Overview of Hadoop and Spark service at CERN

Zbigniew Baranowski, IT-DB Hadoop and Spark Service
May 7th, 2018

Infrastructure data and metadata

- For a long time traditional RDBMS was up to the task
- Evolution of systems and new processing use cases greatly increased requirements for data store backend
  - More data generated (more sensors, higher frequencies)
    - > 100 GB/day
  - New use cases appeared – e.g. analytics, machine learning
    - The initial design of the systems is not optimal for that
- Result -> Hard to scale RBDMS to the new big data use cases

Why Hadoop?

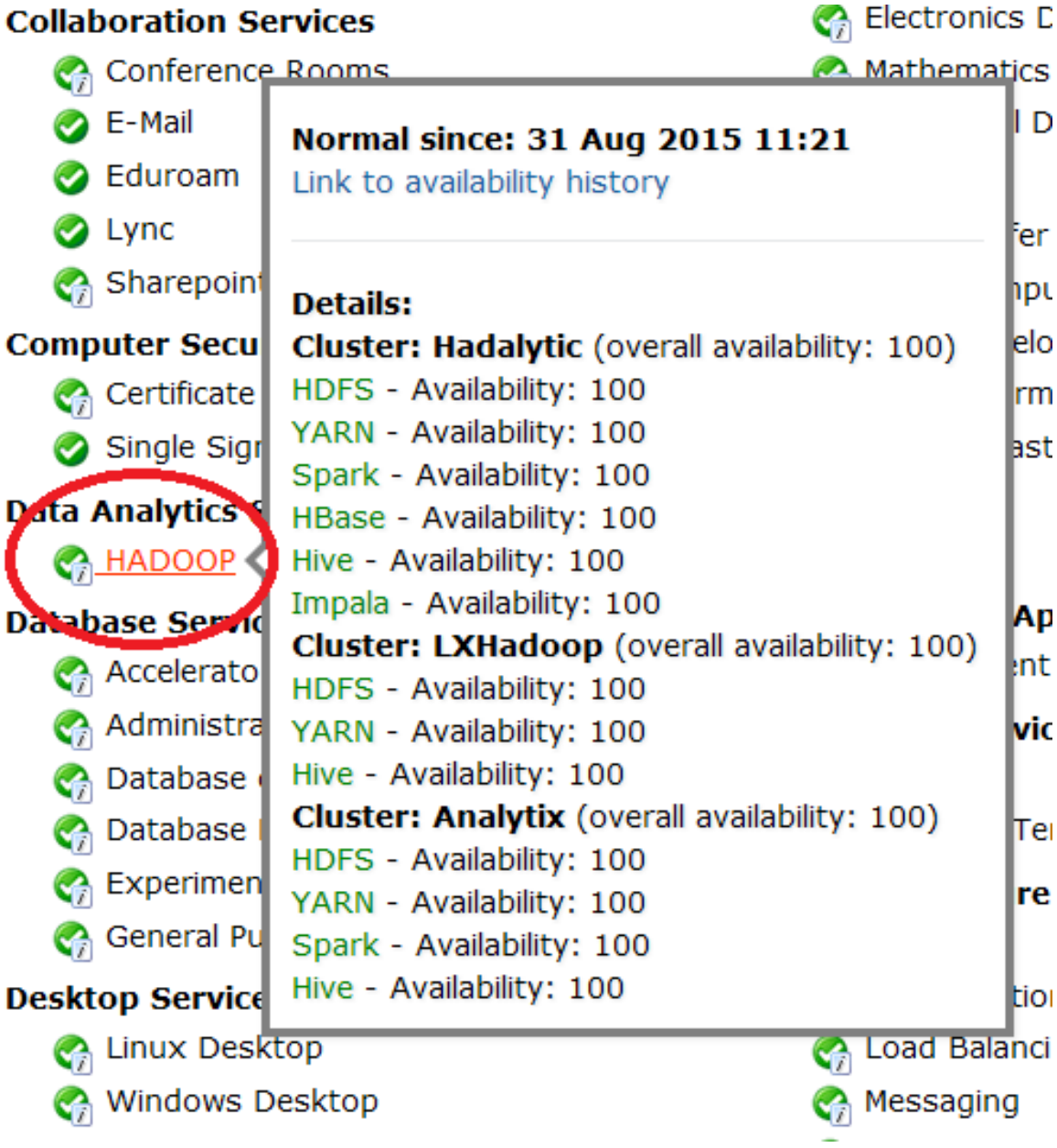
- Already well establish in the industry and open source
- Distributed systems for data processing
  - Can operate at scale by design (shared nothing)
  - Typically on clusters of commodity-type servers/cloud
  - Many solutions target data analytics and data warehousing
  - Can do much more: data ingestion, streaming, machine learning

