www.cern.ch

Data challenges with CERN Technical Infrastructure Monitoring

Matthias Bräger, *CERN*

mbraeger@cern.ch

7th March 2016

BTW 2017, Stuttgart, Germany

About me

- Working at CERN since Dec. 2007
- Responsible for Technical Infrastructure Monitoring (TIM) service at CERN
- Head of the CERN Control and Monitoring Platform (C2MON): http://cern.ch/c2mon

Before CERN:

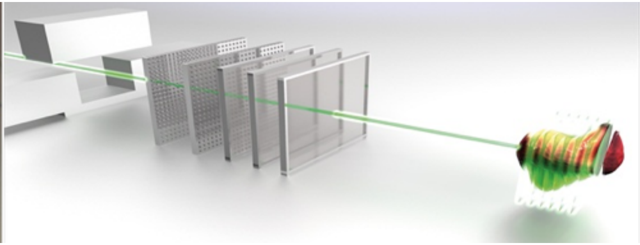
- 2 years at LOGICA space department for ESOC, Darmstadt, Germany
- 4 years Java developer at IBM, Mainz, Germany

European Organization for Nuclear Research

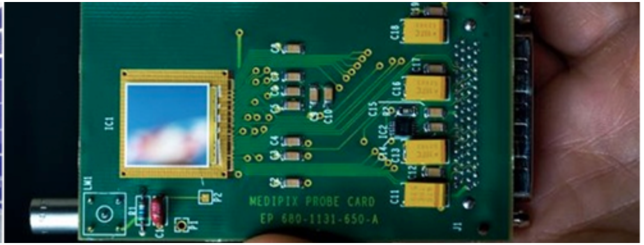
- Founded in 1954 (60 years ago!)
- 21 Member States
- ~ 3'360 Staff, fellows, students...
- ~ 10'000 Scientists from
  113 different countries
- Budget: 1 billion CHF/year

http://cern.ch

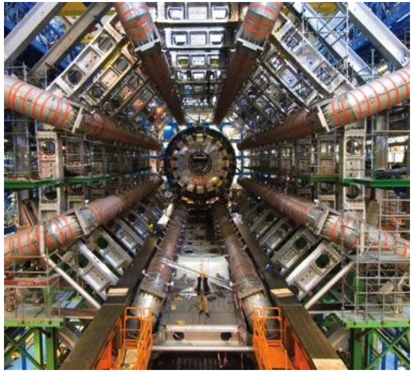
From Physics to Industry



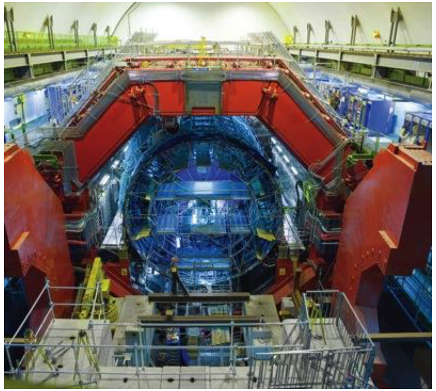


http://cern.ch/knowledgetransfer



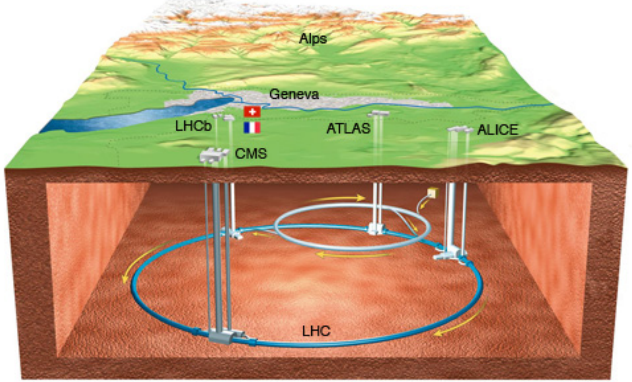
ATLAS

Alice



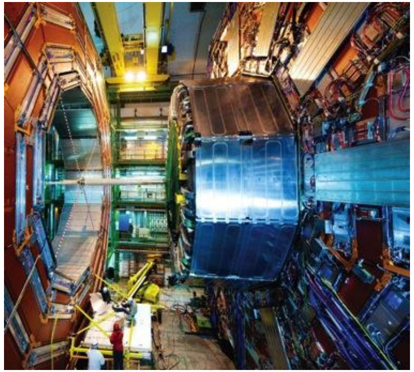
LHC

The worlds biggest machine



Generated 30 Petabytes in 2012

> 100 PB in total!



CMS

LHCb

LHC - Large Hadron Collider

27km ring of superconducting magnets

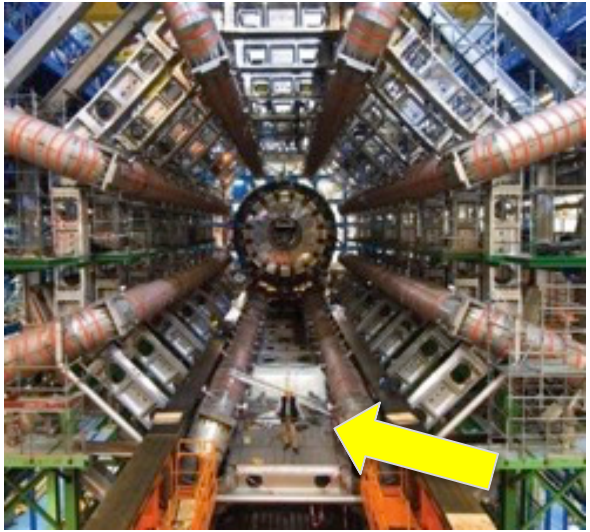
Started operation in 2010 with 3.5 + 3.5 TeV,
4 + 4 TeV in 2012

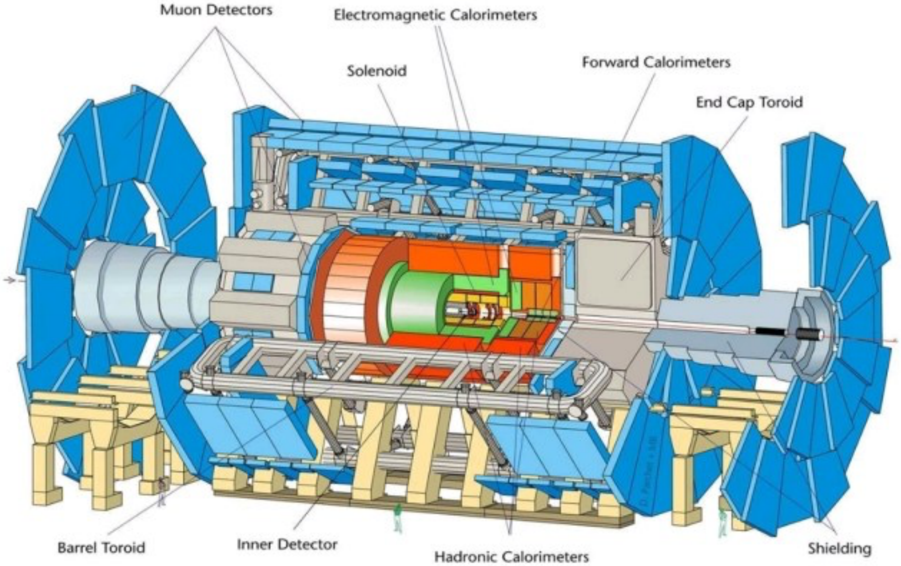
2013 – 2015 in Long Shutdown 1
(machine upgrade)

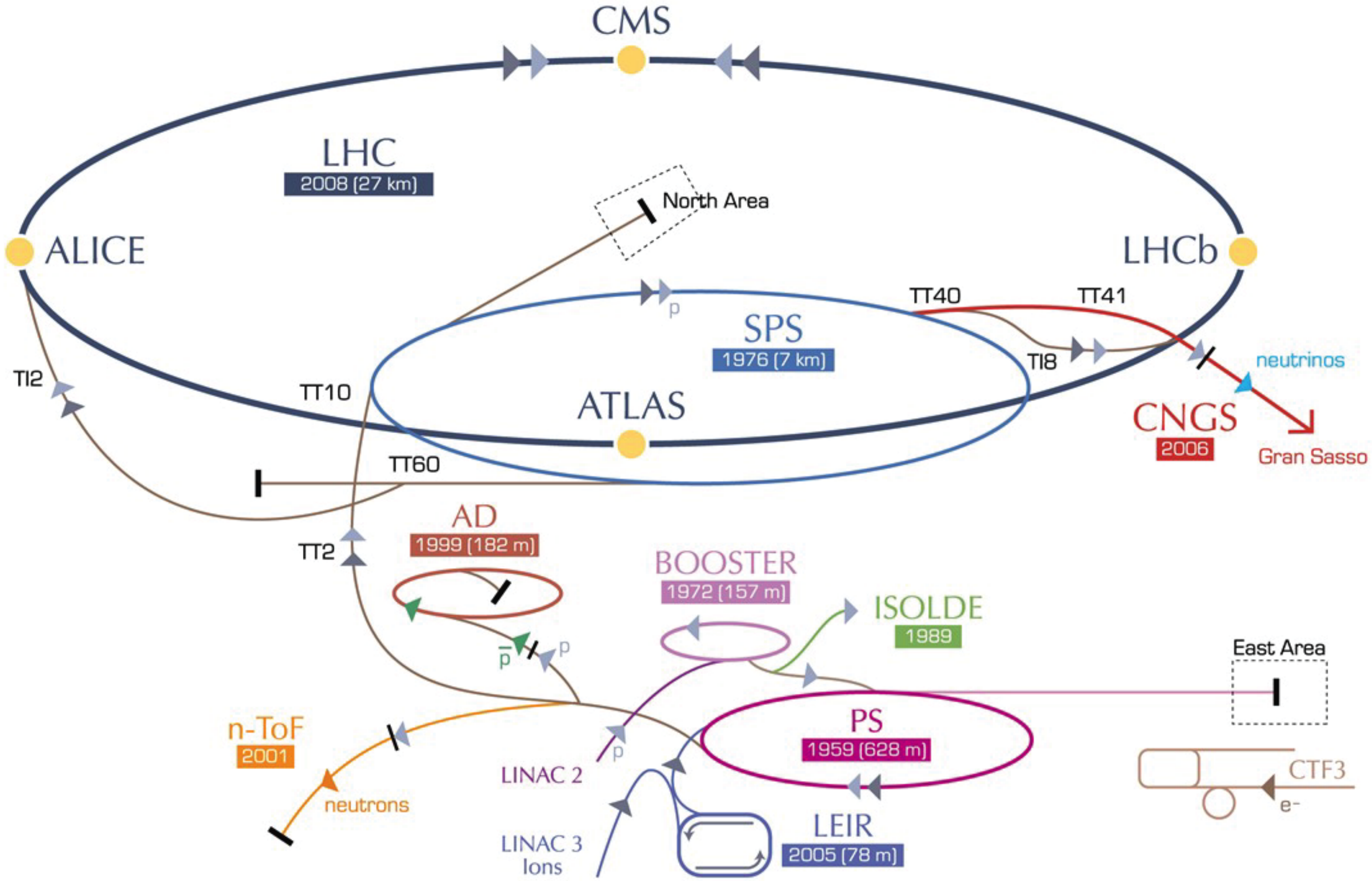
Restarted in April 2015 with 6.5 + 6.5 TeV max

