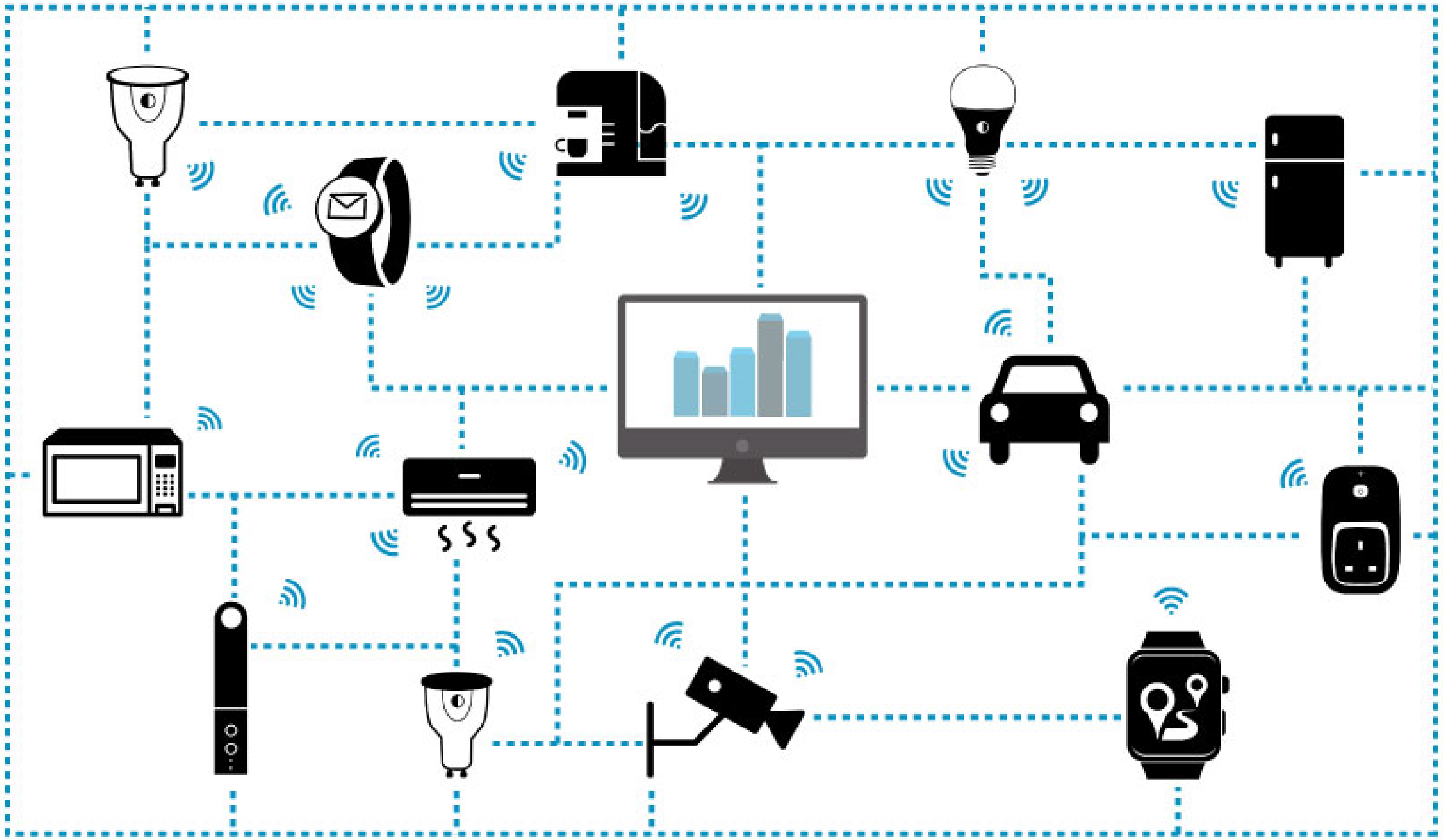
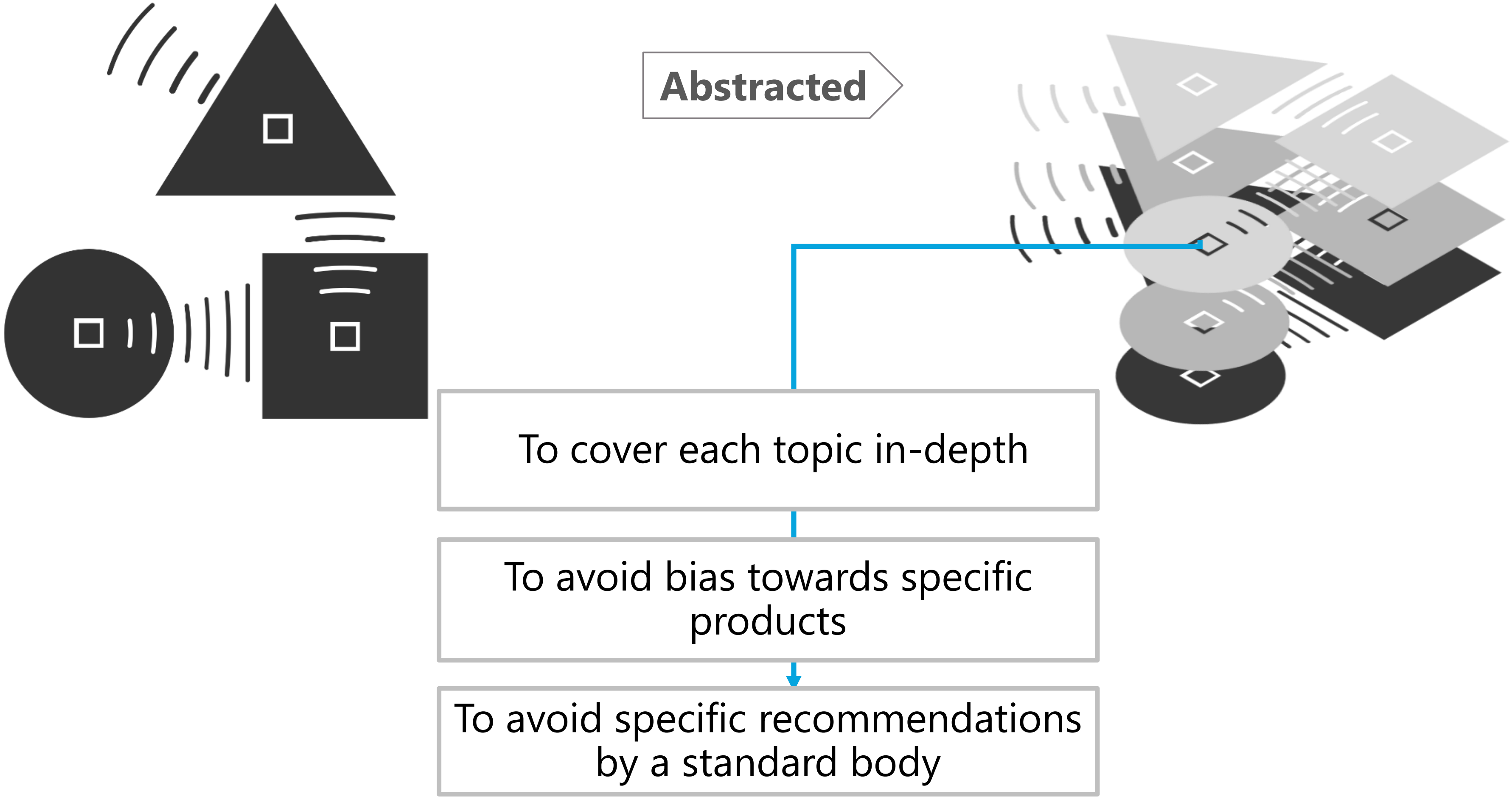
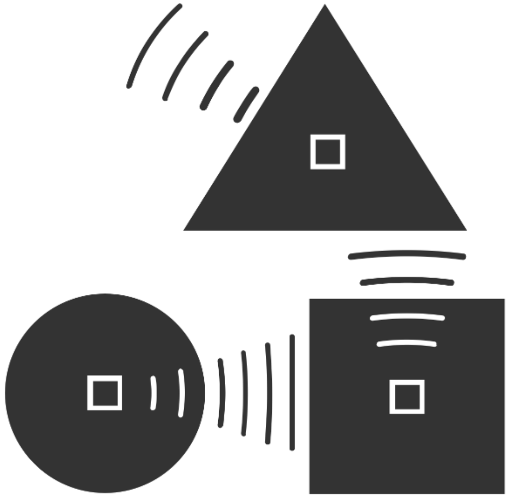


