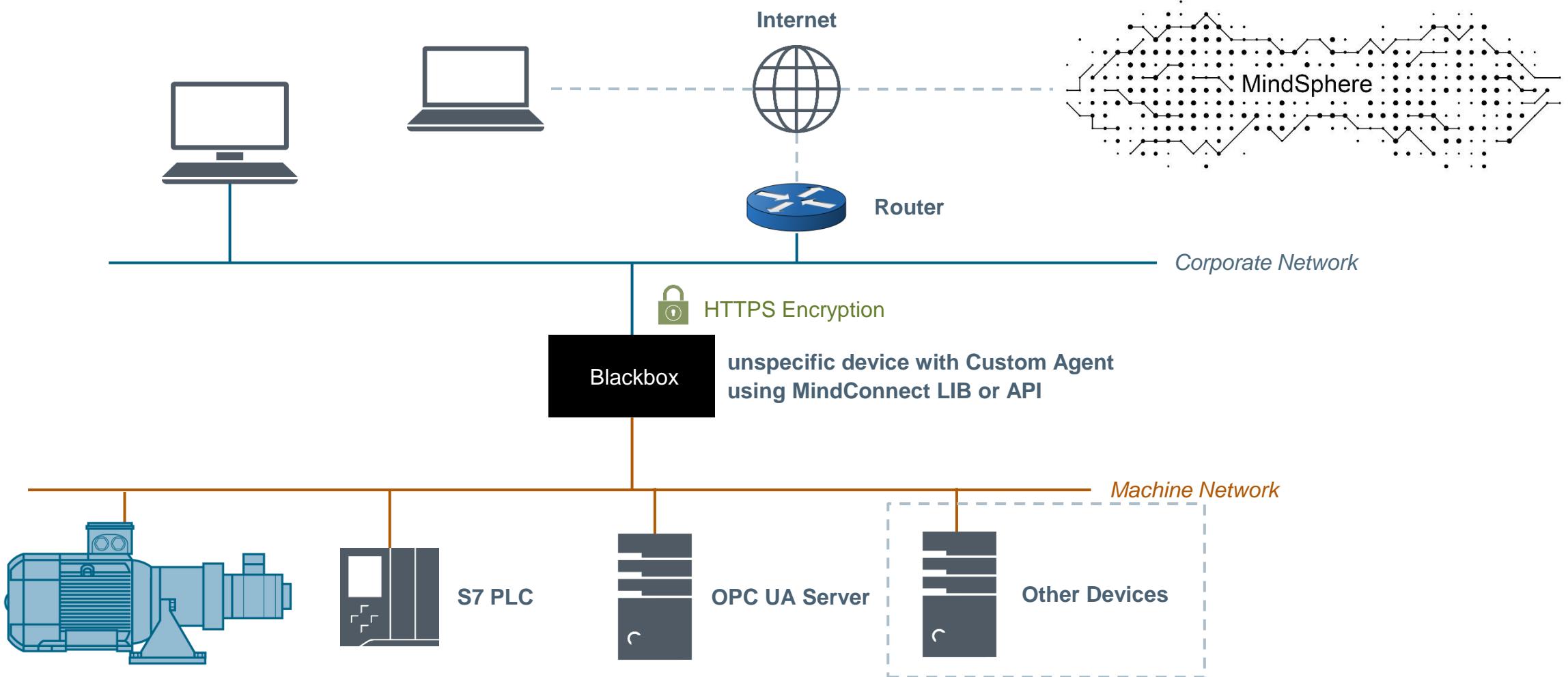
**SIEMENS**  
Ingenuity for life



# Custom Agent Overview

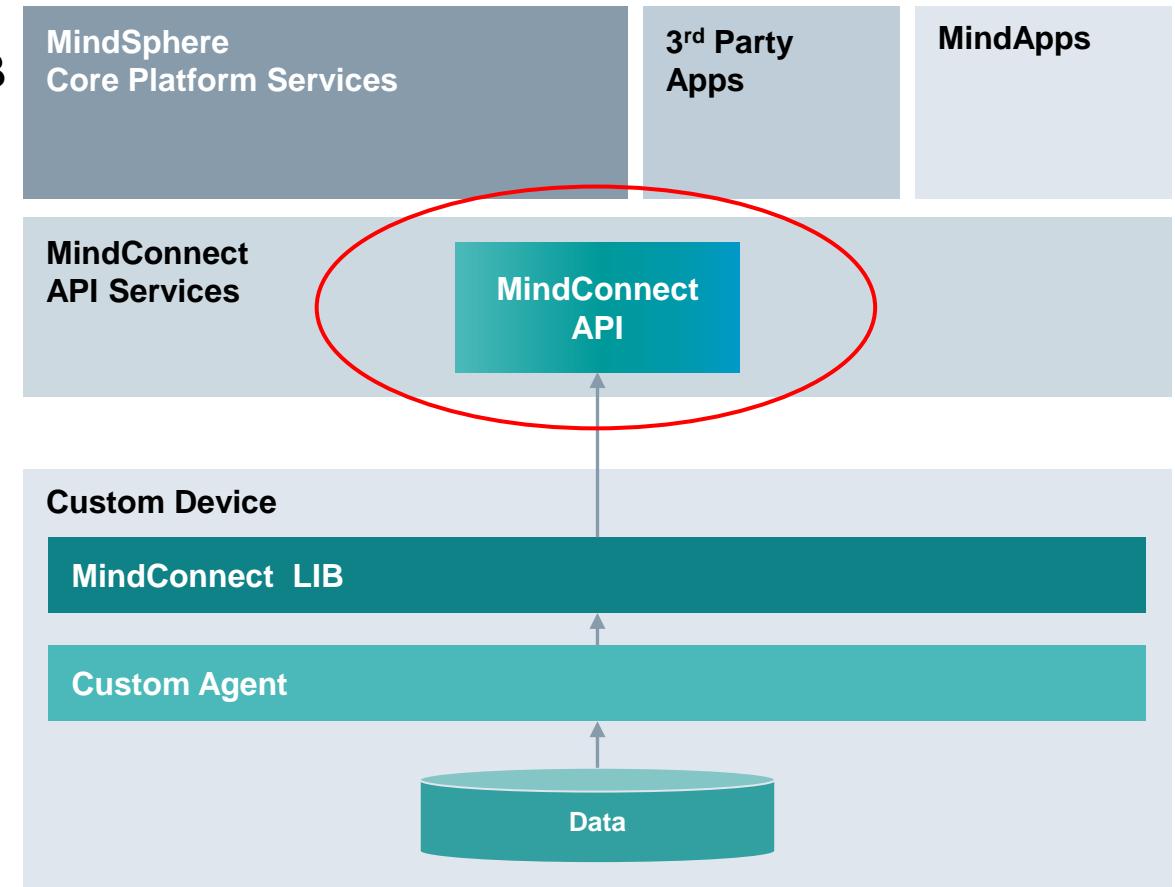
## MindSphere topology using Custom Agent

**SIEMENS**  
Ingenuity for life



# MindConnect LIB (details for developers)

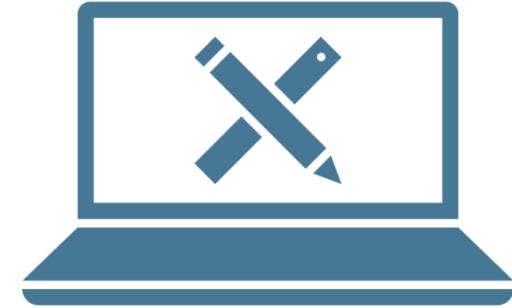
- Software Library written in C
- Small footprint (~170KB, ~64KB Ram) + LIBs ~780KB
- Integrates into any device: (ARM/MIPS, Windows, Linux)
- Enables secure connectivity (e.g. authentication, encryption)
- Firewall friendly
- File transfer (Max. 2 MB/f upload, max. 100 MB/f download)
- Local buffering
- Logical workflows to optimize data collection and upload
- Custom data parser



### API: Application Programming Interface

- A set of subroutines, protocols and tools to build application software  
= Building blocks that make the application development fast and easy
- MindSphere 3.0:
  - 18 API optimized for industrial IoT application development
  - Long term stability for min. 2 years to fulfill industrial needs (long product lifecycles)
  - API versioning and backward compatibility
  - Continuous demand-based functionality extension

Develop your own applications  
running on MindSphere



## Description

### Aggregate Service

Read aggregated values with three durations: 2 minutes, 1 hour, and 1 day



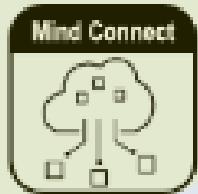
### Dynamic Data Service

Retrieve collected data for an asset as an array of time stamp and value tuples



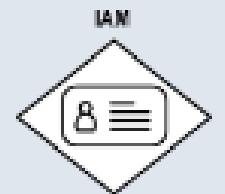
### MindConnect API

Develop custom agents and connect your device or application to MindSphere



### Identity and Access Management

Manage your customers (e.g. add, modify) within your own tenants only



### Agent Management

Create, edit or remove MindConnect elements, onboard and off board agents and set relations to assets



### Asset Management

Represent physical assets from your site in MindSphere. Use models and create instances, set relations to others and create structures such as hierarchies



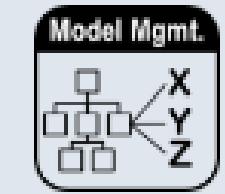
### Event Management

Manage standardized and customized events. Acquire events from the field & other applications



### Model Management

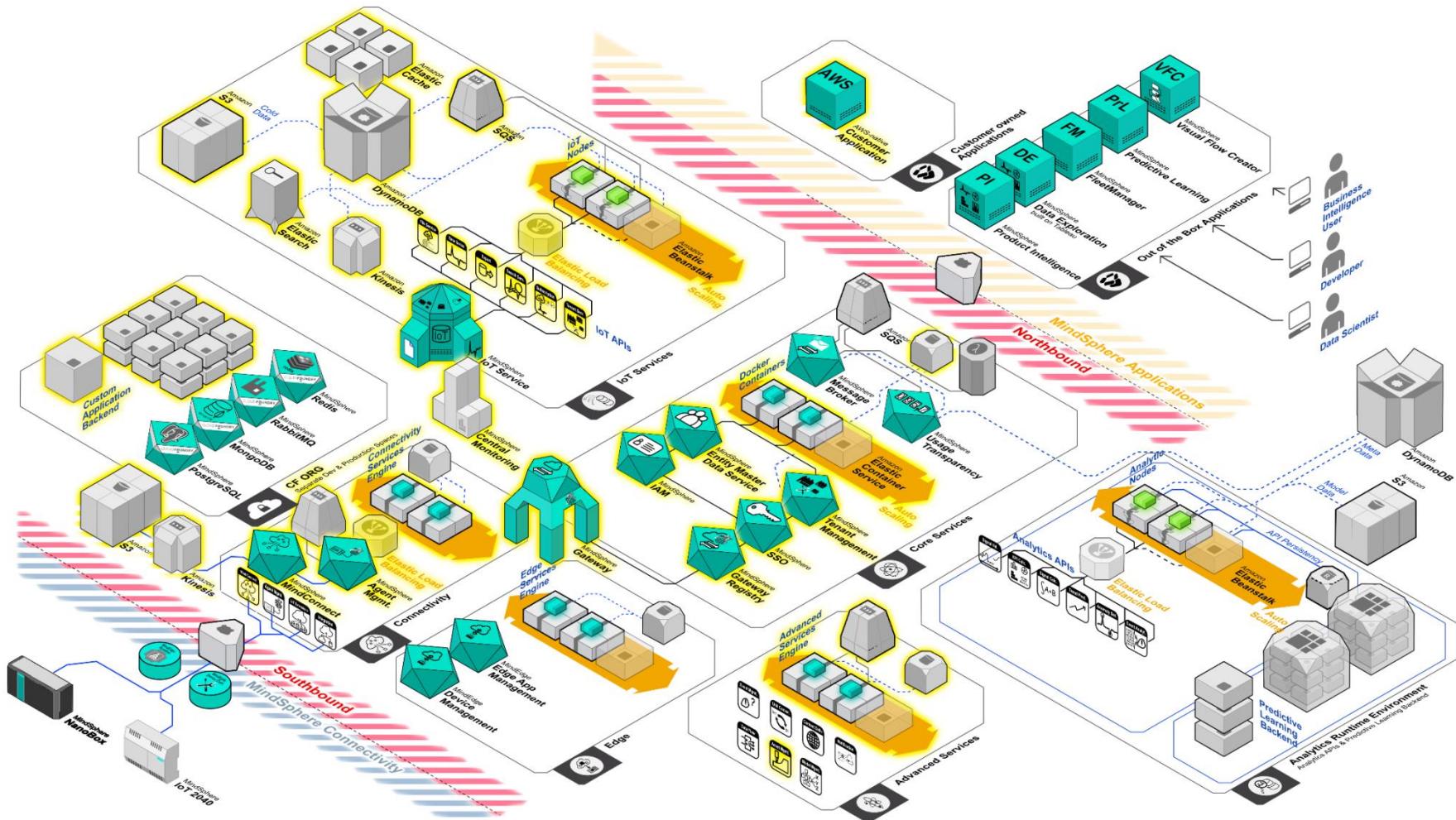
Define your data model and add standardized definitions for a better understanding and usage of your data



