The Internet of Things: Sensor Data Management

Course website: http://site.unibo.it/iot

Prof. Luciano Bononi
luciano.bononi@unibo.it

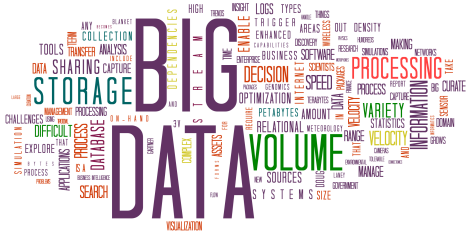
Prof. Marco Di Felice
marco.difelice3@unibo.it

MASTER DEGREE IN COMPUTER SCIENCE
DEPARTMENT OF COMPUTER SCIENCE AND ENGINEERING, UNIVERSITY OF BOLOGNA, ITALY

IoT Sensor Data Management

IoT Sensor Data Management

The word big-data is currently used to identify: (*i*) **data-sources** with specific characteristics, as well as (*ii*) **novel technologies** to manage the data.



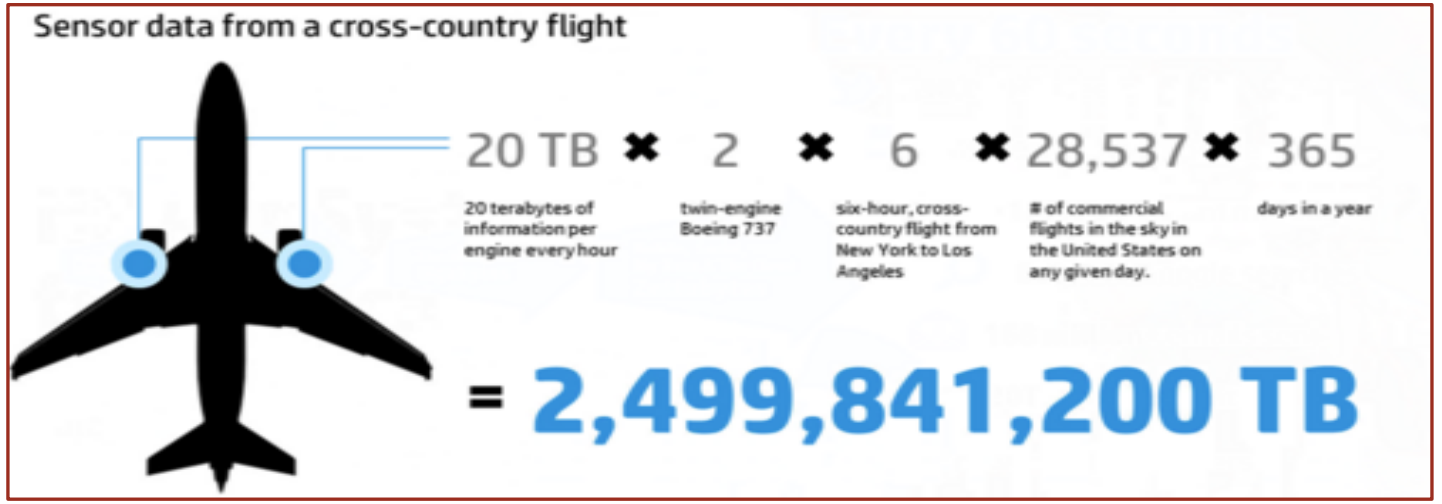
CHARACTERISTICS OF BIG-DATA

- Cannot be managed using conventional technologies of information systems
- Volume → order of Petabytes
- Velocity → data produced at high rate
- Variety → heterogeneous data (text, image, video, etc)
- Value → valuable information can be extracted

