Sanoma Big Data Migration

Sander Kieft

IBM
Data Processing
Division
San Jose, California

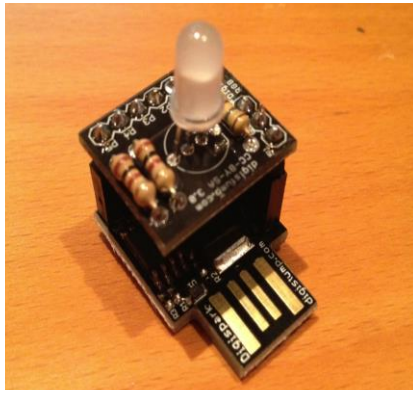
PAA
IBM

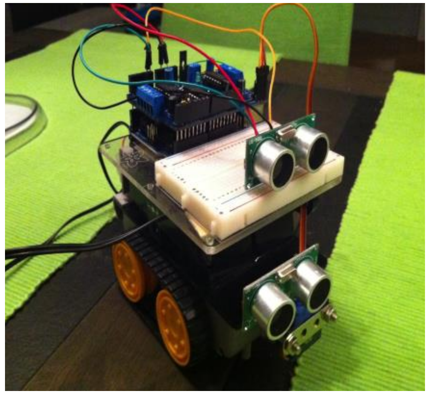
About me

- Manager Core Services at Sanoma
- Responsible for common services, including the Big Data platfor

- Work:
  - Centralized services
  - Data platform
  - Search

- Like:
  - Work
  - Water(sports)
  - Whiskey
  - Tinkering: Arduino, Raspberry PI, soldering stuff





Sanoma, Publishing and Learning company

2+100

2 Finnish newspapers

Over 100 magazines in The Netherland, Belgium and Finland

7

TV channels in Finland and The Netherlands, incl. on demand platforms

30+

Learning applications

200+

Websites

100

Mobile applications on various mobile platforms

Sanoma, Big Data use cases

- Users of Sanoma’s websites, mobile applications, online TV products generate large volumes of data
- We use this data to improve our products for our users and our advertisers
- Use cases:

Dashboards and Reporting

- Reporting on various data sources
- Self Service Data Access
- Editorial Guidance

Product Improvements

- A/B testing
- Recommenders
- Search optimizations
- Adaptive learning/tutoring

Advertising optimization

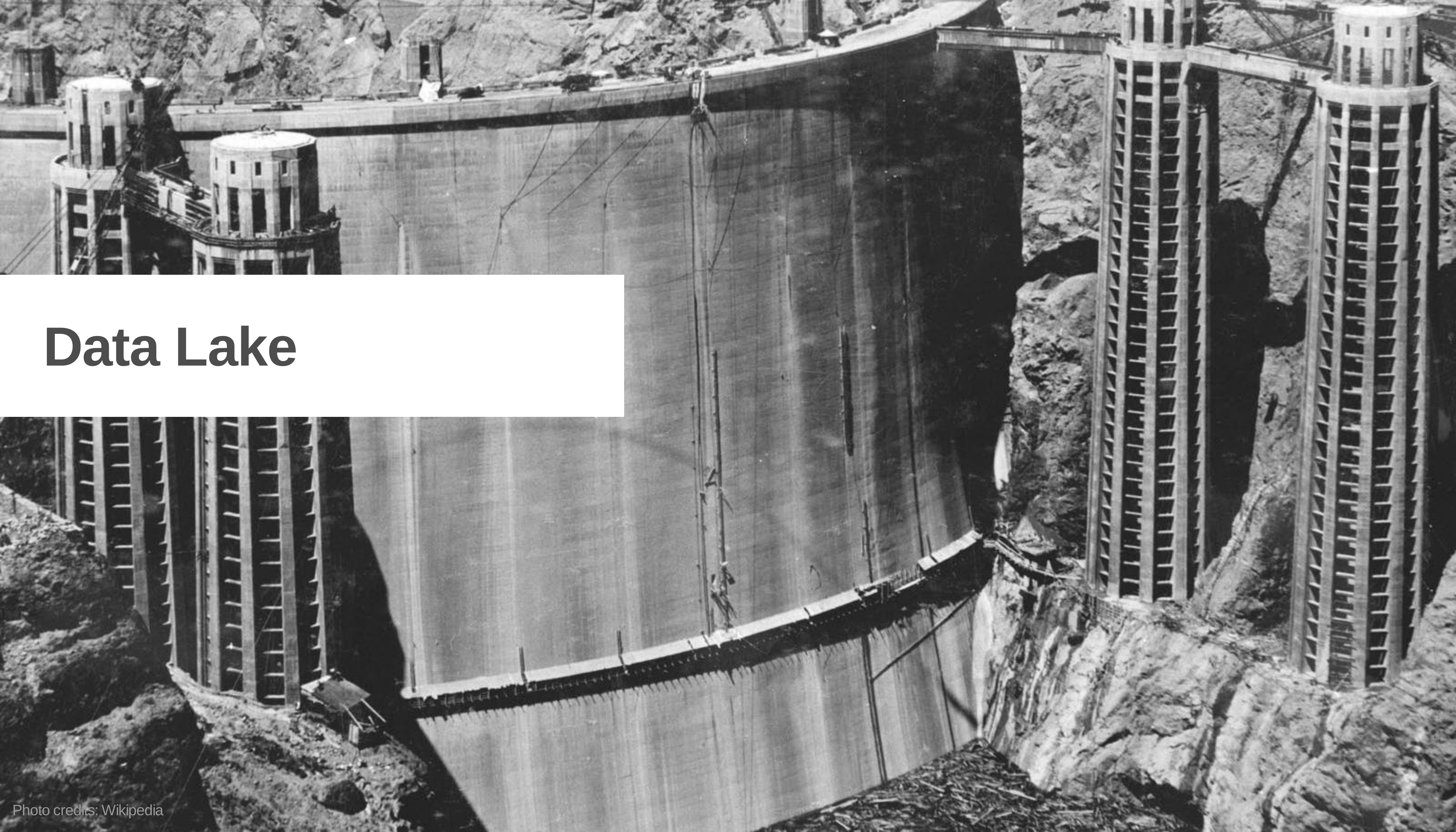
- (Re)Targeting
- Attribution
- Auction price optimization

Photo credits: kennysarmy - https://www.flickr.com/photos/kennysarmy/3207539502/

History

< 2008 | 2009 | 2010 | 2011 | 2012 | 2013 | 2014 | 2015

Google AdWords
Google Analytics
double click
hadoop
HIVE



Data Lake

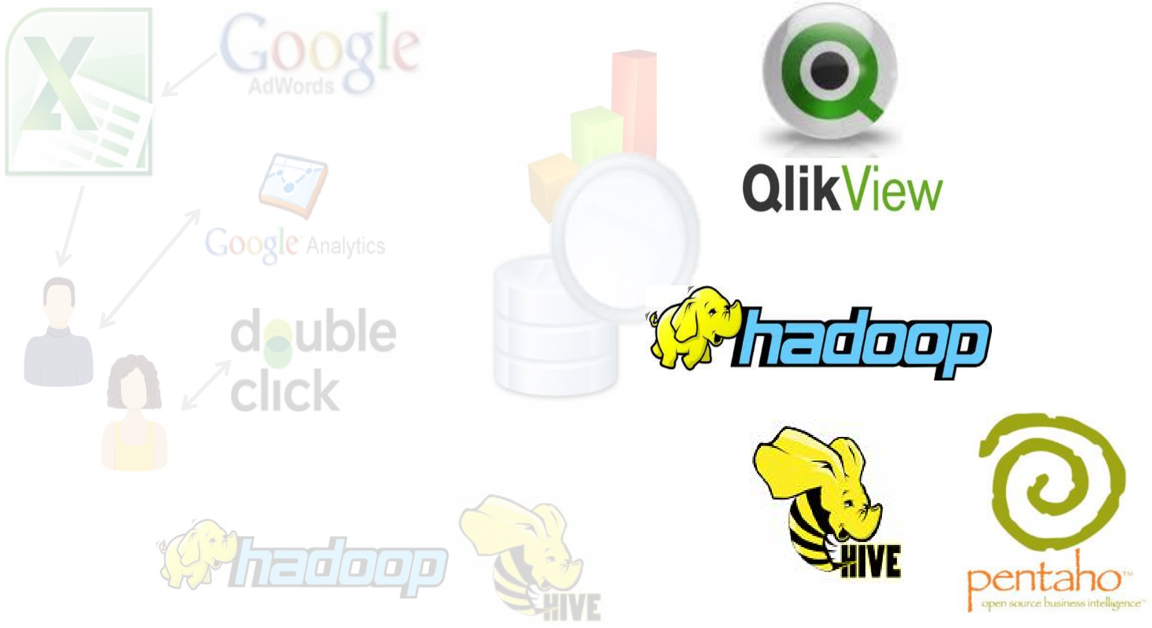
Photo credits: Wikipedia

In science, one man's noise is another man's signal.

