IoT Programming: Introduction

ΕΠΛ 428: IOT PROGRAMMING

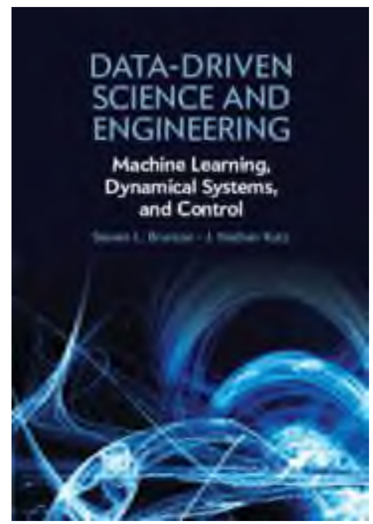
Dr. Panayiotis Kolios
Assistant Professor, Dept. Computer Science,
KIOS CoE for Intelligent Systems and Networks
Office: FST 01, 116
Telephone: +357 22893450 / 22892695
Web: https://www.kios.ucy.ac.cy/pkolios/

Πανεπιστήμιο Κύπρου

BIBLIOGRAPHY

- Pascal Hirmer
- Raj Kamal

- S. Brunton, N. Kutz
- E. Lee, S. A. Seshia



Πανεπιστήμιο Κύπρου

- ☐ IoT Fundamentals: Networking Technologies, Protocols, and Use Cases for the Internet of Things, David Hanes, Gonzalo Salgueiro, Patrick Grossetete, Robert Barton, Jerome Henry, 2017, Cisco Press.

- ☐ Time Series Analysis with Python

```
@misc{bianchi2024tsbook,
  author      = {Filippo Maria Bianchi},
  title       = {Time Series Analysis with Python},
  year        = {2024},
  howpublished = {Online},
  url         = {https://github.com/FilippoMB/python-time-series-handbook}
}
```

COURSE CONTENT

1. Internet of Things (IoT) definition and objectives
2. Applications, use case scenarios and value propositions
3. IoT Architecture and frameworks
4. IoT system blocks: edge, fog, cloud
5. Communications aspects for IoT systems: Internet infrastructure; radio access networks
6. IoT management tools and cybersecurity
7. IoT Devices: sensors, actuators and embedded systems
8. Time-series analysis and prediction
9. Decision and optimization
10. Dynamic systems and control

- **Information Technology** (IT) supports systems for interconnections (Internet) and along with data processing and storage
- **Operational Technology** (OT) is employed to monitor and controls devices and processes on physical systems
    - include assembly lines, utility distribution networks, production facilities, roadway systems
- Traditionally, OT has used dedicated networks with specialized communications protocols to connect these devices / systems
- OT operations are critical:
    - if autonomous car fails, people are in danger
    - if email server fails it may irritate people
- IT & OT merging for efficiency, ease of deployment, etc

Internet of Things

a network of physical things (devices) communicating information using the Internet or other telecommunications technologies and thus enabling monitoring, coordinating or controlling process in the physical environment.

- Convergence of IT & OT poses several challenges
- Scale (consider smart meters for utility operators)
- Quality of Service and Reliability
- Security (threat surface is now greatly expanded)
- Privacy (data gathered on activities of individuals)
- Data analytics (deluge of data and big data analytics)
- Interoperability (various protocols and architectures)
- Availability: Software updates without downtime

PLAN OF THE COURSE

- **Part One – Architecture**

  Process of data and packet communications, as well as protocol layers, TCP/IP, and data networks.

- **Part Two – Sensing & Perception**

  All of the relevant information about the process of sending a wireless signal and combating the effects of the wireless channel.

- **Part Three – Reasoning & Planning**

  Details on IEEE 802.11, IEEE 802.15, Bluetooth, the Internet of Things, and ZigBee.

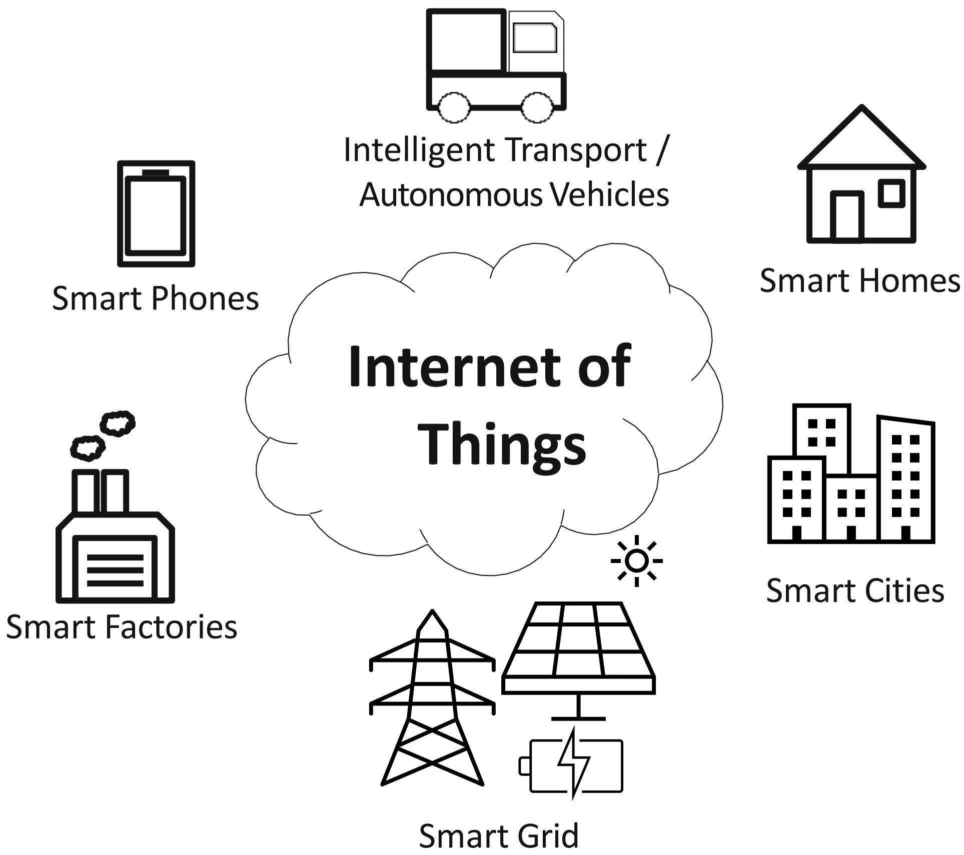
- **Part Four – Control and Actuation**

  Mobile cellular systems principles, LTE, smartphones

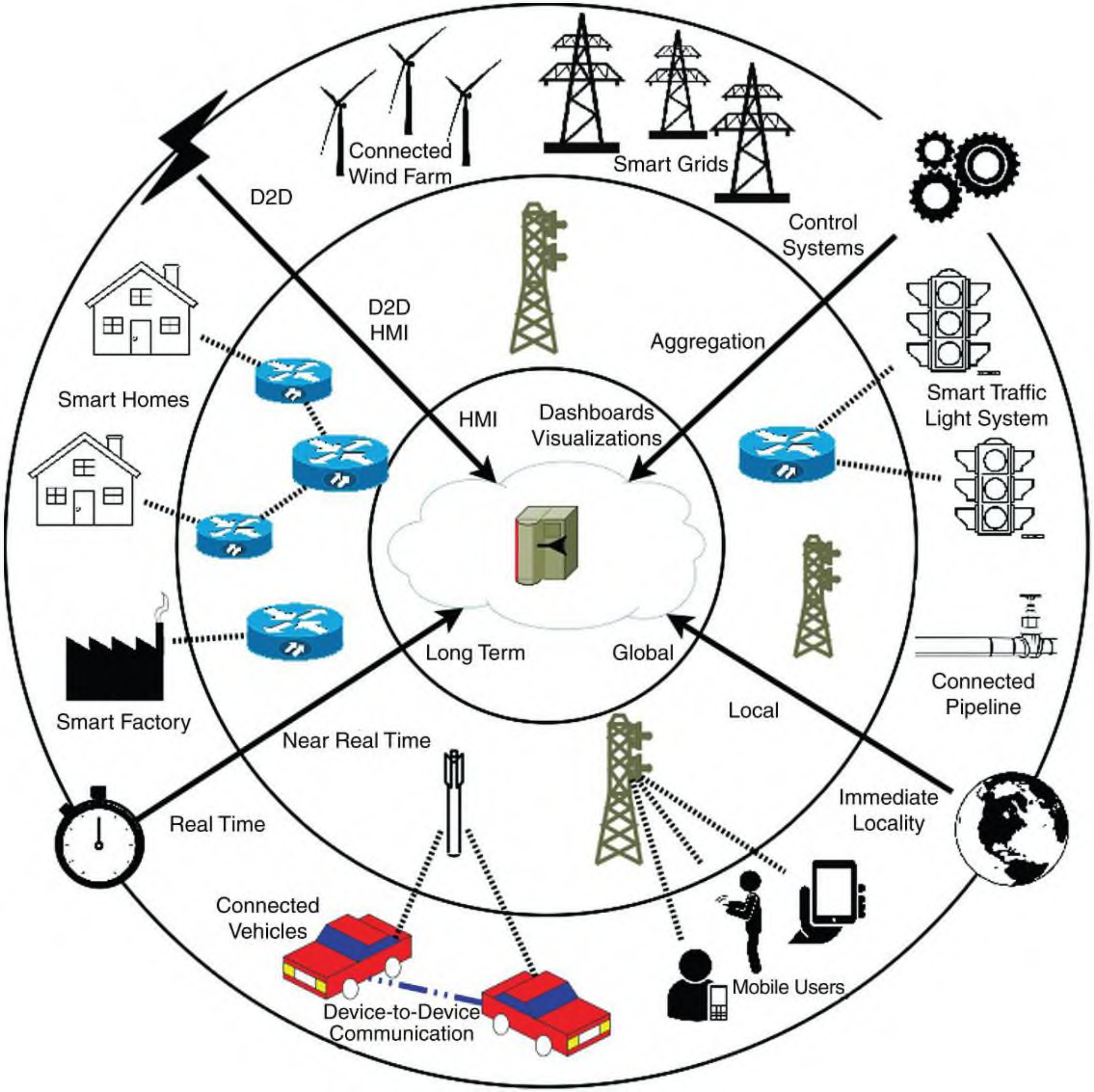
  Mobile applications and app development

  Long-range communications using satellite, fixed wireless, and WiMAX.