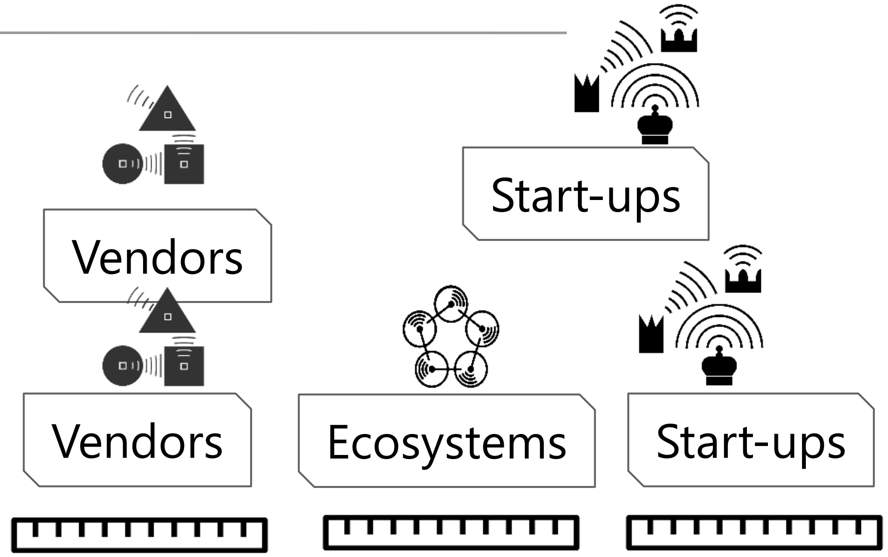
IOT Architecture

Overarching IOT Reference Architecture



Overarching IOT Reference Architecture





IT Value Chain

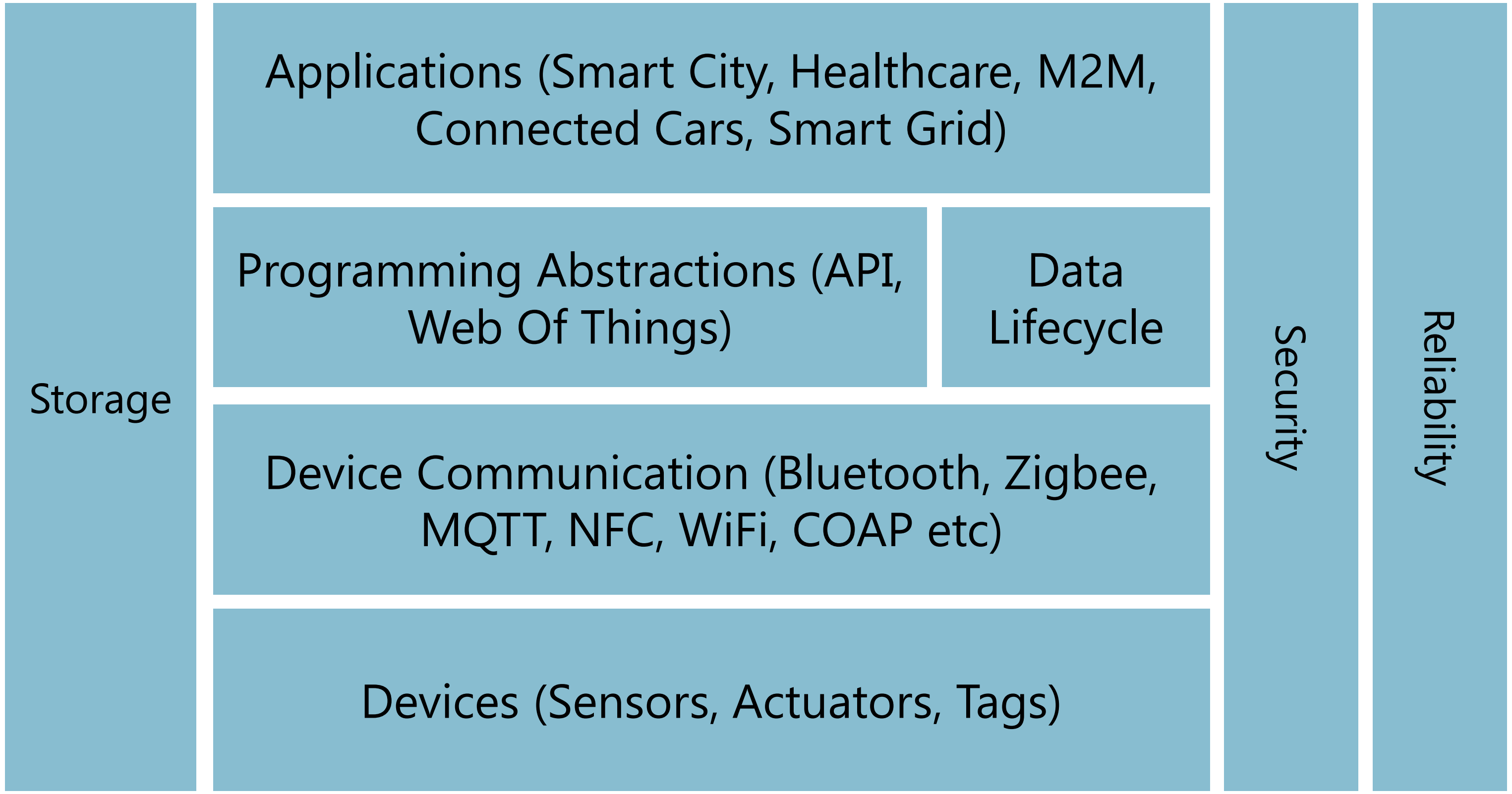
Consumer apps

Business apps

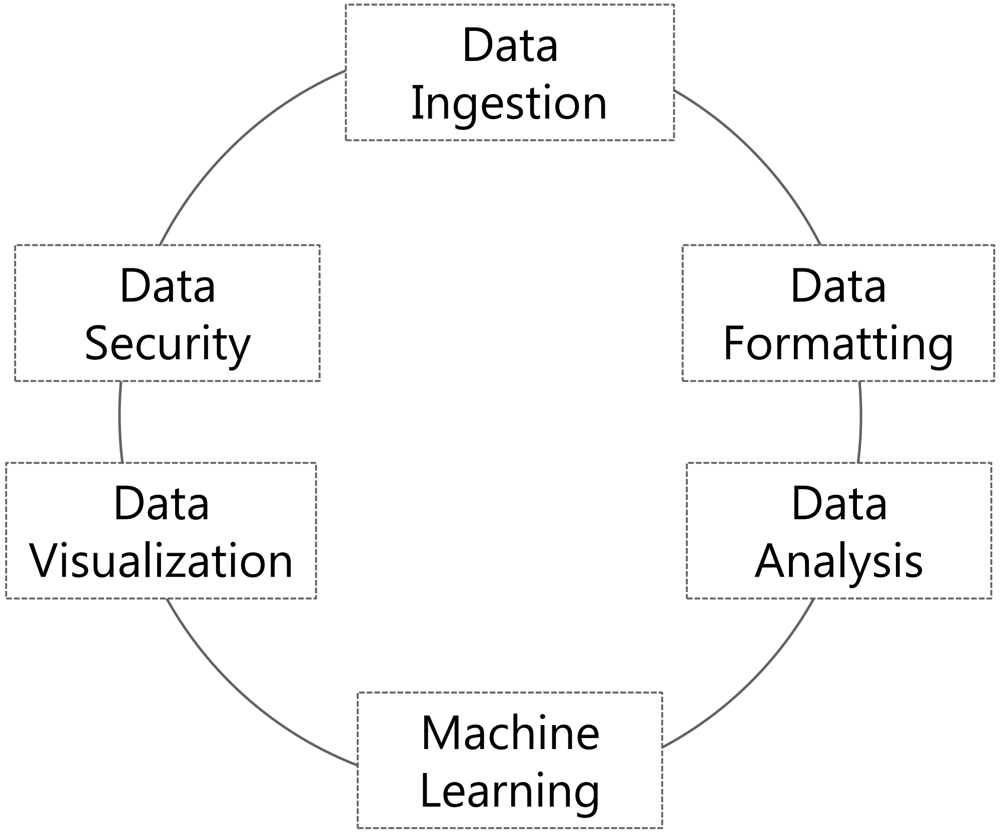
Embedded systems

Standardizing architectural elements, helps regulate the proliferation of architectures

The Different Architectural Layers

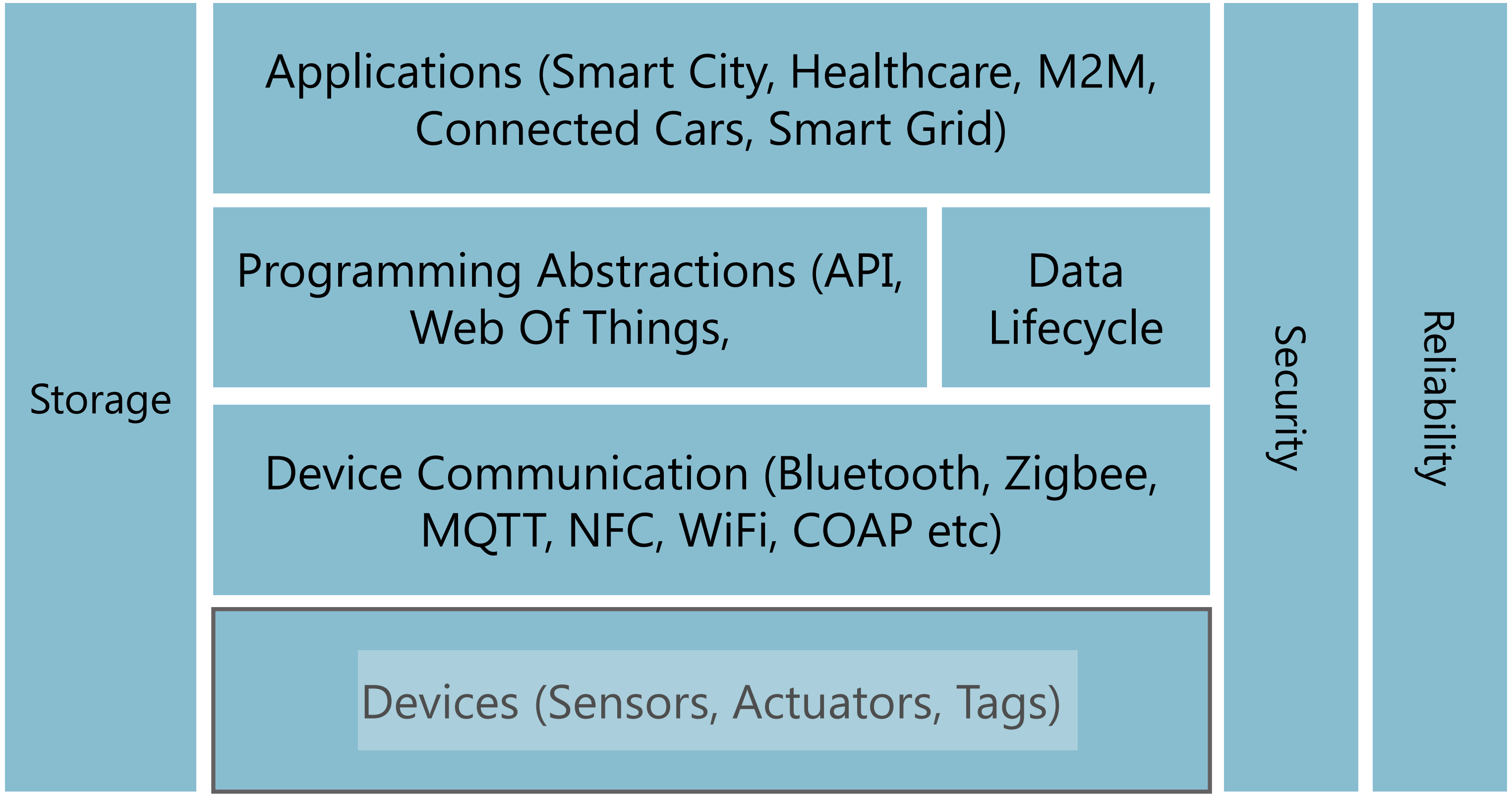


Data Analytics Lifecycle

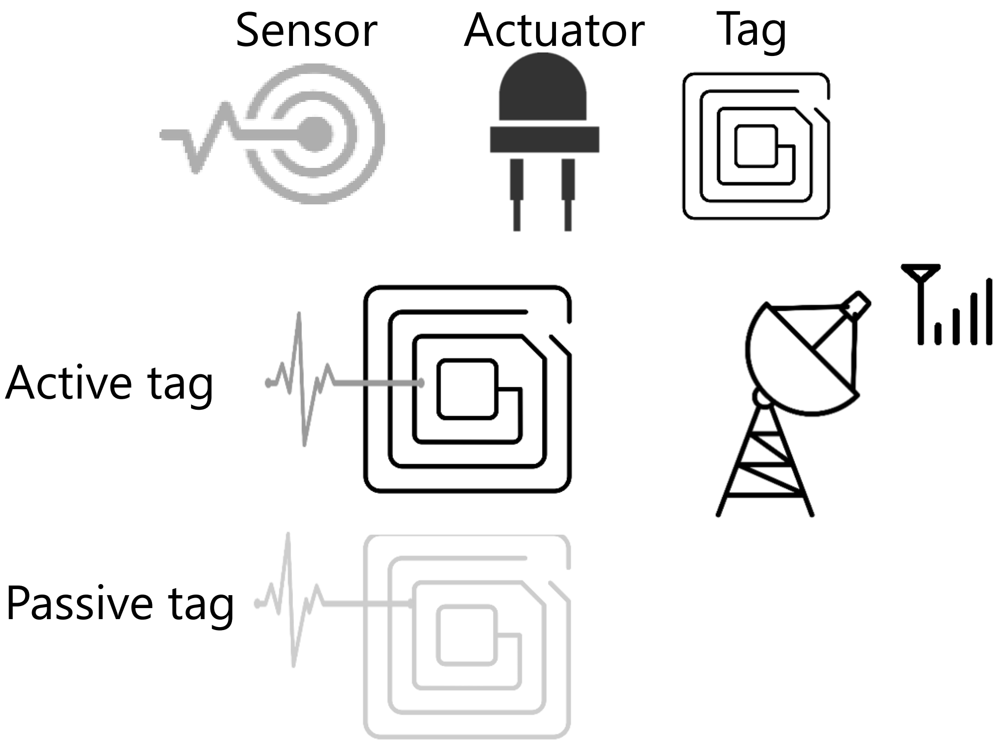


Next: The different architectural layers making up the barebones, minimum common considerations for any IOT deployment