CERN

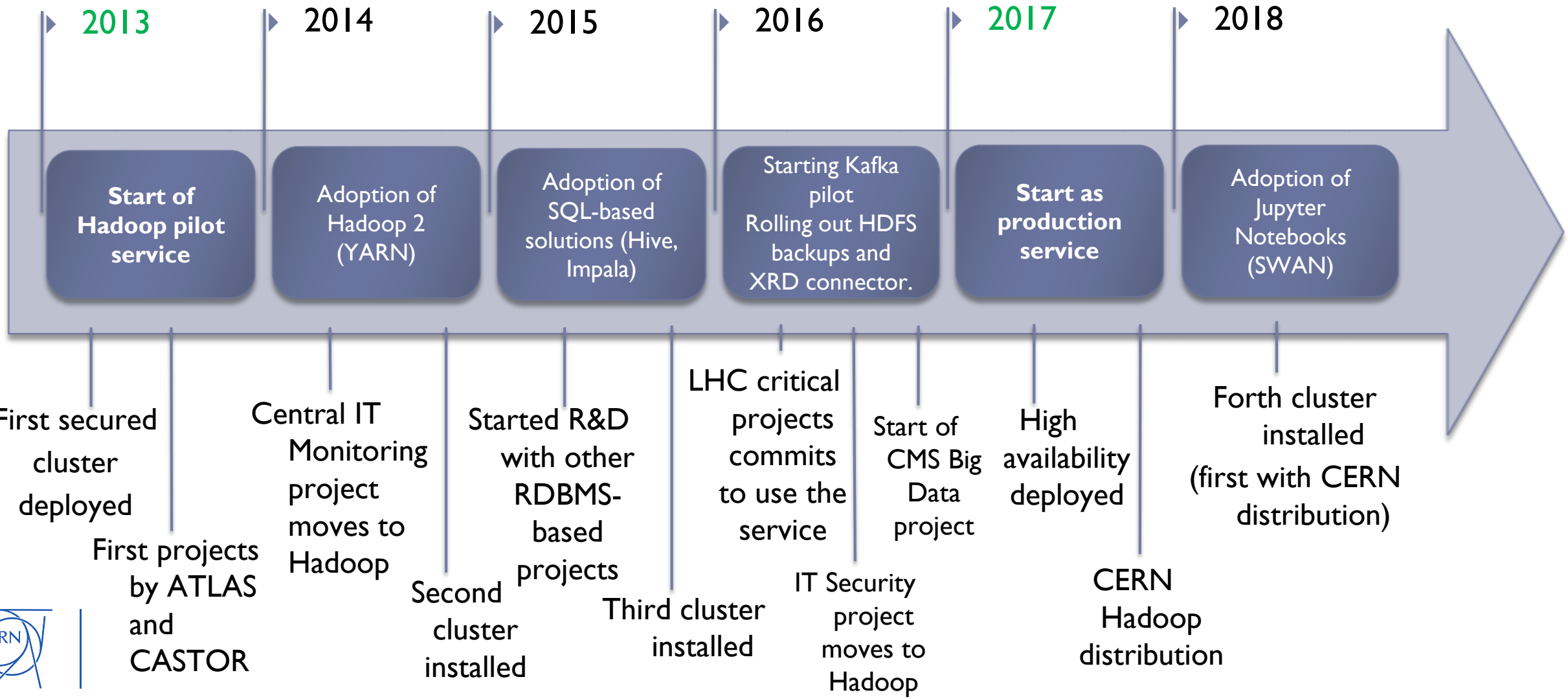
Hadoop Service at CERN

Hadoop Service at CERN IT (since 2013)

- Setup and run the infrastructure
- Provide consultancy
- Support user community
- Running for more than 4 years



Hadoop at CERN - Timeline



Hadoop production deployment @CERN

- Production systems deployed on bare-metal
- Development and QA deployed on VMs

- OS and Hadoop stack installation and configuration done with Puppet
  - CentOS 7
  - we use our custom puppet module to install Hadoop machines

- Hadoop distribution
  - Cloudera (CDH5) rpms (on 3 clusters)
  - CERN custom (based on Apache) distribution (on new 3 clusters)
  - we are in process of migration fully to our custom distribution

- HDFS, YARN and HBase in high availability mode
  - Enables online/rolling service operations (do not require full showdown of the service)

- All clusters run in a secure mode
  - Authentication with Kerberos
  - Authorization based on e-group membership

- New custom monitoring done with ElasticSearch + Grafana
  - Previously we were using Ganglia
  - OS-level monitoring done with the CERN IT monitoring system

- Alerting – custom scripts with sensors
  - Checking availability and usability of Hadoop components
  - OS-level alerting done with the CERN IT monitoring system

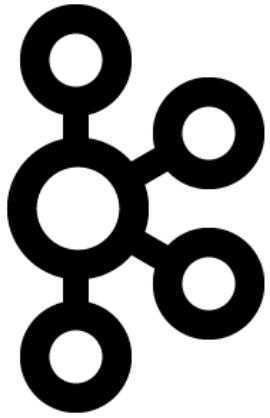
- HDFS Backups to Castor (CERN Storage) done with MapReduce (metadata stored in RDBMS)

