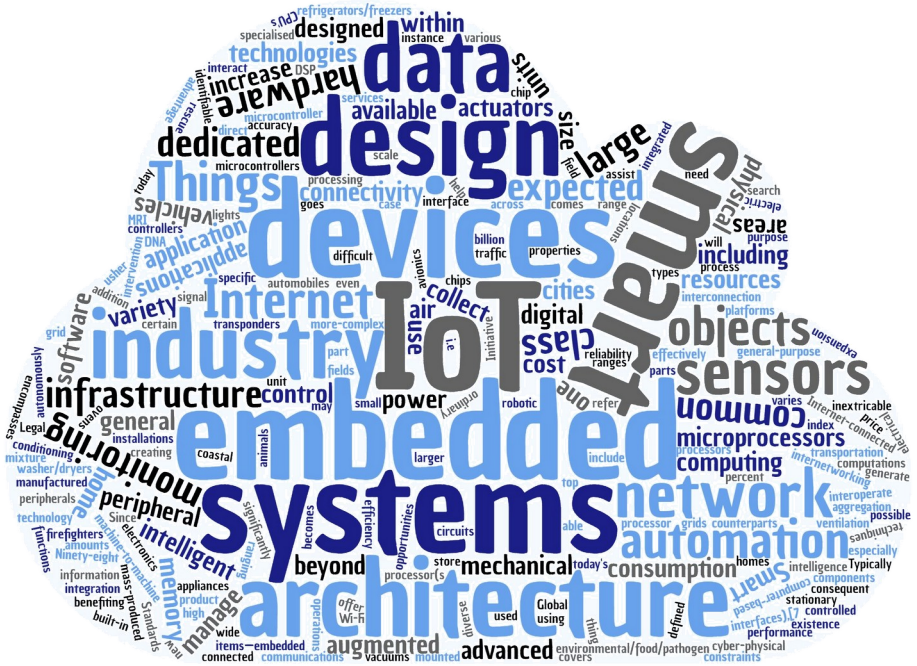
Associate Professorship of Embedded Systems and Internet of Things
Department of Electrical and Computer Engineering
Technical University of Munich

Lecture
IoT Remote Lab

01-Introduction

Ege Korkan

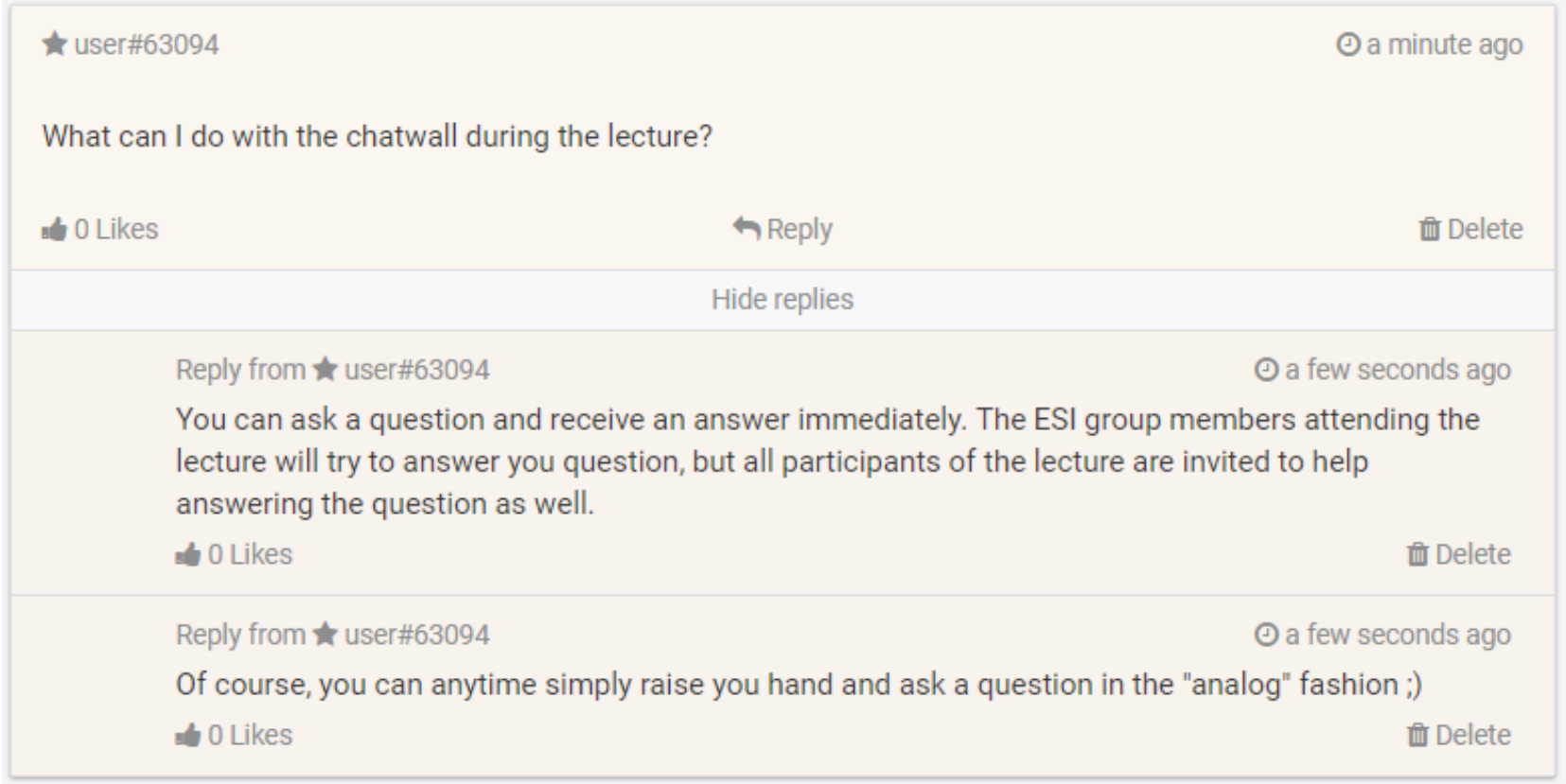


Zoom Guidelines

- The lectures and tutor sessions happen on Zoom meetings following the link sent to you via email.
- Participatition to Zoom sessions is optional
- You can choose a random string for your name
- The Zoom chat will not be recorded and we will not save the chat.
- All the participants except the lecturer is muted. The participants are free to unmute. You can also go to participants, click the hand icon to raise your hand.
- You can also do other things, like asking me to go slower. I have a separate window where I look at the requests from the participants.
- You can ask quick questions in Zoom chat or use the Tweedback link provided in each session.