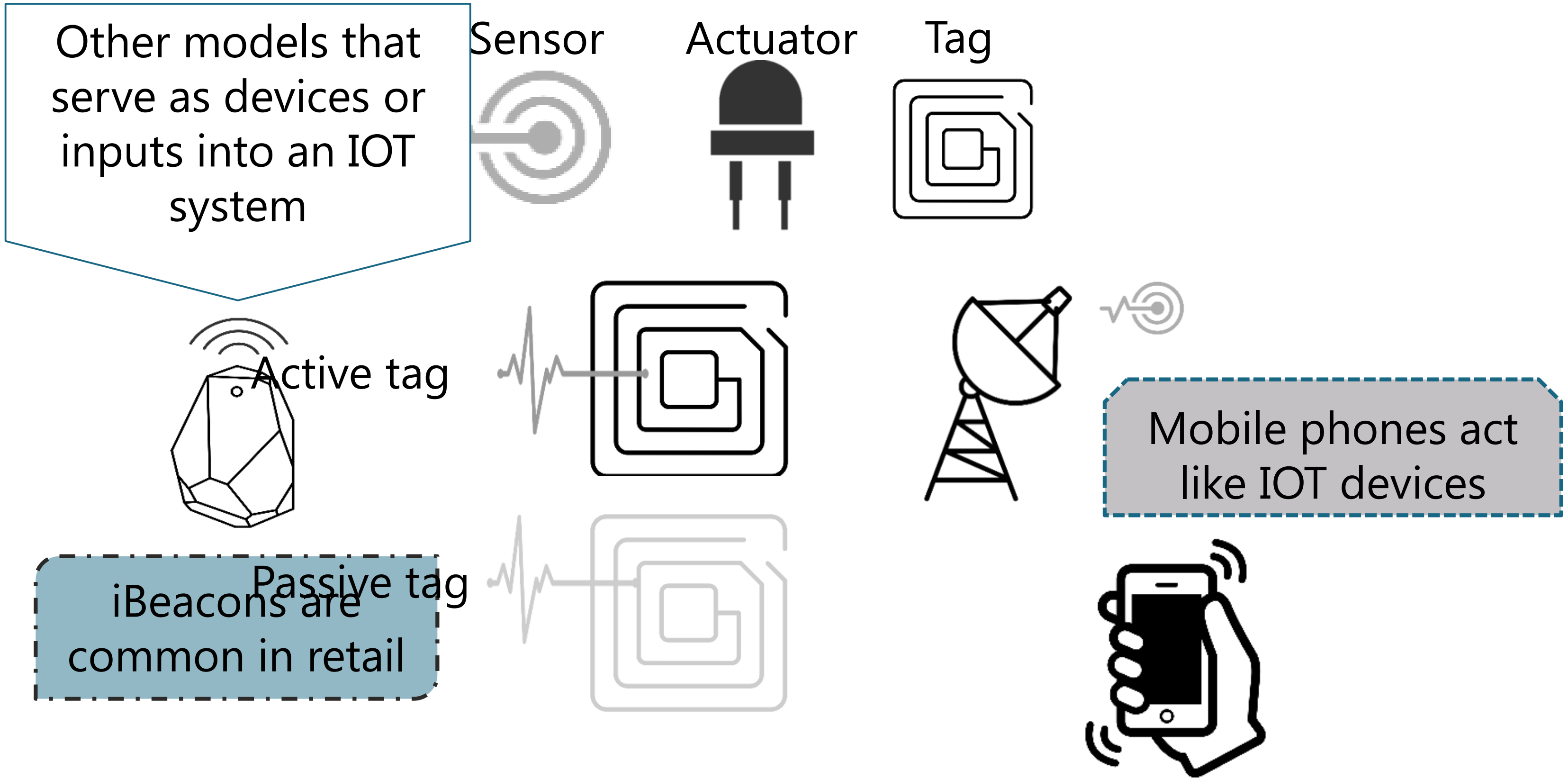
Basic IOT Elements: Devices



Basic IOT Elements: Devices



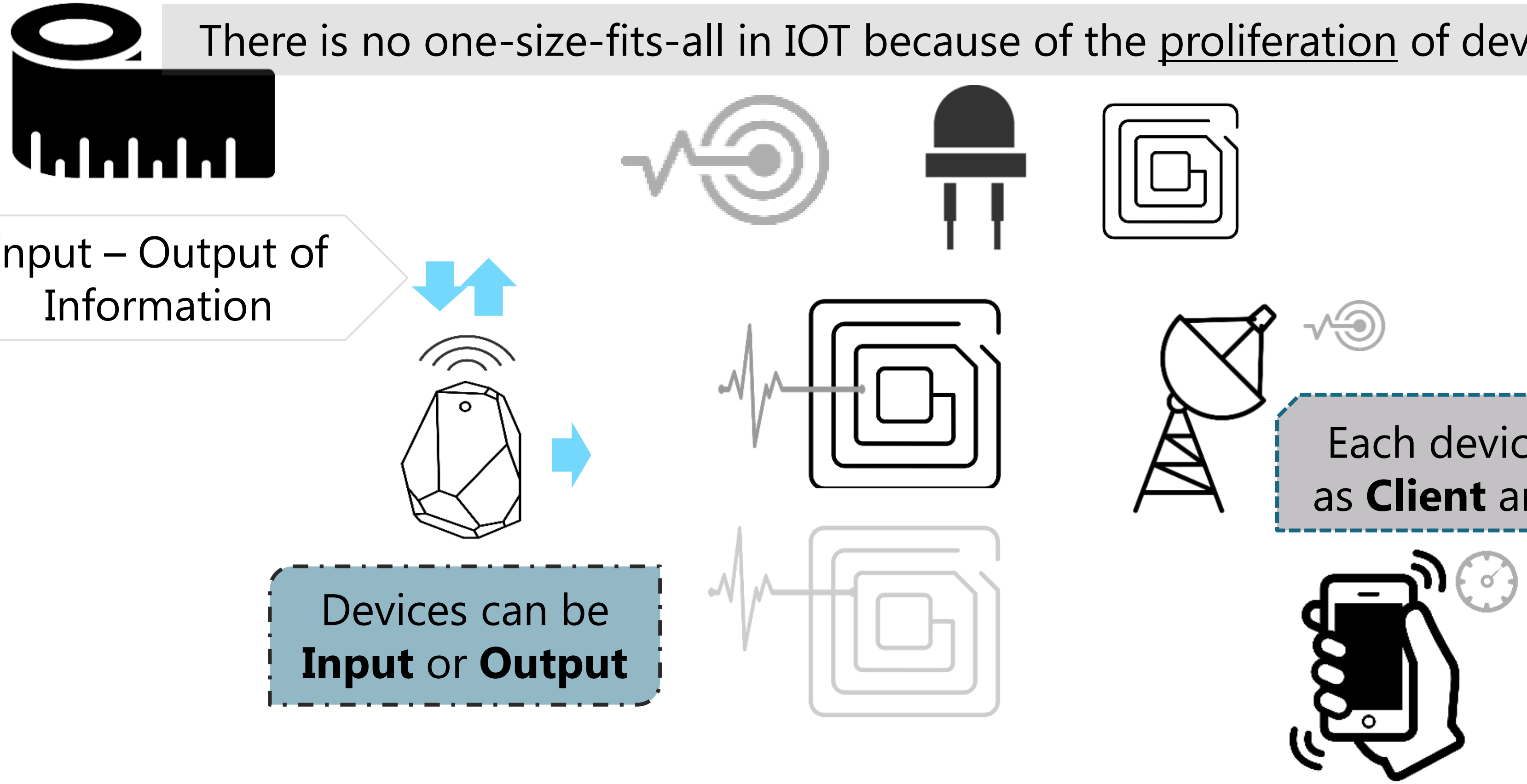
Basic IOT Elements: Devices



Basic IOT Elements: Devices

There is no one-size-fits-all in IOT because of the proliferation of devices

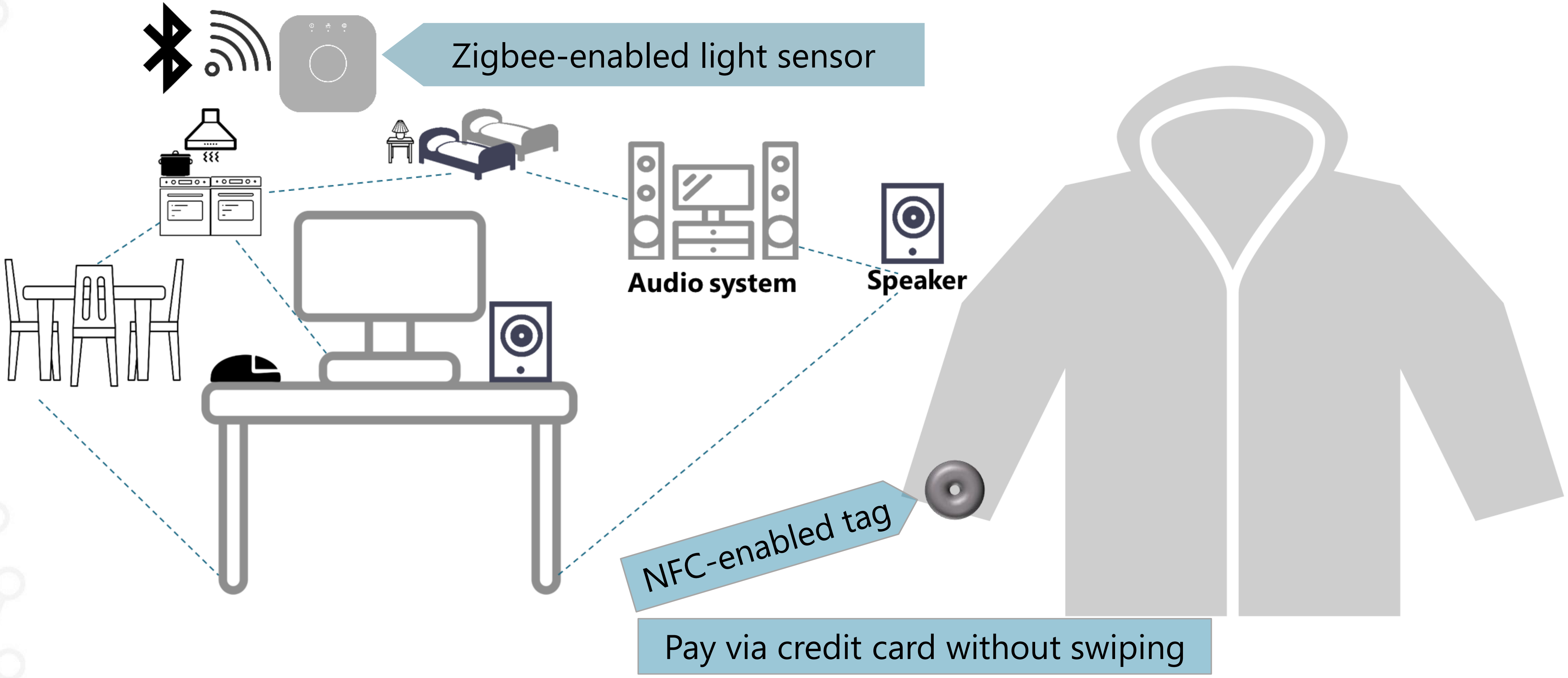
Input – Output of Information



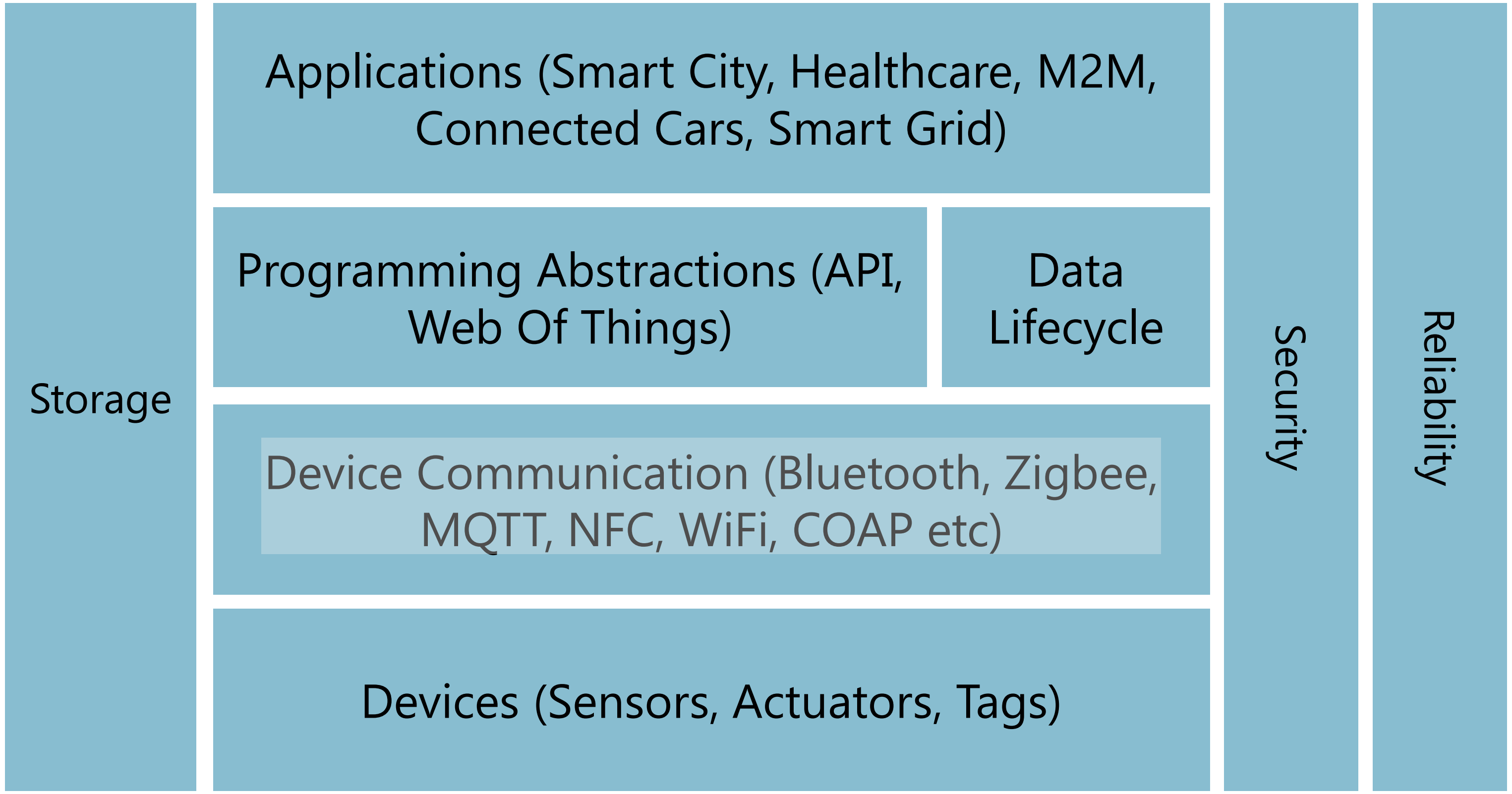
Devices can be **Input** or **Output**

Each device can act as **Client** and **Server**

Device Communication

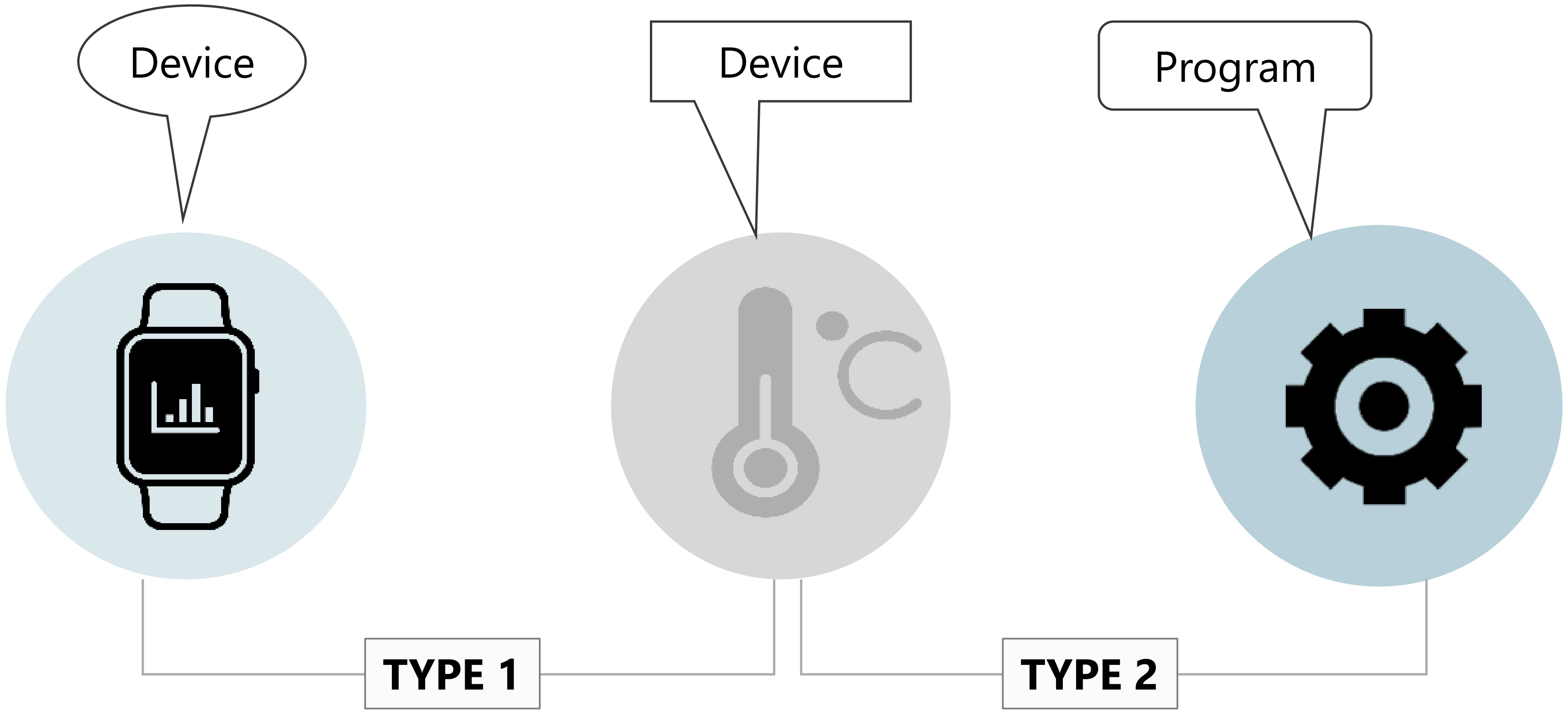


Device Communication



Communication in IOT

Two Types of Communication



IOT Communication

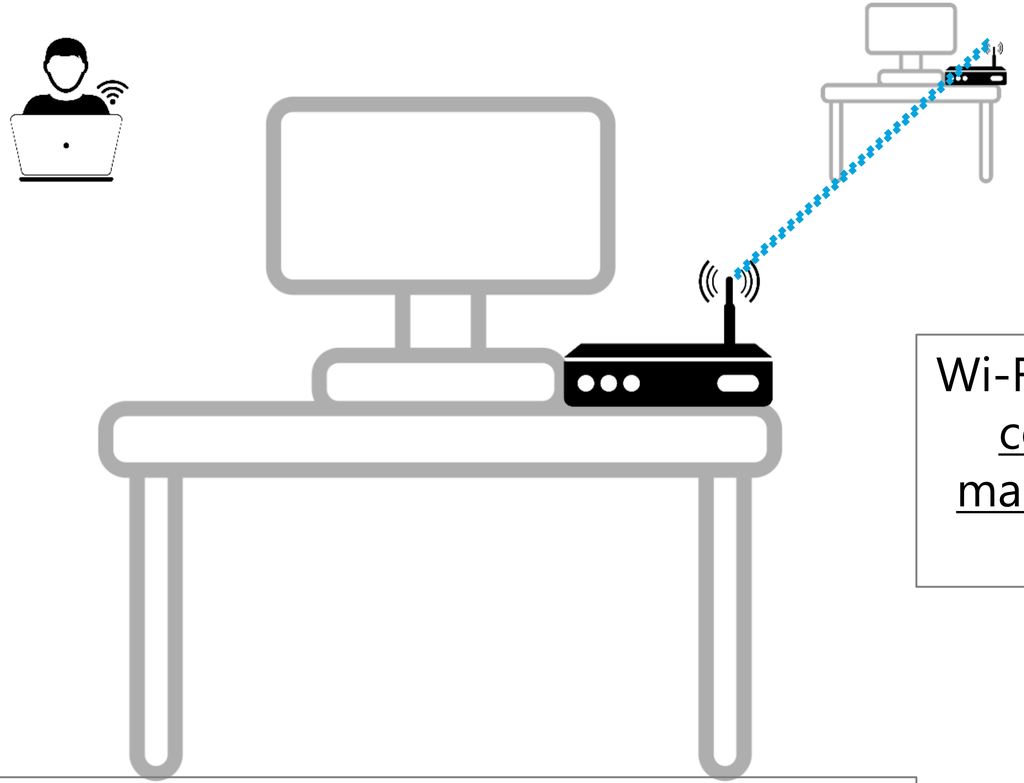
Communication **TYPE 1**

TCP IP (PROTOCOL STACK)
Lower Level Communication
Wi-Fi
Zigbee
Cellular (2g, 3g, LTE)
Bluetooth
Higher Level Communication
MQTT
COAP
XMPP
HTTP (REST)

Typical communication protocols

Lower Level Communication Protocols

Wi-Fi



Examples
Laptops, mobile phones or routers on other end

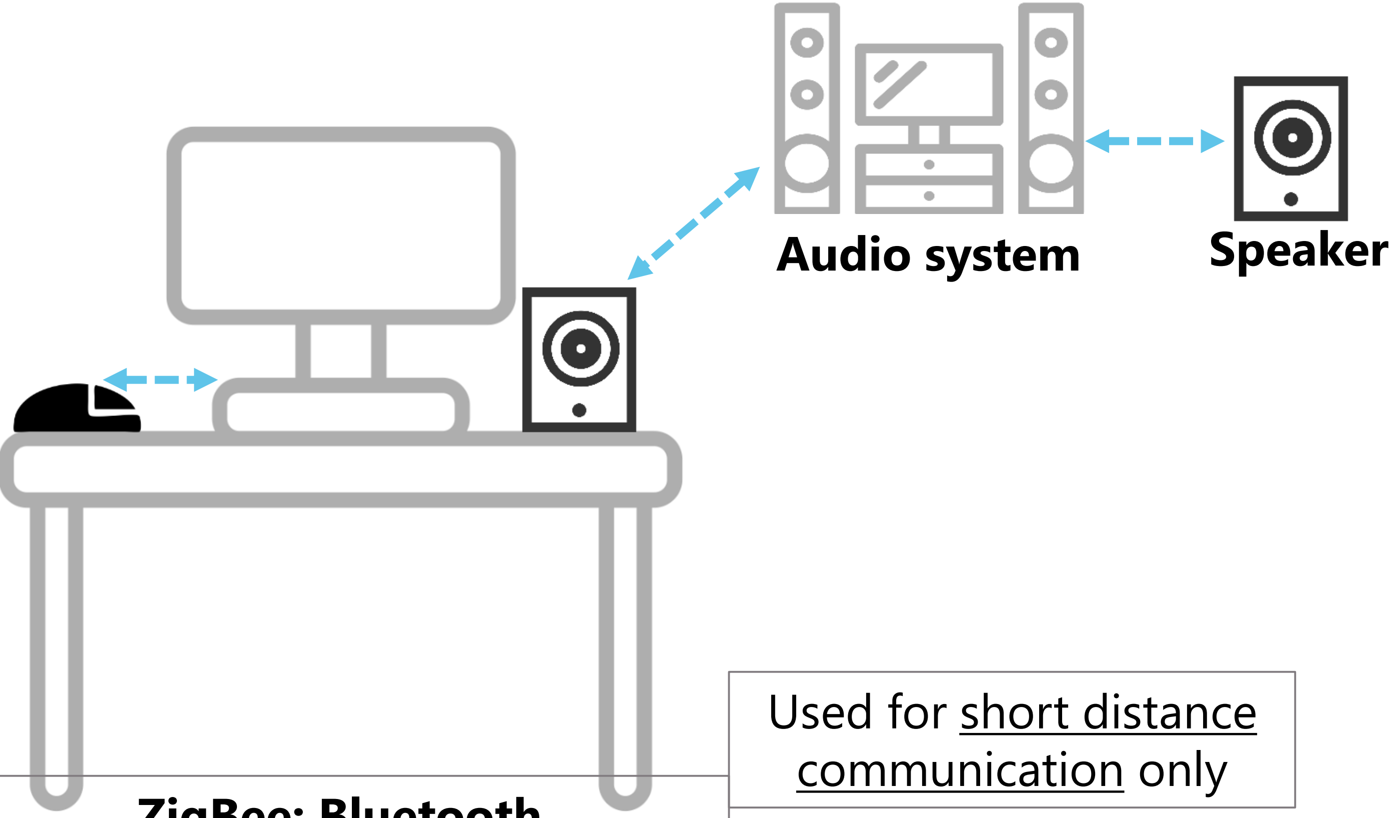
Wi-Fi is commonly used to <u>connect machines to machines</u> like computers to routers