Hadoop production clusters at CERN

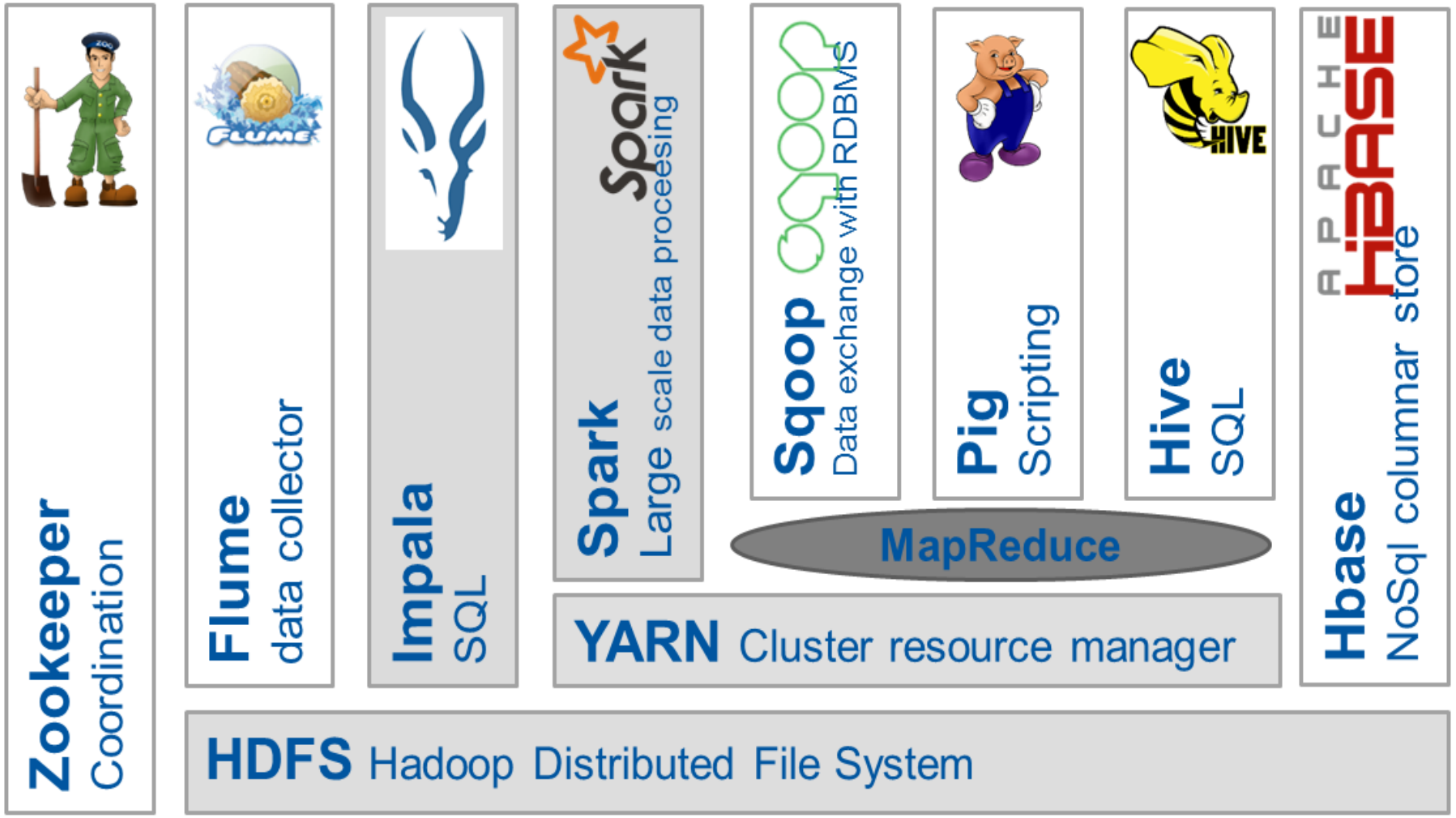
- 4 production clusters
- 2 development

Cluster Name	Configuration	Primary Usage
lxhadoop	18 nodes (Cores – 288,Mem – 912GB,Storage – 1.29 PB)	Experiment activities
analytix	42 nodes (Cores – 524,Mem – 6.9TB,Storage – 6 PB)	General Purpose
hadalytic	14 nodes (Cores – 196,Mem – 768GB,Storage – 2.15 PB)	SQL-oriented engines and data warehouse workloads
nxcals	20 nodes (Cores 480, Mem - 8 TB, Storage – 5 PB, 96GB in SSD)	Accelerator logging (NXCALS) project dedicated cluster

Overview of Available Components



Kafka:
streaming
and ingestion



Data volume (from backup stats July2017)

Application	Current Size	Daily Growth
IT Monitoring	420.5 TB	140 GB
IT Security	125.0 TB	2048 GB
NxCALS	10.0 TB	500 GB
ATLAS Rucio	125.0 TB	~200 GB
AWG	90.0 TB	~10 GB
CASTOR Logs	163.1 TB	~50 GB
WinCC OA	10.0 TB	25 GB
ATLAS EventIndex	250.0 TB	200 GB
USER HOME	150.0 TB	20 GB
Total	**1.5 PB**	**4 TB**