Tweedback for Real-time Q&A During the Lecture

With Tweedback, you can ask questions during the lecture in real-time and anonymously, if you want.
To access the Tweedback chatwall, you access the address tweedback.de/****, where **** stands for the session ID of the specific day, such as https://tweedback.de/pdwn/chatwall for today



Participants

- There are only 25 places available for 68 registrations.
- The allocation happened centrally by the EI Department.
- You should confirm your place right now by writing me an email **now**
- The next sessions will be reserved for the ones with confirmed places. The meeting link will change.

Introduction of Lecturer

Name: Ege Korkan
PhD Candidate, Embedded Systems and Internet of Things
Department of Electrical and Computer Engineering
Office: 4965
Email: ege.korkan@tum.de
Twitter: @egekorkan (personal)
@TUM_EmbeddedIoT (research group)

Introduction of Tutor

Name: Andreas Schrägle
Department of Electrical and Computer Engineering
Email: andreas.schraegle@tum.de

Research Area of Group

Design methodology and hardware/software architecture co-design of resource-constrained distributed embedded systems

Application Areas:
Internet of Things, smart energy and automotive systems

Focus:
Decentralized system architectures

Research Goal:
Enable Cyber-Physical Co-Design across all abstraction levels

My Research Area

Description/Modeling and Verification of Systems of Things in the Web of Things

- Participation in the standardization activities in the World Wide Web Consortium
  - Editor for Binding Templates, Implementation Report
- System Description based on W3C WoT Thing Description
- Automated Thing and System Simulation, Testing

Course Structure and Goals

- 3 SWS on Thursday from 13:15 to 15:30 (Lab room 3971). Lecture + Minimum 1 hour Tutor session
  - Room 4981 for Initial lectures
  - **All exercises are to be done from your own computers (even without the COVID-19)**

Zoom Lectures and Tutor Sessions due to COVID-19 situation

Study Goals

Upon successful completion of the module, the participants are able to explain, apply IoT standards and protocols. The participants can also apply these standards and protocols in different system architectures found in IoT systems. Additionally, they are able to separate application and protocol logic in the implementation of the IoT devices as well as the application logic of the systems.

IoT

… but what do we mean that?

Answergarden Poll

Can you name some „IoT Protocols“?

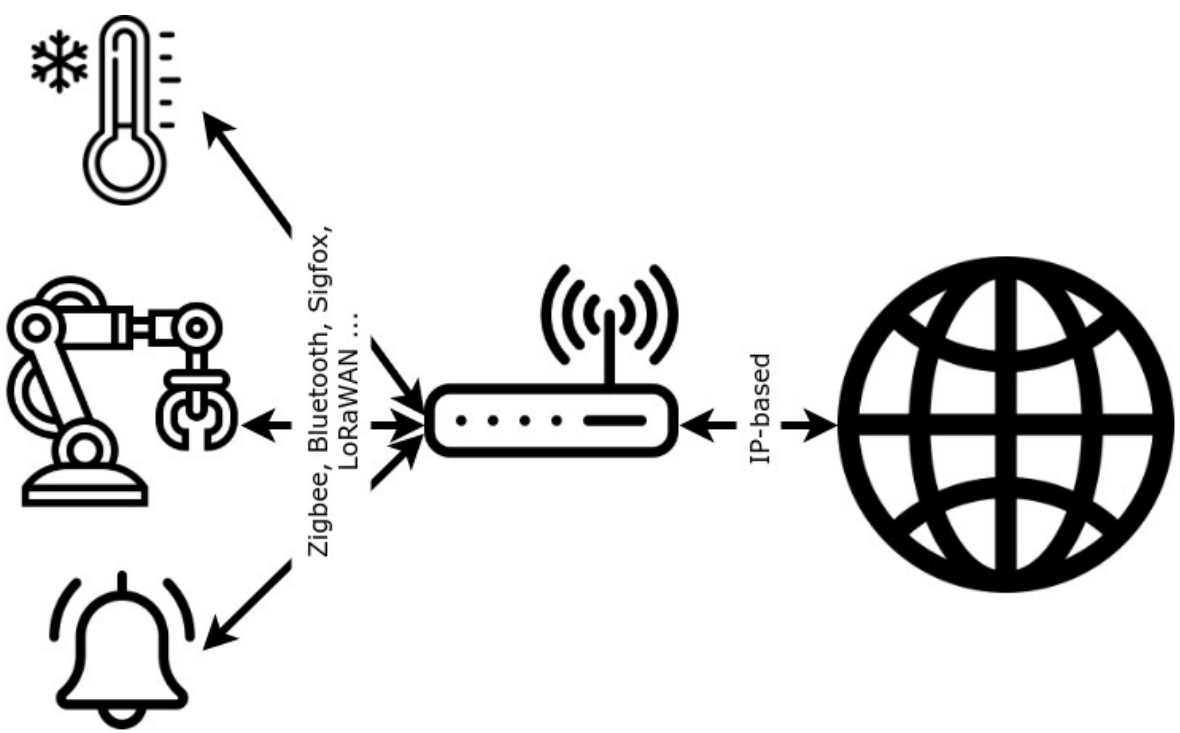
Please provide a short keyword on
https://answergarden.ch/1523278

What about **I**nternet **P**rotocols ?