It can <u>transfer large amounts of data</u> and <u>requires powerful devices</u> to be able do so

Lower Level Communication Protocols

ZigBee

Bluetooth



Used for <u>short distance communication</u> only

ZigBee; Bluetooth

- Low energy
- Low consumption protocols

Higher Level Communication Protocols

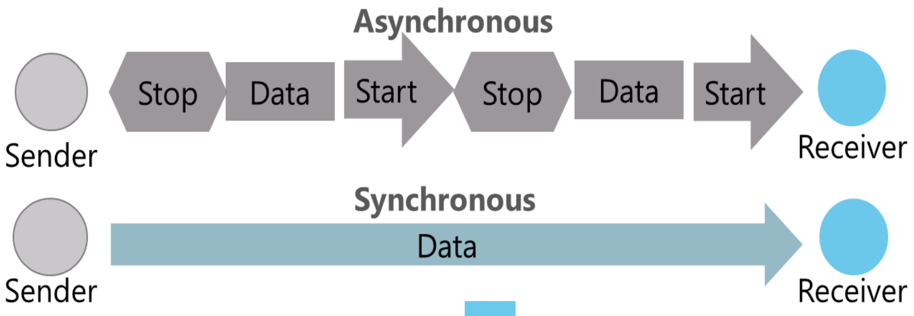
Communication **TYPE 2**

TCP IP (PROTOCOL STACK)
Lower Level Communication
Wi-Fi
Zigbee
Cellular (2g, 3g, LTE)
Bluetooth
Higher Level Communication
MQTT
COAP
XMPP
HTTP (REST)

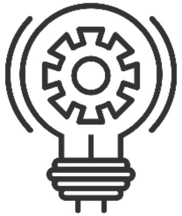
A new set of standards, that address the **specific nuances** of IOT devices has emerged

Higher Level Communication Protocols

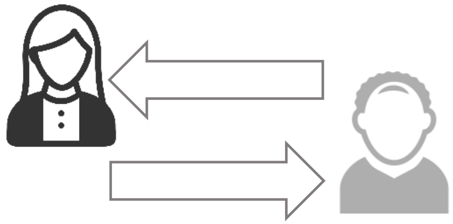
MQTT



It operates on **'Pub Sub'** or **Public Subscribe**



It relies on the notion of **asynchronous communication** based on the MQ series message bus protocol



MQTT is a protocol invented by IBM and donated to the **Open Standards Organization** (OSS)

People can send and subscribe to messages asynchronously, without the need for a **continuous synchronous connection**