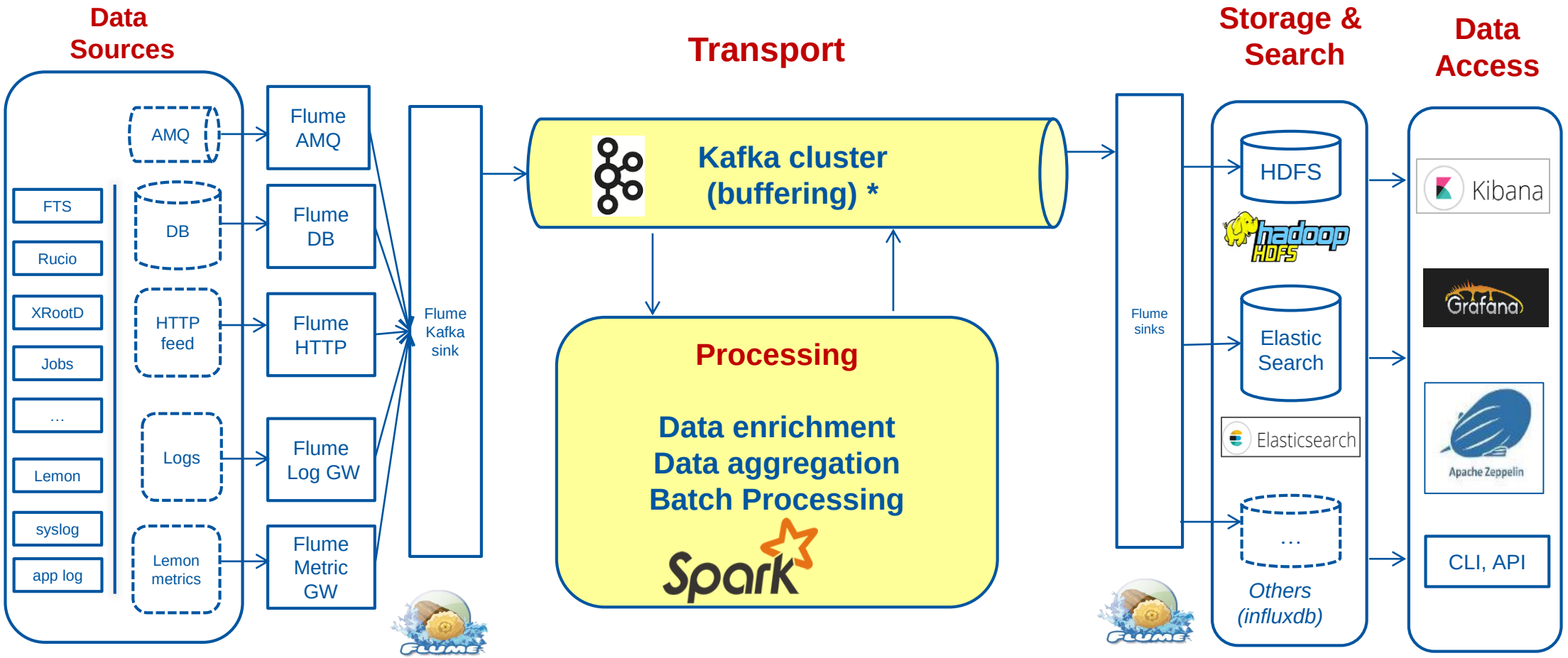
CERN Apache Hadoop distribution

- For core components
  - HDFS and YARN
  - Spark
  - HBase
- Better control of the core software stack
  - In-house compilation
  - Enabling non default features (compression algorithms, R for Spark)
  - Adding critical patches (that are not ported in upstream)
- Streamlined development
  - Available on Maven Central
- RPMs-based – similarly to Cloudera or Hortonworks

(Selected)
Big data projects/use cases

New CERN IT Monitoring infrastructure

Critical for CC operations and WLCG



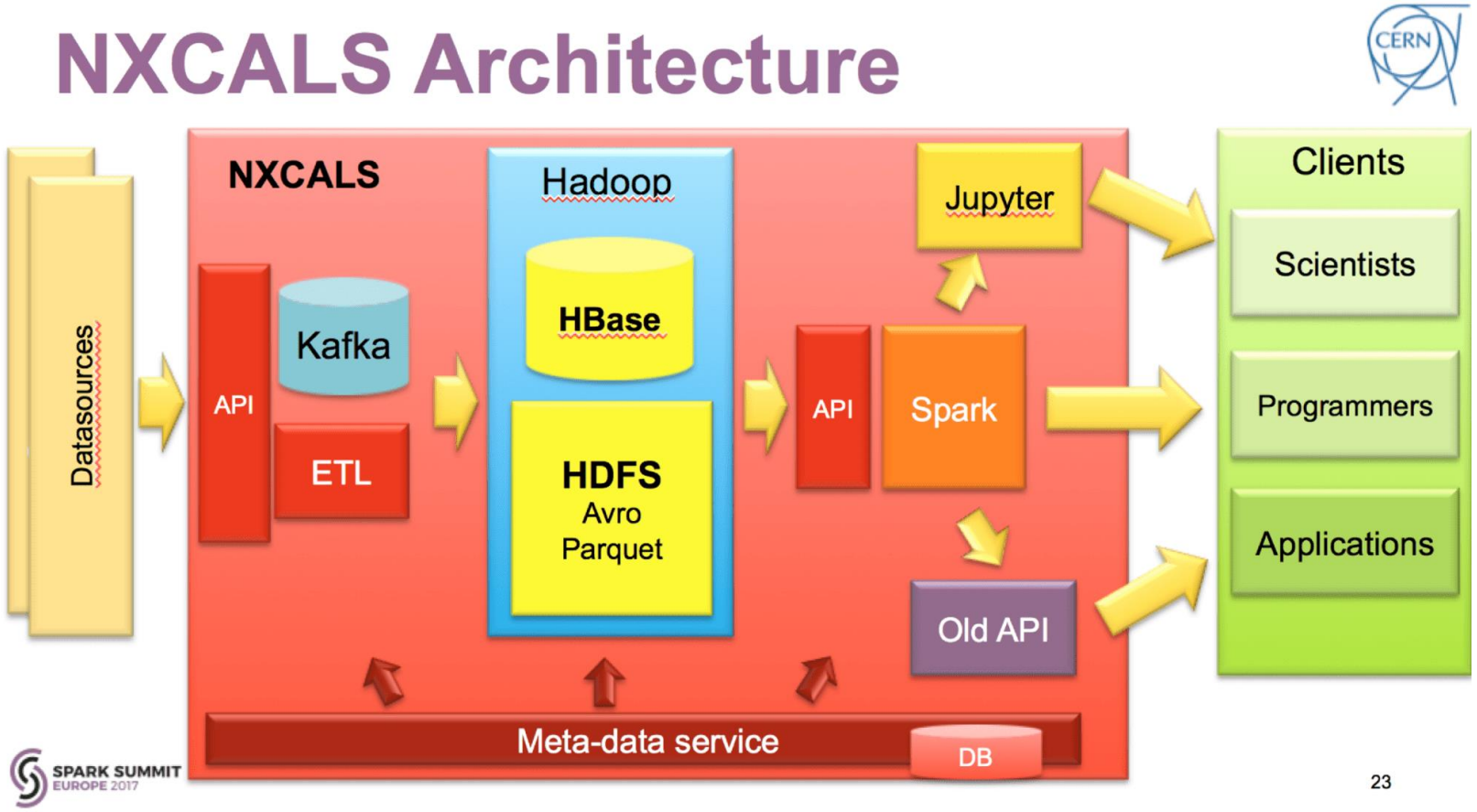
- Data now 200 GB/day, 200M events/day
- At scale 500 GB/day
- Proved effective in several occasions