Some ATLAS facts

- 25m diameter, 46m length, 7'000 tons
- 100 million channels
- 40MHz collision rate (~ 1 PB/s)






CMS
LHC
2008 (27 km)
North Area
ALICE
LHCb
TT40
TT41
p
SPS
1976 (7 km)
TI8
neutrinos
TI2
TT10
ATLAS
CNGS
2006
Gran Sasso
TT60
AD
1999 (182 m)
TT2
BOOSTER
1972 (157 m)
ISOLDE
1989
p̄
p
East Area
n-ToF
2001
PS
1959 (628 m)
LINAC 2
p
CTF3
e–
neutrons
LINAC 3
Ions
LEIR
2005 (78 m)

Log data

Metadata of physics data

Configuration data

Physics data (>100 PB)

Documents

Sensor Data of technical installations

Media data

Others

Is Hadoop used for storing the ~30 PB/year of physics data ?

No ;-(

Experimental data are mainly stored on tape

CERN uses **Hadoop** e.g. for storing the metadata of the experimental data

Physics Data Handling

- Run 1: 30 PB per year
  demanding 100'000 processors
  with peaks of 6 GB/s writing to tape
  spread across 80 tape drives

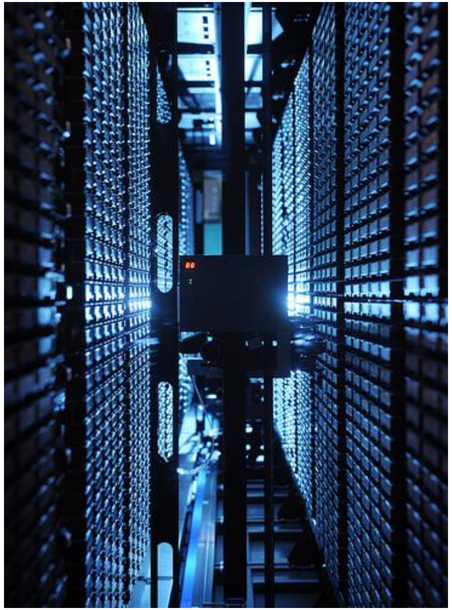
- Run 2: > 50 PB per year
  with peaks of 10 GB/s writing to tape

CERN's Computer Center (1st floor)

Physics Data Handling

2013 already more than 100 PB stored in total!

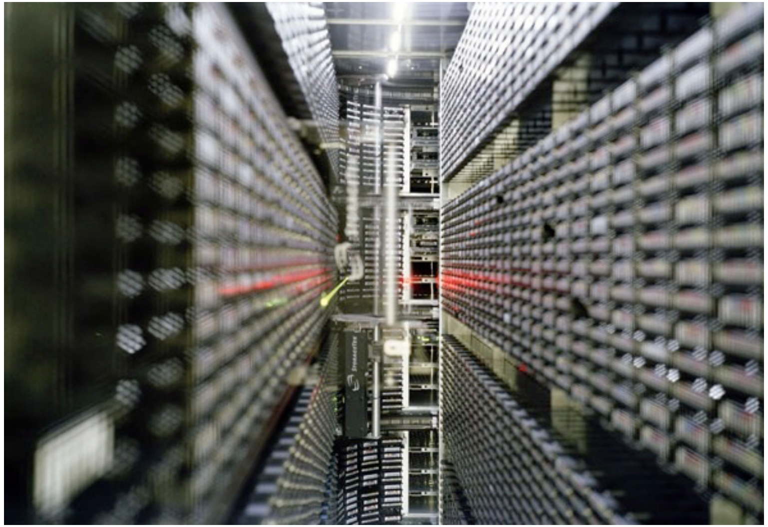
- > 88 PB on 55'000 tapes
- > 13 PB on disk (total disk space 45 PB)
- > 140 PB free tape storage waiting for Run 2



CERN's tape robot

Why tape storage?

- Cost of tape storage is a lot less than disk storage
- No electricity consumption when tapes are not being accessed
- Tape storage size = Data + Copy
  Hadoop storage size = Data + 2 Copies
- No requirement to have all recorded physics data available within seconds



CERN's tape robot

APACHE HBASE @ CERN

3 HBase Clusters

- CASTOR Cluster with ~10 servers
  - ~ 100 GB of Logs per day
  - > 120 TB of Logs in total
- ATLAS Cluster with ~20 servers
  - Event index Catalogue for experimental Data in the Grid
- Monitoring Cluster with ~10 servers
  - Log events from CERN Computer Center

CERN

Metadata from physics event

Metadata are created upon recording of the physics event

<u>Examples 1</u>:

- Tape Storage event log
  - On which tape is my file stored?
  - Is there a copy on disk?
  - List me all events for a given tape or drive
  - Was the tape repacked?

Metadata from physics event

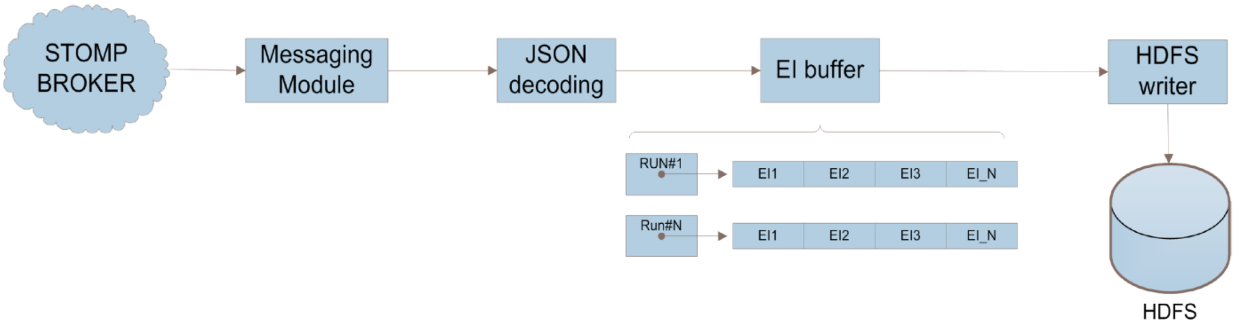
Metadata are created upon recording of the physics event

Examples 2:

- Information about
  - Event number
  - run number
  - timestamp
  - luminosity block number
  - trigger that selected the event, etc.

Example 2: ATLAS EventIndex catalogue

- In 2011 and 2012, ATLAS produced 2 billion real events and 4 billion simulated events



Data are read from the brokers, decoded and stored into Hadoop.

Example 2: ATLAS EventIndex catalogue

The major use cases of the EventIndex project are:

- **Event picking:**
  give me the reference (pointer) to "this" event in "that" format for a given processing cycle.
- **Production consistency checks:**
  technical checks that processing cycles are complete (event counts match).
- **Event service:**
  give me the references (pointers) for “this" list of events, or for the events satisfying given selection criteria

Log data

Metadata of physics data

Configuration data

Physics data (>100 PB)

Documents

Sensor Data of technical installations

Media data

Others

A lot of systems to control and data to store

Controls
Computers

Electricity

Cryogenics

Magnets

85'000 Devices
> 2 Million I/O Endpoints

***Much more** when*
including subsystems!

Safety

Cooling

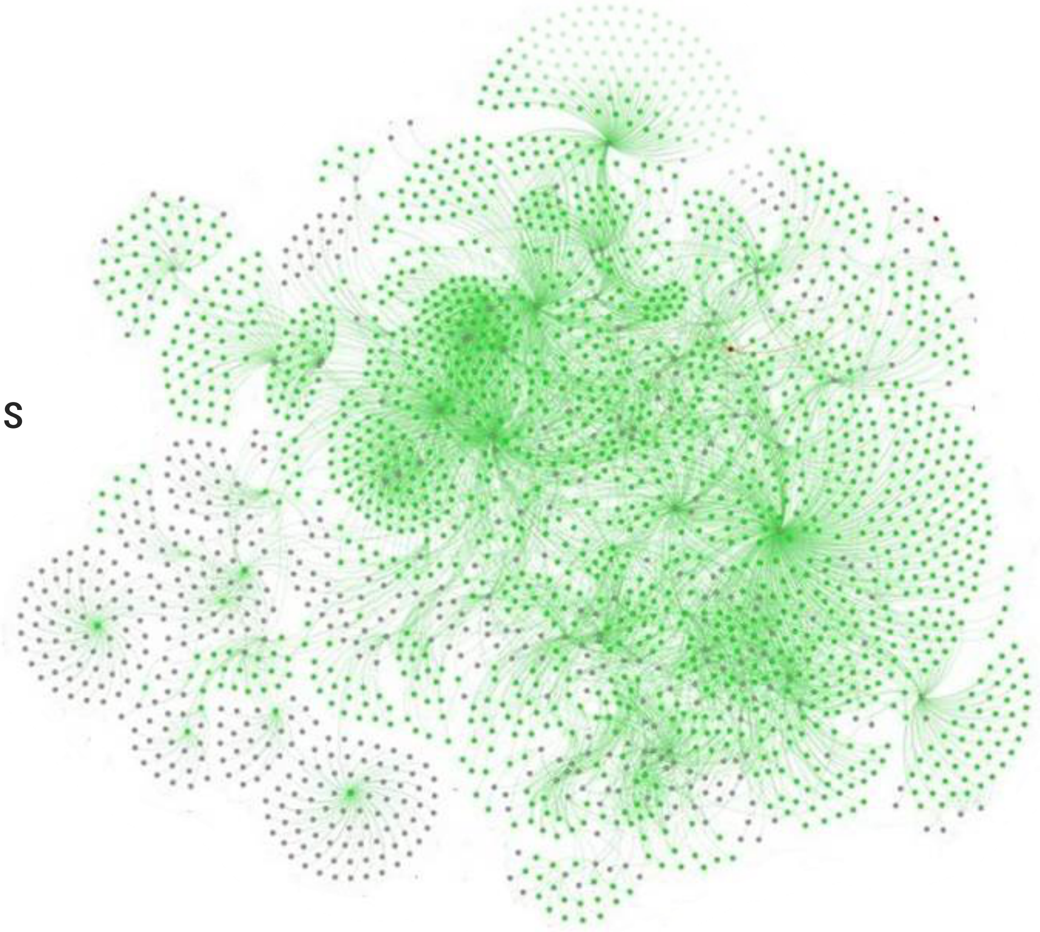
Ventilation

Vacuum

Main systems controlled from one central point: The CERN Control Centre

A complex controls infrastructure...