Higher Level Communication Protocols

COAP

.NEW

COAP is a relatively new phenomenon

Low energy devices

Low bandwidth

COAP is replacing **HTTP** as a means of communicating between IOT devices

COAP is a light weight protocol which is replacing **HTTP**

Unreliable network communication

HTTP is considered a heavy load

Higher Level Communication Protocols

Communication **TYPE 2**

TCP IP (PROTOCOL STACK)
Lower Level Communication
Wi-Fi
Zigbee
Cellular (2g, 3g, LTE)
Bluetooth
Higher Level Communication
MQTT
COAP
XMPP
HTTP (REST)

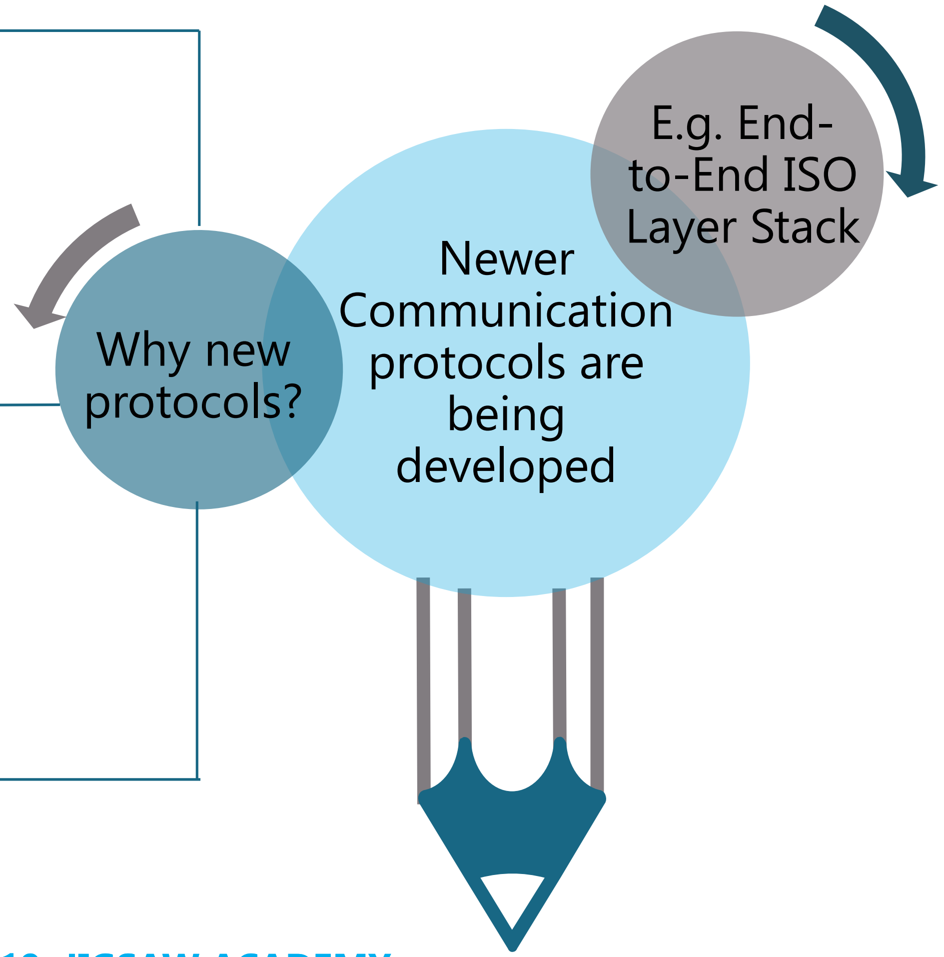
An **older protocol**, (basis for **Zaber),** now being used by devices in the IOT ecosystem

Newer IOT Communication Protocols

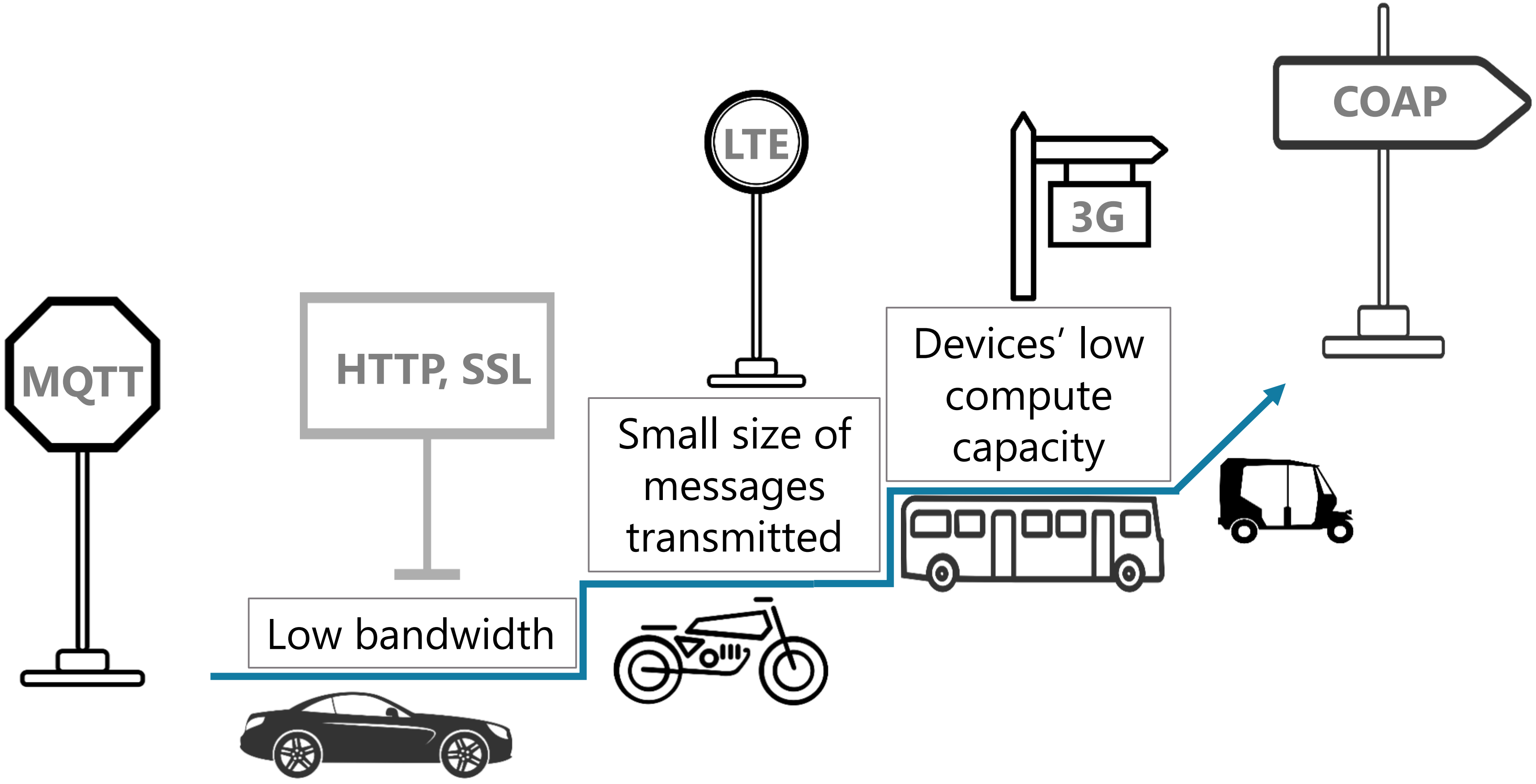
Deploy an end-to-end IOT application, including hardware and software, both have to be compatible

Proprietary protocols are being developed

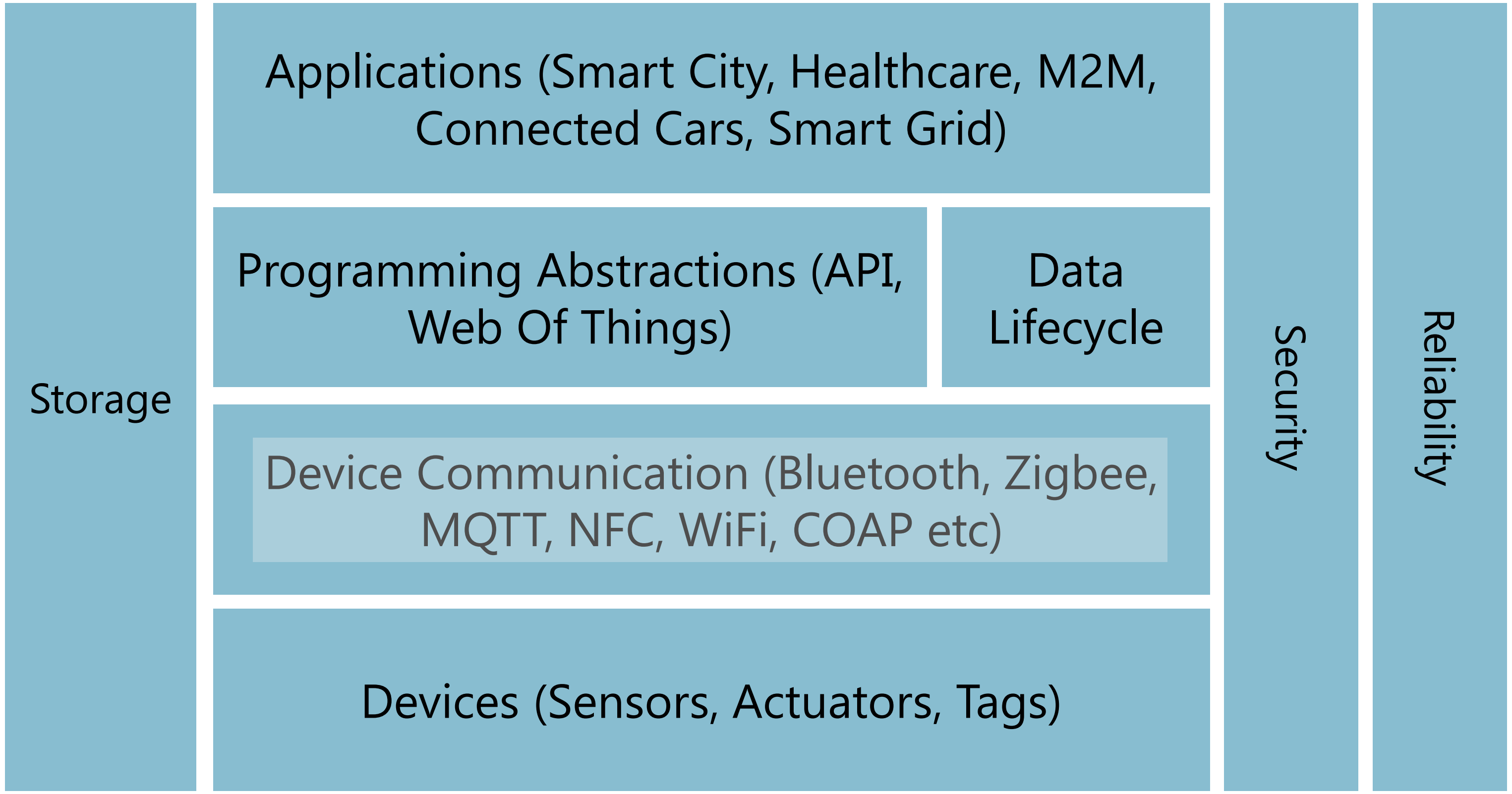
Lack of standardization affects devices and device communication



Newer IOT Communication Protocols



IOT Gateway


Applications (Smart City, Healthcare, M2M, Connected Cars, Smart Grid)
Programming Abstractions (API, Web Of Things)
Data Lifecycle
Device Communication (Bluetooth, Zigbee, MQTT, NFC, WiFi, COAP etc)
Devices (Sensors, Actuators, Tags)
Storage
Security
Reliability

IOT Gateway Devices

IOT gateways are devices consisting of both hardware and software elements

Some of them are pure software

- **Example**: Eclipse Foundation has a new gateway software which enables you to
  - ❑ Work with multiple IOT elements
  - ❑ Create abstractions

IOT Gateway Devices

From a hardware perspective a gateway device

- **Acts** like a firewall in an enterprise scenario which screens all output requests
- **Is able to** do something like a protocol translation (between devices working with different protocols)
- **Is able to** work with and translate to diverse protocols
- **Has** software elements which can do some aggregation, and analytics

IOT Gateway Devices

In some cases, IOT gateway devices can block all but valid packets passing through at network level