Credits: Alberto Aimar, IT-CM-MM

Next Gen. Archiver for Accelerator Logs

Critical system for running LHC - 700 TB today, growing 200 TB/year

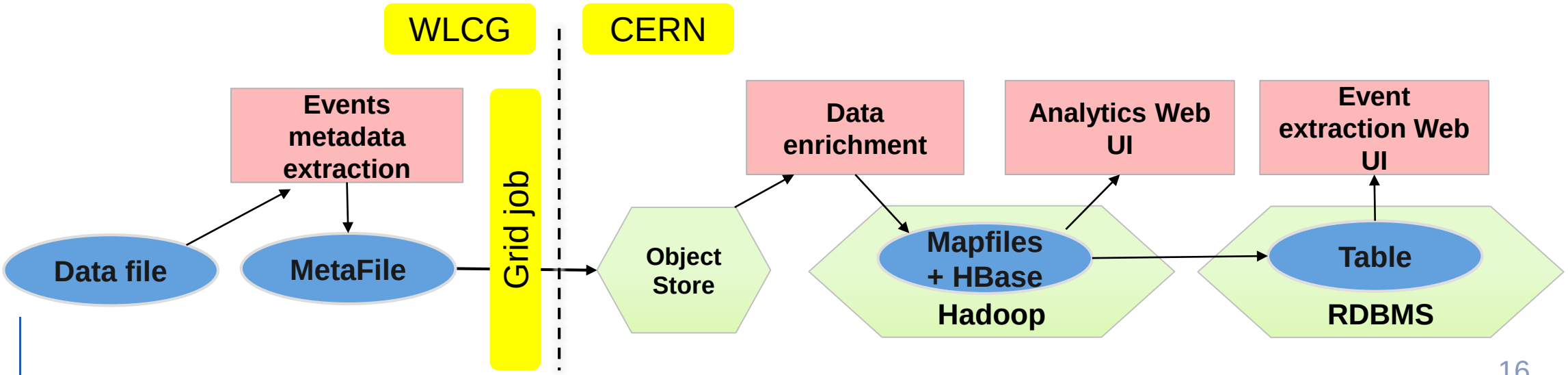
Challenge: service level for critical production



Credit: Jakub Wozniak, BE-CO-DS

The ATLAS EventIndex

- Catalogue of all collisions in the ATLAS detector
  - Over 120 billion of records, 150TB of data
  - Current ingestion rates 5kHz, 60TB/year



SWAN – Jupyter Notebooks On Demand

- Service for web based analysis (SWAN)
  - Developed at CERN, initially for physics analysis
- A web-based interactive interface and platform that combines code, equations, text and visualisations
  - Ideal for exploration, reproducibility, collaboration
- Fully Integrated with Spark and Hadoop at CERN
  - Python on Spark (PySpark) at scale
  - Modern, powerful and scalable platform for data analysis

FILE EDIT VIEW INSERT CELL KERNEL HELP Trusted Python 2

Code

Do the heavylifting in spark and collect aggregated view to panda DF

In [11]:
```
df_loadAvg_pandas = spark.sql("SELECT submitter_host, \
                                      avg(body.LoadAvg) as avg, \
                                      hour(from_unixtime(timestamp / 1000, 'yyyy-MM-dd HH:mm:ss')) as hr \
                                FROM loadAvg \
                                WHERE submitter_hostgroup = 'hadoop/itdb/datanode' \
                                AND dayofmonth(from_unixtime(timestamp / 1000, 'yyyy-MM-dd HH:mm:ss')) = 15 \
                                GROUP BY hour(from_unixtime(timestamp / 1000, 'yyyy-MM-dd HH:mm:ss')), submitter_host")\
                    .toPandas()
```

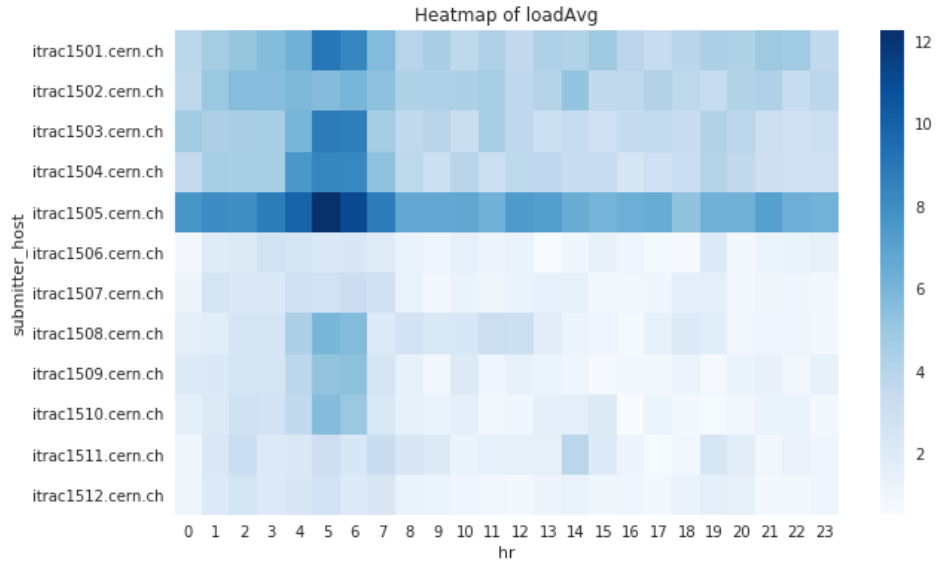
Apache Spark: 90 EXECUTORS 180 CORES Jobs: 1 COMPLETED

Job ID	Job Name	Status	Stages	Tasks	Submission Time	Duration
3	toPandas	COMPLETED	2/2	388 / 388	4 minutes ago	36s

Visualize with seaborn

In [19]:
```
# heatmap of service availability
plt.figure(figsize=(10, 6))
ax = sns.heatmap(df_loadAvg_pandas.pivot(index='submitter_host', columns='hr', values='avg'), cmap="Blues")
ax.set_title("Heatmap of loadAvg")
```

Out[19]: Text(0.5,1,u'Heatmap of loadAvg')