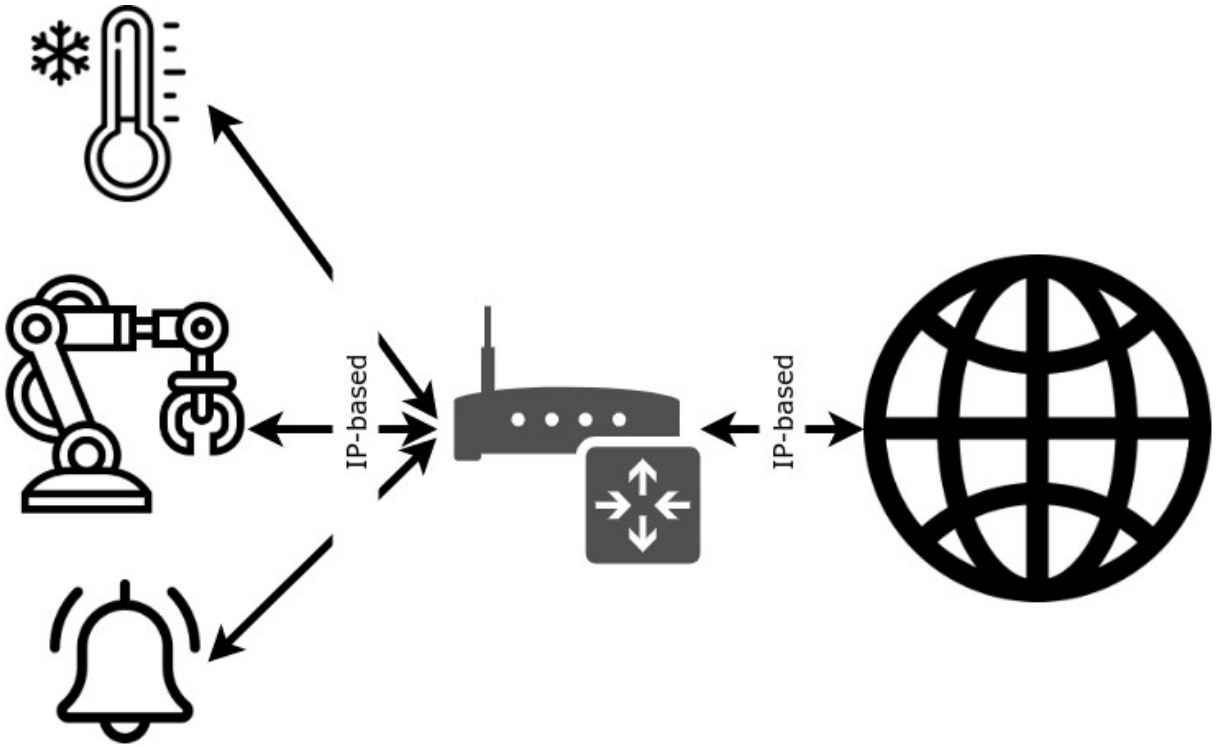
- If we ask about Internet Protocols, would anyone say Zigbee?
- What happened to the **Internet** in Internet of Things?
- Why would not many people name TCP, FTP, SMTP or even HTTP when mentioning IoT protocols?
- If a Philips Hue light bulb connected over Zigbee is an IoT device, why not your wireless keyboard?

Two types of IoT

1) Gateway-ed IoT



2) IP-based IoT



Two types of IoT

But didn‘t we always have „Gateways“?

- Multiple ECUs over CAN bus

- I^2C sensors, shields, HATs

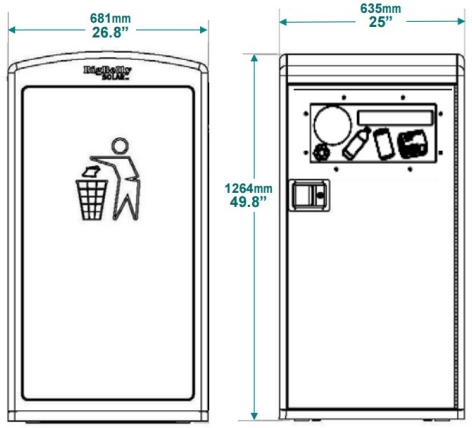
If you think of it, most of the new Gateway-based IoT protocols are wireless!

IoT Examples

Smart Parking

Smart Waste Bin

Smartbelly® Standard Capacity Station



Configurations: Mixed waste or single-stream recycling (with appropriate markings)

Materials

- RoHS compliant
- Galvanized sheet metal steel interior and exterior construction
- Heavy duty plastic side panels for dent and scratch resistance (recycled content)
- Interior Bin: Single bin is leak proof made out of low density polyethylene plastic

Power & Electronics

- Polycrystalline silicon cell PV module (22 watts). (40 watts HE [high energy] upgrade available)
- PV panel protected by polycarbonate bubble
- Spill-proof, sealed maintenance-free battery
- Self-powered unit requires no wiring

Technical Specifications (SB5)

Overall Machine Dimensions

- Height: 49.8" (1264mm)
- Width: 25" (635mm)
- Depth: 26.8" (681mm)
- Weight: 175 lbs (80kg); Shipping weight: 205 lbs (136kg)
- Insertion Opening: Can be configured for multiple recycling or waste streams.
- Bin Volume: 50 gallons (189L)

Features

- Fully automated, microprocessor controlled system senses fullness and machine status
- LED status indicate fullness level, machine status and error codes
- GPRS or CDMA wireless data link for remote monitoring and management
- GPS assisted location service

