Sistemas de Computação Móvel e Ubíqua

2021/2022

IoT and Cloud

2021/2022

2

IoT: what is IoT

"The term ***Internet of Things*** generally refers to scenarios where internet connectivity and computing capability extends to objects, sensors and everyday items not normally considered computers, allowing these devices to **generate, exchange and consume data with minimal human intervention**."

https://www.internetsociety.org/wp-content/uploads/2017/08/ISOC-IoT-Overview-20151221-en.pdf

3

IoT: enabling technologies

Ubiquitous IP connectivity

- Low–cost, high–speed, pervasive IP network connectivity, especially through wireless technology, makes almost everything "connectable''
- Network and Transport protocols for ad-hoc, zero-config and low-power

Miniaturization and computing economics

- Computing and communications technology and sensor devices can be incorporated into any small objects at very low cost.

Advances in **cloud computing** and its services

- Cloud computing can provide the resources needed to store and analyse data in a way that makes it possible to extract information and knowledge from this data.

4

Examples: Human & Home

Devices attached or inside the human body

- Devices (wearables and ingestibles) to monitor and maintain human health and wellness; disease management, increased fitness, higher productivity

Devices at homes and buildings

- Home controllers, including lights, appliances, gardening, and security systems

Examples: Industry and Commerce

Manufacturing

- Monitoring of machines to guarantee they are working correctly
- Monitor products to identify quality issues

Retail

- Better management of inventories
- Smart shelves equipped with weight or RFID sensors for better monitoring of inventory

Examples: Transports and Vehicles

Transports
- Monitoring of cargo – e.g. temperature of drugs, food, etc.
- Better management of inventories

Vehicles
- Condition-based maintenance, usage-based design, presales analytics

Examples: Smart Cities

Adaptive traffic control

Environmental monitoring

Resource management – e.g. water, energy, waste, ...

…

IoT key challenges

Availability – anytime, anywhere, for anyone

Reliability – proper working, resilient to failures

Mobility – from devices (including users)

Performance – QoS, response time, cost

Scalability – more devices, functionality and users

Interoperability – heterogeneous devices, protocols, services

Security and Privacy – authentication, data and users' privacy

Management – configuration, monitoring and deploy of devices, services, etc

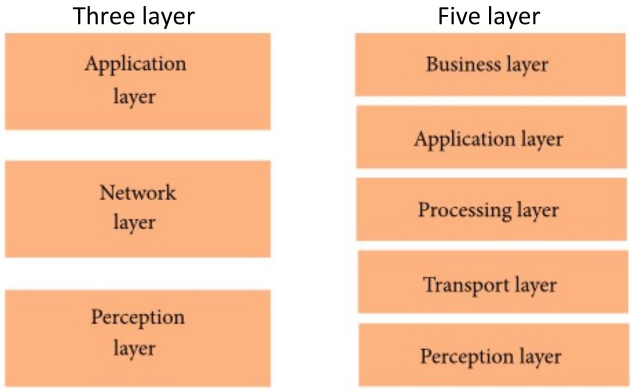
IoT - Platform

An IoT platform provides a comprehensive set of generic services that facilitates the development, deployment, maintenance, analytics as well as intelligent decision making capabilities to an IoT application.

One of the issues that usually needs to be address is that of communication among devices of multiple types

- There is a need of some common standards to hide heterogeneity of various devices by providing a common working environment to them.

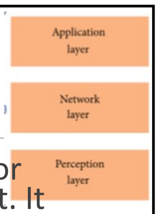
IoT - Architecture



From: Internet of Things: Architectures, Protocols, and Applications
https://www.hindawi.com/journals/jece/2017/9324035/

11

IoT - Three Layer Architecture



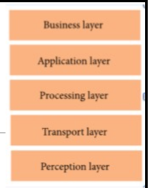
Perception layer is the physical layer, which has sensors for sensing and gathering information about the environment. It senses some physical parameters or identifies other smart objects in the environment.

Network layer is responsible for connecting to other smart things, network devices and servers. Its features are also used for transmitting and processing sensor data.

Application layer is responsible for delivering application specific services to the user. It defines various applications in which the IoT can be deployed (eg. smart homes, smart cities, smart health, …)

12

IoT - Five Layer Architecture



Perception layer …

Transport layer transfers the sensor data from the perception layer to the processing layer and vice versa through networks such as wireless, 4G, LAN, Bluetooth, RFID, NFC, etc. …

Processing layer stores, analyzes, and processes huge amounts of data that comes from the transport layer. Manage and provide a diverse set of services to the lower layers (employ technologies such as databases, cloud computing, and big data processing modules).

Application layer …

Business layer manages the whole IoT system, including applications, business and profit models, and users' privacy.