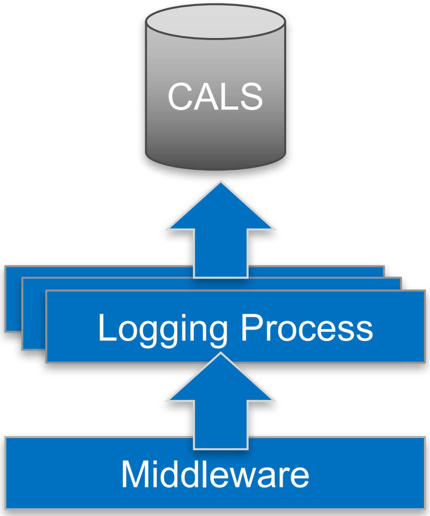
- Each dot is a process
- Each line is a network connections



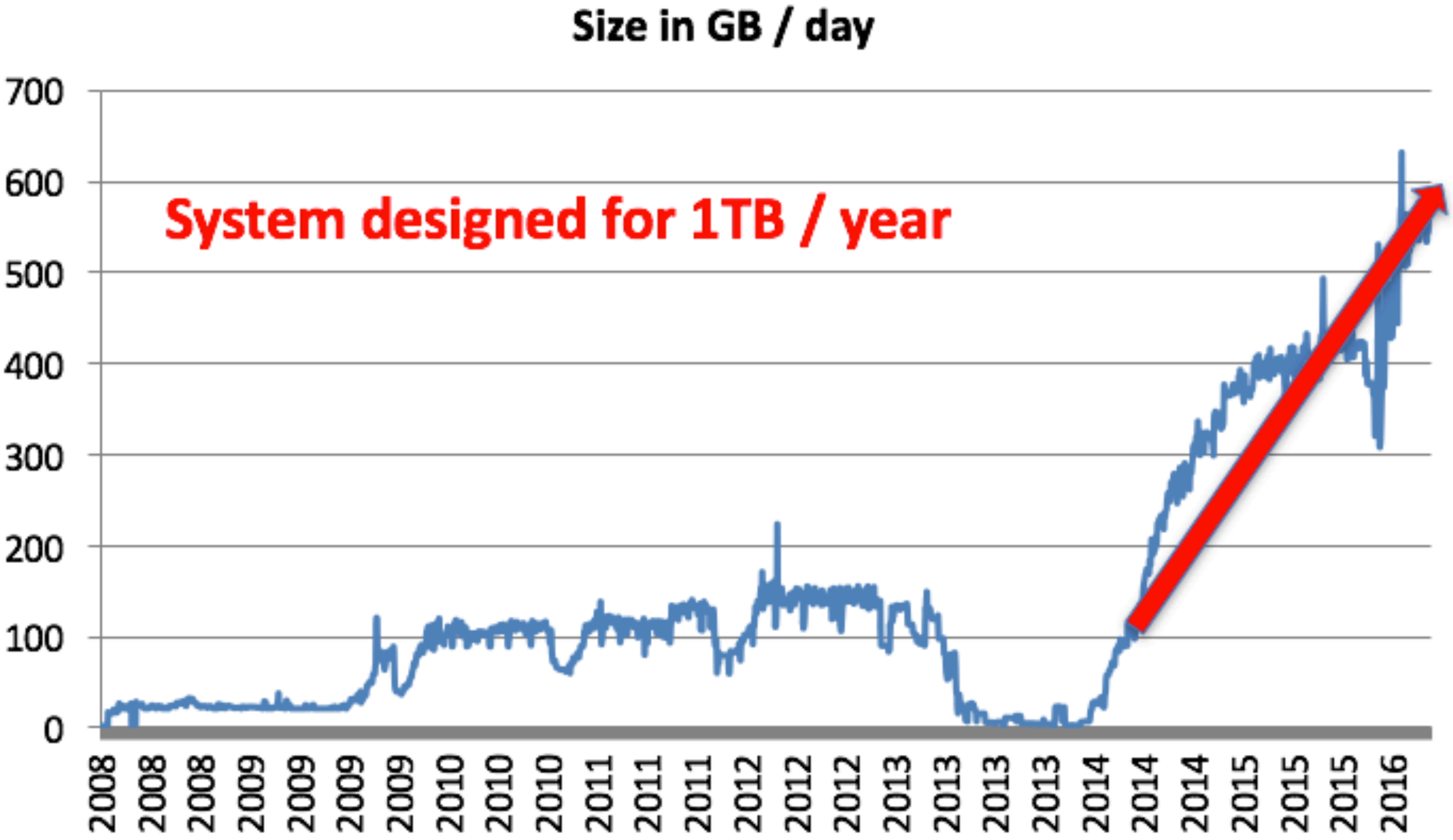
CERN Accelerator Logging Service (CALS)

- Mandate
  - **Stores data** from accelerator complex related devices
  - Information for acc. performance improvement
  - **Decision support** system for management
  - Avoids **duplicate logging** efforts
- In numbers
  - Built for **1 TB / year** throughput
  - Currently **1.2 TB / day** for all DBs
  - 1,500,000 signals
  - 5 billion dp/day, 1.6E12 dp/year
  - 6 million extraction requests per day
  - Soon reaching **Peta Bytes stored (~0.5PB)**

Persistence layer based on Oracle DB



CALS Storage Evolution

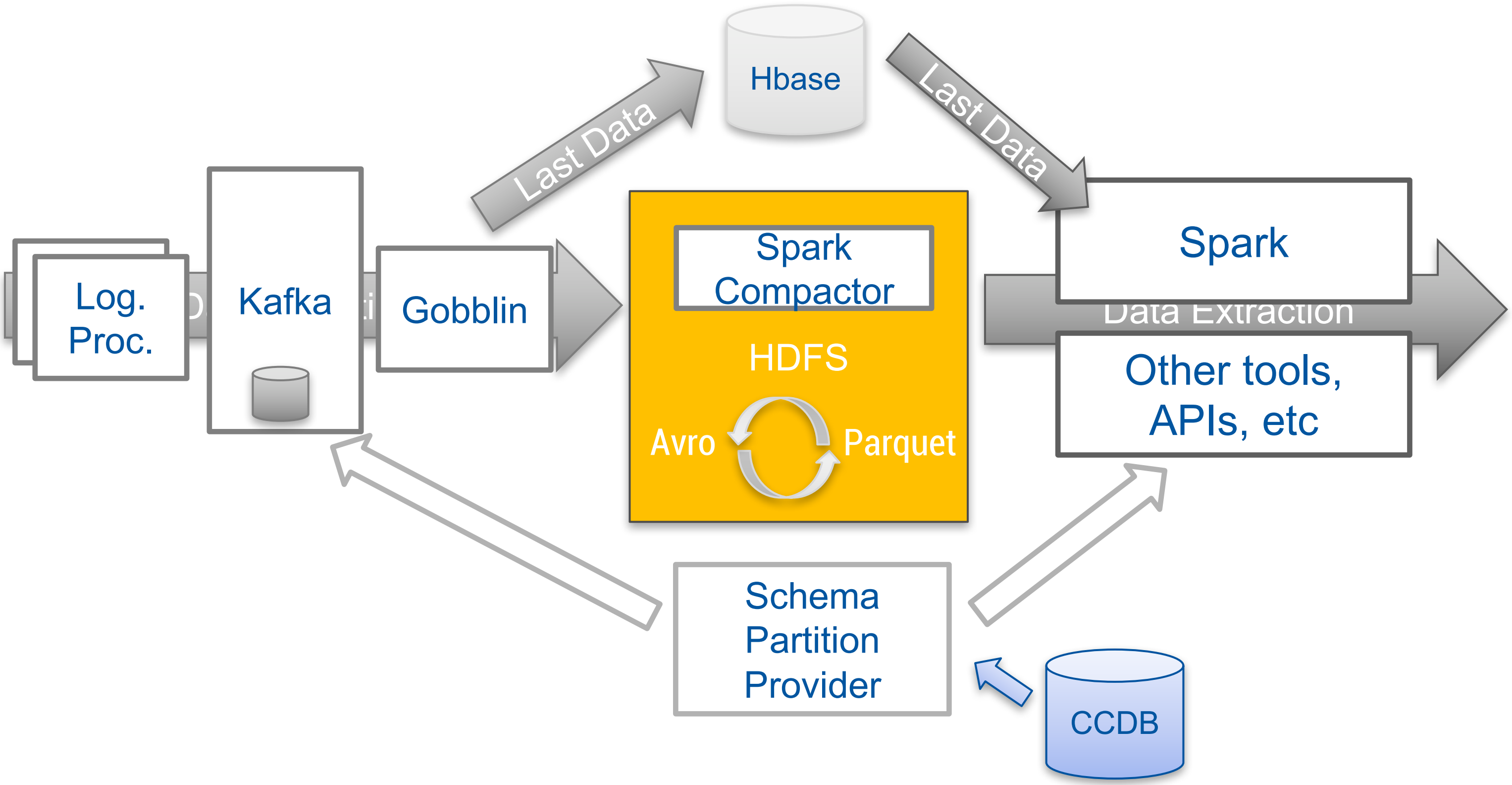


CALS - Current Challenges

- Dramatic increase of data load (in/out)
  - Frequency increase in many sub-systems to 10Hz
  - Very big vector data (2e06) - analog, bunch by bunch, ...
  - Some data sources cannot be filtered
  - Injectors data (request for 20k new devices)

- No support for near-data processing
  - Have to extract all data first to analyze it -> API limited
  - Emerging custom logging systems

Future CALS architecture (NXCALS)



Data Formats

Used as intermediate row-oriented storage

Final column oriented storage in files

Apache Parquet

- Open format
- Based on Google "Dremel" white-paper
- Columnar storage
- Very efficient compression algorithms
  - Delta encodings
  - Binary (bit) packing
  - Dictionary
- Very efficient reads (avoid reading unwanted data)
- Separating metadata and column data

NXCALS Data Partitioning

- Data stored in Parquet files of records {f1,f2,...,fn} partitioned by

  `system/classifier/schema/date`
  - Dynamic records of ANY content
  - They represent a change of "state" in time for some "entities"
  - Schema per entity CAN change over time
- Pros
  - Very accommodating storage system
  - Convenient to gather data statistics i.e. about used space per client/system
  - Convenient to move/backup/restore on demand
  - More optimal for scanning (less data to process)
- Problematic
  - Historical schema changes for a given data source over time (problem of renaming fields over time)

The renaming problem with Parquet

- **Class / version / property rename in the same version of a class**
- Like a migration. If there is a property rename we have to re-subscribe.
- We have to rename the directory or/and move some data around. The actual action depends on the semantics of the operation and how the old data is affected.
- It might complicate the backup. If the backup is just files copied over somewhere we do the same rename/move on the backup.
- We loose track of the history of changes, we might want to keep history of those renames. The original class/version/property is kept in the data files.
- Still somebody might want to ask about a given device/property from the past while this property might not exist any longer. As long as we

More questions about NXCALS?

