OST
Ostschweizer
Fachhochschule

Digital Manufacturing

04 – Cloud Computing

Lukas Kretschmar, MSc FHO in Engineering (ICT & BEP)

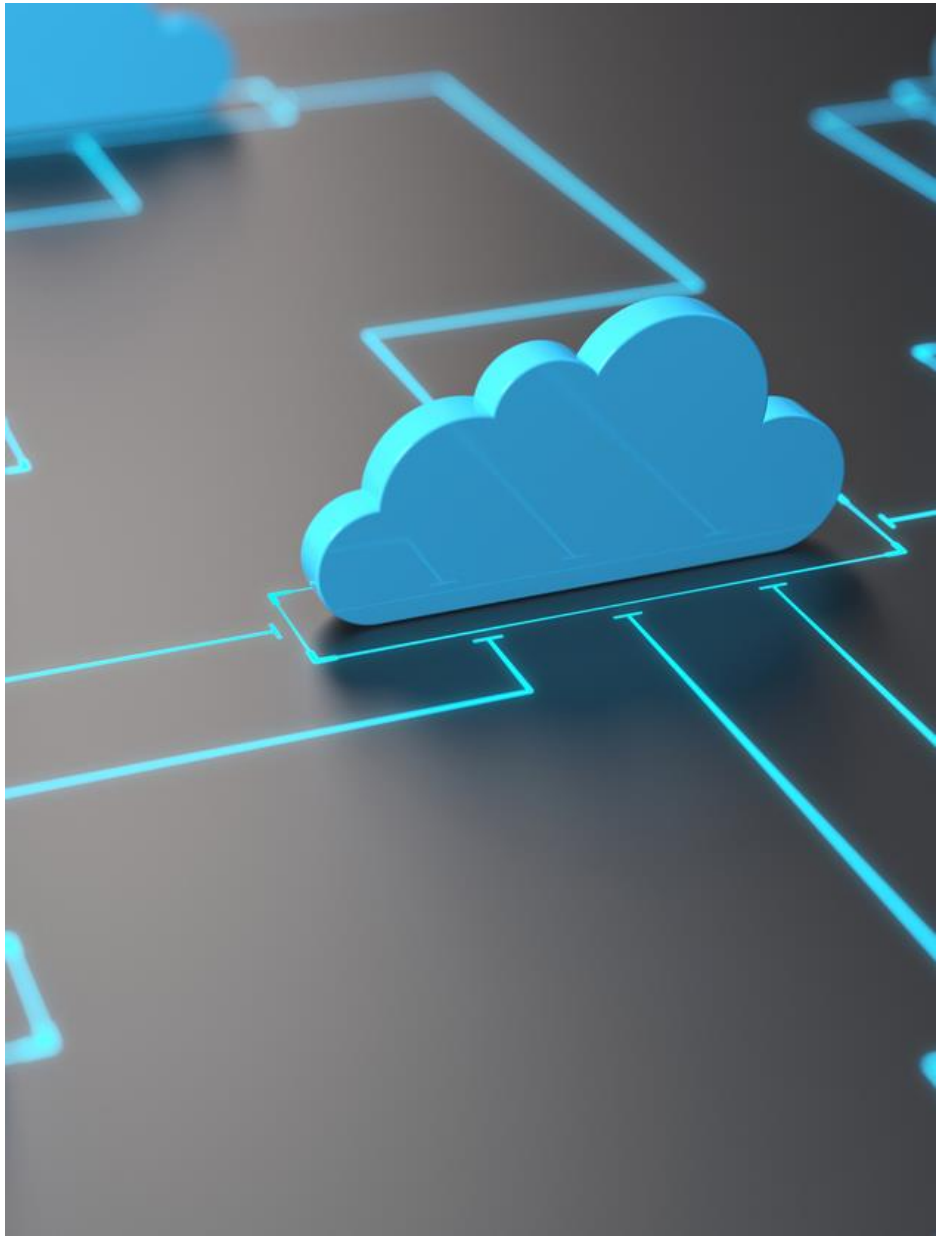
15. November 2021

Industrial Engineering (OST-RJ)

Startup

Today's Objectives

- You can use a REST API
- You can identify the service concepts of cloud providers
- You know the basic pros and cons about the cloud
- You have a basic understanding of key security concepts
- You can identify the parts of JWT



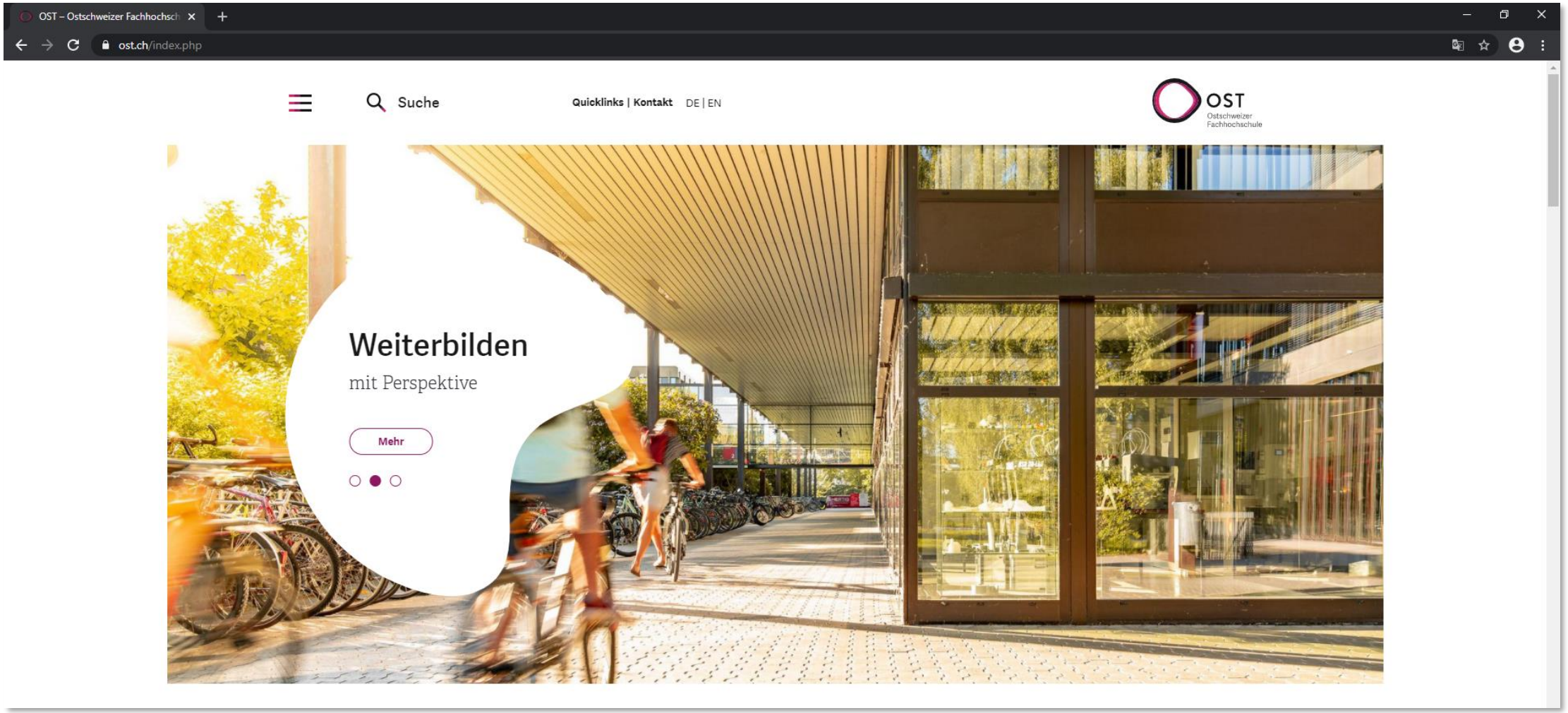
OST

How the Internet works

200 OK

How the Internet works

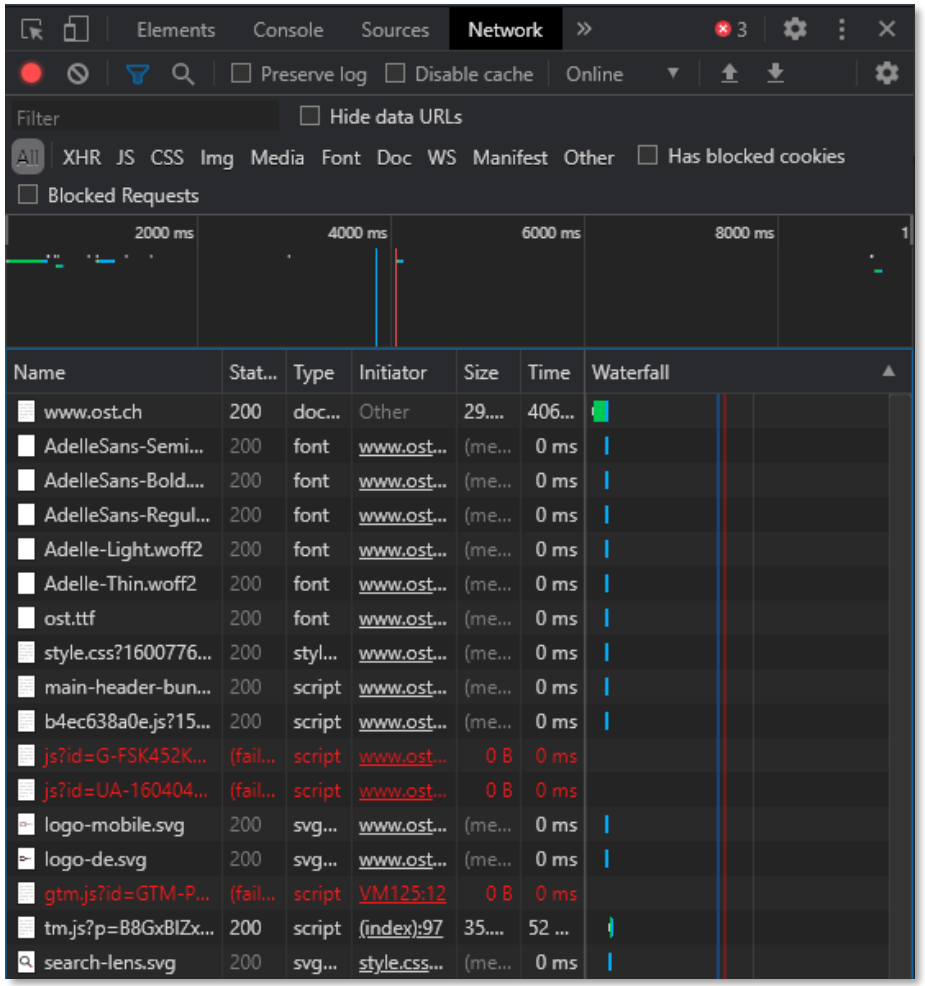
http://www.hsr.ch → https://www.ost.ch



OST

Hands-On: http://www.hsr.ch → https://www.ost.ch

- Let's see what happens when we open http://www.hsr.ch
- And land on https://www.ost.ch

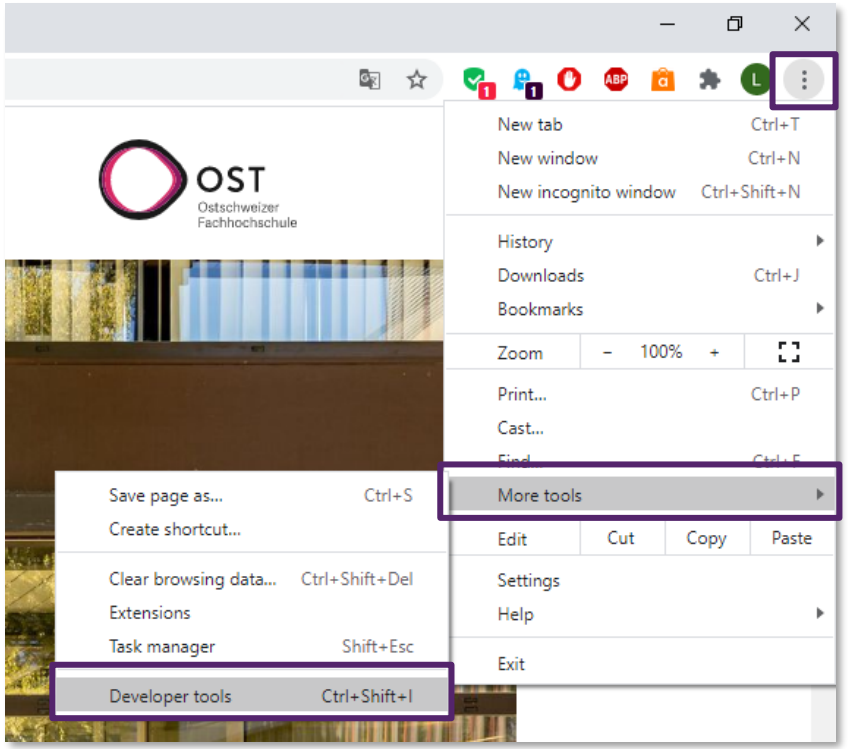


- To see the communication, we need to get behind the scenes
  - Nowadays, every browser is shipped with some features to see the communication
  - Usually, they are called *Developer Tools* or something in that direction