-- Edward Ng

History

< 2008 | 2009 | 2010 | 2011 | 2012 | 2013 | 2014 | 2015

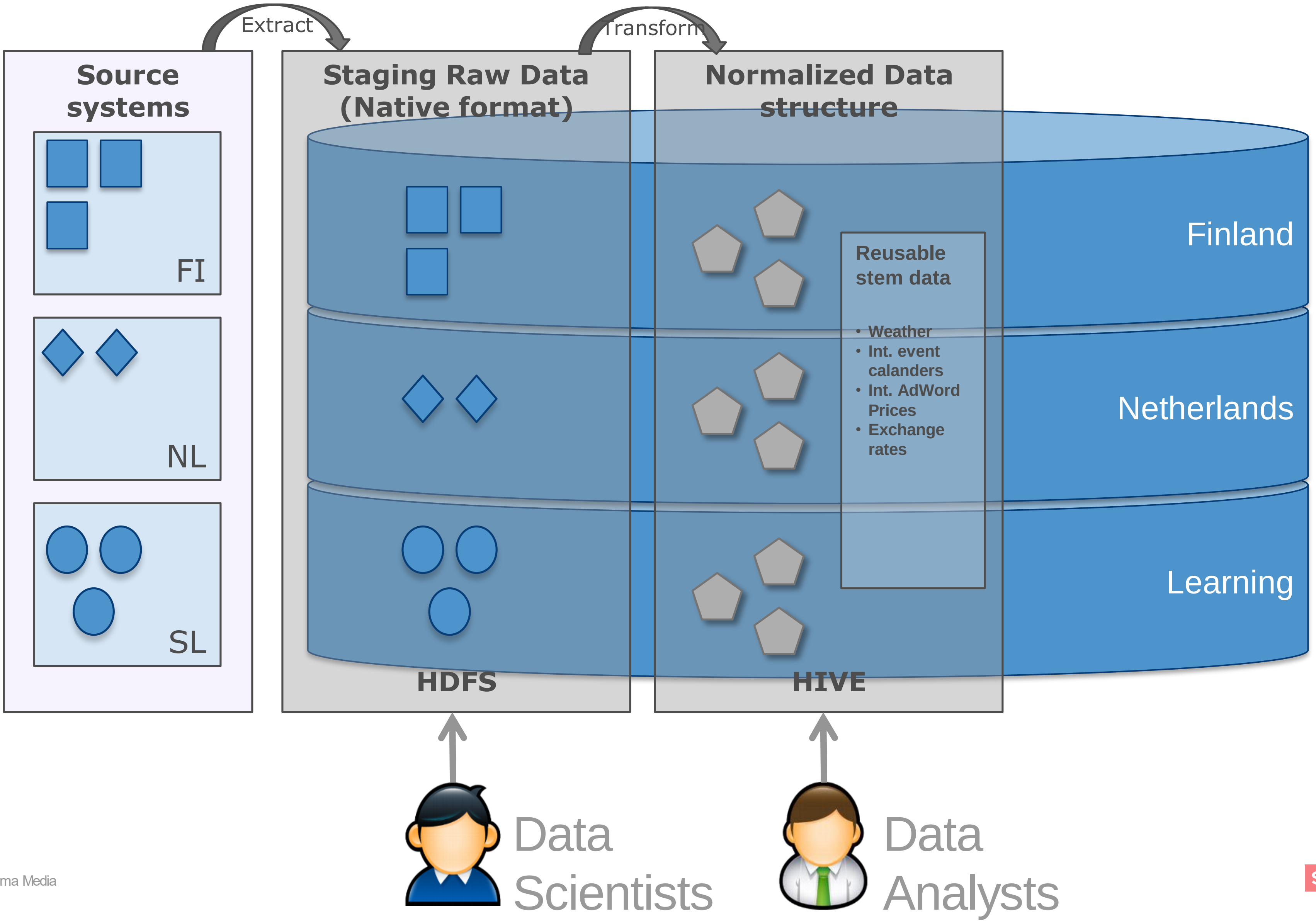


Lake filling

Self service

Photo credits: misternaxal - http://www.flickr.com/photos/misternaxal/2888791930/

Enabling self service



Extract

Transform

Source systems

FI

NL

SL

Staging Raw Data (Native format)

Normalized Data structure

Reusable stem data

- Weather
- Int. event calanders
- Int. AdWord Prices
- Exchange rates

Finland

Netherlands

Learning

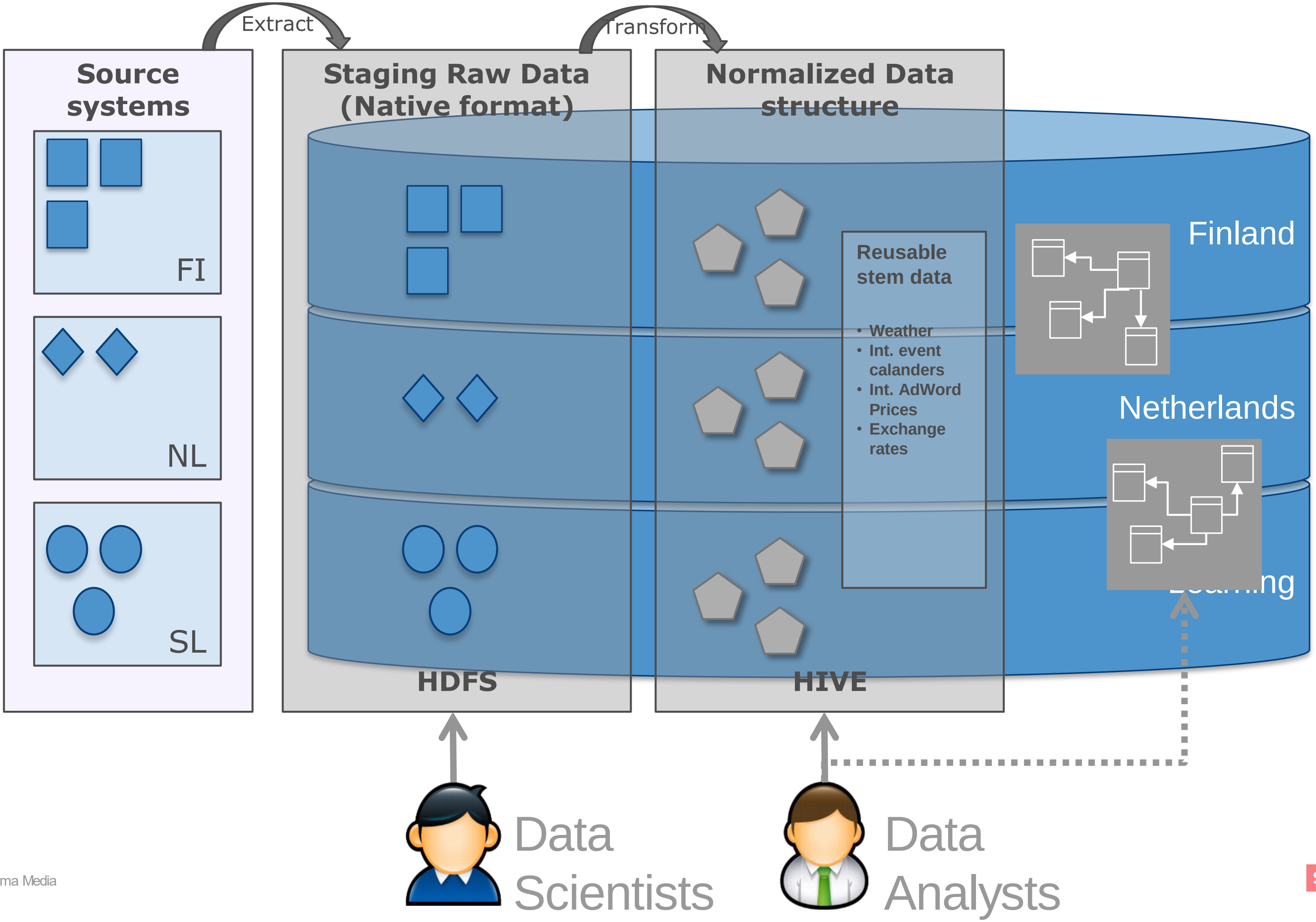
HDFS

HIVE

Data Scientists

Data Analysts

Enabling self service



Extract

Transform

Source systems

FI

NL

SL

Staging Raw Data (Native format)