IoT Sensor Data Management

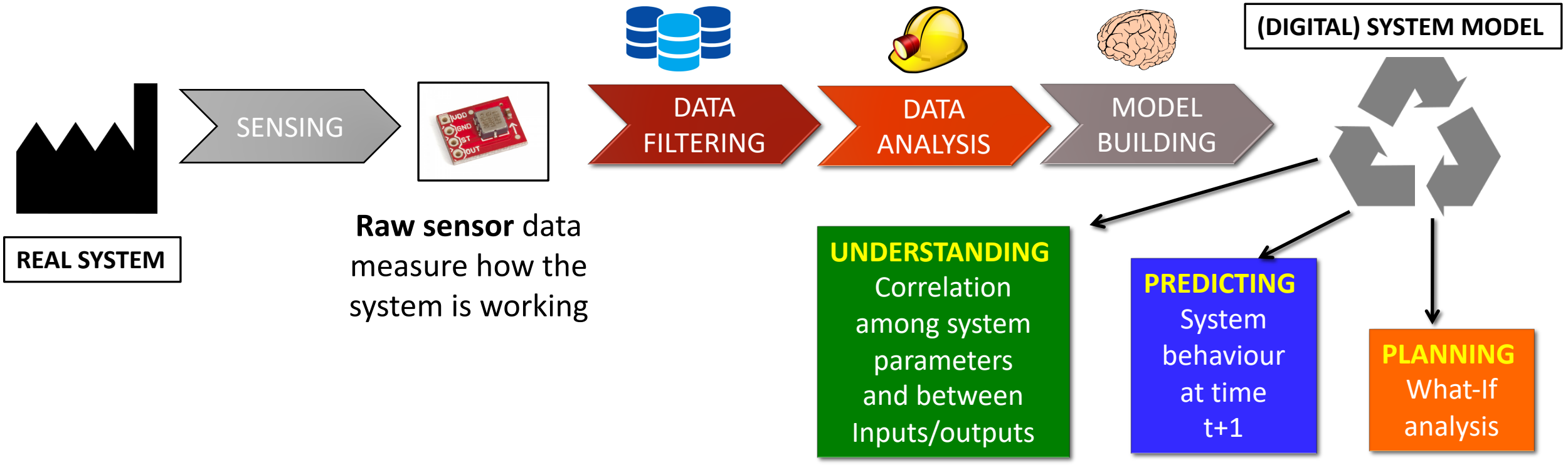
✧**IoT** & **Big-data** are two sides of the same coin!

- ❑ Large-scale IoT deployments can produce huge amounts of data
- ❑ IoT is about data, services, connectivity: data are gathered by objects, transfered, analyzed, and traslated into services



IoT Sensor Data Management

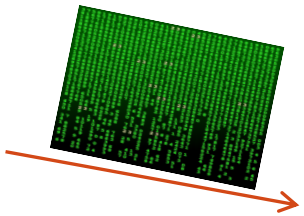
✧**IoT** & **Big-data** are two sides of the same coin!



IoT Sensor Data Management

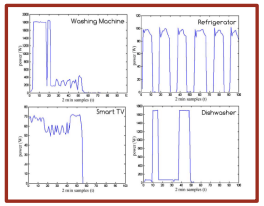
✧**IoT** & **Big-data** are two sides of the same coin!

EXAMPLE: **ENERGY@HOME TELECOM ITALIA (research project)**
http://www.energy-home.it



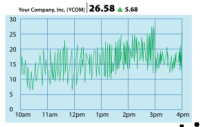
RAW **SENSOR DATA** TRACES

Classification
(i.e. identify the appliance)



Profiling
(i.e. identify users' habbits)

Predicting
(i.e. predict energy consumption)