Hands-On: http://www.hsr.ch → https://www.ost.ch

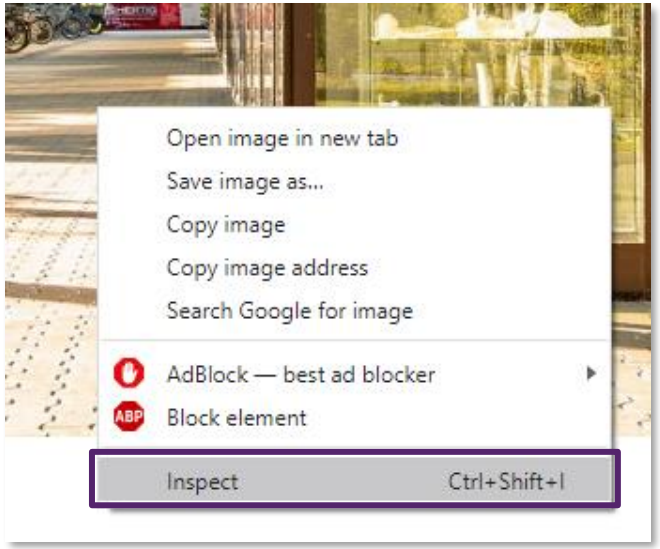
Open them through the menu

- Usually, they use the keyboard combination of *Shift + Ctrl + I*

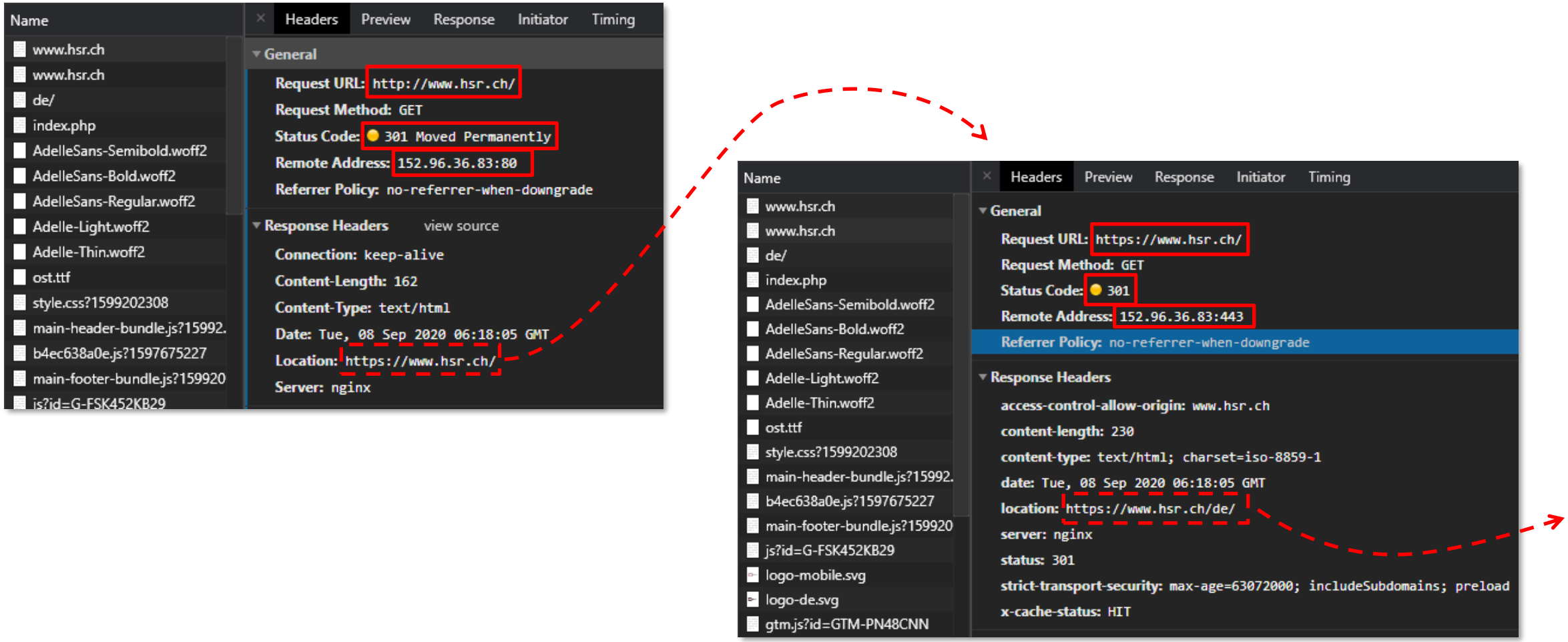


Inspect a websites content

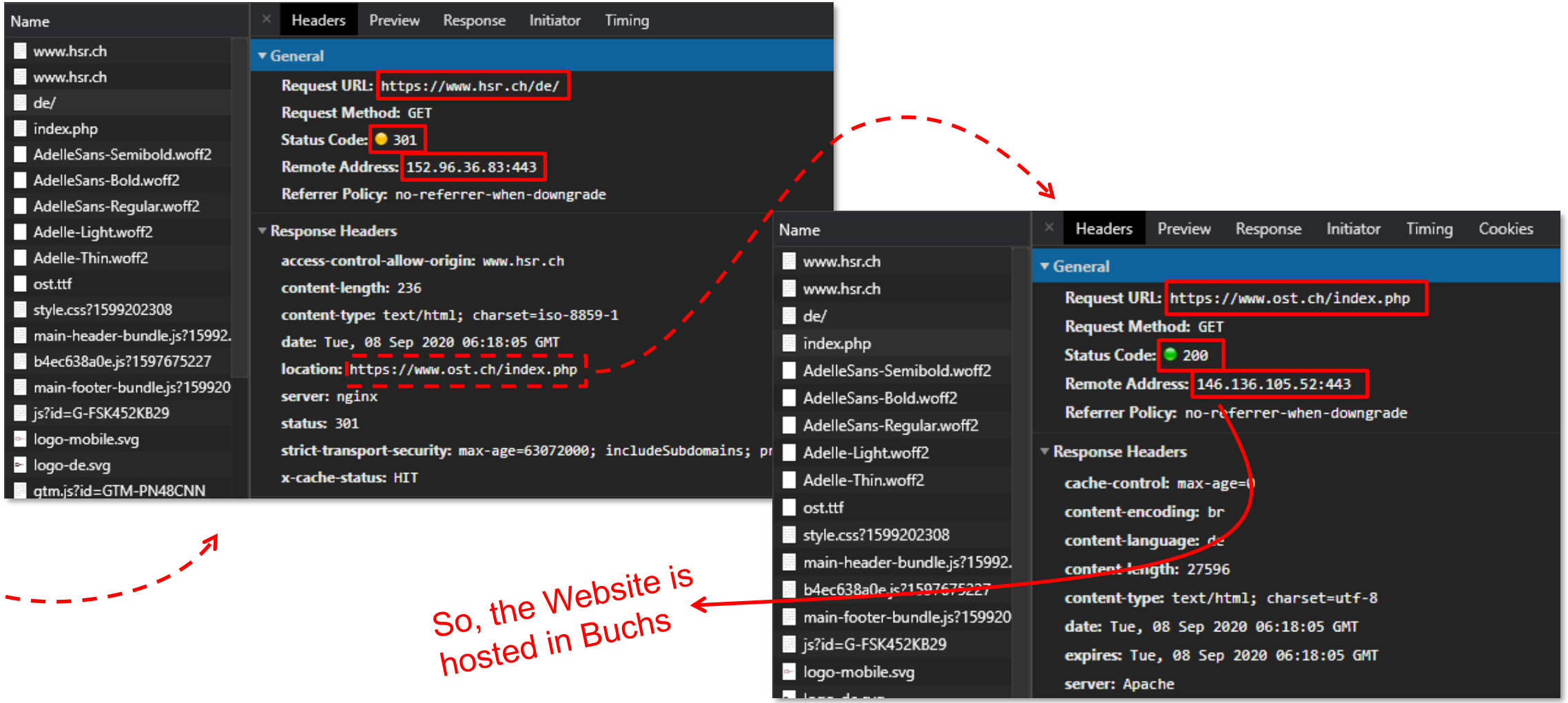
- Click on the page with the right mouse button
- Use the menu item Inspect or Inspect Element



Hands-On: http://www.hsr.ch → https://www.ost.ch

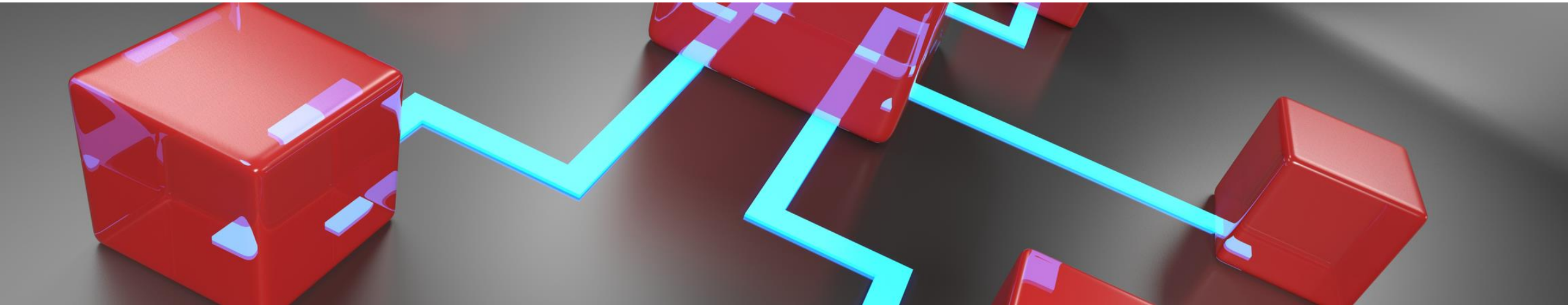


Hands-On: http://www.hsr.ch → https://www.ost.ch



API (Application Programming Interface)

Talking to Services

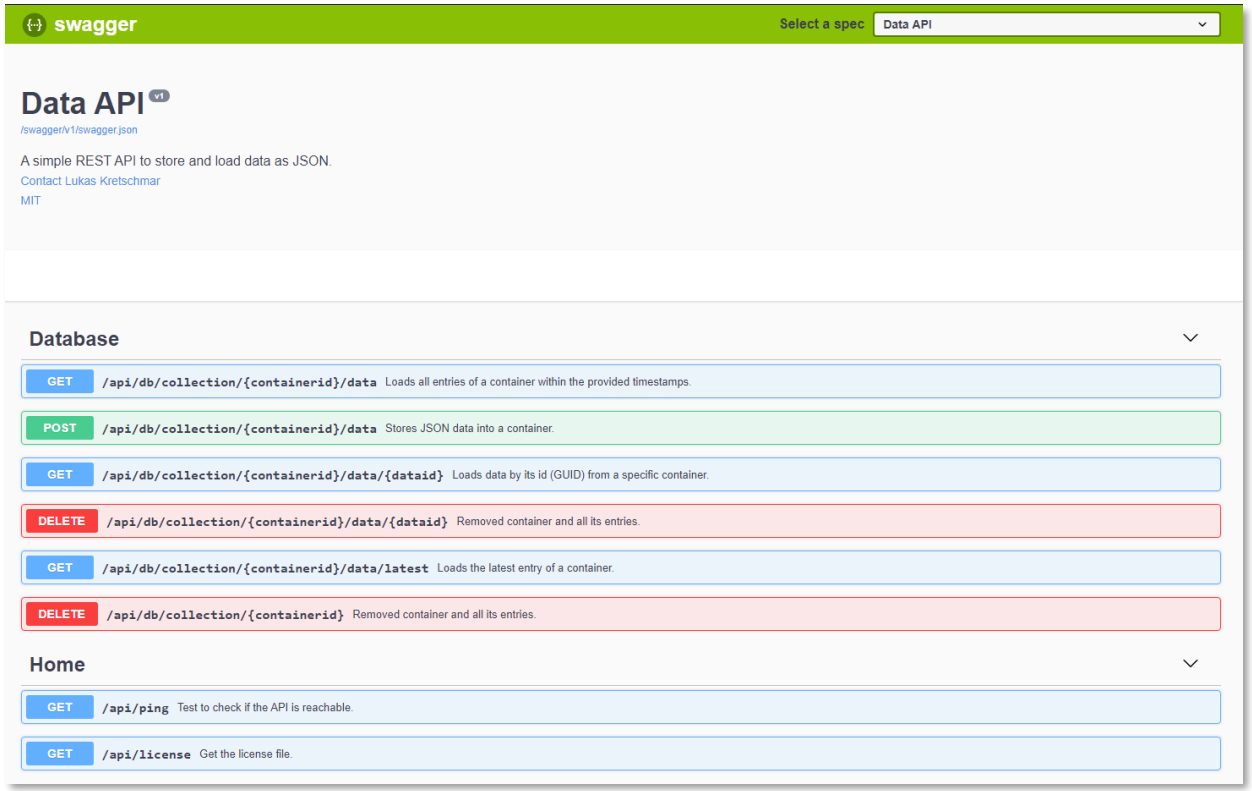


What's an API?

- Don't be confused by the term programming
  - You don't program anything
  - Interface is the term to focus on
- APIs are the entry point to applications
  - Applications offer features through their APIs
  - Applications work together by using their APIs
- APIs are for applications, what GUIs (graphical user interface) are for humans
- An API allows you to
  - Access & manipulate data
  - Trigger actions
  - Hide the implementation
  - Limit the possibilities what you can do with an application
  - Enable modularization of systems
    - SOA (Service-oriented architecture)
    - Applications talking to other applications that talk to other applications

REST

- Representational State Transfer (REST)
  - Simple client-service architecture
  - Builds upon the HTTP protocol
  - Uses JSON
  - Does not keep a state
  - Endpoints can change
    - Subsequent calls do not necessarily need to be handled by the same instance
- Current state-of-the-art type of webservices



https://en.wikipedia.org/wiki/Representational_state_transfer

SOAP

Acronym has been dropped in v1.2

- “Simple Object Access Protocol”
  - Protocol independent
  - Service description and messages are defined in XML
    - XML generates overhead
    - But distinguishes between data and metadata
  - Service description can be exchanged (Metadata Exchange)
    - Service can be generated on the client side
    - Service description is also in XML

- More common between applications within an intranet (local network)
  - OPC-UA (OPC Unified Architecture)

https://en.wikipedia.org/wiki/SOAP

OPC-UA (Open Platform Communications Unified Architecture)

- OPC Unified Architecture
  - Open standard for M2M communication
  - Cross platform
  - Using SOAP
    - Integrated Security
    - Integral information model
      - Faults are part of the model
  - Supported protocols
    - opc.tcp://
    - http://

Binary

Plaintext

- It's really complex
  - Everything is specified
  - System integration is easier
  - But we still rely on the implementation and usage of the vendor

- Request & Response look the same as with HTTP
  - Content is in XML

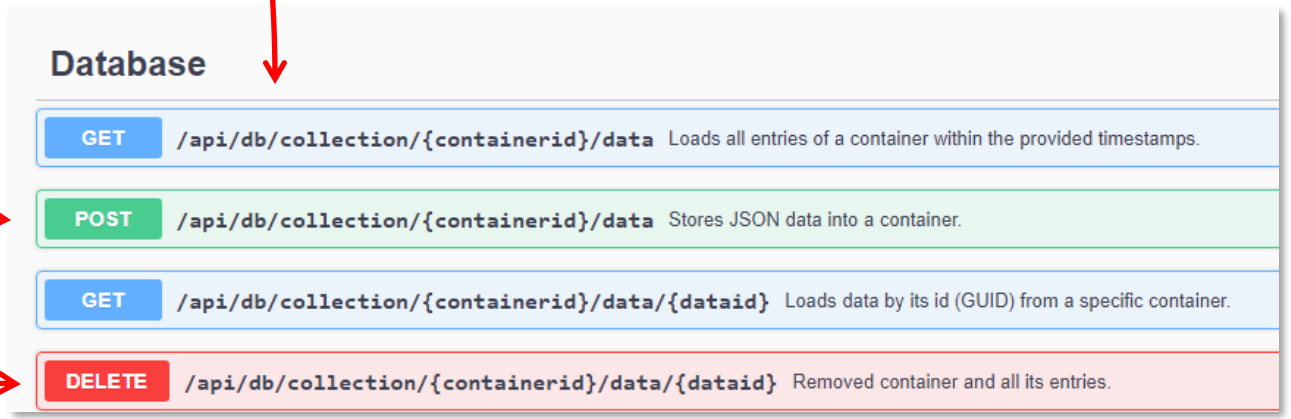
```xml
<?xml version="1.0"?>
<soap:Envelope
        xmlns:soap="http://www.w3.org/2003/05/soap-envelope"
        xmlns:m="http://www.example.org">
    <soap:Header>
    </soap:Header>
    <soap:Body>
    </soap:Body>
</soap:Envelope>
```