13

IoT - Five Layer Architecture

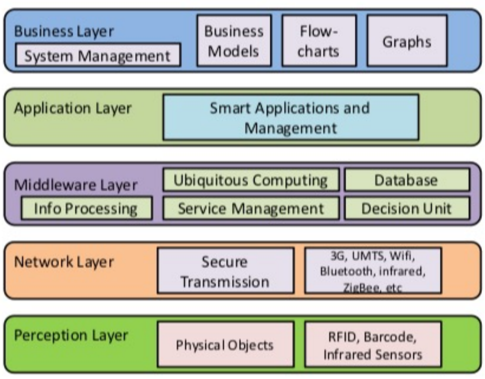
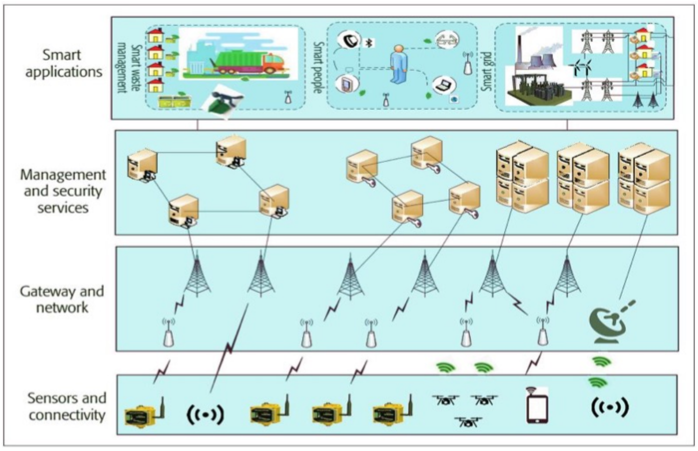


Figure 3: The IoT Architecture.

https://pure.qub.ac.uk/portal/files/81384964/PID2566391.pdf

14

IoT – Communication Infrastructure



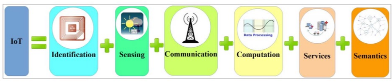
https://ieeexplore.ieee.org/document/7955906

IoT - Elements

To deliver the expected IoT functionalities there are several elements that have to be considered.

From: “Internet of Things: A Survey on Enabling Technologies, Protocols and Applications”

IoT – Elements (1)



Identification

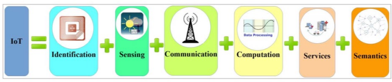
- Identification methods are used to provide a clear identity for each object within the network.
- Identification is crucial for the IoT to name and match services with their demand.

Many identification methods are available for the IoT:

- Name – Object ID, eg. electronic product codes (EPC), ubiquitous codes (uCode)
- Address – address within a communication network, eg. IPv6 and IPv4

17

IoT – Elements (2)

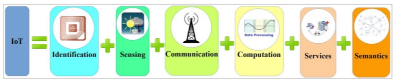


Sensing

- The IoT sensing means gathering data from related objects within the network.
- The collected data is analyzed to take specific actions based on required services.
- The IoT sensors can be smart sensors, actuators or wearable sensing devices.
- e.g., Arduino WiFi, Raspberry PI, embedded hardware, etc..., devices that typically connect to a central management portal to provide the required data by customers.

18

IoT – Elements (3)

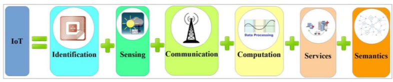


Communication

- The IoT communication technologies connect heterogeneous objects together to deliver specific smart services.
- Typically, the IoT nodes should operate using low power in the presence of lossy and noisy communication links.
- Examples of communication protocols used: WiFi, Bluetooth, ZigBee, GSM/3G/*4G/5G*, …
- More specific communication technologies are also used like RFID, BLE (Bluetooth Low Energy), NFC (Near Field Communication), etc.

19

IoT – Elements (4)



Computation

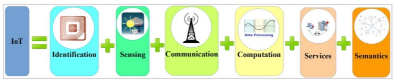
- Processing units (e.g., microcontrollers) and software applications represent the core and the computational ability of the IoT.
- Various hardware platforms for devices such as Arduino, ESP, Intel Galileo, Raspberry PI, …
- A large part of computation in IoT systems is performed at the edge (gateways) or cloud platforms

COMMON OPERATING SYSTEMS USED IN IOT ENVIRONMENTS

Operating System	Language Support	Minimum Memory (KB)	Event-based Programming	Multi-threading	Dynamic Memory
TinyOS	nesC	1	Yes	Partial	Yes
Contiki	C	2	Yes	Yes	Yes
LiteOS	C	4	Yes	Yes	Yes
Riot OS	C/C++	1.5	No	Yes	Yes
Android	Java	-	Yes	Yes	Yes

20

IoT – Elements (5)



Services

IoT services can be categorized under four classes:

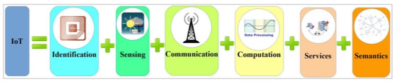
Identity-Related - relate identified real-world objects to virtual-world objects.

Information Aggregation - collect and summarize raw sensory measurements.

Collaborative-Aware - act on top of Information Aggregation Services and use the obtained data to make decision and react accordingly

Ubiquitous - aim to provide Collaborative-Aware Services anytime they are needed to anyone who needs them anywhere.