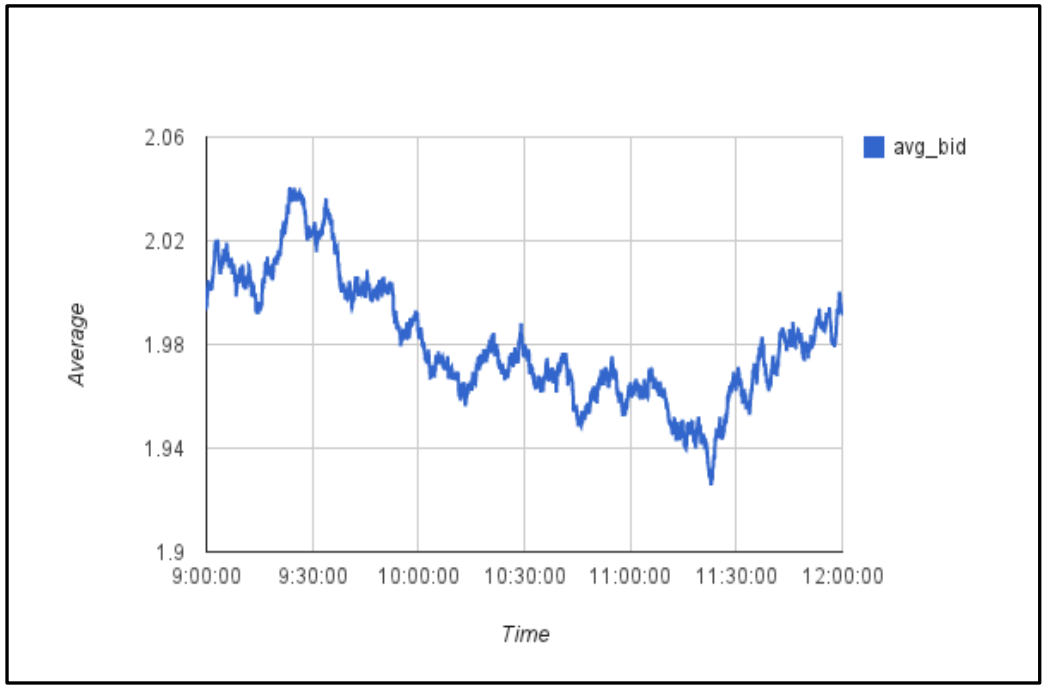


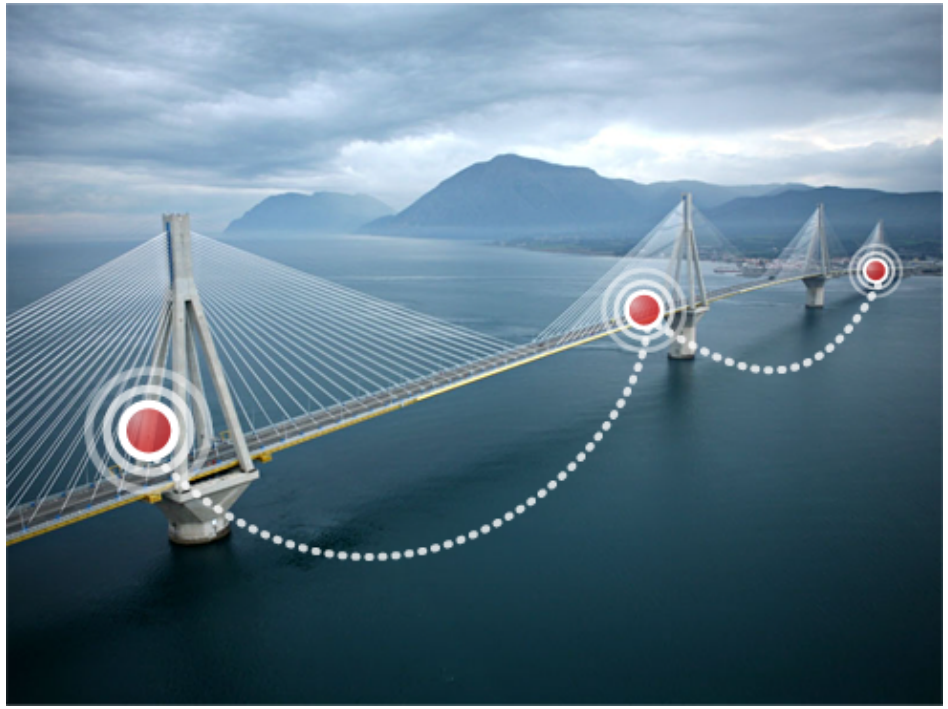
Scheduling
(i.e. intelligent ON/OFF schedule)

- ✧ 5 **smart plugs** for house
- ✧ Around 200 installations in private houses
- ✧ 1 sampling every 2 minutes