HDFS

Normalized Data structure

HIVE

Reusable stem data

- Weather
- Int. event calanders
- Int. AdWord Prices
- Exchange rates

Finland

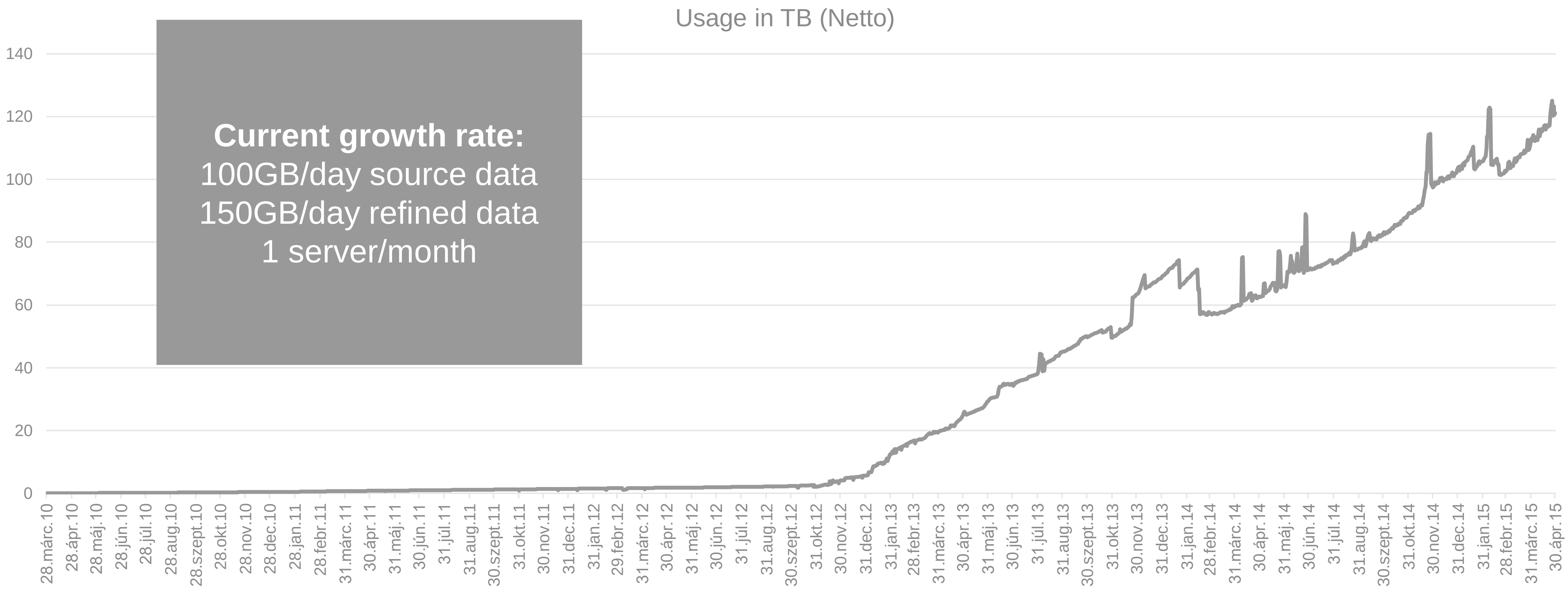
Netherlands

Learning

Data Scientists

Data Analysts

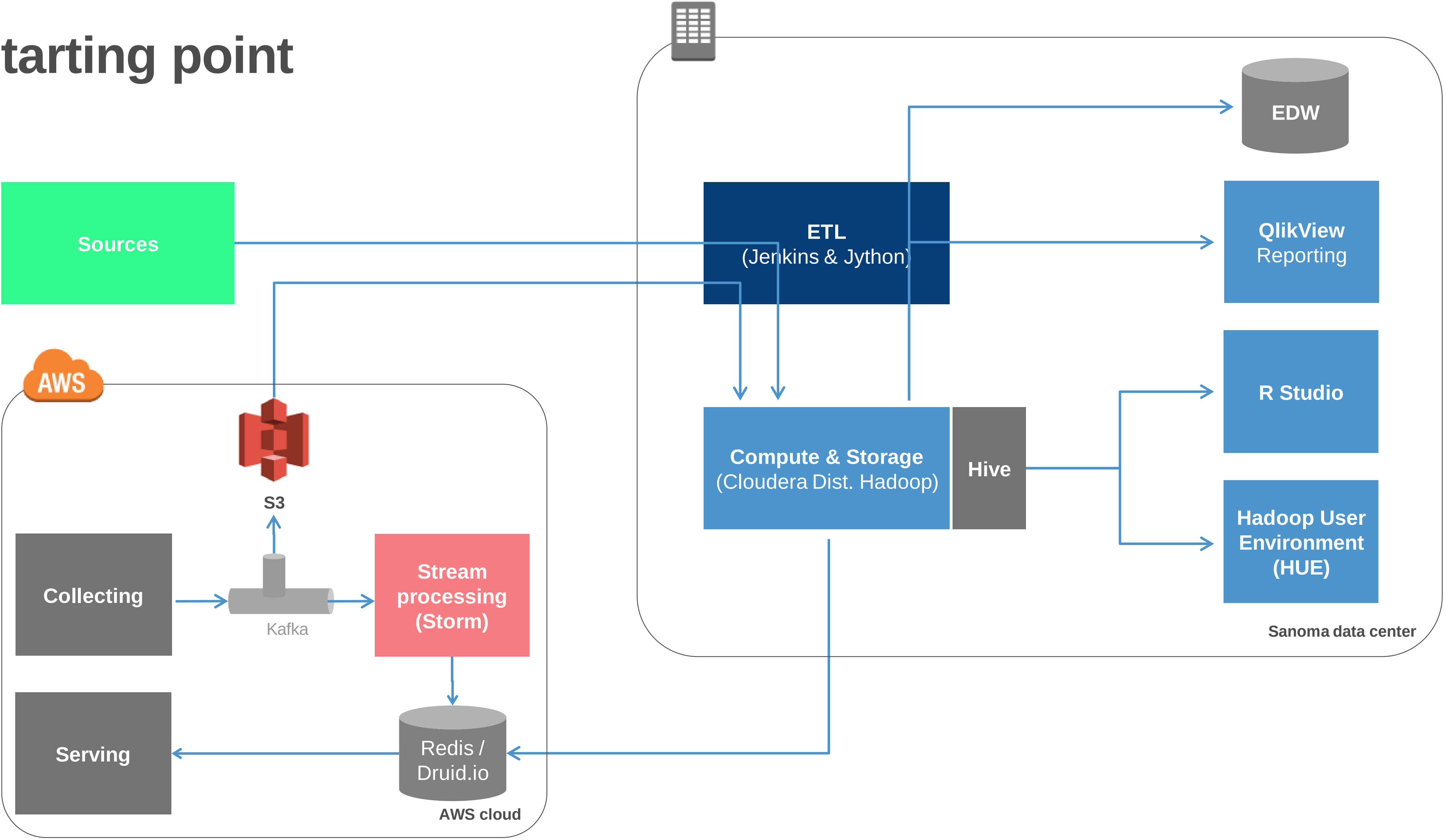
Storage growth



Current growth rate:
100GB/day source data
150GB/day refined data
1 server/month

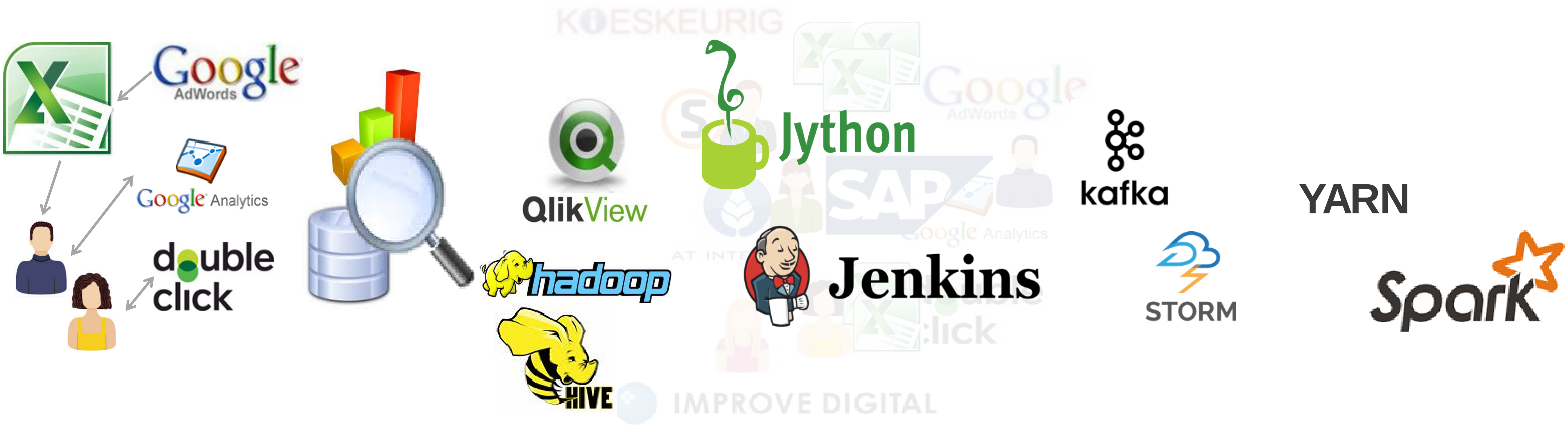
Keeps filling

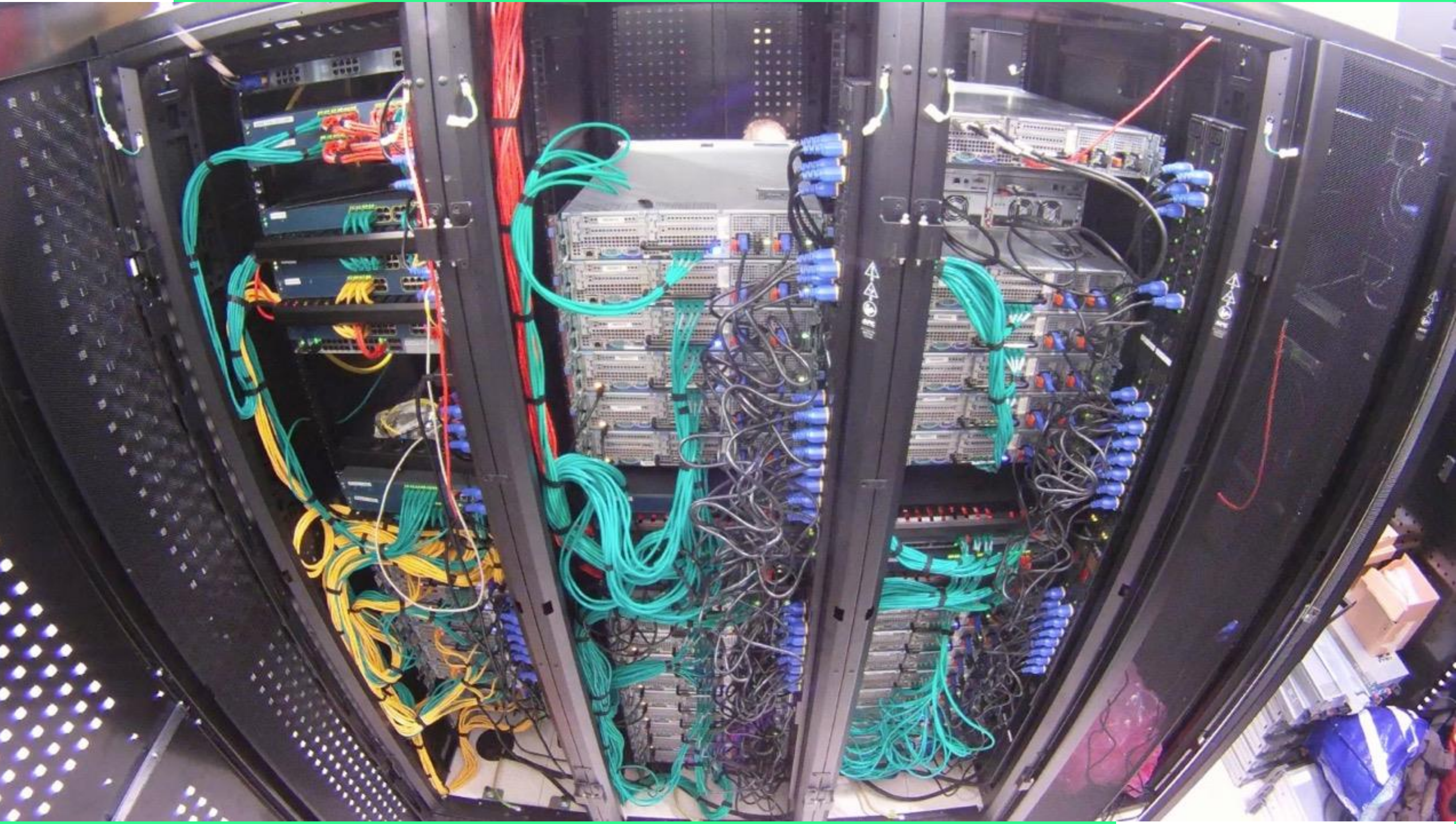
Starting point



Present

< 2008 | 2009 | 2010 | 2011 | 2012 | 2013 | 2014 | 2015





60+

NODES

720TB

CAPACITY

475TB

DATA STORED (GROSS)

