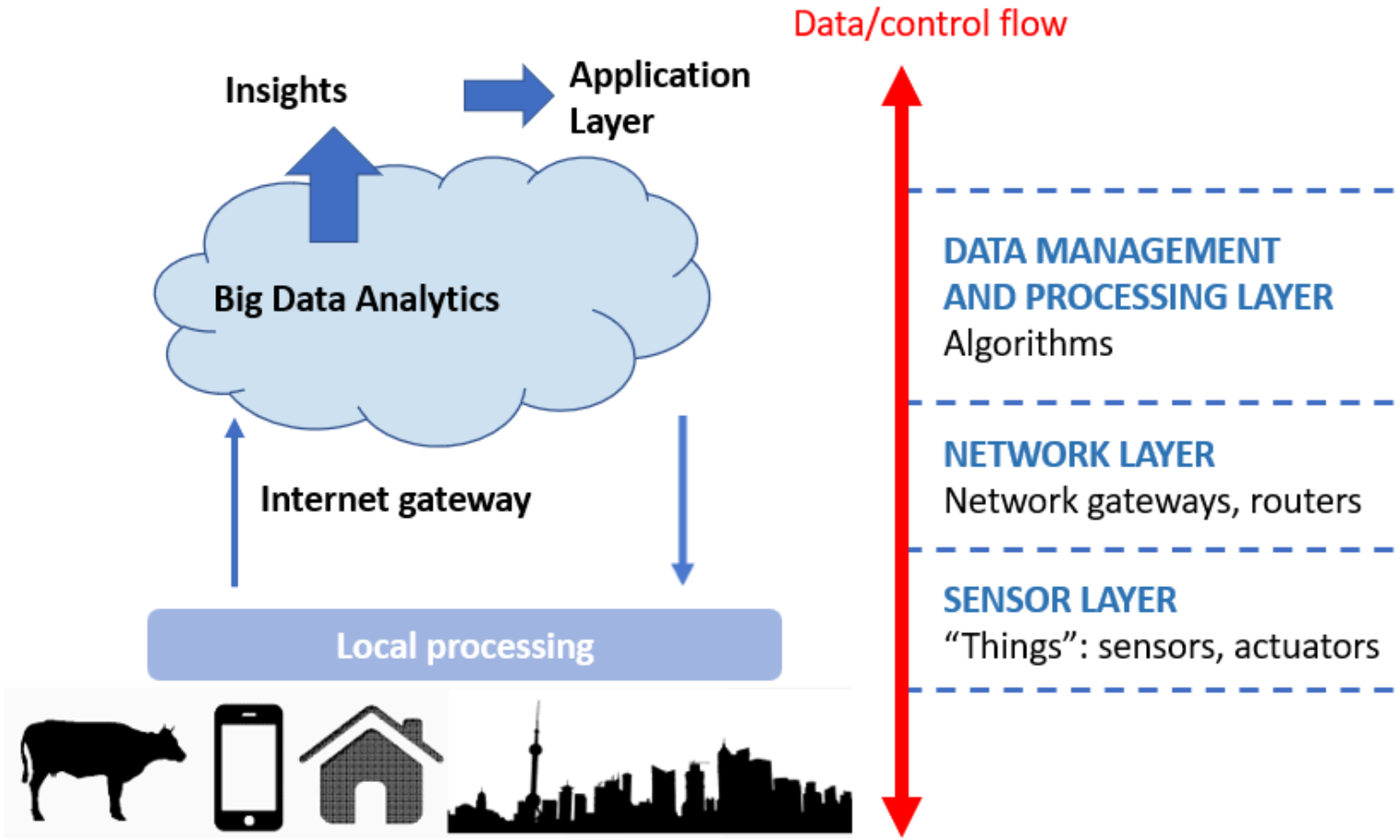
Chapter 1/3

IoT Architecture

What you'll learn in this page

- **Distinguish** the different functional layers in IoT applications.
- **Explain** the general end-to-end structure for IoT applications.
- **Illustrate** different examples of IoT applications.