- Please contact: Jakub.Wozniak@cern.ch

(Thank you Jakub for providing the slides!)

CERN

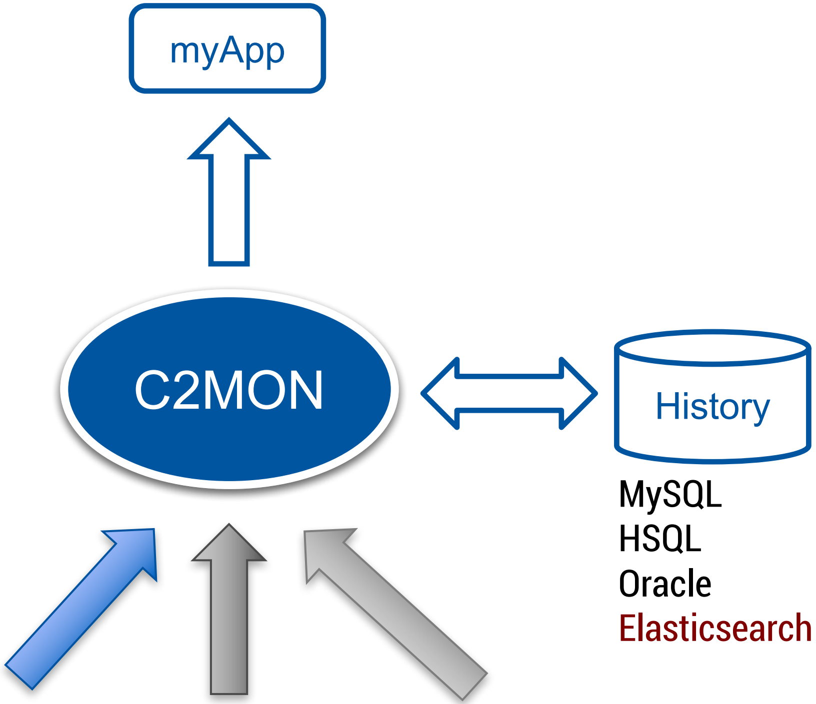
C2MON
CERN Control and Monitoring Platform

The configuration hell

- Many different types of data sources and protocols
- Complex data structure and addressing
- Different data rates

How to subscribe to my data?

We need …

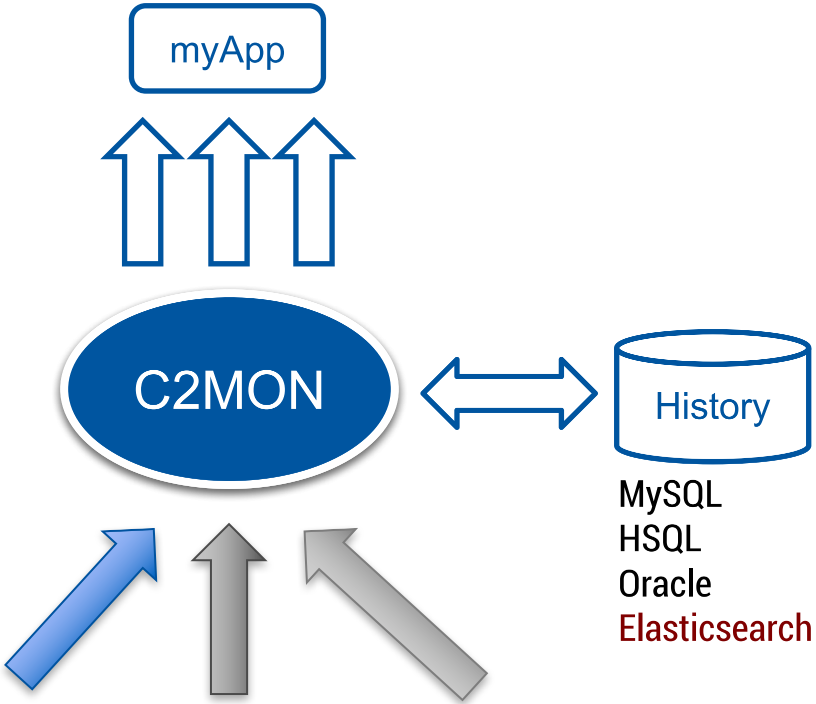


… a platform that:

- handles low level data subscriptions
- monitors the different data sources
- reconfigures acquisition processes at runtime
- **standardises messages** and **data storage**
- reduces data streams to relevant information
- always keeps the latest values available
- provides custom data streams
- provides access to history

… and is modular and open source!

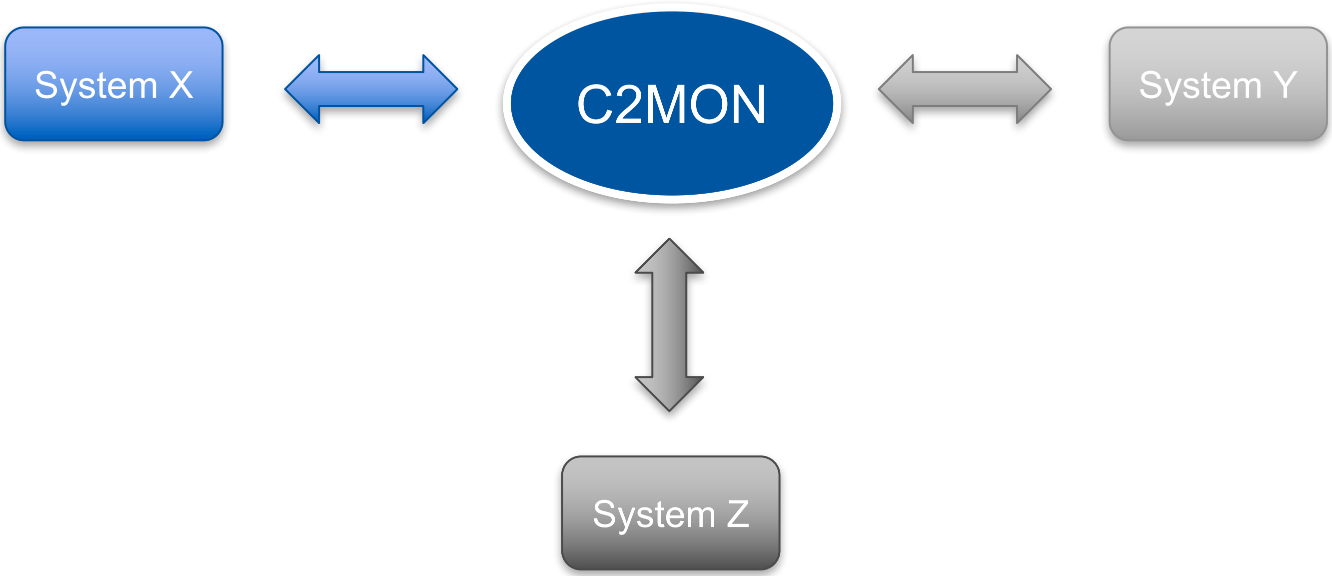
C2MON – A great platform for many use cases



Use C2MON...

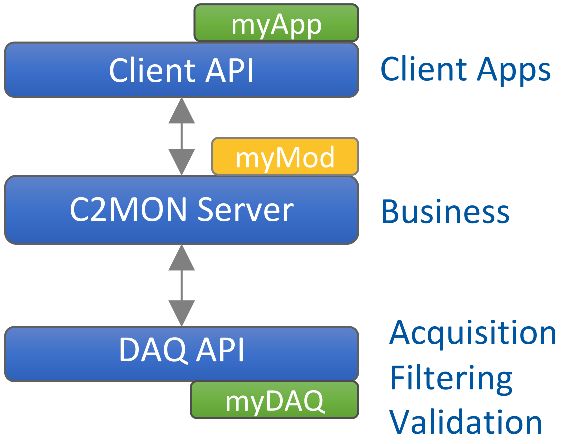
- to feed your analytics framework
- to structure persist your data in ES for offline analytics
- as backbone for your SCADA system
- as configurable data proxy
- to write innovative new Java and Web applications
- ...

Use C2MON to realise IoT scenarios



C2MON - CERN Control and Monitoring Platform

- Modular and **scalable at all layers**
- Optimized for high availability & big data volume
- Server based on In-Memory cache solution



Two big monitoring services (TIM & DIAMON) **running in production with C2MON at CERN**

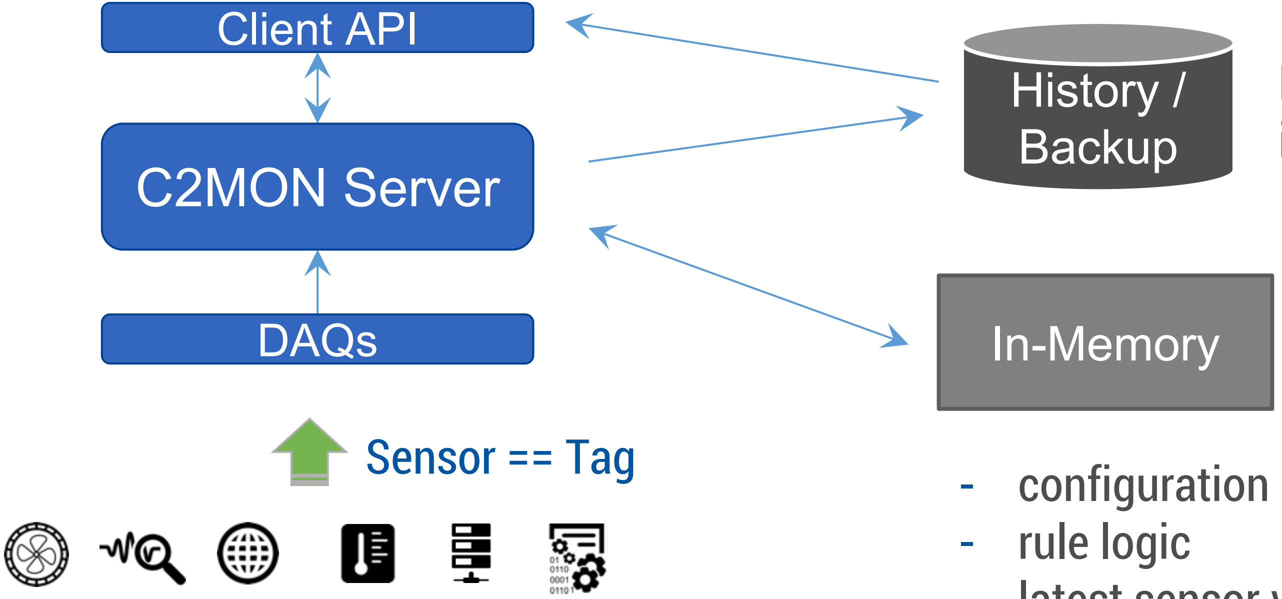
- Central TI alarm system in migration phase
- Other CERN projects in prototyping phase
- TU Berlin first users outside of CERN

Ready for starting Open Source community!

