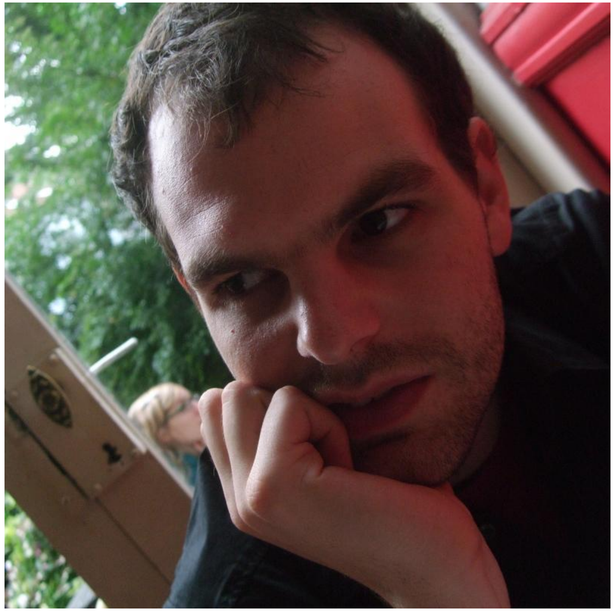
A real-time Lambda Architecture using Hadoop & Storm

NoSQL Matters Cologne 2014 by Nathan Bijnens

technicolor

virdata

Speaker



Nathan Bijnens

Big Data Engineer @ Virdata

@nathan_gs

technicolor
virdata

Computing Trends

Past	Current
Computation (CPUs) Expensive	Computation Cheap (Many Core Computers)
Disk Storage Expensive	Disk Storage Cheap (Cheap Commodity Disks)
DRAM Expensive	DRAM / SSD Getting Cheap
Coordination Easy (Latches Don't Often Hit)	Coordination Hard (Latches Stall a Lot, etc)

Source: Immutability Changes Everything - Pat Helland, RICON2012

Credits

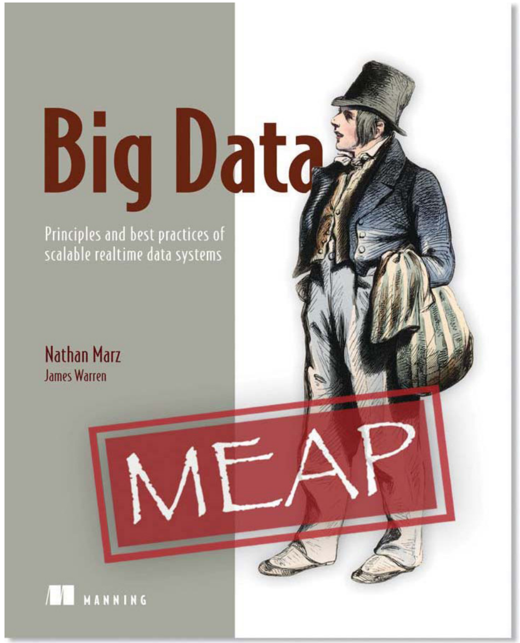
Nathan Marz

- Ex-Backtype & Twitter
- Startup in Stealthmode

Creator of

- Storm
- Cascalog
- ElephantDB

Coined the term Lambda Architecture.


Big Data
Principles and best practices of scalable realtime data systems
Nathan Marz
James Warren
MEAP
MANNING

manning.com/marz

technicolor
virdata

a Data System

Data is more than Information

Not all information is equal.

Some information is derived from other pieces of information.

technicolor virdata

Data is more than Information

Eventually you will reach the most ‘raw’ form of information.

This is the information you hold true, simply because it exists.
Let’s call this ‘data’, very similar to ‘event’.

technicolor
virdata

Events: Before

Events used to **manipulate** the master data.

Events: After

Today, events **are** the master data.

Data System

Let’s store **everything**.

Data System

Data is **Immutable**.

Data System

Data is **Time Based**.

Capturing change

Traditionally

```
INSERT INTO contact (name, city) VALUES (‘Nathan’, ‘Antwerp’)
UPDATE contact SET city = ‘Cologne’ WHERE name = ‘Nathan’
```

Capturing change

in a Data System

INSERT INTO contact (name, city, timestamp) VALUES ('Nathan', 'Antwerp', 2008-10-11 20:00Z)
INSERT INTO contact (name, city, timestamp) VALUES ('Nathan', 'Cologne', 2014-04-29 10:00Z)

Query

The data you query is often **transformed**, aggregated, ...

Rarely used in it's original form.

Query

Query = function (all data)

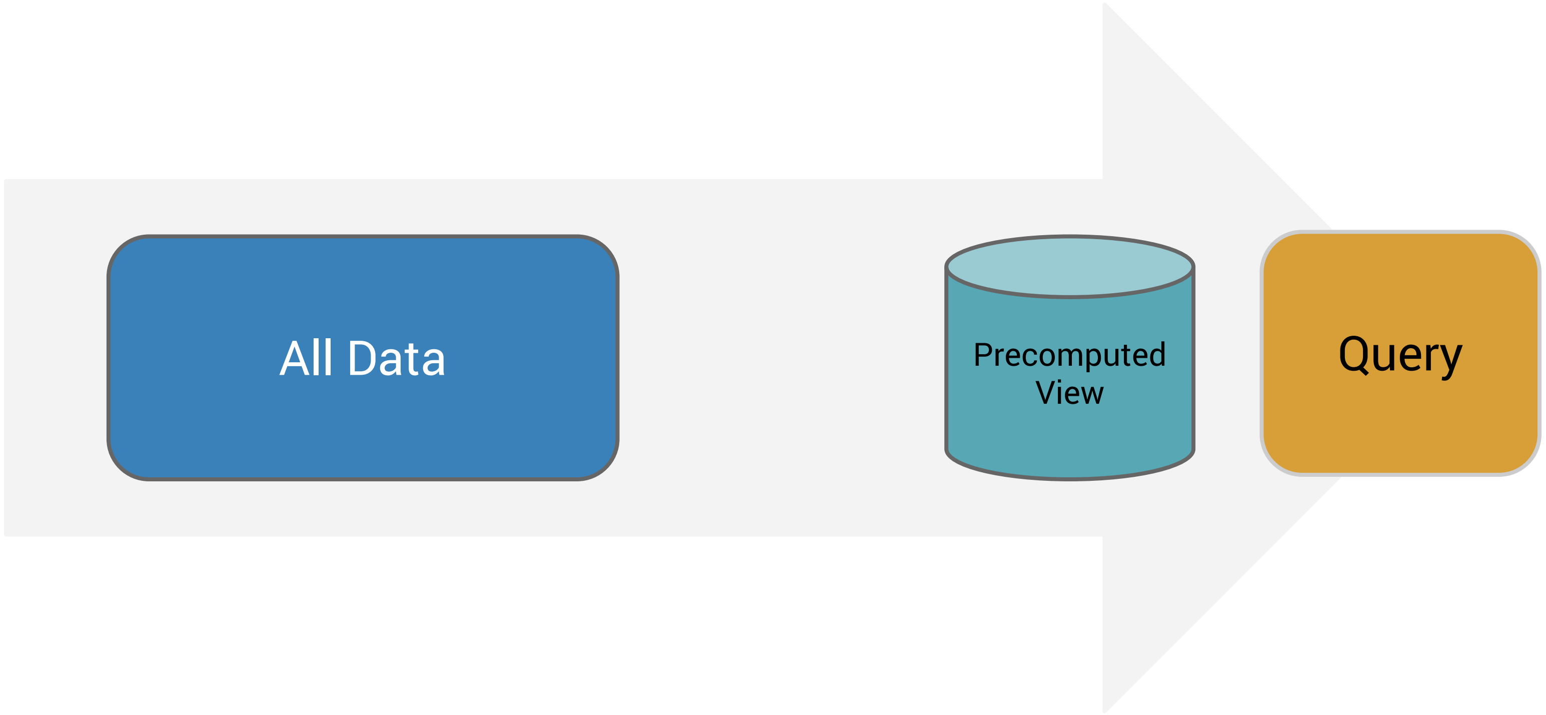
Query: Number of people living in each city