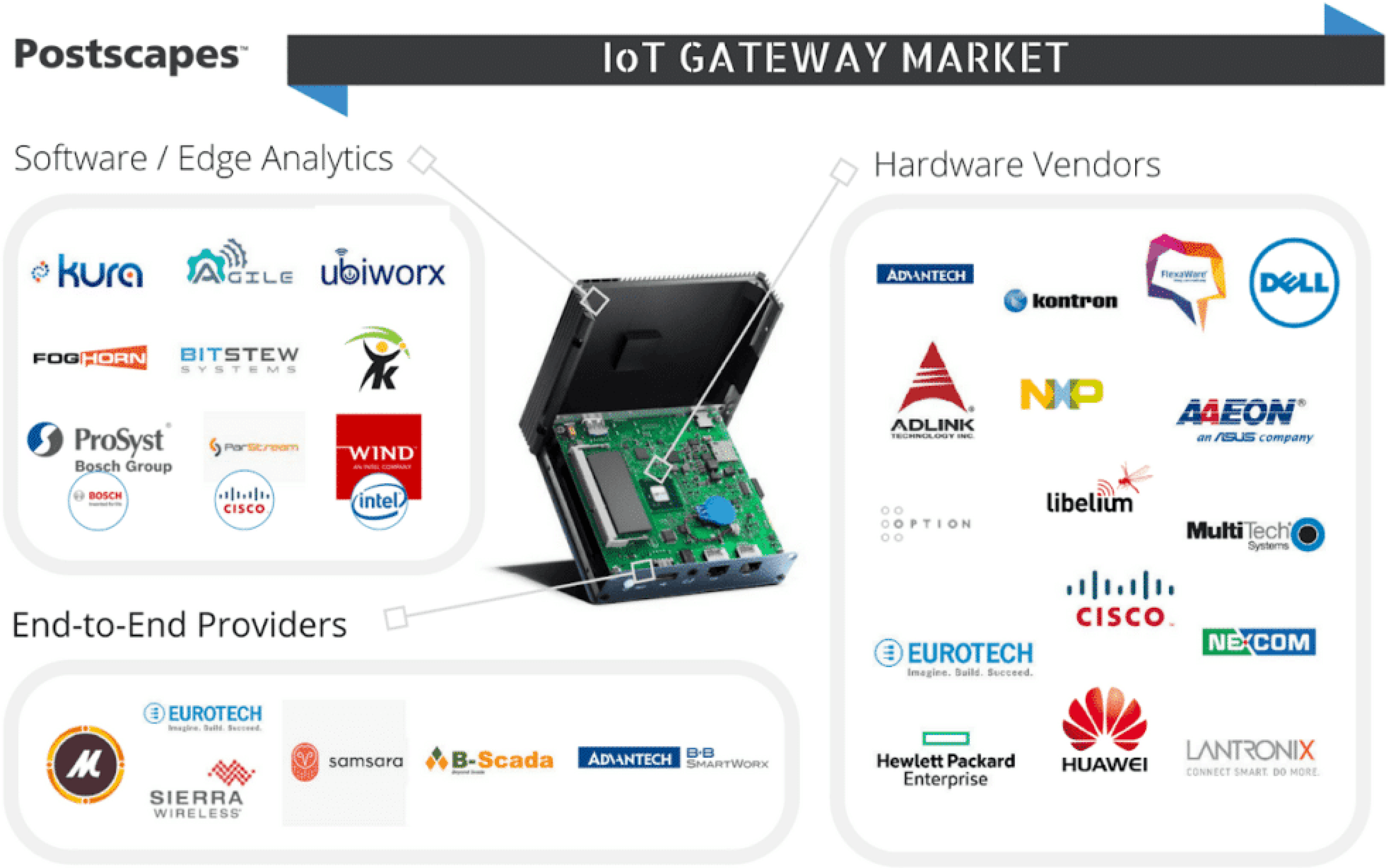
They can channel requests to specific elements depending upon the nature of the request

There is no standard definition of gateways

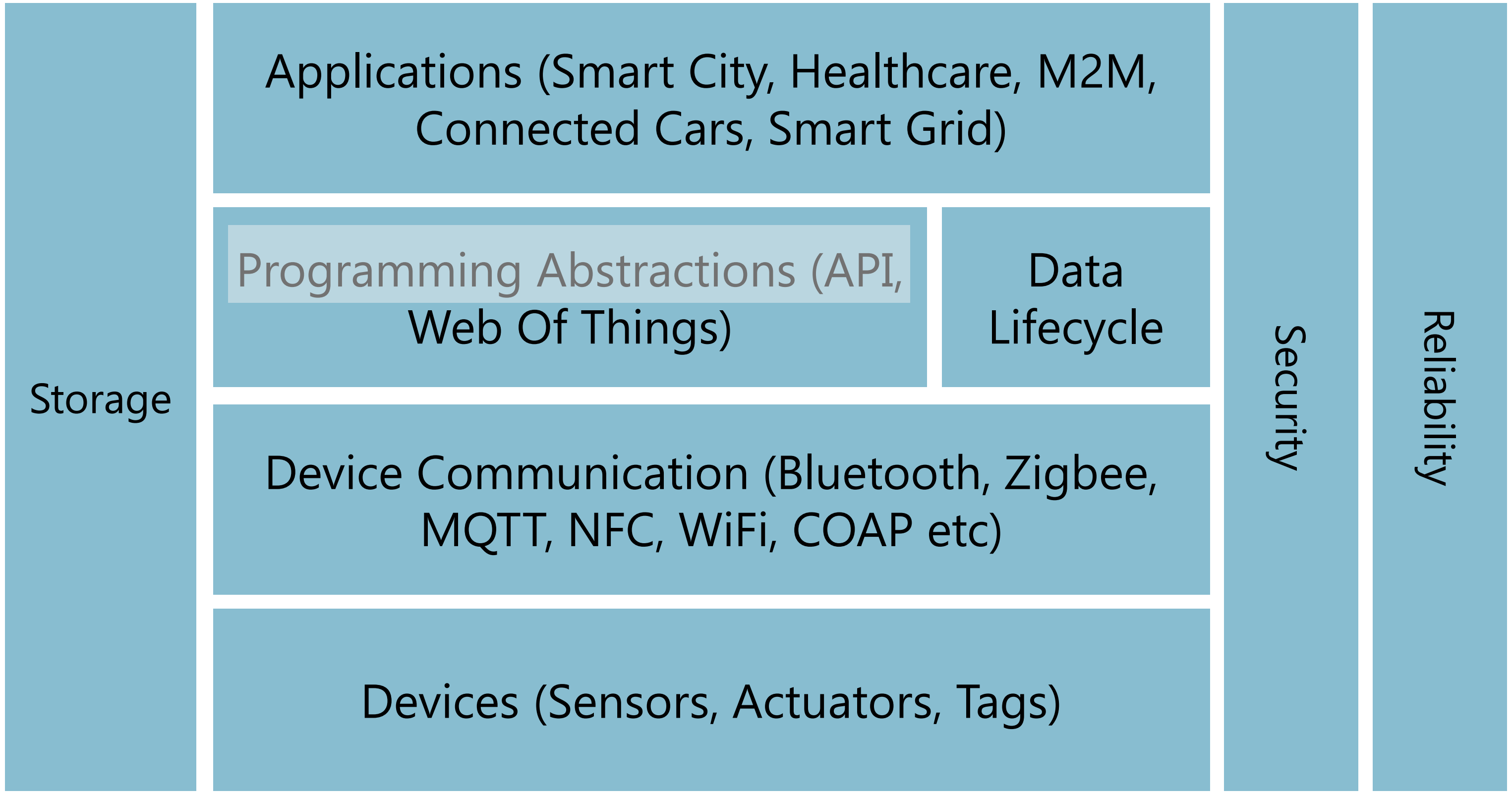
Some have a software-only approach, others a hardware-only approach like Dell, Cisco, and Intel (even though they have very complex software inside)

IOT gateways have emerged as a way to

- Manage the complexity and variety of devices
- Lower the complexity of interaction between the 'outside world' (the internet) and the local deployment of devices


Postscapes
IoT GATEWAY MARKET
Software / Edge Analytics
kura
AGILE
ubiworx
FOGHORN
BITSTEW SYSTEMS
ProSyst Bosch Group
BOSCH
ParStream
CISCO
WIND
intel
End-to-End Providers
EUROTECH Imagine. Build. Succeed.
SIERRA WIRELESS
samsara
B-Scada
ADVANTECH B+B SMARTWORX
Hardware Vendors
ADVANTECH
kontron
FlexaWare
DELL
ADLINK TECHNOLOGY INC.
NXP
AAEON an ASUS company
OPTION
libelium
MultiTech Systems
CISCO
EUROTECH Imagine. Build. Succeed.
NEXCOM
Hewlett Packard Enterprise
HUAWEI
LANTRONIX CONNECT SMART. DO MORE.

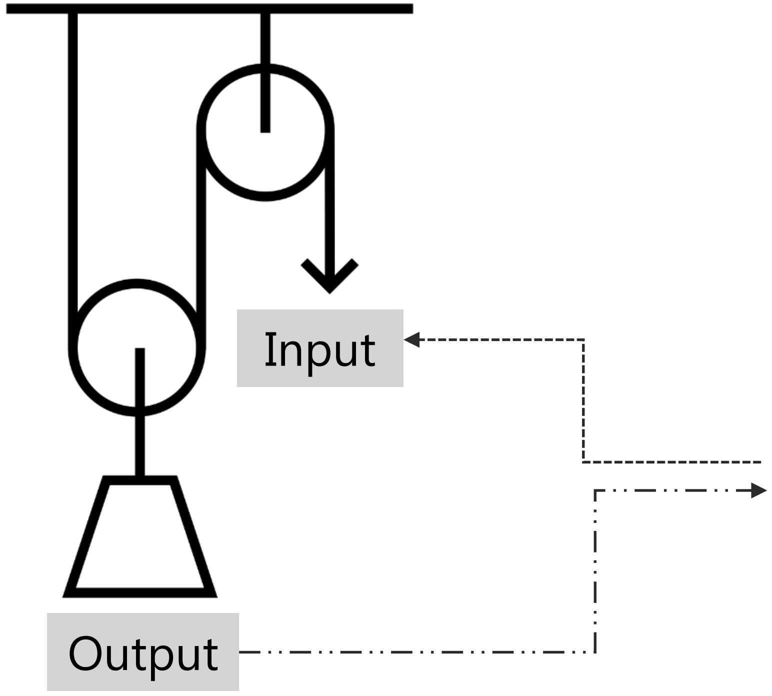
Creating Programmable Objects

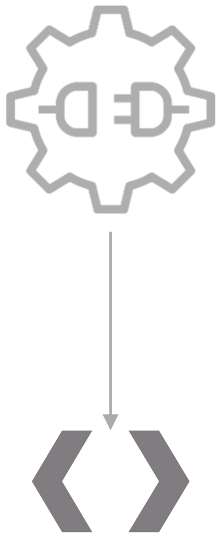


Creating Programmable Objects

Programmable APIs

Sensor: Embedded System





Allows programs to work with the internal embedded system

IOT platforms provide abstractions that expose the underlying embedded system using a structured API

To make API output available to other programs, Universal Interfaces can be deployed

Creating Programmable Objects

Universal Interfaces: Make APIs more programmable

Web of Things

- Relies on a protocol like REST to connect to the internet
- Produces far better generic abstractions when compared to proprietary APIs

REST-based interface

Produces an internet addressable URI to access, manipulate, and claim the output of the underlying device

Device interfaces

Device interfaces and data, in the form of **APIs** and **programming abstractions** enable productive programming with IOT

Creating Programmable Objects

Universal Interfaces: Make APIs more programmable

Achievement

To be able to write an end-to-end computer program with detailed logic, without worrying about the internals of the device, or its input/output