Text

Code

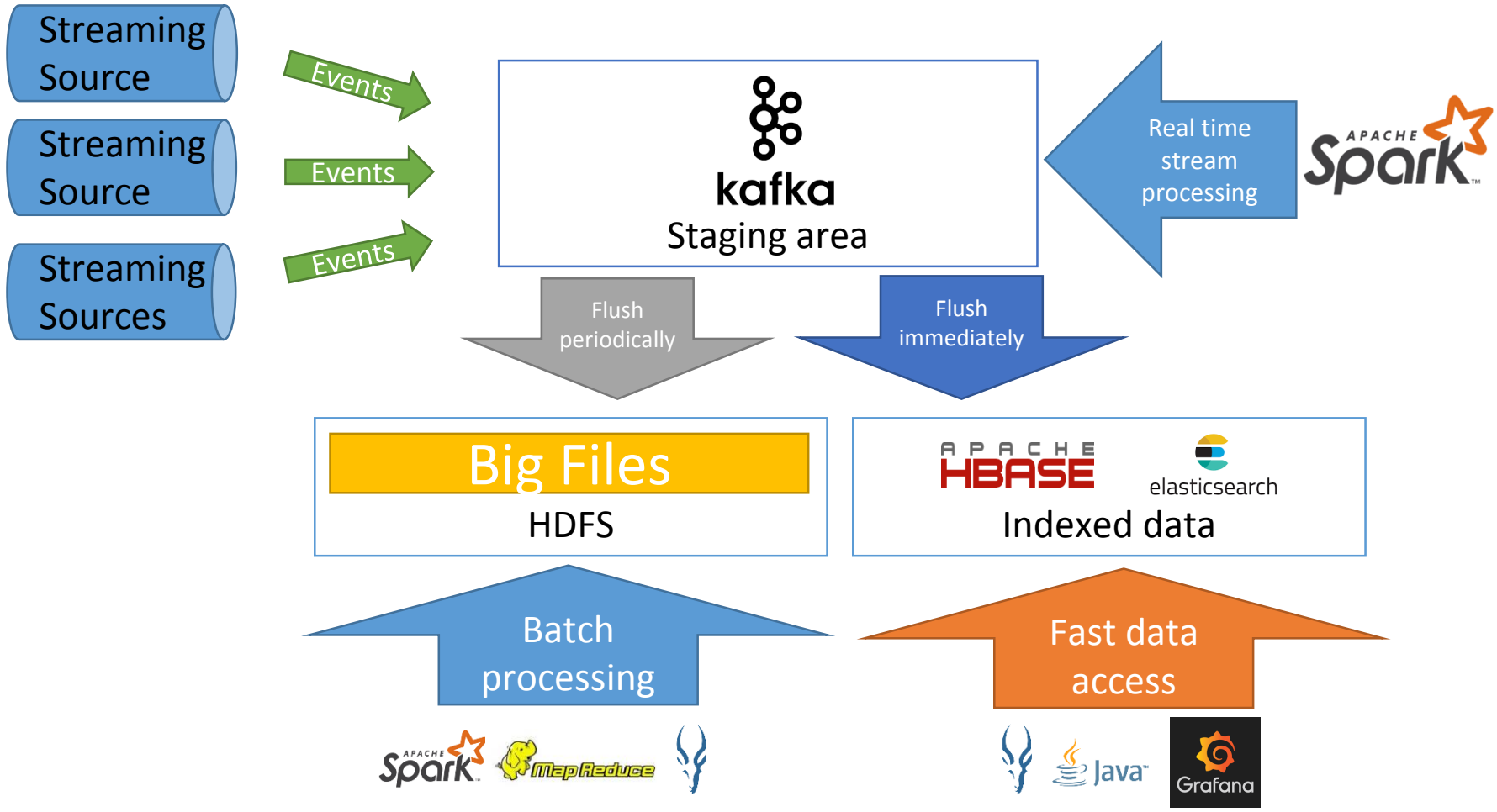
Monitoring

Visualizations

Thoughts and trends observed

Data ingestion – is a challange

- Apache Kafka becomes a standard for modern scalable architectures



Visualization

- Nothing unified provided by for the ecosystem the open source community ecosystem
  - Some efforts with Hue are being done

- Analytics
  - Jupyter Notebooks (pySpark)
  - Zeppelin (Scala Spark)

- Live data
  - ElasticSearch or InfluxDB + Grafana/Kibana (data stored for limited duration)

Apache Parquet – an efficient columnar file format for HDFS

- Internal schema with multiple data types including nested ones
- Multiple encoding applicable on per column-bases
  - RLE, Dictionary, Delta, Bit packing
- Compressions supported
  - Snappy, gzip, LZO
- Column-level statistics per each block/rowgroup

