Chapter3

Understanding Big Data Technology Foundations

Lamda Architecture

Basanta Joshi, PhD

Asst. Prof., Depart of Electronics and Computer Engineering

Program Coordinator, MSc in Information and Communication Engineering

Member, Laboratory for ICT Research and Development (LICT)

Member, Research Management Cell (RMC)

Institute of Engineering

basanta@ioe.edu.np

http://www.basantajoshi.com.np

https://scholar.google.com/citations?user=iocLiGcAAAAJ

https://www.researchgate.net/profile/Basanta_Joshi2

The world is changing !

The model of Generating/Consuming Data has changed !.

Old Model: few companies are generating data, all others are consuming data

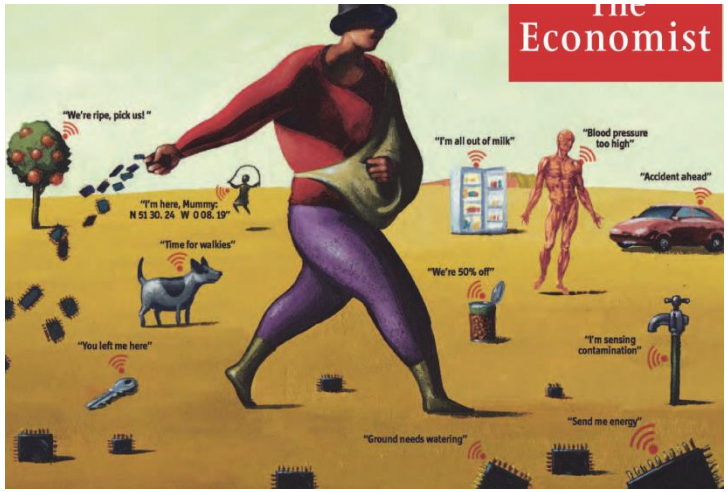
New Model: all of us are generating data, and all of us are consuming data


IN 60 SECONDS..
1 NEW DEFINITION IS ADDED ON urban
1,600+ READS ON Scribd.
13,000+ HOURS MUSIC STREAMING ON PANDORA
12,000+ NEW ADS POSTED ON craigslist
370,000+ MINUTES VOICE CALLS ON skype
98,000+ TWEETS
20,000+ NEW POSTS ON tumblr.
THE LARGEST SOCIAL READING PUBLISHING COMPANY
New Craigslist Ads
320+ NEW twitter ACCOUNTS
13,000+ iPhone APPLICATIONS DOWNLOADED
100+ NEW Linked in ACCOUNTS
1 associatedcontent NEW ARTICLE IS PUBLISHED
THE WORLD'S LARGEST COMMUNITY CREATED CONTENT!!
QUESTIONS ASKED ON THE INTERNET...
100+ Answers.com
40+ YAHOO! ANSWERS
600+ NEW VIDEOS
6,600+ NEW PICTURES ARE UPLOADED ON flickr
50+ WordPress DOWNLOADS
25+ HOURS TOTAL DURATION
70+ DOMAINS REGISTERED
60+ NEW BLOGS
168 MILLION EMAILS ARE SENT
694,445 SEARCH QUERIES
1,700+ Firefox DOWNLOADS
695,000+ facebook STATUS UPDATES
125+ PLUGIN DOWNLOADS
1,500+ BLOG POSTS
79,364 WALL POSTS
510,040 COMMENTS
Google
Google Search
GO-Globe.com

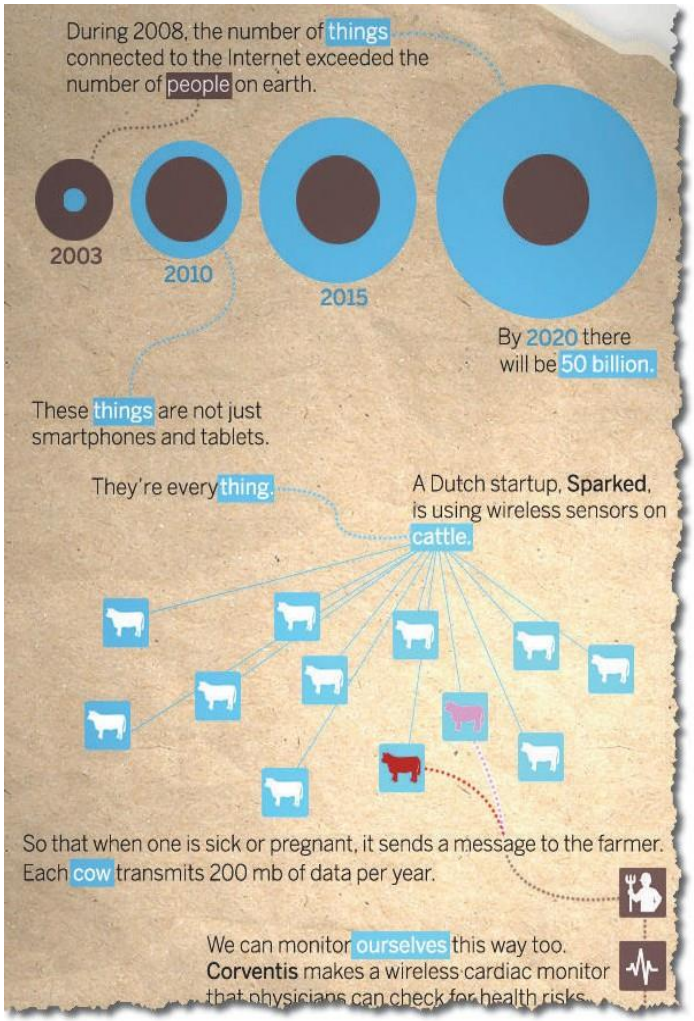
Internet Of Things – Sensors are/will be everywhere

There are more devices tapping into the internet than people on earth

How do we prepare our systems/architecture for the future?



Source: The Economist



Source: Cisco

trivadis
makes IT easier.

2020 IoT Insights

- With 1.3 billion **projected subscriptions by 2023**, IoT is about to experience another boost by the 5G technology.
- By 2022, **Google Home will have the largest IoT devices market share**, at 48%.
- The average number of connected devices **per household in 2020 will be 50**.
- By 2021, **35 billion IoT devices** will be installed around the world.
- The number of connected devices **in 2020 is predicted to hit 50 billion**.