Safety Features

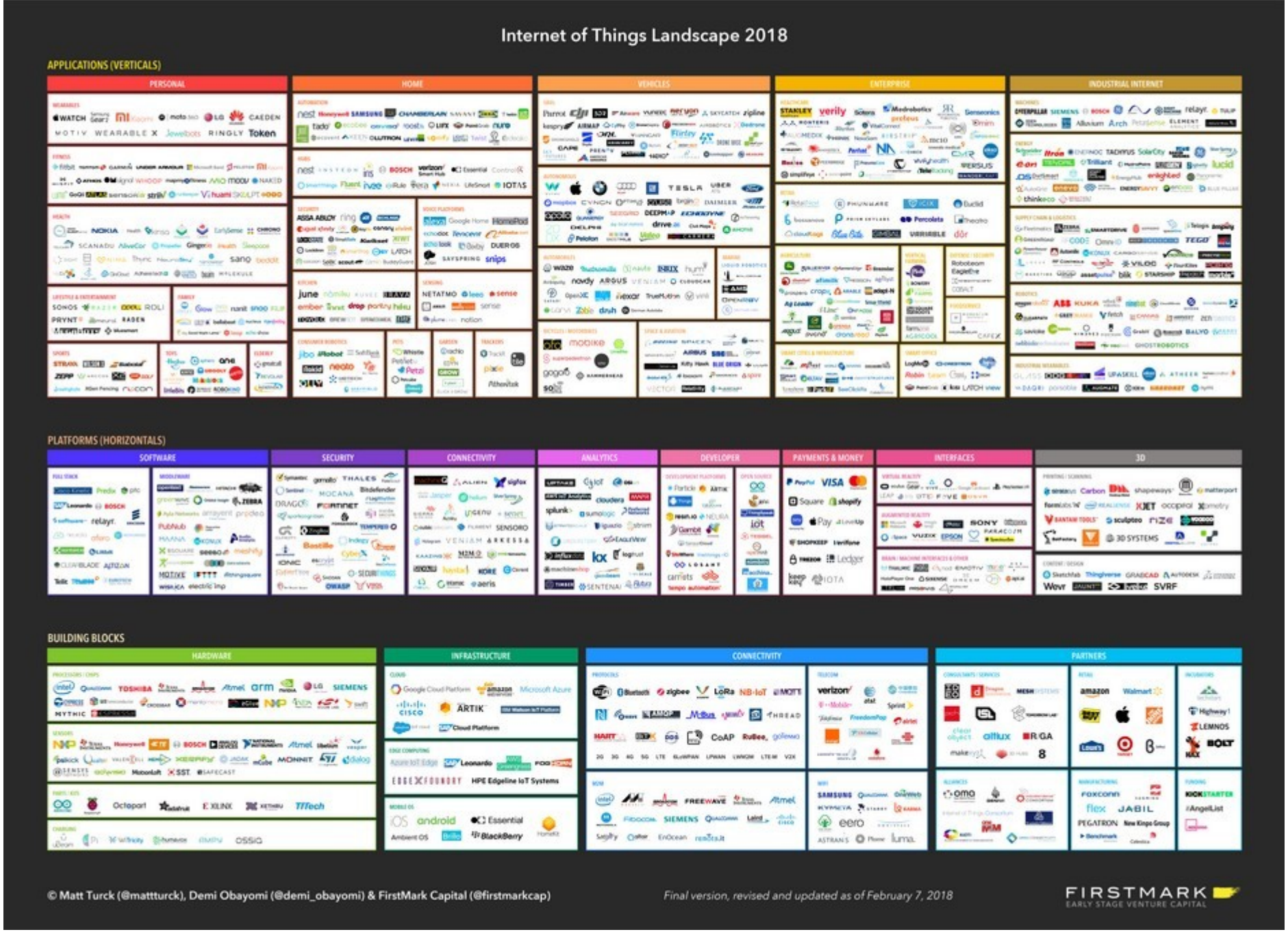
- CE approved
- Fully interlocked access doors protect users and service personnel
- Locked front trash removal door. (Locked rear door option available)
- Separately keyed service access

Durability

- Weather resistant, UV stabilized polyester powder-coat finish on all exterior parts
- Electronic components temperature range of -40ºF to +185ºF (-40ºC to +85ºC)
- Fully weatherized, but in the event of a flood, Smartbelly can withstand up to 40" (1.06m) of water without harming electronics