IoT – Elements (6)



Semantics / analytics

- Semantics in the IoT refers to the ability to extract knowledge smartly by different machines to provide the required services.
- Knowledge extraction includes:
  - discovering and using resources and modeling information.
  - recognizing and analyzing data to make sense of the right decision to provide the exact service.
- Semantic represents the brain of the IoT by sending demands to the right resource.

IoT Platform main features

Device management

Integration

Security

Protocols for data collection

Analytics

Support for visualizations

IoT - Device Management

Maintains a list of connected devices and tracks their operation status.

- Identity management is a key challenge and it has implication in the security of the system.

Handles configuration and firmware (or any other software).

Manages updates.

Provides device level error reporting and error handling.

IoT - Integration Support

Support for integration of different "things" and services is an important feature expected from an IoT software platform.

The API should provide access to the operations and data that needs to be exposed from the IoT platform.

- APIs often use REST to achieve this aim.

IoT - Information Security

Millions of devices being connected with an IoT platform means we need to anticipate a proportional number of vulnerabilities.

The network connection between the IoT devices and the IoT software platform needs to be encrypted with a strong encryption mechanism.

e.g. TLS is used for that.

IoT - Information Security

Authentication is key for achieving security.

Common approach:

1. Devices have unique identities, which are registered in the IoT platform;
2. Devices include cryptographic keys stored in hardware, which are used to authenticate the device in the platform.

IoT – Data Collection Protocols

Protocols used for data communication between the components of an IoT software platform.

An IoT platform may need to be scaled to millions or even billions of devices (nodes).

Lightweight communication protocols should be used to enable low energy use as well as low network bandwidth functionality.

May deal with heterogeneous devices and heterogeneous physical communication technologies.

STANDARDIZATION EFFORTS IN SUPPORT OF THE IOT

Application Protocol		DDS	CoAP	AMQP	MQTT	MQTT-SN	XMPP	HTTP REST
Service Discovery		mDNS				DNS-SD		
Infrastructure Protocols	Routing Protocol	RPL						
	Network Layer	6LoWPAN				IPv4/IPv6		
	Link Layer	IEEE 802.15.4						
	Physical/ Device Layer	LTE-A	EPCglobal	IEEE 802.15.4	Z-Wave			
Influential Protocols		IEEE 1888.3, IPSec				IEEE 1905.1		

MQTT Protocol (note)

MQTT stands for MQ Telemetry Transport.

It is a publish/subscribe, extremely simple and lightweight messaging protocol

- designed for constrained devices and low-bandwidth, high-latency or unreliable networks.
- also attempting to ensure reliability and some degree of assurance of delivery.

These principles also turn out to make the protocol ideal of the emerging “machine-to-machine” (M2M) or “Internet of Things” and for mobile applications where bandwidth and battery power are at a premium.

http://mqtt.org/

IoT - Data Analytics

Data collected from the sensors connected to an IoT platform needs to be analyzed (usually in a cloud)

Main types of analytics which can be conducted on IoT data: real-time, batch, predictive, and interactive analytics.

Real-time analytics conduct online (on-the-fly) analysis of the streaming data.

Batch analytics runs operations on an accumulated set of data. Thus, batch operations run at scheduled time periods and may last for several hours or days.

Predictive analytics is focused on making predictions based on various statistical and machine learning techniques.

Interactive analytics runs multiple exploratory analysis on both streaming and batch data.

IoT key challenges

Availability – anytime, anywhere, for anyone

Reliability – proper working, resilient to failures

Mobility – from devices (including users)

Performance – QoS, response time, cost

Scalability – more devices, functionality and users

Interoperability – heterogeneous devices, protocols, services

Security and Privacy – authentication, data and users' privacy

Management – configuration, monitoring and deploy of devices, services, etc

Edge Computing / Fog Computing

The Internet of Things (IoT) is generating an unprecedented volume and variety of data.

Analyse the most time-sensitive data at the network edge
- close to where it is generated
- instead of sending vast amounts of IoT data to the cloud.

Send selected data to the cloud for
- historical analysis
- longer-term storage.

33

Edge Computing

As possible solution to some of the problems:
- Bring some compute and storage resources to the edge of the network instead of relying on the cloud for everything.
- Improve response time.
- Reduce network and cloud resources loads.

34

Edge Computing

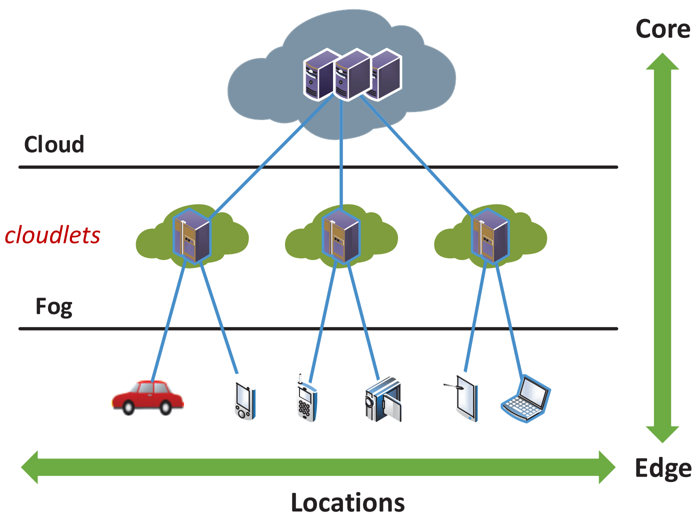
Fog computing - a term originally coined by Cisco. Edge computing is more common currently

In contrast to the cloud, fog platforms have been described as dense computational architectures at the network's edge.