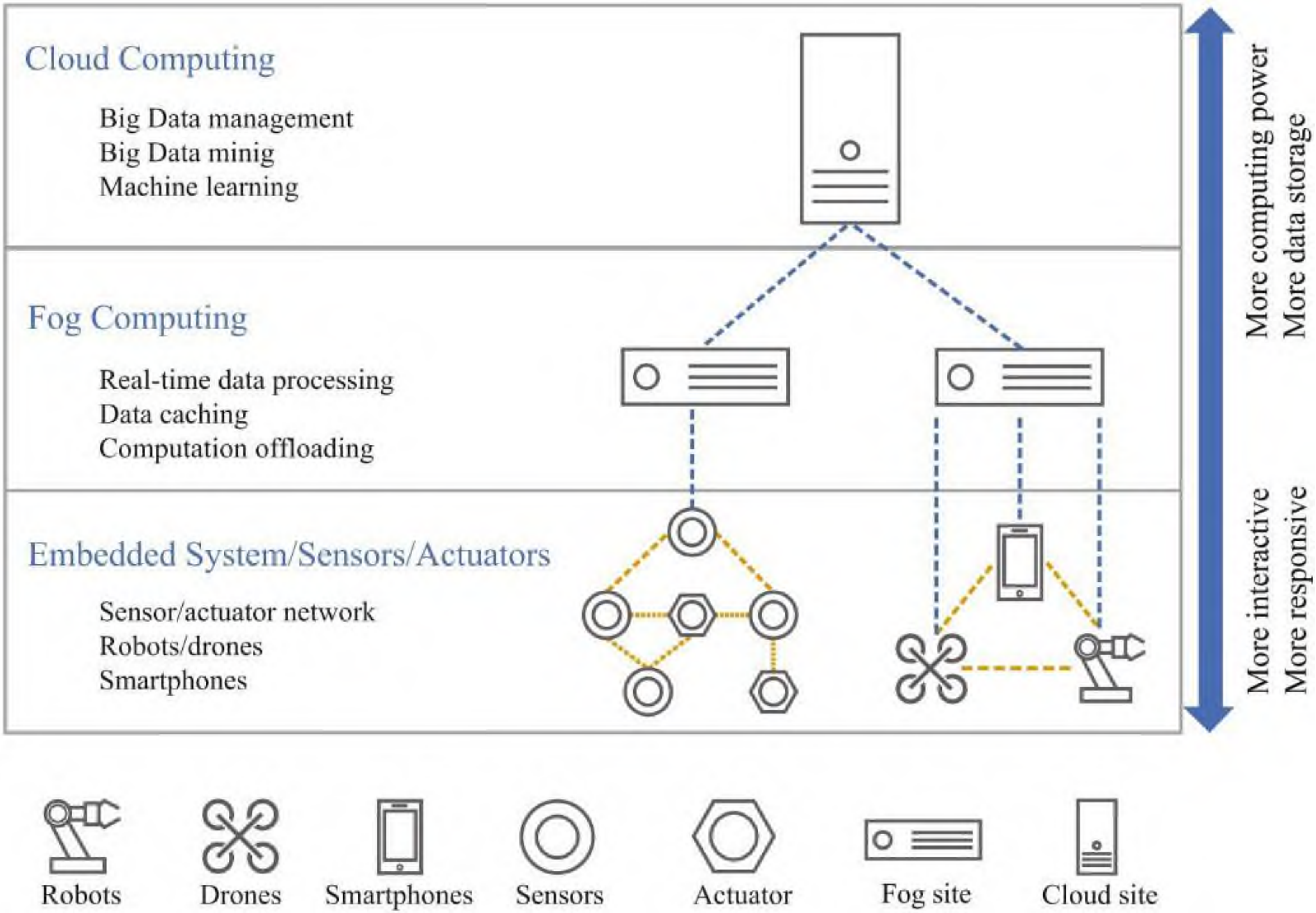
Πανεπιστήμιο Κύπρου



- Heterogeneous **Devices** communicate using Internet protocols to reach common goals. Various sensors and actuators can be attached to these **Devices**.