Tool kits

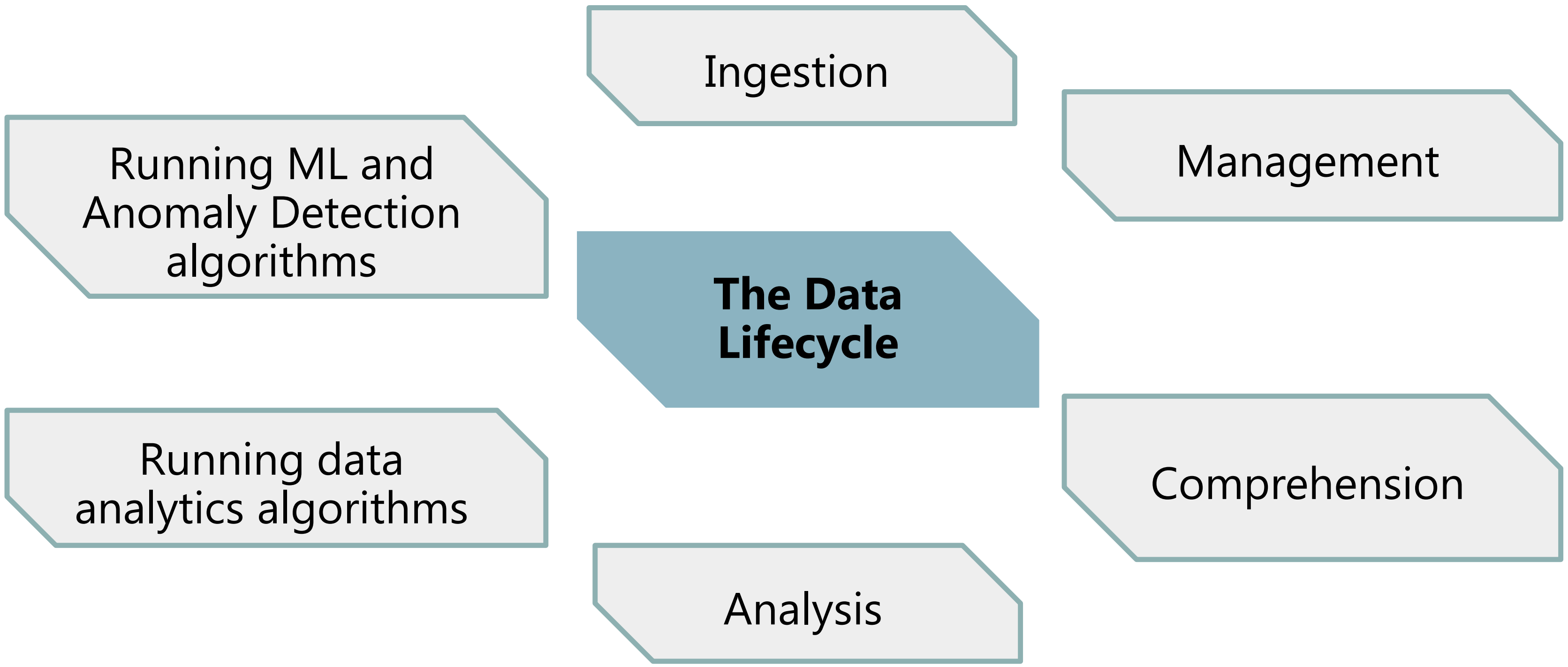
Tool kits with standard APIs and programs to work with, are available from popular tool vendors like **Microsoft**, **Cisco**, and **Dell**

Adoption accelerator

This is an accelerator for the adoption of IOT in broader enterprise and analytics applications

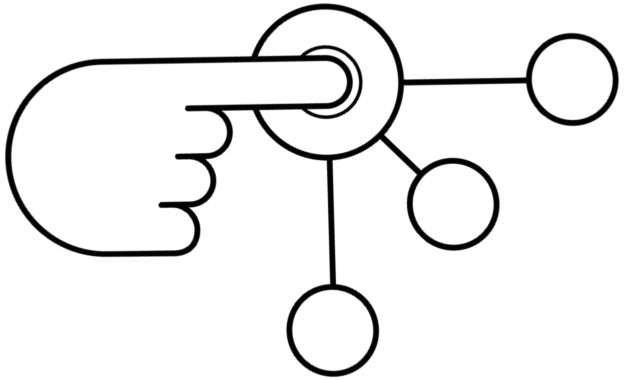
End-To-End Life Cycle

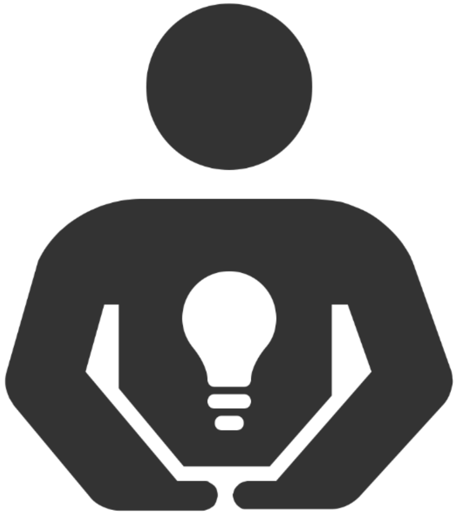
- Programming is important, but so is data
- Device data has an **end-to-end lifecycle**



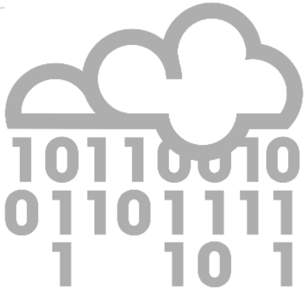
Streaming Data Management

Mechanisms for dealing with large data streams are needed



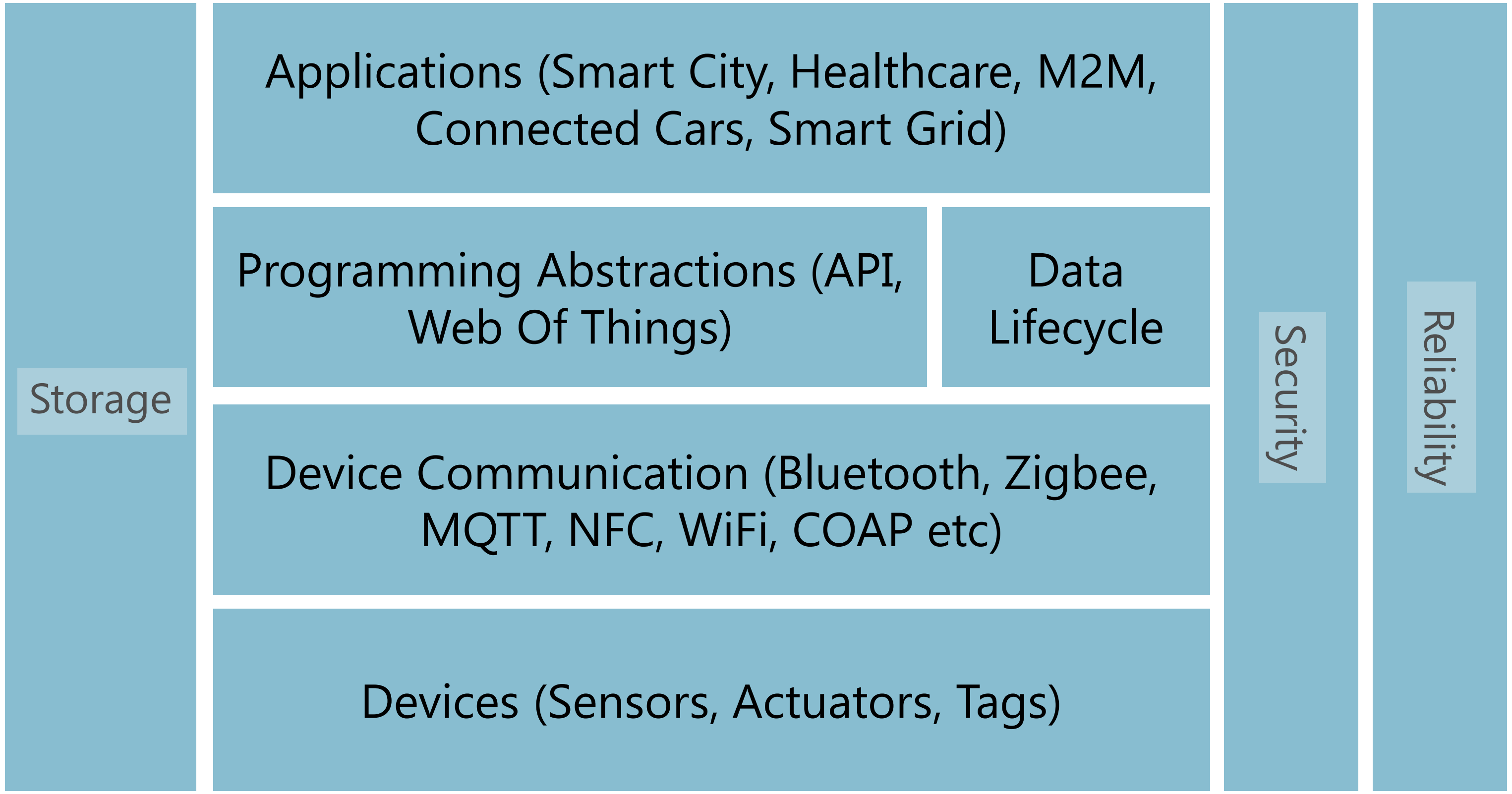


proprietary and **standards-based tools**
Open Source for data science lifecycle management



Non-Functional Requirements: Storage

Architecture covers the non-functional requirements of any system



Non-Functional Requirements: Storage

Architecture covers the non-functional requirements of any system

Growth

Explosion in size of data being generated by sensors and actuators

Storage

Data storage facilitates meaningful analysis and data usage in higher-level applications

Real Time

Processing and discarding real-time data using a streaming protocol is not always possible

Mainstreaming

Cheap storage is helping to mainstream IOT analytics

Storage

The availability of cloud-based storage as a service…

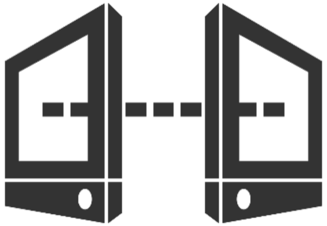
…enables cloud, IOT devices and apps to converse

Storage Models

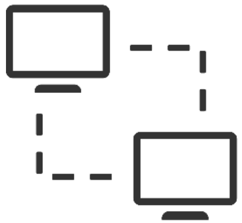
To work with the unreliable storage, place data in a programming abstraction...

...caution customers about the ramifications of cloud-based storage

Local, device-based storage



Gateway storage



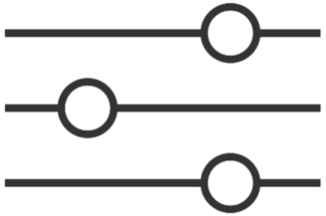
Intranet/Extranet/Internet

<u>Enterprise-level IOT equipment</u> like gateways can also <u>act as storage devices</u>

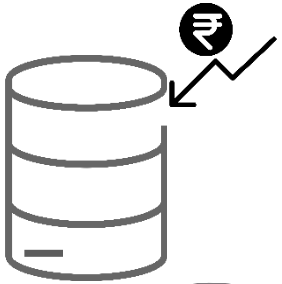
Storage Models

Access to **multiple devices** is no longer a necessity

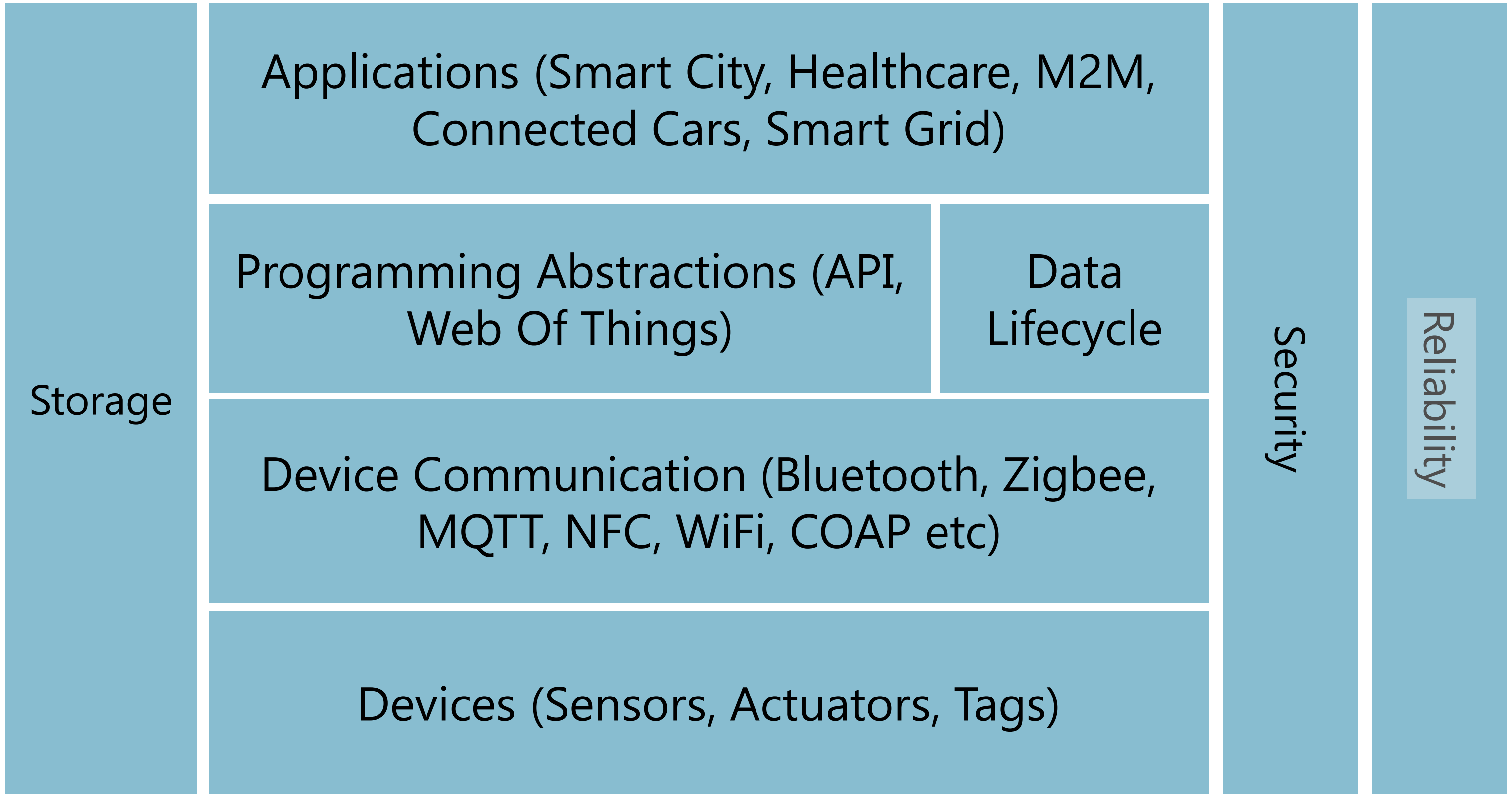
Lower number of parameters to be monitored in an application when space is no longer an issue