# MindSphere Overview

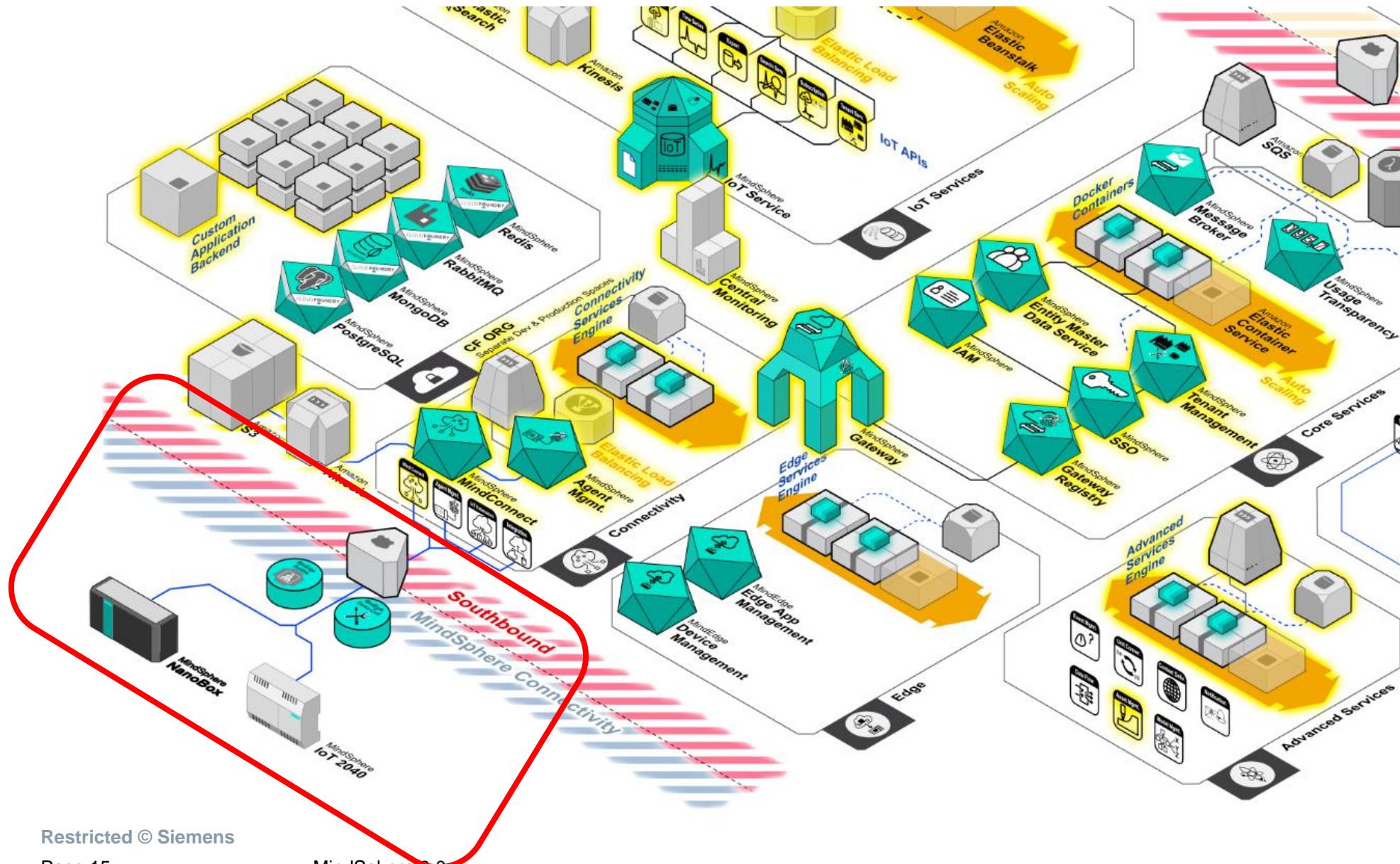
# SIEMENS

Ingenuity for life



# MindSphere Connectivity

**SIEMENS**  
Ingenuity for life



MindSphere on AWS uses an **API** first approach for **early definition** and **stabilization** of APIs

By this way, MindSphere APIs offer:

- **Continuous** demand-based **functionality extension**
- **Long term stability** for min. 2 years to fulfill industrial needs (long product lifecycles),  
**API versioning** and **backward compatibility**



Customers get early access to API specifications

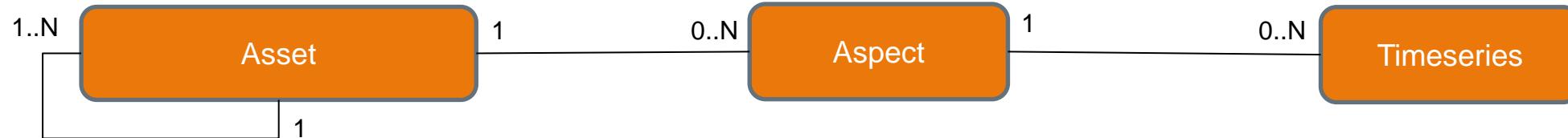
Mocking of APIs enables parallel development for faster delivery

# MindSphere 3.0 IoT Model

## Asset-Level

**SIEMENS**  
Ingenuity for life

Asset	Aspect	Datapoint / Timeseries
<ul style="list-style-type: none"><li>“Asset” is a logical representation of a machine or any device that can provide data to the system</li><li>Hierarchical relations possible</li><li>Data sources, e.g. MindConnect Elements, are mapped to Assets</li></ul>	<ul style="list-style-type: none"><li>“Aspect” contains summarized data points from data sources for a specific evaluations</li><li>Group of datapoints</li><li>Aspect can define multiple variables with pre-defined units / data-types</li></ul>	<ul style="list-style-type: none"><li>Contains historical data from assets as defined by aspects</li><li>Timestamp format: ISO 8601</li></ul>



# Asset Manager

The Asset Manager offers the means to configure Assets, Asset type and Aspect type

# MindSphere 3.0

## Asset Manager



MindSphere Asset Manager

The screenshot shows the 'Basic Settings' for an asset named 'casseoe'. The left sidebar has tabs for 'ASSETS', 'ASSET TYPES', and 'ASPECT TYPES'. The 'ASSETS' tab is selected, showing a list of assets including 'AC\_Location', 'Connectivity\_Training', 'Democase\_Location', 'MCN\_Test1', 'TBAsset1', 'Test\_Asset\_AC', 'Training\_room', and 'UP\_Location'. The main panel displays the 'Basic Settings' for the selected asset 'casseoe'. The asset type is listed as 'BasicEnterprise core.basicenterprise'. The description is 'Root Asset for casseoe tenant'. There are fields for Country, Region, Location, Postal code, Street, Longitude, and Latitude, all of which are currently empty. At the bottom right of the main panel are 'Edit' and 'New' buttons.

**IoT Data Modeler is replaced by Asset Manager.**

- **Asset:** assets can be created, onboarded, modified, cloned, moved, deleted or offboarded.
- **Assets Types:** administration of Assets Types/Templates.
- **Aspect Types:** administration of Aspect Types/Templates.

### Assets

Creates an asset in the three main topics: device types, agent types and hierarchy types. You can configure, read and manage an asset or the root asset. An asset is accessed by its ID.

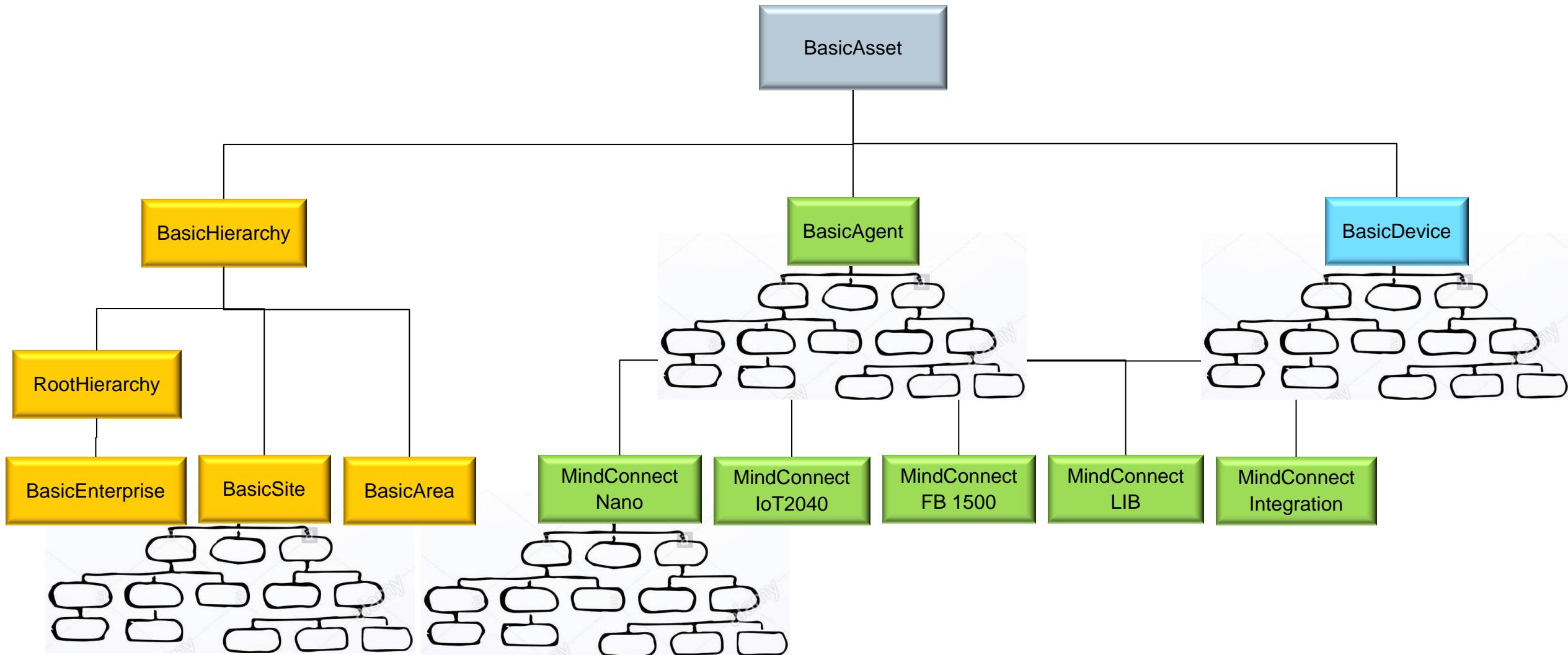
### Aspects

You can manage all aspects of an asset. An aspect is the representation of the data sources of an agent. The measured time series data are stored for aspects. An aspect can contain multiple variables.