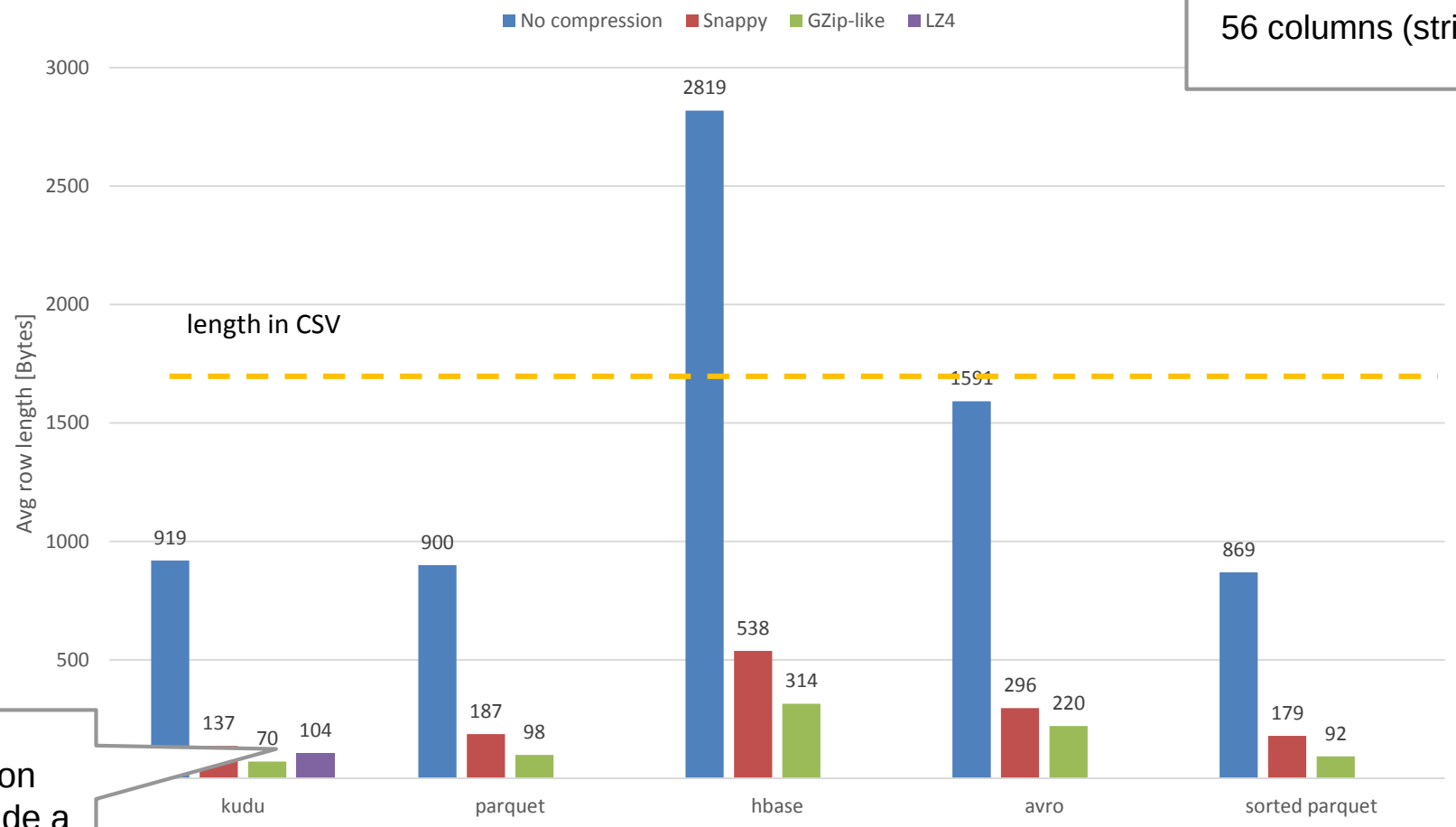
- Advantages
  - Very compact – up 10x smaller than a text-based file
  - Column (vertical) pruning –> less IO
  - Rows group (horizontal) pruning -> less IO

- Supported by most of modern big data processing frameworks
- Recommended for analytic workloads

Data packing by various formats

- and compressions

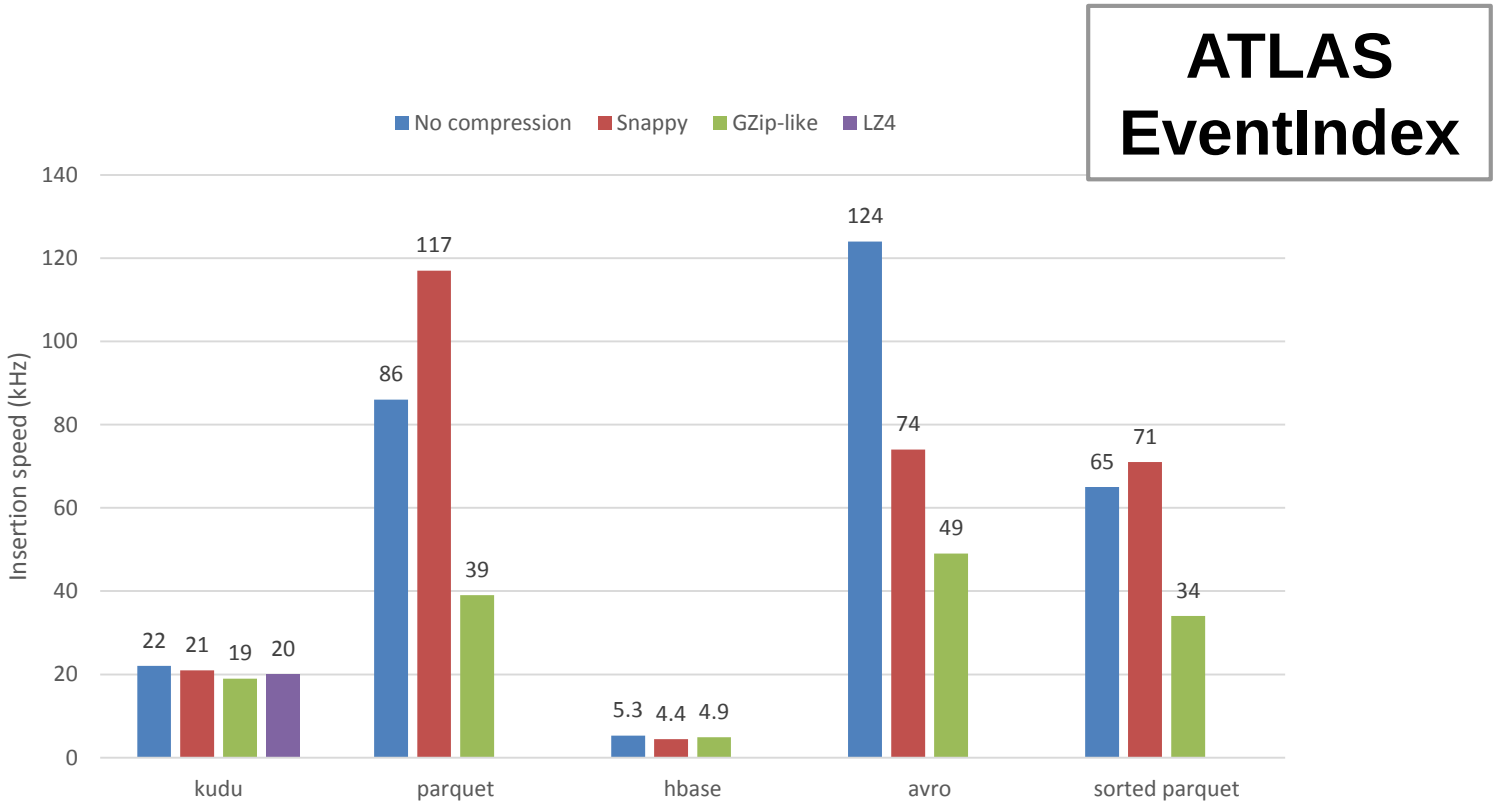
ATLAS EventIndex
56 columns (strings, ints and floats)