https://en.wikipedia.org/wiki/OPC_Unified_Architecture

OST

HTTP Methods

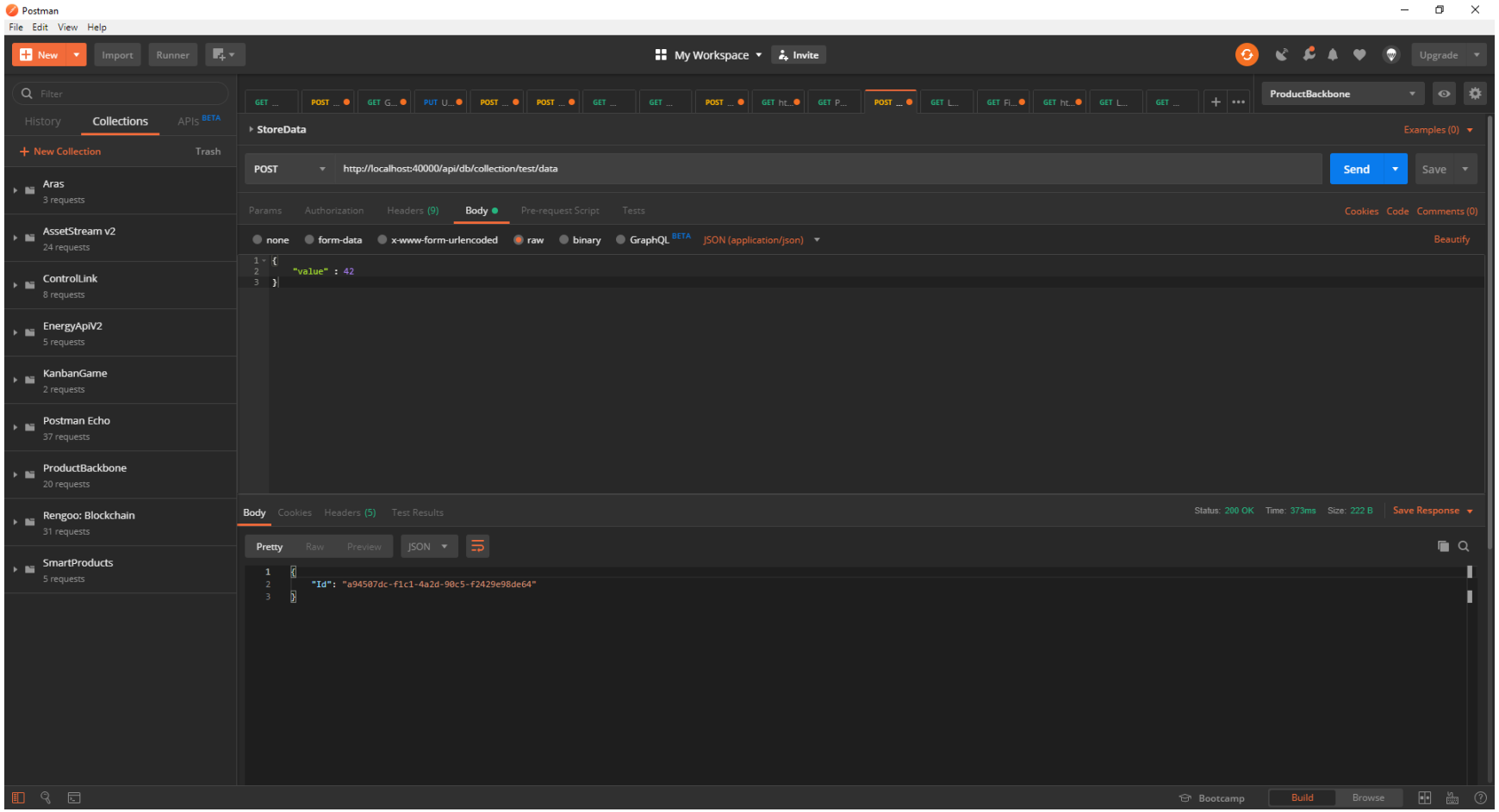
- GET
  - Requesting data from an URI
- POST
  - Store data beneath the URI
- PUT
  - Store/replace data at URI
- DELETE
  - Remove data at URI