# MindSphere 3.0

## Asset and Hierarchy

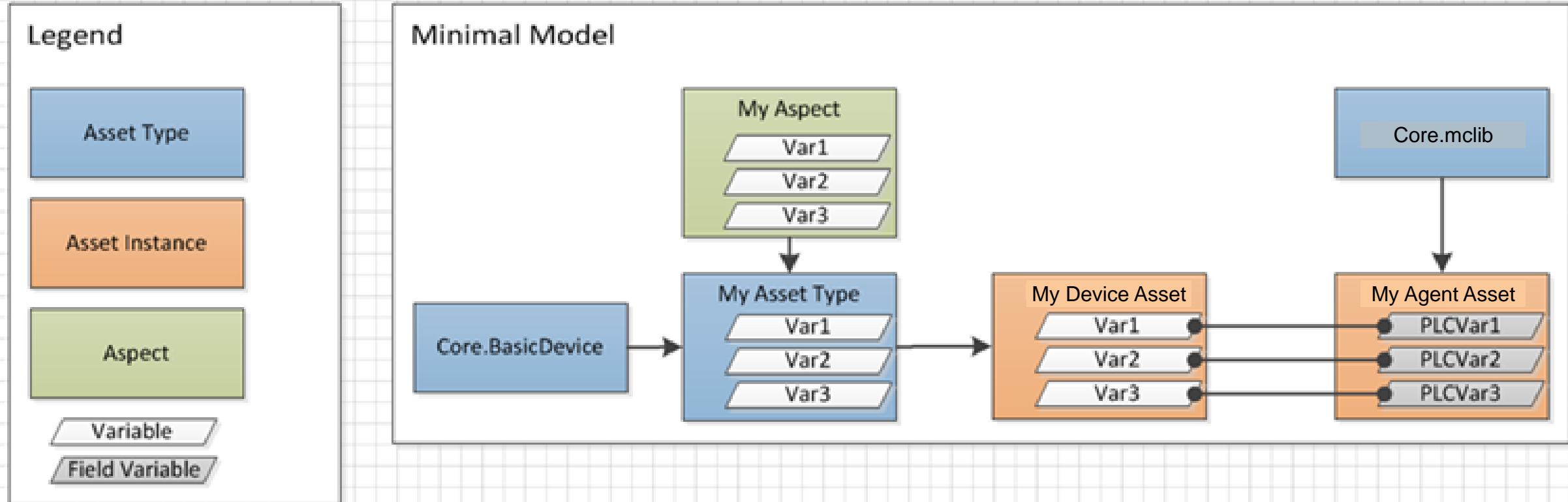
**SIEMENS**  
Ingenuity for life



# MindSphere 3.0

## Asset and Aspect

**SIEMENS**  
Ingenuity for life



T

# Fleet Management

**Always up to date**

The key data for your connected components and systems

# MindSphere 3.0

## Fleet Manager

**SIEMENS**  
Ingenuity for life

Transparency of  
installed base

Analysis for each asset cluster of  
multiple data sources

Request function, defining actions  
after an occurrence of an event

Map View

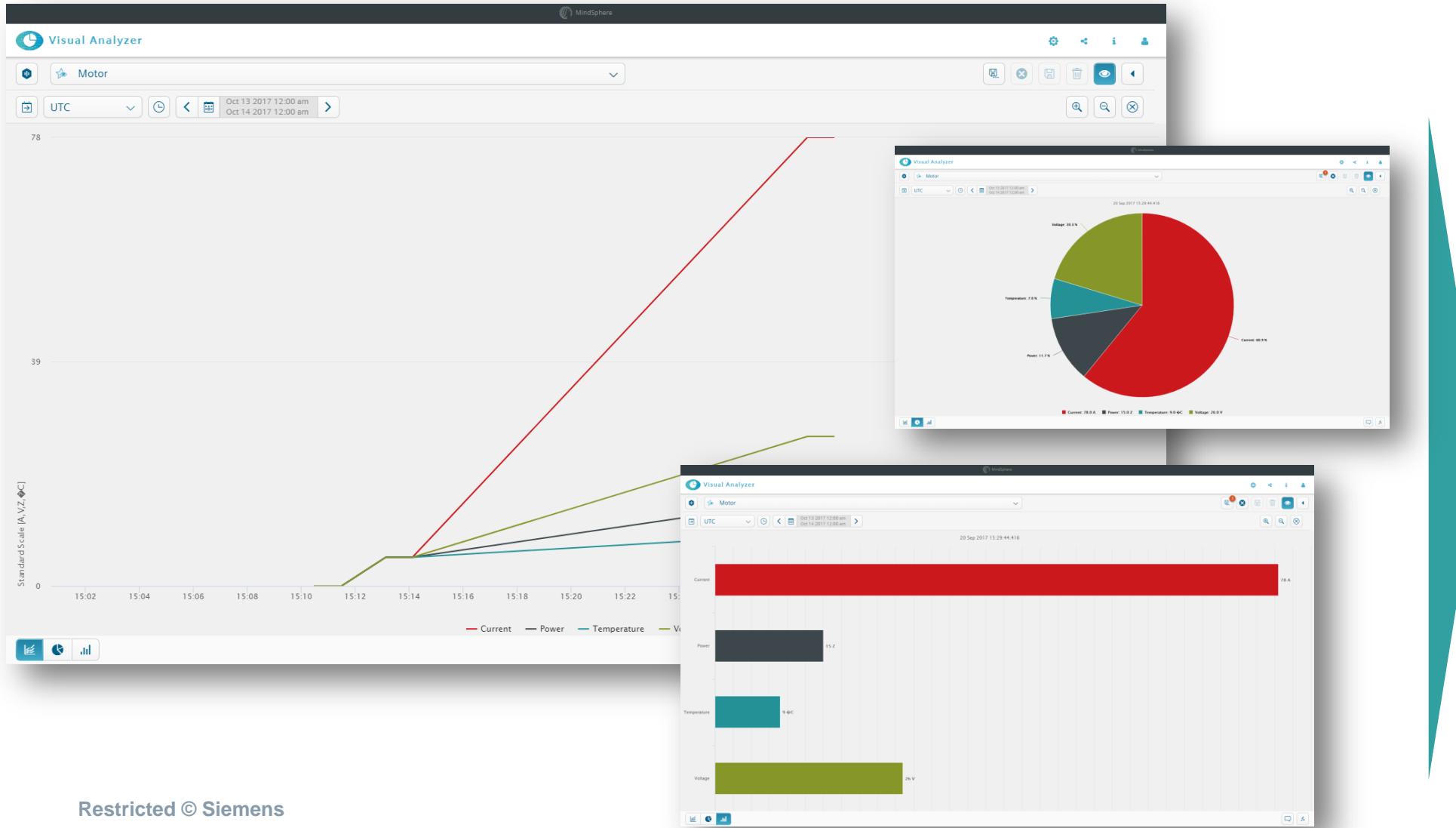
The screenshot shows the MindSphere Fleet Manager interface. On the left, a list view displays several assets: NEW\_GENERIC\_..., TBAsset1 (selected, showing details), Test\_Asset\_AC, testtao, Traingrack\_1, Training\_Agent\_N..., Training\_Box\_1, Training\_Box\_1a, and Training\_room. The selected asset, TBAsset1, has a detailed info card with fields: ID (b75549b996014c60809902662df158c7), Type (casseoe.TBAssetType1), and Description. To the right of the list is a map view titled 'Asset Location' showing the global distribution of assets, with a specific location marked in Africa.

### Equipment transparency:

- Live process data
- Root cause analysis
- Rule triggered events

# Fleet Manager Visual Analyzer

**SIEMENS**  
Ingenuity for life



## Visualize the data:

- Select visualization
- Select time frame
- Save and share current view

# Architecture & API Summary



How is MindSphere designed?

**As an open platform as a service (PaaS) for scalable, global IoT connectivity and application development.**

How shall we develop applications for MindSphere?

**By getting MindAccess Developer plan**

**(S, M, L), we can start using 18 different APIs to make the application development easy and fast.**

# MindSphere Tools for Training

# MindSphere - Tools for Training

## Learning Goals



**Which tools do we use in the training?**

- Virtual Box
- Ubuntu Linux
- Cloud Foundry CLI
- Visual Studio Code
- Postman

Official name: **Representational state transfer**

REST-compliant web services allow requesting systems to access and manipulate textual representations of web resources using a uniform and predefined set of stateless operations

### Core ideas

- Performance
- Scalability
- Simplicity of a uniform Interface
- Visibility of communication between components by service agents
- Statelessness

*Important: REST is an architectural style, not a protocol*

*REST was defined by Roy Fielding in his 2000 PhD dissertation  
Roy Fielding is the co-founder of the Apache HTTP Server project*

### HTTP-based RESTful APIs are defined with the following aspects:

- base URL, such as `http://api.example.com/resources/`
- an internet media type that defines state transition data elements, for example JSON
- standard HTTP methods (e.g., OPTIONS, GET, PUT, POST, and DELETE)

Verb	Usage
GET	Used to retrieve a resource.
POST	Primarily used to create a new resource. It is also used to retrieve resources if the input for the retrieval request is so large that it has to be sent in the request body instead of the header.
PUT	Used to fully replace an existing resource.
PATCH	Used to partially update an existing resource.
DELETE	Used to delete an existing resource.

### Request status codes

Code	Status	Code	Status	Code	Status
<b>200</b>	OK	<b>400</b>	Bad Request	<b>500</b>	Internal Server Error
<b>201</b>	Created	<b>401</b>	Unauthorized	<b>501</b>	Not Implemented
<b>204</b>	No Content	<b>403</b>	Forbidden		
		<b>404</b>	Not Found		
		<b>405</b>	Method Not Allowed		
		<b>406</b>	Not Acceptable		

# Web Development Basics

## JSON



```
{  
    "Herausgeber": "Xema",  
    "Nummer": "1234-5678-9012-3456",  
    "Deckung": 2e+6,  
    "Waehrung": "EURO",  
    "Inhaber":  
    {  
        "Name": "Mustermann",  
        "Vorname": "Max",  
        "maennlich": true,  
        "Hobbys": ["Reiten", "Golfen", "Lesen"],  
        "Alter": 42,  
        "Kinder": [],  
        "Partner": null  
    }  
}
```

JSON: JavaScript Object Notation

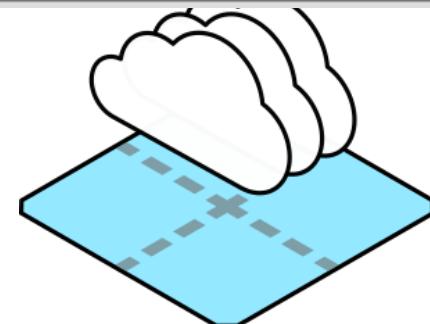
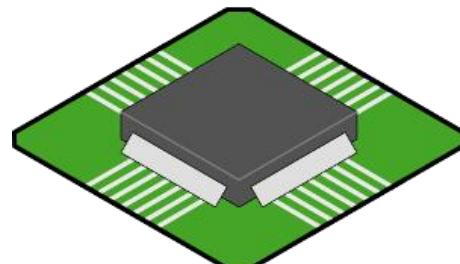
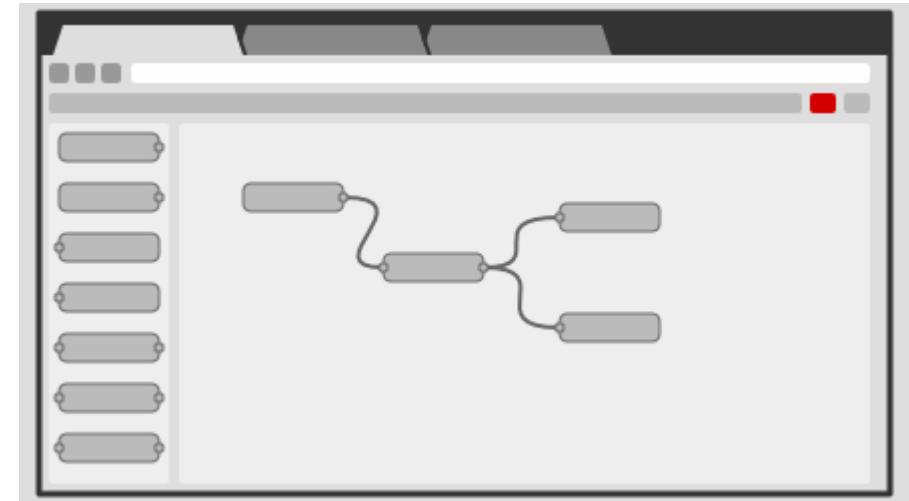
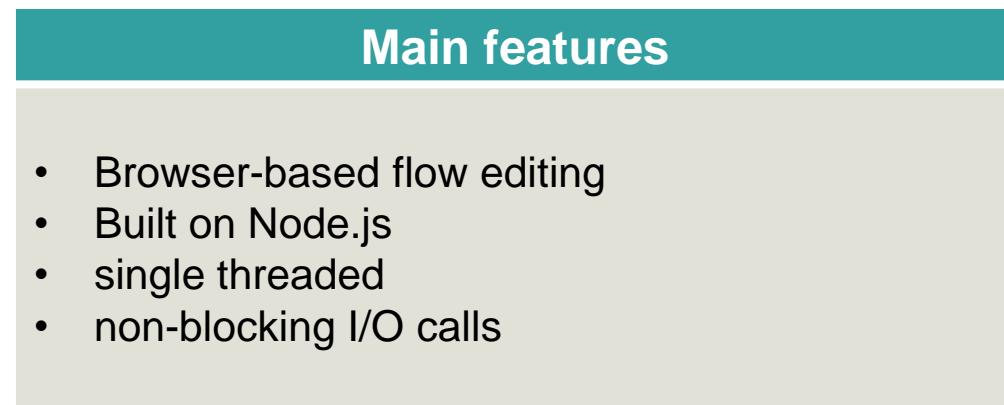
We use the following tools in the training:

- Node-Red
- Microsoft Visual Studio Code
- Postman
- Browser (Chrome)

**Where do you get these tools?**

Node-RED is a programming tool for wiring together hardware devices, APIs and online services in new and interesting ways.

It provides a browser-based editor that makes it easy to wire together flows using the wide range of nodes in the palette that can be deployed to its runtime in a single-click



### Preperations:

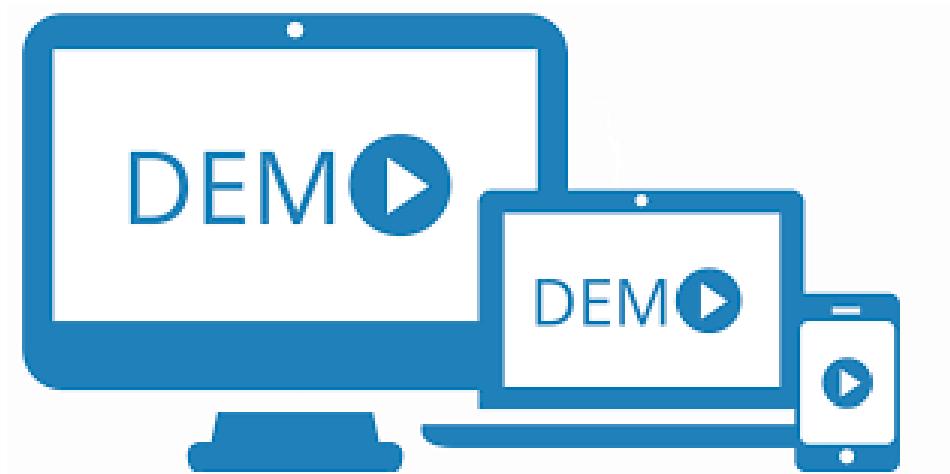
- Everybody should have already installed the Virtual Box on your notebook.
- BIOS flag for virtualization (VT-X) has to be enabled.

### What to do next:

- Install Node-Red
- Install Postmann

Download and install Postman

Use Postman to call MindSphere API



Download and install Postman

Use Postman to call MindSphere API



### Task 1: Download and install Postman

- Use the link <https://www.getpostman.com/> to download and install postman
- Import the postman collection provided by your trainer.
- Follow the instruction of your trainer to set up the collection and test your first API call of Siemens MindSphere

### Task 2: Download and install Visual Studio Code

### Task 3: Download and install Node-JS and Node Red

# MindSphere - Tools for Training

## Summary

**Which tools do we use in the training?**

- Tools are provided
- Install Virtual Box and the latest OVA file



# Overview of MindConnect API

Jan, 2018

# MindConnect API

## Learning Goals



- Overview of MindConnect API

The MindConnect Service exposes an API that enables shop floor devices to send data securely and reliable to MindSphere. It opens the MindSphere platform to custom applications to collect and send data which shall be stored and used by applications in the cloud.

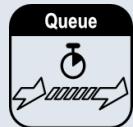
The MindConnect Service enables the development of custom data collectors, also referred as custom agents. This software applications act as data sources that upload the collected data into the MindSphere.

### Description

#### Aggregate Service

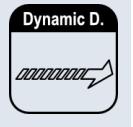
Use Aggregate Service to read aggregated values

- Read aggregates with three durations: 2 minutes, 1 hour, and 1 day.



#### Dynamic Data Service

Use Dynamic Data Service to retrieve collected data for an asset as an array of Time stamp and value tuples



Imp

#### MindConnect API

Use MindConnect API to develop custom agents and connect your device or application to MindSphere



New

#### Agent Management

Use Agent Management to create, edit or remove MindConnect elements, on-board and off board



#### Identity and Access Management

Use Customer Management APIs to manage your customers (e.g. add new ones or change existing) for own tenants only



New

#### Asset Management

Represent physical assets from your site in MindSphere. Use models and create instances, set relations to others and create structures such as hierarchies.