Internet of Things applications and services have a similar structure despite a high diversity in the devices that are used to implement the applications or services, and the domains for the applications themselves. For the devices, the simplest way to understand this structure is to follow a model that partitions IoT devices along three functional layers: **a sensing layer** that integrates sensors and algorithms for processing sensor data, **a networking layer** which integrates the networking interfaces, protocols, and architectures needed for exchanging data between IoT devices, and **a data management layer** which is responsible for storage, management, and other functions that relate to the data the applications generate and require. During this course we follow this structure and dedicate one chapter for each of the functional layers in turn. The figure below illustrates the layers, the kinds of devices that operate on them, and the role of these devices in the broader IoT ecosystem. We next give an overview of the different functional layers and the general structure of IoT devices.



The functional layers of IoT applications.

Functional Layers of IoT Applications

Sensor Layer (Chapter 3): From the device's point-of-view, the "lowest" layer consists of **sensors** that provide inputs for the IoT application or service. Early IoT only required devices to be connected, identified, and addressed whereas nowadays most smart devices embed sensors that allow the devices to measure information about the people

using the devices, the devices themselves, or the environment where the devices operate. For example, smartphones integrate over 20 sensor modalities such as motion sensors, light sensors, proximity sensors, and multimedia sensors. Besides integrating sensors, most smart devices have processing capability to support processing the sensor data on the devices. This makes it possible to derive information that is relevant for applications that run on the device. From computing architecture point-of-view, sensors are **input** devices as the sensor data they provide forms the input for IoT applications.

Networking Layer (Chapter 4): The networking layer connects devices with each other and the world. As with the sensor layer, the role and characteristics of the networking layer have evolved over the years. The first significant development was the shift to IPv6 which increased the number of available IP addresses (from 2^{32} to 2^{128}). The emergence of mobile computing was another important milestone as it enabled devices to use connectivity while on the move instead of restricting network access to stationary settings. A recent development is the emergence of low-power local connectivity technology, which allows devices to connect to other devices directly without demanding access through the Internet. This is also known as device-to-device (or D2D) networking. Finally, network interfaces themselves have started to become "things". Indeed, modern routers, set top boxes, base stations, and other network devices continually analyse network conditions (including the wireless signal environment) and integrate AI, machine learning, and other processing to optimize their behaviour.

Data Management Layer (Chapter 5): The data management layer is responsible for **storing**, **managing**, and **retrieving** data. In many applications, data management layer also integrates processing support that analyses or otherwise modifies sensor readings

thought the core tasks relate to managing the life-cycle of information. Data management is typically performed at multiple levels. The devices forming the "things", on top of which applications build on, usually need support for managing sensor data internally. At the same time there is a need to manage data from multiple devices - typically in the cloud - or between the device and the cloud, known as edge or fog computing. Indeed, most IoT applications fuse sensor data (inputs) from multiple devices and provide services that analyse and use this data to provide services to end users. As stated earlier, IoT is also increasingly a source of big data. Thus, in a large-scale IoT deployment the requirements for the data management layer align with the requirements of big data processing, requiring dedicated frameworks that have been designed to process such data (such as Spark or Hadoop). In recent times, the boundaries between the layers have started to blur with edge and fog computing starting to take parts of the processing, including data management, closer to the things themselves. In practical terms, parts of the advanced data management can operate on the intermediate devices (such as routers or network base stations) instead of needing dedicated servers or cloud computing access.

Application Layer: The final layer provides the interface for users to interact with the IoT application. Depending on the nature of the application, the information that is provided to the user may come from the device itself, other devices in the vicinity, or through a remote interface (cloud or dedicated server).