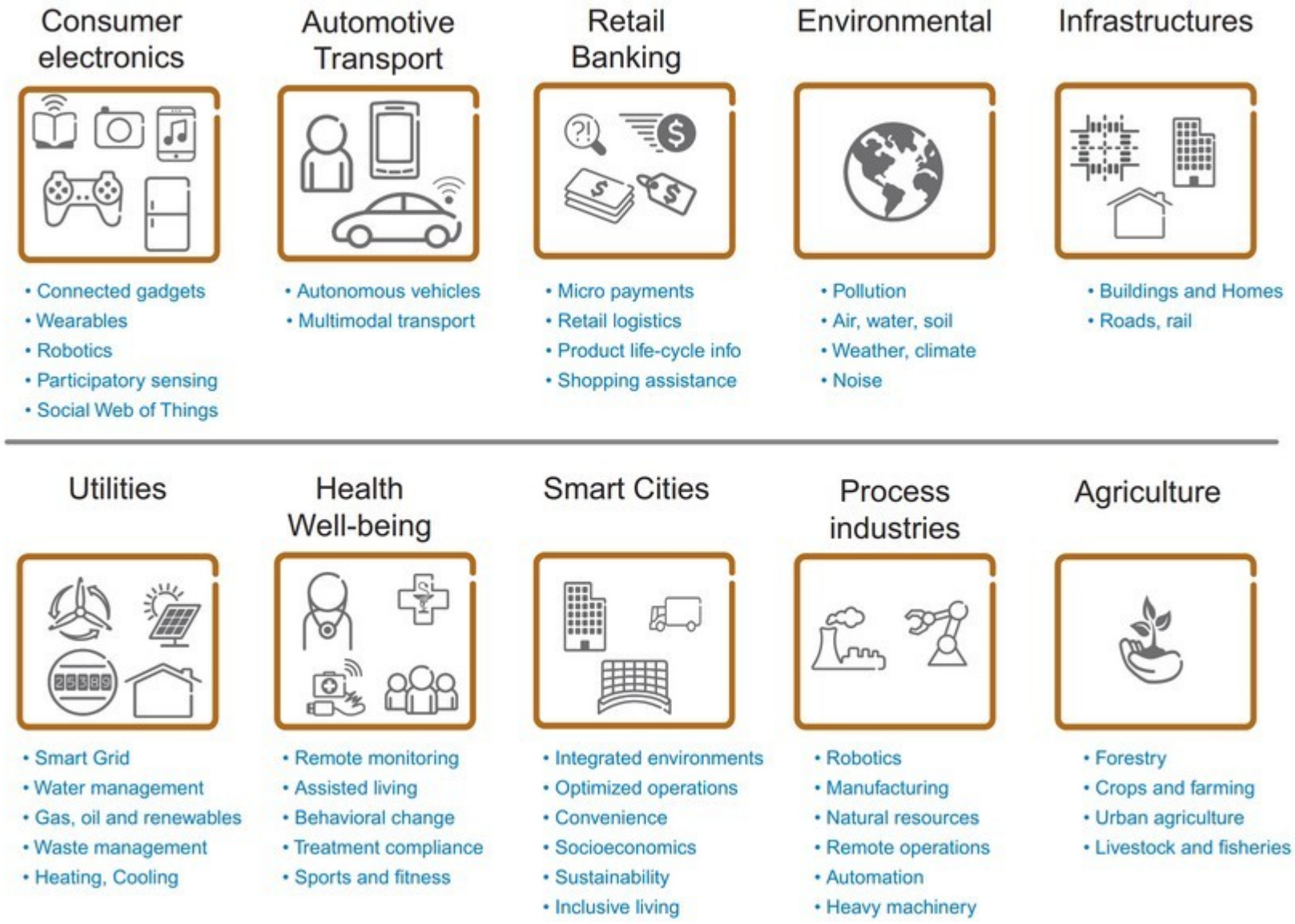
Smart Home: Amazon Echo Dot

2018



Source: http://mattturck.com/iot2018/

IoT Application Areas



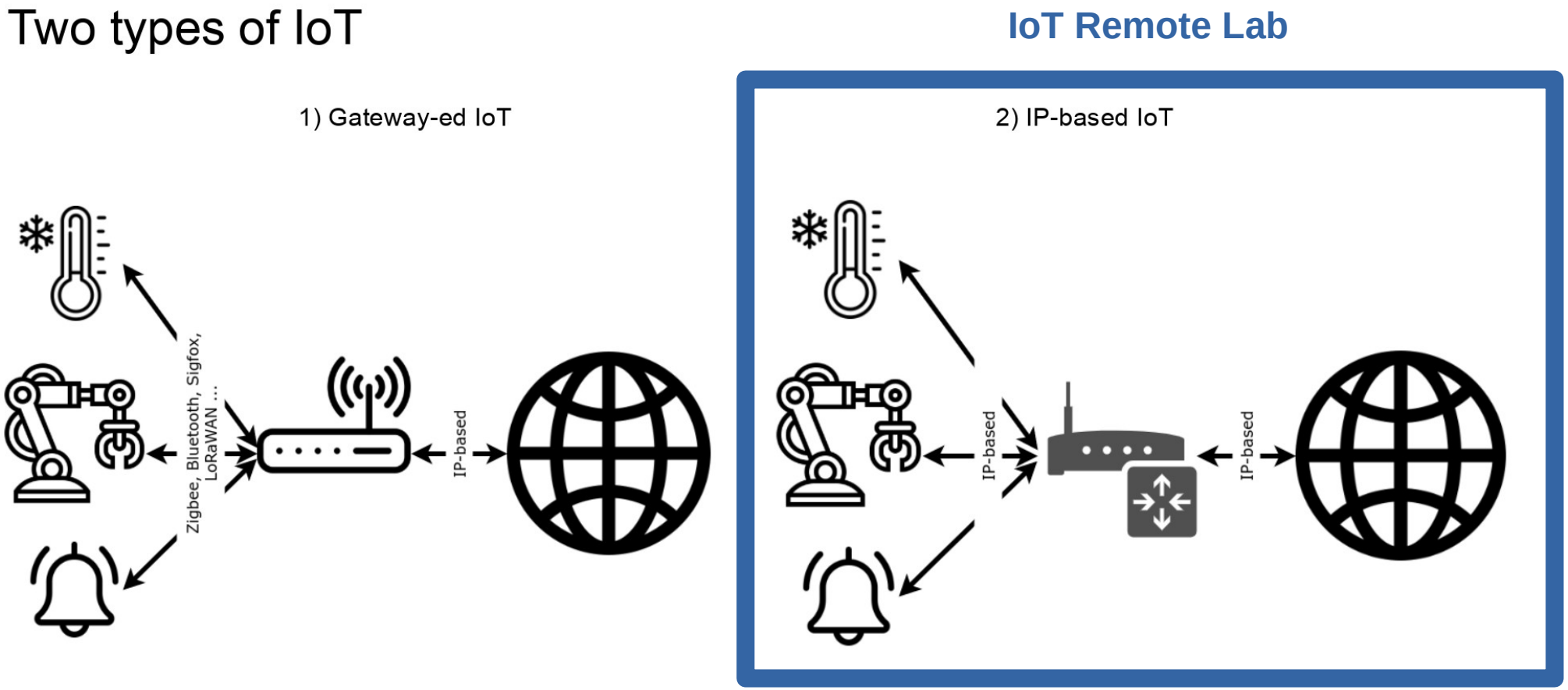
Source: Holler, Jan, et al. *From Machine-to-machine to the Internet of Things: Introduction to a New Age of Intelligence*. Academic Press, 2014.

IoT Remote Lab

… the real deal!

IoT of IoT Remote Lab

- In this lab, your tasks will be centered on IP-based IoT devices



At least, that is what you will see

IoT of IoT Remote Lab

This means that you will not learn protocols such as Zigbee, Bluetooth etc.