180TB

DATA STORED (NETTO)

150GB

DAILY GROWTH

50+

DATA SOURCES

200+

DATA PROCESSES

3000+

DAILY HADOOP JOBS

200+

DASHBOARDS

275+

AVG MONTHLY DASHBOARD USERS

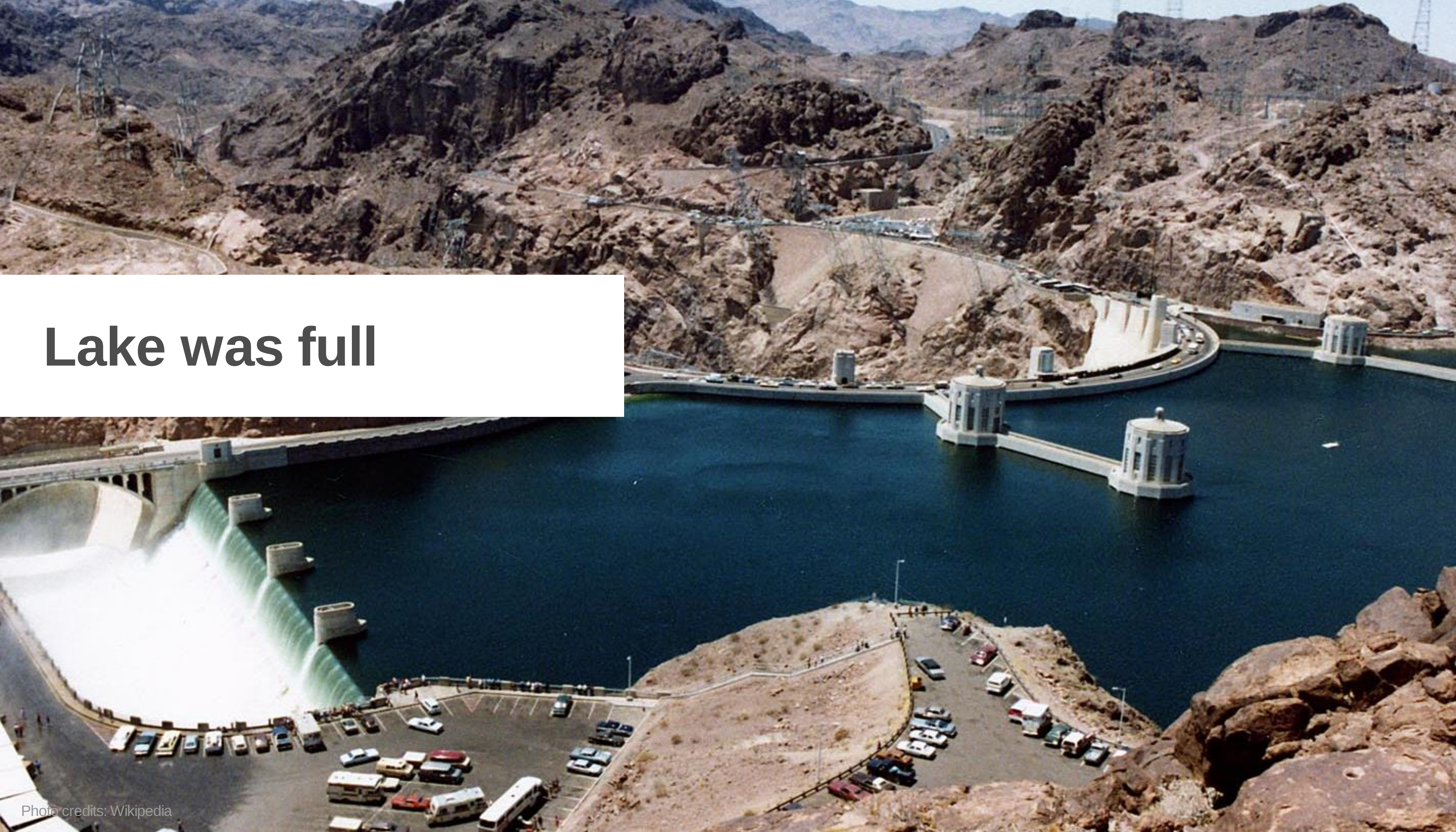
Challenges

Positives

- Users have been steadily increasing
- Demand for ..
  - more real time processing
  - faster data availability
  - higher availability (SLA)
  - quicker availability of new versions
  - specialized hardware (GPU's/SSD's)
  - quicker experiments (Fail Fast)

Negatives

- Data Center network support end of life
- Outsourcing of own operations team
- Version upgrades harder to manage
- Poor job isolation between test, dev, prod and interactive, etl and data science workloads
- Higher level of security and access control