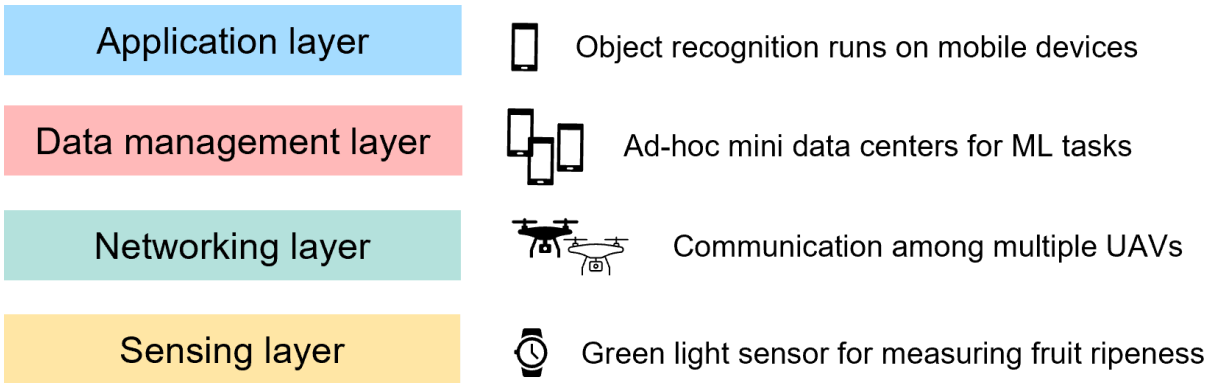
Security and Privacy (Chapter 6): Security and privacy are not actual layers. They are cross-cutting aspects that should be taken into consideration in all software and hardware - not just in IoT. Nevertheless, with IoT increasing the amount and nature of

devices that operate in our environments, naturally there are new types of security and privacy issues that emerge from this development. In terms of security, low-end IoT devices typically have no sophisticated security measures, unlike higher-end devices such as smartphones or laptops. This makes it easier to break the security and access the devices. A wide range of remote attacks against IoT devices have already been conducted and demonstrated in academic contexts. Examples range from remote access to cars - and even remotely stopping their engine while being driven- to accessing wireless pacemakers. Remote access to IoT devices is also the leading mechanism for creating botnets and executing denial of service attacks. Besides remote access, there have been ransomware attacks, e.g., on networked coffee machines and hospital bedside monitoring stations. Thus, security vulnerabilities in IoT context can have dire consequences for people using the devices. As for privacy, IoT devices increasingly integrate sensors. This means that increased data about humans and their environments is being collected. At the same time, these devices have different owners, which makes controlling disclosure of data more challenging. End users also often struggle to understand what all is possible to deduct from the sensor data. Privacy problems can also originate from security vulnerabilities, e.g., unauthorized remote access to a baby monitoring device can allow people to spy on a person's home or children.

Research and Challenges

Research on IoT typically focuses on challenges within the individual layers rather than attempting to advance all components at the same time. For example, research on the sensing layer can investigate new sensing modalities for IoT devices or re-purposing existing sensors for new purposes, whereas networking layer research focuses on new

communication interfaces for IoT applications or improved protocol designs. Note that these topics are broader than IoT and it is important to **contextualize** the research with respect to IoT challenges. For example, measurement technology is developed within the field of metrology (measurement science) and distinct research fields have their own strands of measurement technology research (e.g., development of air quality sensing technology within atmospheric sciences). Only once this is placed in an IoT context does the research become related to IoT. Examples of contextualizing research within IoT range from optimizing the integration of sensors, algorithms, or other components to function within a specific device or IoT domain, to adapting the data management layer to support data matching the characteristics of a specific IoT deployment. Further examples of how to contextualize research in a specific functional layer within the IoT domain are illustrated in the figure below.