Characteristics of such platforms reportedly include low latency, location awareness and use of wireless access.

Benefits include real-time analytics and improved security.

35



36

Edge Computing

The fog can be viewed as a cloud, which is close to the "ground".

Data can be stored, processed, filtered, and analyzed on a **cloudlet** or **smart gateway** at the edge of the network before sending it to the cloud through expensive communication media.

The fog and cloud paradigms can go together for a better performance of IoT applications.

37

Edge Computing

Edge tasks:

- collecting sensor data, preprocessing and filtering
- communicating with the cloud, sending only necessary data
- provide computation, storage and networking to IoT devices
- provide cloud services closer to IoT
- monitoring power consumption of IoT devices
- monitoring activities and services of IoT devices
- ensuring security and privacy of data.

39

Edge Architecture - features

Low latency: less time is required to access computing and storage resources on fog nodes (smart gateways).

Location awareness: as the edge is located on the edge of the network, it is aware of the location of the applications and their context. This is beneficial as context awareness is an important feature of IoT applications.

Distributed nodes: edge nodes are distributed unlike centralized cloud nodes. Multiple edge nodes need to be deployed in distributed geographical areas in order to provide services to mobile devices in those areas. (e.g. in vehicular networks, deploying fog nodes at highways can provide low latency data/video streaming to vehicles).

41

Edge Architecture - features

Mobility: the edge supports mobility as smart devices can directly communicate with smart gateways present in their proximity.

Real time response: edge nodes can give an immediate response unlike the cloud, which has a much greater latency.

Interaction with the cloud: edge nodes can further interact with the cloud and communicate only that data, which is required to be sent to the cloud.

42

IoT - Platforms

Summary

- IoT Platform connects the data network to the sensors and provides insights using backend applications to make sense of plethora of data generated by hundreds of sensors.
- IoT Platform fills the gap between the device sensors and data networks.

There are many IoT platforms available, that provide option to deploy IoT applications on the go.

43

IoT – Amazon Web Service

Amazon Web Services (AWS) provides trusted, cloud-based solutions.

AWS IoT is a cloud platform that lets connected devices easily and securely interact with cloud applications and other devices.

AWS IoT can support billions of devices and trillions of messages, and can process and route those messages to AWS endpoints and to other devices reliably and securely.

https://aws.amazon.com/iot-platform/

44

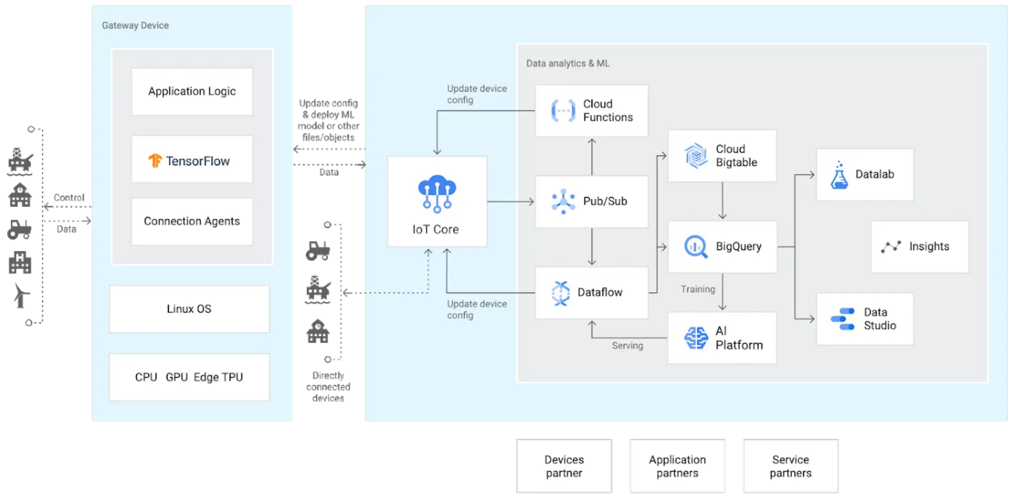
IoT – Google Cloud

Google Cloud IoT is a set of fully managed and integrated services that allow you to easily and securely connect, manage, and ingest IoT data from globally dispersed devices at a large scale, process and analyze/visualize that data in real time, and implement operational changes and take actions as needed.

https://cloud.google.com/solutions/iot/

45

IoT – Google Cloud



46

IoT – Microsoft Azure

Microsoft Azure is a growing collection of integrated cloud services that developers use to build, deploy, and manage applications through Microsoft global network of data centers.