There are some more, but not relevant here

OST

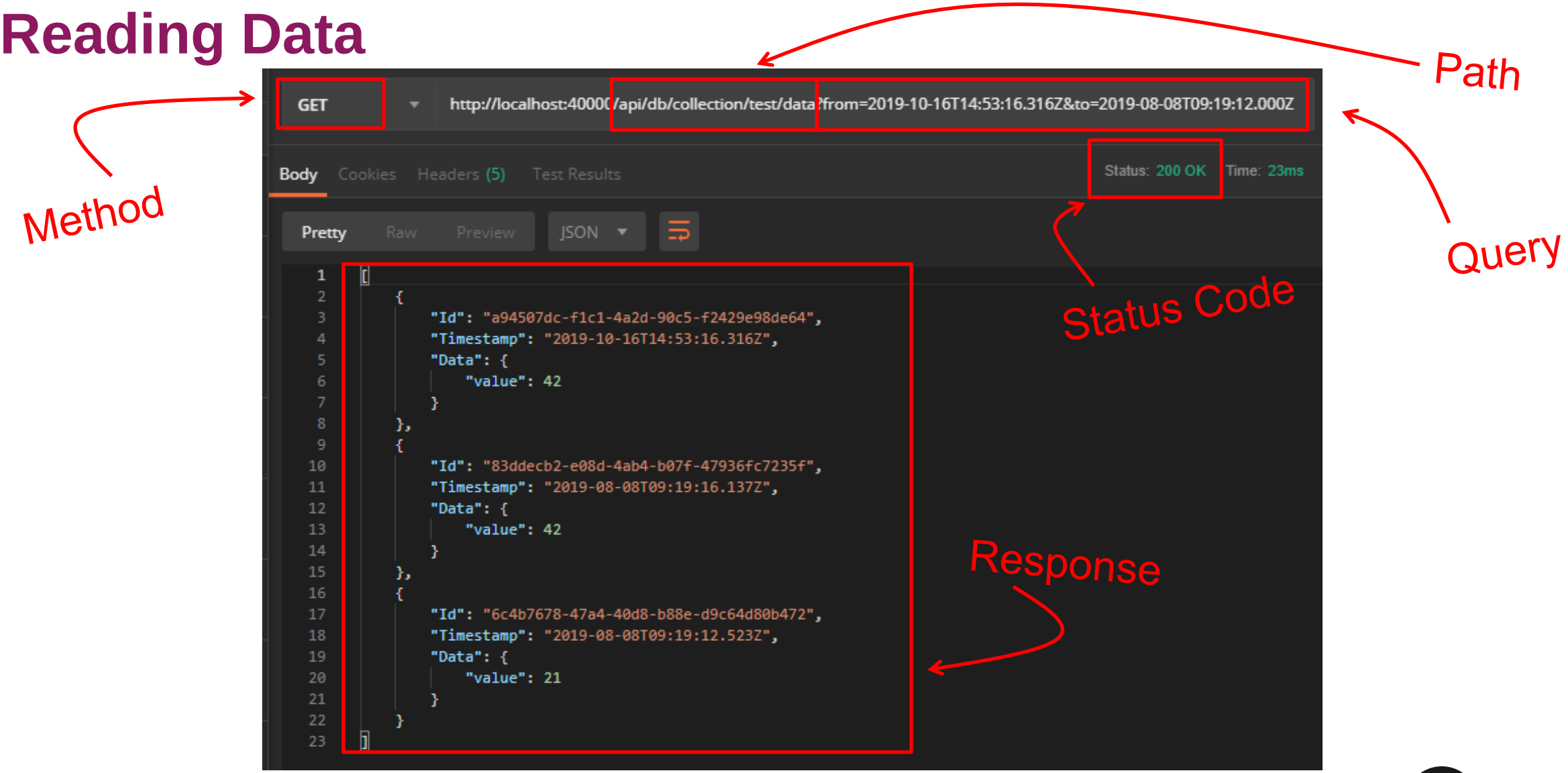
Postman



https://www.postman.com

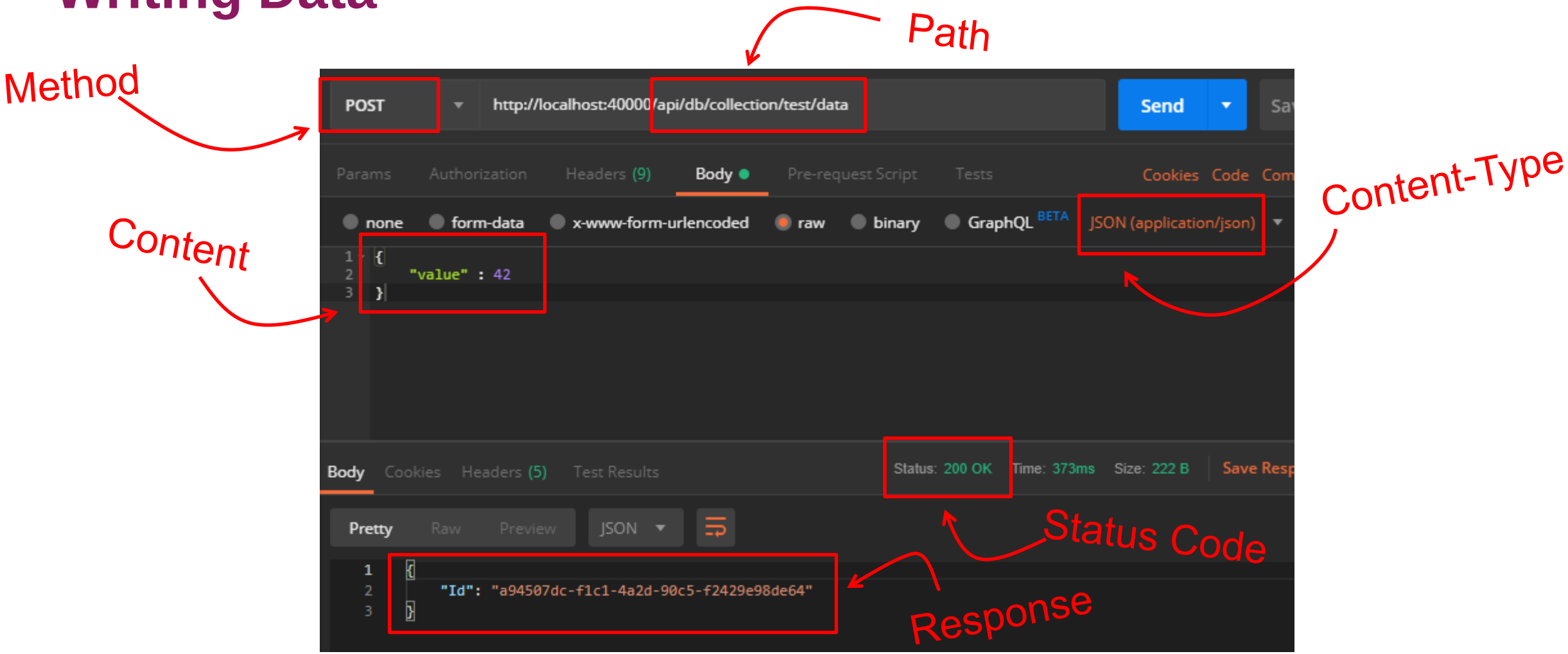
OST

Reading Data



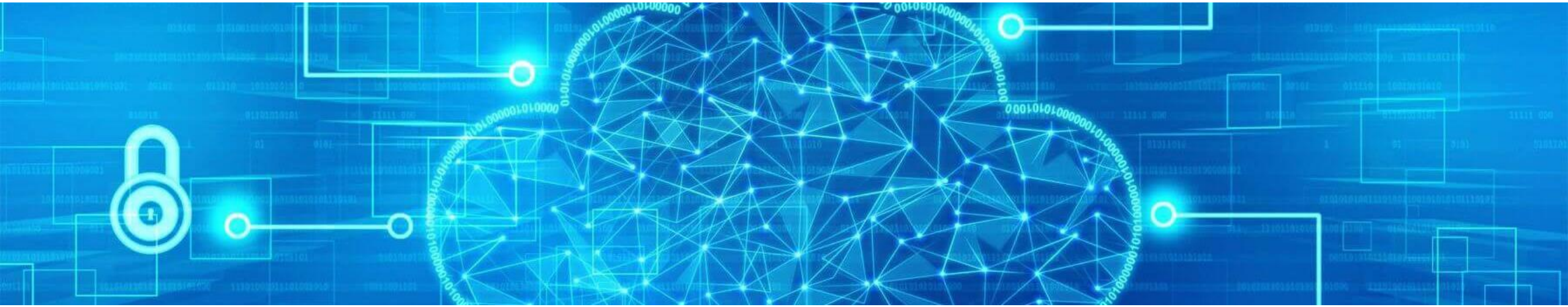
OST

Writing Data

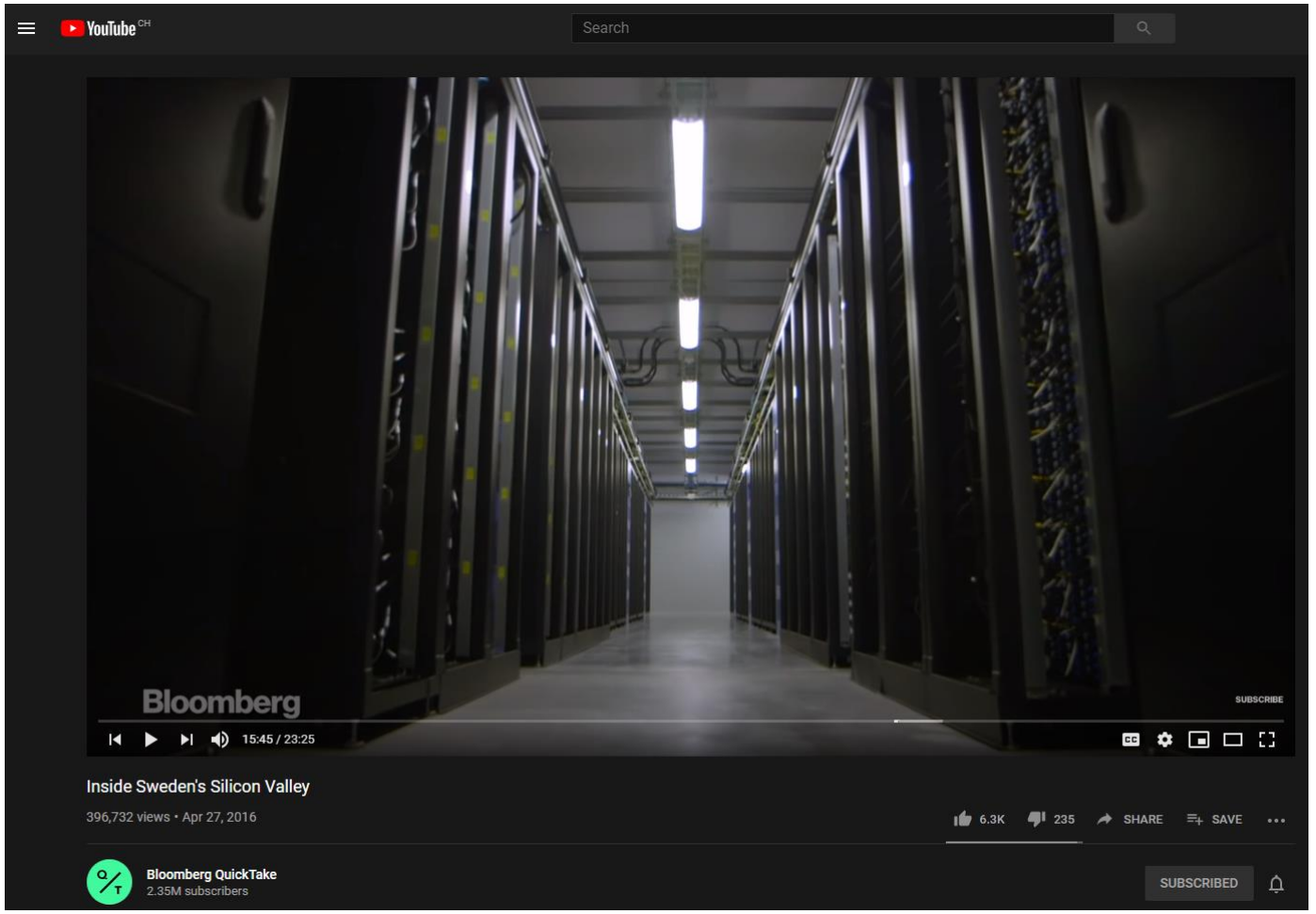


The Cloud

It's just someone else's Computer



Hello Cloud



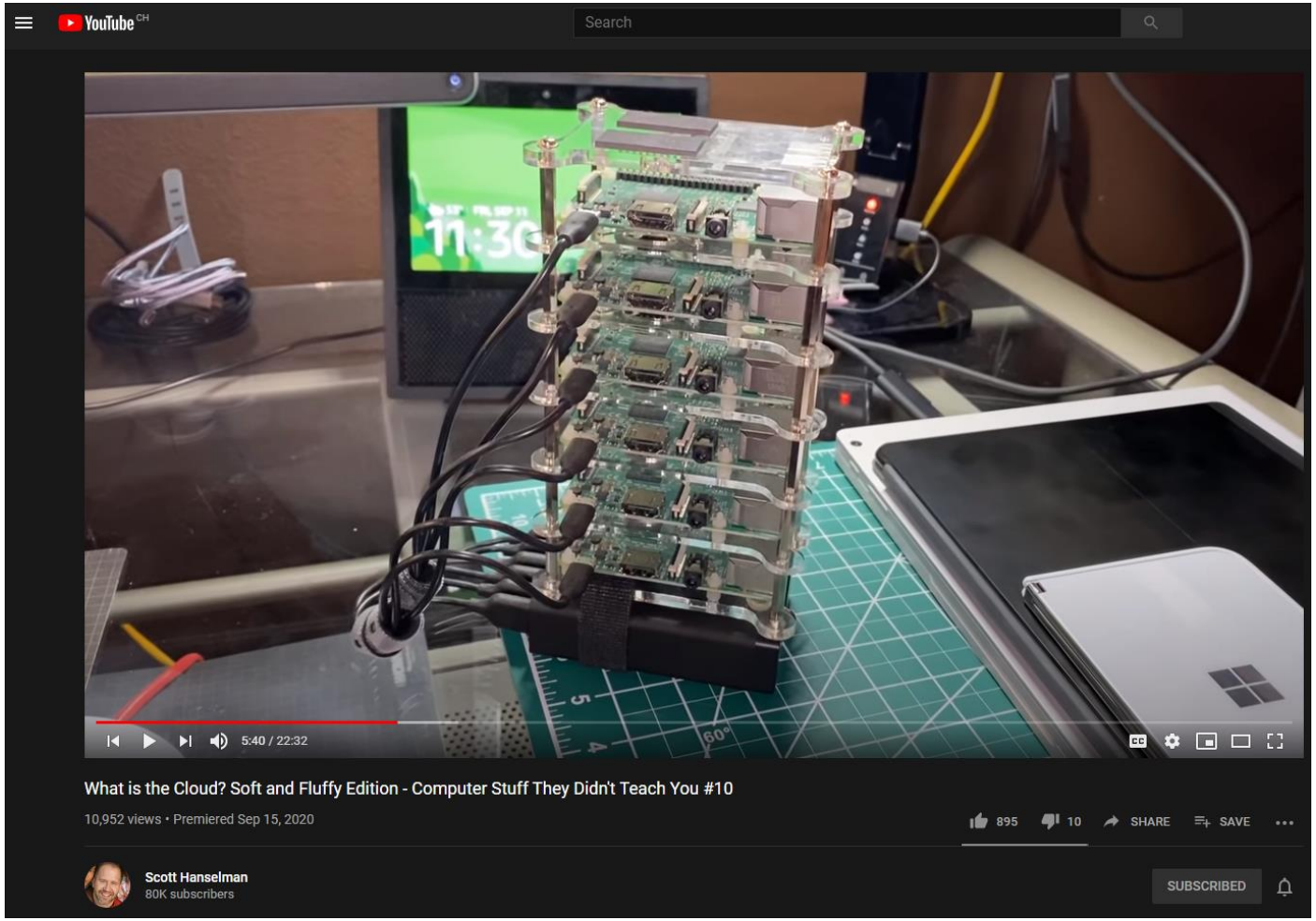
For those who are interested

https://youtu.be/d915p1aXlkE?list=PLqq4LnWs3olU-bP2R9uD8YXbt02JjocOk&t=858

Bloomberg Hello World Playlist: https://www.youtube.com/playlist?list=PLqq4LnWs3olU-bP2R9uD8YXbt02JjocOk

OST

Introduction to the Cloud by Scott Hanselman

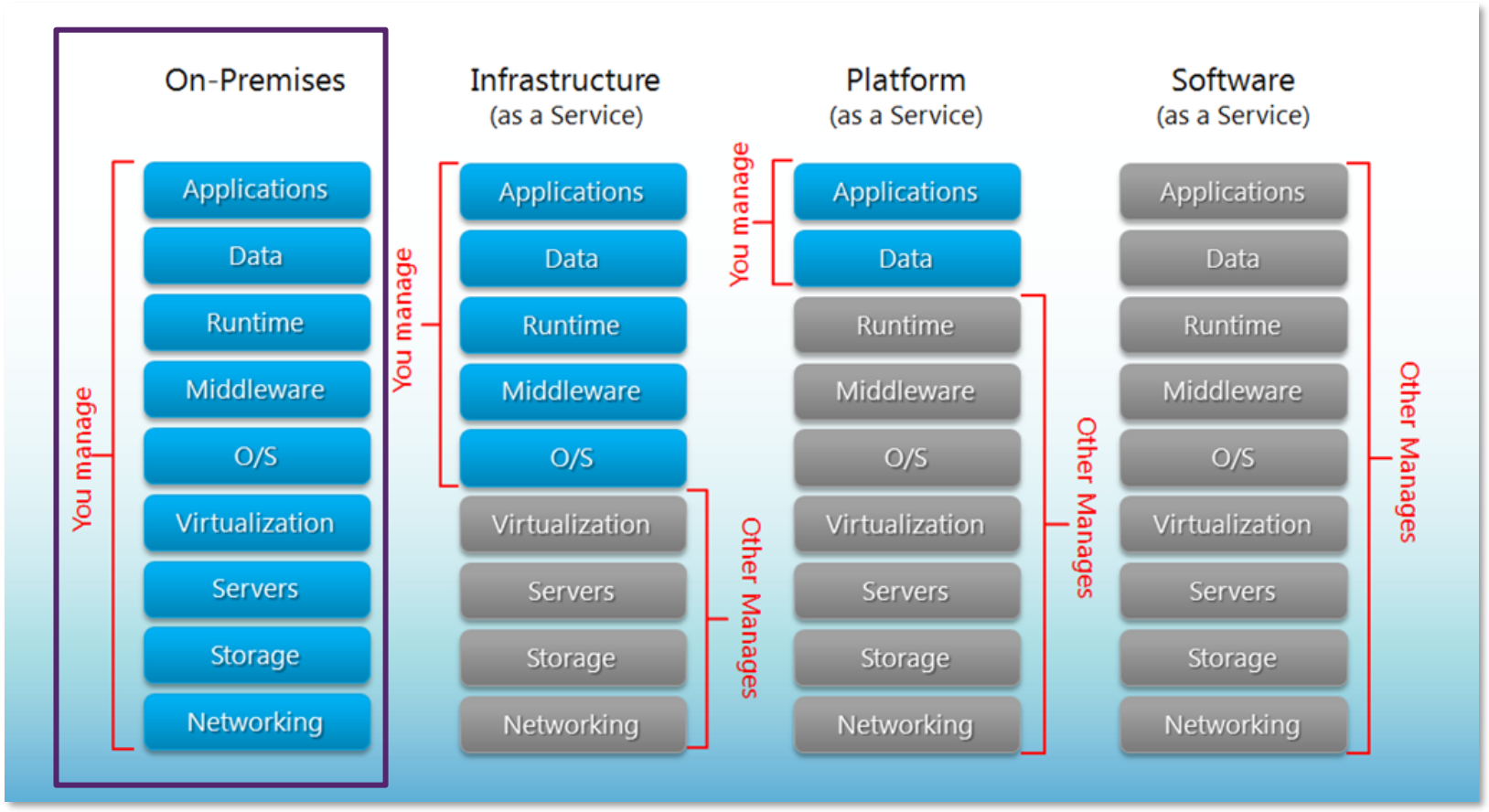


For self-study if interested

https://www.youtube.com/watch?v=BO6jvQ88ICQ

OST

XaaS (Everything as a Service)



Microsoft Azure

aws

MindSphere

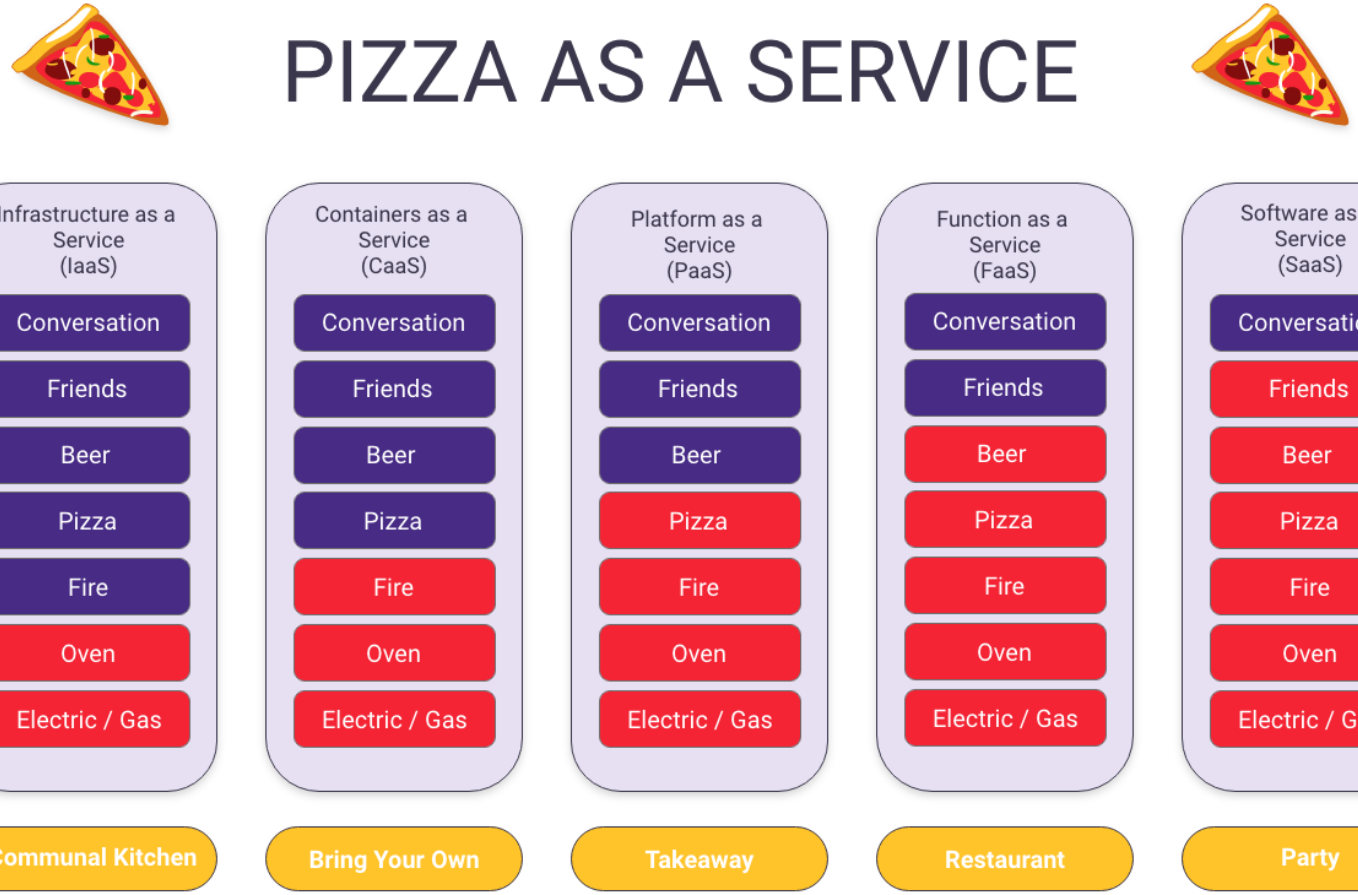
NETFLIX

Dropbox

OST

XaaS (Everything as a Service)

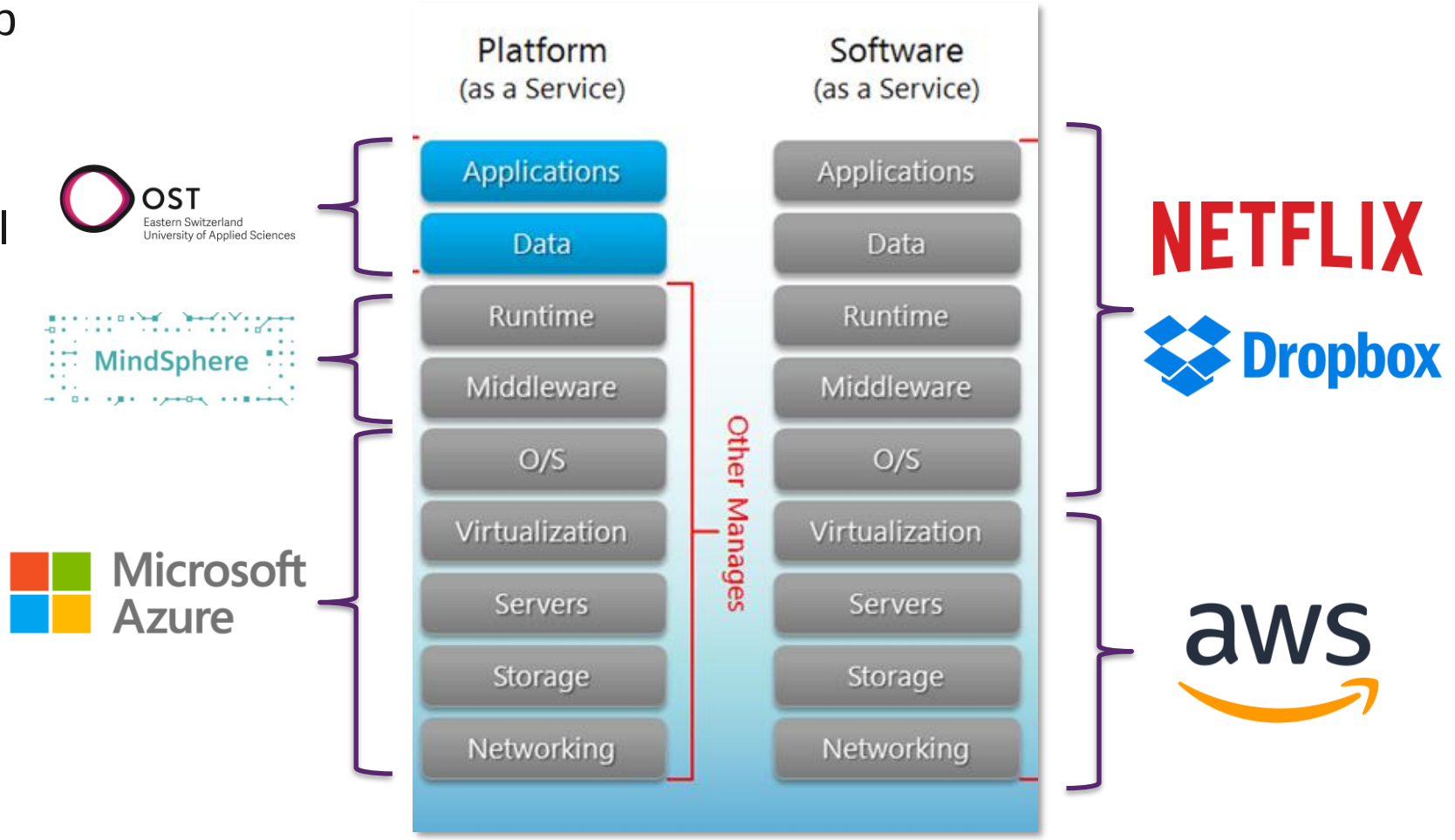
Well - it also works like this



OST

XaaS (Everything as a Service)