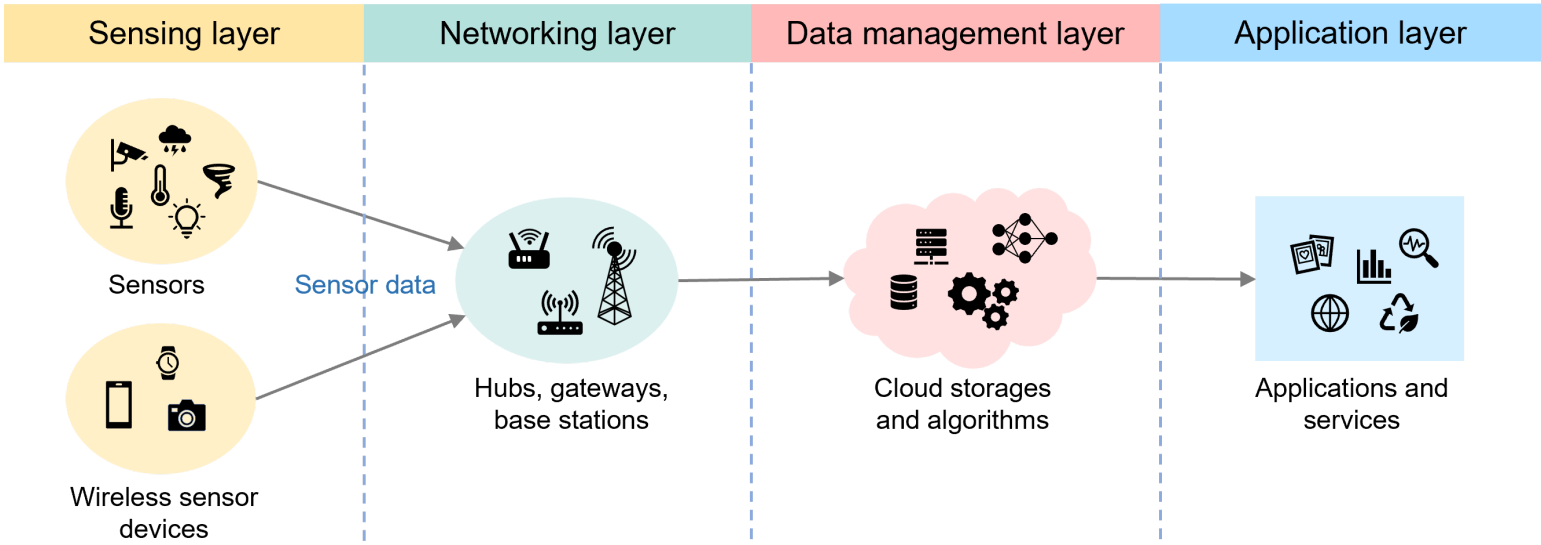
Examples of research along different layers.

Beyond challenges related to research, there are practical challenges that span the entire IoT domain and that need to be considered when building IoT applications. One such challenge is the lack of **standards** as IoT technologies remain fragmented. While there are standardization bodies that relate to IoT, none provides an integrated framework for IoT deployments, instead focusing on specific technologies or domains (especially prominent in the field of wireless communications). What makes standardization challenging is that the field is evolving rapidly, and standardization processes typically take years (often 5-10). Thus, standards often become relevant only once the ecosystem consolidates around a common set of technologies and practices. Another related and critical challenge is **interoperability** as IoT ecosystems integrate a variety of devices from different manufacturers and that use different technologies. Research projects often are limited by the difficulty of having IoT devices interact with each other and thus a stronger focus on interoperability between devices and the data they produce would be an essential driver for the field. Finally, **legal frameworks** are needed to govern the IoT ecosystem, particularly related to security and privacy.

End-to-End IoT Architecture

The layered model provides a reference point for understanding different functionality in IoT solutions, but it provides no insights into how to design IoT applications. The most common way to implement IoT applications is to rely on the end-to-end architecture model shown in the figure below. In this model, sensor enabled IoT devices connect to a **hub** or **gateway** which then transfers the information to a server or a cloud that processes the sensor measurements. Users interact with the IoT devices and the

information they gather through applications and services that are implemented on top of the cloud service.



An overview of end-to-end IoT architecture.