IoT Sensor Data Management

✧Time-series → Sequence of timestamp plus values





IoT Sensor Data Management

✧**Time-series** → **Sequence of timestamp plus values**

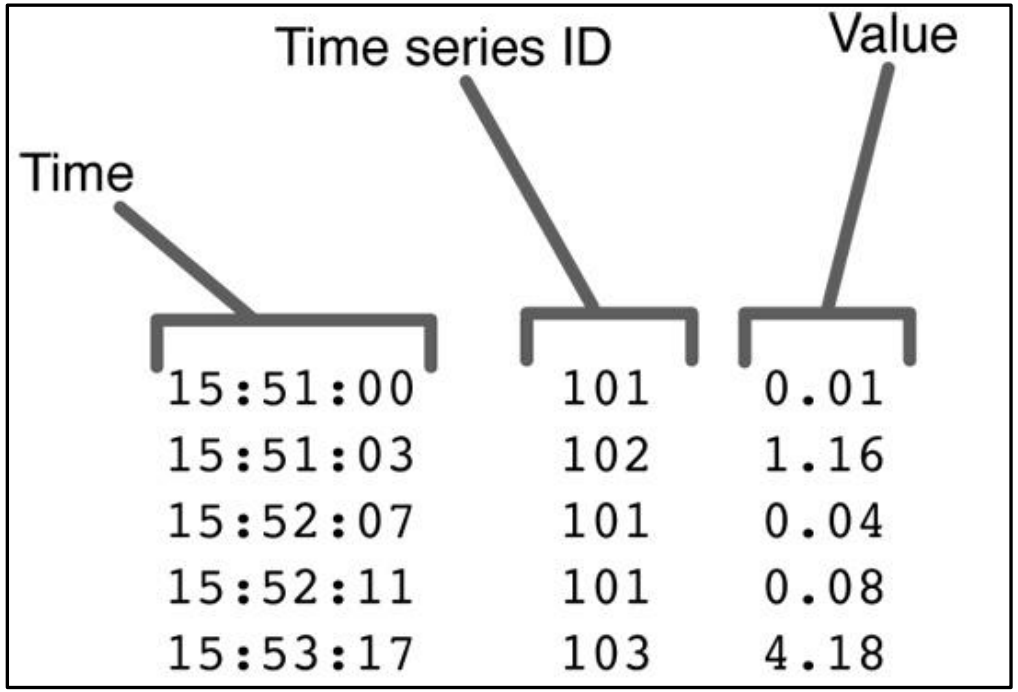
- ❑ Data are **immutable**.
- ❑ Writing in **append**.
- ❑ Reading **contiguous sequence** of samples data.
- ❑ Highly **compressible data**.
- ❑ **Deleting** usually across large time period
- ❑ **High precision** for short period of time.
- ❑ Single value is not **so important**

IoT Sensor Data Management

- **Relational Database Management Systems (RDBMS)**
  - Based on the relational model first proposed by Edgar F. Codd (1970)
  - Employ **SQL** language
  - Support **ACID** properties
  - Scheme-based (structured) database.
  - Components: Tables (relations), primary keys, foreign keys, NULL values.

IoT Sensor Data Management

✧ Time-series implemented on RDMBS



PROBLEMS

- ✧ **Scalability** (i.e. need to store large amount of time-series data)
- ✧ **Performance** (e.g. support for range-based operations)
- ✧ Aymmetric **CRUD** operations