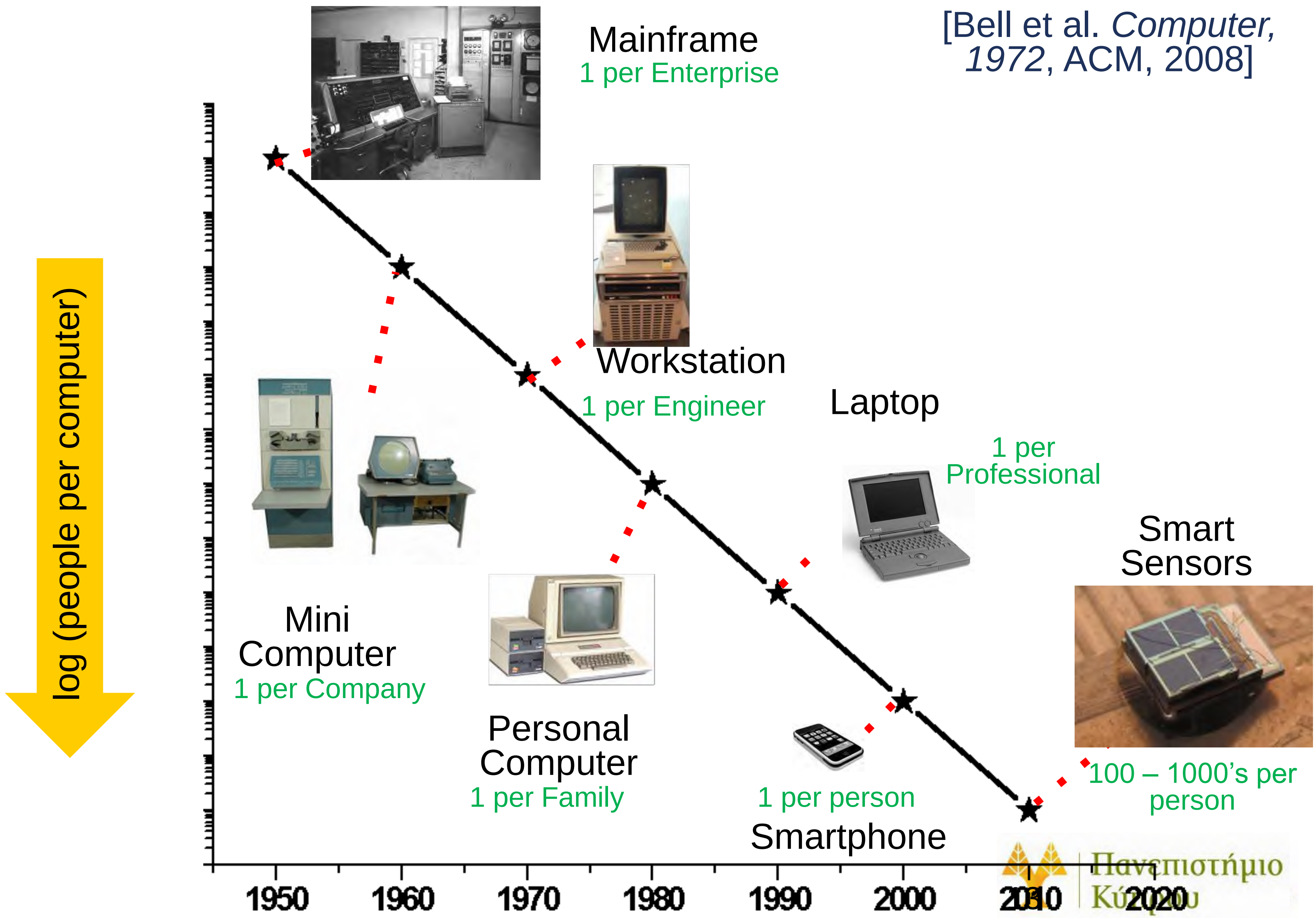


Πανεπιστήμιο Κύπρου

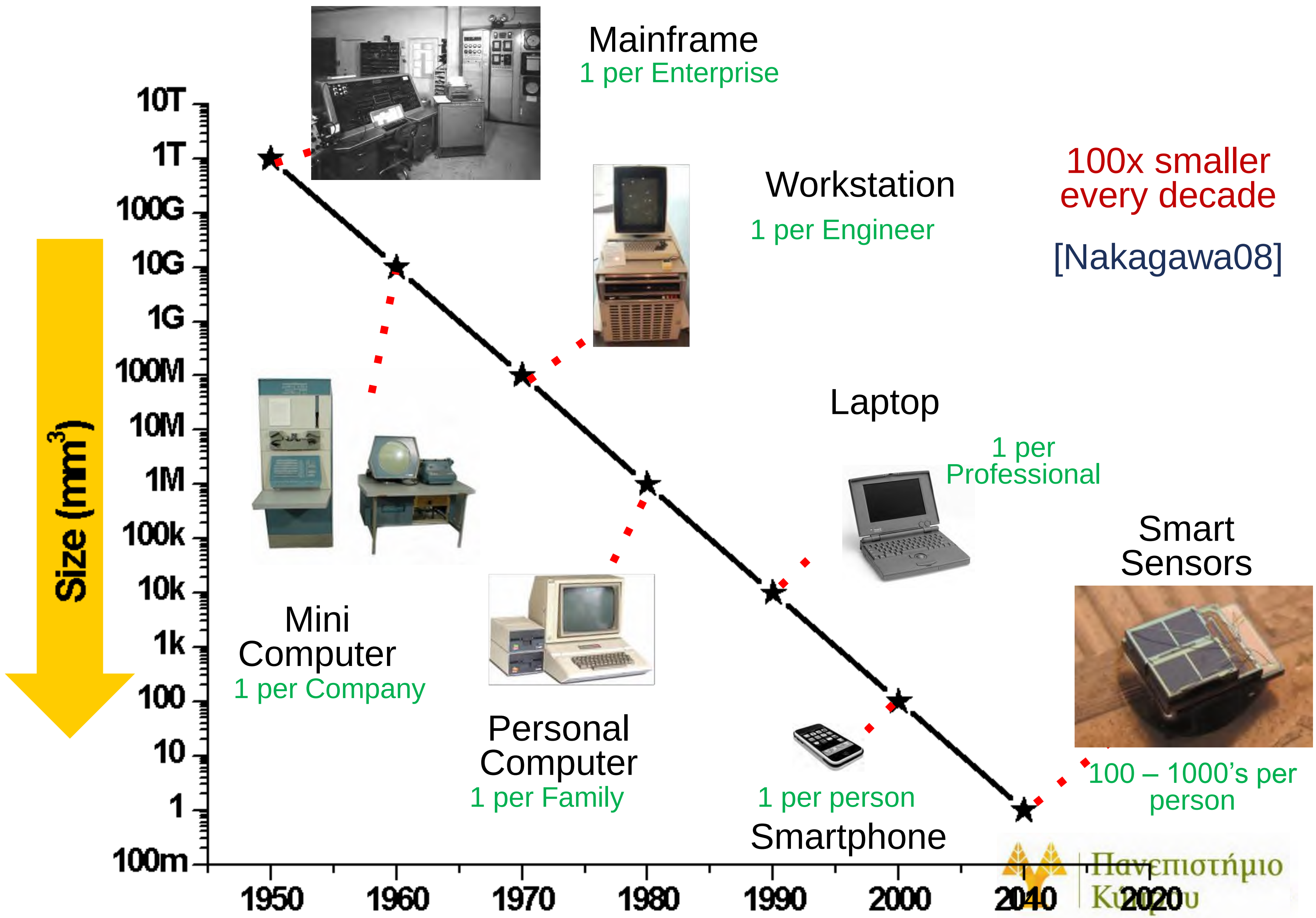


Πανεπιστήμιο Κύπρου

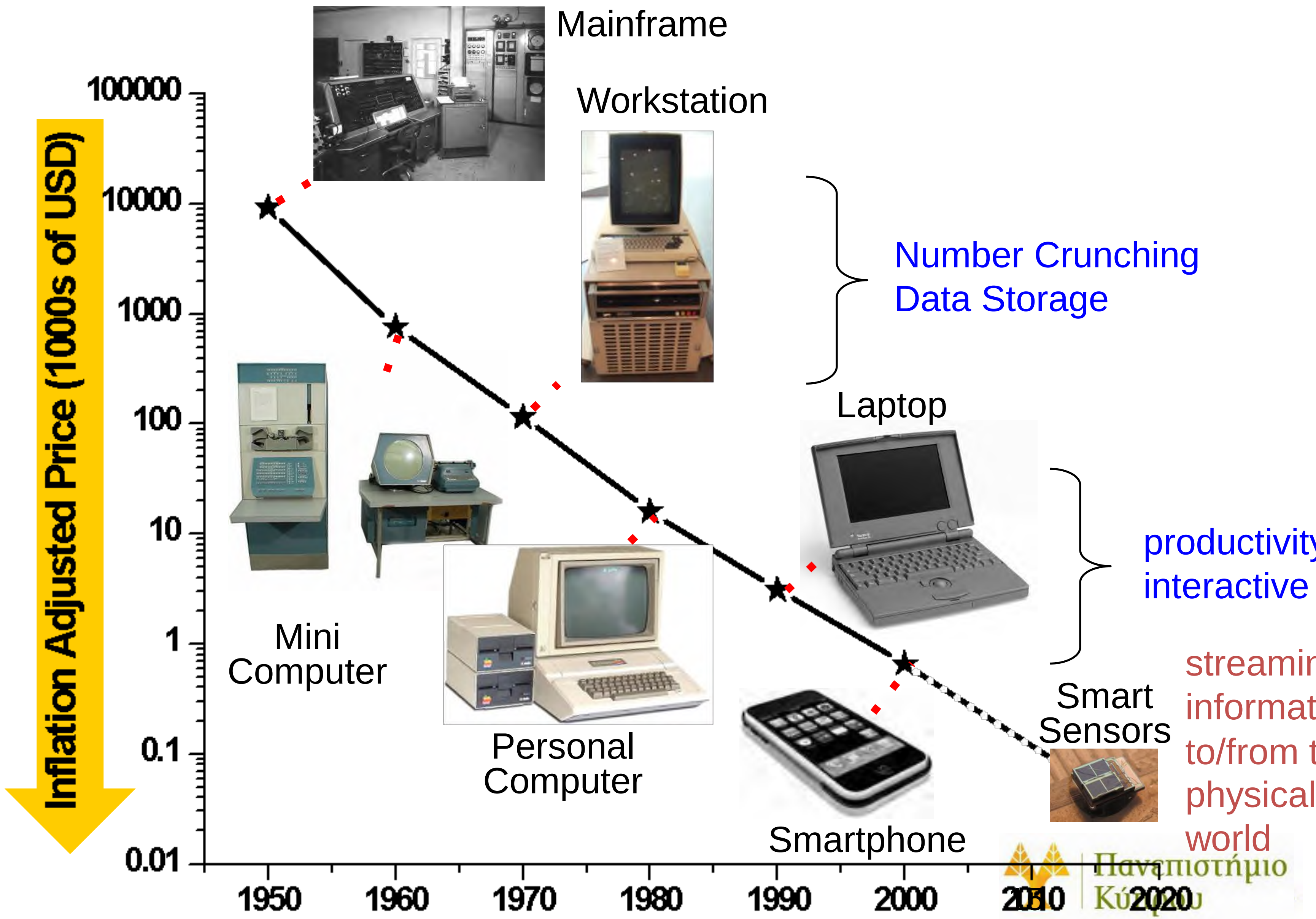
Number of Computers/Person Grows Over Time



Computer Volume Shrinks by 100x Every Decade



Price falls dramatically, enabling new applications



It's not just information technology anymore:

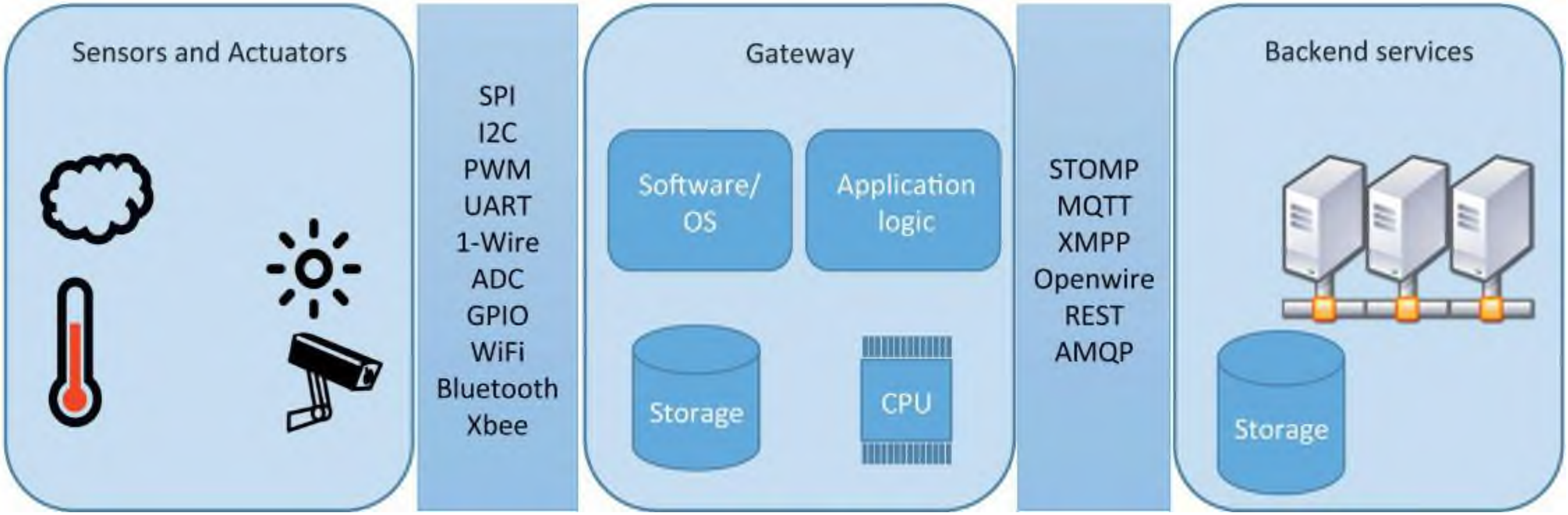
- Cyber + Physical
- Computation + Dynamics
- Security + Safety

Contradictions:

- Adaptability vs. Repeatability
- High connectivity vs. Security and Privacy
- High performance vs. Low Energy
- Asynchrony vs. Coordination/Cooperation
- Scalability vs. Reliability and Predictability
- Laws and Regulations vs. Technical Possibilities
- Economies of scale (cloud) vs. Locality (fog)
- Open vs. Proprietary
- Algorithms vs. Dynamics

Innovation:

Cyber-physical systems require new engineering methods and models to address these contradictions.