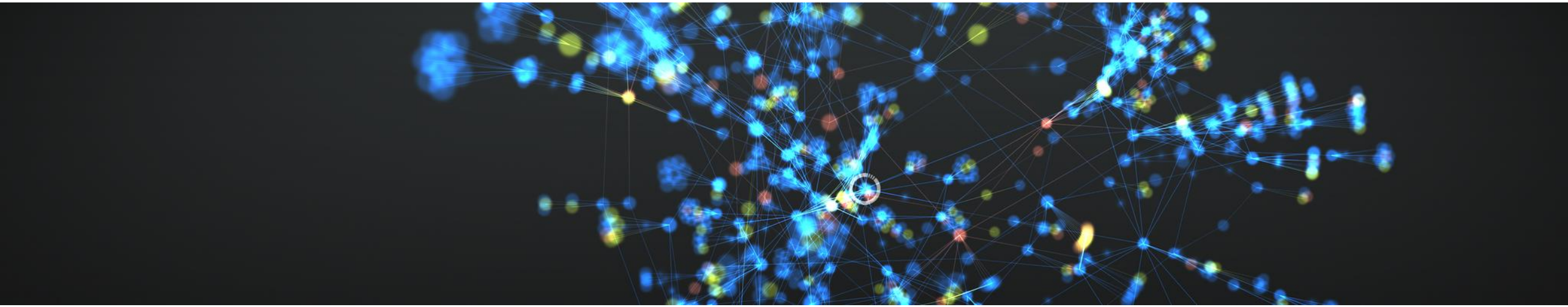
- Services can build on top of each other
- Not every provider has every layer under control
  - May be managed by another provider



OST

Data Modeling for MindSphere

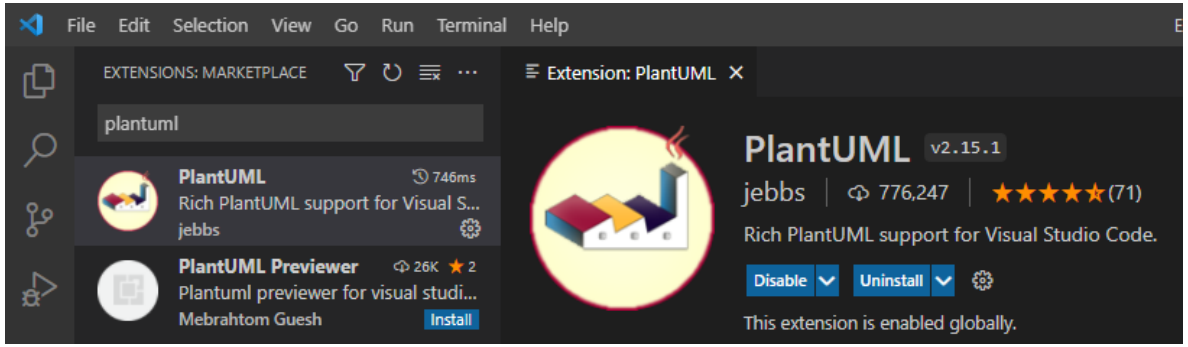
We need some Structure

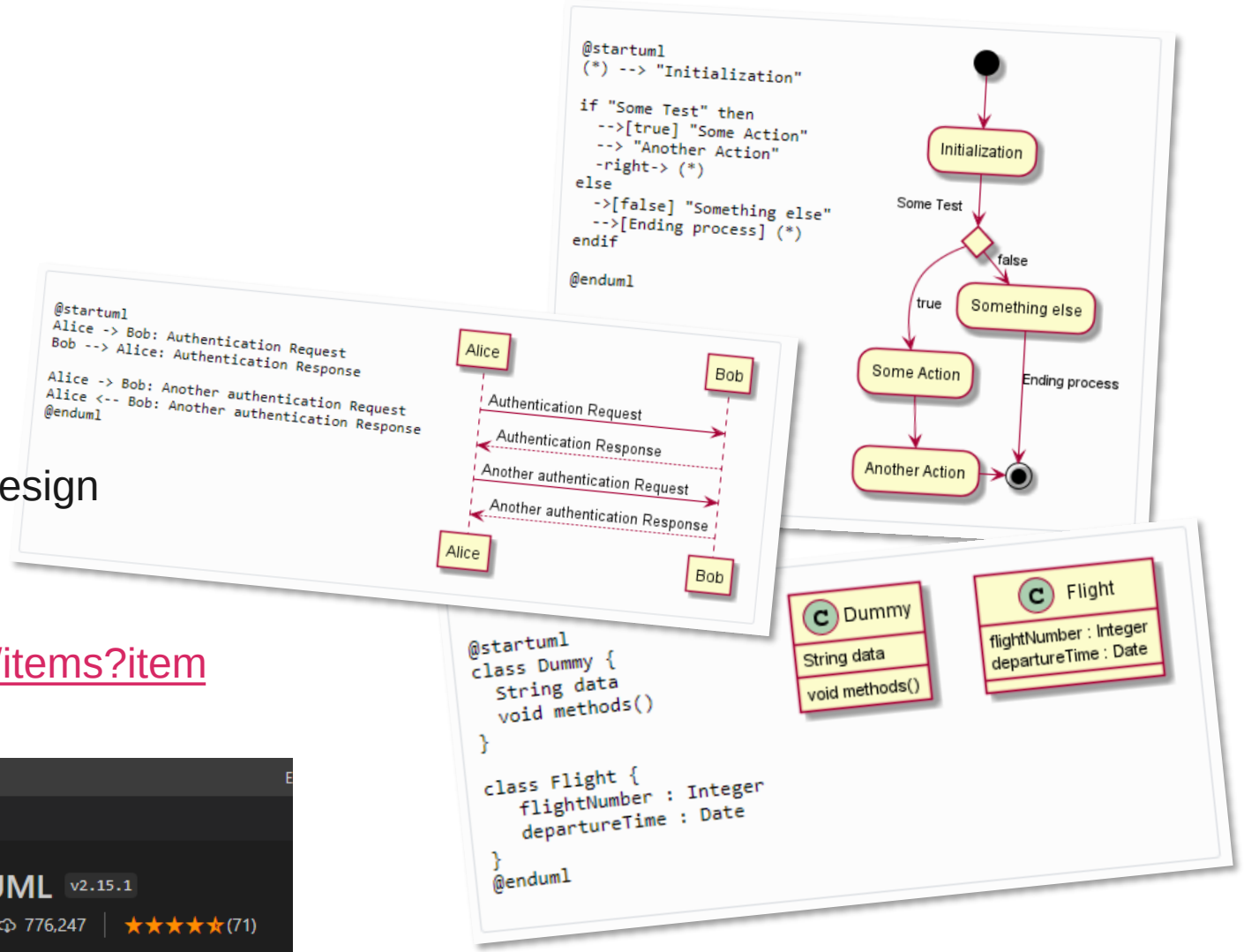


OST

PlantUML

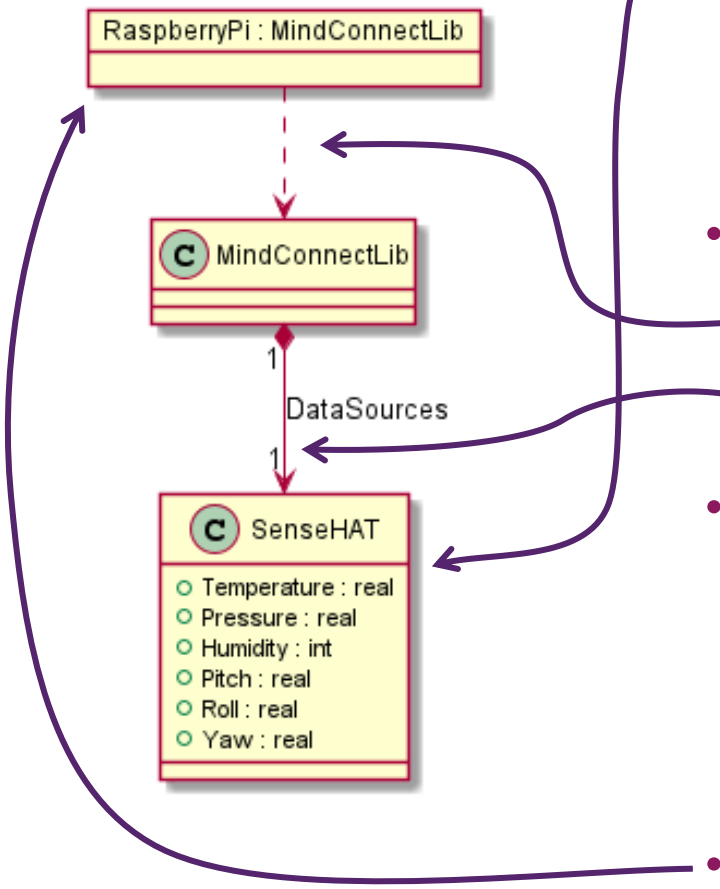
- https://plantuml.com
- Coding your Diagrams
  - Pro: Simple way to build your model
  - Cons: Limited flexibility on layout & design
- VS Code Extension
  - https://marketplace.visualstudio.com/items?itemName=jebbs.plantuml





OST

Class Diagrams



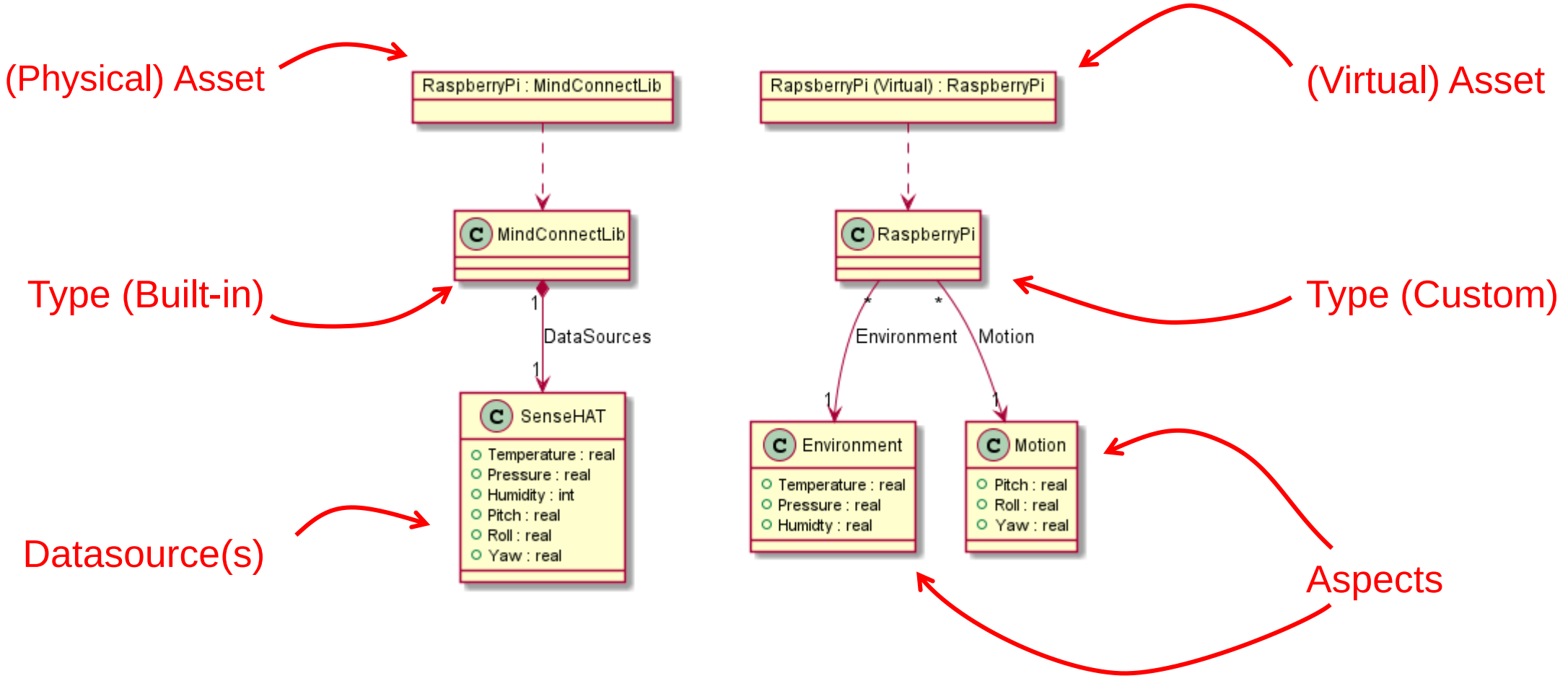
- A class is a definition of a concept (object)
  - It contains properties and operations (functions) as members
  - Every member has a visibility (+ public, - private, # protected, ~ package)
- Classes have relationships
  - Dependencies
  - Associations
- Associations can be detailed
  - Multiplicity
  - Direction
  - Aggregation/Composition
- We can also define instances of classes (realizations of objects)

Modeling for MindSphere

- In MindSphere, everything is an Asset
- We distinguish between to different concepts of Assets
  - Physical Asset
    - The model of a real device
    - Can receive data
  - Virtual Asset
    - View on your data
    - Can map on Physical Assets
      - Partially
      - 1:1
      - Multiple
- Every Asset builds upon a Type
  - For the Physical Asset, we rely on a built-in Type (**MindConnectLib**)
  - For the Virtual Asset, we define our own Type
- Our custom Type builds upon Aspects
  - An Aspect is a group of variables
  - A variable is mapped on a data point