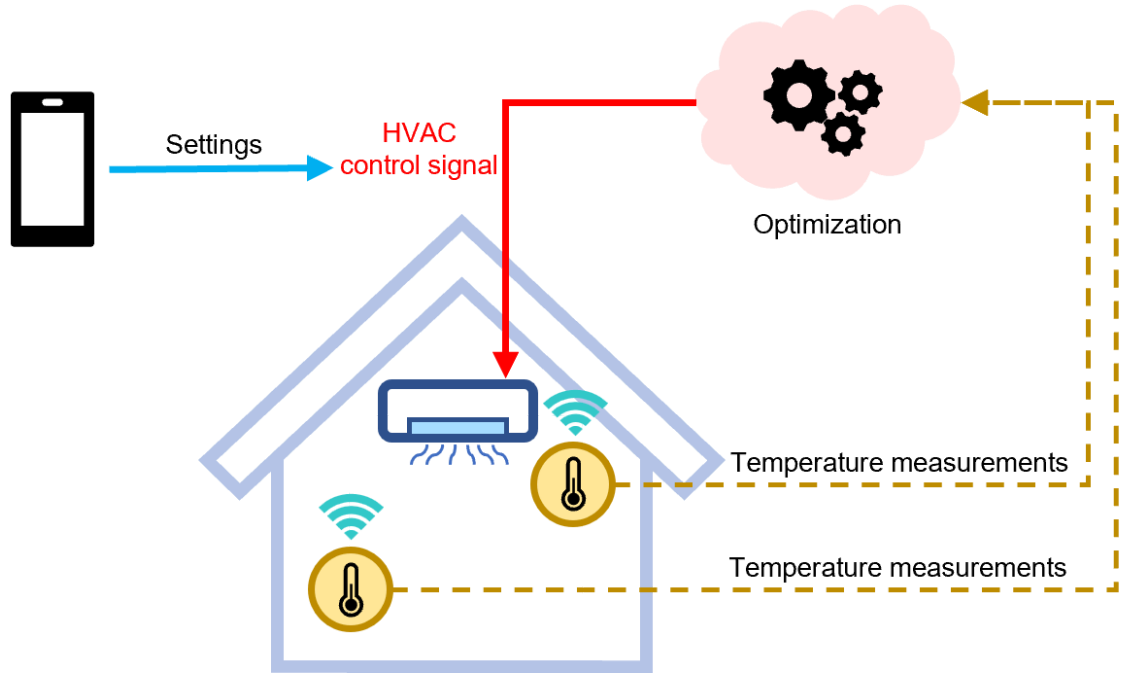
The end-to-end model should be understood as a service chain that links IoT devices with different capabilities with end-users. The sensor enabled IoT devices are assumed to have the least resources and to operate under the heaviest constraints. For example, low-end sensor nodes typically have heavy constraints on processing, memory, energy, and other resources, whereas the hubs and gateways typically have fewer resource constraints but need sufficiently fast and efficient network performance. Similarly, cloud and other data management needs to process data efficiently and flexibly regardless of the devices that produce the data. In terms of functional layers, each device similarly integrates each of the layers but has different requirements for implementing the

necessary functionality. For example, the things need networking to communicate with the hubs or gateways, and they also need data management to store, process and manage the measurements internally. Similarly, the gateway typically combines software and hardware sensors to optimize its performance. Thus, the end-to-end architecture model offers a viewpoint for analysing larger deployments of IoT devices, whereas the functional layers offer a view of the functions that are needed on each of the devices.

Application Examples

To put the concepts into practice, we next consider two examples of IoT applications with different scales and analyse the different devices and functional layers of the devices involved in these applications.

Smart HVAC (Heating, Ventilating and Air Conditioning): The figure below shows a high-level view of an intelligent (or smart) HVAC system that uses wireless sensor nodes to capture environmental parameters. These are communicated to a server (effectively serving as the hub or the gateway) which then generates policies governing the operation of heating, ventilation, or air conditioning as requested. The server also interacts with the cloud, e.g., to collect energy usage statistics from the energy provider. The policies, as well as statistics about system use and resource consumption (e.g., energy use, and operational time) can be investigated on a smartphone interface that interacts with the server and a cloud. The smartphone app also allows adjusting the policies or turn of any HVAC functionality.