You will also **not** learn things such as:

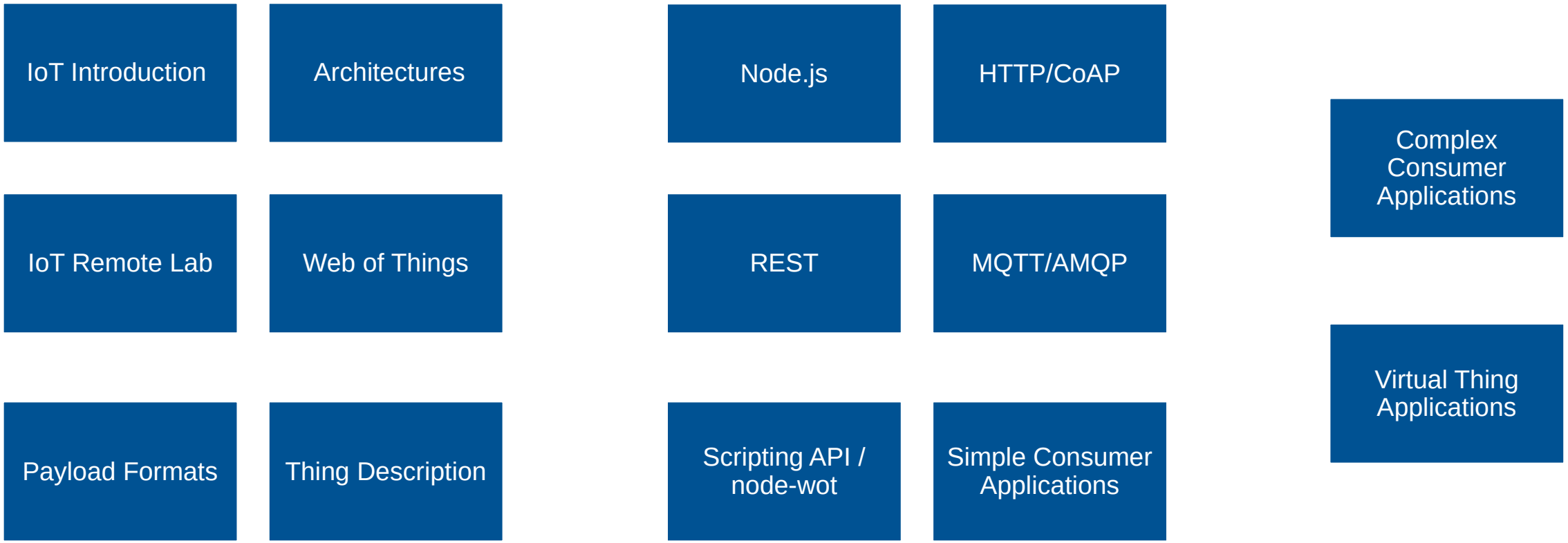
- How to make low power devices for IoT
- A new hardware architecture for IoT devices
- A new security mechanism that works on constrained devices
- Cloud architectures and platforms to collect data from IoT devices

Then what?

Course Contents (without COVID-19)

IoT Introduction	Architectures	Node.js	HTTP/CoAP	Sensors and Actuators
IoT Remote Lab	Web of Things	REST	MQTT/AMQP	Serial Protocols
Payload Formats	Thing Description	Scripting API / node-wot	Consumer Applications	Thing Applications
Deliverable 1		Deliverable 2		Deliverable 3

Course Contents (with COVID-19)



Deliverables and Grading

There is no exam, grading done based on 3 delivered homeworks.

Deliverables:

1) Preliminary exercises (~30% of the grade)
2) Client-side applications (~50% of the grade)
3) Thing-side application project OR Complex Client-side applications (~20% of the grade)

   ~5% of the grade is based on how well you use git

Since this is the second semester of this lab. There are still some rough edges in the given tasks and implementations. We appreciate all your feedback :)

Deliverables Submission

Last semester, the submissions were on Moodle. This semester we are moving to ArTEMIS (https://artemis.ase.in.tum.de/). Just log in once with your TUM credentials so that I can add you.

The course is not ready there yet.

Support Materials

We do not follow a certain book, mostly Web pages!

These include:

- Standardization documents from W3C, IETF, OASIS etc.
- Documentations of libraries
- Guide-like documents
- Nice and informative websites

You will also need to understand such documents.

We will upload the lectures to Panopto (https://tum.cloud.panopto.eu) and linked in Moodle

Development Environment

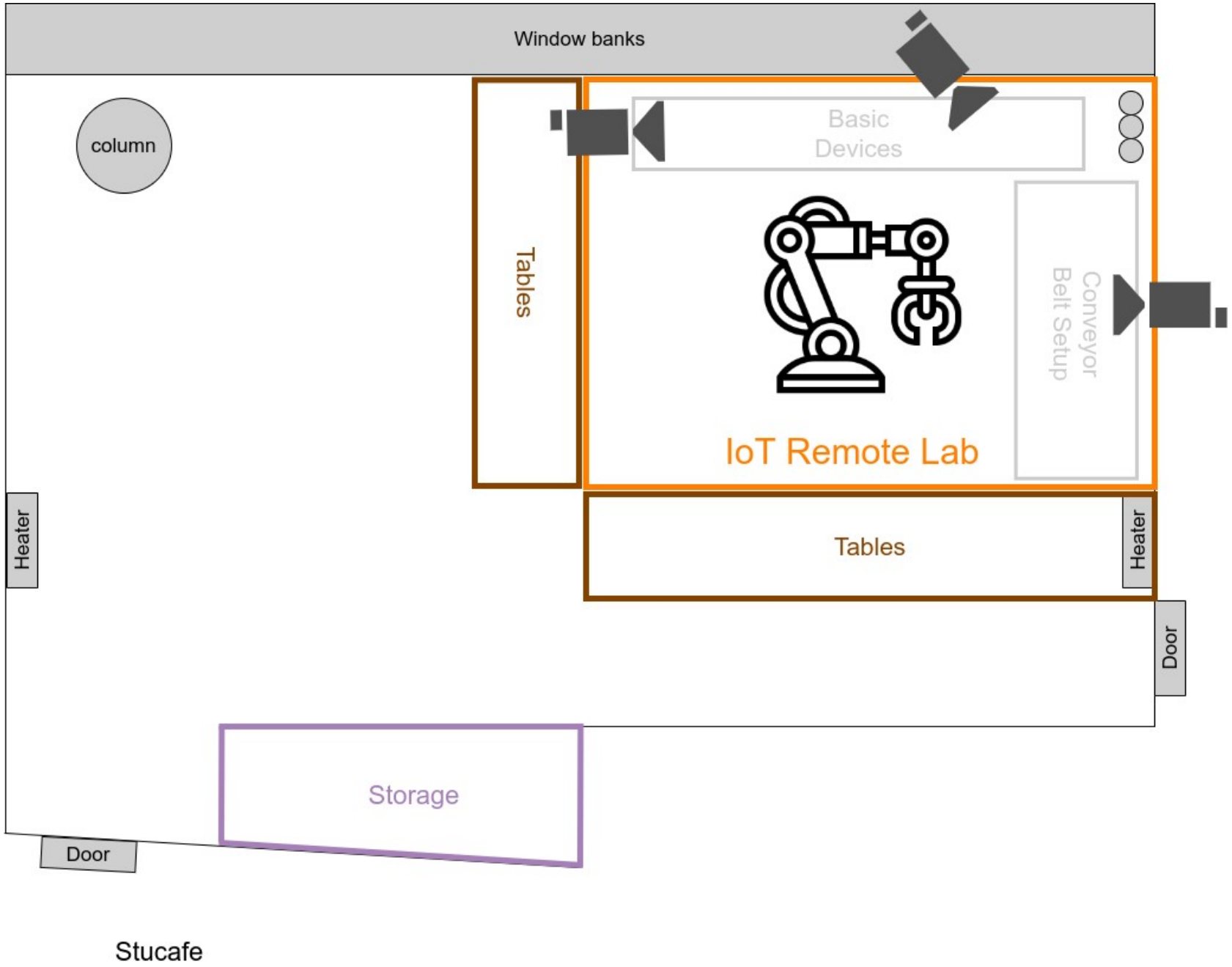
- We are not bound to a specific environment.
- Recommendation: use Visual Studio Code as an IDE/Editor
- What you will need:
  - A git repository where you can use to develop and use to send us the deliverables
  - Something to write, render/preview markdown files
  - Something to write JSON files and quickly validate them
  - Something to write Node.js code
  - Web Browser (Chrome-based or Firefox is recommended. We cannot evaluate other browsers)
  - A REST/HTTP client. Easy-to-use clients for MQTT and CoAP are also recommended