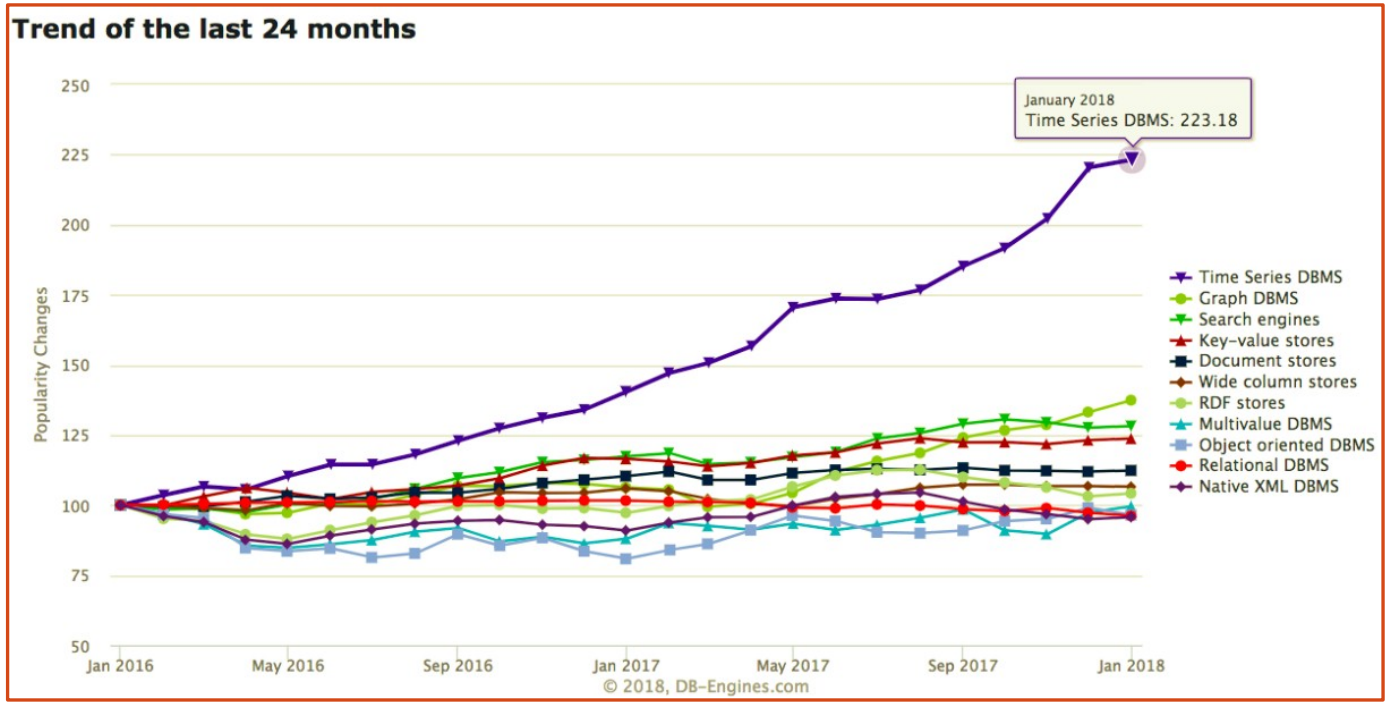
IoT Sensor Data Management

✧ NOSQL Database Management Systems

- ❑ Set of tools and logic models, alternative or complementary to RDBMS (noREL).
- ❑ Support **BASE** properties.
- ❑ Do not employ the SQL language.
- ❑ Scheme-less (un/semi-structured) database.
- ❑ Families: Key-values DB, Document-based DB, Column-based DB, Graph-based DB.

IoT Sensor Data Management

✧ Time-series Database



RANK	DBMS	SCORE	
APR. 2018		APR. 2018	APR. 2016
1	InfluxDB	10.76	+6.84
2	Kdb+	3.08	+1.87
3	RRDtool	2.75	+0.22
4	Graphite	2.19	+0.63
5	OpenTSDB	1.70	+0.29
6	Druid	1.06	+0.83
7	Prometheus	1.06	+0.91
8	KairosDB	0.43	+0.24
9	eXtremeDB	0.33	+0.13
10	Riak TS	0.27	+0.25

Source: DB-Engines 23 Systems in Ranking, April 2018

IoT Sensor Data Management

✧ **Time-series Database** → Dedicated DBMS optimized for managing large volumes of time-series data