All compression algorithms provide a lot of savings

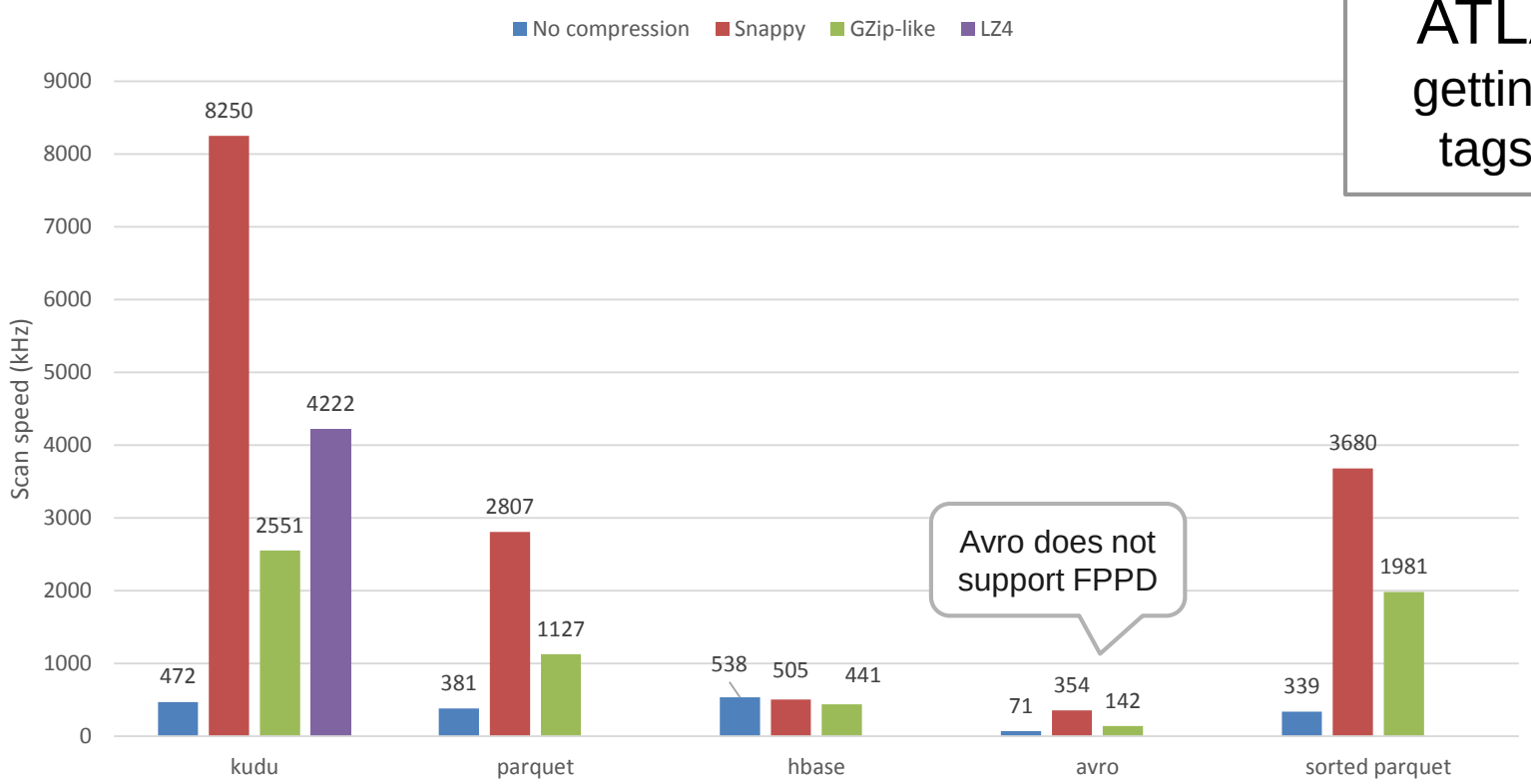
Measured insertion speed

- Per client thread (the higher the better)



Data scanning efficiency (using Impala)

- By non-PK column per scanner thread (higher the better)
- **With** filter predicate push down



ATLAS EventIndex
getting events with exact tags (**equal** predicate)

Hadoop and Spark on a private cloud?

- Appears to be a good solution when storage locality is not needed
  - Functional test and development
  - Non-IO intensive workloads
  - Reading from external storages (AFS, EOS, foreign HDFS)
- Spark clusters (without HDFS and YARN) - on containers (Kubernetes)
  - possible candidates for Spark clusters for physics data processing reading from EOS (or from remote HDFS)
  - Streaming jobs reading from Kafka

Conclusions

- Hadoop, Spark, Kafka services at CERN IT
  - Analytics, streaming, logging/controls
- BigData is growing at CERN
  - Many projects started and running
  - The service is evolving
  - Experience and community
- The technologies evolves rapidly on that field
  - Opportunities and challenges