Imp

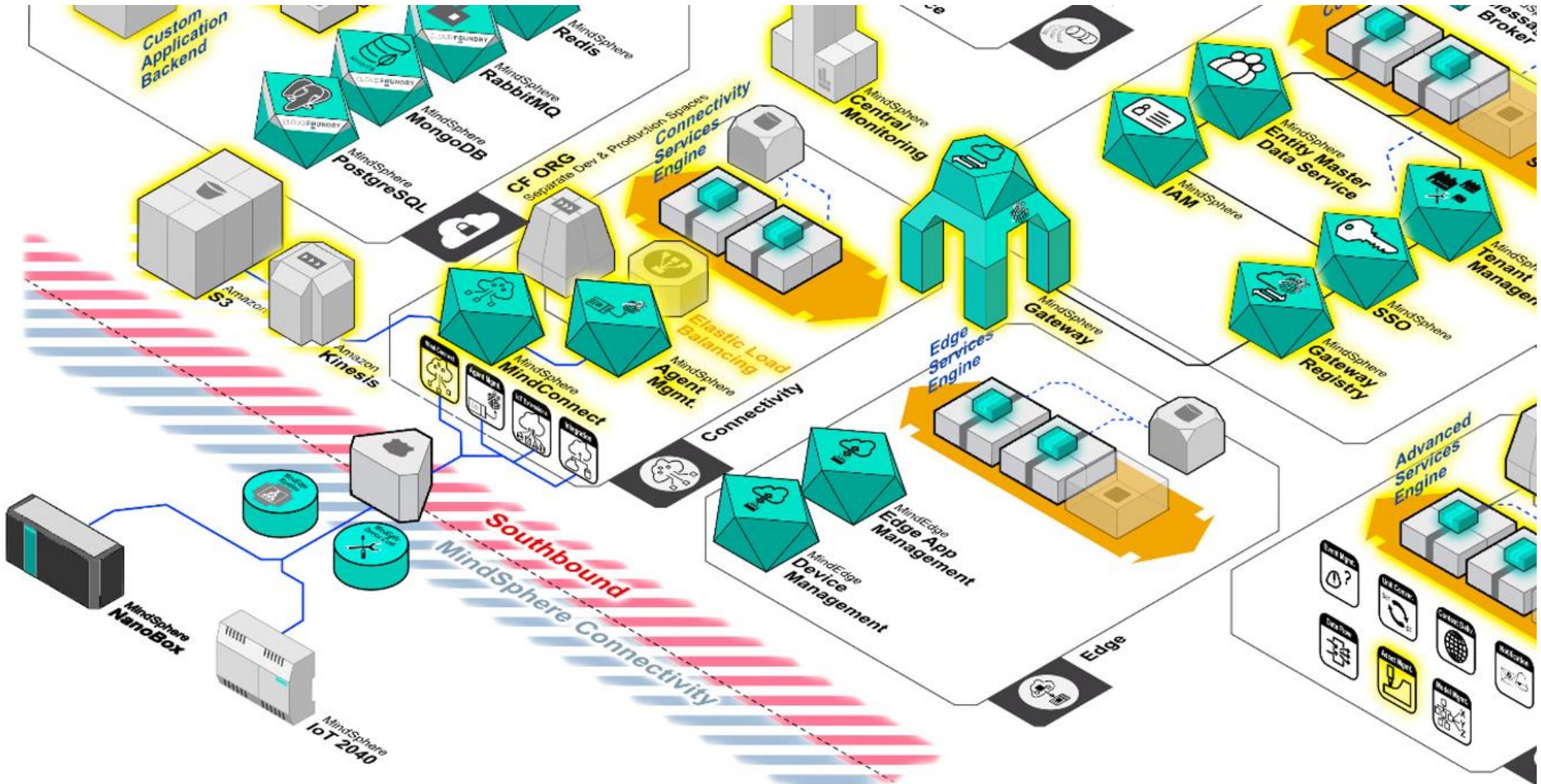
#### Event Management

Manage standardized and customized events. Acquire events from the field & other applications.



# MindConnect API Overview

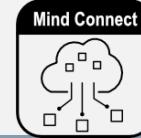
**SIEMENS**  
Ingenuity for life



The custom agent needs a field side network infrastructure to forward and route outbound HTTPs requests to the Internet.

MindConnect supports multiple agent device classes: strong hardware platforms as well as resource constrained devices. All target agent platforms must comply to the following minimum requirements:

- HTTP processing
- TLS
- JSON parsing
- JSON Web Token (JWT) generation
- HMAC generation (preferably SHA2 based hashing)



# MindConnect API

## General Workflow

- Provisioning an Agent
- Acquiring Boarding Configuration
- Onboarding the Agent
- Acquiring Access Token
- Configuring Data Source
- Consuming Exchange Services



### Provisioning an Agent

Get Boarding Configuration

Onboarding Agent

Acquiring Access Token

DataSource Configuration

Create Data Point Mapping

Exchange Data

In order to create an agent, an **Asset (Entity) Type** and an **Asset (Entity)** needs to be created beforehand. In MindSphere every Agent Entity needs to extend from base EntityType **core.basicAgent**.

Asset Type and Asset can be created on MindSphere User Interface with **Asset Manager**. Or through the IoT Service API:

(`<>_gateway_url>>/api/iottypes/v3/entitytypes/{{_tenant}}.{{_agent_entity_type}}`)

Other step includes creating an Asset from the Asset Type, later we will use the created Entity to create the Agent.

# MindConnect API

## Create the Agent



### Provisioning an Agent

Get Boarding Configuration

Onboarding Agent

Acquiring Access Token

DataSource Configuration

Create Data Point Mapping

Exchange Data

After the Entity has been created we can create an Agent that will be bound to this Entity.  
The Agent can be created on MindSphere UI with **Asset Manager**

MindSphere provides also "<>URL<>/api/agentmanagement/v3/agents" endpoint to provision an Agent below is a sample request:

POST ... HTTP1/1

Content-Type: Application/Json

...

{

```
"entityId": "3b27818ea09a46b48c7eb3fb878349f",
"name": "Nanobox Agent",
"securityProfile": "SHARED_SECRET"
```

}

# MindConnect API

## Demonstration



Provisioning an Agent

Get Boarding Configuration

Onboarding Agent

Acquiring Access Token

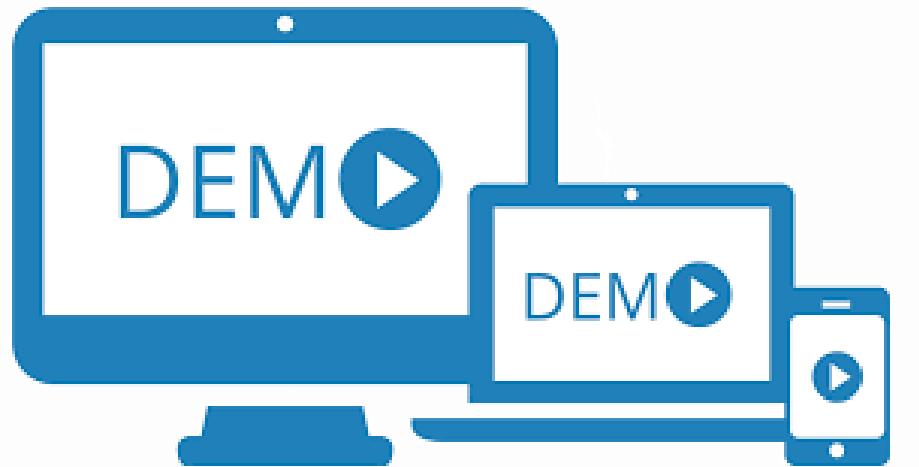
DataSource Configuration

Create Data Point Mapping

Exchange Data

Create Asset and Asset Type with MindSphere Asset Manager

Create Asset and Asset Type with API through Postman



### Provisioning an Agent

Get Boarding Configuration

Onboarding Agent

Acquiring Access Token

DataSource Configuration

Create Data Point Mapping

Exchange Data

**Task 1: Create Asset and Asset Type with MindSphere Asset Manager**

**Task 2 (optional): Create Asset and Asset Type with API through Postman**

**Task 3: Create Agent and associate the Agent with Asset**

# Hands-on



### Task 1: Launchpad and Asset Manager

#### Log into MindSphere Tenant

- <https://academy2.eu1.mindsphere.io/> and open the Asset Manager

#### Create Aspect and Device Asset

- Click the tab **Aspect** and click the button **New**  to add new aspect type.
  - Name your Aspect with name Aspect\_<your name>
- Creating Aspect containing variables with data type and unit. For example:
  - **Name:** Poti      **Data type:** Double      **Unit:** V;
  - **Name:** Temperature    **Data type:** Double      **Unit:** °C;
  - **Name:** Ratotion      **Data type:** Double      **Unit:** rpm;
- Click the tab **Types** and open the standard type **BasicDevice**. click the button **New**  to add new asset type
  - Enter the name “Device\_Asset\_Type\_<your name>”
  - Select the new Aspect created in the last step and click the Add button to associate the Aspect with your new Asset type.
- Go to the folder **01\_MindConnect\_API\_Tokyo** and Create Asset Instance
  - Select the new asset type create in last step as asset type for your new asset.
  - Enter the name “Device\_Asset\_<your name>”

### Task 2: Create Agent Asset Type and Asset

Log into MindSphere training tenant and open the **Asset Manager**

- <https://academy2.eu1.mindsphere.io>

#### Create Asset Type and Asset

- Click the tab **Types** and click the button **New**  to add new asset type:
  - Select **core.basicagent** as the parent asset type
  - Enter the name “Agent\_Asset\_Type\_<your name>”
  - Save the new Asset Type by clicking the save button
- Click the tab **Assets** and select folder **01\_MindConnect\_API\_Tokyo** as location for your new Asset:
- Click the button **New**  to add new asset:
  - Select the agent asset type you created in the last step
  - Enter the name “Agent\_Asset\_<your name>”
  - Save the new asset by clicking the save button
- Select the new Asset on Mindsphere UI and note the internal ID of your new asset, for example:
  - <https://<URL>..../details/asset?selected=ecf3bc01ec8445b2b71ace859ba2bfaa>

### Task 4 (Optional): Use Postman to create Asset Type and Asset

Start Postman and use the REST API call to create asset type

- Use the [PUT](#) request to create Asset Type

```
PUT https://gateway.eu1.mindsphere.io/api/assetmanagement/v3/assettypes/{{_tenant}}.{{_agent_entity_type}}
```

- Use following JSON object as body of your PUT request and send the request to create the asset type

```
{
  "name": "{{your_asset_type_name}}",
  "category": "dynamic",
  "description": "Test agent for {{tenant_name}} tenant",
  "parentTypeId": "core.basicagent",
  "instantiable": true,
  "scope": "private",
  "variables": [
    {
      "name": "temperature",
      "dataType": "STRING",
      "unit": "C",
      "searchable": true,
      "length": 3
    }
  ]
}
```

- Note the returned message

### Create Asset

- Use the **POST** request to create Asset

```
POST https://gateway.eu1.mindsphere.io/api/assetmanagement/v3/assets
```

- Use following JSON object as body of your PUT request

```
{  
  "name": "{{_entity_name}}",  
  "typeId": "{{_tenant}}.{{_agent_entity_type}}",  
  "parentId": "{{_location_new_asset}}"  
}
```

- Note the result returned by MindSphere. The unique ID of asset is required for the next exercises.
- Optional: try to update the Asset with **PUT** request, or delete the Asset with **DELETE** request. You have to however make sure that a new asset is available for the next steps.

### Task 5: Use Postman to create Agent

Start Postman and use the REST API call to create agent

- Use the **POST** request to create agent

```
POST https://gateway.eu1.mindsphere.io/api/agentmanagement/v3/agents
```

- Use following JSON object as body of your PUT request and send the request to create the asset type

```
{  
  "entityId": "{{_asset_id}}",  
  "name": "{{_agent_name}}",  
  "securityProfile": "SHARED_SECRET"  
}
```

- Note the result returned by MindSphere. The unique ID of agent is required for the next exercises.
- Optional: try to update the Agent with **PUT** request, or delete the Agent with **DELETE** request. You have to however make sure that a new agent is available for the next steps.

Provisioning an Agent

**Get Boarding Configuration**

Onboarding the Agent

Acquiring an Access Token

DataSource Configuration

Create Data Point Mapping

Exchange Data

Upon successful creation the Agent a **201 CREATED** response received. With JSON body holding additional data about the newly provisioned Agent. Initially, the boarding status of the newly created agent is **NOT\_ONBOARDED**. The Agent Management creates the onboarding configuration for the related agent.

HTTP/**1.1 201** Created

Content-Type: Application/Json

...

{

```
"id": "3b27818ea09a46b48c7eb3fb878349f",
"eTag": "2",
"entityId": "3b27818ea09a46b48c7eb3fb878349f",
"name": "Nanobox Agent",
"securityProfile": "SHARED_SECRET"
```

}

# MindConnect API

## Acquiring Boarding Configuration



Provisioning an Agent

**Get Boarding Configuration**

Onboarding the Agent

Acquiring an Access Token

DataSource Configuration

Create Data Point Mapping

Exchange Data

After agent creation, agent needs to acquire **boarding configuration** in order to board to MindSphere. If agent is not boarded, it can not use any of the services offered by Mindsphere. Boarding functionality is provided by **Boarding endpoint**.