Where is the lab?!

Concretely, IoT Remote Lab

- It is in Room 3971 (above the stucafé)
  - The devices are found here and are surrounded by a wire fence
  - There are cameras that look at the devices
  - The server is here
- You can be there but you don‘t have to be there for most of the tasks
  - For interacting with robots or similar devices, we will do a slot based access.
  - Last semester, no one had to come to the university to finish the assignments

Top View


Window banks
column
Tables
Basic
Devices
Conveyor
Belt Setup
IoT Remote Lab
Heater
Tables
Heater
Door
Storage
Door
Stucafe

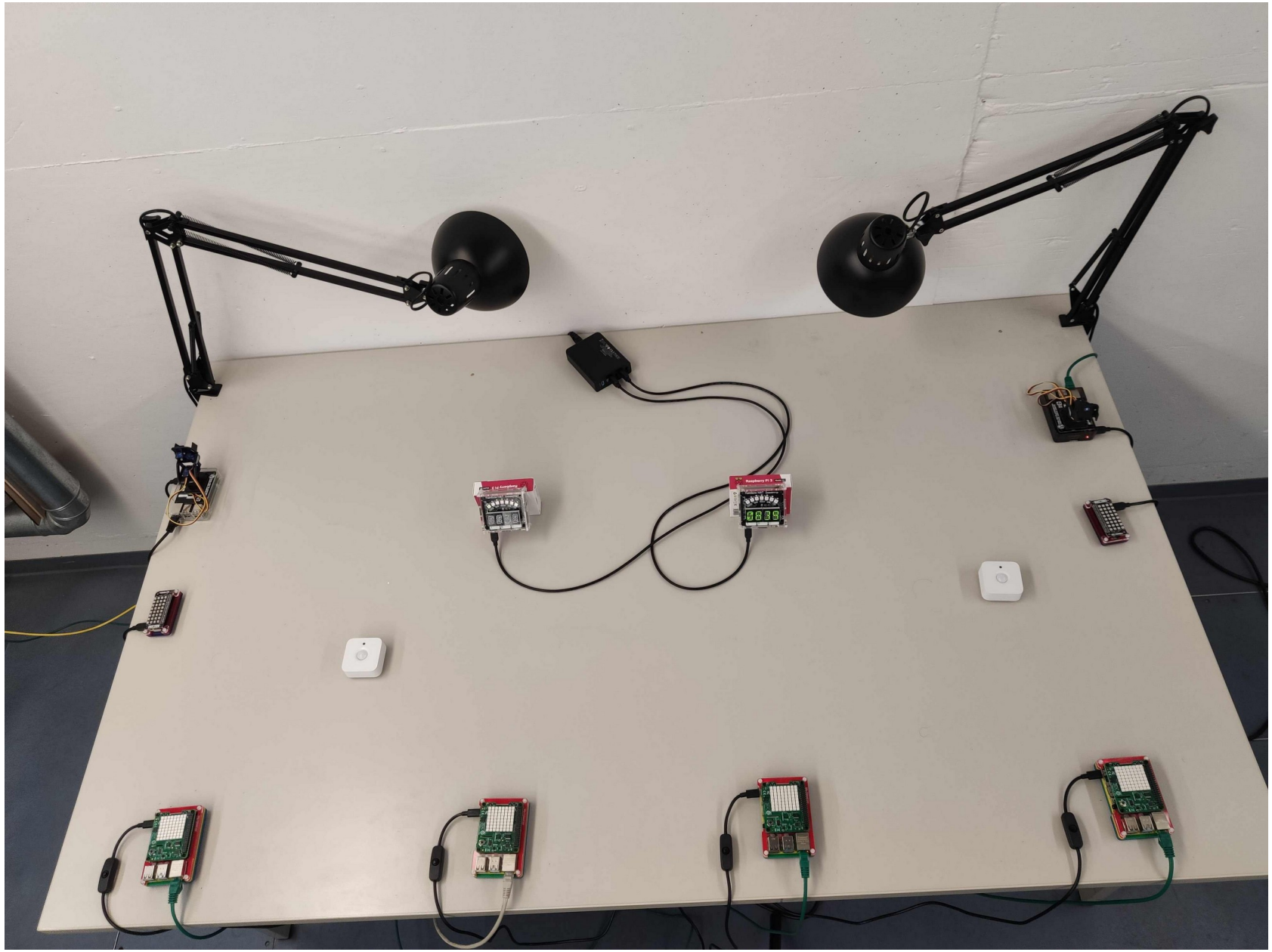
Devices (all with a certain Internet protocol)