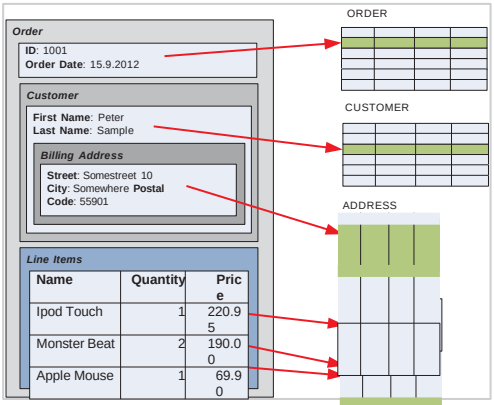
- https://techjury.net/blog/how-many-iot-devices-are-there/#gref

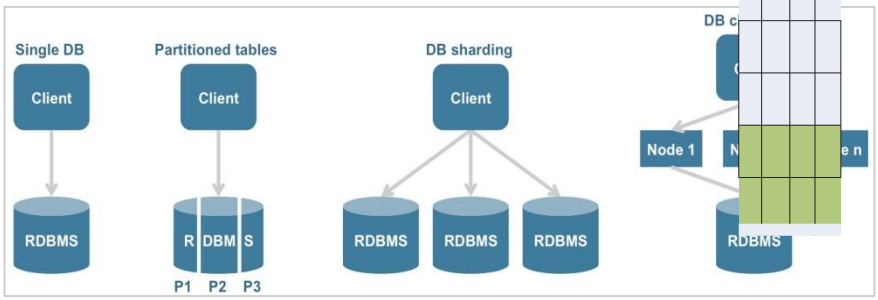
The world is changin

new data stores

Problem of traditional (R)DBMS approach:

- Complex object graph
- Schema evolution
- Semi-structured data
- Scaling





Polyglot persistence

- Using multiple data storage technologies (RDMBS + NoSQL + NewSQL + In- Memory)

- The world is changing ! New platforms evolving (i.e. Hadoop Ecosystem)

Data as an Asset – Store everything?

Data is just too valuable to delete!
We must store anything!

It depends …
Big Data technologies allow **to store the raw information** from new and existing data sources so that you can **later use it** to create **new data-driven products**, which you haven't **thought about today**!

Nonsense!
Just store the data **you know you need today**!

AGENDA

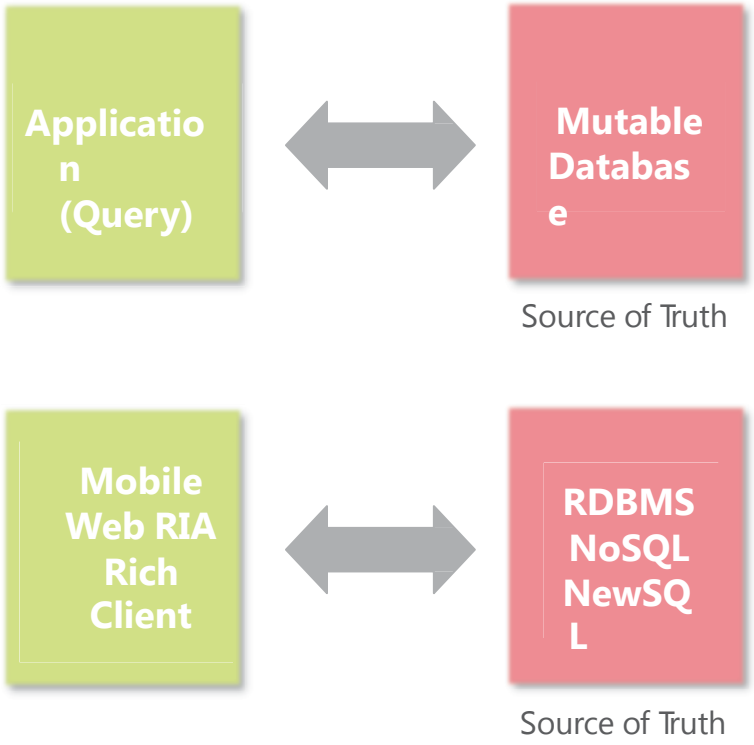
1. Big Data and Fast Data, what is it?
2. **Architecting (Big) Data Systems**
3. The Lambda Architecture
4. Use Case and the Implementation
5. Summary and Outlook

What is a data system?

- A (data) system that manages the **storage and querying of data** with a **lifetime measured in years** encompassing **every version** of the application to ever exist, every **hardware failure** and every **human mistake** ever made.

- A data system answers **questions based on information** that was **acquired in the past**

How do we build (data) systems today – Today's Architectures



Source of Truth **is mutable**!

- CRUD pattern

What is the problem with this?

- Lack of Human Fault Tolerance
- Potential loss of information/ data

Lack of Human Fault Tolerance

Bugs will be deployed to production over the lifetime of a data system

Operational mistakes will be made

Humans are part of the overall system
- Just like hard disks, CPUs, memory, software
- design for human error like you design for any other fault

Examples of human error
- Deploy a bug that increments counters by two instead of by one
- Accidentally delete data from database
- Accidental DOS on important internal service

Worst two consequences: **data loss** or **data corruption**

As long as an error **doesn‘t lose or corrupt** good data, you can **fix** what went wrong

Lack of Human Fault Tolerance – Immutability vs. Mutability

An immutable system captures historical records of events

Each **event** happens at a **particular time** and is **always true**

The **U** and **D** in CRUD

A mutable system updates the current state of the world

Mutable systems inherently **lack** human fault-tolerance

Name	City	Timestamp
Guido	Berne	1.8.1999
Albert	Zurich	10.5.1988

Name	City	Timestamp
Guido	Berne	1.8.1999
Albert	Zurich	10.5.1988
Guido	**Basel**	**1.4.2013**