Connect quickly and scale with efficiency:
- Easily scale from just a few sensors to millions of simultaneously connected devices.

Analyze and act on untapped data:
- Collect data from devices and sensors, and use built-in capabilities to visualize - and act on - that data.
- Set up real-time analytics by using SQL-based syntax in a scalable, high-performance.

Integrate:
- Easily integrate Azure IoT Suite with your systems and applications

Enhance security of IoT solutions with per-device authentication:
- Set up individual identities and credentials for each of the connected devices

https://azure.microsoft.com/en-us/overview/iot/

47

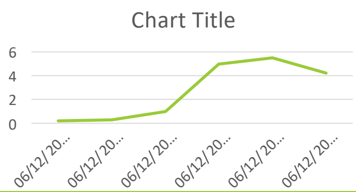
(Many) IoT applications are about time series

Many IoT applications register the values of a variable (e.g. temperature) with an associated timestamp.

This is a **Time Series**.

There are currently many databases specialized in handling time series: time-series databases.

Timestamp	Value
2016-12-06 08:58:00	0.2
2016-12-06 08:58:05	0.3
2016-12-06 08:58:10	1.0
2016-12-06 08:58:15	5.0
2016-12-06 08:58:17	5.5
2016-12-06 08:58:20	4.2



49

Requirements: write dominate

It should always be possible to execute writes.

Write scale is huge - example from server monitoring

2,000 servers, VMs, containers, or sensor units
1,000 measurements per server/unit
every 10 seconds
= 17,280,000,000 distinct points per day

Read scale is smaller

- E.g. Facebook Gorilla reports "couple orders of magnitude lower"
- Automated systems watching "important" time series
- Dashboards for humans
- Human operators wishing to diagnose an observed problem

51

Requirements: state transitions

Identify issues that occur on monitored data.

TSDB should support fine-grained aggregations over short-time windows.

TSDB should have the ability to identify state transitions within tens of seconds.

52

Requirements: high availability and fault tolerance

TSDB should support write and reads even in the presence of network partitions.

TSDB should replicate data to survive server failure.

Other requirements

ACID guarantees are not a requirement, but...

...high percentage of writes must succeed at all times (**some may fail**... typically not a problem under high load). Why?

... recent data is of higher value than older data.

Design of a TSDB

Problem: scale of data is enormous… just do the math considering you have N sensors, sampling values every T seconds and each value occupies K bytes.

Solution 1: compression of the data

Time series compression

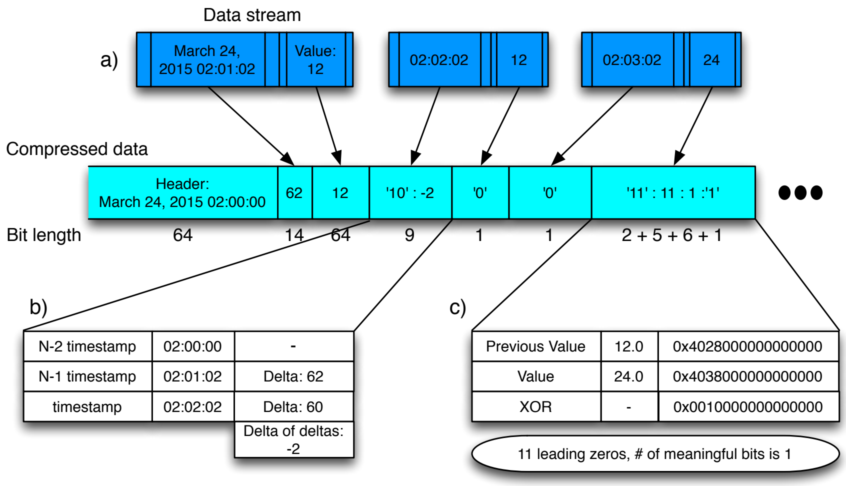
Compresses data points within a time series.

E.g.: Facebook Gorilla

Each data point is a pair of 64 bit values representing the time stamp and value at that time.

Timestamps and values are compressed separately using information about previous values – storing deltas is cheaper.

Time series compression



57

Design of a TSDB

Problem: scale of data is enormous… just do the math considering you have N sensors, sampling values every T seconds and each value occupies K bytes.

Often there is no need to keep all data forever.

Solution 2: Discard data that is not necessary or summarize old data.

Keep all data values only for the last K days. For older data, just keep an aggregation – e.g. average, minimum, maximum for each hour/day.

65

Design of a TSDB

Problem: need to write fast, read fast

Solution: new storage designs, keep indices in memory

Indexing time series

Need to support fast writes…

… and fast reads

Database indexes (B-trees) are not appropriate for time series database

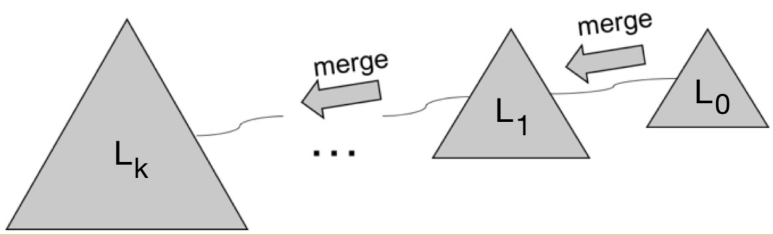
Time series databases indexes usually based on LSM trees

Log-structured merge tree (LSM-tree)

An LSM-tree consists of a hierarchy of storage levels that increase in size.