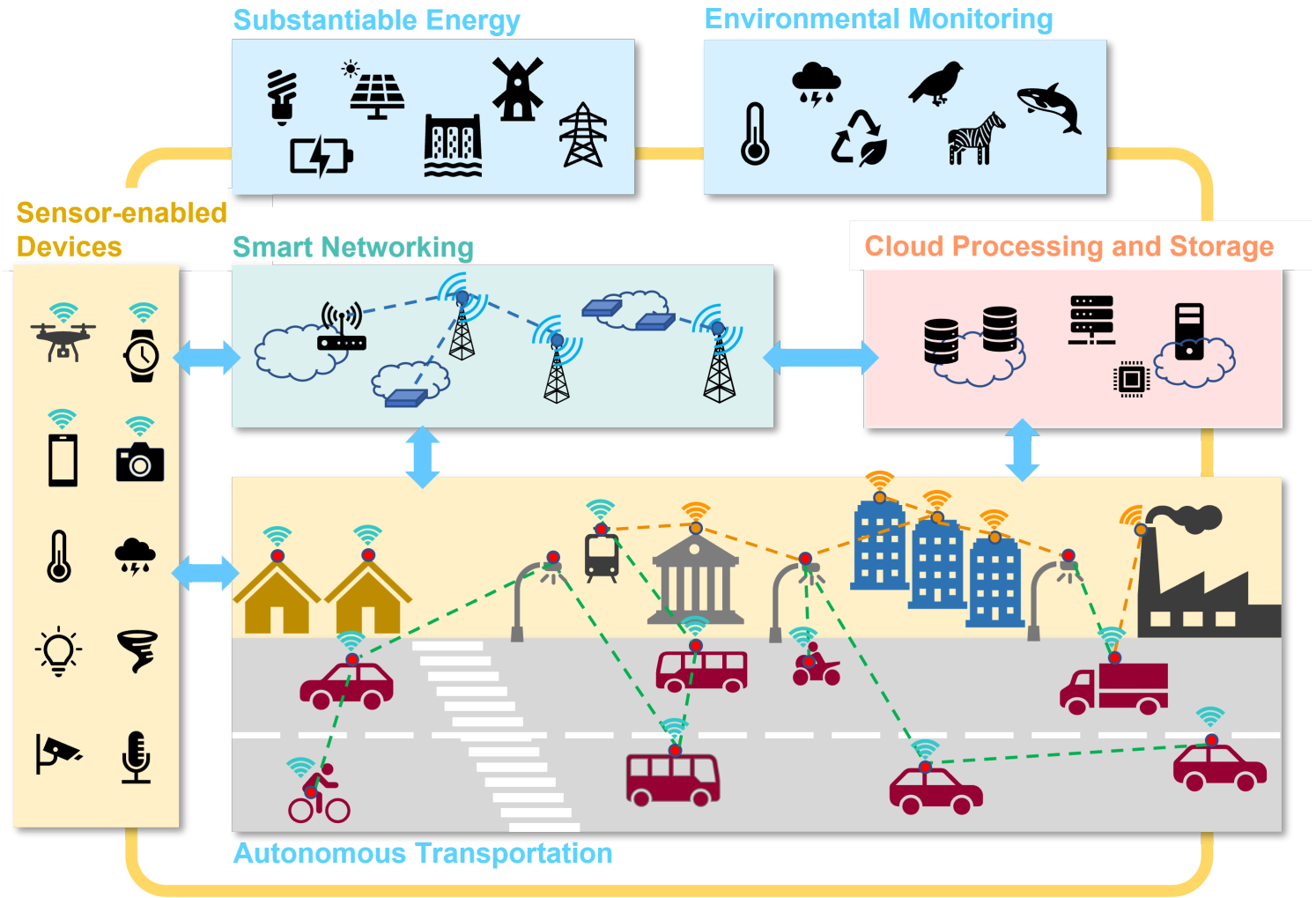


Prototype of a smart HVAC system for dynamically adapting temperature to occupants.