Name	City
Guido	Berne
Albert	Zurich

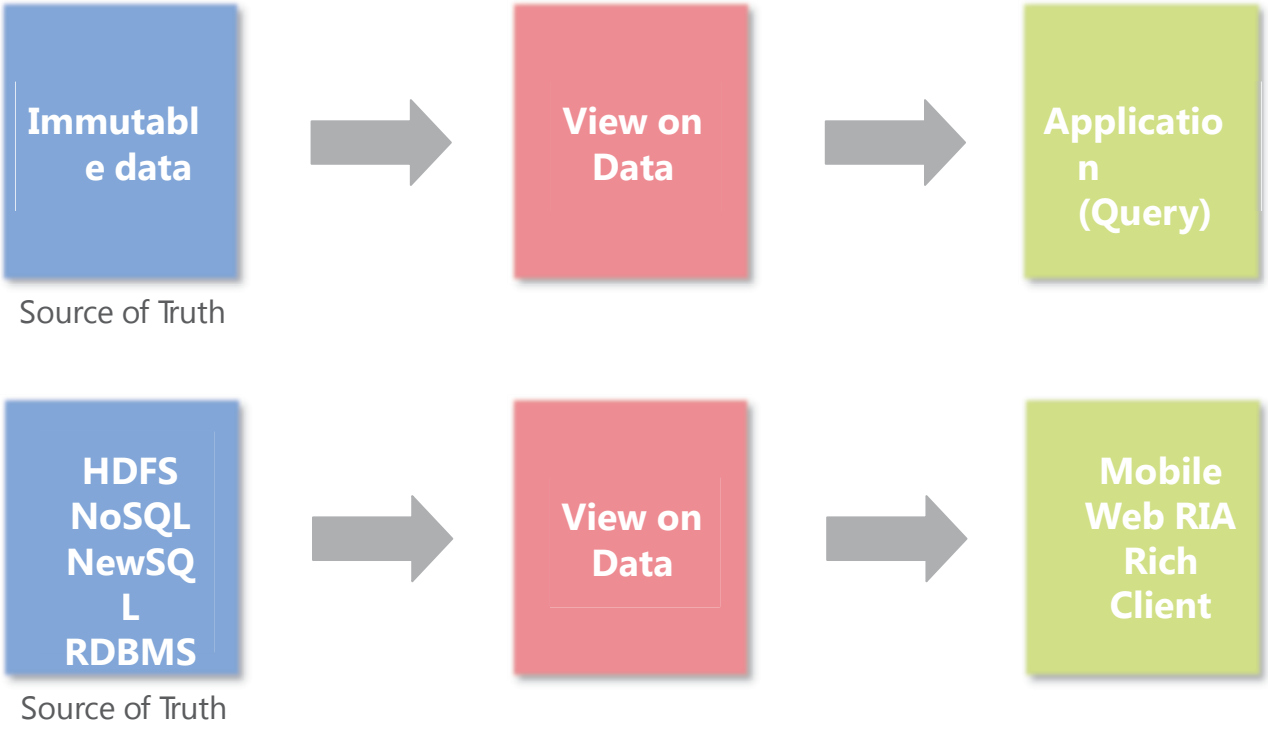
Name	City
Guido	**Basel**
Albert	Zurich

Immutability restricts the range of errors causing data loss/data corruption Vastly more **human fault-tolerant**

Conclusion: Your **source of truth** should always be **immutable**

A different kind of architecture with immutable source of truth

Instead of using our traditional approach ! why not building data systems like this

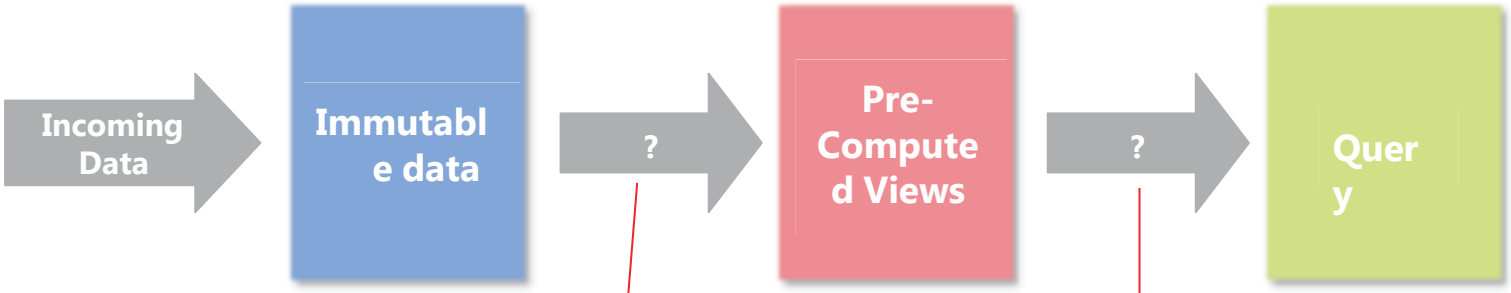


How to create the views on the Immutable data?

On the fly

Materialized, i.e. Pre-computed ?