Person	City	Timestamp
Nathan	Antwerp	2008-10-11
John	Cologne	2010-01-23
Dirk	Antwerp	2012-09-12
Nathan	Cologne	2014-04-29

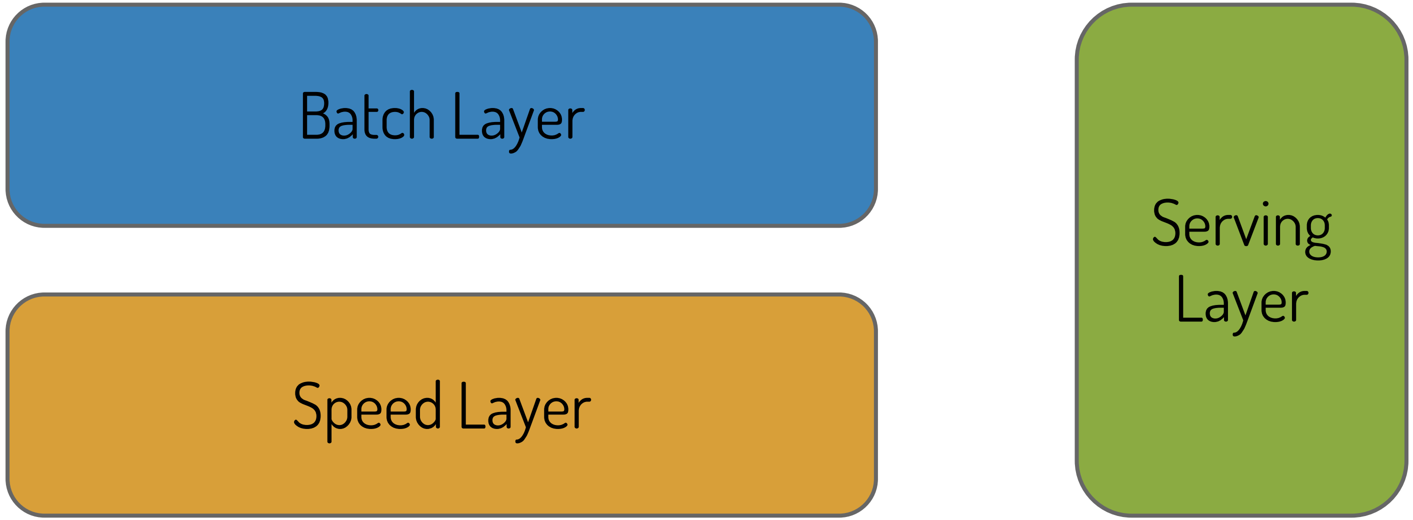
$\rightarrow$

City	Count
Antwerp	1
Cologne	2

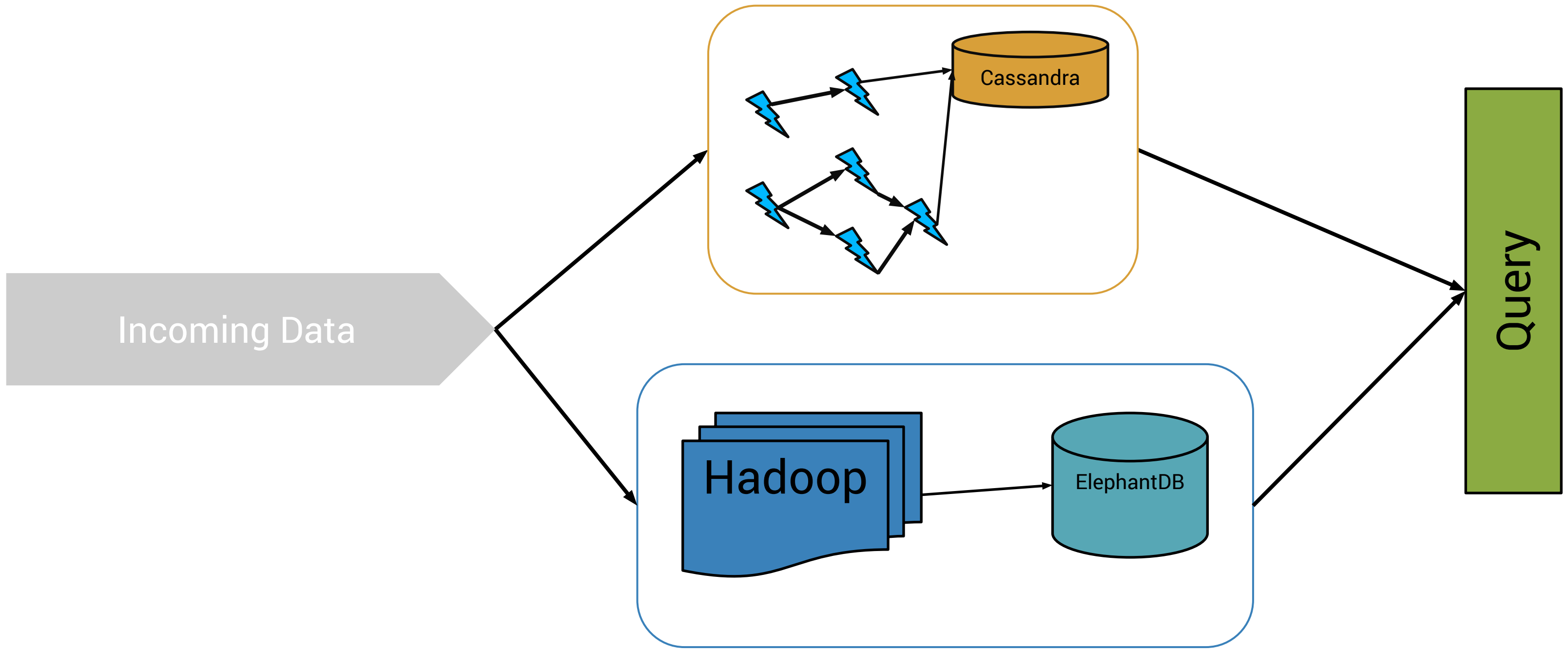
Query



Layered Architecture

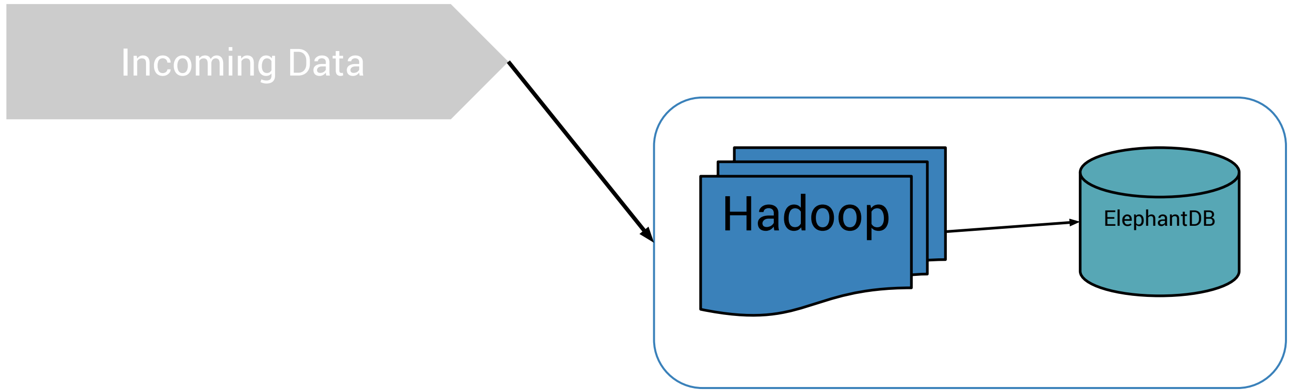


Layered Architecture



Batch Layer

Batch Layer



Batch Layer

Unrestrained computation.

The batch layer can calculate anything, given enough time...

Batch Layer

No need to De-Normalize.

The batch layer stores the data normalized, the generated views are often, if not always denormalized.

Batch Layer

Horizontally scalable.

Batch Layer

High Latency.

Let’s for now pretend the update latency doesn’t matter.

Batch Layer

Functional computation, based on immutable inputs, is idempotent.