The first level, L0, is stored in memory – used to buffer updates. When this level gets full, it is merged with the other levels.

The other levels are stored on disk.



LSM-tree: operations (cont.)

A simple lookup consists in:
- Searching the value in L0
- If not found, continue searching in the following levels
  - For efficiency, each level records a summary of the elements present, as a Bloom filter

Range lookups consist in:
- Executing a range search in every level
- Slow, but…
  - If searching for recent values, they will be in L0 (if large enough)
  - The way merging works makes values added at similar times to be in close levels

Data model – e.g. InfluxDB

temperature,device=dev1,building=b1 internal=80,external=18 1443782126

Measurement — Tags (tagset all together) — Fields — Timestamp

Data divided in a sequence of time series.

Each field has its unique identifier.

Key for a value includes the identifier of the field and the timestamp.

temperature,device=dev1,building=b1#internal → 1

1 → (1443782126,80)

temperature,device=dev1,building=b1#external → 2

2 → (1443782126,18)

key space is ordered

Key	Value
1,1443782126	80
1,1443782127	81
2,1443782126	18

From SQL schema to InfluxDB schema

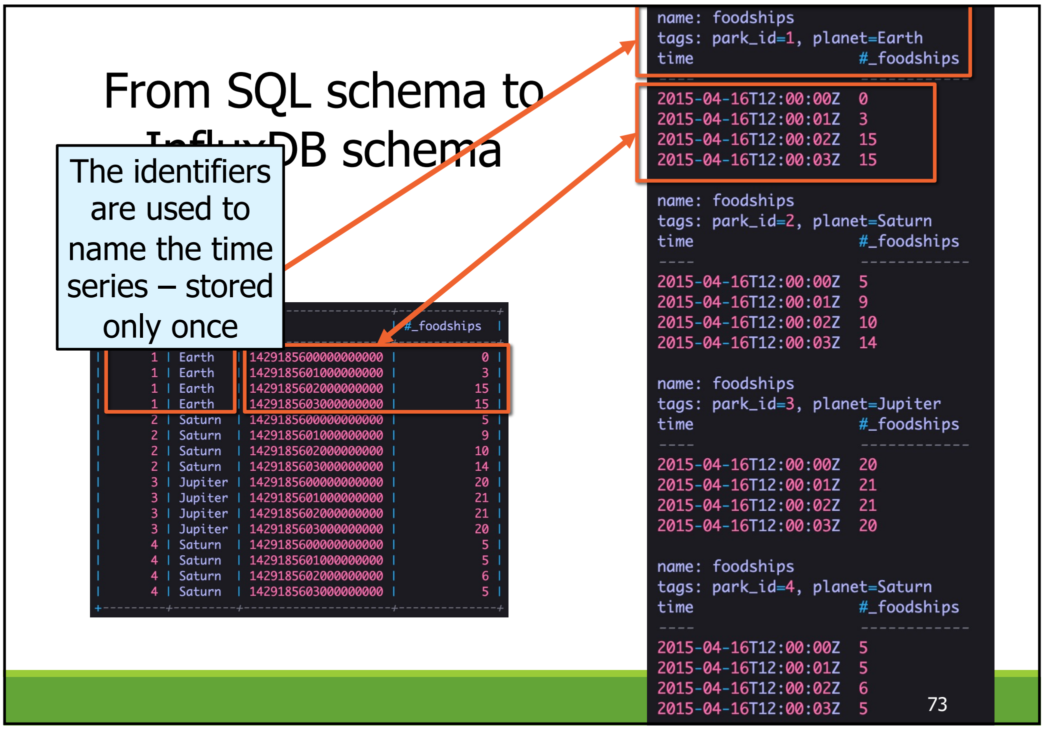
park_id	planet	time	#_foodships
1	Earth	1429185600000000000	0
1	Earth	1429185601000000000	3
1	Earth	1429185602000000000	15
1	Earth	1429185603000000000	15
2	Saturn	1429185600000000000	5
2	Saturn	1429185601000000000	9
2	Saturn	1429185602000000000	10
2	Saturn	1429185603000000000	14
3	Jupiter	1429185600000000000	20
3	Jupiter	1429185601000000000	21
3	Jupiter	1429185602000000000	21
3	Jupiter	1429185603000000000	20
4	Saturn	1429185600000000000	5
4	Saturn	1429185601000000000	5
4	Saturn	1429185602000000000	6
4	Saturn	1429185603000000000	5