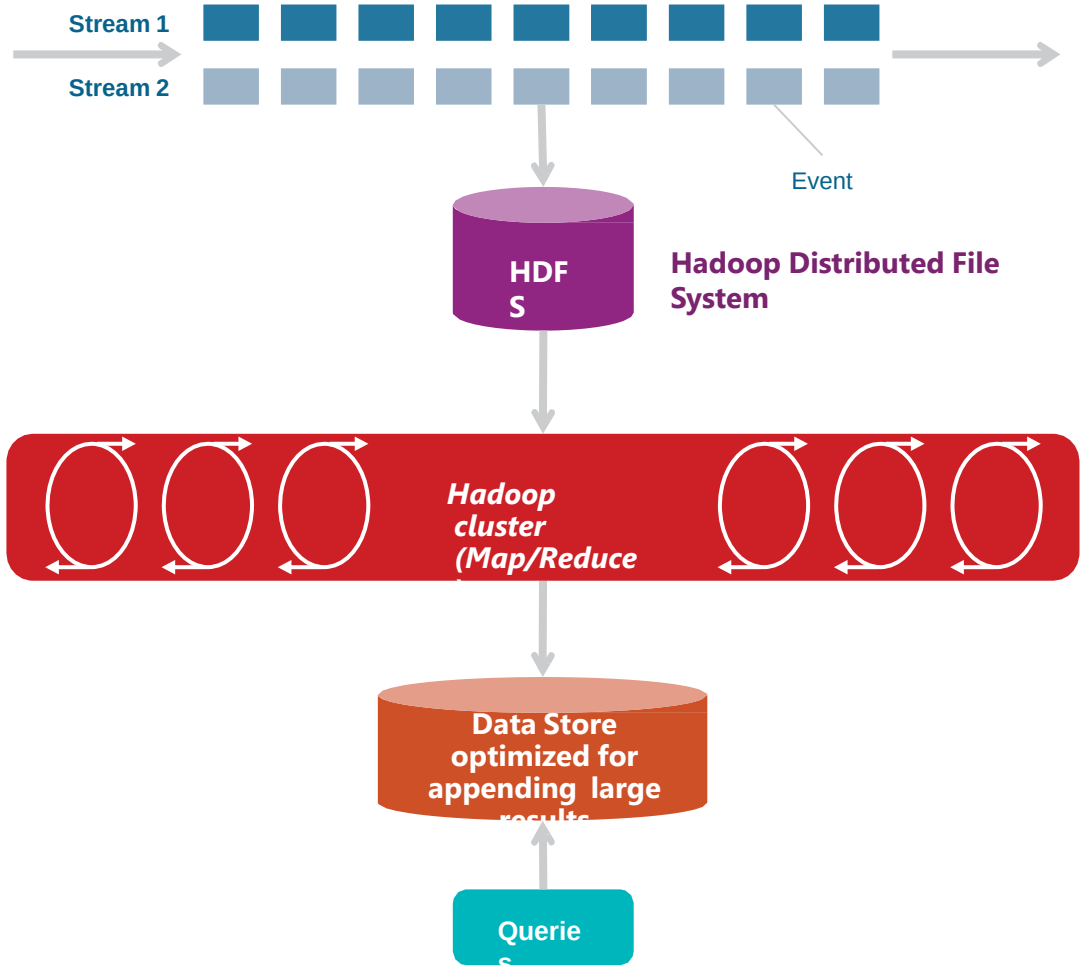
(Big) Data Processing



How to compute the materialized views ?

How to compute queries from the views ?

Today Big Data Processing means Batch Processing !



Big Data Processing - Batch

Favorite Product List Changes

1.2.13	Add	iPAD 64GB
10.3.13	Add	Sony RX-100
11..3.13	Add	Canon GX-10
11.3.13	Remove	Sony RX-100
12.3.13	Add	Nikon S-100
14.4.13	Add	BoseQC-15
15.4.13	Add	MacBook Pro 15
20.4.13	Remove	Canon GX10

Raw information => data

Current Favorite Product List

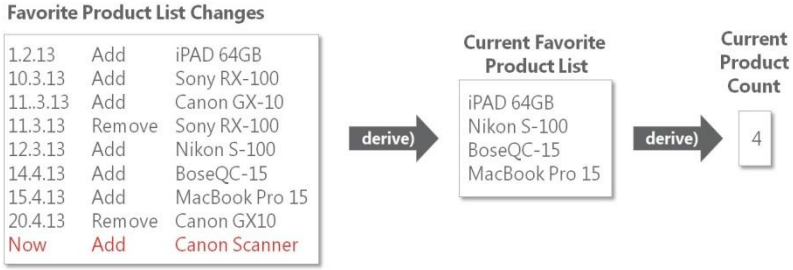
iPAD 64GB
Nikon S-100
BoseQC-15
MacBook Pro 15