In the example above, the networked (wireless) sensors effectively are the only devices that have a dedicated sensing layer. The sensor nodes should also partly implement the data management layer as they at least should offer access control functionality (i.e., prevent unauthorized access) even if they would not store the measurements. The server acts as the hub of gateway and stores any configuration parameters or policies that are needed to control the overall application. The actuators responsible for the HVAC operations have the most restricted functionality as they only offer an interface for receiving command parameters from the server and integrate limited data management support for translating the commands into actions (e.g., to release hot or cold air into the space) and for providing security support (such as access control).

Finally, the application layer is implemented as a mobile or web application that interacts with the server and allows remote control of the HVAC parameters.

Smart Cities: The figure below highlights the general idea of smart cities with a focus on using IoT to improve the sustainability of the city. The vision is that future cities would deploy massive amounts of sensors which would then be used to improve operations in the city.



A concept of smart city with sensor-based parking, waste management and environment monitoring.

Smart Santander is an early proof-of-concept project and implementation of the smart city concept. The project integrated over 10,000 sensors in the city of Santander, Spain to support different smart applications. Examples of applications that were provided include:

- Smart parking, checking availability of parking spots in the city and offering automated payment.
- Environmental monitoring that provides feedback on conditions within the city. Examples include air quality, litter recognition, water monitoring, and light and noise pollution monitoring.
- Traffic monitoring, taxis, buses, and police cars transmit location measurements. This information is then used to estimate congestion, mobility, and city functionality.
- Intelligent spaces, such as parks, use sensors to improve safety and comfort. For example, using motion sensors to detect when to use lights in parks.
- Smart waste management, using sensors on waste bins to determine optimal waste pickup schedules. Sensors could potentially be used to recognize waste types and support recycling practices.

In this example, the most common type of device is a low-end sensor node which generates sensor data about the city environment. The sensor nodes should also offer networking and data management. Specifically, similarly to the HVAC example, the sensor nodes should be able to store and process measurements, and they should integrate limited security features to prevent unauthorized access. The project initially ran during a time when data centres were not commonly available and thus it relied on dedicated servers to offer the hub and data management functionality. In modern implementation of smart cities, the sensor data would be expected to be sent to a cloud data centre which is responsible for storage and processing. Finally, the applications themselves operated on top of the server hub, whereas in modern implementations they would rely on distributed applications operating on top of a cloud.

The two examples described above also highlight how the two generic architectural models are sufficient for diverse IoT applications and how they have even survived the test of time. Indeed, both application examples relied on a server for managing the data and providing the application interface, whereas modern applications have separated the role of the server hub and data processing centre to improve scalability and to be able to serve a larger number of users.

Summary

- The functions on an IoT device can be understood through a layered functional model that comprises of a **sensor** layer (input), a **networking** layer (connectivity), and a **data management** layer (data processing) which processes data from low-end devices through multiple devices into something that can be used to deliver applications and services to users.
- IoT research mostly focuses on individual layers, but the important thing is to contextualize the research within IoT: how IoT devices or ecosystems can harness the research?
- IoT applications can be seen as service chains that can be implemented following an end-to-end architecture model where low-end devices with sensors generate the data that other components use, hubs or gateways serve as the connection point between backend and sensor nodes, and cloud or another backend is responsible for aggregating data and providing the final application interface to end users.

Exercises

Exercise:

1.3. Concepts

TRIES 2 POINTS 0/3

Instructions

Consider the concepts listed below. Please choose all options that match.

Which layer(s) should integrate support for security and privacy?

- Communications Layer
- Sensor Layer
- Data Management Layer

ⓘ Select all correct options.

In IoT architecture, a hub refers to a component that

- low-end devices connect to.
- is synonymous with a cloud service provider.
- disseminates data.

ⓘ Select all correct options.

Which of the following options are correct? Pick all that apply.

- The IoT ecosystem is heavily standardized.

The IoT device ecosystem is unified and comprises of devices that can easily operate with each other.

Interroperability is one of the key challenges for IoT deployments.

IoT use is governed by many legal frameworks.

ⓘ Select all correct options.