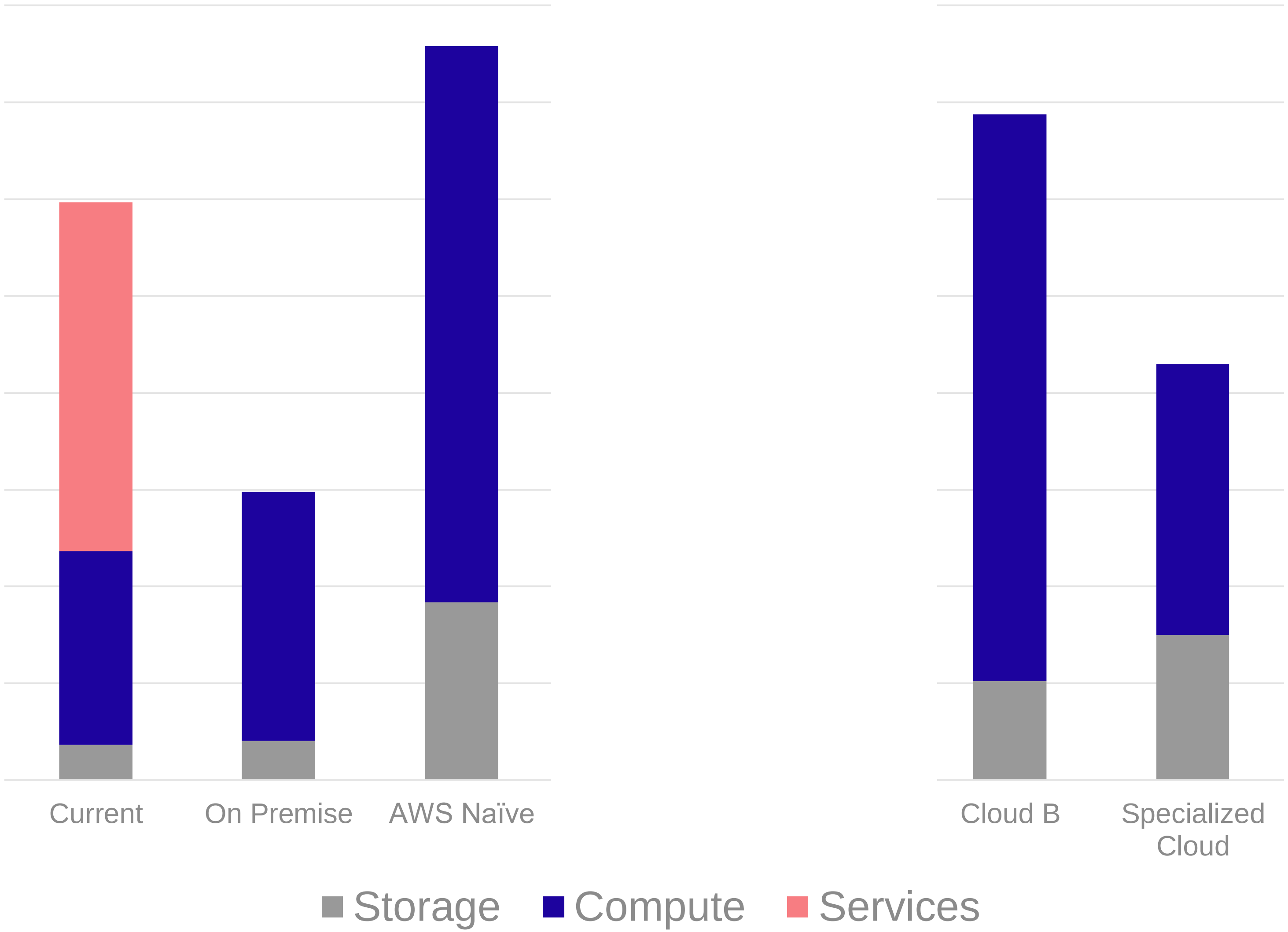


Lake was full

Photo credits: Wikipedia

Big Data Hosting Options

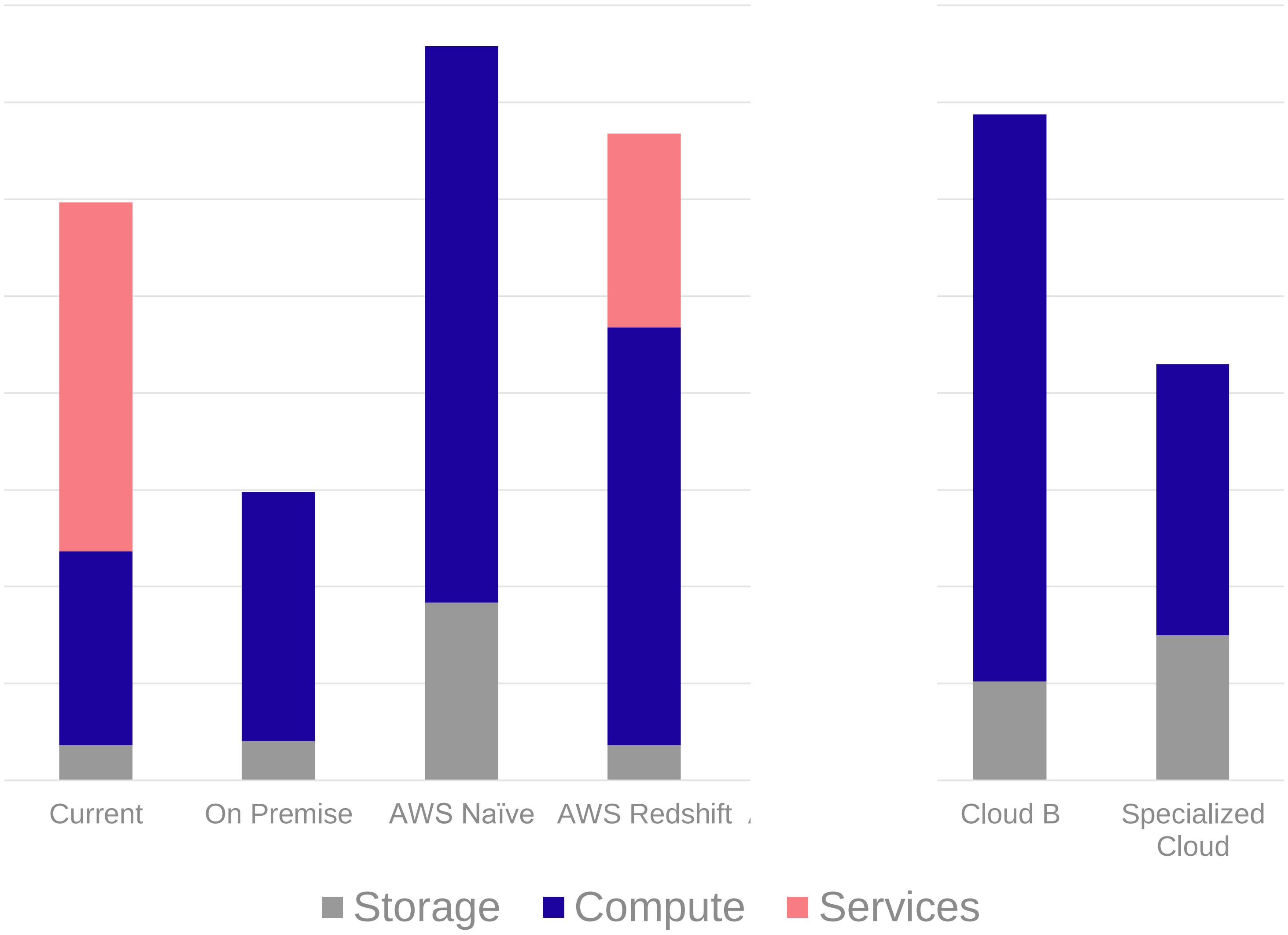
- On Premise
  - New Hardware
- Generic Cloud
  - Provider A
  - Provider B
- Specialized Cloud



Big Data Hosting Options

- On Premise
  - New Hardware
- Generic Cloud
  - Provider A
  - Provider B
- Specialized Cloud

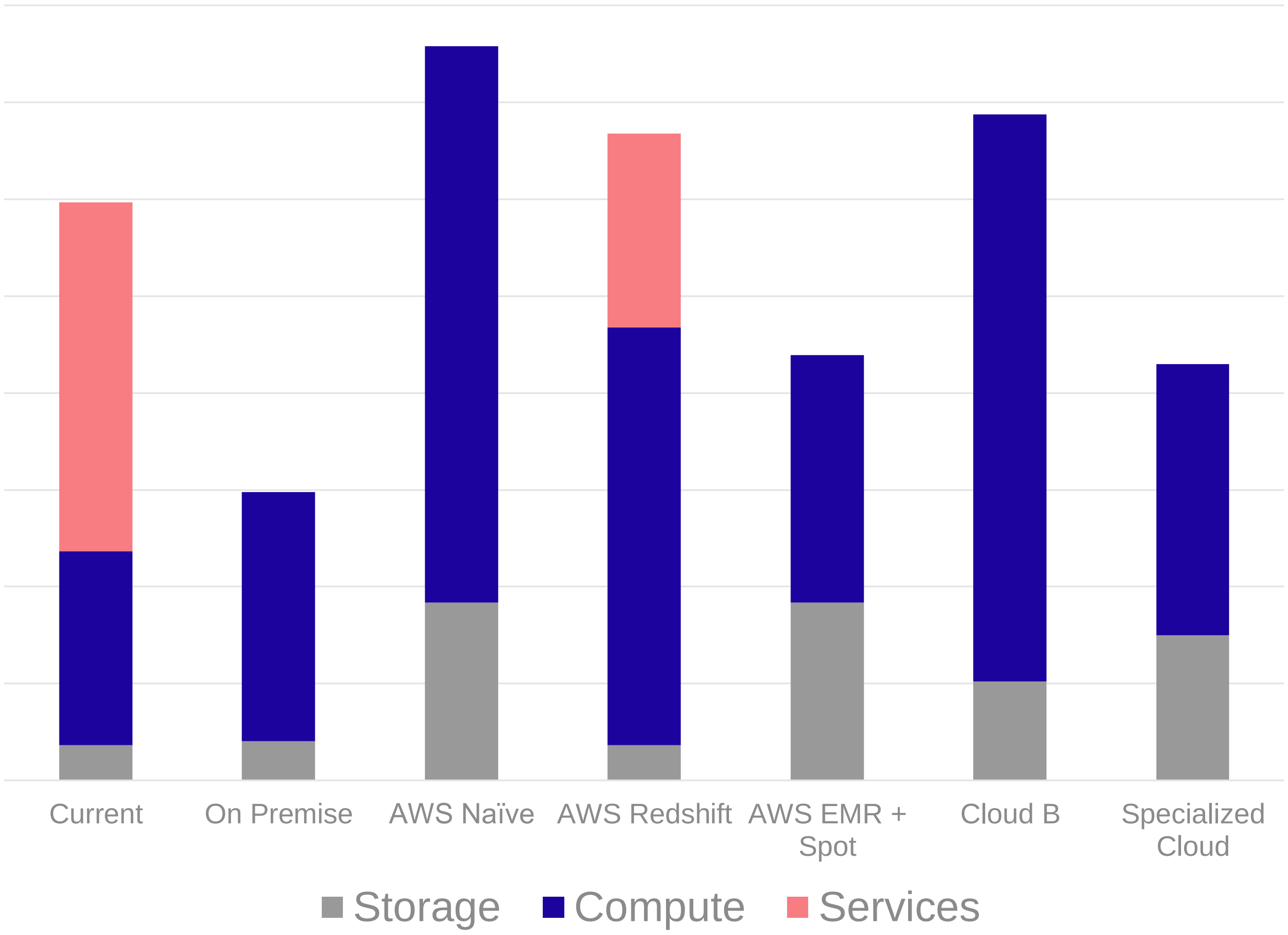
- Adding data services from cloud provider to the comparison



Big Data Hosting Options

- On Premise
  - New Hardware
- Generic Cloud
  - Provider A
  - Provider B
- Specialized Cloud

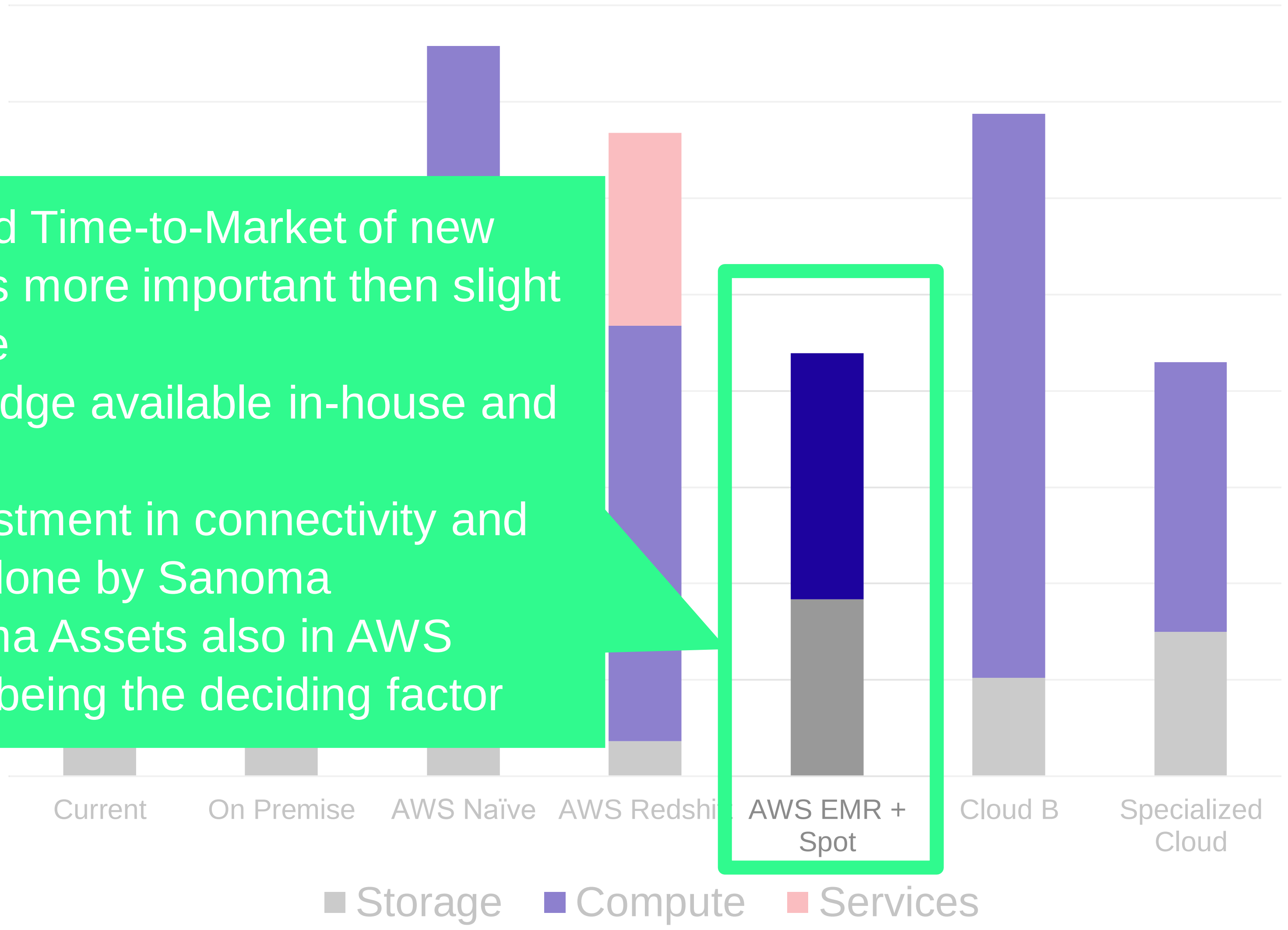
- Adding data services from cloud provider to the comparison
- Replacing portion of processing with Spot Pricing