Batch Layer

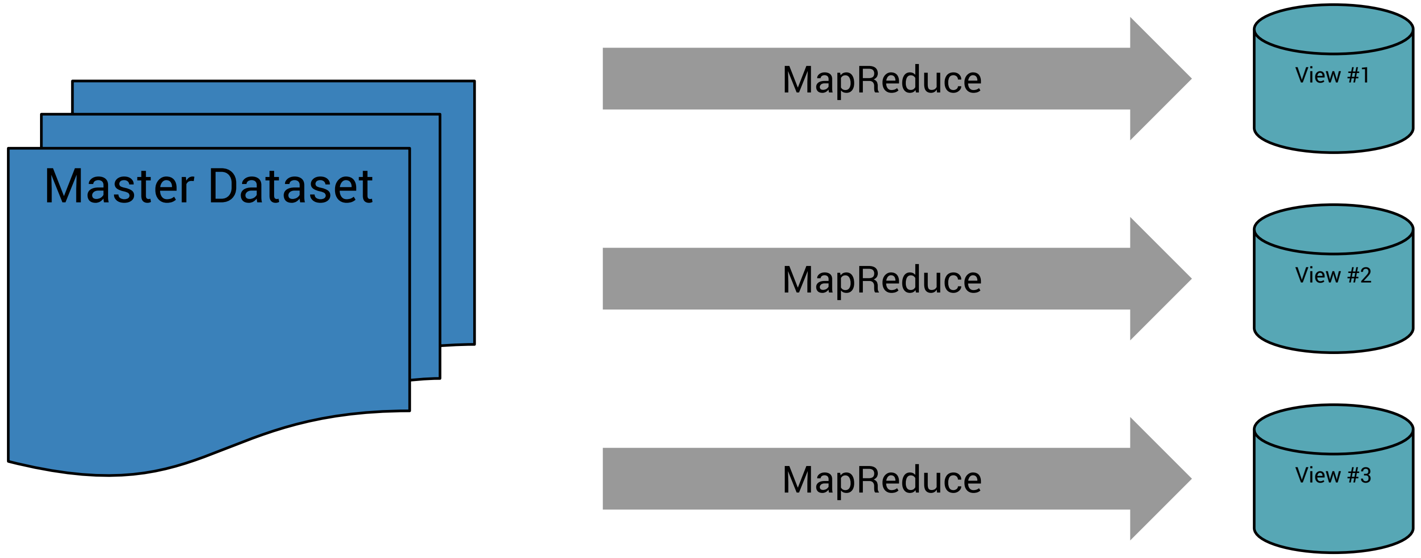
Stores a master copy of the data set

… append only

Batch Layer

hadoop

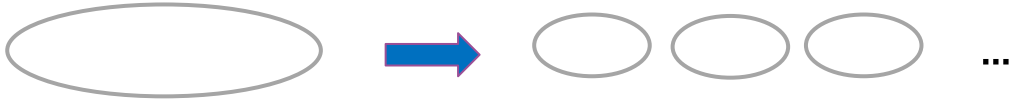
Batch: view generation



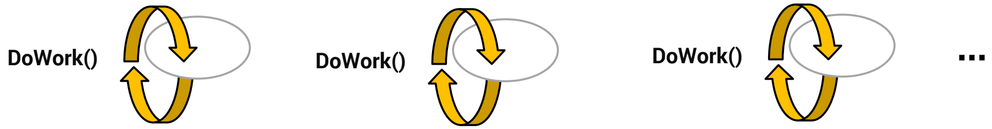
MapReduce

MAP

1. Take a large data set and divide it into subsets

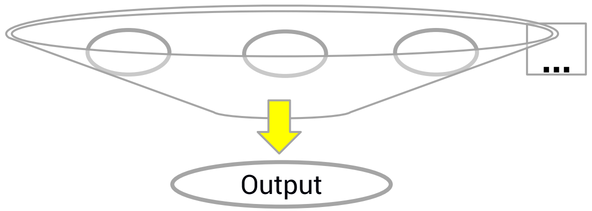


2. Perform the same function on all subsets

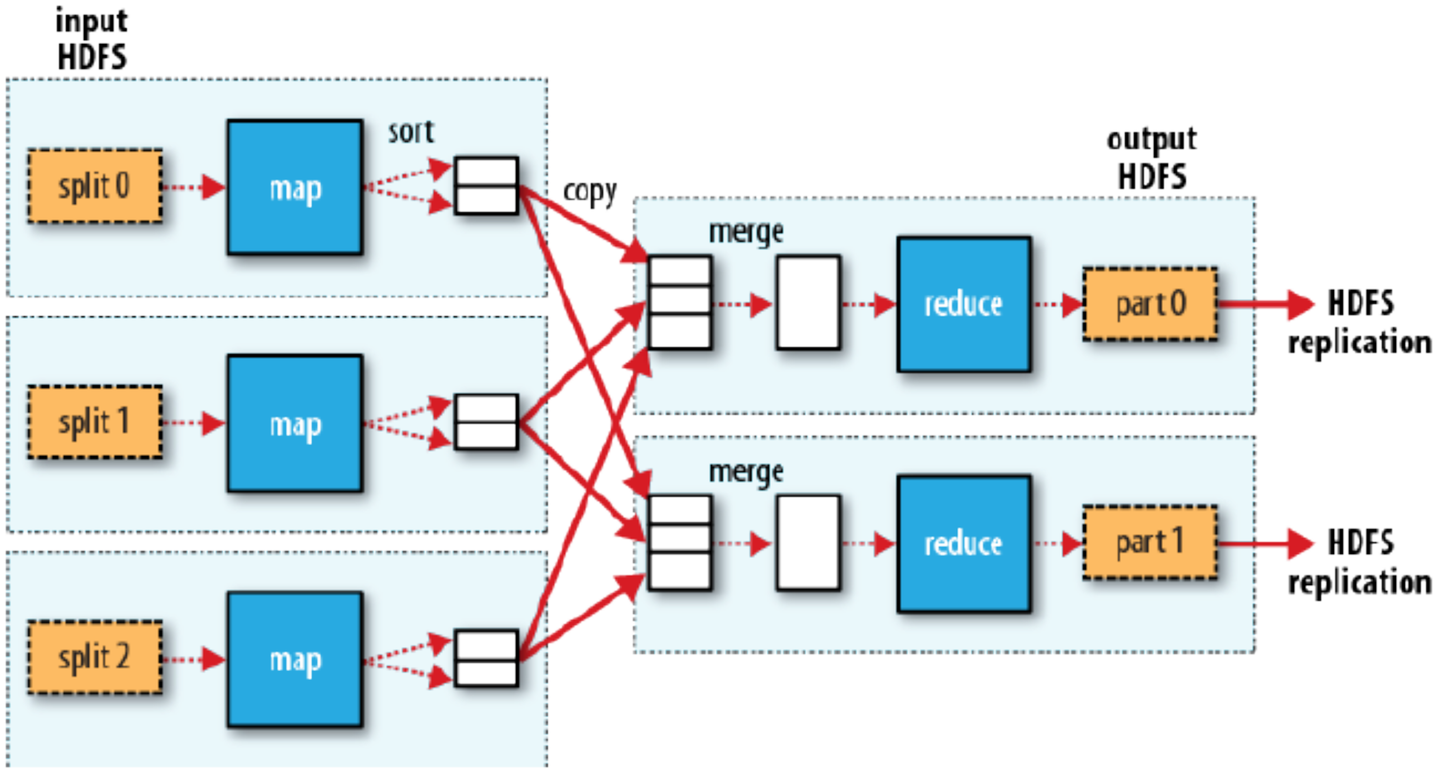


REDUCE

3. Combine the output from all subsets



MapReduce



technicolor
virdata

Serialization & Schema

Catch errors as quickly as they happen.
Validate on write vs on read.

Catch errors as quickly as they happen.
Validate on write vs on read.

Serialization & Schema

CSV is actually a serialization language that is just poorly defined.