Sensors and Actuators
SPI
I2C
PWM
UART
1-Wire
ADC
GPIO
WiFi
Bluetooth
Xbee
Gateway
Software/
OS
Application
logic
Storage
CPU
STOMP
MQTT
XMPP
Openwire
REST
AMQP
Backend services
Storage

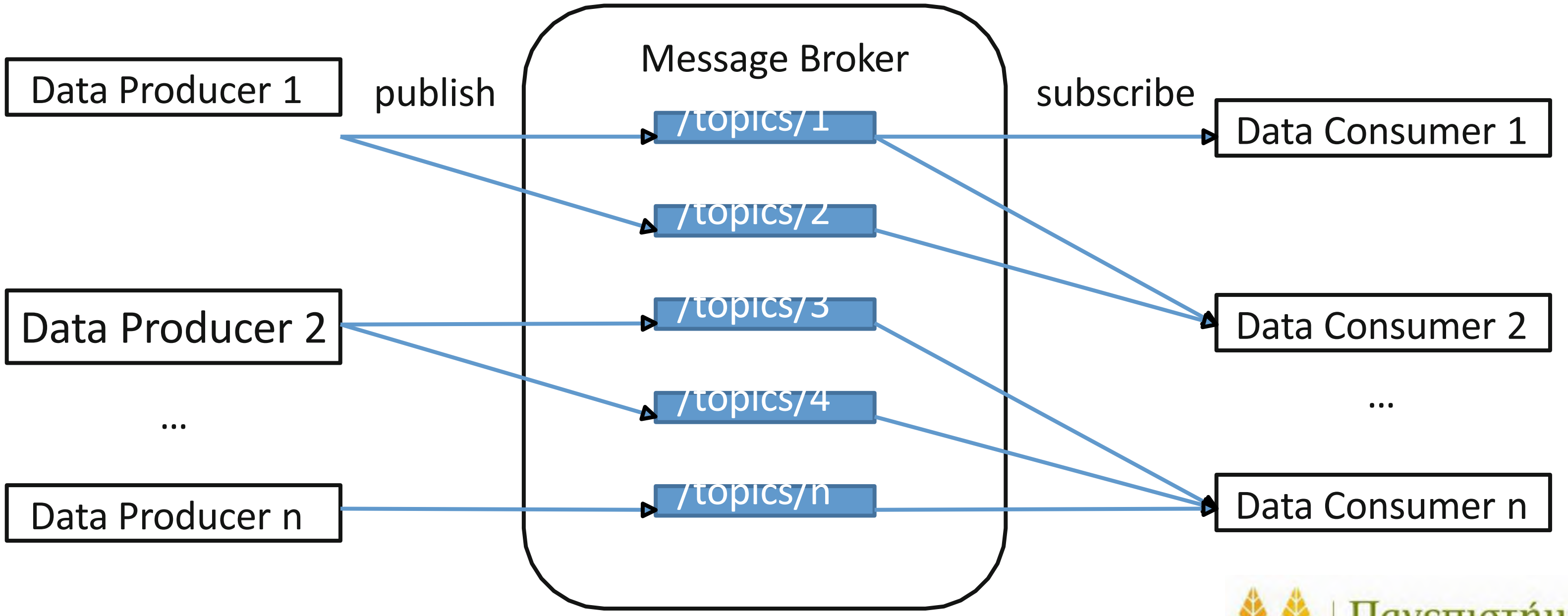
- An Internet of Things **Device** is the hardware component offering computing capabilities and resources (e.g., CPU, RAM) as well as wireless or wired network interfaces (e.g., LAN, WiFi, Bluetooth, Zigbee, LoRa).
- A device can be connected to or be embedded with a multitude of different sensors and actuators.
- A device is capable to connect to the sensors and actuators through hardware interfaces (e.g., GPIO) and run programs to extract, pre-process, interpret and transfer data to other devices or other receiving entities in the Internet (e.g., dashboard applications operated in the cloud).
- A device is either plugged in or most frequently battery powered and thus need to duty-cycle for prolonged operation

Πανεπιστήμιο Κύπρου

- A **Sensor** does not provide any computing or network capabilities. A sensor is capable of measuring certain metrics of the environment, e.g., temperature, humidity, velocity, etc. A sensor can be analog and digital and can be accessed through the hardware interfaces of devices.
- An **Actuator** does not provide any computing or network capabilities. An actuator is capable of controlling the environment, e.g., turn on a switch. An actuator can be analog and digital and can be accessed through the hardware interfaces of devices.

- For exchanging messages in the Internet of Things, two communication paradigms have been established:
  - publish-subscribe
  - request- response
- Both paradigms come with different protocols and should be selected based on the requirements of the scenarios

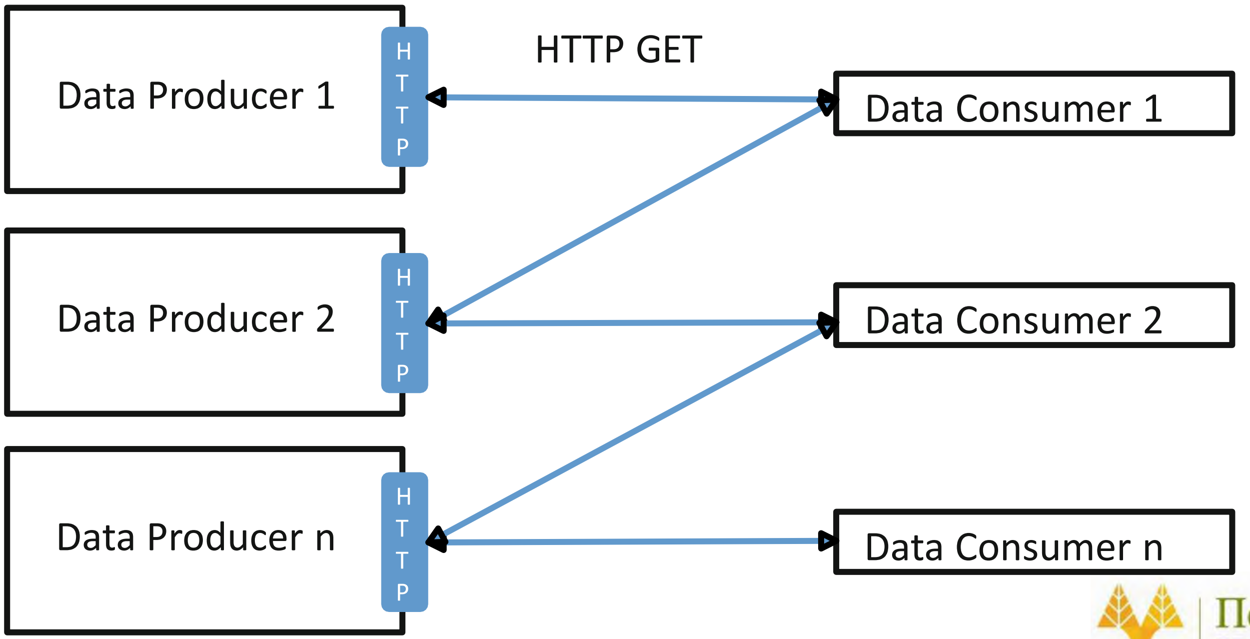
- Consists of two entities:
  - (1) the publisher, i.e. the data producer,
  - (2) one or more subscribers, i.e., the data consumers.
- Achieves one-to-many communication, i.e. multiple devices consume messages generated by one device, for example, an autonomous vehicle communicating a dangerous situation to the other traffic participants.



Πανεπιστήμιο Κύπρου

- Message broker is the central component being responsible for distributing messages to subscribers
- The publishers send messages to a so-called topic. A topic is a hierarchical path that is used by the subscribers to define which kind of data should be received.
    - *<level_1>/<level_2>/…/<level_n>*
    - *e.g. myhome / livingroom / temperature*
- A famous example for a protocol realizing publish-subscribe is MQTT which defines quality of service parameters for delivering messages as:
    - at-most-once / exactly-once / at-least-once
- Famous implementations are Mosquitto, RabbitMQ, or Apache Kafka
- Many of these implementations are very lightweight and use UDP instead of TPC to ensure a high performance.