Sending an **HTTP GET** request

`<<URL>>/api/agentmanagement/v3/agents/{{agent_id}}/boarding/configuration`  
with an agent Id in the query parameter; returns following information:

Parameter	Description	Remarks
baseUrl	Holds the southgate url	10 GB
iat	Initial Access Token, A JSON Web Token that is being signed by the AgentIAM, required to do initial registration.	IAT are subject to expiration and by default they have 7 days of validity, if IAT expired; Agent needs to call this endpoint to acquire a new configuration.
clientCredentialProfile	Agents security profile, set during provisioning step.	RSA_3072, SHARED_SECRET
clientId	Agent Id	
tenant	Agents tenant	
expiration	Holds IAT expiration time	Default value is 7 days

# MindConnect API

## Acquiring Boarding Configuration

Provisioning an Agent

Get Boarding Configuration

Onboarding the Agent

Acquiring an Access Token

DataSource Configuration

Create Data Point Mapping

Exchange Data

```
{  
  "content": {  
    "baseUrl": "https://southgate.eu-central.mindsphere.io",  
    "iat":  
"eyJhbGciOiJSUzI1NiIsInR5cCI6IkpXVCJ9.eyJpc3MiOiJTQ0kiLCJzdWIiOiJ0ZXN0LWFnZW50IiwiYX  
VkJpbIkFJQU0iXSviaWF0IjoxNTEzNjA3NjQ3LCJuYmYi0jE1MTM2MDc2NDcsImV4cCI6MTUxNDIxMjQ0Ny  
wianRpIjoiODBkNGViYTAtZTQwMC0xMWU3LTlmMDYtM2QzOTg3NTQ1MjNjIiwic2NvcGUI0lsiSUFUI0sIn  
RlbmFudCI6InRlc3QtdGVuYW50IiwiY2xpZW50X2NyZWRlbnRpYlxzX3Byb2ZpbGUI0lsiUlNBXzMwNzIiXS  
wic2NoZW1hcyI6WyJ1cm46c211bwVuczptaw5kc3BoZXJ1OnYxIl19.F/C44dH0orAVrKXDxcdGTgDFGsYzx  
OLR84j49S1qQMcX4GKq0HGfN7mm9N2tAUo8+z9ISRIW+D/n81ui0F4ro+cyjGFbstatsRVH8hFFbZe4WcSCh  
BHpwmA9IB0L0iFo/8C/7FppjhmHn1JNTQrfdkPJbtgQ1jt1Y46Bqw2pKp3wPUPJb21nLhVwMv1AiAbISg17F  
azhzuV7n12pJxdtpEB+mYZZe16P9amW217iyaDyRZ0+UhldHyKYwX0in5SpvJ1gzFlhMNwYWhG8zk6EJ80YY  
490hTYLHtvoEm8IIxarPWwmMztwrEXvNvQi6svZ1NCgbkuSTwdEyJDzPqmbw==",  
    "clientCredentialProfile": [  
      "SHARED_SECRET"  
    ],  
    "clientId": "5fa51b64-dce2-11e7-9296-cec278b6b50a",  
    "tenant": "6e70640a-dce2-11e7-9296-cec278b6b50a"  
  },  
  "expiration": "2017-12-18T14:04:01.076Z"  
}
```

Provisioning an Agent

## Get Boarding Configuration

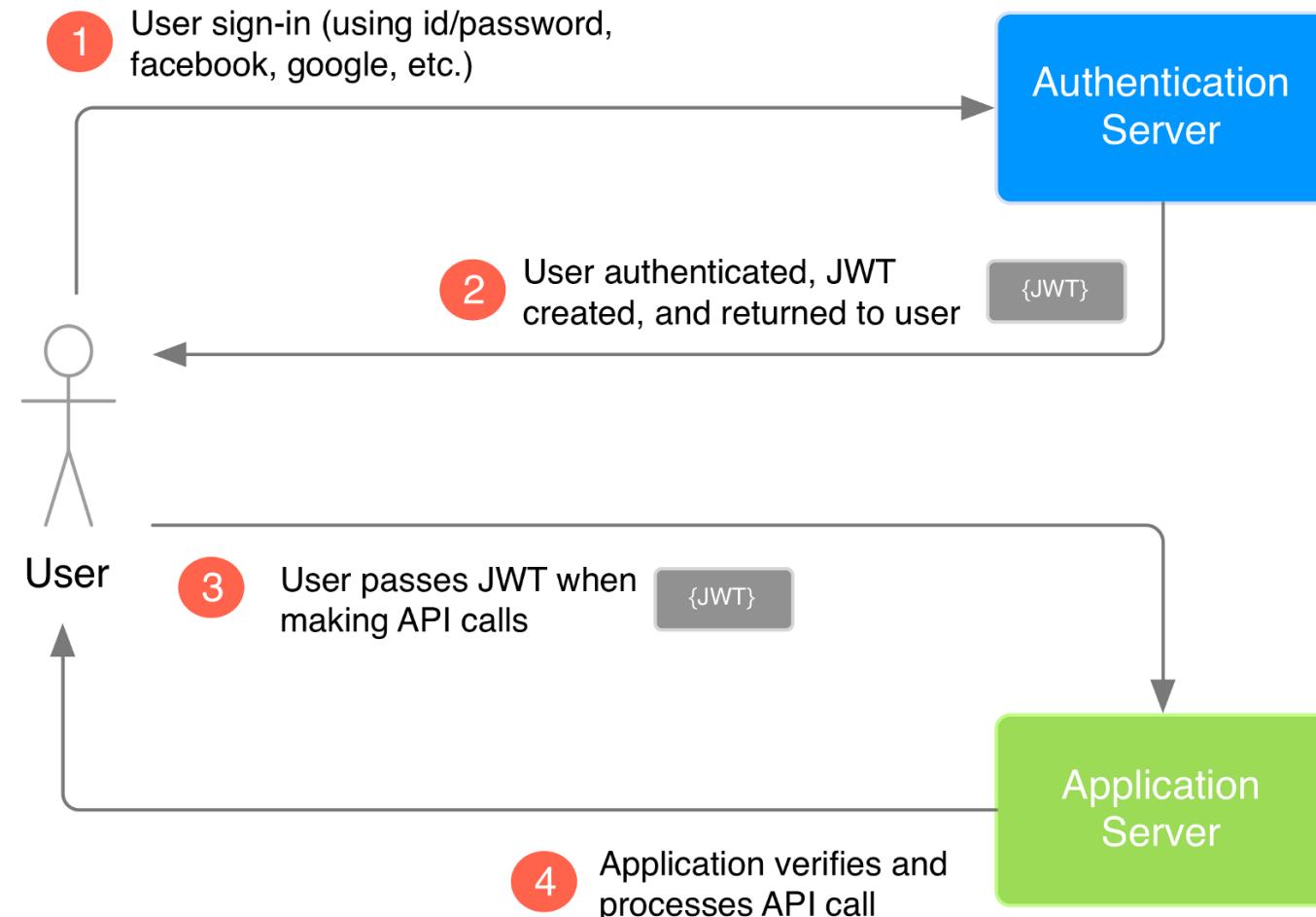
Onboarding the Agent

Acquiring an Access Token

DataSource Configuration

Create Data Point Mapping

Exchange Data



Provisioning an Agent

Get Boarding Configuration

Onboarding the Agent

Acquiring an Access Token

DataSource Configuration

Create Data Point Mapping

Exchange Data

Get Boarding Information of your Agent



Provisioning an Agent

Get Boarding Configuration

Onboarding the Agent

Acquiring an Access Token

DataSource Configuration

Create Data Point Mapping

Exchange Data

**Task 1: Get Boarding Information of your Agent**

**Task 2: Create the Agent Asset with type MindConnect LIB and generate boarding information**

# Hands-on



### Task 1: Get boarding configuration of your agent

Start Postman and use the REST API call to get the boarding configuration

- Use the **GET** request to create agent

```
GET https://gateway.eu1.mindsphere.io/api/agentmanagement/v3/agents/{agent_id}/boarding/configuration
```

- Note the result returned by MindSphere. The **IAT** token of agent is required for the next exercises

```
{  
  "content": {  
    "baseUrl": "https://southgate.eu1.mindsphere.io",  
    "iat": "eyJraWQiOiJrZXktaw.....",  
    "clientCredentialProfile": [  
      "SHARED_SECRET"  
    ],  
    "clientId": "f03714282d374383abff1cf486d231c",  
    "tenant": "casseoe"  
  },  
  "expiration": "2018-05-21T14:40:09.000Z"  
}
```

- Use the **GET** request to get the boarding status

```
GET https://gateway.eu1.mindsphere.io/api/agentmanagement/v3/agents/{agent_id}/boarding/status
```

## Hands-On Exercise 3: Boarding Information

### Task 2: Create the Agent Asset with type MindConnect LIB and generate boarding information

Log into MindSphere training tenant and open the **Asset Manager**

- <https://academy2.eu1.mindsphere.io>

#### Create Asset Type and Asset

- Click the tab **Assets** and select folder **01\_MindConnect\_API\_Tokyo** as location for your new Asset:
- Click the button **New**  to add new asset:
  - Select the asset type MindConnect LIB (core.mclib) as your asset type
  - Enter the name “Agent\_Asset\_mclib\_<your name>”
  - Save the new asset by clicking the save button

#### Generate the onboarding token

- Select the new Asset on Mindsphere UI and note the internal ID of your new asset, for example:
  - <https://<URL>..../details/asset?selected=ecf3bc01ec8445b2b71ace859ba2bfaa>
- Click the section MindConnect Lib and Open the overview tab of MindConnect Lib
- Click button **Download Onboarding Key**
- Note the onboarding information including IAT has been generated.
- The **IAT** token of agent is required for the next exercises.

Provisioning an Agent

Get Boarding Configuration

Onboarding the Agent

Acquiring an Access Token

DataSource Configuration

Create Data Point Mapping

Exchange Data

### Initial contact (Onboarding)

We need to ensure that agents, that connect the first time to the MindSphere, become a valid communication partner having a known human owner.

### Regular interaction

Once an agent has become a known and authorized entity in MindSphere, we need to ensure that each time it reconnects, we are still talking to the same agent.

After acquiring the **IAT** ( Initial access token ), Agent needs to register with MindSphere. This registration process takes place in accordance with the **OAuth 2.0** Authorization protocol.

Agent sends an HTTP POST request to the MindSphere, The way request constructed varies slightly depending on the Agent Security Profile.

Provisioning an Agent

Get Boarding Configuration

Onboarding the Agent

Acquiring an Access Token

DataSource Configuration

Create Data Point Mapping

Exchange Data

Agent sends an **HTTP POST** request with an **empty body**, MindSphere receives the request, validates it (IAT validation, mandatory headers validation etc) and if validation succeeds, creates an Agent **SHARED\_SECRET** and stores it in its database with a predefined SHARED\_SECRET expiration time (7 days by default). MindSphere creates another token called **Registration Access Token (RAT)** for the client. When the client\_secret expiration occurs client is expected to sent another Registration Update Request (*Key Rotation*) RAT token instead of IAT so the registration can be renewed.

POST /api/agentmanagement/v3/register

POST /api/agentmanagement/v3/register HTTP/1.1

Content-Type: application/json

Accept: application/json

Authorization: Bearer

eyJhbGciOiJSUzI1NiIsInR5cCI6IkpXVCJ9.eyJpc3MiOiJTQ0kiLCJzdWIiOiJ0

ZXN0LWFnZW50IiwiYXVkcjpbIkFJQU0iXSwiaWF0Ijox...

... //Initial Access Token (IAT) acquired

==

Host: southgate.<region-env>.mindsphere.io

{ } //empty JSON Object

Provisioning an Agent

Get Boarding Configuration

Onboarding the Agent

Acquiring an Access Token

DataSource Configuration

Create Data Point Mapping

Exchange Data

```
HTTP/1.1 201 Created
Content-Type: application/json
Cache-Control: no-store
Pragma: no-cache
{
  "client_id": "008ceaf9dff443ee94205f903aa545c4",
  "token_endpoint_auth_method": "client_secret_jwt",
  "grant_types": [
    "client_credentials"
  ],
  "client_secret": "ly7zmXbz0Izla6B82x1AYRVBcKDcCFatwmnHQmR8Ycw",
  "client_secret_expires_at": 1514206768,
  "registration_access_token": "eyJraWQiOiJrZXktawQtMSI.....",
  "registration_client_uri": "https://southgate.eu-central-rc.mindsphere.io/api/agentmanagement/v3/regi
ster/008ceaf9dff443ee94205f903aa545c4"
}
```

Provisioning an Agent

Get Boarding Configuration

Onboarding the Agent

Acquiring an Access Token

DataSource Configuration

Create Data Point Mapping

Exchange Data

Parameter	Description	Remarks
<code>client_id</code>	Holds agent id	
<code>token_endpoint_auth_method</code>		
<code>grant_types</code>	Agents security profile, set during provisioning step.	
<code>client_secret</code>	Generated Client Secret, Client will use when requesting access token	
<code>client_secret_expires_at</code>	Time generated client secret will expire	Client needs to re-register with RAT if expiration occurs. Default value is 7 days
<code>registration_access_token</code>	Registration Access Token, client needs this token to renew its registration	
<code>registration_client_uri</code>	URI to use when client renews its registration	

Provisioning an Agent

Get Boarding Configuration

Onboarding the Agent

Acquiring an Access Token

DataSource Configuration

Create Data Point Mapping

Exchange Data

By default, each registered agent (either RSA 3072 or Shared Secret security profile) has its credentials valid for 7 days. After this period MindSphere will not grant access tokens to the agent and it will render the agent credentials invalid. The agent needs to **re-register (Key-Rotation)** to MindSphere providing its Registration Access Token (RAT). The agent needs to perform this request to the URL value of registration\_client\_uri. This URL is provided in the JSON body of the HTTP response 201 Created to a successful registration.

PUT /api/agentmanagement/v3/register/{agent\_id}

PUT /api/agentmanagement/v3/register/{agent\_id} HTTP/1.1

Content-Type: application/json

Accept: application/json

Authorization: Bearer

eyJhbGciOiJSUzI1NiIsInR5cCI6IkpXVCJ9.eyJpc3MiOiJTQ0kiLCJzdWIiOiJ0ZXN0LWFnZW50IiwiYXVkIjpbIkFJQU0iXSviaWF0Ijox...