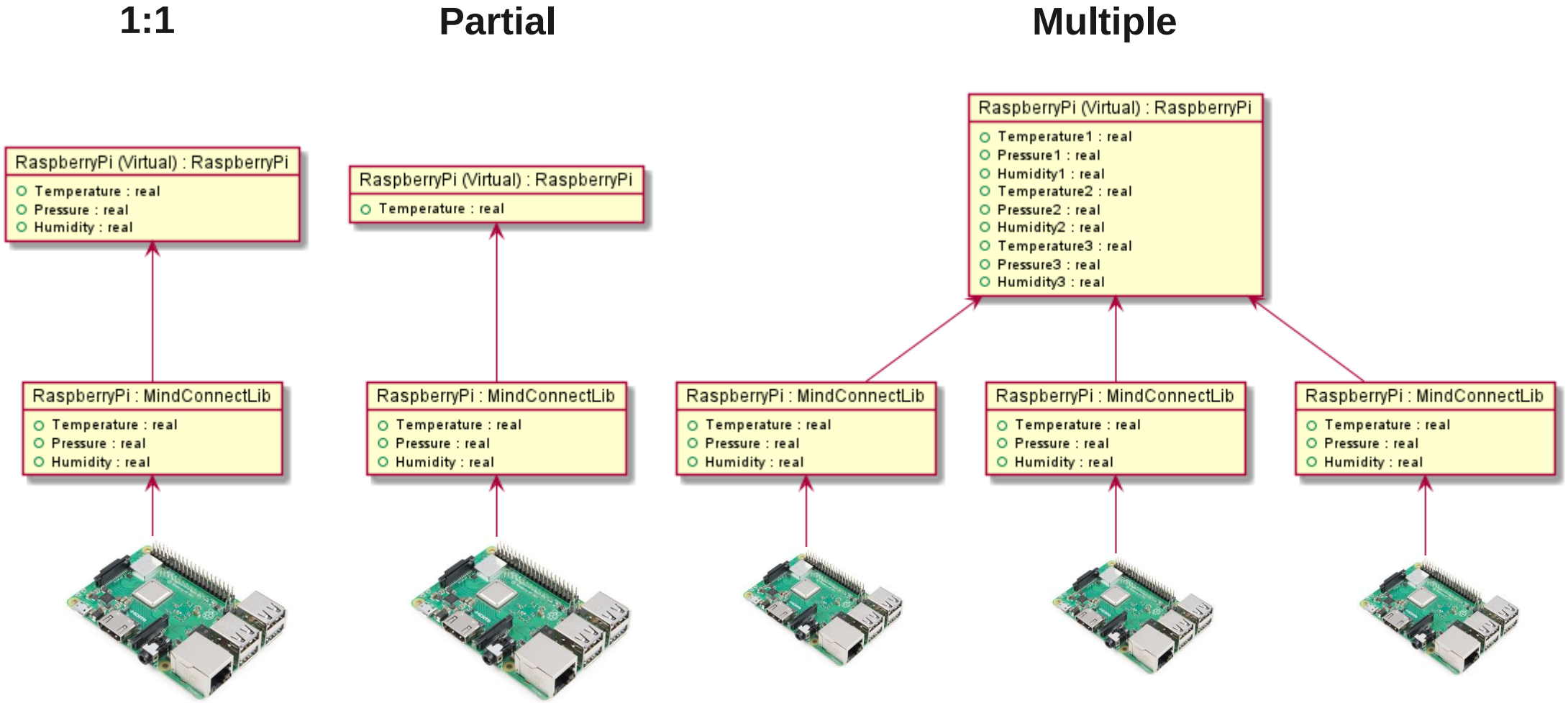
➢ See exercise for a hands-on tutorial

Modeling your Assets



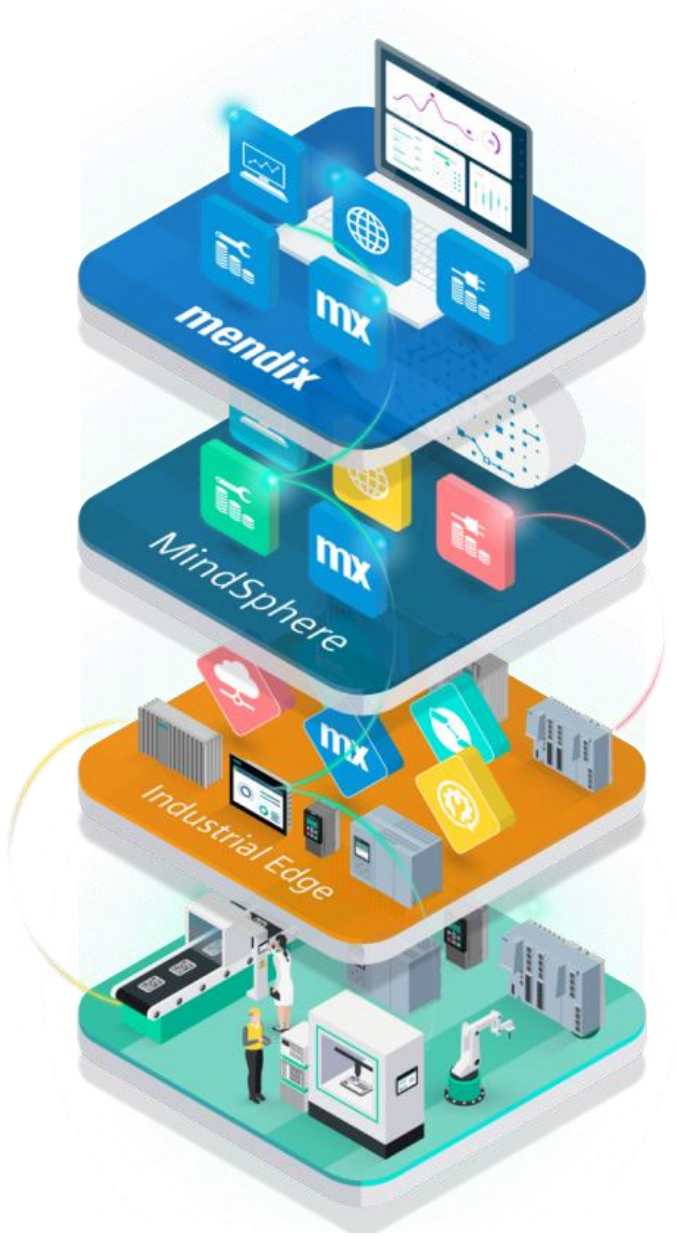
OST

Why a Physical and Virtual Asset?



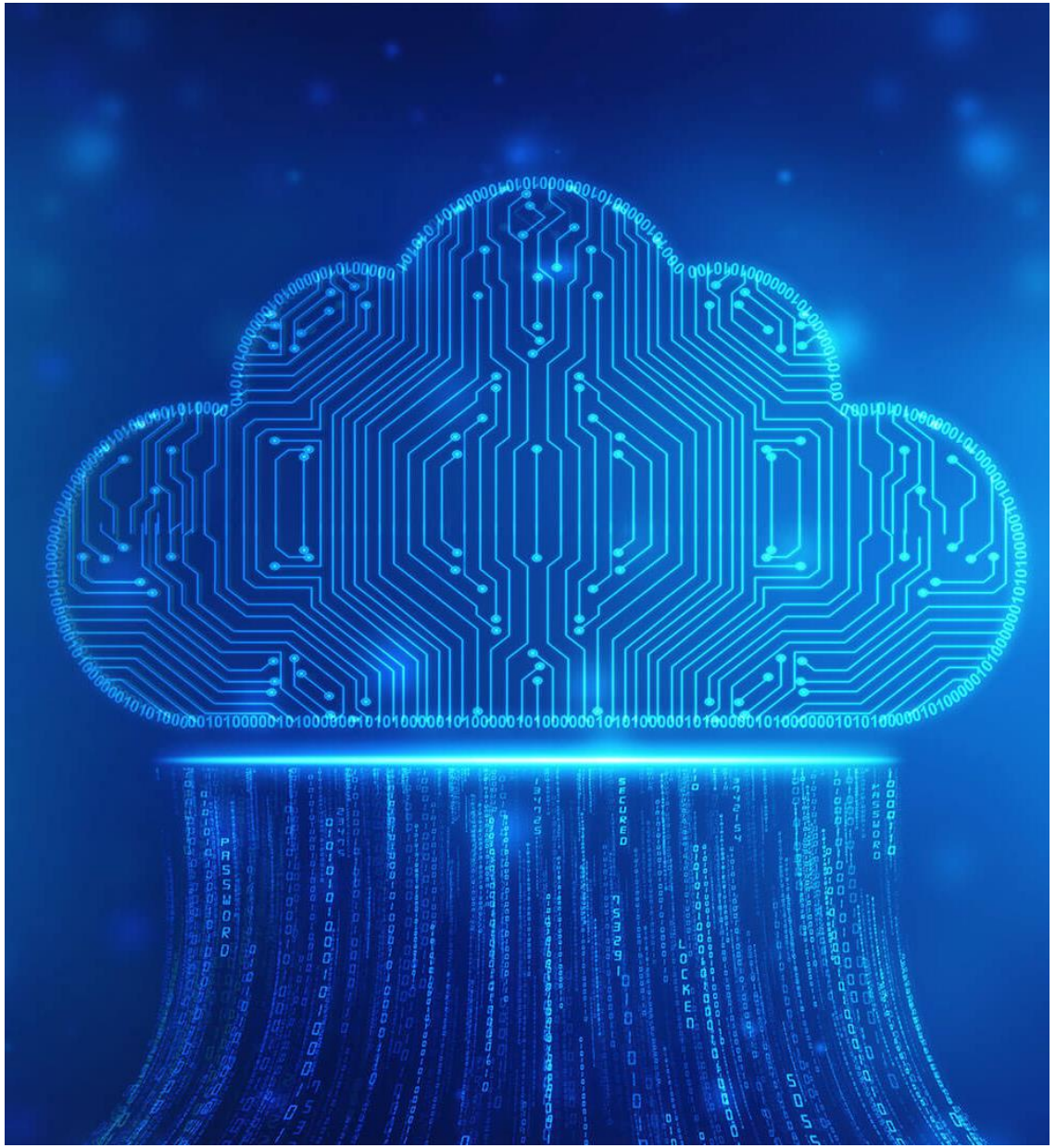
Working with MindSphere

Exercise



Cloud Computing

Lecture



Pros

Major

- **On-demand elasticity**
  - You can get more resources when you need them (e.g., CPU, RAM, Storage, Servers/VMs)
  - You can free resources when you no longer need them
- **Security is no longer your concern** (to some extent)
  - You still need some expertise
  - But not to the same degree as with on-premises solutions

Minor

- Pay as you go
  - Only pay for what you use
- Usage of shared infrastructure
  - Reduced costs
- Increased focus on the application layer
  - Tools and features are your main concern
- Don't worry about the hardware
  - You don't own the hardware

Cons

Major

- **You depend on connectivity**
  - If you are disconnected from the internet, you are disconnected from your data and applications
  - Always make sure that
    - **You can operate with just (limited) offline capabilities**
    - You do not lose data
      - Delayed upload is not a problem
- **You don't know where your data is stored**
  - Some data should not be stored in the cloud
  - You don't know which route your data takes to the data center

Minor

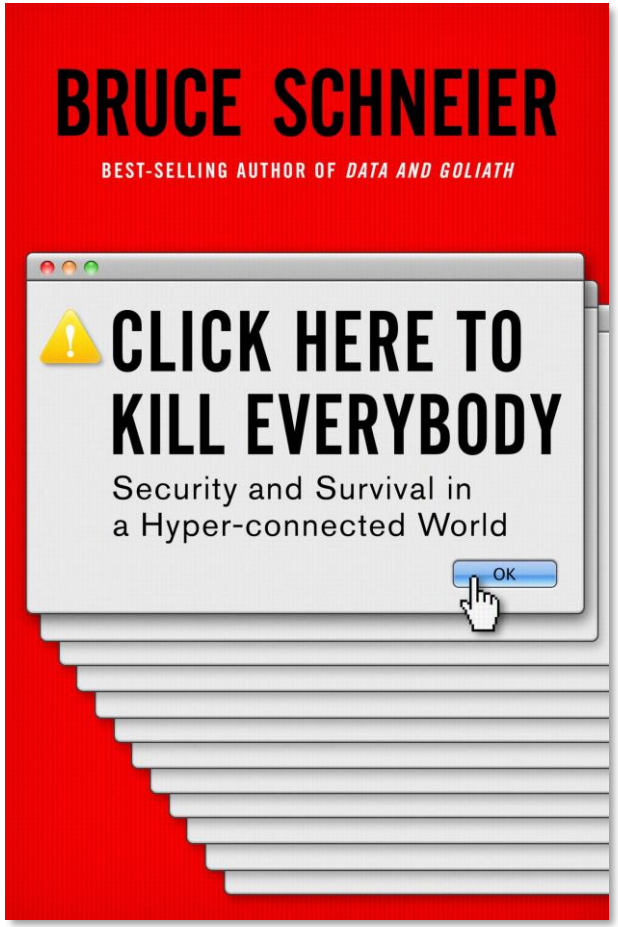
- You lose direct control over the underlying hardware
  - But if this is an issue, moving to the cloud is probably not a good idea
- Hard to diagnose performance issues, due to limited visibility and virtualization
  - You depend on tech support of the provider
- You depend on the features of your provider
  - Evaluate providers before you commit
  - Switching providers is possible, but will take some effort

Security Fundamentals

Protecting your Stuff

Why Security?

- You use it every day
  - Encrypted hard drive
  - Browsing with https://
  - Messengers (WhatsApp, iMessenger, Threema, etc)
- It is an important topic
- It will be more important
  - You can't avoid it
- "Security by Obscurity"
  - Using complexity to avoid security measures
  - Using secrecy as security method

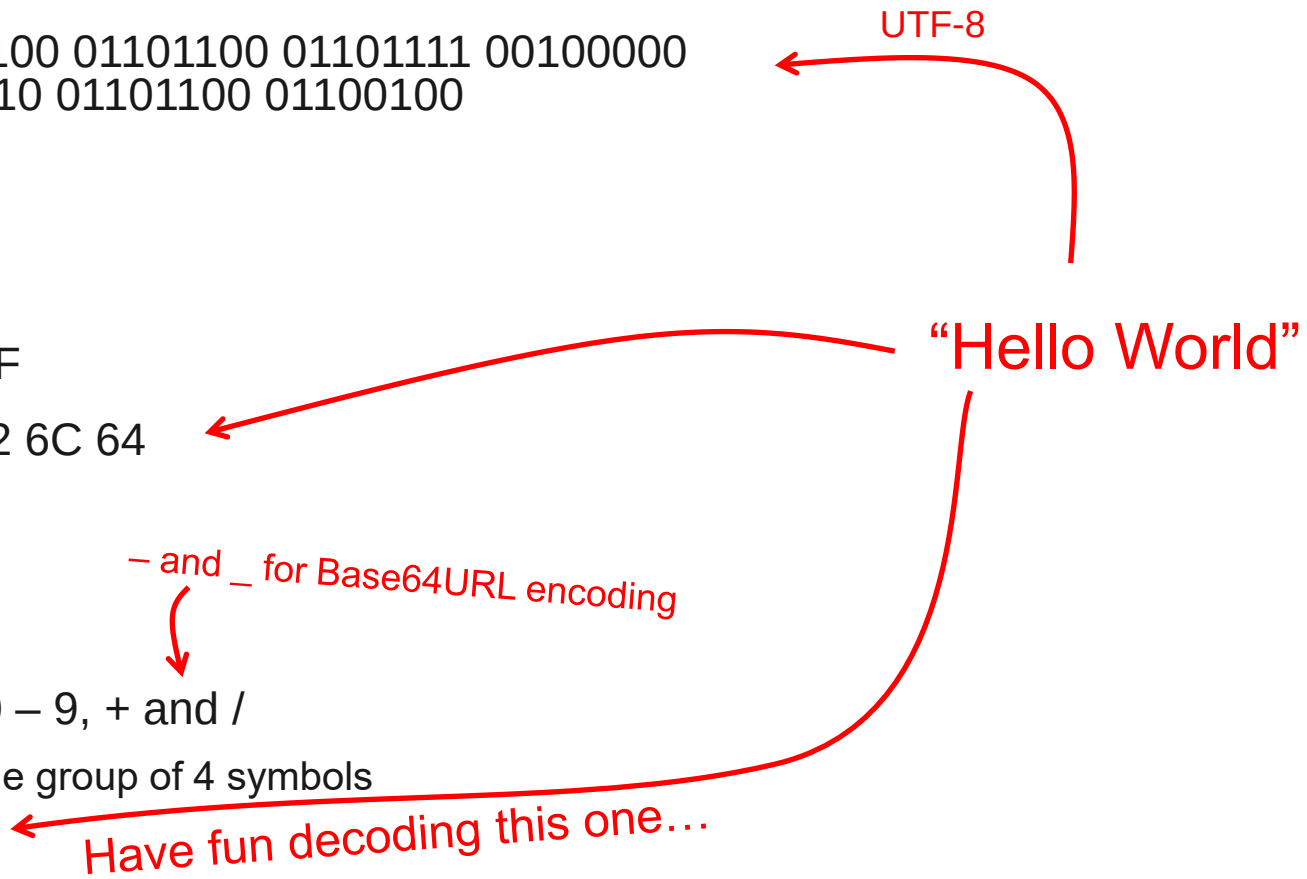


https://www.schneier.com/books/click-here/

OST

Data Representation

- Binary
  - 2 Symbols = 0 and 1
  - 01001000 01100101 01101100 01101100 01101111 00100000 01010111 01101111 01110010 01101100 01100100
- Hexadecimal
  - 16 Symbols = 0 – 9 and A - F
  - 48 65 6C 6C 6F 20 57 6F 72 6C 64
- Base64
  - 64 Symbols = A – Z, a – z, 0 – 9, + and /
    - "=" symbol at the end to fill the group of 4 symbols
  - SGVsbG8gV29ybGQ=