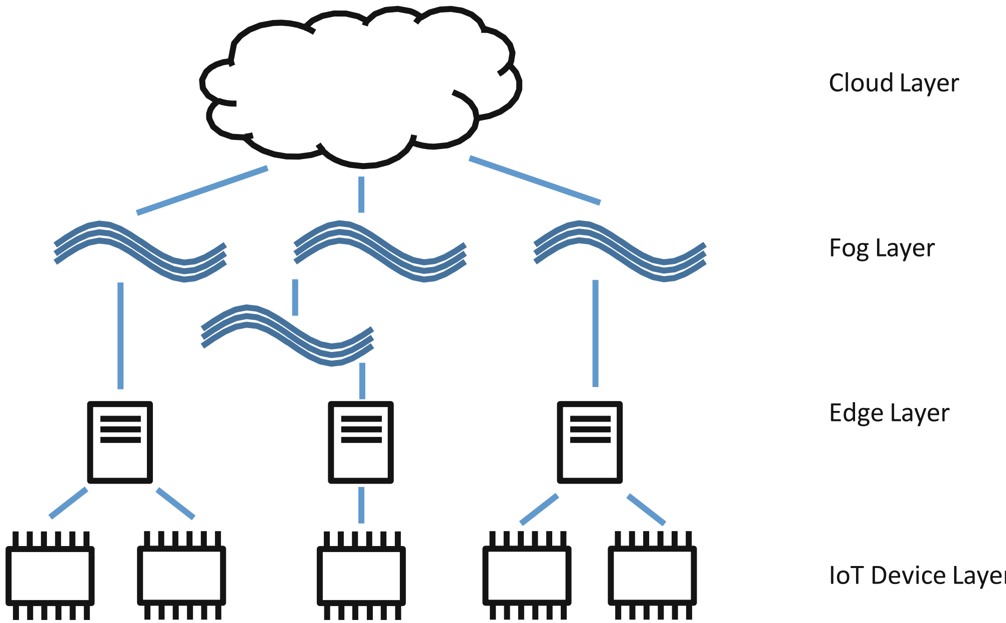
- 1-to-1 communication and message exchange
  - more robust and secure since the recipients are well-known and, usually, the retrieval of each message is acknowledged by the recipient (message broker not required)
  - Request-Response is usually done using TPC and HTTP
  - However lead to a significant overhead when sending the same message to several recipients and the address (e.g., URL) of the recipient needs to be known



Πανεπιστήμιο Κύπρου

- **LoRa** is a long-range low data rate protocol favored in smart city applications. LoRa requires less energy than other communication standards, especially over larger distances. Disadvantages of LoRa comprise a rather low transmission rate.
- **Bluetooth** Low Energy (BTLE) is a very low energy short range protocol that has become the standard for communication in Smart Homes or in communicating with wearables
- **ZigBee** has become a very well adapted standard for communication in Smart Homes. Its advantages are similar to BTLE the very low energy consumption and offers a longer range than BTLE with 100–300 m line- of-sight. Zigbee however, requires a gateway that provides an ad-hoc network
- **WiFi** is a high data rate higher energy consumption protocol that has also been extensively used in Smart Homes
- **5G** networks are the new communication standard for high-bandwith and fast communication, which enables new applications, such as autonomous cars with the need to communicate constantly and transfer a large amount of data. 5G is very reliable, has a large bandwidth and distance and offers many features, such as device discovery and localization. However, it is a licensed technology and has a relatively high energy consumption

Edge and Fog Computing

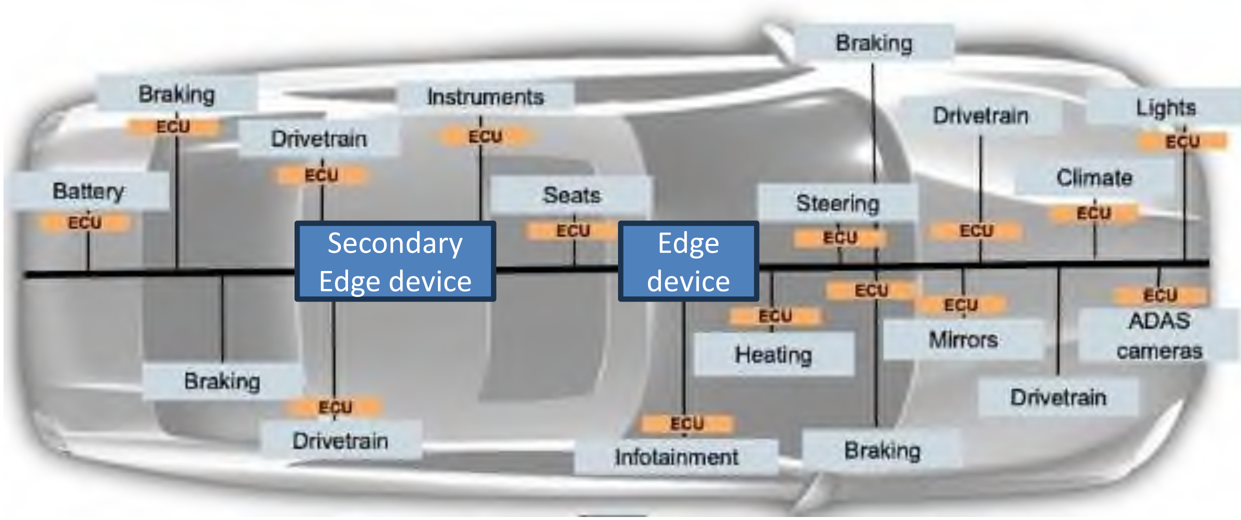


- IoT devices tend to have limited computing resources
- Hence data and computing operations could be transferred to another location, such as a Cloud, to be processed.
- However, using distant backend clouds, entail high network latency, which hinders building real-time
- Thus Edge and Fog Computing have been introduced

Πανεπιστήμιο Κύπρου

ELECTRONIC CONTROL UNITS (ECU)

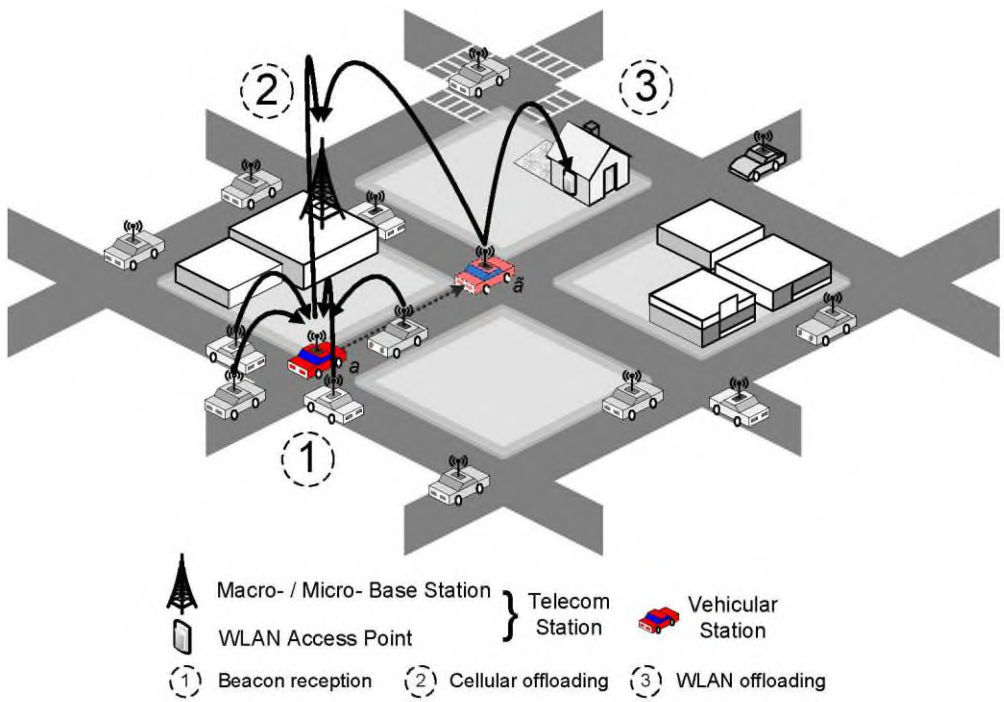
- 70 to 100 ECUs in modern luxury cars, close to 100M LOC



- located very close to sensors and actuators to minimize latency
- Edge devices are able to aggregate, process, and interpret the data from all ECUs of the car