Serialization & Schema

Use a format with a schema

- Thrift
- Avro
- Protocolbuffers

Could be combined with Parquet.

Added bonus: it’s faster and uses less space.

technicolor virdata

Batch View Database

Read Only database

No **random** writes required.

Batch View Database

Every iteration produces the views from scratch.

Batch View Databases

Pure Lambda databases

- ElephantDB
- SploutSQL

Databases with a batch load & read only views

- Voldemort

Other databases that could be used

- ElasticSearch/Solr: generate the lucene indexes using MapReduce
- Cassandra: generate sstables
- ...

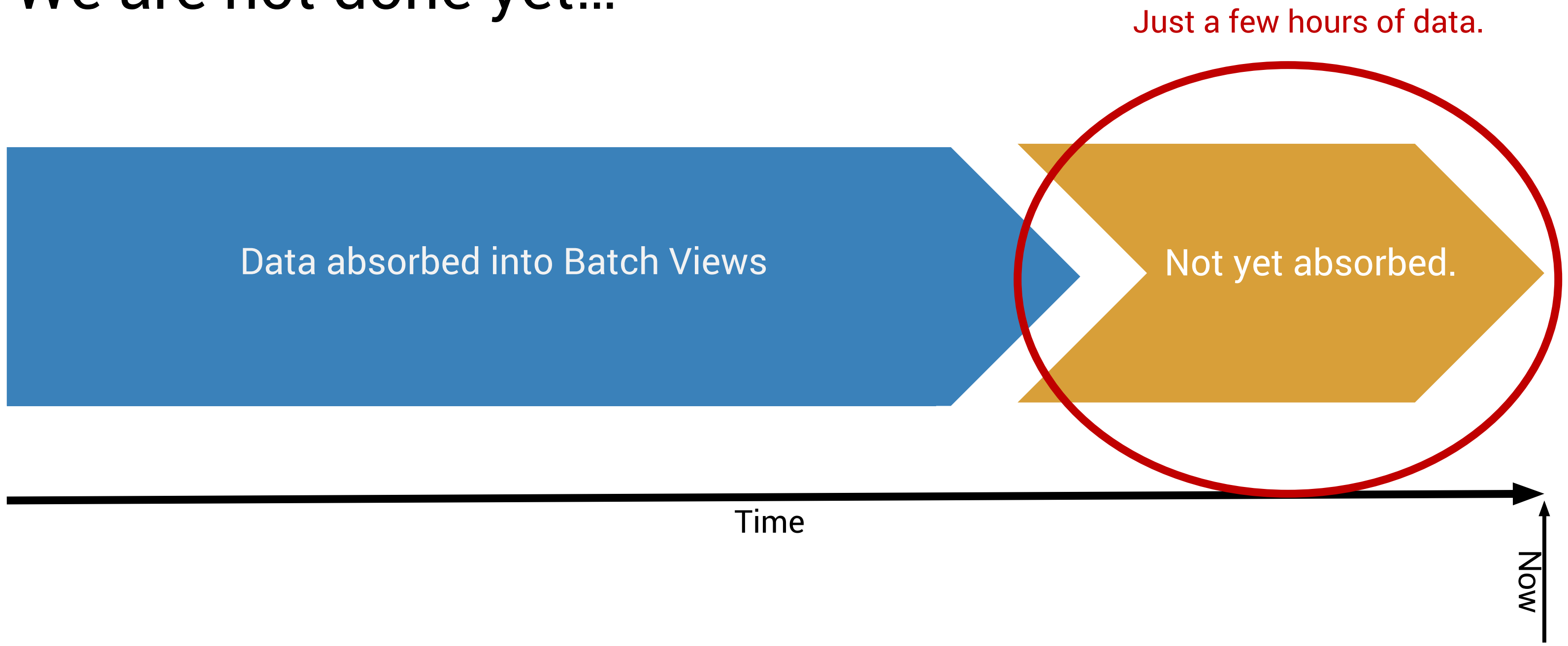
Batch Layer

Eventually consistent

Without the associated complexities.