Big Data Hosting Options

- On Premise
  - New Hardware
- Generic Cloud
  - Provider A
  - Provider B
- Specialized Cloud

- Adding data servi
  provider to the comparison
- Replacing portion of processing with Spot Pricing

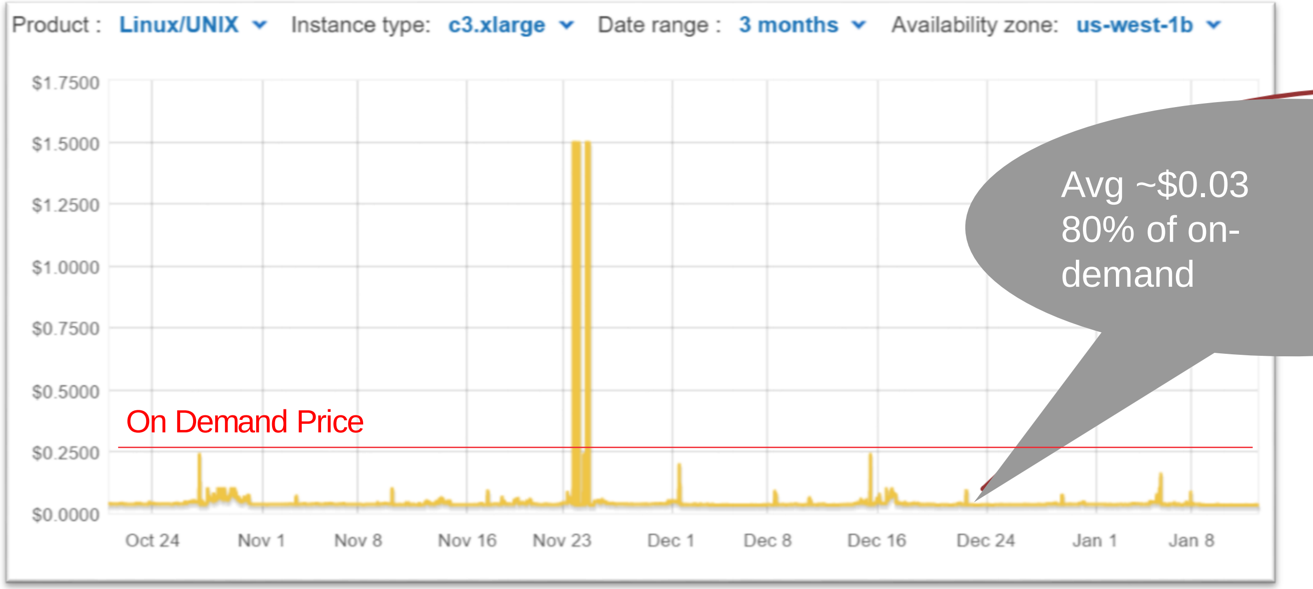


- Flexibility and Time-to-Market of new data services more important then slight cost increase
- More knowledge available in-house and in market
- Already investment in connectivity and automation done by Sanoma
- Other Sanoma Assets also in AWS
- Spot pricing being the deciding factor

Amazon – Instance Buying Options

	c3.xlarge, 4 vCPU, 7.5 GiB
▪ On-demand Instances: hourly pricing	$0.239 per Hour
▪ Reserved Instances: – Up to 75% cheaper than on-demand pricing – 1 – 3 year commitment – (Large) upfront costs; typical breakeven at 50-80% utilization	$0.146 – $0.078 per Hour
▪ Spot Instances: – AWS sells excess capacity to highest bidder – Hourly commitments at a price you name – Can be up to 90% cheaper that on-demand pricing	~ $0.03 per Hour (> 80% reduction)

Amazon - Spot Prices



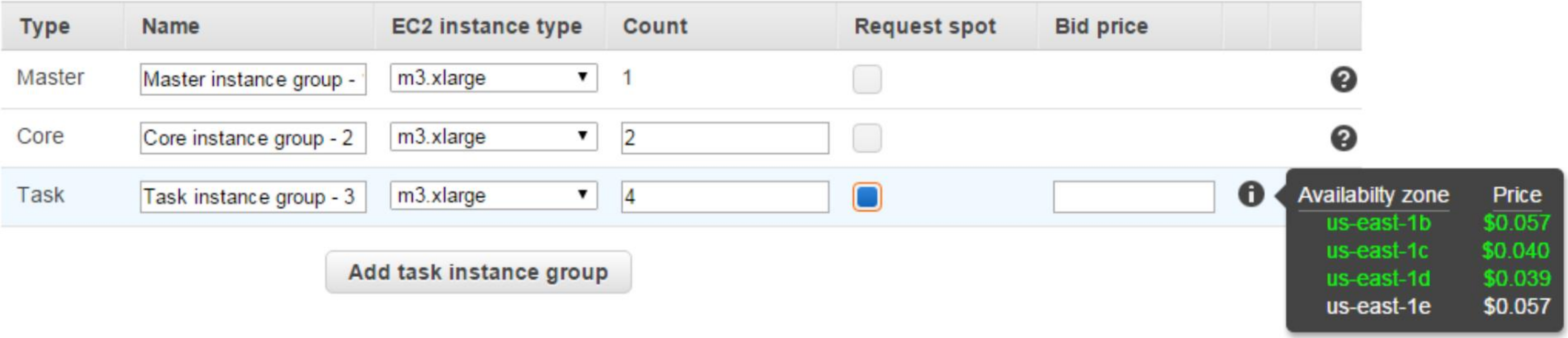
Product : Linux/UNIX Instance type: c3.xlarge Date range : 3 months Availability zone: us-west-1b

Avg ~$0.03
80% of on-demand

- AWS determines market price based on supply and demand
- Instance is allocated to highest bidder
- Bidder pays market price
- Allocated instance is terminated (with a 2 minute warning) when market price increases above your bid price
- Diversification of instance families, instance types, availability zones increases continuity

- Spot Instances can take a while to provision with a different workflow than a traditional on-demand model.
- Guard against termination with adequate pricing, but don't try and prevent it. Automation is key.

Amazon – Spot Pricing



Type	Name	EC2 instance type	Count	Request spot	Bid price
Master	Master instance group -	m3.xlarge	1	□	
Core	Core instance group - 2	m3.xlarge	2	□	
Task	Task instance group - 3	m3.xlarge	4	■	

Availabilty zone	Price
us-east-1b	$0.057
us-east-1c	$0.040
us-east-1d	$0.039
us-east-1e	$0.057

Via Console or via cli:

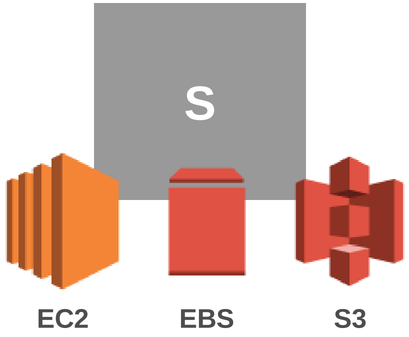
```
aws emr create-cluster --name "Spot cluster" --ami-version 3.3 InstanceGroupType=MASTER,
InstanceType=m3.xlarge,InstanceCount=1, InstanceGroupType=CORE,
BidPrice=0.03,InstanceType=m3.xlarge,InstanceCount=2 InstanceGroupType=TASK,
BidPrice=0.10,InstanceType=m3.xlarge,InstanceCount=3
```

Destination Amazon

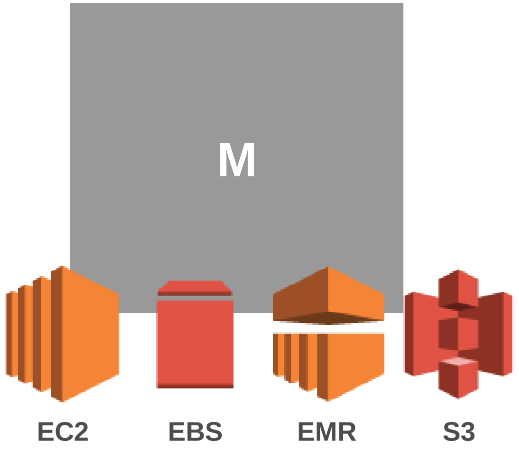
Photo credits: Amauri Aguiar- https://www.flickr.com/photos/amauriaguiar/3204101173/