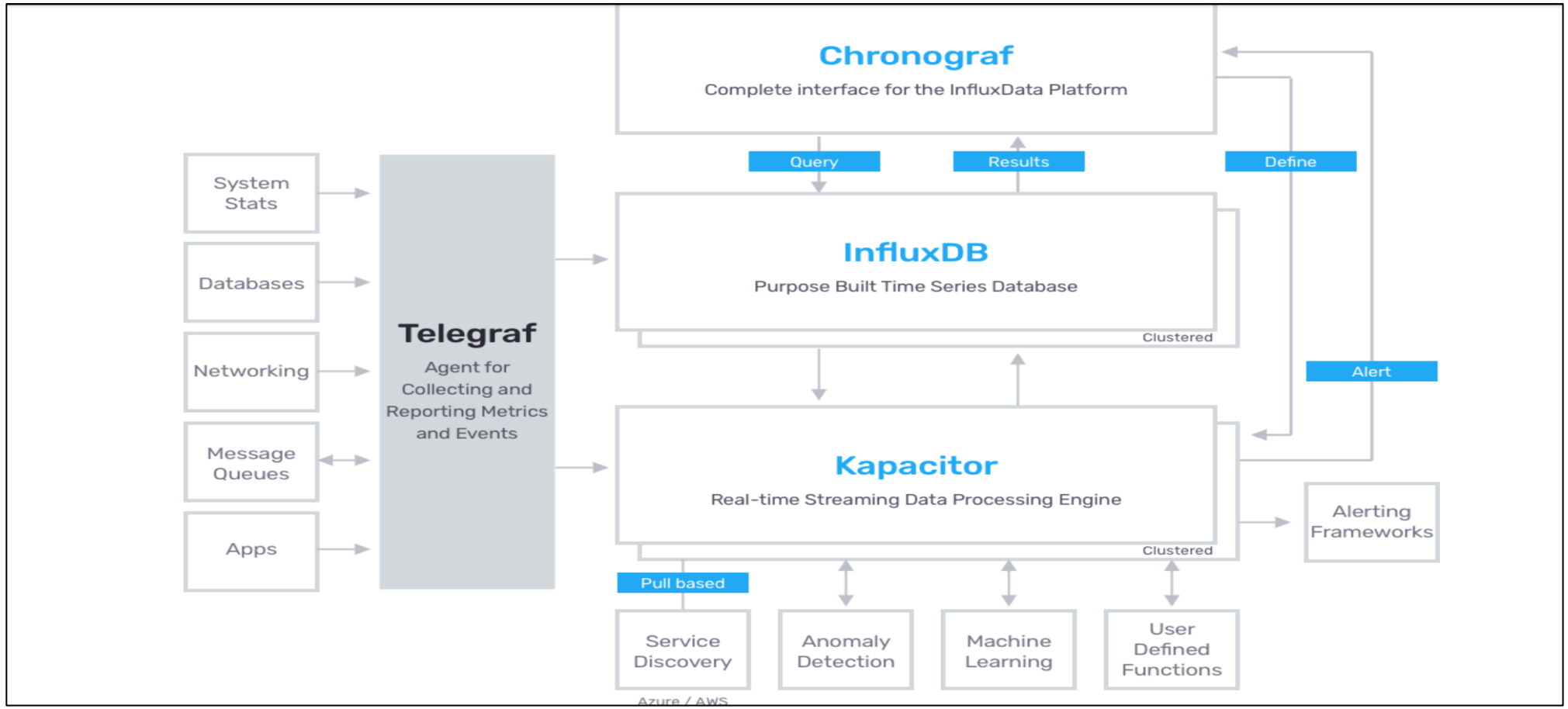
- ❑ Optimized data storage and sharding
- ❑ Operational support (e.g. range-based queries)
- ❑ Time-granularity management
- ❑ Time-series analytics and mining

IoT Sensor Data Management

✧ InfluxDB (https://www.influxdata.com)

- ❑ Open-source time-series database (InfluxData)
- ❑ Written in **GO** language
- ❑ SQL-like query language (InfluxQL) (<1.8)
- ❑ JS-like query language (Flux) (>=1.8)
- ❑ GUI, Command Line Interface (CLI) and HTTP APIs
- ❑ Support for distributed deployments
- ❑ Integration with time-series tools for data acquisition, analytics and visualization (e.g. Telegraf, Grafana)

IoT Sensor Data Management



IoT Sensor Data Management

✧ InfluxDB (https://www.influxdata.com)

- ❑ Time-Structured Merge Tree (TSM) → data structure used to contain sorted, compressed series data.
- ❑ Time Series Index (TSI) → address millions of unique time series, regardless of the amount of memory on the server hardware.
- ❑ Automatic downsampling and data rentention procedures.