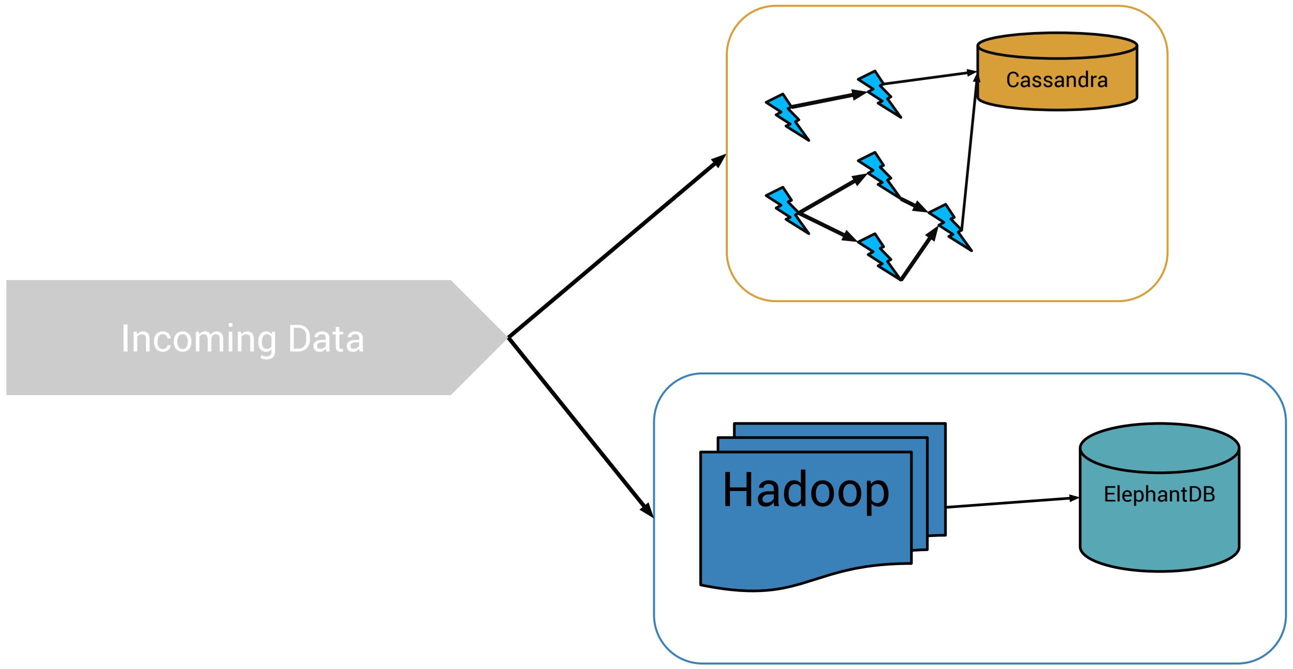
Batch Layer

We are not done yet…



Speed Layer

Speed Layer



Speed Layer

Stream processing.

Speed Layer

Continuous computation.

Speed Layer

Storing a limited window of data.

Compensating for the last few hours of data.

Speed Layer

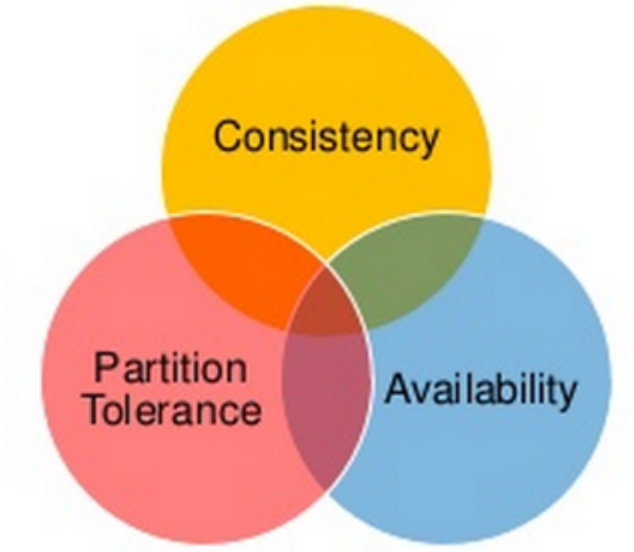
All the complexity is isolated in the Speed Layer.

If anything goes wrong, it’s auto-corrected.

CAP

You have a choice between:

- Availability
  - Queries are eventual consistent
- Consistency
  - Queries are consistent



technicolor virdata

Eventual accuracy

Some algorithms are hard to implement in real-time.
For those cases we could estimate the results.

Speed Layer

Storm

Storm

Message passing

Storm

Distributed processing

Storm

Horizontally scalable.