CERN

http://cern.ch/C2MON

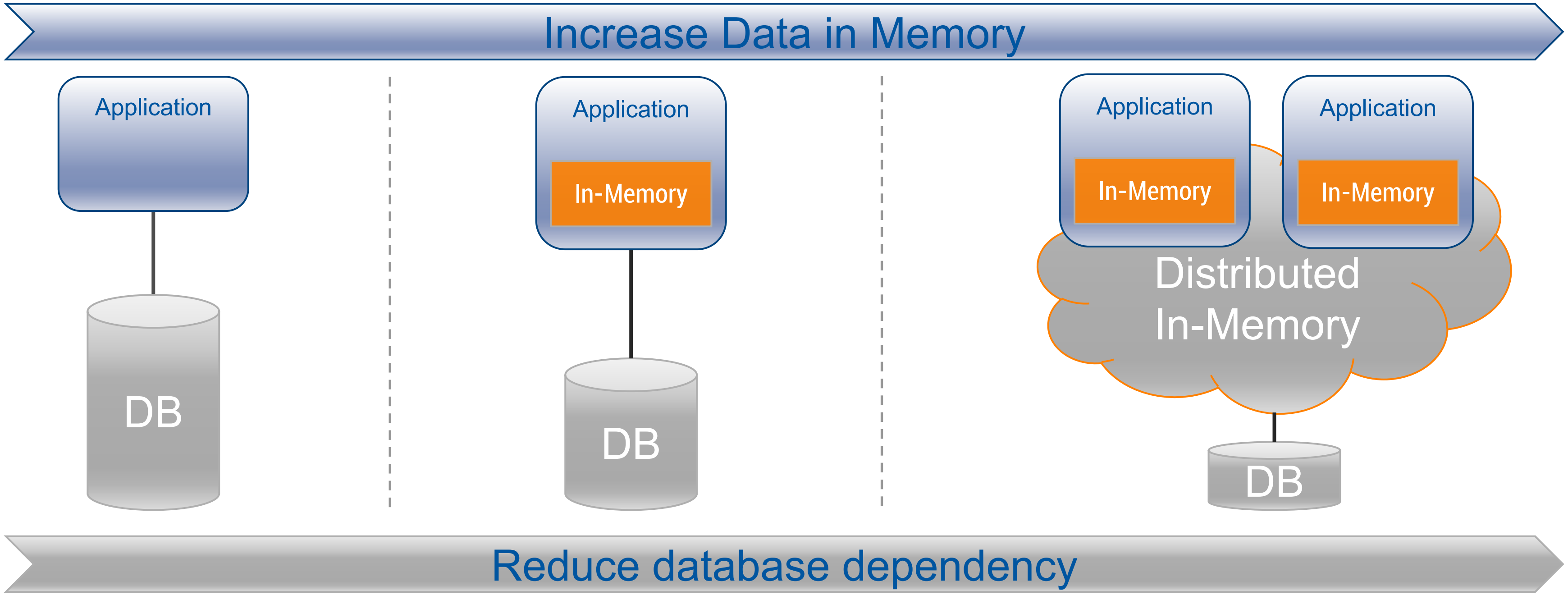
Architecture



No downtime, if DB is not available.

- configuration
- rule logic
- latest sensor values
- assuring high availability

In-Memory approach:
Scale with data and processing needs



In-Memory Data Grid solutions

Popular Open Source solutions:

redis

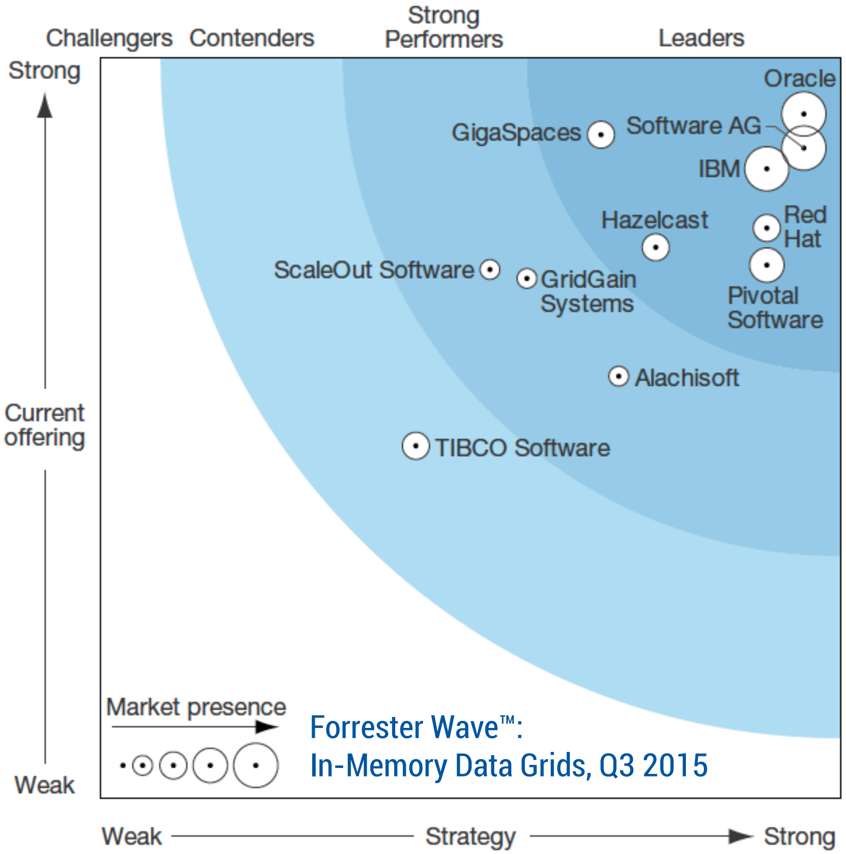
TERRACOTTA

EHCACHE

MEMCACHED

apache Ignite

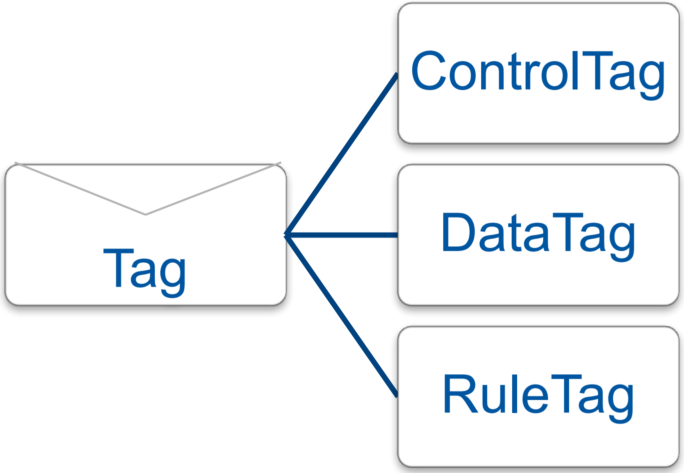
hazelcast



Forrester Wave™: In-Memory Data Grids, Q3 2015

The Tag family

- id
- name
- value
- quality
- timestamp
- metadata



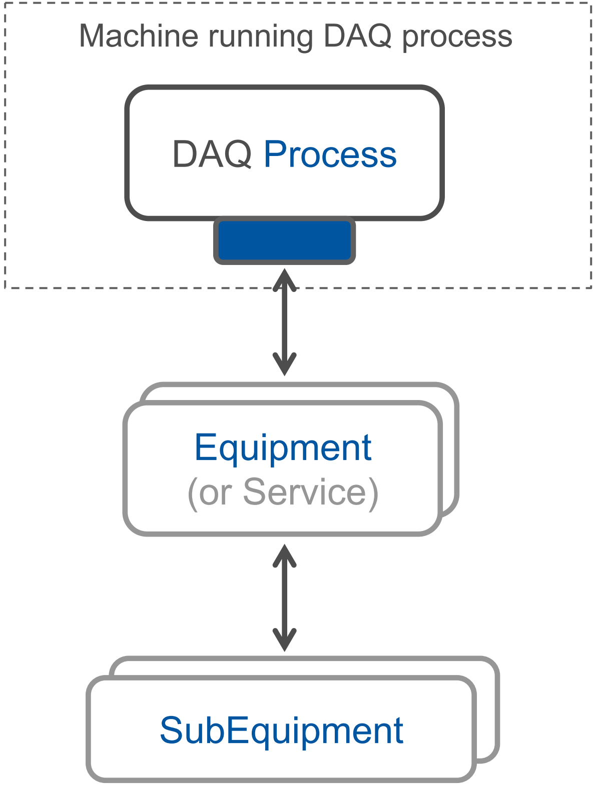
- Internally used for Process and Equipment surveillance
- Used for data acquisition *
- (#123 + #234) > 2 [ERROR], true [OK]

* Support of primitive arrays and arbitrary Objects

C2MON Acquisition layer

DAQ Process takes care of:

- Equipment/Service monitoring
- Data acquisition for configured Tags
- Raw data validation & filtering
- Sending data to server tier