Current Product Count

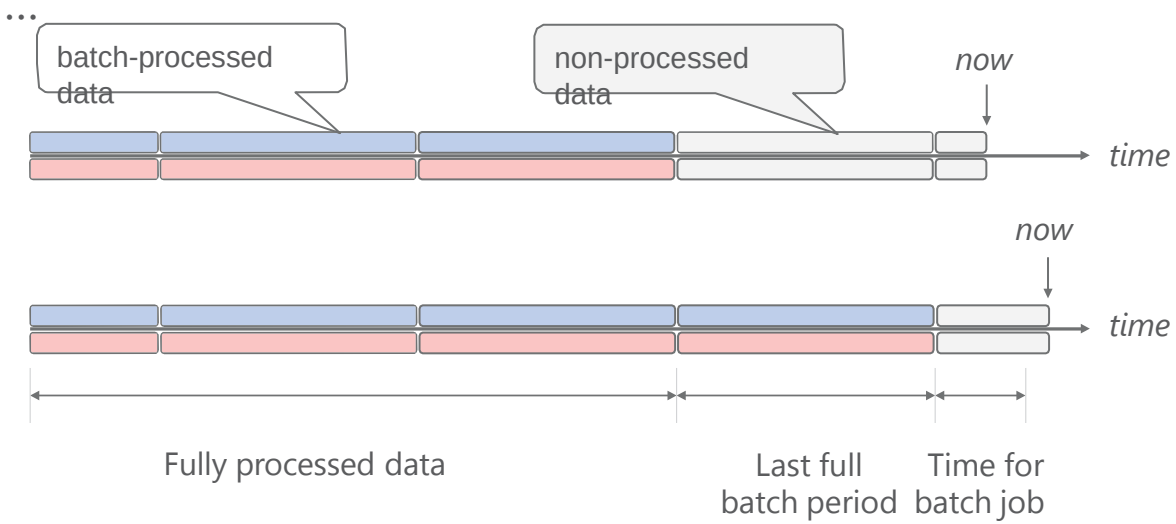
4

Information => derived

Big Data Processing – Batch

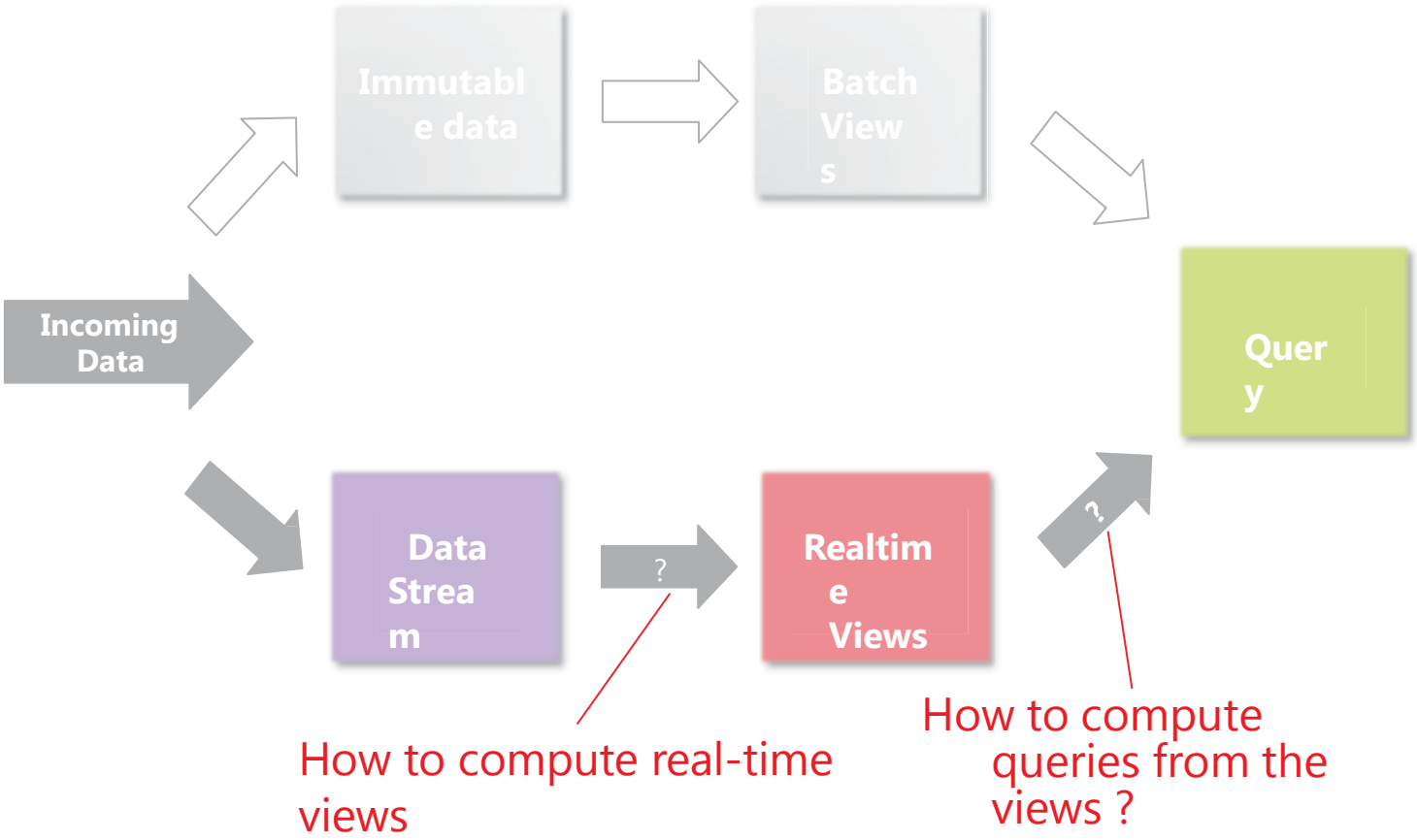