... //Registration Access Token (RAT)

Host: authorizationserver.mindsphere.com

```
{  
    "client_id": "agentid"  
}
```

# MindConnect API

## Refreshing a Registration



Provisioning an Agent

Get Boarding Configuration

Onboarding the Agent

Acquiring an Access Token

DataSource Configuration

Create Data Point Mapping

Exchange Data

If the RAT is correct and the provided client Id matches, a new registration access token is returned to the client. The old registration access token must be discarded by the client, MindSphere discards the old RAT automatically.

MindSphere returns an HTTP response 200 OK. The structure of the JSON body is the same as for the initial registration of the agent:

```
{  
    "client_id": "f03714282d374383abff1cf486d231c",  
    "token_endpoint_auth_method": "client_secret_jwt",  
    "grant_types": [  
        "client_credentials"  
    ],  
    "client_secret": "OsROMak-0iGET_P7hr-Dd_Vl6PzP4M6n9r8ssX2YF4M",  
    "client_secret_expires_at": 1526996947,  
    "registration_access_token":  
    "eyJraWQiOiJrZXktawQtMSIsInR5cCI6IkpXVCIsImFsZyI6IlJTJmJU2I.....",  
    "registration_client_uri":  
    "https://southgate.eu1.mindsphere.io/api/agentmanagement/v3/register/f03714282d3743  
    83abff1cf486d231c"  
}
```

# MindConnect API

## Demonstration



Provisioning an Agent

Get Boarding Configuration

Onboarding the Agent

Acquiring an Access Token

DataSource Configuration

Create Data Point Mapping

Exchange Data

Onboarding the Agent

Key rotation



Provisioning an Agent

Get Boarding Configuration

Onboarding the Agent

Acquiring an Access Token

DataSource Configuration

Create Data Point Mapping

Exchange Data

Task 1: Onboarding the agent

Task 2: Key rotation

# Hands-on



### Task 1: Use IAT to onboard and offboard the agent

Start Postman and use the REST API call to register/onboard the agent

- Create the **POST** request to register/onboard agent

```
POST https://gateway.eu1.mindsphere.io/api/agentmanagement/v3/register
```

- In the header part of post request, use **IAT** token as value

**Authorization:** Bearer

```
eyJraWQiOiJrZXktawQtMSIsInR5cCI6IkpXVCIsImFsZyI6IlJTMjU2In0.eyJpc3Mi0iJTQ0kiLCJzdWIi0iJmMDM3MTQyODJkMzc0Mz
gzYWJmZjFjZmQ0ODZkMjMxYyIsImF1ZCI6IkFJQU0iLCJpYXQiOjE1MjYzMdgsIm5iZiI6MTUyNjMwODgwOSwiZXhwIjoxNTI2OTEz
NjA5LCJqdGkiOiJkNzI0ZmQ3ZC1jZGQ5LTRiNGEtYTMwYi05ZjdiNzQ4YWQ2OTEiLCJzY29wZSI6Ik1BVCIsInRlbiI6ImNhC3N1b2UiLC
JjbG1lbnRfY3JlZGVudG1hbHNfcHJvZmlsZSI6WyJTSEFSURFU0VDUkVUI10sInNjaGVtYXMiOlsidXJuOnNpZW1lbnM6bWluZHNwaGVy
ZTp2MSJdfQ.FTMV4kE1jCETuDrvBLHD5brwJwomrCKkfCzPgEbom20nvr1Mk8sAEiQBDchYS6Pk9.....
```

- In the body part of post request, use empty JSON as value

```
{ } //empty JSON Object
```

## Hands-On Exercise 4: Onboarding the Agent

Start Postman and use the REST API call to register/onboard the agent

- Save the response, especially the registration\_access\_token (RAT)

```
{  
  "client_id": "f03714282d374383abff1cf486d231c",  
  "token_endpoint_auth_method": "client_secret_jwt",  
  "grant_types": [  
    "client_credentials"  
  ],  
  "client_secret": "otUcSKW4Ai9Vr8YNOTgaH1TEik0JzmPtba-w3xvwlcyc",  
  "client_secret_expires_at": 1526917298,  
  "registration_access_token": "eyJraWQiOiJrZXktawQtMSIsInR5cCI6IkpXVCIsImFsZyI6IlJTdjU2In0.....",  
  "registration_client_uri":  
    "https://southgate.eu1.mindsphere.io/api/agentmanagement/v3/register/f03714282d374383abff1cf486d231c"  
}
```

- After the onboard/register call, use the **GET** request to get the boarding status

```
GET https://gateway.eu1.mindsphere.io/api/agentmanagement/v3/agents/{agent_id}/boarding/status
```

- Use the **POST** request to offboard the agent

```
POST https://gateway.eu1.mindsphere.io/api/agentmanagement/v3/agents/{agent_id}/boarding/offboard
```

### Task 2: Onboard the Agent with type MindConnect LIB

#### Generate the onboarding token

- Go to MindSphere Asset Manager and open the agent asset with type MindConnect LIB
- Open the overview tab of MindConnect Lib and click button **Download Onboarding Key**
- Copy the **IAT** value generated.

#### Start Postman and use the REST API call to register/onboard the agent

- Create the **POST** request to register/onboard agent

```
POST https://gateway.eu1.mindsphere.io/api/agentmanagement/v3/register
```

- In the header part of post request, use **IAT** token as value

```
Authorization: Bearer eyJraWQiOiJrZXktawQtMSIsInR5cCI6IkpXVCIsImFsZyI6IlJTmjU2In0.eyJpc3MiOiJTQ0ki.....
```

- In the body part of post request, use empty JSON as value

```
{} //empty JSON Object
```

- Save the response, especially the registration\_access\_token (**RAT**)

```
{
...
"registration_access_token": "eyJraWQiOiJrZXktawQtMSIsInR5cCI6IkpXVCIsImFsZyI6IlJTmjU2In0....",
...
}
```

### Task 3: Key rotation

- Use the **POST** request to refresh the registration (key rotation)

```
PUT https://gateway.eu1.mindsphere.io/api/agentmanagement/v3/register/{agent-id}
```

- In the header part of post request, use **RAT** token as value

```
Authorization: Bearer eyJraWQiOjIjZXktawQtMSIsInR5cCI6IkpXVCIsImFsZyI6IlJTJU2In0.eyJpc3MiOiJTQ0k.....
```

- In the body part of post request, use empty agentid as value

```
{  
    client_id: agentid  
}
```

- Save the response, especially the registration\_access\_token (**RAT**) and client\_secret

```
{  
...  
"client_secret": "otUcSKW4Ai9Vr8YNOTgaH1TEik0JzmPtBA-w3xvwCyc",  
"client_secret_expires_at": 1526917298,  
"registration_access_token": "eyJraWQiOjIjZXktawQtMSIsInR5cCI6IkpXVCIsImFsZyI6IlJTJU2In0.....",  
"registration_client_uri":  
"https://southgate.eu1.mindsphere.io/api/agentmanagement/v3/register/f03714282d374383abff1cf486d231c"  
}
```

Provisioning an Agent

Get Boarding Configuration

Onboarding the Agent

Acquiring an Access Token

DataSource Configuration

Create Data Point Mapping

Exchange Data

Each onboarded agent is required to get an access token in order to use any services offered by MindSphere. MindSphere grants access tokens to clients for which token generation and grants comply with the rules of the OAuth 2.0 authorization framework.

MindSphere receives token requests (`HTTP POST`) from clients and grants access tokens back to them (`HTTP 200 OK`). Before granting an access token, the Access Token service performs the following checks:

- 1.The token request contains mandatory and expected headers.
- 2.The client is a known client (exists in the database).
- 3.The client status is **ONBOARDED**.

Provisioning an Agent

Get Boarding Configuration

Onboarding the Agent

Acquiring an Access Token

DataSource Configuration

Create Data Point Mapping

Exchange Data

Granting of access tokens is provided by the token endpoint of the [Agent Management Service](#). This is a sample client access token request expected by MindSphere:

```
POST /api/agentmanagement/v3/oauth/token HTTP/1.1
Content-Type: application/x-www-form-urlencoded
Host: localhost: 8061
content-length: 510
Connection: keep-alive
grant_type=client_credentials&client_assertion_type=urn:ietf:params:oauth:client-
assertion-type:jwt-bearer&client_assertion=eyJhbGciOiJSUzI1NiIsInR5cCI6IkpXVCJ9.....
```

Provisioning an Agent

Get Boarding Configuration

Onboarding the Agent

Acquiring an Access Token

DataSource Configuration

Create Data Point Mapping

Exchange Data

```
grant_type=client_credentials&client_assertion_type=urn:ietf:params:oauth:client-assertion-type:jwt-bearer&client_assertion=eyJhbGciOiJSUzI1NiIsInR5cCI6IkpXVCJ9.....
```

Parameter	Description	Remarks
grant_type	Defines the OAuth 2.0 access token flow requested by the client. MindSphere supports only client_credentials.	
client_assertion_type	Defines the assertion type (assertions are defined in RFC 7521). MindSphere supports only the jwt-bearer type assertion urn:ietf:params:oauth:client-assertion-type:jwt-bearer (for details see RFC 7523, Section 2.2 "Using JWTs for Client Authentication").	
client_assertion	Contains the assertion (self-signed JWT) signed (by the client) with client secret or client public key depending on the client security profile.	

# MindConnect API

## Signed JWT



Provisioning an Agent

Get Boarding Configuration

Onboarding the Agent

Acquiring an Access Token

DataSource Configuration

Create Data Point Mapping

Exchange Data

Client requests contain a self-signed JWT, conveyed with the body parameter `client_assertion`. This client assertion is required to be signed with the **credential data** provided at the registration:

```
{  
  "alg": "HS256",  
  "typ": "JWT"  
  
  {  
    "iss": "f03714282d374383abff1cf486d231c", //client_id  
    "sub": "f03714282d374383abff1cf486d231c", //client_id  
    "aud": [  
      "southgate"  
    ],  
    "iat": 1526393304, // date.getTime()  
    "nbf": 1526393304,  
    "exp": 1526998104,  
    "jti": "826d75fa-4f16-4aaa-ad85-807b8a61957b", //unique id  
    "schemas": [  
      "urn:siemens:mindsphere:v1"  
    ],  
    "ten": „academy2"  
  }  
}
```

Provisioning an Agent

Get Boarding Configuration

Onboarding the Agent

Acquiring an Access Token

DataSource Configuration

Create Data Point Mapping

Exchange Data

When all tests have been successfully passed, MindSphere returns an HTTP response **200 OK** that holds an access token. The returned access token is actually a signed JWT, which is valid for 1 hour by default

```
{  
    "access_token": "eyJraWQiOiJrZXktawQtMSIsImF.....zgw",  
    "token_type": "Bearer",  
    "expires_in": 3600,  
    "scope": [  
        "mdsp:core:DefaultAgent"  
    ],  
    "jti": "0d0f345f-b036-4484-851a-deced2022144"  
}
```

# MindConnect API

## Demonstration



Provisioning an Agent

Get Boarding Configuration

Onboarding the Agent

Acquiring an Access Token

DataSource Configuration

Create Data Point Mapping

Exchange Data

Create and sign the JWT

Acquire Access Token



Provisioning an Agent

Get Boarding Configuration

Onboarding the Agent

Acquiring an Access Token

DataSource Configuration

Create Data Point Mapping

Exchange Data

Task 1: Create and sign the JWT

Task 2: Acquire Access Token

# Hands-on



## Hands-On Exercise 5: Acquire Access Token

### Task 1: Create signed JWT

Open in the web browser [www.jwt.io](http://www.jwt.io)

- Create the jwt token with following value

```
HEADER: ALGORITHM & TOKEN TYPE
{
  "alg": "HS256",
  "typ": "JWT"
}

PAYLOAD: DATA
{
  "iss": "0b53fdd222904162afc5d8fd99a5120c",
  "sub": "0b53fdd222904162afc5d8fd99a5120c",
  "aud": [
    "southgate"
  ],
  "iat": 1530550328,
  "nbf": 1530550328,
  "exp": 1540550328,
  "jti": "826d75fa-4f16-4bbb-ad85-807b8a61957c",
  "schemas": [
    "urn:siemens:mindsphere:v1"
  ],
  "ten": "academy2"
}
```

Use the client\_secret to sign the jwt

- Enter the client\_secret generated in last step to sign the token

```
VERIFY SIGNATURE
HMACSHA256(
  base64UrlEncode(header) + "." +
  base64UrlEncode(payload),
  tPnYHEsVbVyXupFbTadqX1
)  secret base64 encoded
```

## Hands-On Exercise 5: Acquire Access Token

### Task 2: Acquire Access Token

Start Postman and use the REST API call to acquire the access token

- Create the **POST** request to register/onboard agent

```
POST https://gateway.eu1.mindsphere.io/api/agentmanagement/v3/oauth/token
```

- In the Header part of post request, use following value

```
Content-Type: application/x-www-form-urlencoded  
Host: localhost: 8061  
content-length: 510  
Connection: keep-alive
```

- In the body part of post request, use following as value:

```
grant_type=client_credentials&client_assertion_type=urn:ietf:params:oauth:client-assertion-type:jwt-bearer&client_assertion={{Signed_JWT}}
```

