

# Digital Twin & Data Collection

Riku Ala-Laurinaho

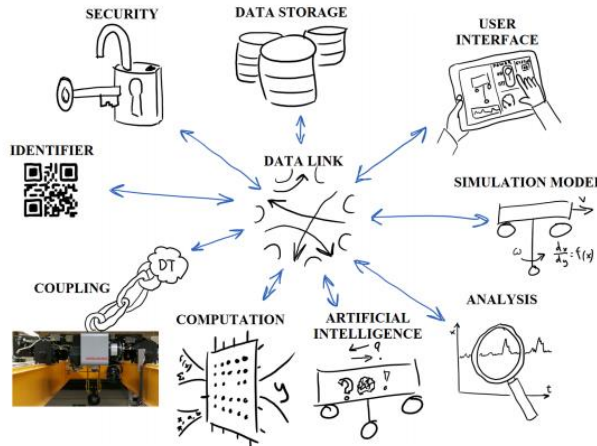
30.10.2019



Aalto-yliopisto  
Aalto-universitetet  
Aalto University

# Data collection is an essential part of DT

- Digital twin needs real data from its counterpart to reach its full potential
- Most of the DT features are based on **accurate** information about the counterpart
- No output without input



Physical product



With DT



With DT & simulation



With DT & real data



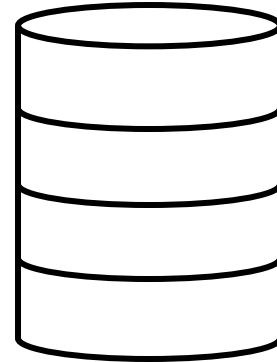
# Structure of the presentation

- What can a DT do with data and what kind of data is needed for that?
- Requirements for data collection
- Solution for data collection



# What does data collection enable?

- **Confirming the simulation results or assumptions made in the development phase of the product**
- **Analyzing the use of the product, improving the design**
- **Optimization of operation**
- **Couples physical product and DT**
  - Running (forward) simulation
- **Predictive maintenance**



# What kind of data is needed by DT?

- Boring, yet truthful answer: depends on the use case & product
- In scientific literature, sometimes it is stated that DT *completely* mirrors its physical counterpart [1]. However, this is not feasible.
- What should be then monitored?
  - Weak parts
  - Performance
  - Adding sensors based on need
  - Oversensing
  - Virtual sensors

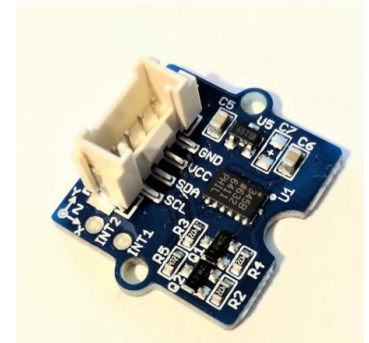


# Requirements for data collection

- **High quality data**
  - Sufficient sample rate and sensitivity
- **Addition of new sensors, removing unnecessary**
- **Changing measurement settings**
- **Changing the network parameters**
- **Software updating**
  - Bug fixes
  - New features



Source: <https://www.baaahq.org/news/dont-wait-until-its-too-late-update-your-software-now>



# Solution: Sensor Management System

- Easy addition of new sensor nodes
- Easy addition of new sensor types
- Software updates & remote management of sensors over Internet
- Support for multiple data formats and both HTTP and MQTT
- Web User Interface & REST API



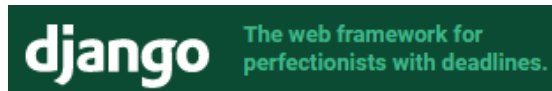
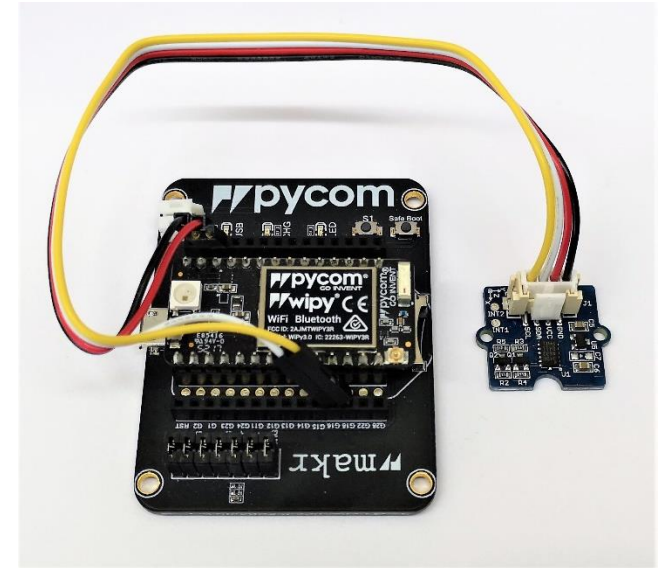
SMP Home Sensors Communication technologies Protocols Instructions User: admin Logout							
Browse sensors							
Order by: Id Inverse							
Search by keyword:							
Id Name Model Location Date added Last modified Status Actions							
1	hookAcceleration	ADXL345	AIIC Hall, K3	Oct. 15, 2019, 1:05 p.m.	Nov. 6, 2019, 3:17 p.m.	Measuring	Info Edit Delete
2	BridgePosition	ADXL345	AIIC hall, K3	Nov. 12, 2019, 8:57 a.m.	Nov. 12, 2019, 8:57 a.m.	Waiting-for-update	Info Edit Delete