But we are not done yet

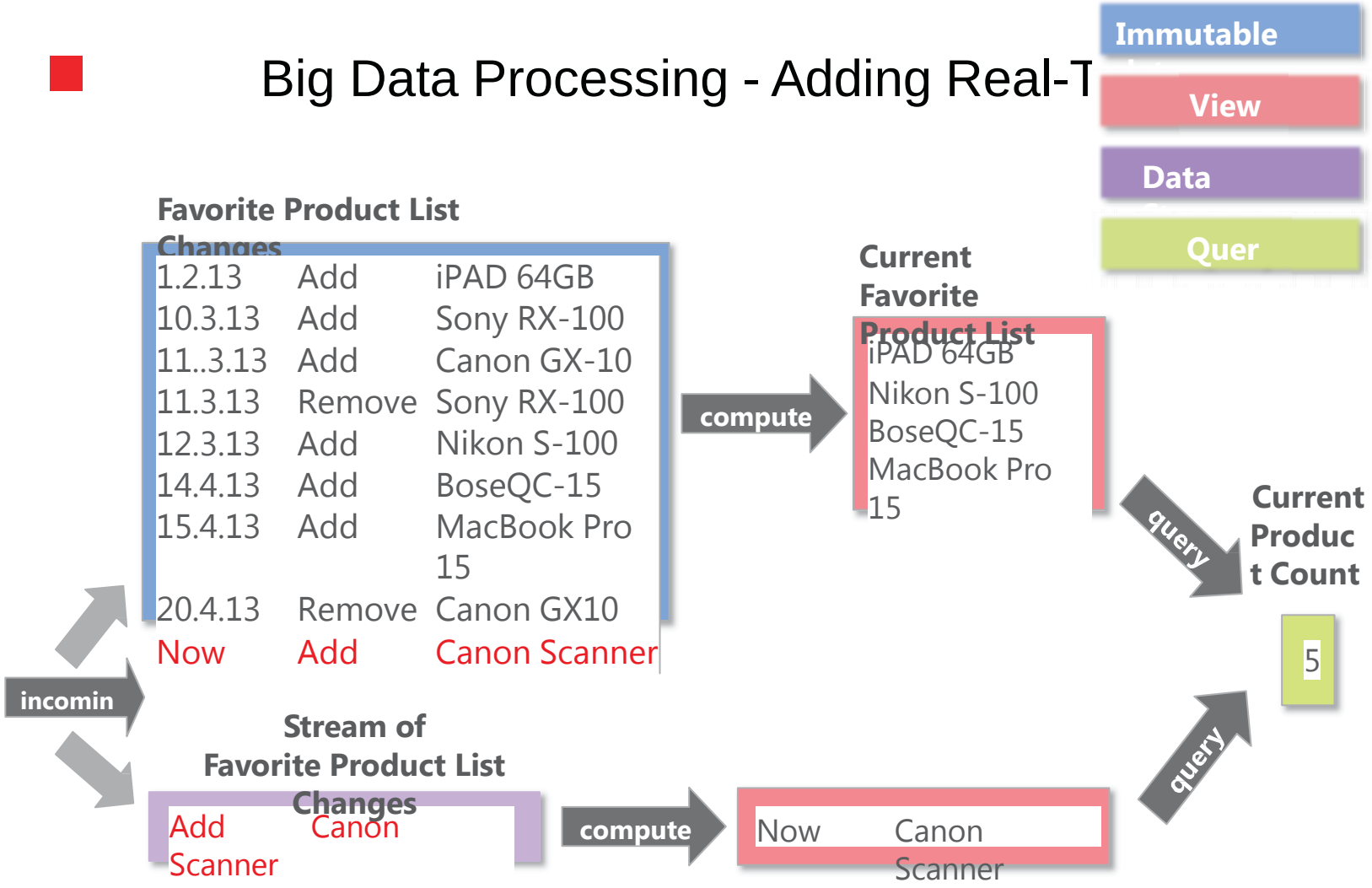


- Using only batch processing, leaves you always with a portion of non- processed data.