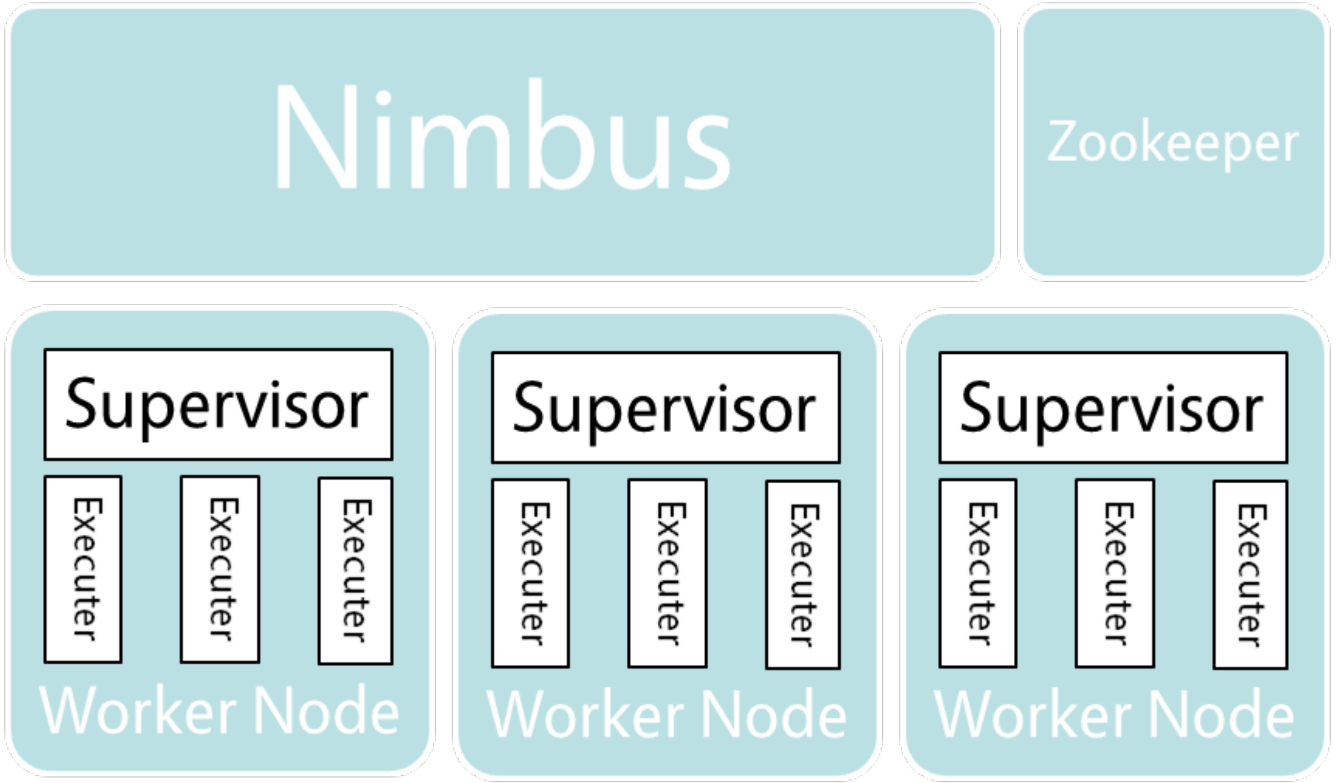
Storm

Incremental algorithms

Storm

Fast.

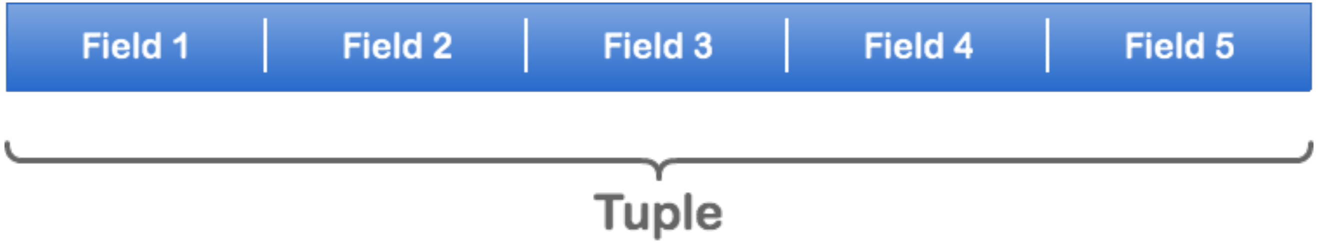
Storm



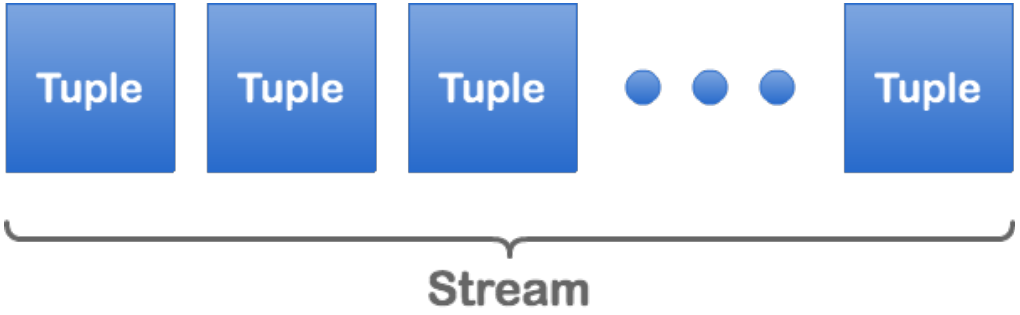
technicolor virdata

Storm

Tuple



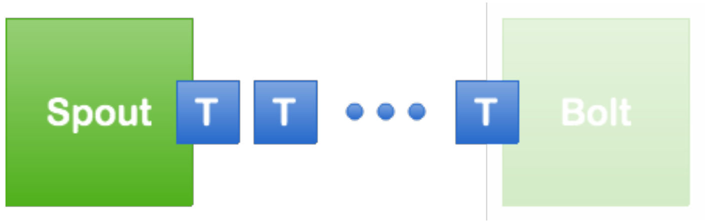
Stream

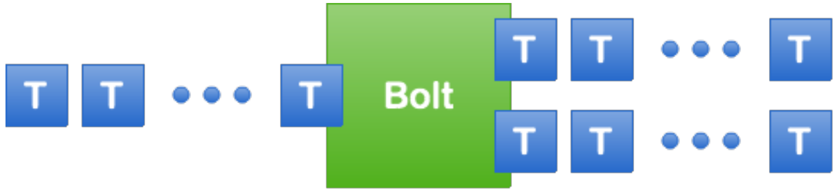


technicolor virdata

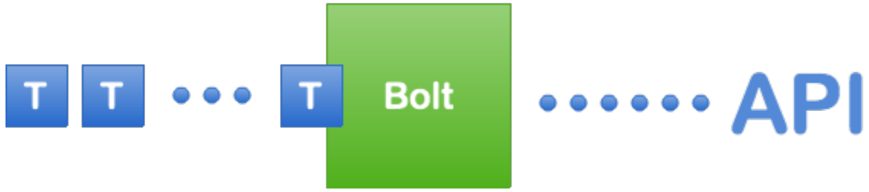
Storm

Spout





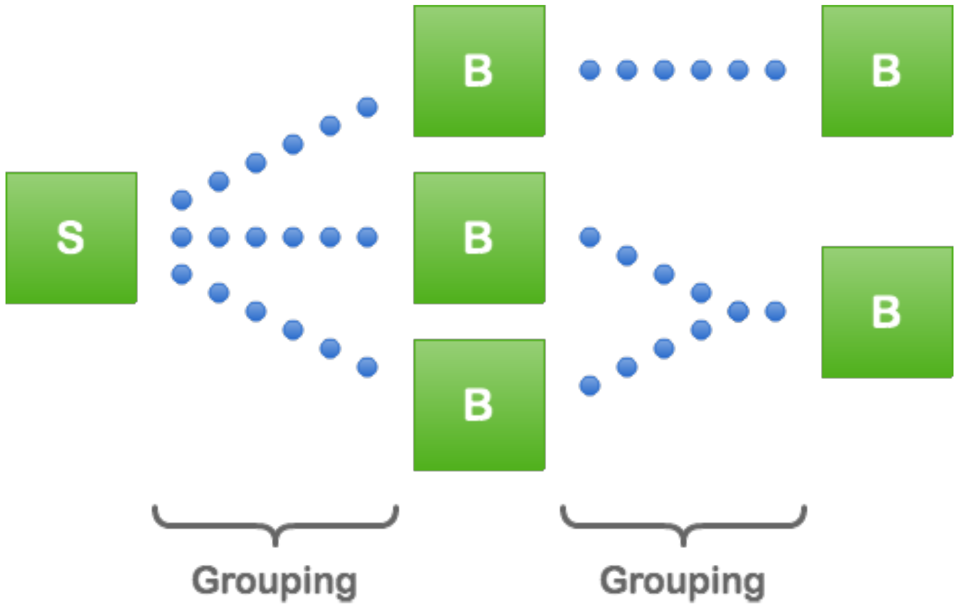
Bolt



technicolor virdata

Storm

Grouping



Data Ingestion

Queues & Pub/Sub models are a natural fit.