- **Note:** The {{Signed\_JWT}} has been created in the last step

## Hands-On Exercise 5: Acquire Access Token

- Save the result including Agent Access Token (AAT)

```
{  
  "access_token":  
    "eyJraWQiOiJrZXktaWQtMSIsImFsZyI6IlJTmjU2IiwidHlwIjois1dUIn0.eyJqdGkiOiI5MWY1MDQ1OS1hNjQ4LTQwM2ItYjU3ZS1lY  
    jUzNDgzYmI4MDIiLCJzY29wZSI6WyJtZHNwOmNvcmU6RGVmYXVsdEFnZW50I10sImlzcyI6Imh0dHBzOi8vc291dGhnYXR1LmV1MS5taW5  
    kc3BoZXJ1LmlvL2FwaS9hZ2VudG1hbmcFnZW1lbnQvIiwic3ViIjoimGI1M2ZkZDIyMjkwNDE2MmFmYzVkoGZkOT1hNTEyMGMiLCJ6aWQiO  
    iJhZ2VudGlhbSIsImF1ZCI6WyJzb3V0aGdhGUixSwiaWF0IjoxNTMwNTUwNzI2LCJ1eHAiOjE1MzA1NTQzMjYsInNjaGVtYXMiOlSidXJ  
    uOnNpZW1lbmM6bwluZHNwaGVyZTppYW06djEiXSwidGVuIjoiYWNhZGVteTIiLCJjYXQiOjJhZ2VudC10b2tlbjp2MSIsImdyYW50X3R5c  
    GUioiJjbG1lbmRfY3J1ZGVudGlhbHMifQ.sXdZjsamInF_MFENcf7d4v2RuGVU_63H_ptaFzhJw3NsJ3lqCM8C1CmW8bWdix0NCCDX4BPm  
    OFxRtmWHcHRQV8fxqUFcmB14A8BKpl29D1Y1k6cgt4IJyd-6rmYK9B80T0ScY7mpUZx9SJ5csYJfMfq-bg3YX80uaxvd-  
    aXI193B5Aj25dDmVqosfcwjuN6rdXLPWv5-1Xy-  
    1CkZbCheowOYhm281KsHALdk10yk7RjZqsEKcKmEg3ftCdZ5_USvBuDL3FhJ8zm3T3J9-RNlo90c-  
    NdszGB7ry6Bdr_HTyOF2mpqD1fVq4IWycGIF6kcqCsexIhmdE1PAiH8LhB8JQ",  
  "token_type": "Bearer",  
  "expires_in": 3600,  
  "scope": [  
    "mdsp:core:DefaultAgent"  
  ],  
  "jti": "91f50459-a648-403b-b57e-eb53483bb802"  
}
```

Provisioning an Agent

Get Boarding Configuration

Onboarding the Agent

Acquiring an Access Token

**DataSource Configuration**

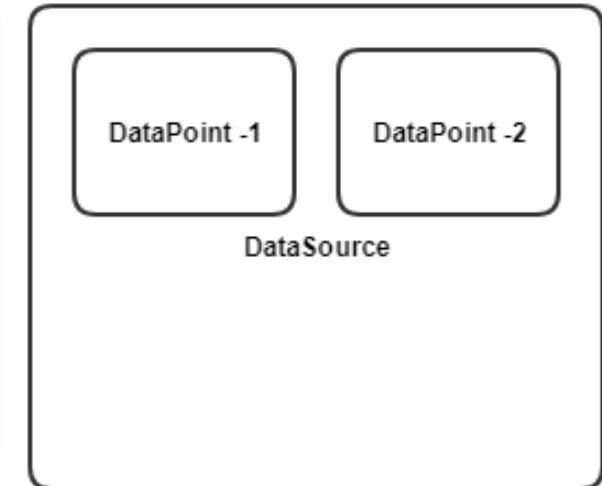
Create Data Point Mapping

Exchange Data

If Agent needs to upload data to MindSphere, MindSphere needs additional configuration to know how to interpret agents data streams. This configuration requires 3 definitions within MindSphere.

- DataSource Provisioning
- Property Set (Data Points) Provisioning
- Creating a Mapping for Property Set and DataSource

**"DataSource"** is a logical group that holds so called **"Data Points"**. Data points hold meta data about a specific metric that the Agent generates or measures. An example is: "if an agent measures ambient temperature and pressure data. Each of this two measurement needs to be defined as separate DataPoints".



Provisioning an Agent

Get Boarding Configuration

Onboarding the Agent

Acquiring an Access Token

**DataSource Configuration**

Create Data Point Mapping

Exchange Data

Mindsphere provides **dataSourceConfiguration** endpoint to create DataPoints and the DataSource they are grouped into.

**PUT [/api/agentmanagement/v3/agents/{id}/dataSourceConfiguration](#)**

```
HTTP/1.1 PUT api/agentmanagement/v3/agents/{id}/dataSourceConfiguration
Content-Type: application/json
If-Match: etag
Authorization: Bearer eyx...
{
  "configurationId": "string",
  "dataSources": [
    {
      "name": "string",
      "description": "string",
      "dataPoints": [
        {
          "id": "string",
          "name": "string",
          "description": "string",
          "type": "int",
          "unit": "string",
          "customData": {}
        }
      ],
      "customData": {}
    }
  ]
}
```

Provisioning an Agent

Get Boarding Configuration

Onboarding the Agent

Acquiring an Access Token

**DataSource Configuration**

Create Data Point Mapping

Exchange Data

If successfull MindSphere will respond with HTTP 200 OK with JSON body that holds the created DataSource Configuration. **"configurationId"** must be taken note since, during consuming exchange services this parameter needs to be provided back to MindSphere.

```
{  
    "id": "string",  
    "eTag": "2",  
    "configurationId": "string",  
    "dataSources": [  
        {  
            "name": "string",  
            "description": "string",  
            "dataPoints": [  
                {  
                    "id": "string",  
                    "name": "string",  
                    "description": "string",  
                    "type": "int",  
                    "unit": "string",  
                    "customData": {}  
                }  
            ],  
            "customData": {}  
        }  
    ]  
}
```

# MindConnect API

## Demonstration



Provisioning an Agent

Get Boarding Configuration

Onboarding the Agent

Acquiring an Access Token

**DataSource Configuration**

Create Data Point Mapping

Exchange Data

Configure and upload data source



Provisioning an Agent

Get Boarding Configuration

Onboarding the Agent

Acquiring an Access Token

DataSource Configuration

Create Data Point Mapping

Exchange Data

### Task 1: Configure and upload data source

# Hands-on



## Hands-On Exercise 6: Upload Data Configuration

### Task 1: Construct the Data Source Configuration

Use the following format to construct the data source. Note you need to make sure that the unit and type match the unit and type defined in the MindSphere Device asset

```
{  
  "configurationId": "string",  
  "dataSources": [  
    {  
      "name": "string",  
      "description": "string",  
      "dataPoints": [  
        {  
          "id": "string",  
          "name": "string",  
          "description": "string",  
          "type": "int",  
          "unit": "string",  
          "customData": {}  
        }  
      ],  
      "customData": {}  
    }  
  ]  
}
```

### Task 2: Upload Data Source Configuration

Start Postman and use the REST API call to upload data configuration

- Create the **PUT** request to upload data configuration

**PUT [https://gateway.eu1.mindsphere.io/api/agentmanagement/v3/agents/{agent\\_id}/dataSourceConfiguration](https://gateway.eu1.mindsphere.io/api/agentmanagement/v3/agents/{agent_id}/dataSourceConfiguration)**

- In the Header part of put request, use the **Agent Access Token** as authorization

**Authorization:** Bearer

```
eyJraWQiOiJrZXktawQtMSIsInR5cCI6IkpXVCIsImFsZyI6IlJTMjU2In0.eyJpc3MiOiJTQ0kiLCJzdWIiOiJmMDM3MTQyODJkMzc0Mz
gzYWJmZjFjZmQ0ODZkMjMxYyIsImF1ZCI6IkFJQU0iLCJpYXQiOjE1MjYzMdgsIm5iZiI6MTUyNjMwODgwOSwiZXhwIjoxNTI2OTEz
NjA5LCJqdGkiOiJkNzI0ZmQ3ZC1jZGQ5LTRiNGEtYTMwYi05ZjdiNzQ4YQ20TEiLCJzY29wZSI6Ik1BVCIsInRlbiI6ImNhC3N1b2UiLC
JjbG11bnRfY3J1ZGVudG1hbHNfcHJvZmlsZSI6WyJTSEFSRURFU0VDUkVUI10sInNjaGVtYXMiOlSidXJuOnNpZW11bnM6bwluZHNwaGVy
ZTp2MSJdfQ.FTMV4kEljCETuDrvbLHD5brwJwomrCKkfCzPgEbom20nvr1Mk8sAEiQBDchYS6Pk9.....
```

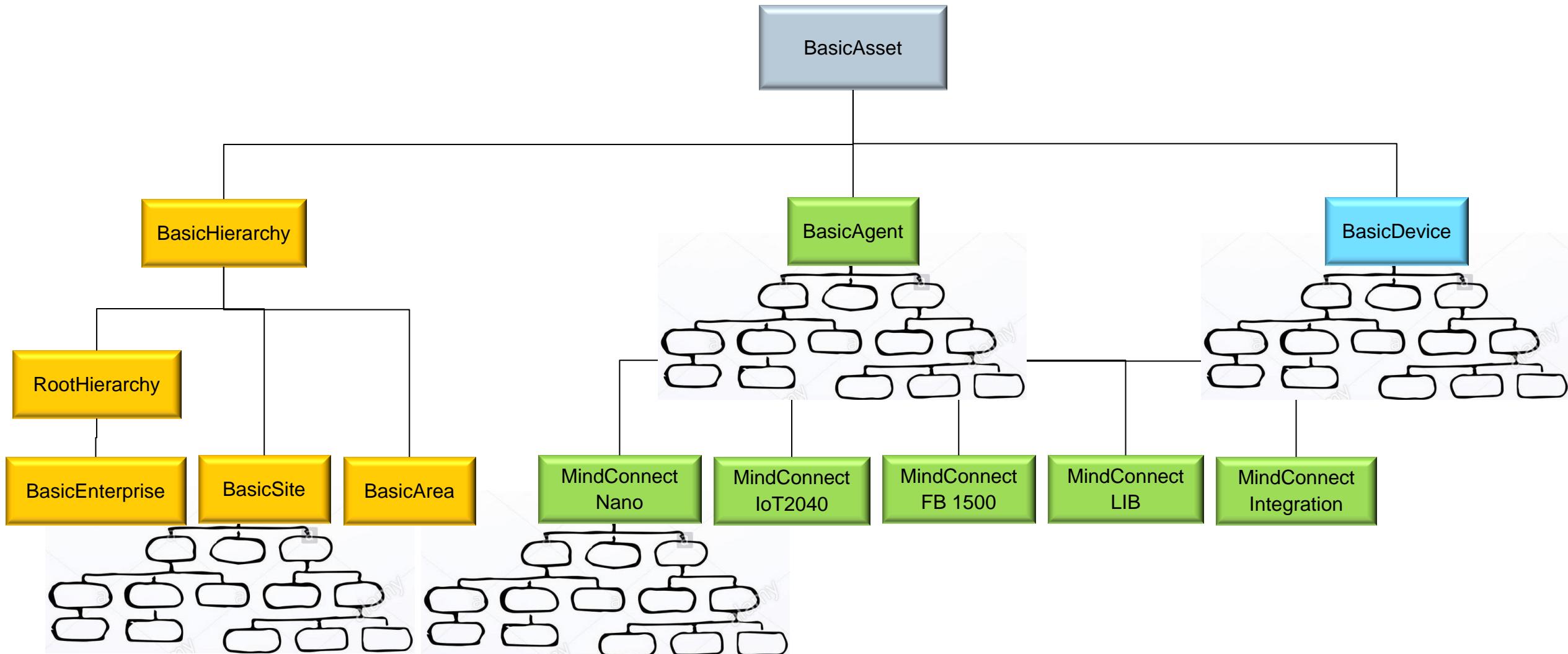
- In the body part of post request, use the data source structure created in the last step as value.
- Send the request and check the result.
- You may use the following request to get the uploaded data source. Or on MindSphere UI, go to the agent asset to check the data source configuration.

**Get [https://gateway.eu1.mindsphere.io/api/agentmanagement/v3/agents/{agent\\_id}/dataSourceConfiguration](https://gateway.eu1.mindsphere.io/api/agentmanagement/v3/agents/{agent_id}/dataSourceConfiguration)**

# MindSphere 3.0

## Asset and Hierarchy

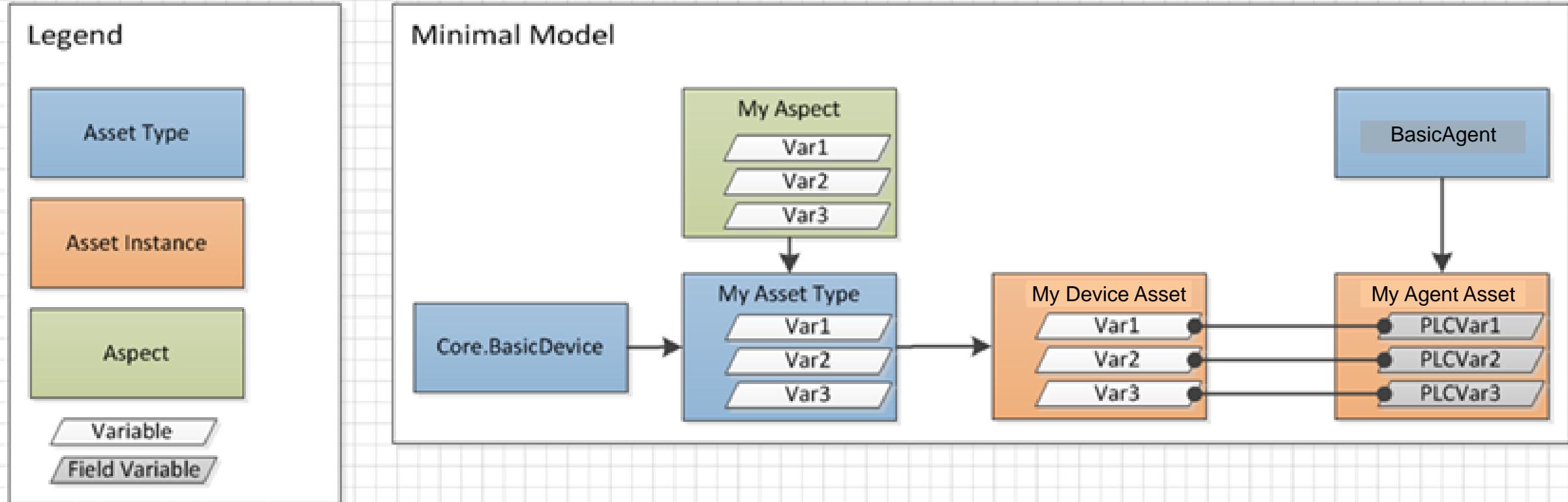
**SIEMENS**  
Ingenuity for life



# MindSphere 3.0

## Asset and Aspect

**SIEMENS**  
Ingenuity for life



T

Provisioning an Agent

Get Boarding Configuration

Onboarding the Agent

Acquiring an Access Token

DataSource Configuration

Create Data Point Mapping

Exchange Data

At this step we have a DataSource (Grouping Object for the Data Points) and a Property Set (Grouping object for the Properties) already defined in the system, a Mapping needs to be defined so that MindSphere can interpret data flowing through agent to MindSphere. DataSource configuration holds meta data about Agent side where as Property Set holds Meta data about IOT side. MindSphere needs additional configuration to map from Data Point meta to Property meta, This information is called a mapping. This mapping done one by one, for each data point to property. In other words we need to map Data Points to Properties.