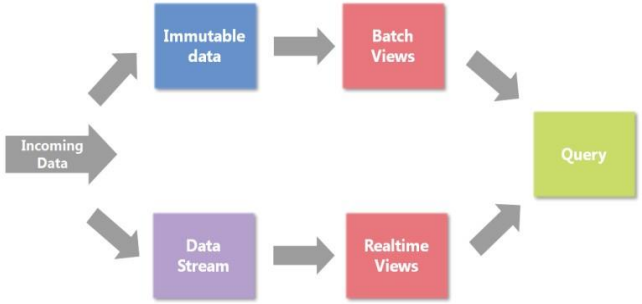
Big Data Processing - Adding Real-Time

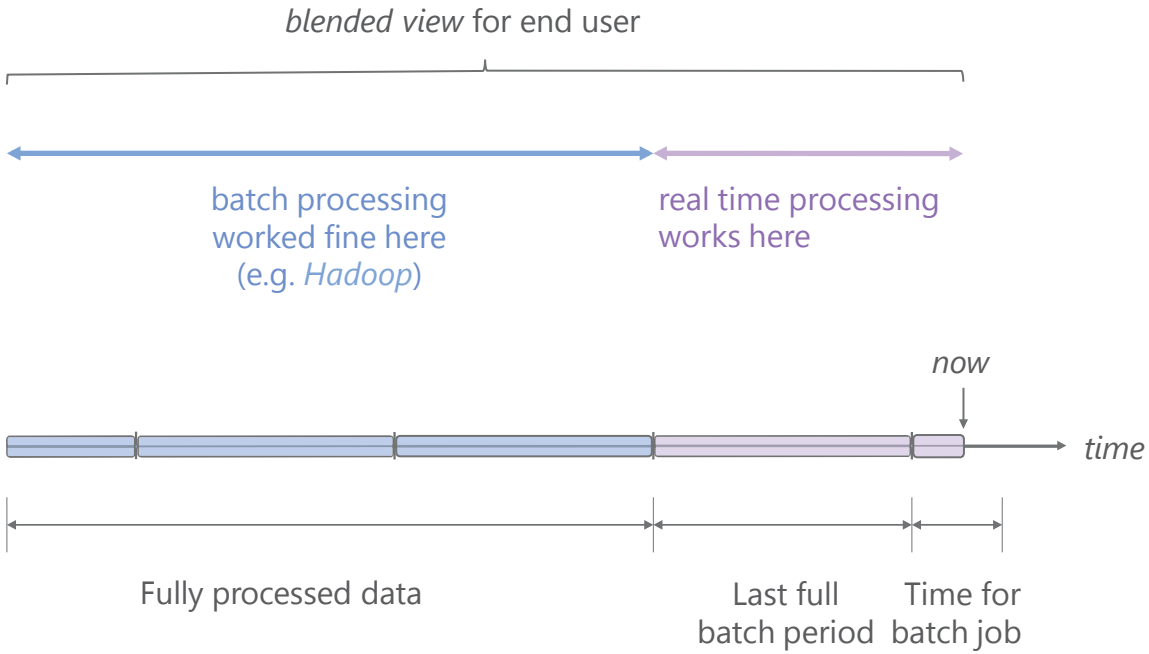


Big Data Processing - Adding Real-T



Big Data Processing - Batch & Real Time



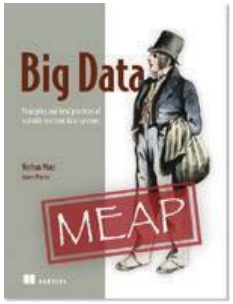


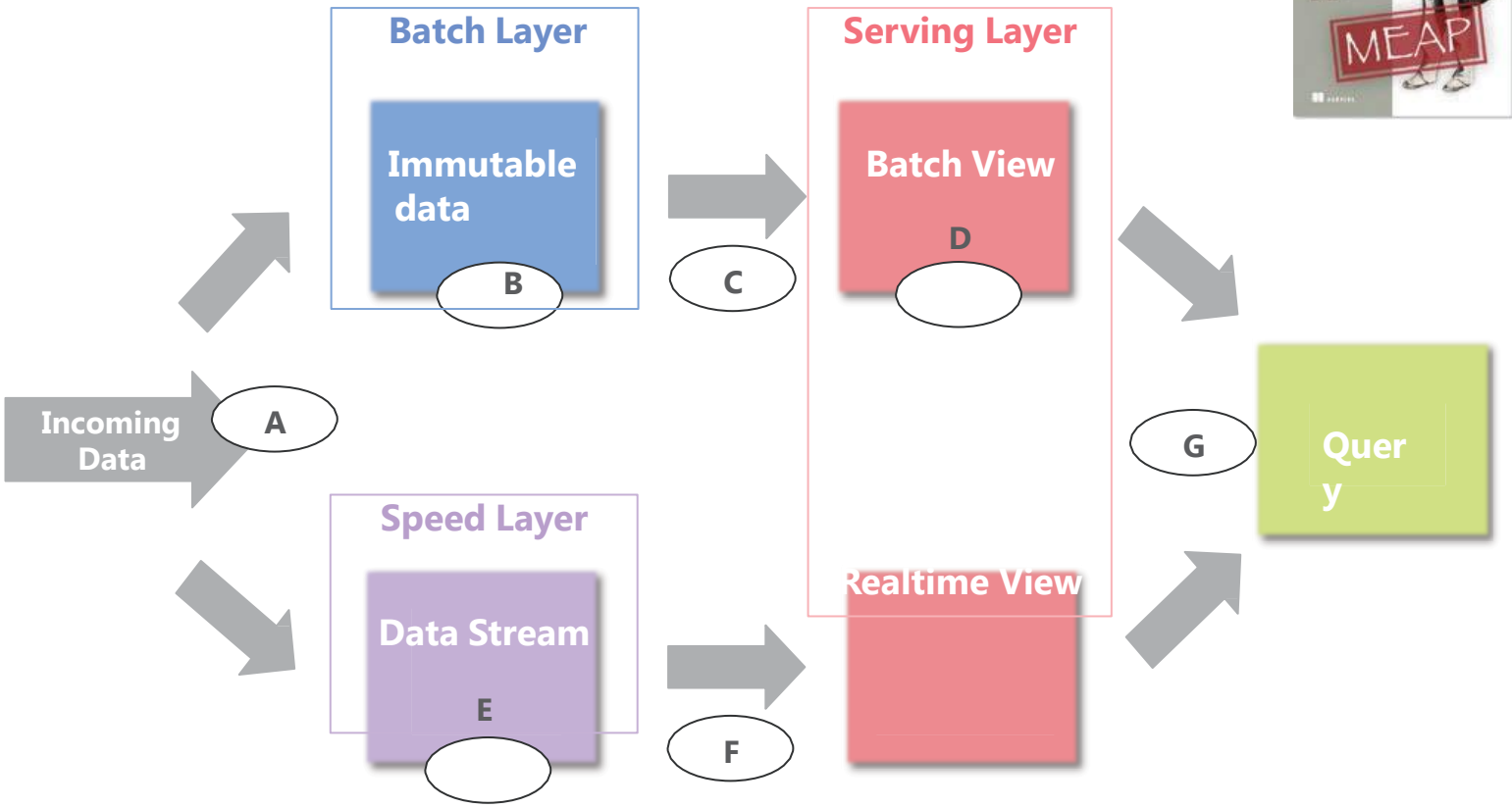
Adapted from Ted Dunning (March 2012):
http://www.youtube.com/watch?v=7PcmbI5aC20

AGENDA

1. Big Data and Fast Data, what is it?
2. Architecting (Big) Data Systems
3. **The Lambda Architecture**
4. The Use Case and the Implementation
5. Summary and Outlook

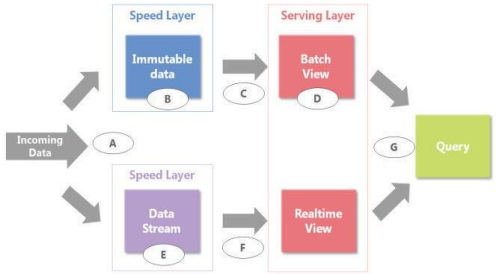
Lambda Architecture





Lambda => Query = function(all data)

Lambda Architectur



A. All data is sent to **both** the **batch and speed layer**

B. Master data set is an **immutable, append-only** set of data

C. Batch layer **pre-computes** query functions from scratch, result is called Batch Views. Batch layer **constantly re-computes** the batch views.

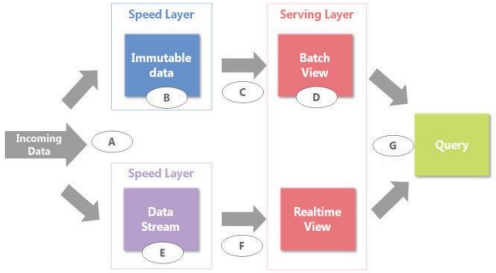
D. Batch views are **indexed** and **stored** in a **scalable database** to get particular values very quickly. Swaps in new batch views when they are available

E. Speed layer **compensates** for the high latency of updates to the Batch Views

F. Uses fast **incremental algorithms** and read/write databases to produce real- time views

G. Queries are resolved by getting results from **both** batch and real-time views

Lambda Architecture



Batch

Stores the immutable constantly growing dataset

Computes arbitrary views from this dataset using BigData technologies (can take hours)

Can be always recreated

Speed

Computes the views from the constant stream of data it receives Needed to compensate for the high latency of the batch layer Incremental model and views are transient