Πανεπιστήμιο Κύπρου

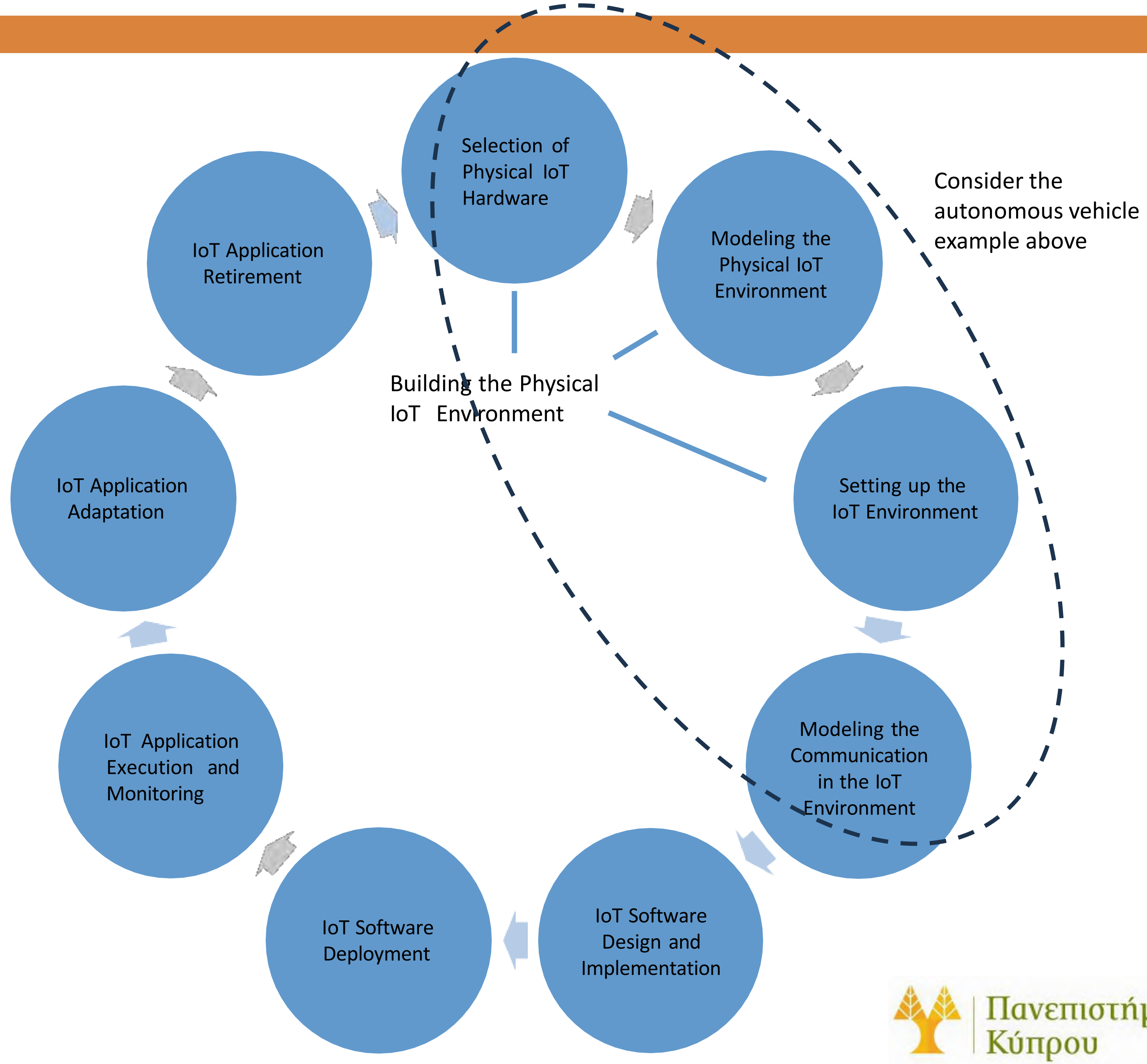
- Fog Computing provides scalable computing infrastructure, close to where is needed.
- Multiple edge devices communicate with the fog hardware and transfer data to be processed or analyzed.
- For long-term analysis and storage, data is transferred from the fog environment to backend clouds.
- In this case latency and efficiency usually are no issue



Πανεπιστήμιο Κύπρου

LIFE CYCLE OF IOT APPLICATIONS

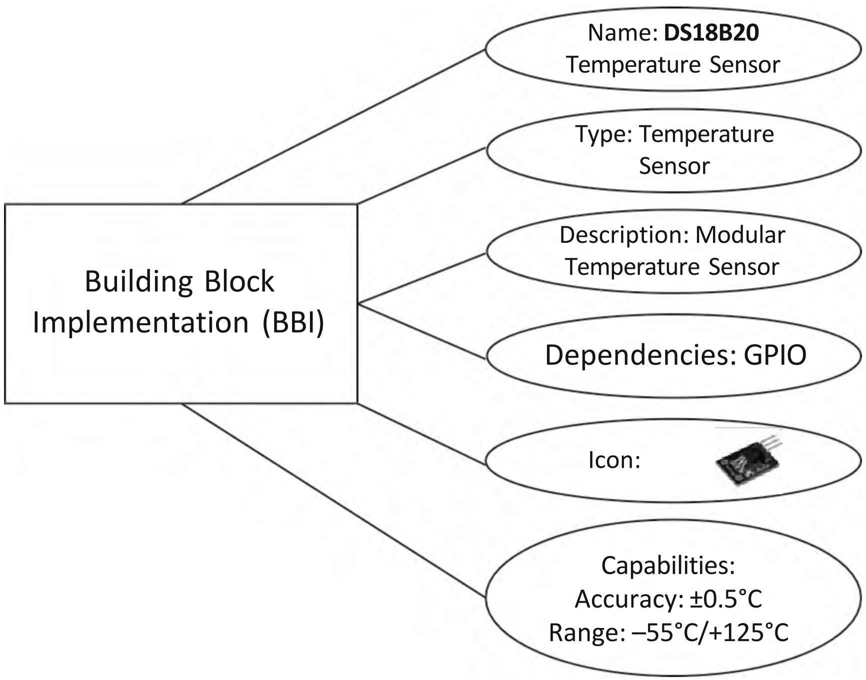
Πανεπιστήμιο Κύπρου



Πανεπιστήμιο Κύπρου

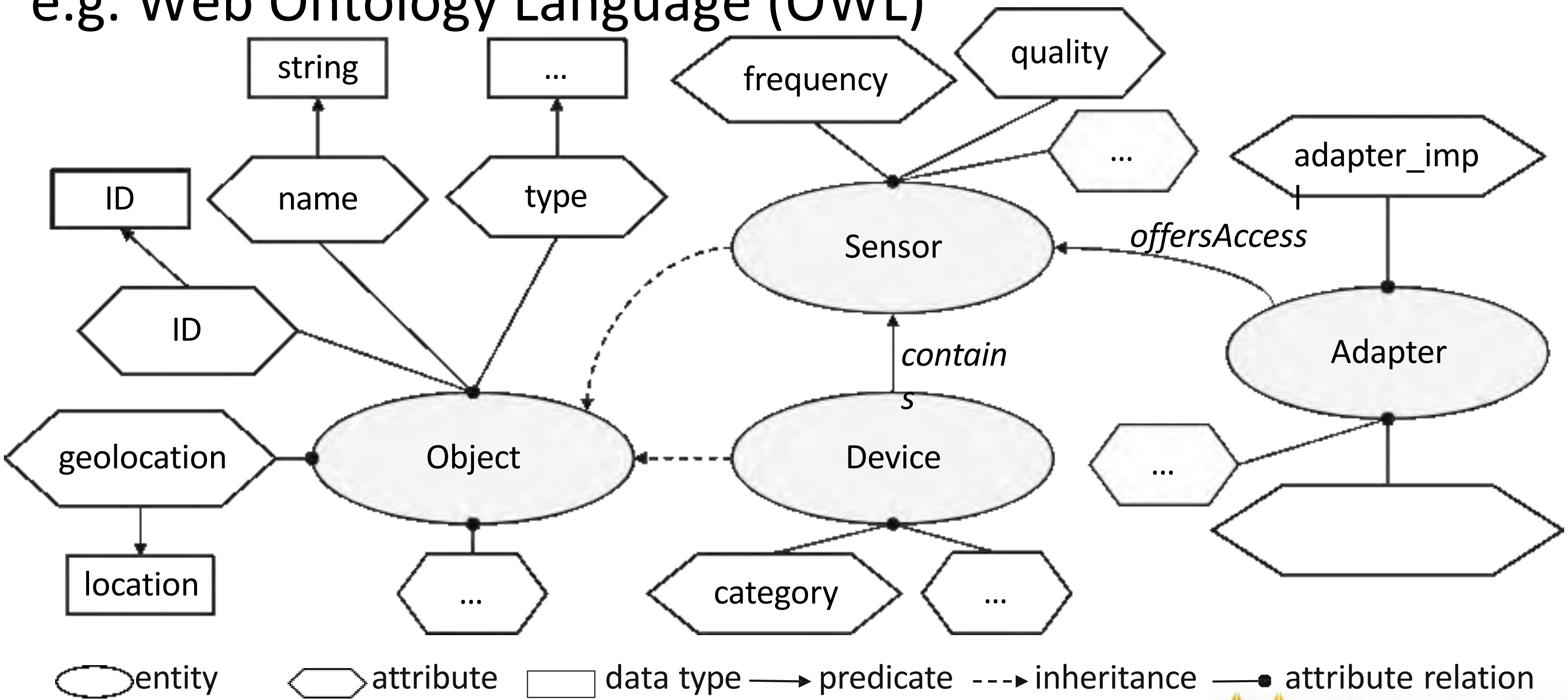
STEP 1: IOT HARDWARE

- DEVICE – SENSOR - ACTUATOR



- Network technology (e.g., WIFI, BLE)
- Costs, efficiency, security, privacy application requirements

- IoT environment model contains representations of
  I. devices, sensors, and actuators of the IoT environment
  II. the connections among them
- Ontology models describe semantics between entities e.g. Web Ontology Language (OWL)