Raw data validation & filtering on DAQ layer

Dynamic Filtering

- Dynamic Time dead-band filtering for Protecting against data bursts

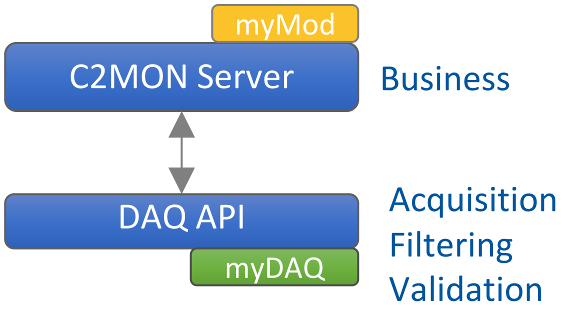
Static Filtering

- Static time dead-band filtering
- Value redundancy
- Value dead-band filtering

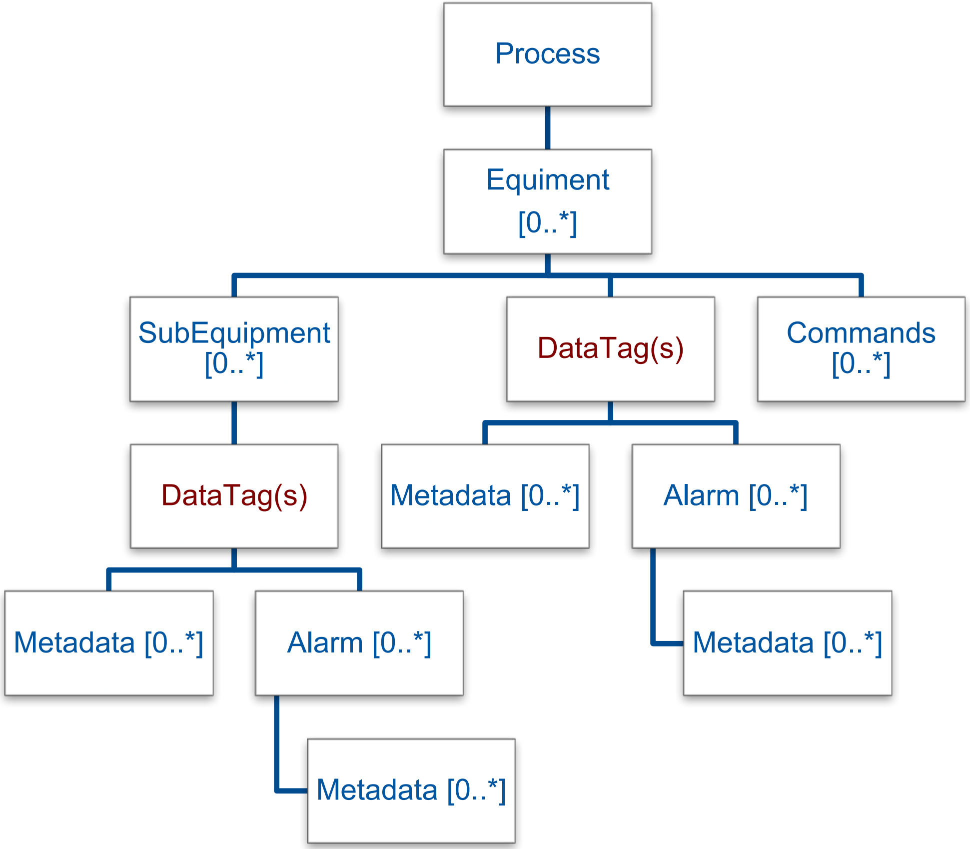
Data Validation

- Value in defined range?
- Correct value data type?
- Source timestamp in the future?
- Outdated information?

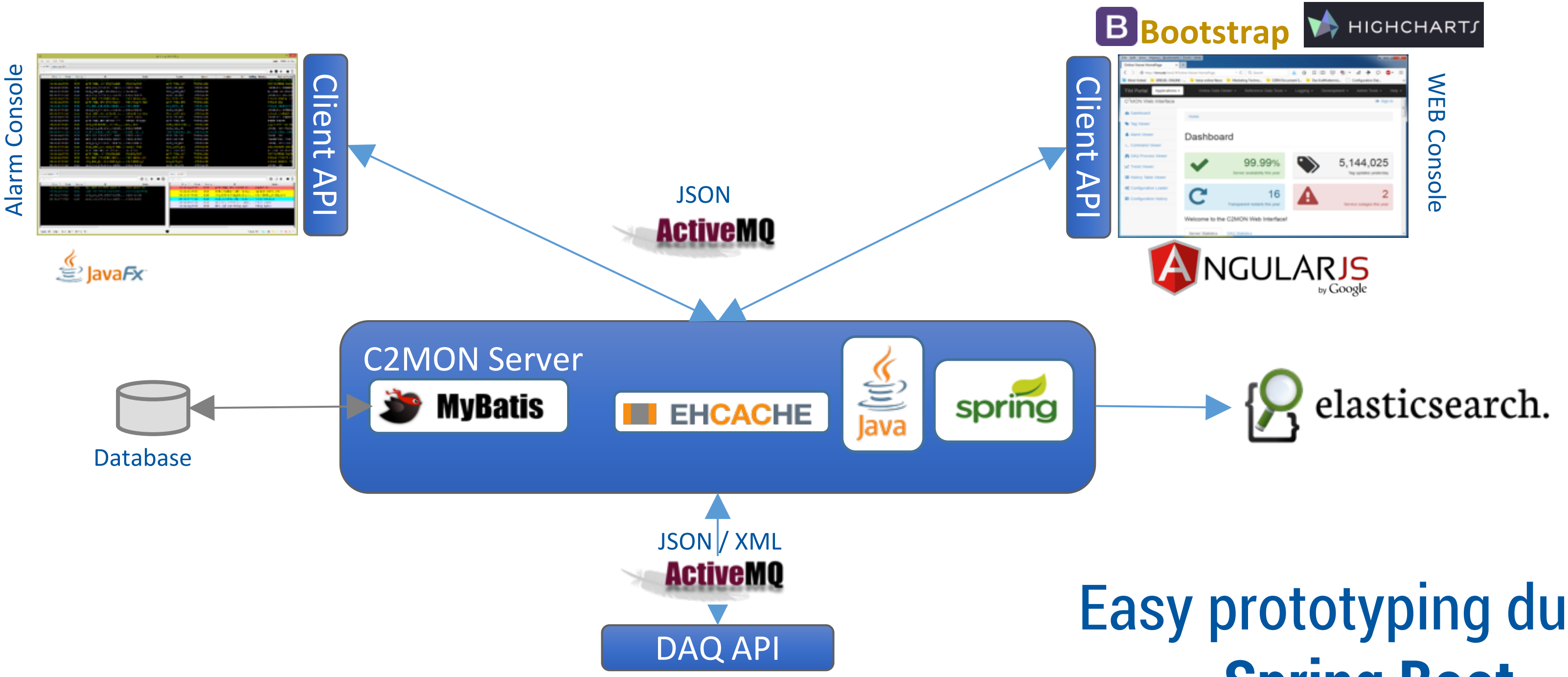
Configurable by Tag



Basic configuration structure



Open Source in all layers



Easy prototyping due to **Spring Boot**

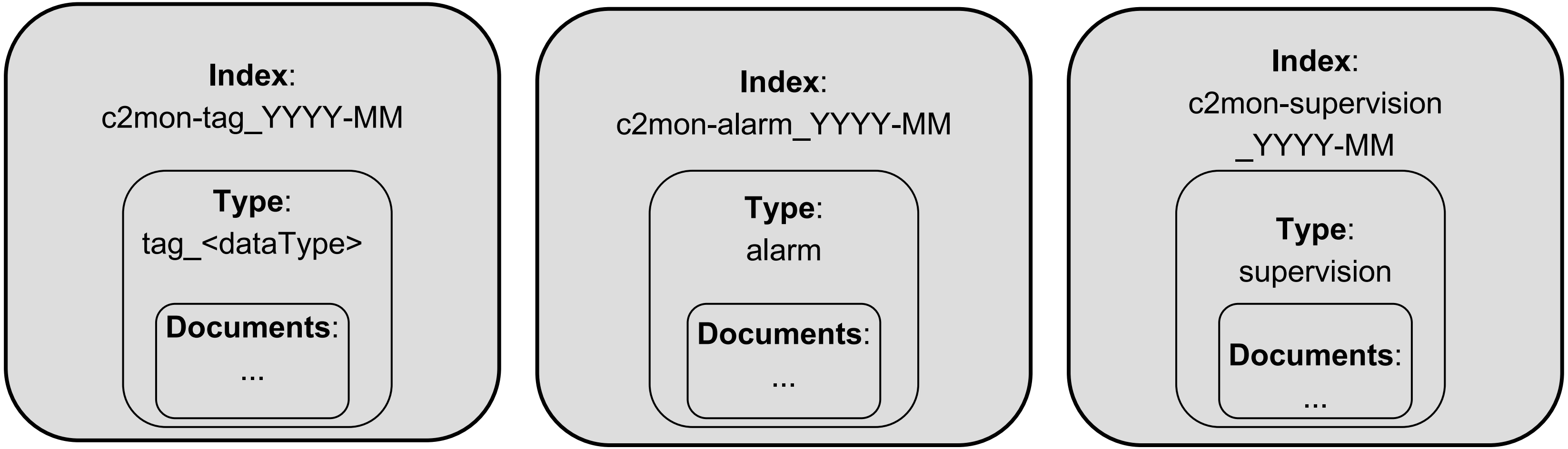
elasticsearch. as timeseries data storage

kibana ← **C2MON** → Grafana

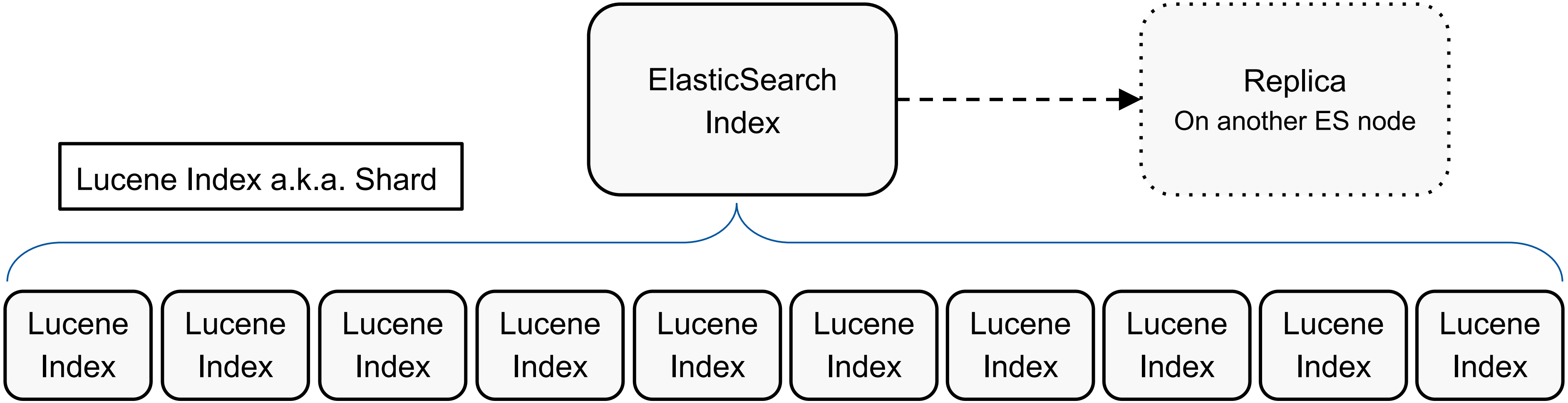
Motivation for using Elasticsearch

- Provide better and faster charting
- Improve Dashboard playback functionality
- Provide longer data storage (1 - 2 yrs)
- Simplify generation of tag and alarm statistics
- Enable data analytics e.g. with Spark
- Query data through HTTP POST
- Provide Open Source alternative to Oracle storage