IoT Sensor Data Management

✧ InfluxDB (https://www.influxdata.com)

- Timestamp → RFC3339 UTC format (yyyy-mm-ddThh:mm:ssZ)
- Field keys → string metadata, similar to column name
- Field values → actual measured data
- Tag-sets → optimal, extra-information about the measurements
  - **Tag** keys → string meta-data, similar to field keys
  - **Tag** values → string values
- **Measurement** → Container to hold the timestamps, fields and tags (similar to a table in a RDBMS)

Source: https://www.linkedin.com/pulse/introduction-influxdb-kristof-bruylants

IoT Sensor Data Management

✧ InfluxDB (https://www.influxdata.com)

- **Buckets** → collection of data-points, containing:
  - **Measurement**
  - **Tag-sets**
  - **Retention policy** → period that datapoints are being stored in InfluxDB, which is called DURATION but also the number of versions that should be kept on the cluster, as REPLICATION.

Source: https://www.linkedin.com/pulse/introduction-influxdb-kristof-bruylants

IoT Sensor Data Management

✧ InfluxDB (https://www.influxdata.com)

```
series          retention          measurement     tag set
number          policy
------          ------             ------          ------
series 1        default            census          location = 1, scientist = smith
series 2        default            census          location = 2, scientist = smith
series 3        default            census          location = 1, scientist = jones
series 4        default            census          location = 2, scientist = jones
```

```
name: census
-----------------
time                    butterflies   honeybees     location      scientist
2015-08-18T00:00:00Z    1             30            1             jones
```

Source: https://www.linkedin.com/pulse/introduction-influxdb-kristof-bruylants

IoT Sensor Data Management

✧ InfluxDB (https://www.influxdata.com)

❑ Command-Line Interface (CLI)

```
user@mypc:$influx
Connected to http://localhost:8086 version 1.5.2
InfluxDB shell version: 1.5.2
```

❑ HTTP-based API

```
curl -i -XPOST http://localhost:8086/query --data-urlencode "q=CREATE
DATABASE mydb”
curl -i -XPOST 'http://localhost:8086/write?db=mydb' --data-binary
'cpu_load_short,host=server01,region=us-west value=0.64 1434055562000000000'
```

Source: https://docs.influxdata.com/influxdb/v1.5/tools/shell/#write-data-to-influxdb-with-insert

IoT Sensor Data Management

✧ InfluxDB (https://www.influxdata.com)

❑ Basic Commands (InfluxQL)

- ✧ `CREATE DATABASE name`
- ✧ `USE DATABASE name`
- ✧ `SHOW DATABASES`
- ✧ `CLEAR DATABASE name`
- ✧ `CREATE RETENTION POLICY name ON measurement DURATION 1d REPLICATION 1`
- ✧ `INSERT treasures,captain_id=pirate_king value=2`

Source: https://docs.influxdata.com/influxdb/v1.5/query_language/data_exploration/

IoT Sensor Data Management

✧ InfluxDB (https://www.influxdata.com)

❑ InfluxQL language for data exploration

```
SELECT <field_key>[,<field_key>,<tag_key>] FROM <measurement_name>[,<measurement_name>]
```

```
SELECT * FROM "h2o_feet"
```

```
SELECT_clause FROM_clause WHERE <conditional_expression> [(AND|OR) <conditional_expression> [...]]
```

```
SELECT * FROM "h2o_feet" WHERE "water_level" > 8
```

```
SELECT_clause FROM_clause [WHERE_clause] GROUP BY [* | <tag_key>[,<tag_key]]
```

```
SELECT MEAN("water_level") FROM "h2o_feet" GROUP BY "location"
```

Source: https://docs.influxdata.com/influxdb/v1.5/query_language/data_exploration/

IoT Sensor Data Management

✧ InfluxDB (https://www.influxdata.com)

❑ FLUX language for data exploration (Influx 2.0)

- ✓ Functional approach to data exploration and processing
- ✓ **Inspired by Javascript language**
- ✓ The pipe-forward operator (**|>**) allow to chain multiple data operations
- ✓ Each data operation manipulates a table and produces a new table as output

Source: https://docs.influxdata.com/influxdb/v1.5/query_language/data_exploration/

IoT Sensor Data Management

✧ InfluxDB (https://www.influxdata.com)

❑ FLUX language for data exploration (Influx 2.0)

```
from(bucket:"example-bucket")
|> range(start:-1h)
```