In order to create a mapping **dataPointMappings**

**Endpoint: (<<\_gateway\_url>>/api/mindconnect/v3/dataPointMappings)**

HTTP/1.1 POST

Content-Type: Application/Json

Authorization: Bearer eyc..

```
{  
    "agentId": "11961bc396cd4a87a9b26b723f5b7ba0",  
    "dataPointId": "DP0001",  
    "entityId": "83e78008eadf453bae4f5c7bef3db550",  
    "propertySetName": "ElectricalProperties",  
    "propertyName": "Voltage"  
}
```

Provisioning an Agent

Get Boarding Configuration

Onboarding the Agent

Acquiring an Access Token

DataSource Configuration

Create Data Point Mapping

Exchange Data

In order to create a mapping **dataPointMappings**

**Endpoint: (<<\_gateway\_url>>/api/mindconnect/v3/dataPointMappings)**

Parameter	Description	Remarks
agentId	Unique agent id	
dataPointId	DataSource Configurations Data Point Id	
entityId	Entities Id, this Agent created from	
propertySetName	Property Set name created for this Agent. (Entity EntityType PropertySet)	Client needs to re-register with RAT if expiration occurs. Default value is 7 days
propertyName	Related property name within the property set we are binding to data point	

Provisioning an Agent

Get Boarding Configuration

Onboarding the Agent

Acquiring an Access Token

DataSource Configuration

Create Data Point Mapping

Exchange Data

If successfull MindConnect responds with 201 Created with json body that holds created mapping configuration. This step only maps one data point to one property if more mapping needed this step must be repeated.

```
{  
    "id": "4fad6258-5def-4d84-a4c2-1481b209c116",  
    "agentId": "11961bc396cd4a87a9b26b723f5b7ba0",  
    "dataPointId": "DP0001",  
    "dataPointUnit": "%",  
    "dataPointType": "INT",  
    "entityId": "83e78008eadf453bae4f5c7bef3db550",  
    "propertySetName": "ElectricalProperties",  
    "propertyName": "Voltage",  
    "propertyUnit": "%",  
    "propertyType": "INT",  
    "qualityEnabled": true  
}
```

# MindConnect API

## Exchange Data



Provisioning an Agent

Get Boarding Configuration

Onboarding the Agent

Acquiring an Access Token

DataSource Configuration

Create Data Point Mapping

Exchange Data

The MindSphere **exchange** endpoint of the [MindConnect API](#) provides the agent with the capability of uploading its data to MindSphere. This data can be of type:

- Time Series
- File Upload

The format conforms to a subset of the HTTP multipart specification, but only permits nesting of 2 levels. Below is a sample exchange message content:

```
--f0Ve5iPP2ySppIcDSR6Bak
Content-Type: multipart/related;boundary=penFL6sBQHJJUN3HA4ftqc

--penFL6sBQHJJUN3HA4ftqc
Content-Type: application/vnd.siemens.mindsphere.meta+json

{
  "type": "item",
  "version": "1.0",
  "payload": {
    "type": "standardTimeSeries",
    "version": "1.0",
    "details": {
      "configurationId": "{{_configuration_id}}"
    }
  }
}
```

multipart messages

--initial boundary

--boundary1  
**Metadata**

--boundary1  
**Payload**

--boundary1--  
--initial boundary--

# MindConnect API

## Workflow details



Provisioning an Agent

Get Boarding Configuration

Onboarding the Agent

Acquiring an Access Token

DataSource Configuration

Create Data Point Mapping

Exchange Data

```
--penFL6sBQHJJUN3HA4ftqC
Content-Type: application/json

[
  {
    "timestamp": "2017-02-01T08:30:03.780Z",
    "values": [
      {
        "dataPointId": "{{_datapoint_id_1}}",
        "value": "9856",
        "qualityCode": "0"
      },
      {
        "dataPointId": "{{_datapoint_id_2}}",
        "value": "3766",
        "qualityCode": "0"
      }
    ]
  }
]
--penFL6sBQHJJUN3HA4ftqC--
--f0Ve5iPP2ySppIcDSR6Bak--
```

multipart messages

```
--initial boundary

--boundary1
Metadata

--boundary1
Payload

--boundary1-
--initial boundary--
```

# MindConnect API

## Workflow details



Provisioning an Agent

Get Boarding Configuration

Onboarding the Agent

Acquiring an Access Token

DataSource Configuration

Create Data Point Mapping

Exchange Data

The MindSphere **exchange** endpoint of the [MindConnect API](#) provides the agent with the capability of uploading its data to MindSphere. This data can be of type:

- Time Series
- File Upload

The format conforms to a subset of the HTTP multipart specification, but only permits nesting of 2 levels. Below is a sample exchange message content:

```
--f0Ve5iPP2ySppIcDSR6Bak
Content-Type: multipart/related;boundary=penFL6sBQHJJUN3HA4ftqC

--penFL6sBQHJJUN3HA4ftqC
Content-Type: application/vnd.siemens.mindsphere.meta+json
{
  "type": "item",
  "version": "1.0",
  "payload": {
    "type": "standardTimeSeries",
    "version": "1.0",
    "details": {
      "configurationId": "{{_configuration_id}}"
    }
  }
}
```

Provisioning an Agent

Get Boarding Configuration

Onboarding the Agent

Acquiring an Access Token

DataSource Configuration

Create Data Point Mapping

Exchange Data

Create data mapping

Prepare the time series to be uploaded

Exchange Data



Provisioning an Agent

Get Boarding Configuration

Onboarding the Agent

Acquiring an Access Token

DataSource Configuration

Create Data Point Mapping

Exchange Data

**Task 1: Create data mapping**

**Task 2: Prepare the time series to be uploaded**

**Task 3: Exchange Data**

# Hands-on



### Task 1: Create Data Mapping

Start Postman and use the REST API call to create the Data Mapping

- Create the **POST** request to upload data

**Post** <https://gateway.eu1.mindsphere.iCo/api/mindconnect/v3/dataPointMappings>

- In the Body part of put request, use the following structure to create data mapping

```
HTTP/1.1 POST  
Content-Type: Application/Json  
Authorization: Bearer eyc..  
{  
    "agentId": "11961bc396cd4a87a9b26b723f5b7ba0",  
    "dataPointId": "DP0001",  
    "entityId": "83e78008eadf453bae4f5c7bef3db550",  
    "propertySetName": "ElectricalProperties",  
    "propertyName": "Voltage"  
}
```

Optionally you can use the MindSphere UI to create the data mapping. Note you need both Agent Asset and Device Asset to create the mapping

### Task 2: Construct the Time Series Data to be uploaded

Use the following format to construct the time series data:

```
--f0Ve5iPP2ySppIcDSR6Bak
Content-Type: multipart/related;boundary=penFL6sBQHJJUN3HA4ftqC

--penFL6sBQHJJUN3HA4ftqC
Content-Type: application/vnd.siemens.mindsphere.meta+json

{
  "type": "item",
  "version": "1.0",
  "payload": {
    "type": "standardTimeSeries",
    "version": "1.0",
    "details": {
      "configurationId": "{{_configuration_id}}"
    }
  }
}
```

multipart messages

```
--initial boundary
--boundary1
Metadata
--boundary1
Payload
--boundary1--
--initial boundary--
```

## Hands-On Exercise 7: Exchange Data

Use the following format to construct the time series data.

```
--penFL6sBQHJJUN3HA4ftqC
Content-Type: application/json

[
  {
    "timestamp": "2017-02-01T08:30:03.780Z",
    "values": [
      {
        "dataPointId": "{{_datapoint_id_1}}",
        "value": "9856",
        "qualityCode": "0"
      },
      {
        "dataPointId": "{{_datapoint_id_2}}",
        "value": "3766",
        "qualityCode": "0"
      }
    ]
  }
]

--penFL6sBQHJJUN3HA4ftqC--
--f0Ve5iPP2ySppIcDSR6Bak--
```

multipart messages

```
--initial boundary

--boundary1
Metadata

--boundary1
Payload

--boundary1-
--initial boundary--
```

### Task 3: Exchange Timeseries Data

Start Postman and use the REST API call to exchange data

- Create the **POST** request to upload data

**Post** <https://gateway.eu1.mindsphere.io/api/mindconnect/v3/exchange>

- In the Header part of put request, use the **Agent Access Token** as authorization

**Authorization:** Bearer

```
eyJraWQiOiJrZXktawQtMSIsInR5cCI6IkpXVCIsImFsZyI6IlJTMjU2In0.eyJpc3MiOiJTQ0kiLCJzdWIiOiJmMDM3MTQyODJkMzc0Mz
gzYWJmZjFjZmQ0ODZkMjMxYyIsImF1ZCI6IkFJQU0iLCJpYXQiOjE1MjYzMdgsIM5iZiI6MTUyNjMwODgwOSwiZXhwIjoxNTI20TEz
NjA5LCJqdGkiOiJkNzI0ZmQ3ZC1jZGQ5LTRiNGEtYTMwYi05ZjdiNzQ4YQ20TEiLCJzY29wZSI6Ik1BVCIsInRlbii6ImNhC3N1b2UiLC
JjbG1lbnRfY3J1ZGVudG1hbHNfcHJvZmlsZSI6WyJTSEFSRURFU0VDUkVUI10sInNjaGVtYXMi0lsidXJuOnNpZW1lbnM6bwluZHNwaGVy
ZTp2MSJdfQ.FTMV4kEljCETuDrvbLHD5brwJwomrCKkfCzPgEbom20nvrlMk8sAEiQBDchYS6Pk9.....
```

- In the body part of post request, use the data structure created in the last step as value.
- Send the request and check the result with MindApp **Fleet Manager**
- You may use the following request to get the time series.

**Get** [https://gateway.eu1.mindsphere.io/api/iottimeseries/v3/timeseries/{asset\\_id}/{aspect}?from=2018-07-10T00:00:00Z&to=2018-07-23T23:50:00Z](https://gateway.eu1.mindsphere.io/api/iottimeseries/v3/timeseries/{asset_id}/{aspect}?from=2018-07-10T00:00:00Z&to=2018-07-23T23:50:00Z)

# MindSphere Connectivity Summary & Feedback 2018

## SIOS: Siemens Industry Online Support

Provide the last technical documents from Fact Sheets to Quick Start Guides and Customer Documents

The screenshot shows the Siemens Industry Online Support (mySupport) interface. The top navigation bar includes links for 'Industry Online Support Deutschland', 'English', 'Kontakt', 'Hilfe', 'Support Request', and 'Site Explorer'. The main content area is titled 'Industry Online Support mySupport' and shows a user profile for 'Alexander Foerster'. On the left, there's a sidebar with a navigation menu containing items like 'Persönliche Nachrichten', 'Anfragen' (selected), 'Produkte', 'Benachrichtigungen', 'Filter', 'Favoriten', 'Tagging', 'Zuletzt gesehene Beiträge', 'Dokumentation', 'Persönliche Daten', 'CAX-Daten', and 'Extranets'. The central part of the screen displays a list of 'Meine Anfragen' (My Requests). The first request listed is for 'CALVATIS; Reconnection der MCN nicht automatisch, Daten werden nicht in VA angezeigt' (SR-Number: 1-4470743224, Status: In work, Created: 28.06.2016 07:47). The second request is for 'CALVATIS; Reconnection der MCN nicht automatisch, Daten werden nicht in VA angezeigt' (SR-Number: 1-4469723096, Status: HQ Requested, Created: 27.06.2016 10:37).

<https://support.industry.siemens.com>



The screenshot shows the 'MindSphere - Getting Started' document. It features a table of contents on the right side with the following sections and page numbers: Preface (1), Safety Notes (1), Getting Started with MindSphere (2), Prepare MindConnect Nano for MindSphere (3), Onboard MindConnect Nano to MindSphere (4), Configure Data Collection (5), Perform visual analysis on an Asset (6), Appendix (7), and Glossary (8). The document is valid from V1.1 Version and was published on 06/2016.

# MindSphere

## Connect to Us



<b>Website</b>	<a href="http://www.mindsphere.io">www.mindsphere.io</a>
<b>MindSphere Community</b>	<a href="https://community.plm.automation.siemens.com/t5/MindSphere/ct-p/MindSphere">https://community.plm.automation.siemens.com/t5/MindSphere/ct-p/MindSphere</a>
<b>MindSphere General</b>	<a href="mailto:contact@mindsphere.io">contact@mindsphere.io</a>
<b>MindSphere Support Team</b>	<p>Call Center: +49 (0)911 895 7 272 (Germany)</p> <p>Online Support: <a href="http://www.siemens.com/automation/service&amp;support">http://www.siemens.com/automation/service&amp;support</a></p> <p>Support Request: <a href="http://www.siemens.com/automation/support-request">http://www.siemens.com/automation/support-request</a></p>
<b>MindSphere Academy</b>	<a href="mailto:training@mindsphere.io">training@mindsphere.io</a>

# Thank you.