Amazon Migration Scenario’s

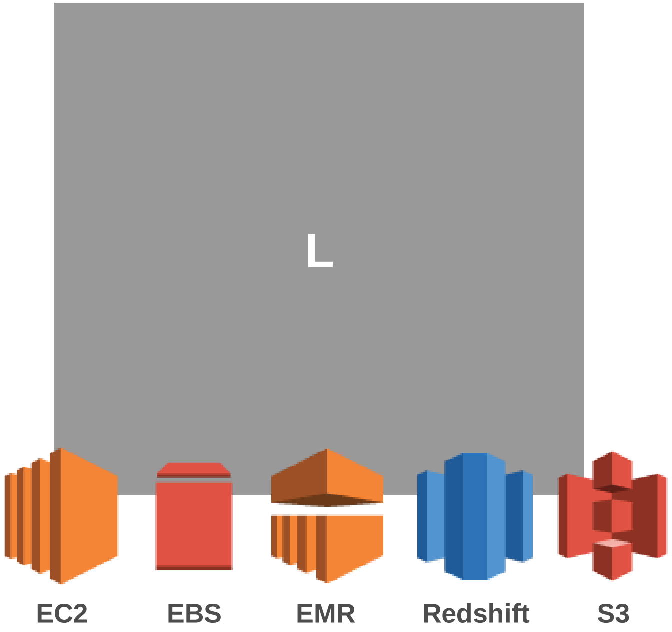
- Three scenario’s evaluated for moving



- All services on EC2
- Only source data on S3



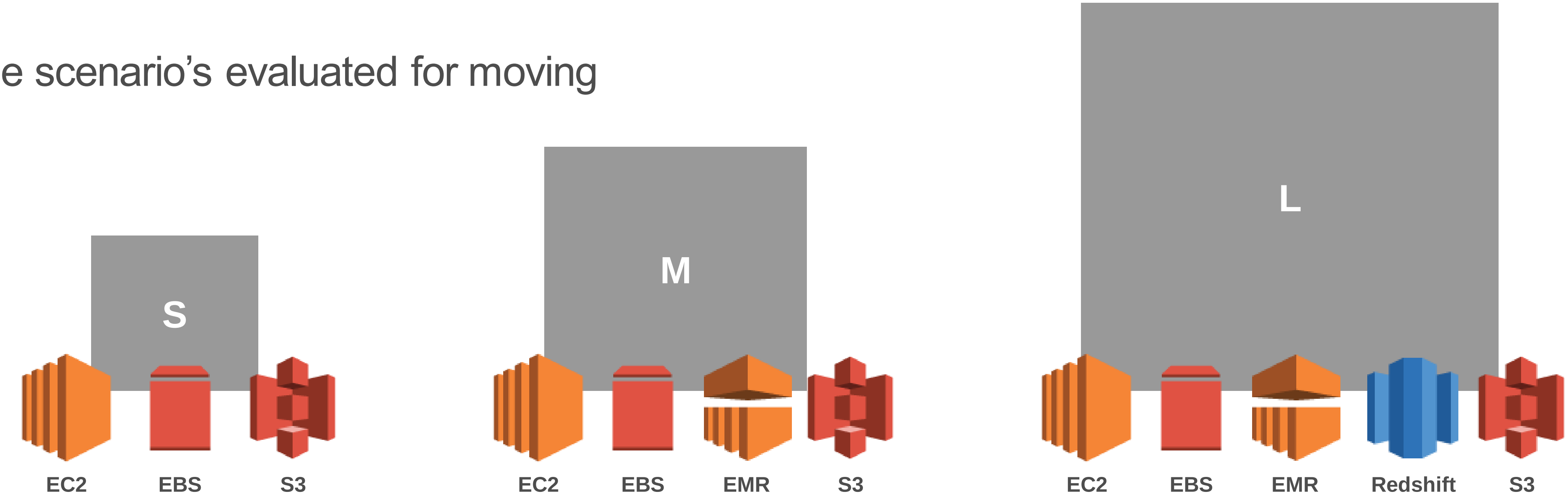
- All data on S3
- EMR for Hadoop
- EC2 only for utility services not provided by EMR



- All data on S3
- EMR for Hadoop
- Interactive querying workload is moved to Redshift instead of Hive

Amazon Migration Scenario’s

- Three scenario’s evaluated for moving



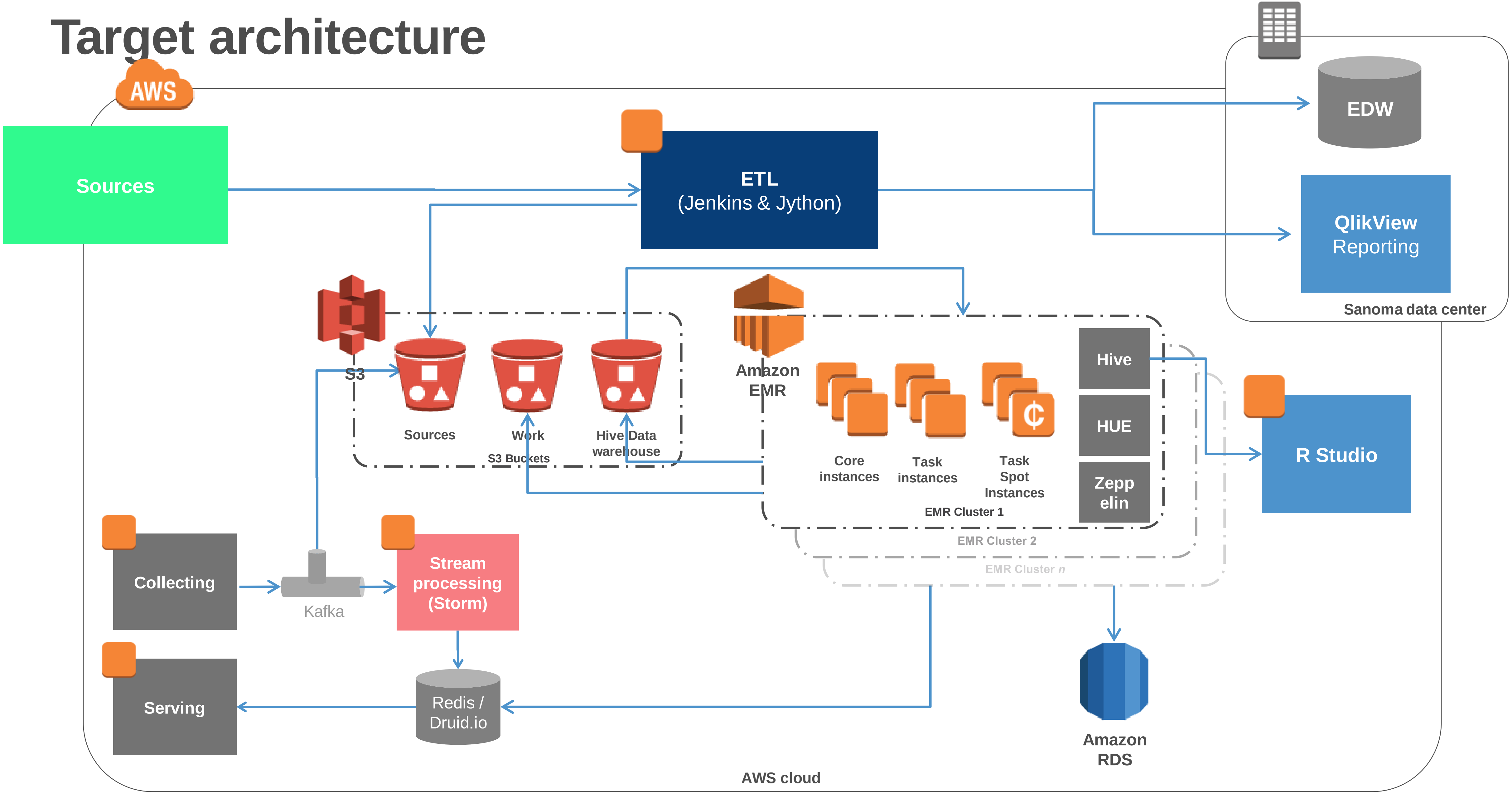
- All services on EC2
- Only source data on S3

- All data on S3
- EMR for Hadoop
- EC2 only for utility services not provided by EMR

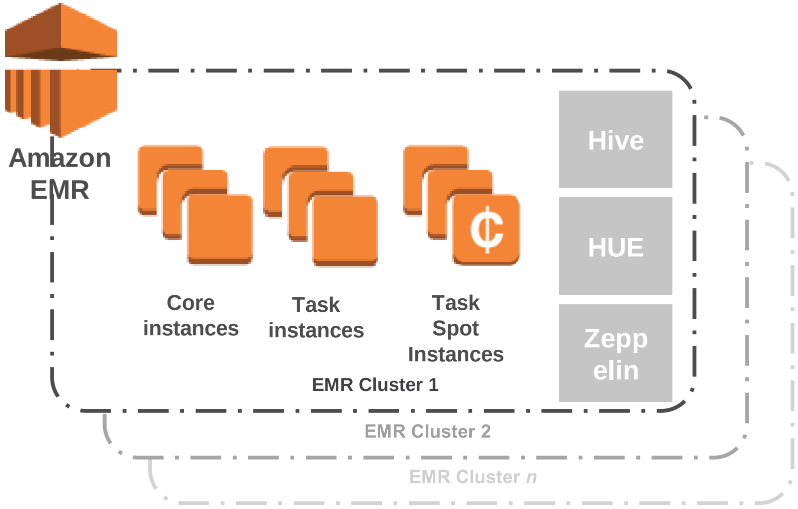
- All data on S3
- EMR for Hadoop
- Interactive querying workload is moved to Redshift instead of Hive