Elasticseach Structure



ElasticSearch Index



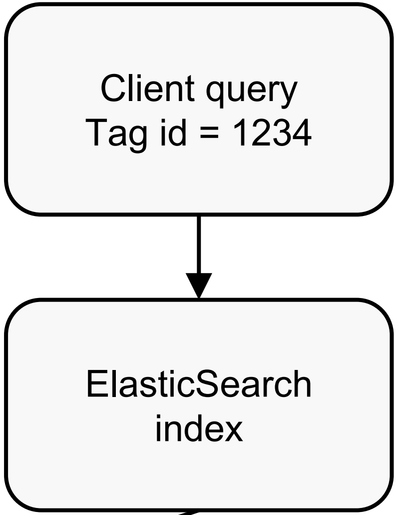
Inverted Index between ElasticSearch and Lucene.

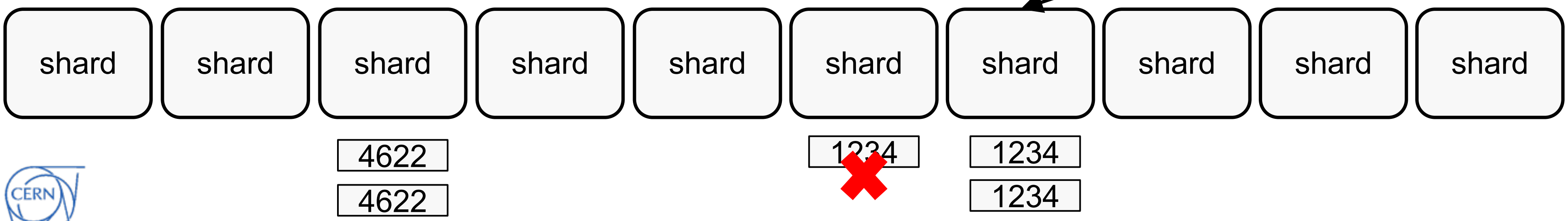
Routing

```
{
   "c2mon-tag_2017-03" : {
      "mappings" : {
         "tag_float" : {
            "_routing" : {
               "required" : true
            }
         }
      }
   }
}
```

```
IndexRequest indexNewTag =
new IndexRequest(index, type)
.source(json)
.routing(String.valueOf(tag.getId()));
```

shard = hash(routing) % number_of_primary_shards





Performance by leveraging all ES features

- Mapping: set parameters for better performance on retrievals.
- Routing: a query on tagId will hit only 1 shard.
- Aliases: for each tagid, faking index per tag. Other possibilities (e.g., last day...).
- Pagination: retrieve first *N* results and then fetch next *N*, ...
- Filters: denormalized data; filter the results according to TIM metadata.

Elasticsearch Document example


```
{
  "_index": "tim-tag_2017-02",
  "_type": "type_float",
  "_id": "AVqG87fNdUmPrPQhaeQP",
  "_score": null,
  "_routing": "195222",
  "_source": {
    "id": 195222,
    "name": "EA.MEY.EMD109*43:U_T_R",
    "description": "MESURE_TENSION",
     "value": 18128,
    "metadata": {
      "responsiblePerson": "JOHN DOE",
      "site": "MEY",
      "pointAttribute": "U_T_R",
      "otherEquipCode": "EMD109*43",
      "subsystem": "ELEC UPS",
       "location": "513"
     },
     ...
  }
}
```

C2MON example
Technical Infrastructure Monitoring (TIM)

- Operational since 2005
- Used to monitor and control infrastructure at CERN
- **24/7** service
- ~ 100 different main users at CERN

- Since Jan. 2012 based on
new server architecture with C2MON

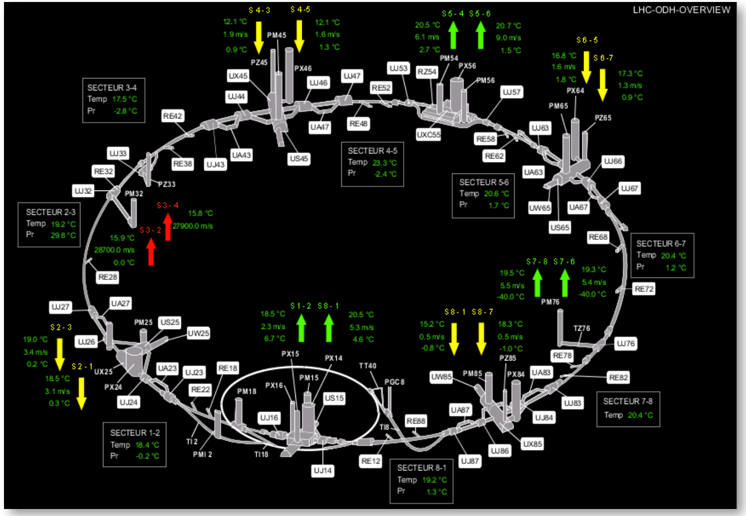
CERN Control Center at LHC startup

C2MON example

Technical Infrastructure Monitoring (TIM)

- ~ 90'000 sensors
- ~ 50'000 alarms
- ~ 400 million raw data values
- ~ 3 million after filtering

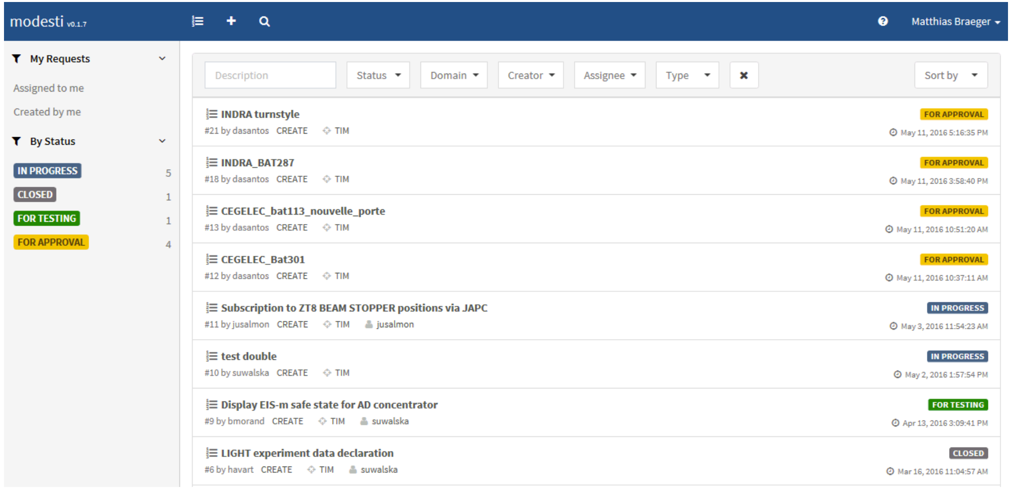
- **20-30 Gb/month** in Elasticsearch



TIM Dashboard Example

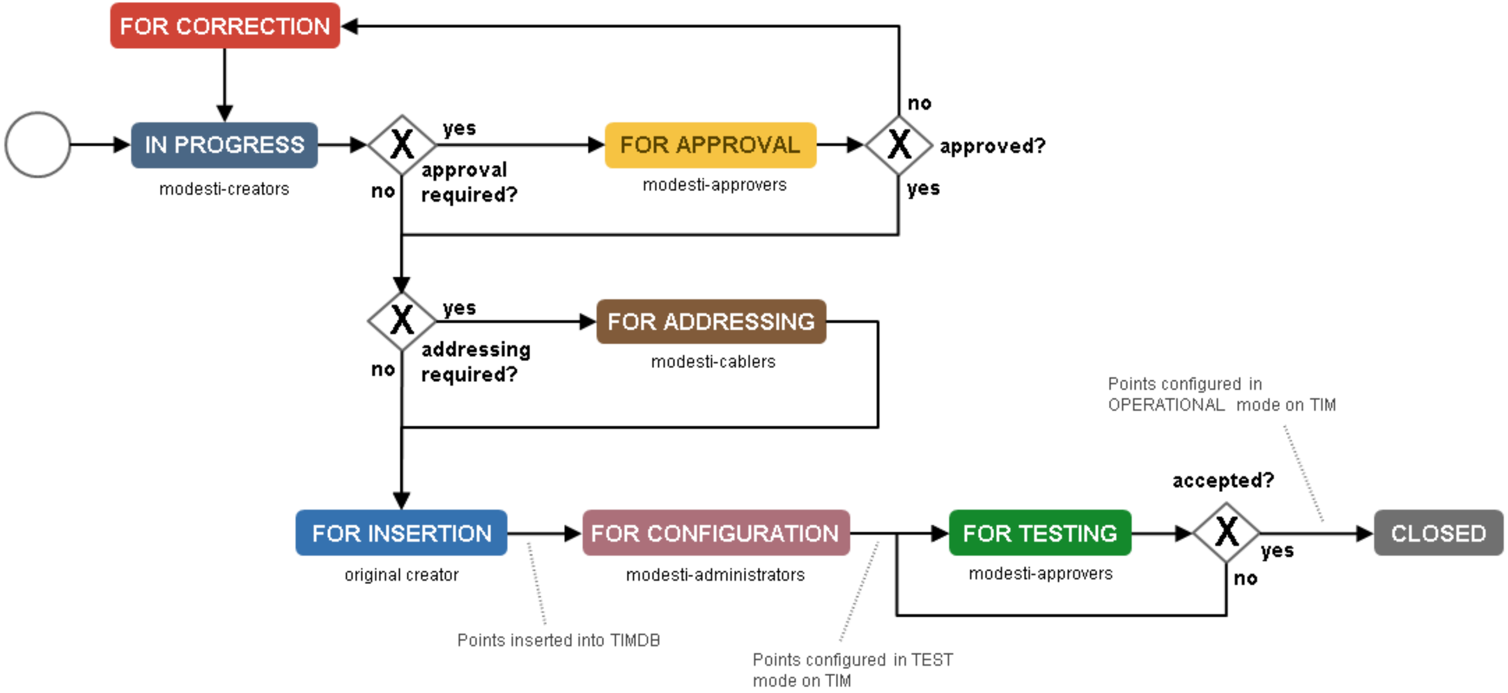
TIM – Main features

- Unifies sensor data from a multitude of sources and protocols
- Provides simple dashboarding and access to historical values
- Central configuration management
- Filters raw data streams