Data Ingestion

- Kafka
- Flume
- Scribe
- *MQ
- ...

Speed Layer Views

The views need to be stored in a random writable database.

Speed Layer Views

The logic behind a R/W database is much more complex than a read-only view.

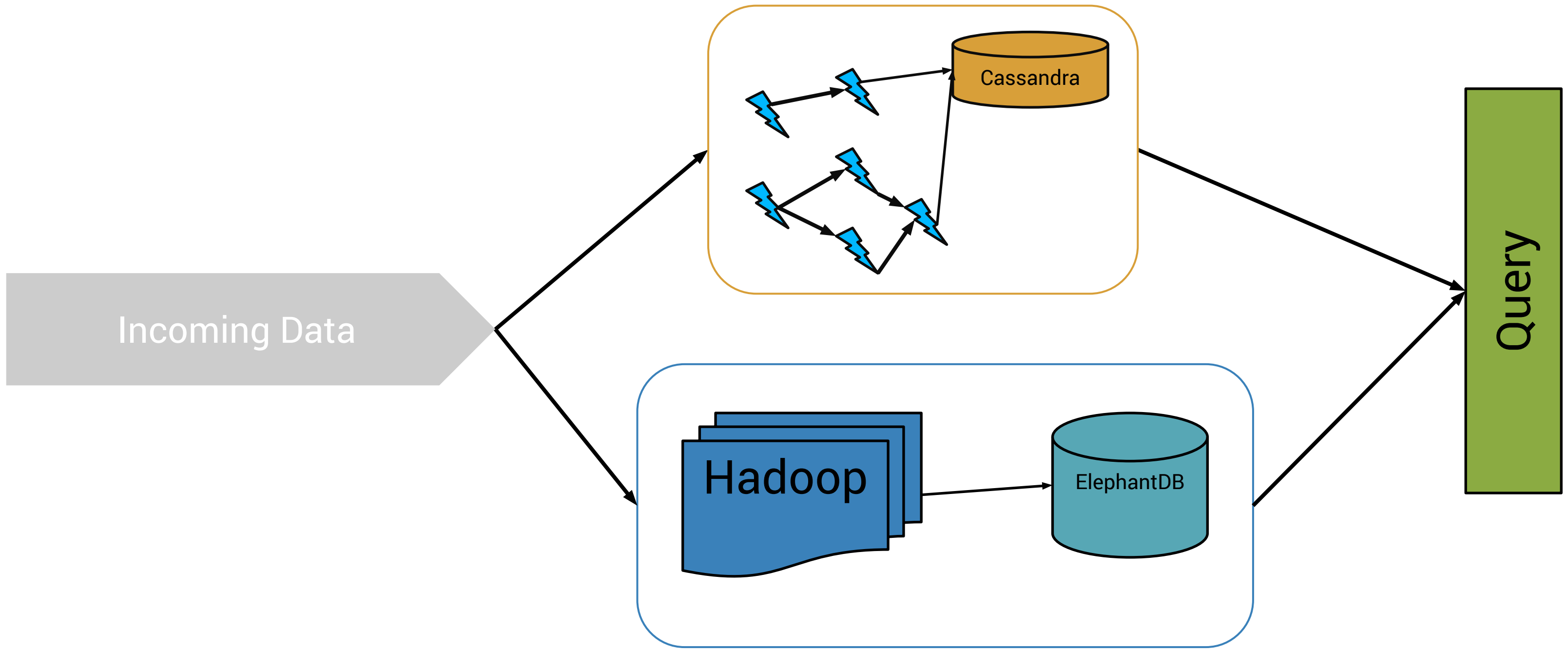
Speed Layer Views

The views are stored in a Read & Write database.

- Cassandra
- Hbase
- Redis
- SQL
- ElasticSearch
- ...

Serving Layer

Serving Layer



Serving Layer

Random reads.

Serving layer

This layer queries the batch & real-time views and merges it.

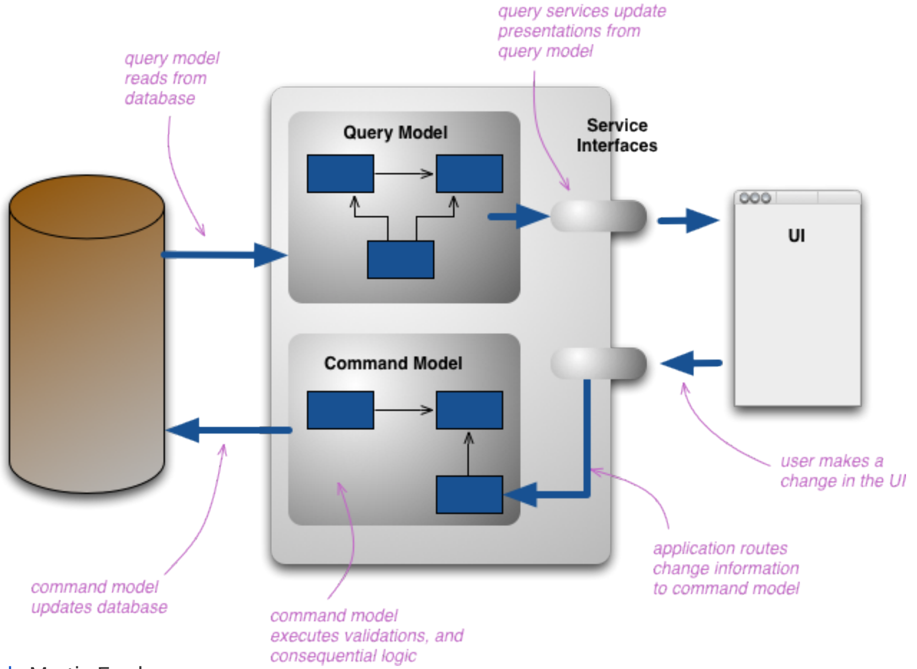
technicolor
virdata

Serving Layer

How to query an Average?