Easier to leverage spot pricing, due to data on S3

Target architecture



Node types



- Basic Cluster

1x	MASTER	m4.2xlarge
25x	CORE	d2.xlarge
40x	TASK	r3.2xlarge

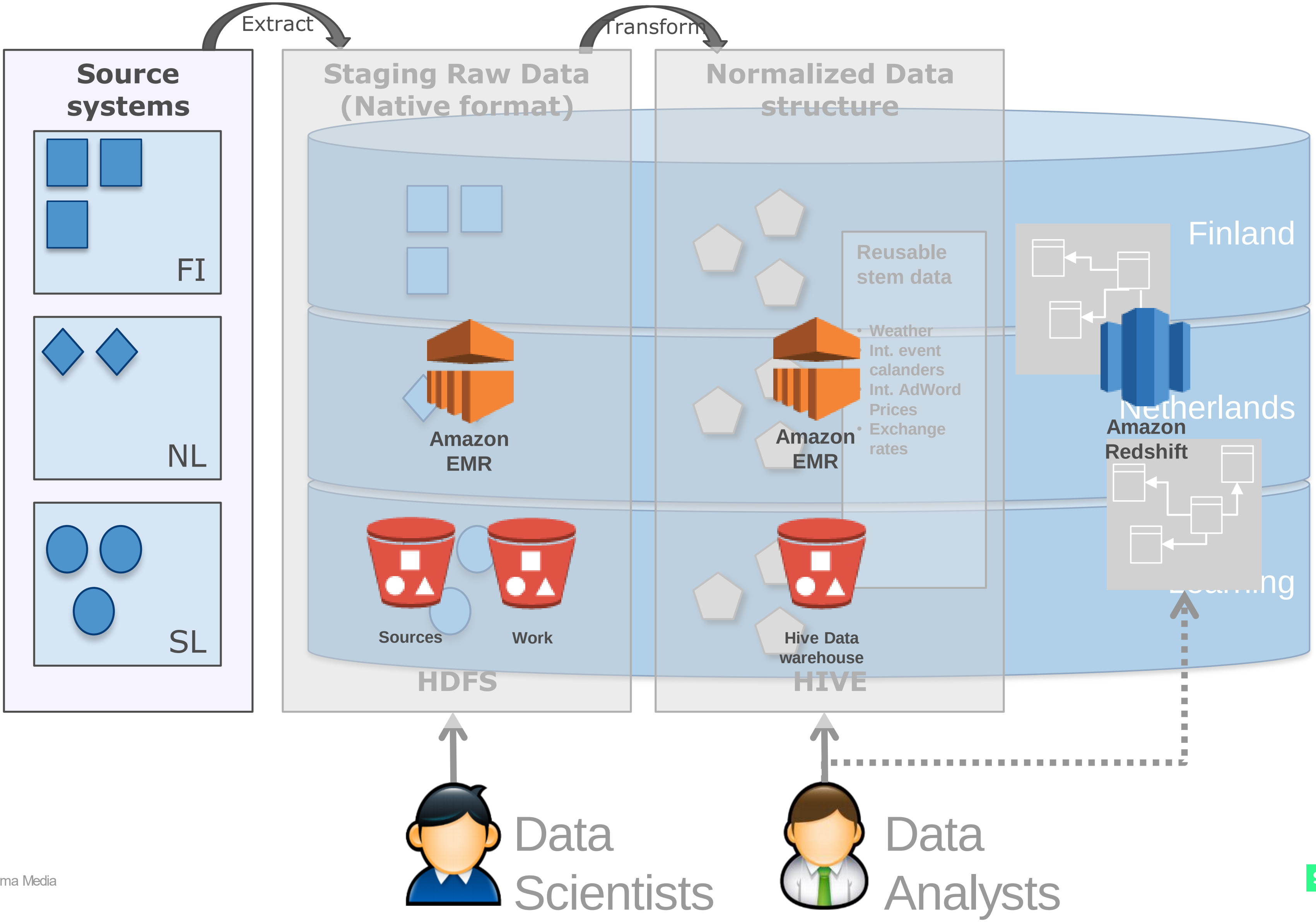
- Basic Cluster with spot pricing:

1x	MASTER	m4.2xlarge
25x	CORE	d2.xlarge
20x	TASK	r3.2xlarge (On Demand)
40x	TASK	r3.2xlarge (Spot Pricing)

Possible bidding strategies:

- Bid on-demand price
- Diversification of
  - instance families
  - instance types
  - availability zones
- Bundle and rolling prices
  - 5x 0,01, 5x 0,02, 5x 0,03

Enabling self service – In Amazon



Extract

Transform

Source systems

FI

NL

SL

Staging Raw Data (Native format)

Amazon EMR

Sources

Work

HDFS

Normalized Data structure

Reusable stem data

- Weather
- Int. event calanders
- Int. AdWord Prices
- Exchange rates

Amazon EMR

Hive Data warehouse

HIVE

Finland

Netherlands

Amazon Redshift

...ng

Data Scientists

Data Analysts

Migration

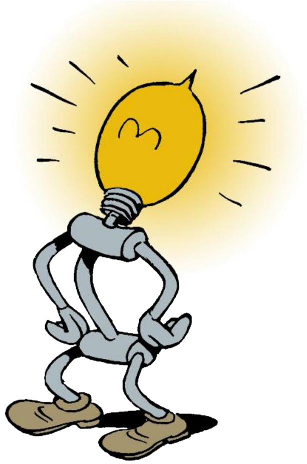
Migration

- Migration of the data from HDFS to S3 took long time
  - Large volume of data
  - Migration of source data & data warehouse

- Using EMR required more rewrites then initially planned
  - Due to network isolation of EMR Cluster it’s harder to initiate jobs from outside the cluster
  - Jobs had to be rewritten to mitigate side effects EMRFS
  - INSERT OVERWRITE TABLE xx AS SELECT xx Has different behavior

- Data formats Hive
  - Hive on EMR doesn’t support RC-files
  - Had to convert our Data Warehouse to ORC

Learnings



- We’re almost done with the migration. Running in parallel now.
- Setup solves our challenges, some still require work
- Missing Cloudera Manager for small config changes and monitoring
- EMR not ideal for long running clusters

S3

- Bucket structure impacts performance
- Setup access control, human error will occur
- Uploading data takes time. Start early!
  - Check Snowball or new upload service

EMR

- Check data formats! RC vs ORC/Parquet
- Start jobs from master node
- Break up long running jobs, shorter independent
- Spot pricing & pre-empted nodes /w Spark
- HUE and Zeppelin meta data on RDS
- Research EMR FS behavior for your use case

S3 Bucket Structure

- Throughput optimization
- S3 automatically partitions based upon key prefix

Bucket: example-hive-data

Object keys:

- warehouse/weblogs/2016-01-01/a.log.bz2
- warehouse/advertising/2016-01-01/a.log.bz2
- warehouse/analytics/2016-01-01/a.log.bz2
- warehouse/targeting/2016-01-01/a.log.bz2

Partition Key: example-hive-data/w

Bucket: example-hive-data

Object keys:

- weblogs/2016-01-01/a.log.bz2
- advertising/2016-01-01/a.log.bz2
- analytics/2016-01-01/a.log.bz2
- targeting/2016-01-01/a.log.bz2

example-hive-data/w
Partition Keys: example-hive-data/a
example-hive-data/t

Spot pricing & pre-empted nodes /w Spark

- Spark:
  - Massively parallel
  - Uses DAGs instead of mapreduce for execution
  - Minimizes I/O by storing data in RDDs in memory
  - Partitioning-aware to avoid network-intensive shuffle
  - Ideal for iterative algorithms

- We use spark a lot for Data Science
- ETL Processing slowly moving to Spark too

- Problem with Spark and EMR:
  - No control over the node where Application Master lands
  - Spark Executors are termination resilient, master is not.

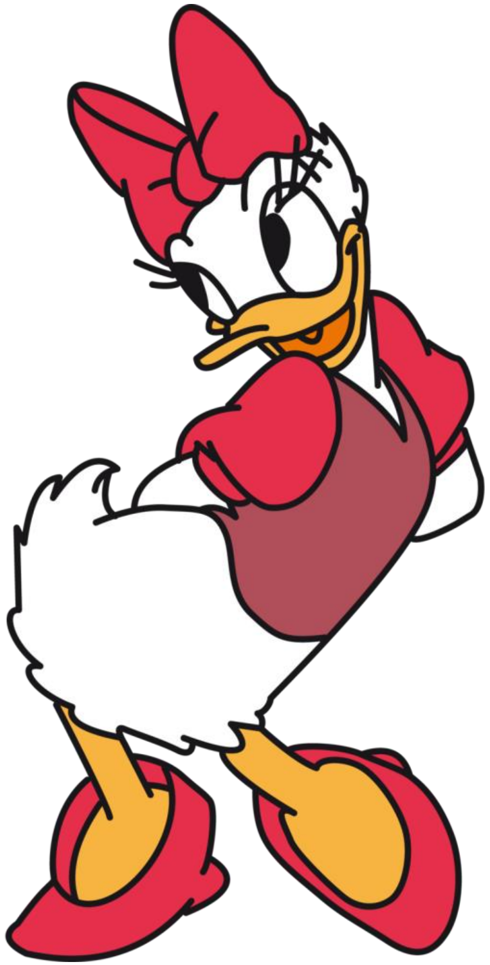
- Possible solutions:
  - Run separate cluster for Spark workload
  - Assign node labels, but current implementation is crude and is exclusive

Conclusion

- Amazon is a wonderful place to run your Big Data infrastructure
- It’s flexible, but costs can grow rapidly
- Many options for cost control available, but might impact architecture

- Take your time testing and validating your setup
  - If you have time, rethink whole setup
  - No time, move as-is first and then start optimizing

- Much faster to iterate your solution when everything is at Amazon then partly AWS/On Premise



sanoma

Thank you! Questions?

Twitter: @skieft