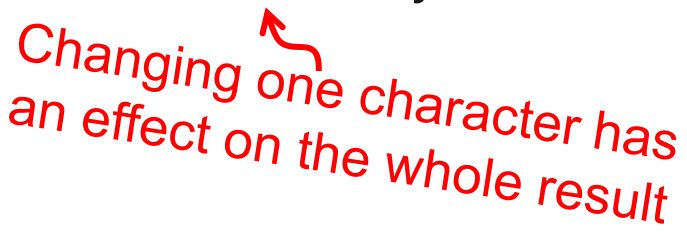
Cryptographic Hash Functions

- Definition
  - $Hash : \Sigma \longrightarrow \Sigma^*, x \longmapsto x^*$
  - Σ denotes the set of every possible input
  - Σ^* denotes the set of every possible output
  - Hash(x) is O(n), n = |x|
- Creates a fixed length output x^* representing the input
  - e.g., SHA-256 creates a 256-bit output (2^{256} = 64 HEX symbols)
  - Input length |x| does not matter
- Collision Resistance
  - $\forall x \in \Sigma, \forall y \in \Sigma, x \neq y : Hash(x) \neq Hash(y)$
    - Two different inputs cannot have the same output
    - Theoretical possible, but not in practice
- Hiding
  - $Hash(x) = x^* : \nexists Hash^{-1}(x^*) = x$
    - It's not possible to get the input from the output
    - Except by trying every possible input

Hash Examples

“Smart Factory" = dcb6fe8fd753123cef283a601dbe6ffe3fd1d69b819d21c346dfc4e1ed9bcb01

“SmartFactory" = 84d0681e3e1757a3194156f806be354c0d2496b235a0accb2d1c75cf605b7a6b



- Used for storing your passwords
  - This way, your password is not stored in plain text
    - And therefore, cannot be retrieved
  - A salt (random data) is added to prevent pre-generating hashes (rainbow tables)
    - https://en.wikipedia.org/wiki/Rainbow_table
  - Checking your password is a comparison of two hashes
    - If equal, the given password must be the same as the one that is known by the system

Symmetric Cryptography

- Using the same *key* for encryption and decryption
  - $msg \oplus key = msg_{enc}$
  - $msg_{enc} \oplus key = msg$
- Algorithms
  - AES (Advanced Encryption Standard)
  - DES (Data Encryption Standard)
  - IDEA (International Data Encryption Algorithm)
- Advantages
  - It's fast
  - It's simple
- Disadvantages
  - If someone gets the key, he can decrypt the data
  - Does not scale in a network where every connection needs a unique key
- Examples
  - Encrypt and decrypt local data

https://en.wikipedia.org/wiki/Symmetric-key_algorithm

Asymmetric Cryptography

- We have two keys
- A private key key_{priv}
  - Only known by you (thus private)
- A public key key_{pub}
  - Known by everybody (publicly available)
- Algorithms
  - RSA
- Advantages
  - It scales well in networks
- Disadvantages
  - Slow (compared to symmetric encryption)
  - Can potentially by solved since one key is known
    - Relays on features of large prime numbers
    - It's just hard to crack, but not impossible

https://en.wikipedia.org/wiki/Public-key_cryptography

OST

Two Ways of using Asymmetric Cryptography

Encryption

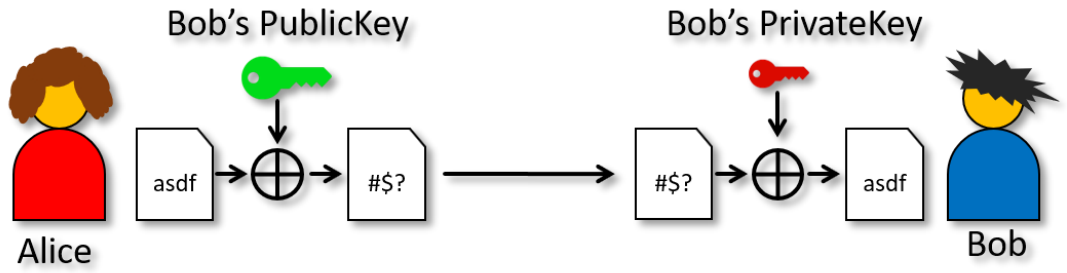
https://

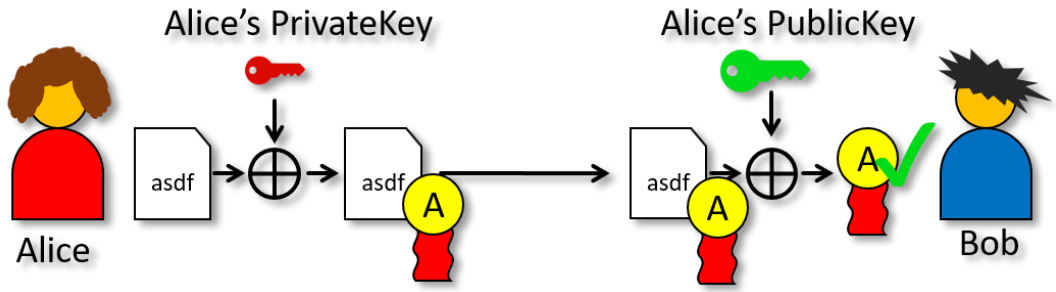
- $msg \oplus key_{pub} = msg_{enc}$
- $msg_{enc} \oplus key_{priv} = msg$
- Only the one with the corresponding private key key_{priv} (e.g., the recipient) can decrypt msg_{enc}

Signing

Digital Signatures

- $msg \oplus key_{priv} = sig_{msg}$
- $sig_{msg} \oplus key_{pub} = msg$
- Everyone can verify the signature sig_{msg} was created by the owner of the private key key_{priv}





OST

What’s behind the MindSphere Token?


```
{
   "content": {
      "baseUrl": "https://southgate.eu1.mindsphere.io",
      "iat":
"eyJraWQiOiJrZXktaWQtMSIsInR5cCI6IkpXVCIsImFsZyI6IlJTMjU2In0.eyJpc3MiOiJTQ0kiLCJzdWIiOiJiZGE0NmE5MjJhNWY0M2UzYTRjZmNkYTkwMWZhMjhhYSIsImF1ZCI6IkFJQU0iLCJpYXQiOjE1NzA1NDk3MjksIm5iZiI6MTU3MDU0OTcyOSwiZXhwIjoxNTcxMTU0NTI5LCJqdGkiOiJiMjcxOGJhZS1hMTNiLTRiYjctODI0NS1mZmMzMTIxMWU5Y2UiLCJzY29wZSI6IklBVCIsInRlbiI6ImhzciIsInRlbl9jdHgiOiJtYWluLXRlbmFudCIsImNsaWVudF9jcmVkZW50aWFsc19wcm9maWxlIjpbIlNIQVJFRF9TRUNSRVQiXSwic2NoZW1hcyI6WyJ1cm46c2llbWVuczptaW5kc3BoZXJlOnYxIl19.RE76sDFKsSDQiZD2lzHxYL7pXwYD3NfeKsAAJ_0-ua_90JuYAUXfLIIW0w4I09GZ9upGY9DiE6aW5FjUiRVHJre36IKsTwyVq9o7UO2HBNyTLMP8aHuW14unm96sRmEx_sAGUNT2GDiegWDN1WNDVqqt-CNpxef8O9v8pc7by7RH3DzLwphqpTwHXqEnFruqTOwbdD38SEr5o90rCd9HCprBfVroFiLzXX6RJ_L1r6nejtMLC5vQnfH0qNYK9qFRFq7OSi9WkZb-AWZBvD_FfhVvCe0WJ7EBw36kzgNeWxEpyZrDGi_CyfMKyxUtMEu-lPN8eWaW_OaoBG0_sPsCUQ",
      "clientCredentialProfile": [
         "SHARED_SECRET"
      ],
      "clientId": "bda46a922a5f43e3a4cfcda901fa28aa",
      "tenant": "hsr"
   },
   "expiration": "2019-10-15T15:48:49.000Z"
}
```


OST

What's behind the MindSphere Token?


```
{
   "content": {
      "baseUrl": "https://southgate.eu1.mindsphere.io",
      "iat":
"eyJraWQiOiJrZXktaWQtMSIsInR5cCI6IkpXVCIsImFsZyI6IlJTMjU2In0.eyJpc3MiOiJTQ0kiLCJzdWIiOiJiZGE0NmE5MjJhNWY0M2UzYTRjZmNkYTkwMWZhMjhhYSIsImF1ZCI6IkFJQU0iLCJpYXQiOjE1NzA1NDk3MjksIm5iZiI6MTU3MDU0OTcyOSwiZXhwIjoxNTcxMTU0NTI5LCJqdGkiOiJiMjcxOGJhZS1hMTNiLTRiYjctODI0NS1mZmMzMTIxMWU5Y2UiLCJzY29wZSI6IklBVCIsInRlbiI6ImhzciIsInRlbl9jdHgiOiJtYWluLXRlbmFudCIsImNsaWVudF9jcmVkZW50aWFsc19wcm9maWxlIjpbIlNIQVJFRF9TRUNSRVQiXSwic2NoZW1hcyI6WyJ1cm46c2llbWVuczptaW5kc3BoZXJlOnYxIl19.RE76sDFKsSDQiZD2lzHxYL7pXwYD3NfeKsAAJ_0-ua_90JuYAUXfLIIW0w4I09GZ9upGY9DiE6aW5FjUiRVHJre36IKsTwyVq9o7UO2HBNyTLMP8aHuW14unm96sRmEx_sAGUNT2GDiegWDN1WNDVqqt-CNpxef8O9v8pc7by7RH3DzLwphqpTwHXqEnFruqTOwbdD38SEr5o90rCd9HCprBfVroFiLzXX6RJ_L1r6nejtMLC5vQnfH0qNYK9qFRFq7OSi9WkZb-AWZBvD_FfhVvCe0WJ7EBw36kzgNeWxEpyZrDGi_CyfMKyxUtMEu-lPN8eWaW_OaoBG0_sPsCUQ",
      "clientCredentialProfile": [
         "SHARED_SECRET"
      ],
      "clientId": "bda46a922a5f43e3a4cfcda901fa28aa",
      "tenant": "hsr"
   },
   "expiration": "2019-10-15T15:48:49.000Z"
}
```


It's a combination of several Base64 values

The token – this is sent with every request

Remember - Base64
It looks like gibberish, but it isn't!

Id of your asset – see, it's a GUID

Until when this token must be used for the first time

OST

JWT (JSON Web Token)

- Common way to grant access to an API
- Uses JSON to structure claims
  - Claims are your granted rights
- Token is signed to ensure its validity
  - A token could be changed
  - But you can detect the changes when you regenerate the signature

Header	{
	"alg" : "RS256",
	"typ" : "JWT"
	}
		Metadata of the token
		e.g., Algorithm used

Payload	{
	"ten" : "hsr",
	"iat" : 1570549729,
	"exp" **:** 1571154529
	}
		Data of the token
		e.g., who you are, what you are allowed to do, when the token was generated,
		how long the token is valid, etc.

Signature	Hash(secret, base64urlEncoding(header) + '.' + base64urlEncoding(payload))	Signature of the token To ensure the token is valid and no information was changed

Token = base64urlEncoding(header) + '.'
+ base64urlEncoding(payload) + '.'
+ base64urlEncoding(signature)

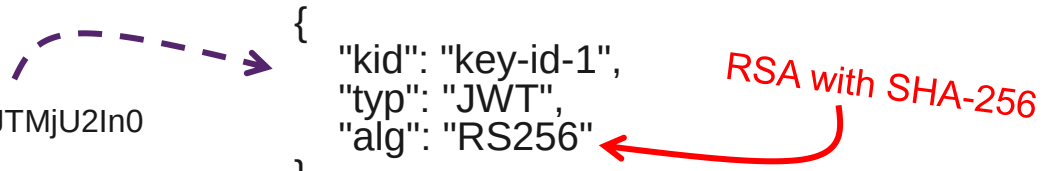
https://en.wikipedia.org/wiki/JSON_Web_Token

OST

What’s behind the MindSphere Token?

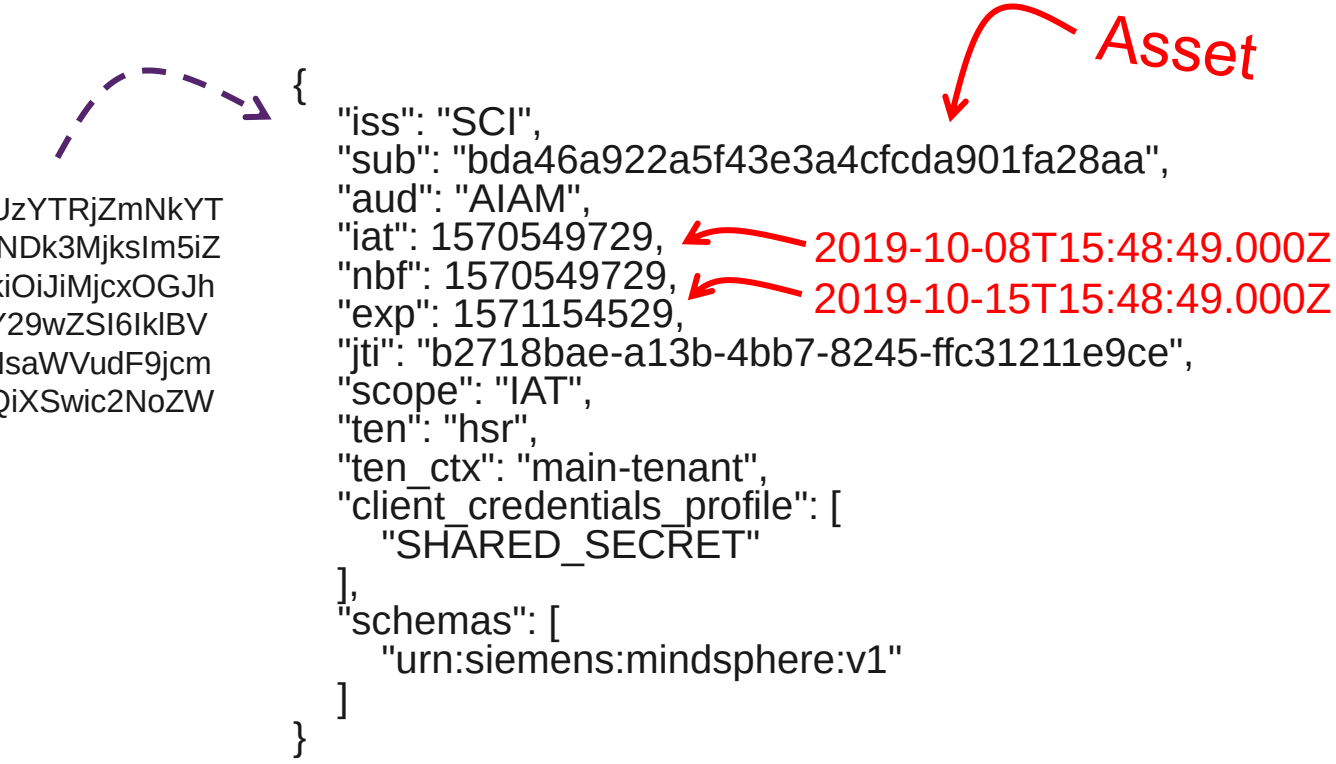
- Header

eyJraWQiOiJrZXktaWQtMSIsInR5cCI6IkpXVCIsImFsZyI6IlJTMjU2In0



- Payload

eyJpc3MiOiJTQ0kiLCJzdWIiOiJiZGE0NmE5MjJhNWY0M2UzYTRjZmNkYTkwMWZhMjhhYSIsImF1ZCI6IkFJQU0iLCJpYXQiOjE1NzA1NDk3MjksIm5iZiI6MTU3MDU0OTcyOSwiZXhwIjoxNTcxMTU0NTI5LCJqdGkiOiJiMjcxOGJhZS1hMTNiLTRiYjctODI0NS1mZmMzMTIxMWU5Y2UiLCJzY29wZSI6IklBVCIsInRlbiI6ImhzciIsInRlbl9jdHgiOiJtYWluLXRlbmFudCIsImNsaWVudF9jcmVkZW50aWFsc19wcm9maWxlIjpbIlNIQVJFRF9TRUNSRVQiXSwic2NoZW1hcyI6WyJ1cm46c2llbWVuczptaW5kc3BoZXJlOnYxIl19



What’s behind the MindSphere Token?