Query sul bucket «example-bucket», seleziona tutte le time-series create nell'ultima ora

Source: https://docs.influxdata.com/influxdb/v1.5/query_language/data_exploration/

IoT Sensor Data Management

✧ InfluxDB (https://www.influxdata.com)

❑ FLUX language for data exploration (Influx 2.0)

```
from(bucket:"example-bucket")
|> range(start:-1h)
|> r._measurement == «temperature»
   and r.room == «kitchen»
```

Query sul bucket «example-bucket», seleziona tutte le time-series create nell'ultima ora, relative alla measurement «temperature» e dove il tag «room» è pari a «kitchen»

Source: https://docs.influxdata.com/influxdb/v1.5/query_language/data_exploration/

IoT Sensor Data Management

✧ InfluxDB (https://www.influxdata.com)

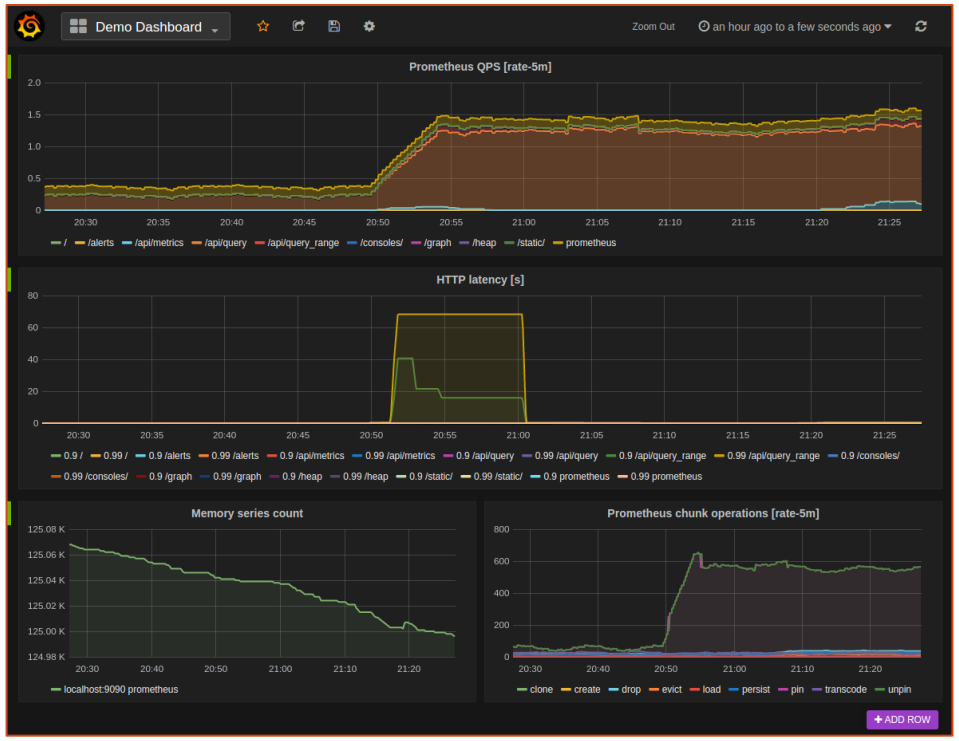
❑ FLUX language for data exploration (Influx 2.0)

```
from(bucket:"example-bucket")
|> range(start:-1h)
|> r._measurement == «temperature»
   and r.room == «kitchen»
|>aggregateWindow(every: 1m, fn: mean)
```

Query sul bucket «example-bucket», seleziona tutte le time-series create nell'ultima ora, relative alla measurement «temperature» e dove il tag «room» è pari a «kitchen». Aggrega i dati ogni minuto, calcolando la media

IoT Sensor Data Management

✧ Grafana (https://grafana.com)



Source: https://docs.influxdata.com/influxdb/v1.5/query_language/data_exploration/

IoT Sensor Data Management

✧ InfluxDB (https://www.influxdata.com)

❑ FLUX Query in Grafana

```
from(bucket: «XXX»)
|>range(start: -ZZZh)
|>filter(fn: (r) => r["_measurement"] == »YYY")
```