Submit

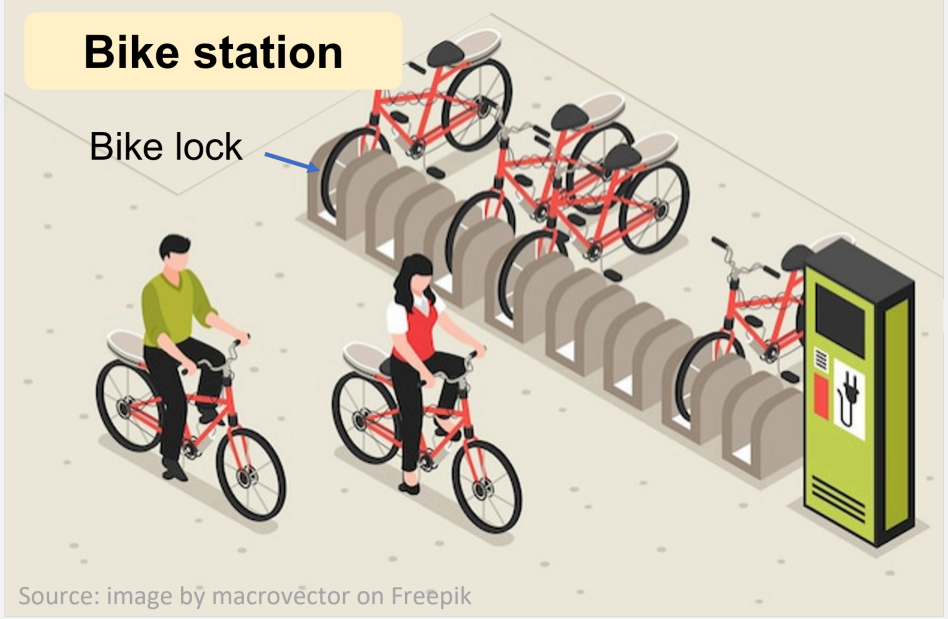
Exercise:

1.3. Identifying IoT architecture layers

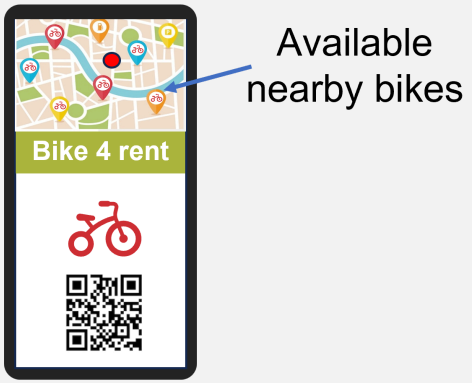
TRIES 2 POINTS 0/3

Instructions

Consider a bike rental service shown in the figure below where users can rent and return bikes using a smartphone app. The bikes have a RFID chip and the station operates a RFID reader to detect bikes that are returned or removed from the bike station. Please pick the layers that are relevant for the given service functionality.



Source: image by macrovector on Freepik



Mobile app for bike rent

Validating the user that is renting a bike.

- Networking Layer
- Privacy and Security
- Sensor Layer
- Data Management Layer

ⓘ Select all correct options.

The station detecting that a bike has been returned.

- Networking Layer
- Privacy and Security
- Sensor Layer
- Data Management Layer

ⓘ Select all correct options.

Find available bikes that are within a city block from the user's current location.

- Networking Layer
- Privacy and Security
- Sensor layer
- Data management layer

Select all correct options.

Submit

Credit: Image by macrovector on Freepik

↑ Chapter front page

You've reached the end of this topic.

Proceed to the next topic

Next page:
Smart Objects

About

Privacy

The University of Helsinki MOOC Center makes high-quality online education possible by developing and researching educational software and online learning materials. Teachers both within and without the University of Helsinki rely on our tools to create impactful teaching materials. Our popular Massive Open Online Courses (MOOCs) have been available through MOOC.fi since 2012.

This website is powered by an open source software developed by the University of Helsinki MOOC Center. Star the project on GitHub: Project Github.

MOOC.fi

HELSINGIN YLIOPISTO
HELSINGFORS UNIVERSITET
UNIVERSITY OF HELSINKI