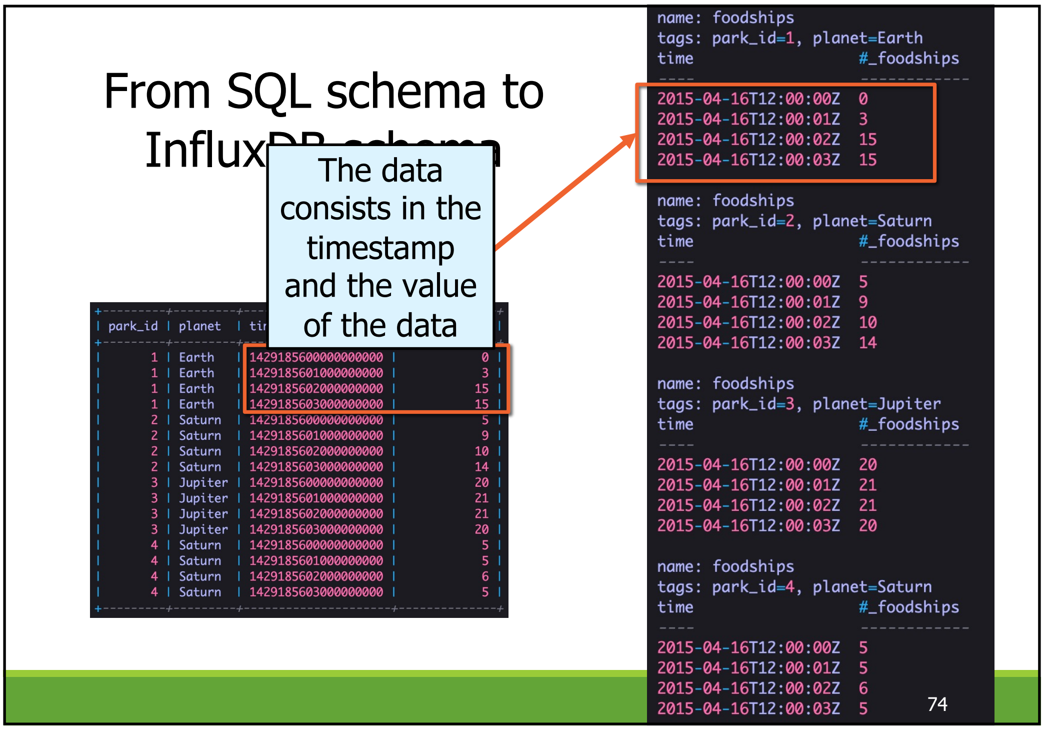
```
name: foodships
tags: park_id=1, planet=Earth
time                  #_foodships
----                  -----------
2015-04-16T12:00:00Z  0
2015-04-16T12:00:01Z  3
2015-04-16T12:00:02Z  15
2015-04-16T12:00:03Z  15

name: foodships
tags: park_id=2, planet=Saturn
time                  #_foodships
----                  -----------
2015-04-16T12:00:00Z  5
2015-04-16T12:00:01Z  9
2015-04-16T12:00:02Z  10
2015-04-16T12:00:03Z  14

name: foodships
tags: park_id=3, planet=Jupiter
time                  #_foodships
----                  -----------
2015-04-16T12:00:00Z  20
2015-04-16T12:00:01Z  21
2015-04-16T12:00:02Z  21
2015-04-16T12:00:03Z  20

name: foodships
tags: park_id=4, planet=Saturn
time                  #_foodships
----                  -----------
2015-04-16T12:00:00Z  5
2015-04-16T12:00:01Z  5
2015-04-16T12:00:02Z  6
2015-04-16T12:00:03Z  5
```