- Signature

  RE76sDFKsSDQiZD2lzHxYL7pXwYD3NfeKsAAJ_0-ua_90JuYAUXfLIIW0w4I09GZ9upGY9DiE6aW5FjUiRVHJre36IKsTwyVq9o7UO2HBNyTLMP8aHuW14unm96sRmEx_sAGUNT2GDiegWDN1WNDVqqt-CNpxef8O9v8pc7by7RH3DzLwphqpTwHXqEnFruqTOwbdD38SEr5o90rCd9HCprBfVroFiLzXX6RJ_L1r6nejtMLC5vQnfH0qNYK9qFRFq7OSi9WkZb-AWZBvD_FfhVvCe0WJ7EBw36kzgNeWxEpyZrDGi_CyfMKyxUtMEu-lPN8eWaW_OaoBG0_sPsCUQ

- We can’t do much with that since it’s a hash
- This is used within MindSphere to check if the header and payload are correct
  - If true, sent data is accepted
  - If not, the data is simply rejected
    - You are not interested in knowing what has changed, just that something has changed

Dashboard with Edge2Web

Exercise

Project

Self-Study

Introduction to the Project

Show me the Data

Task 2: Development of a Dashboard

Task Formulation

➢ Same groups as you had in task 1

1. Visualize data on your device
2. Model your assets in MindSphere
3. Establish communication to your Asset and check if data is received
4. Develop a minimal dashboard to visualize the data sent from your device
5. Extend the dashboard with further data visualizations from a second device

➔ Supporting Documents

- Node-RED Dashboard Manual
- Tutorial (Exercises)
- Persona

Expected Deliverables

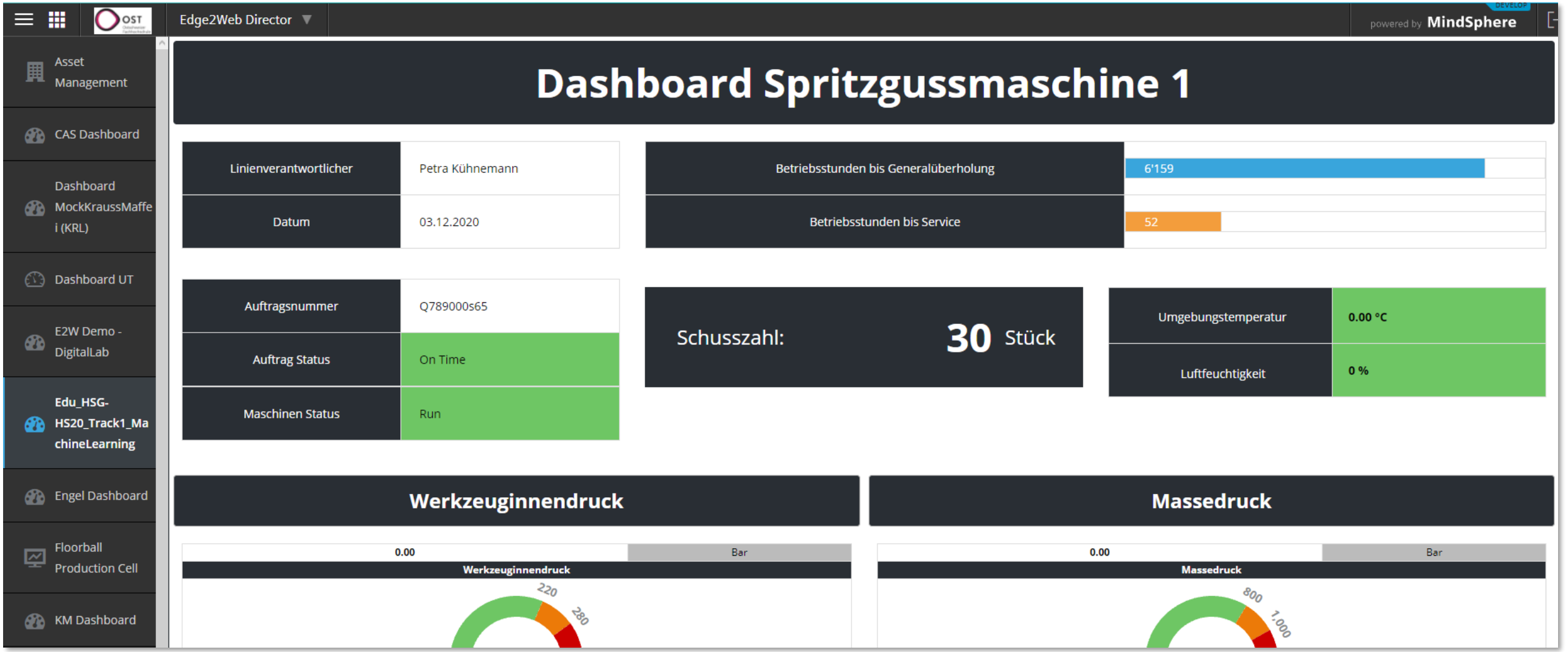
- Dashboard with 6 tiles showing at least 12 data points
- Documentation of your project
  - Thoughts and implications of the persona for the UI (1 page)
  - Paper prototype/design concept (4 pages)
  - Finding/conclusion (0.5+ page)
  - Reflection on the task (0.5+ page)

  ➔ *The submission of the results has to be done on Moodle until 14.12.2021 (PDF-Format)*
- Presentation (approximately 5 minutes) of the dashboard on 14.12.2021

  ➔ *The submission of the presentation has to be done on Moodle until 14.12.2021 (PDF-Format)*

OST

Result



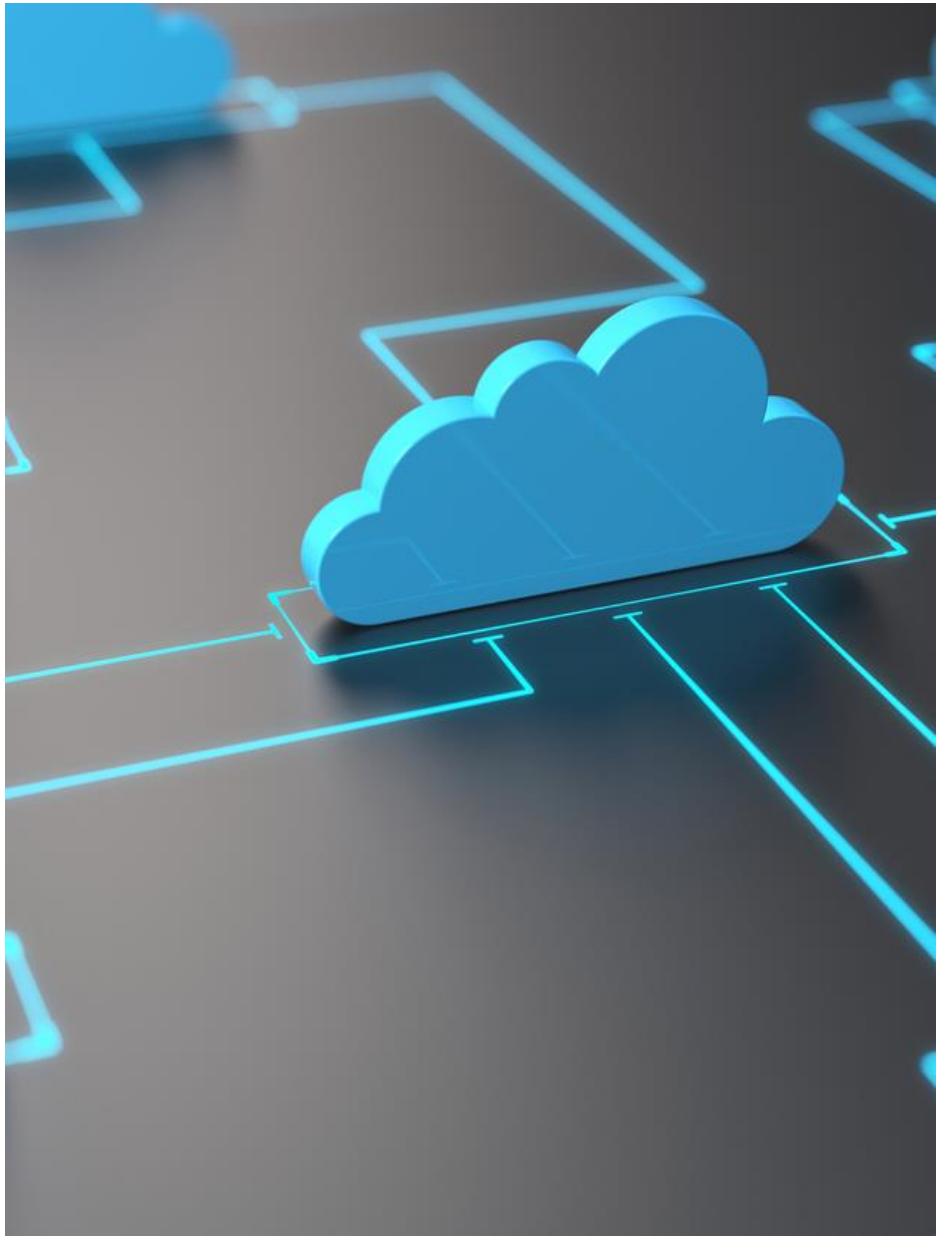
Project

Exercise

OST

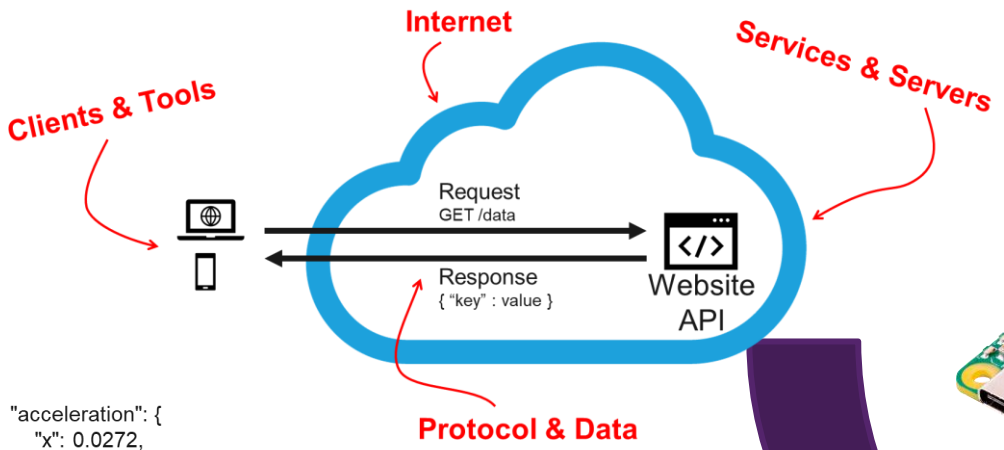
Today's Objectives

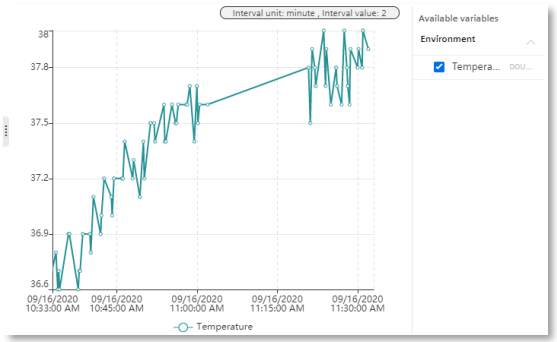
- ✓ You can use a REST API
- ✓ You can identify the service concepts of cloud providers
- ✓ You know the basic pros and cons about the cloud
- ✓ You have a basic understanding of key security concepts
- ✓ You can identify the parts of JWT

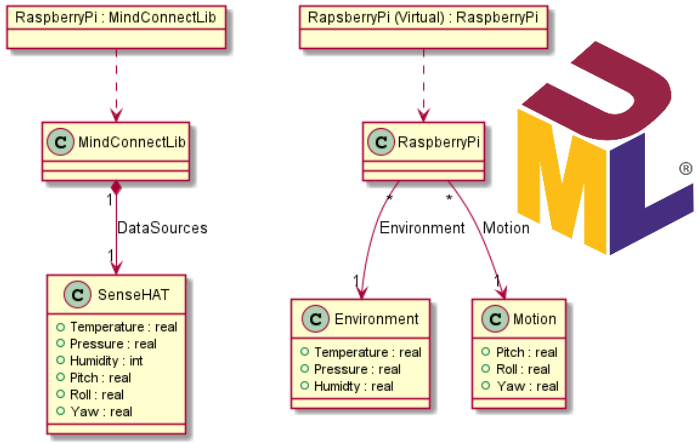


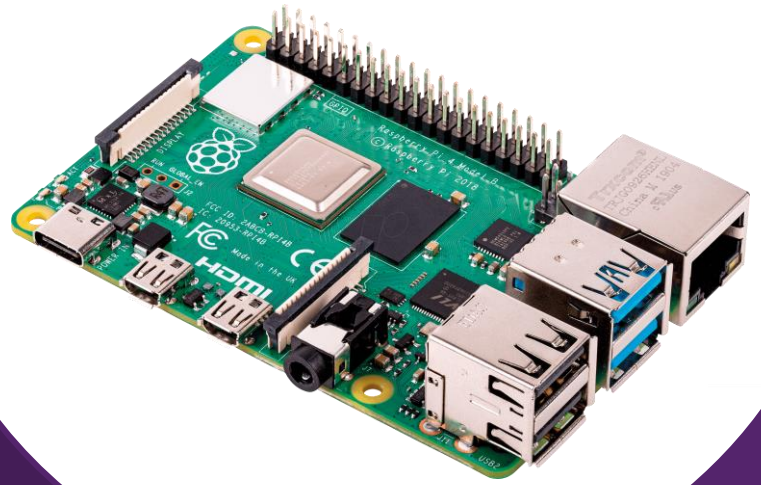
OST

Closing Remarks










```
{
   "acceleration": {
      "x": 0.0272,
      "y": 0.0238,
      "z": 1.0052
   },
   "gyroscope": {
      "x": 0.002,
      "y": 0,
      "z": -0.0015
   },
   "orientation": {
      "roll": 1.4011,
      "pitch": 358.4654,
      "yaw": 249.031
   },
   "compass": 249
}
```