Workflow based sensor and alarm declaration

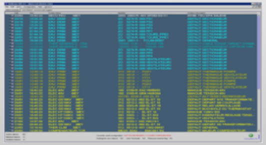
Activiti BPMN 2.0 workflow



MongoDB for instant search

- MongoDB is a schemaless, object-oriented datastore allows rapid development
- JSON all the way down
- Replication and sharding out-of-the-box

Client Tier


Alarm Console


Data Analysis


TIM Viewer


Web Apps


Grafana

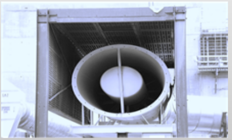
- > 90k data sensors
- > 50k alarms

TIM Server

based on C2MON

- > 1200 commands
- > 1300 rules

Data Acquisition & Filtering


Cooling


Safety Systems

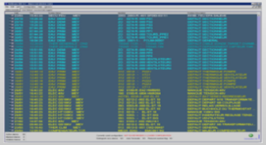

Electricity


Access


Network and Hardware Controls


Cryogenics

Client Tier



Alarm Console

Data Analysis

TIM Viewer

Web Apps

Grafana

> 90k data sensors
> 50k alarms

TIM Server

based on C2MON

> 1200 commands
> 1300 rules

Data Acquisition & Filtering

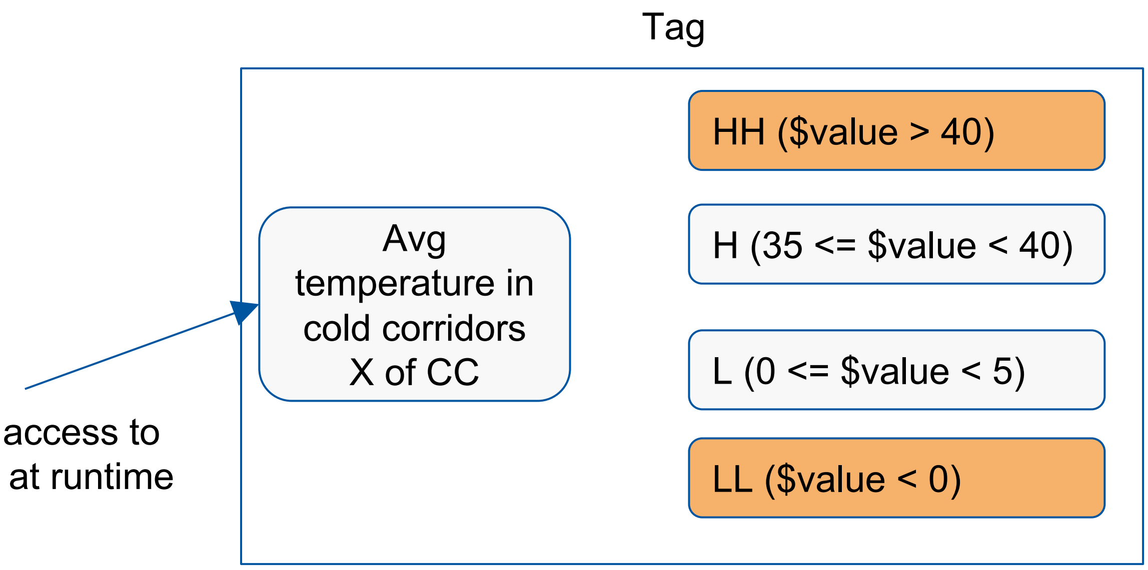
~ 400 million raw values per day

Filtering →

~2 million updates

Renovation of C2MON Rule Engine

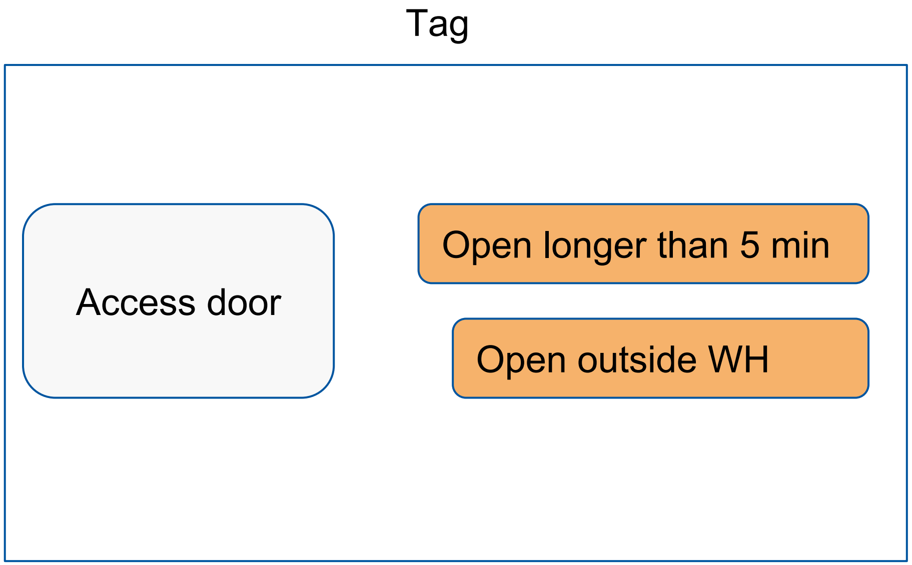
Complex rules and expressions



Requires access to metadata at runtime

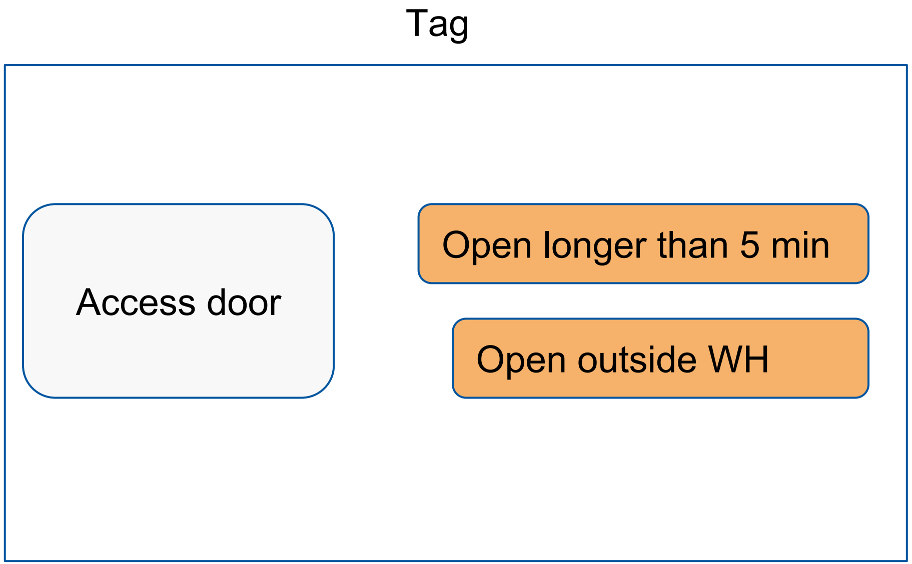
Coming soon!

Time based alerts



Coming soon!

Time based alerts



Coming soon!

The future Rule design

Introduce a new expression Language based on **Groovy script**

- Groovy can be injected and compiled at runtime to C2MON cluster
- Can take advantage of In-Memory cache, Elasticsearch and other 3rd party solutions

Example for a possible DSL:

"Average of accumulated temperature sensor data of last 5 min from building 864"

avg(q(name:'*temperature', location:'864', '5m'))

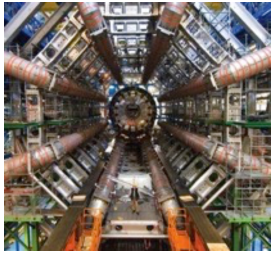
Credits & References

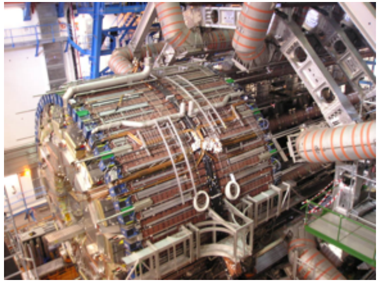
Many thanks to:

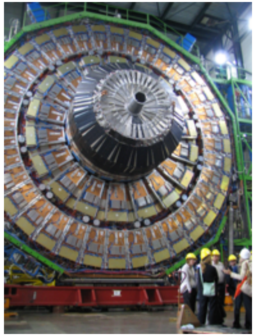
- Sebastien Ponce (CERN), for providing information about CASTOR
- Rainer Toebbicke (CERN), for providing information about CERN HBASE service
- Jan Iven (CERN), for being helpful finding information about existing CERN Hadoop projects
- Jakub Wozniak for providing information about NXCALS
- The entire TIM/C2MON team, which does a fantastic job!

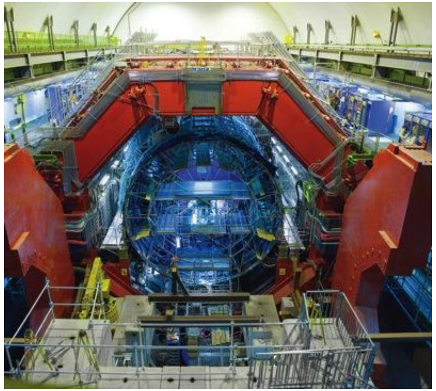
References:

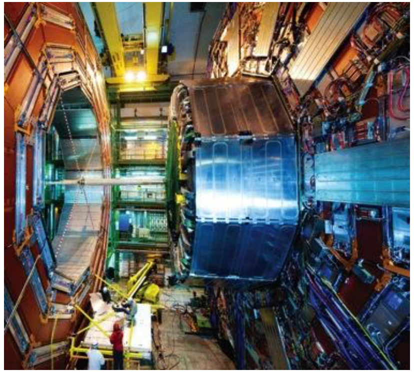
- C2MON: http://cern.ch/c2mon
- The ATLAS EventIndex: https://cds.cern.ch/record/1690609
- Agile Infrastructure at CERN - Moving 9'000 Servers into a Private Cloud, Helge Meinhard (CERN): http://vimeo.com/93247922











Questions?

Thank you for your attention!

Matthias.Braeger@cern.ch