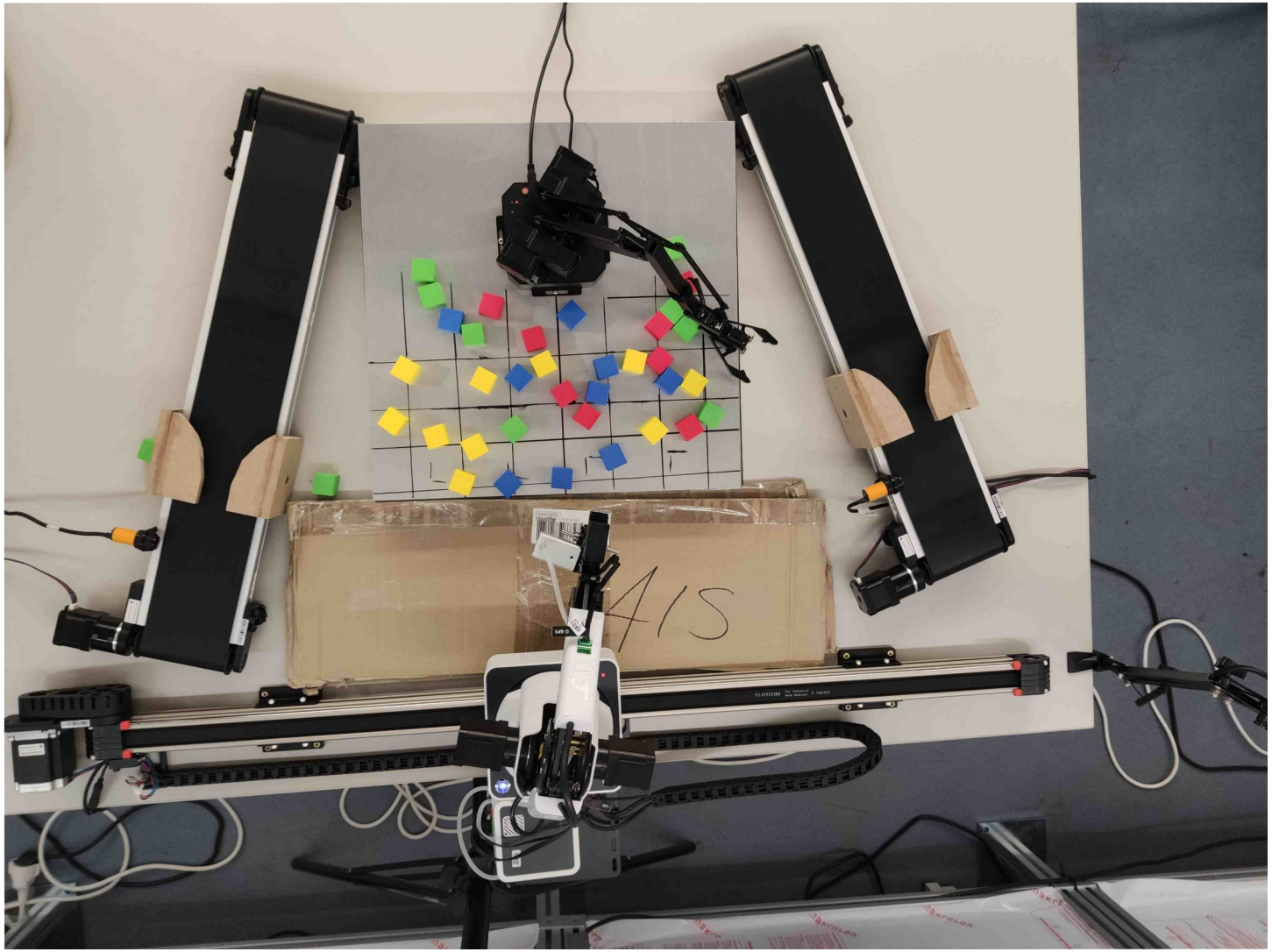
Can be used anytime:

- SenseHAT on a Raspberry Pi for temperature, humidity, pressure measurements and an LED matrix
- Philips HUE light bulbs and sensors
- Raspberry Pi with Pan and Tilt module
- Unicorn pHAT
- Rainbow HAT
- More coming this semester

Only with slot-based access:

- Dobot magician on a slider with conveyor belt setup
- Uarm Swift Pro
- Universal Robots UR10





Your Devices! (without Corona)

- The third deliverable will be to develop a device on your own.
- The implementations we choose will be added to wotify.org and be used in next semesters in IoT Remote Lab.
- An implementation should:
  - **Work: Pass our tests, boot and work automatically**
  - Have good documentation: in the code and outside of the code
  - Be interesting and different from the already existing ones

Your Devices! (with Corona)

- The third deliverable will be to develop a virtual device on your own.
- The implementations will be used in next semesters in IoT Remote Lab as simulated devices.
- An implementation should:
  - **Work: Pass our tests, start automatically**
  - Have good documentation: in the code and outside of the code
  - Be interesting and more than just replying without state management

Web Interface

- Web RTC based video streams from IP Cameras (via login)
- List of Thing Descriptions of devices (via login)
- General information about the lab (anytime)

Available at esiremotelab.esi.ei.tum.de (under maintenance for these couple weeks)

Wrap-Up

- You will be interacting with IoT devices from your computer, no ssh or VPN
- You will have 3 deliverables that will make up your grade
- Technical information/teaching starting next week!

ALL YOUR QUESTIONS SHOULD BE POSTED IN MOODLE FIRST

Contact me at ege.korkan@tum.de for other things (should be nothing much)