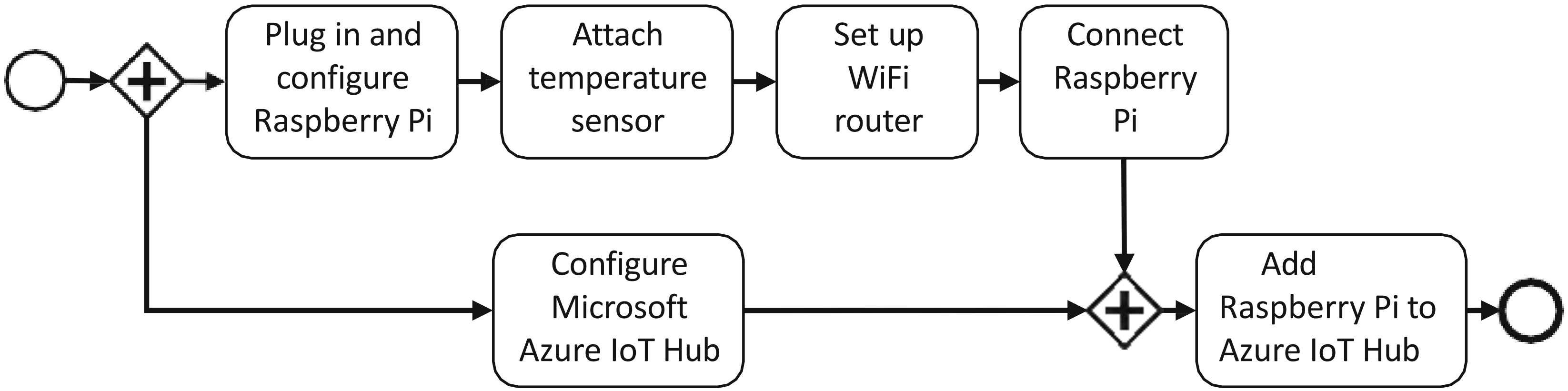


- XML representation provides structure with description and unique identifiers

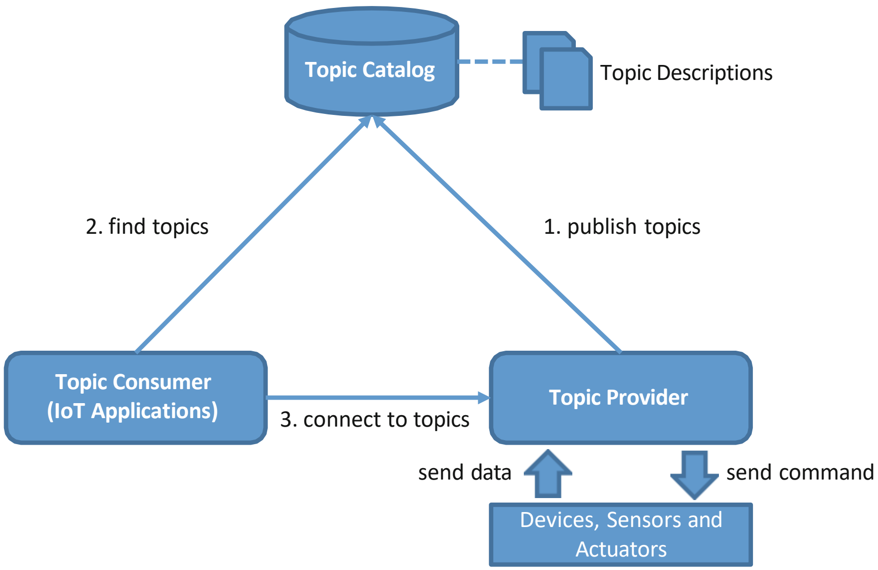
```
  </sml:output>
  </sml:OutputList>
  </sml:outputs>

  <!-- Sensor Location -->
  <sml:position>
  <gml:Point gml:id="stationLocation"
  srsName="http://www.opengis.net/def/crs/EPSG/0/4326">
  <gml:coordinates>47.8 88.56</gml:coordinates>
  </gml:Point>
  </sml:position>
 </sml:PhysicalComponent>
```

Πανεπιστήμιο Κύπρου

- After selecting the physical hardware components
- After creating the IoT Environment Model
    - Then setting up the IoT hardware
    - e.g. attaching a temperature sensor to a Raspberry Pi and connecting it via WiFi to the Microsoft Azure IoT Hub



- Use of abstraction through topics
- Serve as endpoints to access device’s data or control devices
- Usually multiple topics per device
- Provide means to describe and find topics
    - holistic topic descriptions
    - topic catalog to browse the topic descriptions
    - an effective way to find suitable topics that offer access to the devices’ sensors and actuators



Πανεπιστήμιο Κύπρου

- Topic descriptions can have the following attributes
  - **data type:** type of values provided by the topic, e.g. boolean, true if parking space occupied
  - **location:** contains the location type and value, e.g. GPS
  - **message format:** format of message provided, e.g. JSON, or XML
  - **message structure:** metamodel type and version, e.g., JSON schema or XML schema
  - **middleware endpoint:** endpoint hosting the topic, e.g., broker running on a server
  - **owner:** topic provider, e.g., City of Stuttgart
  - **path:** of the topic, e.g., /parking-space-monitor
  - **protocol:** exchange protocol, e.g. MQTT or HTTP

- Topics can be found, based on a *location*, sensor or actuator *type*, data *types*, or data *units*

- Example of a topic description based on JSON


```
{ "data_type": "boolean",
  "hardware_type": "occupation detection sensor",
  "location": {
     "location_type": "city name",
     "location_value": "Stuttgart"
  },
  "message_format": "JSON","m
  essage_structure": {
   "metamodel_type": "JSON schema",
   "metamodel":"{"title": "provider_schema",
              "type": "object","properties":{
                "value": {"type": "boolean"},
                "timestamp": {"type": "integer"},
                "time_up": {"type": "string"} },
              "required": ["value", "timestamp"]}"
  },
  "middleware_endpoint": "http://example.com",
  "owner": "city-of-stuttgart",
  "path": "/parking-space-monitor",
  "protocol": "MQTT",
  "topic_type": "subscription"
}
```


Πανεπιστήμιο Κύπρου

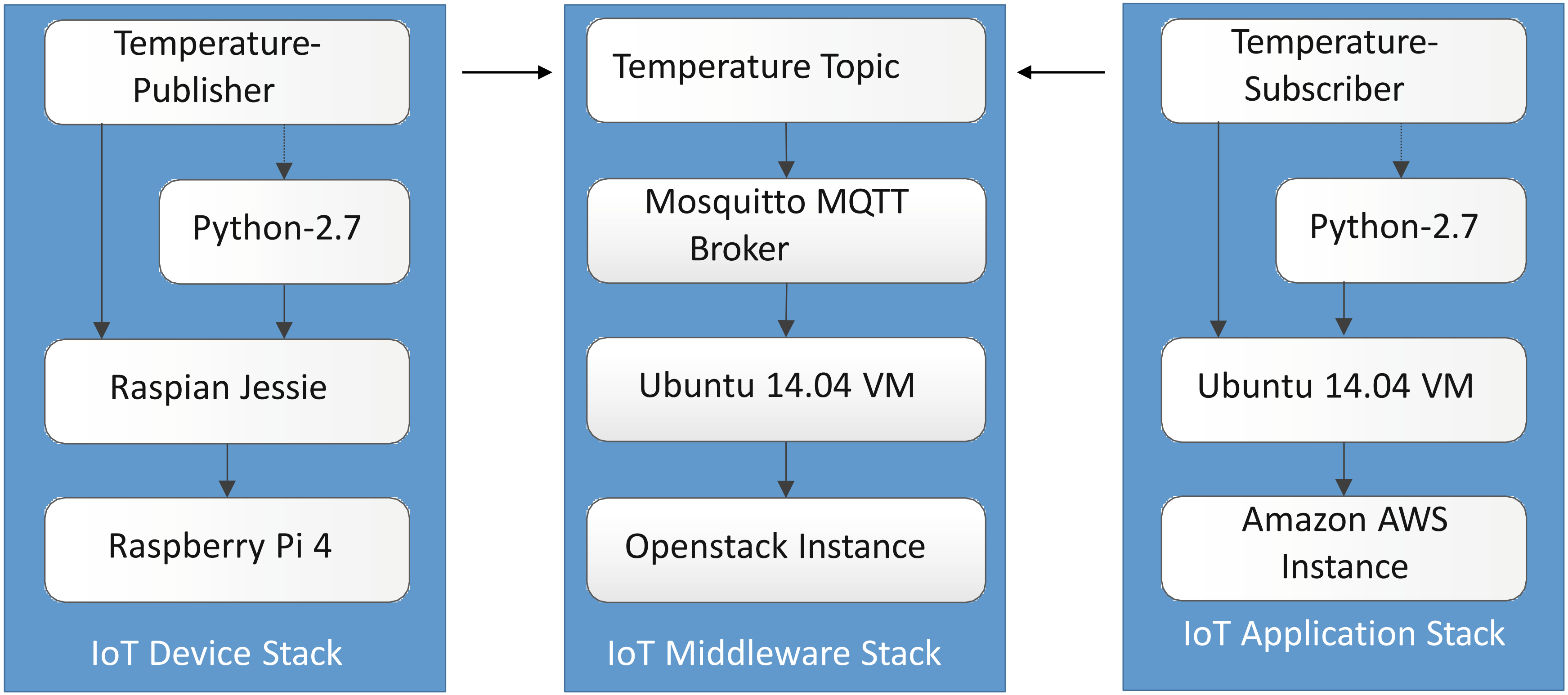
REST API Topic Catalog

```
POST /topics HTTP/1.1
Content-Type: application/json

{ "data_type": "boolean",
  "hardware_type": "occupation detection sensor", ... }

HTTP/1.1 201 CREATED
topic_id: 7321
```

```
GET /topics HTTP/1.1
Accept: application/json

HTTP/1.1 200 OK
[ {"data_type": "boolean", ...},
  {"data_type": "float", ...}, ... ]
```