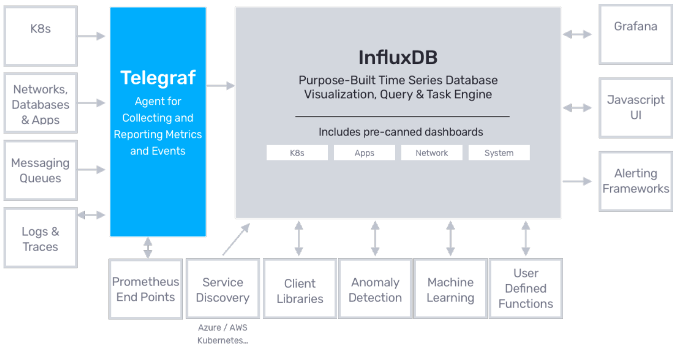

From SQL schema to InfluxDB schema
The identifiers are used to name the time series – stored only once
#_foodships
1 | Earth | 1429185600000000000 | 0
1 | Earth | 1429185601000000000 | 3
1 | Earth | 1429185602000000000 | 15
1 | Earth | 1429185603000000000 | 15
2 | Saturn | 1429185600000000000 | 5
2 | Saturn | 1429185601000000000 | 9
2 | Saturn | 1429185602000000000 | 10
2 | Saturn | 1429185603000000000 | 14
3 | Jupiter | 1429185600000000000 | 20
3 | Jupiter | 1429185601000000000 | 21
3 | Jupiter | 1429185602000000000 | 21
3 | Jupiter | 1429185603000000000 | 20
4 | Saturn | 1429185600000000000 | 5
4 | Saturn | 1429185601000000000 | 5
4 | Saturn | 1429185602000000000 | 6
4 | Saturn | 1429185603000000000 | 5
name: foodships
tags: park_id=1, planet=Earth
time #_foodships
2015-04-16T12:00:00Z 0
2015-04-16T12:00:01Z 3
2015-04-16T12:00:02Z 15
2015-04-16T12:00:03Z 15
name: foodships
tags: park_id=2, planet=Saturn
time #_foodships
2015-04-16T12:00:00Z 5
2015-04-16T12:00:01Z 9
2015-04-16T12:00:02Z 10
2015-04-16T12:00:03Z 14
name: foodships
tags: park_id=3, planet=Jupiter
time #_foodships
2015-04-16T12:00:00Z 20
2015-04-16T12:00:01Z 21
2015-04-16T12:00:02Z 21
2015-04-16T12:00:03Z 20
name: foodships
tags: park_id=4, planet=Saturn
time #_foodships
2015-04-16T12:00:00Z 5
2015-04-16T12:00:01Z 5
2015-04-16T12:00:02Z 6
2015-04-16T12:00:03Z 5
73


From SQL schema to InfluxDB schema
The data consists in the timestamp and the value of the data
park_id | planet
1 | Earth | 1429185600000000000 | 0
1 | Earth | 1429185601000000000 | 3
1 | Earth | 1429185602000000000 | 15
1 | Earth | 1429185603000000000 | 15
2 | Saturn | 1429185600000000000 | 5
2 | Saturn | 1429185601000000000 | 9
2 | Saturn | 1429185602000000000 | 10
2 | Saturn | 1429185603000000000 | 14
3 | Jupiter | 1429185600000000000 | 20
3 | Jupiter | 1429185601000000000 | 21
3 | Jupiter | 1429185602000000000 | 21
3 | Jupiter | 1429185603000000000 | 20
4 | Saturn | 1429185600000000000 | 5
4 | Saturn | 1429185601000000000 | 5
4 | Saturn | 1429185602000000000 | 6
4 | Saturn | 1429185603000000000 | 5
name: foodships
tags: park_id=1, planet=Earth
time #_foodships
2015-04-16T12:00:00Z 0
2015-04-16T12:00:01Z 3
2015-04-16T12:00:02Z 15
2015-04-16T12:00:03Z 15
name: foodships
tags: park_id=2, planet=Saturn
time #_foodships
2015-04-16T12:00:00Z 5
2015-04-16T12:00:01Z 9
2015-04-16T12:00:02Z 10
2015-04-16T12:00:03Z 14
name: foodships
tags: park_id=3, planet=Jupiter
time #_foodships
2015-04-16T12:00:00Z 20
2015-04-16T12:00:01Z 21
2015-04-16T12:00:02Z 21
2015-04-16T12:00:03Z 20
name: foodships
tags: park_id=4, planet=Saturn
time #_foodships
2015-04-16T12:00:00Z 5
2015-04-16T12:00:01Z 5
2015-04-16T12:00:02Z 6
2015-04-16T12:00:03Z 5
74

Design of a TSDB

Challenge 3: need to import data from multiple source.

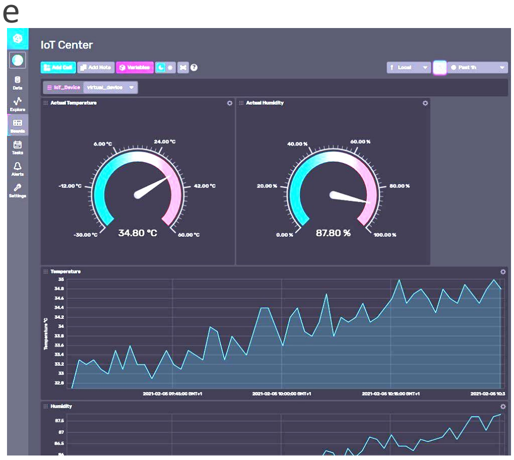
Solution: Open protocols – e.g. MQTT – and connection to multiple sources.



Design of a TSDB

Challenge 4: need to provide information to users/operators.

Solution: Rich dashboards.



Bibliography

"Internet of Things: A Survey on Enabling Technologies, Protocols and Applications", Ala Al-Fuqaha, et al., *IEEE Communications Surveys & Tutorials, 2015.* https://ieeexplore.ieee.org/document/7123563

"The Emergence of Edge Computing", Mahadev Satyanarayanan, Computer IEEE, 2017. https://ieeexplore.ieee.org/document/7807196

"Mobile cloud computing: A survey", Niroshinie Fernando, et al., Future Generation Computer Systems, 2014. https://link.springer.com/article/10.1007/s11036-013-0477-4 (only the topics on the slides)

Gorilla: A Fast, Scalable, In-Memory Time Series
- https://www.vldb.org/pvldb/vol8/p1816-teller.pdf