Cheap storage has a multiplier effect on the IOT ecosystem



Reliability of Communication (Protocols)



Reliability of Communication (Protocols)

Reliability in IOT refers to reliable **communication**

Data from devices should not be **lost in transit** on their way…

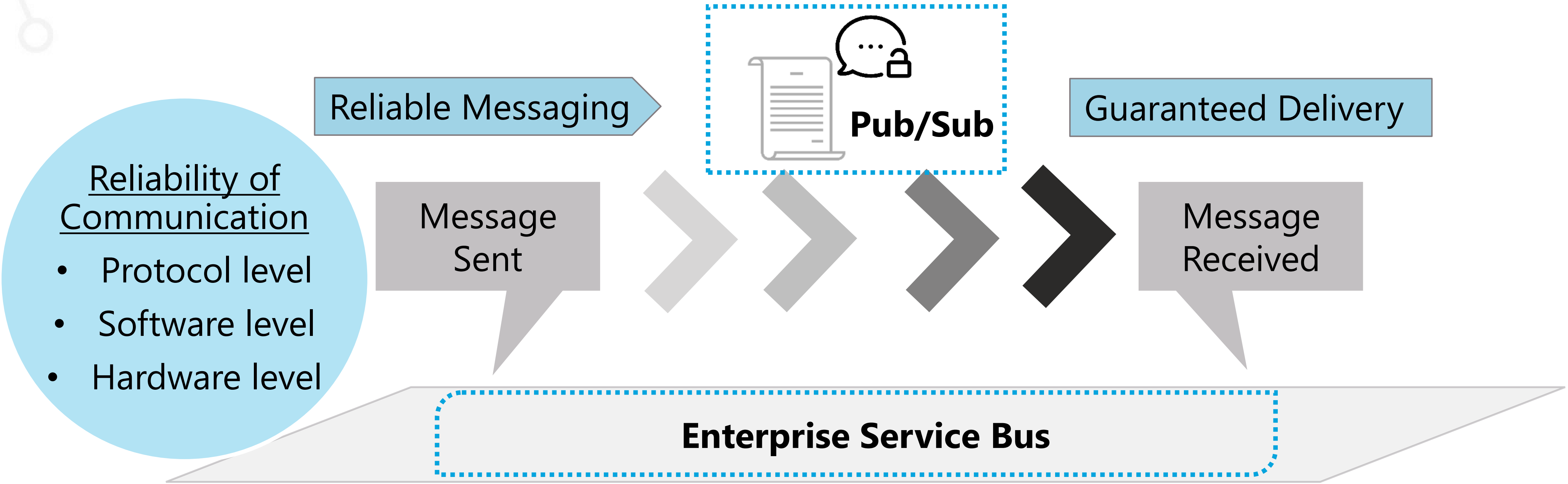
…to a target device

…to an internet service/internet storage

Protocol must ensure that **delivery** is **guaranteed**

New protocols like **MQTT** enable more **reliable messaging** from one source to the other

Reliability of Communication (Software)



Increases reliability of messages being sent between IOT devices

There is a role for strong reliable messaging middleware in IOT deployments

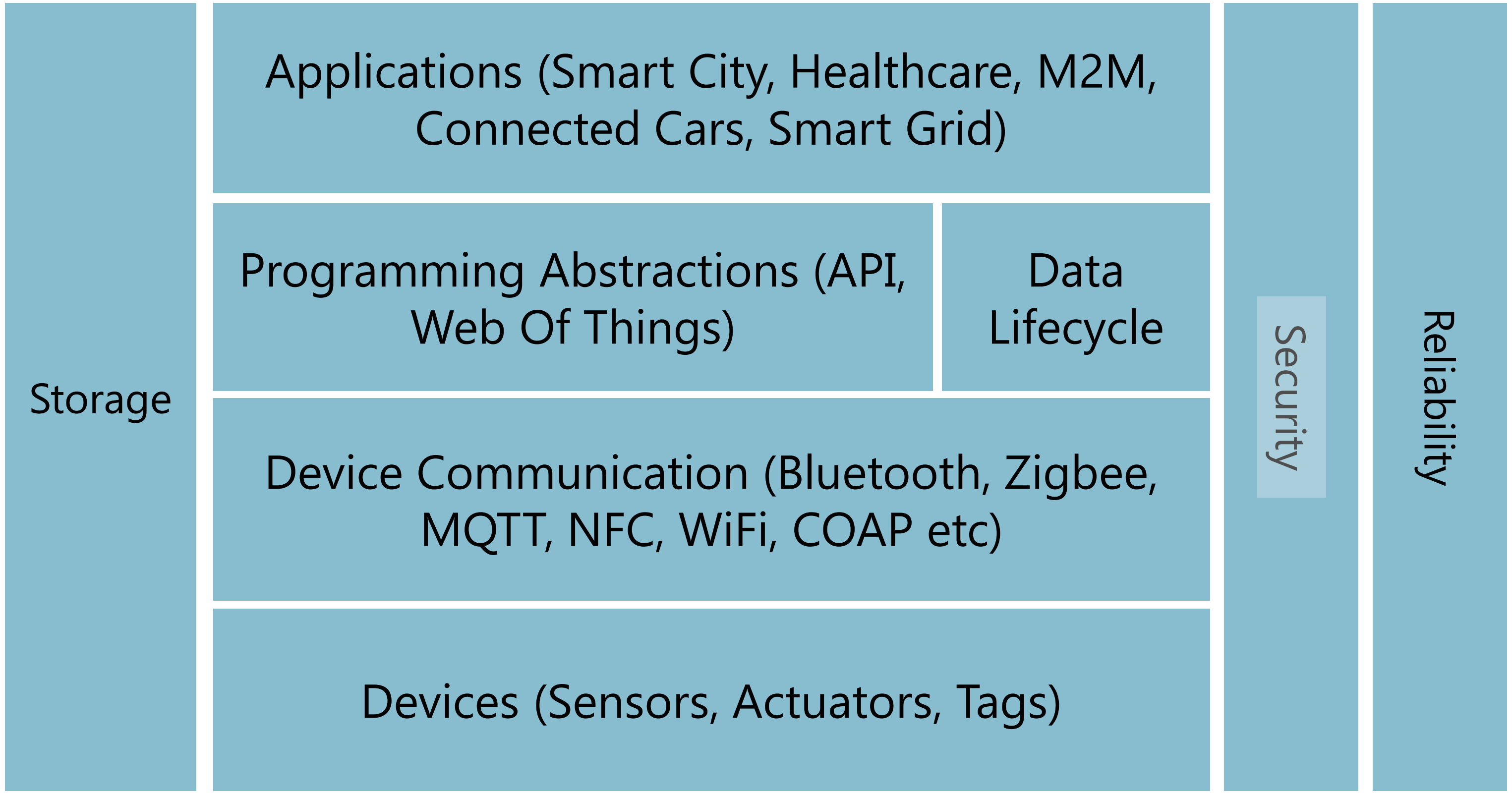
Appropriate Quality Control Measures

Reliability is crucial at the software level, the hardware level, and at the communication level

Certification of devices is also important from security perspective

Appropriate quality control procedures in fabrication and deployment of hardware (critical)

Security of Devices



Security of Devices

Create a rugged environment

Ensure safety of sensors

Do not place in a high temperature zone

Make sure sensors function in the field and are:

- Sensing data
- Transmitting data properly

Device (Google Nest)

Sensor

Boiler

Actuator

Hardware

Software

Security of Software

It protects the sensor from rogue software

Security software is important; each sensor requires one

Shared keys between software developers is a good safety measure

Keying in a unique security code to enable updates, is critical

Security of Data

Data management must meet **CIA** standards of **Confidentiality, Integrity, Availability**

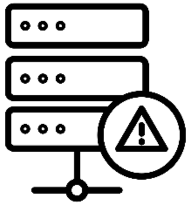
Only **authorised software**, or personnel with the right credentials, should be able to access it

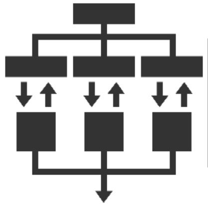
Data should not be manipulated or tampered with

Confidentiality measures such as digital keys or signatures protect critical sensor data

Security and Availability of Data

Availability Ensuring that devices are not flooded with too many requests (or denial of service attacks/DoS)





Awareness of expected throughput from devices is critical

Throughput shouldn’t be overwhelmed with demands for large amounts data at the same time

Authentication or Authorization

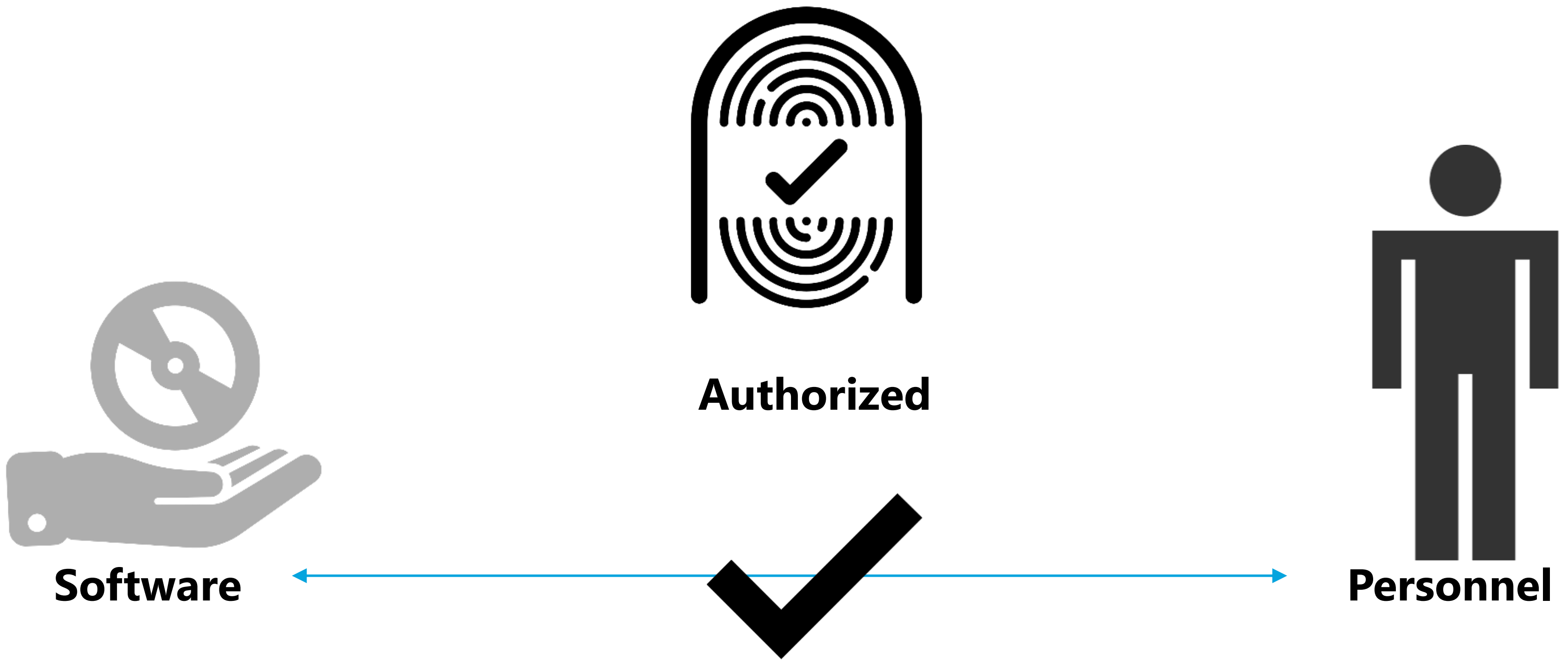
Authorized?

Software

Personnel

Devices

Authentication or Authorization



Authentication or Authorization