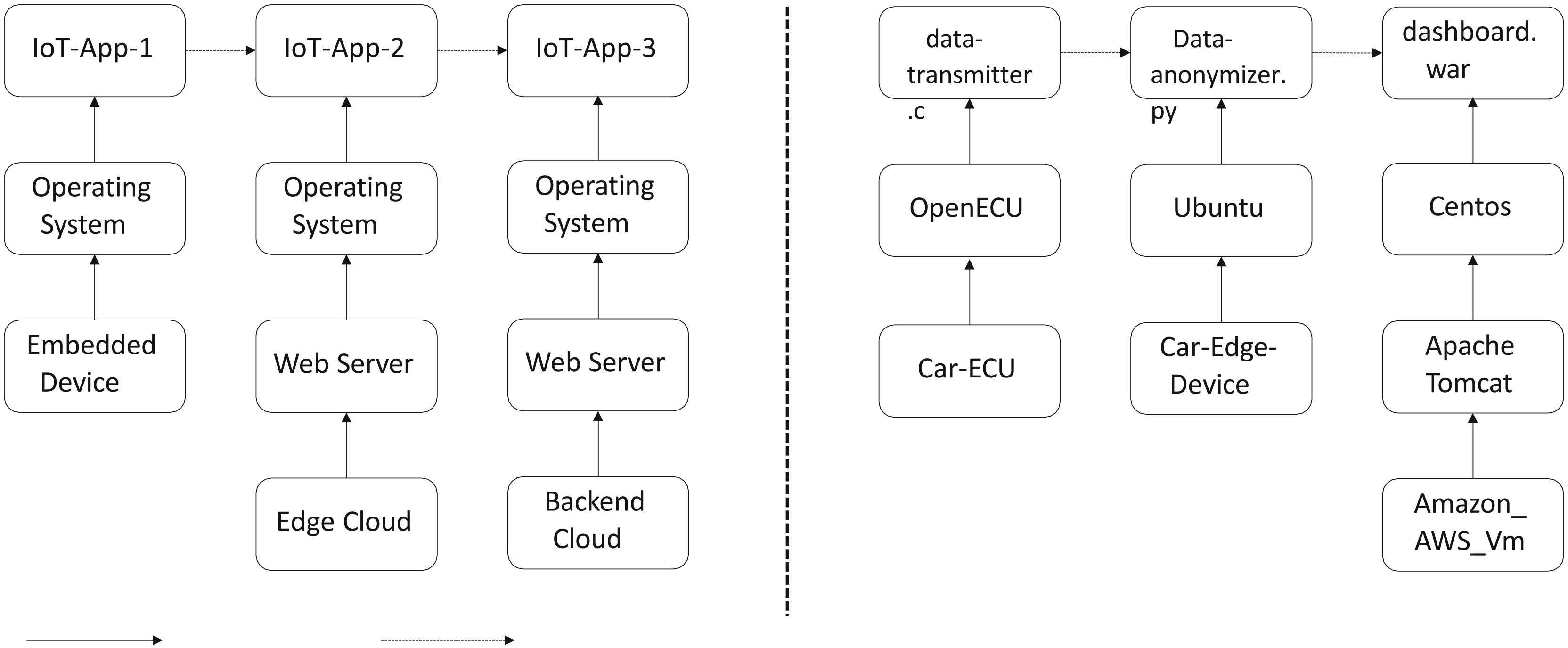
```
POST /topics HTTP/1.1
Accept: application/json Content-Type: application/json

{ "filters": {
    "location": {
      "location_type": "city name",
      "location_value": "Stuttgart"},
    "hardware_type": "occupation detection sensor"}}
```

Step 5: Software Building Blocks



- Pipeline for data flow in an IoT application

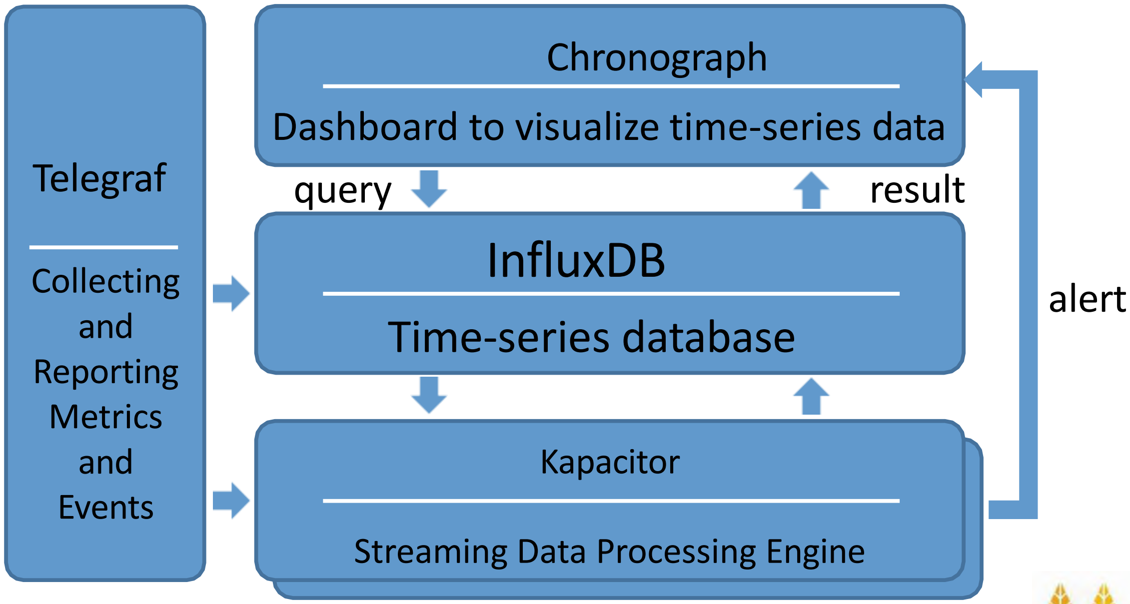
STEP 6: IOT SOFTWARE DEPLOYMENT



Πανεπιστήμιο Κύπρου

- IoT is a highly decentralized and distributed environment
- Standard protocols & uniform message exchange for monitoring is needed
- Centralized monitoring systems are easier to manage and, however, provide a single-point- of-failure
- Decentralized monitoring is more complex to manage but more robust

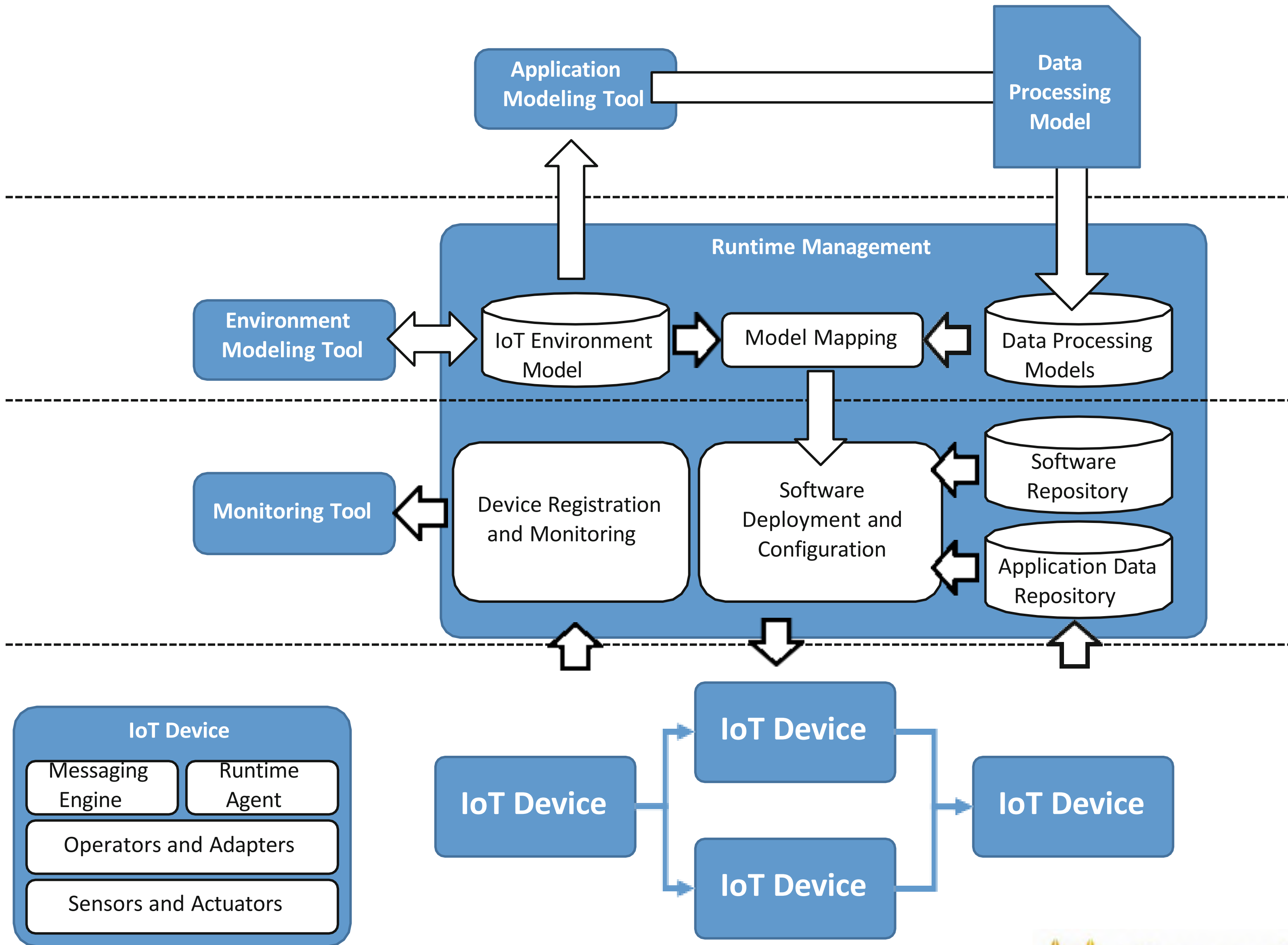
Architecture of the TICK stack



Πανεπιστήμιο Κύπρου

- Software updates and adaptations are necessary
    - to fix bugs and issues
    - to add new functionalities.
- However, software updates tend to be very complex in distributed IoT applications, especially when they should be done online, i.e., while the application is still running.
- Involved models need to adaptable and reused easily, to minimize re-testing and deployment
- In case of failing devices, IoT applications should be able to adapt to the new condition without failing completely (fault tolerant)

LIFECYCLE ARCHITECTURE



Πανεπιστήμιο Κύπρου