Side note: CQRS

CQRS

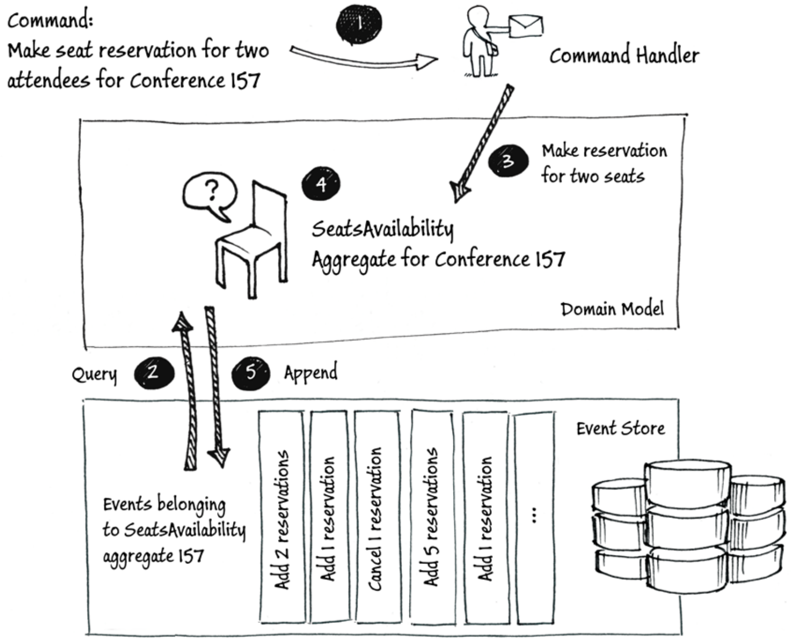

query services update presentations from query model
query model reads from database
Query Model
Service Interfaces
UI
Command Model
user makes a change in the UI
application routes change information to command model
command model updates database
command model executes validations, and consequential logic

Source: martinfowler.com/bliki/CQRS.html - Martin Fowler

CQRS & Event Sourcing

Event Sourcing

- Every command is a new event.
- The event store keeps all events, new events are appended.
- Any query loops through all related events, even to produce an aggregate.



Command:
Make seat reservation for two attendees for Conference 157
1
Command Handler
3
Make reservation for two seats
4
SeatsAvailability Aggregate for Conference 157
Domain Model
Query
2
5
Append
Event Store
Events belonging to SeatsAvailability aggregate 157
Add 2 reservations
Add 1 reservation
Cancel 1 reservation
Add 5 reservations
Add 1 reservation
...

source: CQRS Journey - Microsoft Patterns & Practices

technicolor virdata

Lambda Architecture

Lambda Architecture

The Lambda Architecture can discard any view, batch and real-time, and just recreate everything from the master data.

Lambda Architecture

Mistakes are corrected via recomputation.

Write bad data? Remove the data & recompute.
Bug in view generation? Just recompute the view.

Lambda Architecture

Data storage is highly optimized.

Lambda Architecture

Immutability changes everything.

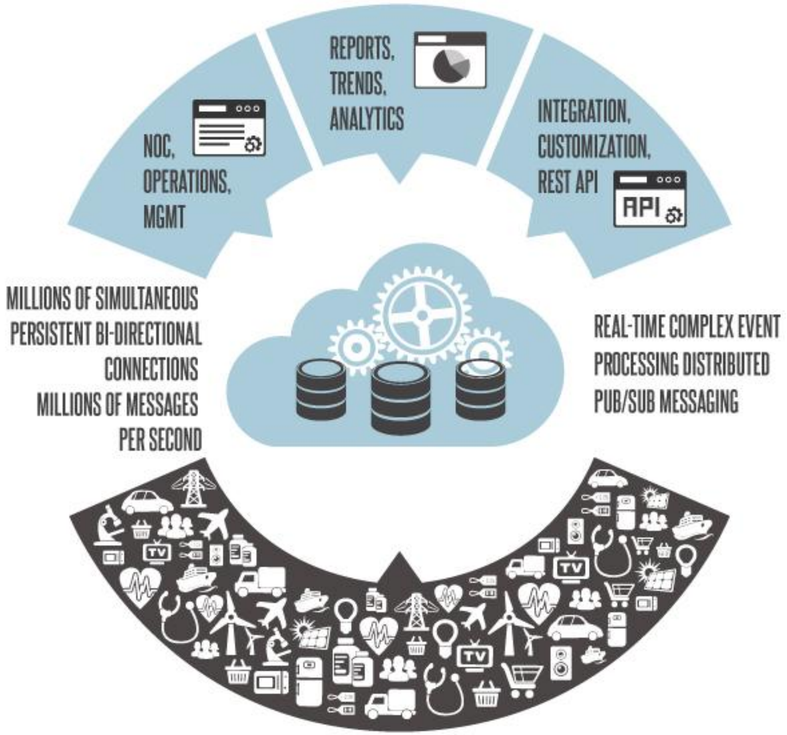
Questions?

@nathan_gs #nosql14
nathan@nathan.gs / slideshare.net/nathan_gs
lambda-architecture.net / @LambdaArch / #LambdaArch

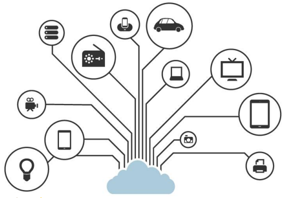
virdata

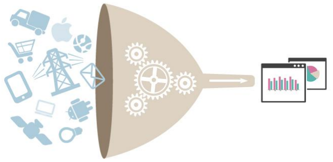
Virdata is the cross-industry cloud service/platform for the Internet of Things. Designed to elastically scale to monitor and manage an unprecedented amount of devices and applications using concurrent persistent connections, Virdata opens the door to numerous new business opportunities.

Virdata combines Publish-Subscribe based Distributed Messaging, Complex Event Processing and state-of-the-art Big Data paradigms to enable both historical & real-time monitoring and near real-time analytics with a scale required for the Internet of Things.



NOC, OPERATIONS, MGMT
REPORTS, TRENDS, ANALYTICS
INTEGRATION, CUSTOMIZATION, REST API
API
MILLIONS OF SIMULTANEOUS PERSISTENT BI-DIRECTIONAL CONNECTIONS
MILLIONS OF MESSAGES PER SECOND
REAL-TIME COMPLEX EVENT PROCESSING DISTRIBUTED PUB/SUB MESSAGING
TV





technicolor
virdata

Acknowledgements

I would like to thank Nathan Marz for writing a very insightful book, where most of the ideas in this presentation come from.

Parts of this presentation has been created while working for datacrunchers.eu, I thank them for the opportunities to speak about the Lambda Architecture both at clients and at conferences. DataCrunchers is the first Big Data agency in Belgium.

Schema's & Pictures:

Computing Trends: Immutability Changes Everything - Pat Helland, RICON2012

MapReduce #1: PolybasePass2012.pptx - David J. DeWitt, Microsoft Gray Systems Lab

MapReduce #2: Introduction to MapReduce and Hadoop - Shivnath Babu, Duke

CQRS: martinfowler.com/bliki/CQRS.html - Martin Fowler

CQRS & Event Sourcing: CQRS Journey - Adam Dymitruk, Josh Elster & Mark Seemann, Microsoft Patterns & Practices

Thank you

@nathan_gs
nathan@nathan.gs

technicolor
virdata