# Hardware & software

- Pycom WiPy ESP32-based microcontroller, micropython
- 3-axis accelerometer ADXL345, Garmin LIDAR-Lite v3HP...
- Django, Django REST framework, jQuery, Bootstrap...



Source: <https://buy.garmin.com/en-US/US/p/578152>



Source: <https://www.djangoproject.com/start/overview/>

**A?** Aalto-yliopisto  
Aalto-universitetet  
Aalto University



Source: <https://www.cablefree.net/wireless-technology/history-of-wifi-technology/>



# User Interface & API

- Sensor nodes can be added, modified and removed from Web user interface or REST API
- Web user interface provides ease of use

## Browse sensors

Order by:  ☐ Inverse

Search by keyword:

Id	Name	Model	Location	Date added	Last modified	Status	Actions
1	<a href="#">hookAcceleration</a>	<a href="#">ADXL345</a>	AIIC Hall, K3	Oct. 15, 2019, 1:05 p.m.	Nov. 6, 2019, 3:17 p.m.	Measuring	<a href="#">Info</a> <a href="#">Edit</a> <a href="#">Delete</a>
2	<a href="#">BridgePosition</a>	<a href="#">ADXL345</a>	AIIC hall, K3	Nov. 12, 2019, 8:57 a.m.	Nov. 12, 2019, 8:57 a.m.	Waiting-for-update	<a href="#">Info</a> <a href="#">Edit</a> <a href="#">Delete</a>

## Api Root

The default basic root view for DefaultRouter

GET /api/v1.0/

```
HTTP 200 OK
Allow: GET, HEAD, OPTIONS
Content-Type: application/json
Vary: Accept

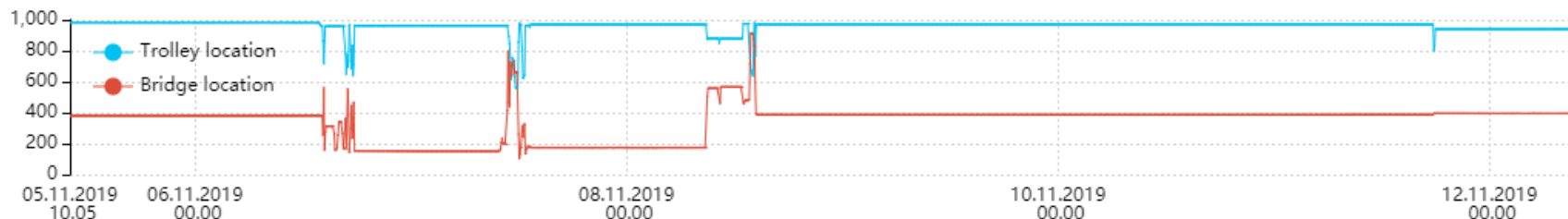
{
  "sensors": "https://riku.digikaksonen.fi/api/v1.0/sensors/",
  "models": "https://riku.digikaksonen.fi/api/v1.0/models/",
  "wlans": "https://riku.digikaksonen.fi/api/v1.0/wlans/",
  "nb-iots": "https://riku.digikaksonen.fi/api/v1.0/nb-iots/",
  "http": "https://riku.digikaksonen.fi/api/v1.0/http/",
  "https": "https://riku.digikaksonen.fi/api/v1.0/https/",
  "mqtt": "https://riku.digikaksonen.fi/api/v1.0/mqtt/",
  "dataformats": "https://riku.digikaksonen.fi/api/v1.0/dataformats/",
  "samplerates": "https://riku.digikaksonen.fi/api/v1.0/samplerates/",
  "sensitivitis": "https://riku.digikaksonen.fi/api/v1.0/sensitivitis/",
  "valuepairs": "https://riku.digikaksonen.fi/api/v1.0/valuepairs/"
}
```

# Data?

- Management system has "nothing to do with the collected data" i. e. it does not have access to it
- External IoT platforms can be used to store and visualize data
  - Good option: Remion Regatta platform
- External systems can change the measurement setting via REST API



Ilmatar location (1w\*)



# Conclusion

- **Data collection is an essential part of DT**
- **Large amount of sensors necessitates management system**
- **Solution: Sensor Management Platform**
  - Code open-sourced in GitHub, please collaborate