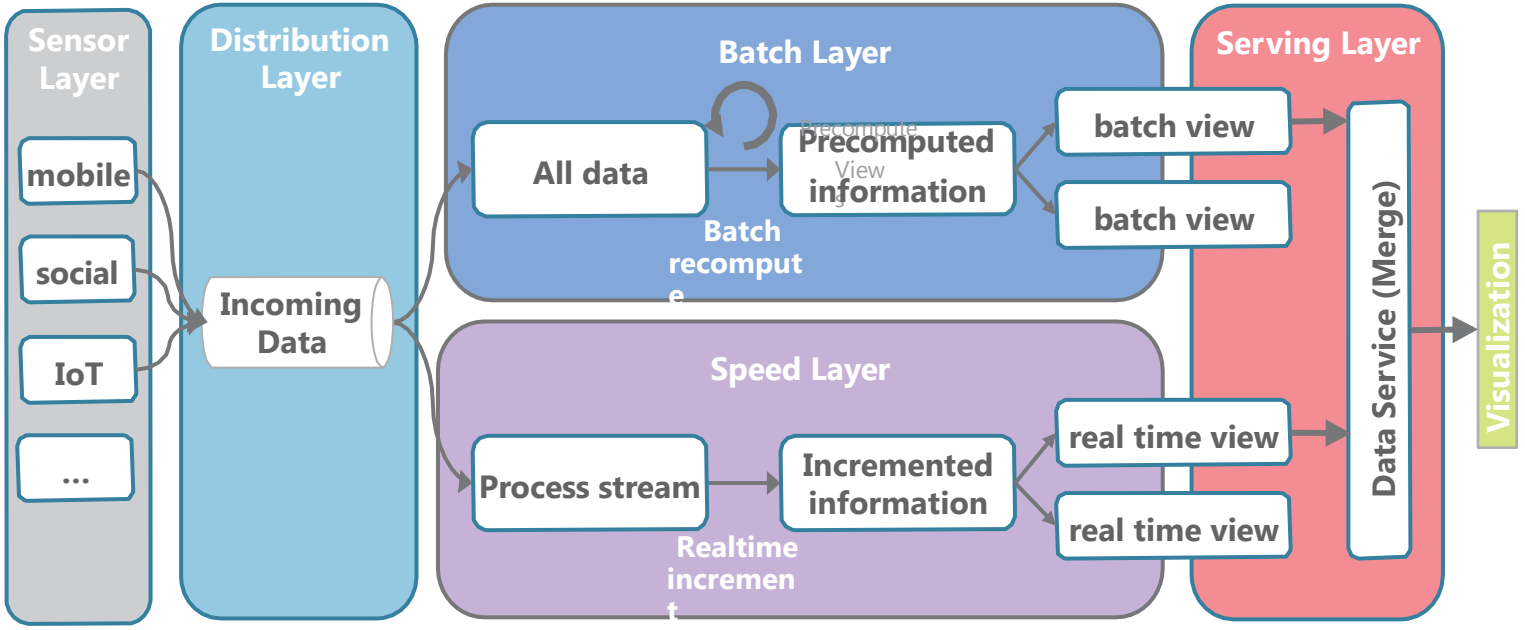
Serving

Responsible for indexing and exposing the pre-computed batch views so that they can be queried

Exposes the incremented real-time views

Merges the batch and the real-time views into a consistent result

Lambda Architecture



Adapted from: Marz, N. & Warren, J. (2013) Big Data. Manning.

AGENDA

1. Big Data and Fast Data, what is it?
2. Architecting (Big) Data Systems
3. The Lambda Architecture
4. **Use Case and the Implementation**
5. Summary and Outlook

Project Definition

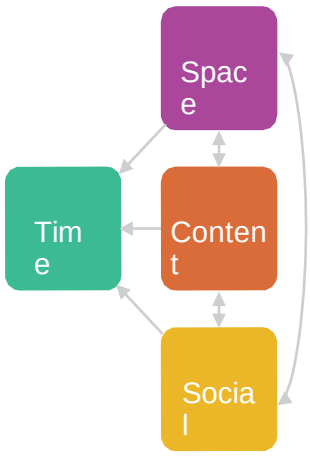
- Build a platform for analyzing Twitter communications in retrospective and in real-time
- Scalability and ability for future data fusion with other information is a must
- Provide a Web-based access to the analytical information
- Invest into new, innovative and not widely-proven technology
  - PoC environment, a pre-invest for future systems

Anatomy of a tweet

Time | Space | Content | Social | Technic

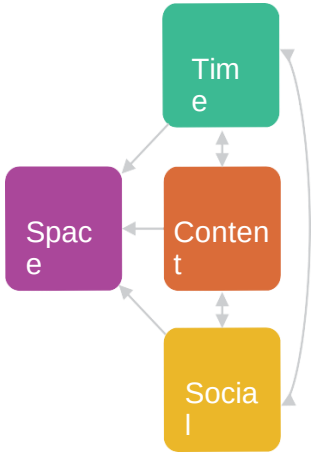

```
{
"created_at":"Sun Aug 18 14:29:11 +0000 2013",
"id":369103686938546176, "id_str":"369103686938546176",

"text":"Baloncesto preparaci\u00f3n Eslovenia, Rajoy derrota a Merkel. #quelosepash",
"source":"\u003ca href=\"http:\/\/twitter.com\/download\/iphone\" rel=\"nofollow\"
\u003eTwitter for iPhone\u003c\/a\u003e",
"truncated":false,
"in_reply_to_status_id":null, "in_reply_to_status_id_str":null,
"in_reply_to_user_id":null, "in_reply_to_user_id_str":null,
"in_reply_to_screen_name":null, "user":{
"id":15032594, "id_str":"15032594",
"name":"Juan Carlos Romo\u2122",
"screen_name":"jcsromo", "location":"Sopuerta, Vizcaya", "url":null,
"description":"Portugalujo, saturado de todo, de baloncesto no. Twitter personal.",
"protected":false,
"followers_count":1331, "friends_count":1326, "listed_count":31,
"created_at":"Fri Jun 06 21:21:22 +0000 2008", "favourites_count":255,
"utc_offset":7200, "time_zone":"Madrid", "geo_enabled":true, "verified":false,
"statuses_count":22787, "lang":"es", "contributors_enabled":false,
"is_translator":false,
...
"profile_image_url_https":"https:\/\/si0.twimg.com\/profile_images\/2649762203\
be4973d9eb457a45077897879c47c8b7_normal.jpeg",
"profile_banner_url":"https:\/\/pbs.twimg.com\/profile_banners\/15032594\/
1371570460",
"profile_link_color":"2FC2EF",
"profile_sidebar_border_color":"FFFFFF", "profile_sidebar_fill_color":"252429",
"profile_text_color":"666666", "profile_use_background_image":true,
"default_profile":false, "default_profile_image":false, "following":null,
"follow_request_sent":null, "notifications":null},

"geo":{
"type":"Point","coordinates":[43.28261499,-2.96464655]},
"coordinates":{"type":"Point","coordinates":[-2.96464655,43.28261499]},
"place":{"id":"cd43ea85d651af92",
"url":"https:\/\/api.twitter.com\/1.1\/geo\/id\/cd43ea85d651af92.json",
"place_type":"city",
"name":"Bilbao", "full_name":"Bilbao, Vizcaya", "country_code":"ES",
"country":"Espa\u00f1a",
"bounding_box":{"type":"Polygon","coordinates":[[[-2.9860102,43.2136542],
[-2.9860102,43.2901452],[-2.8803248,43.2901452],[-2.8803248,43.2136542]]]},
"attributes":{}},

"contributors": null, "retweet_count":0, "favorite_count":0,

"entities":{"hashtags":[{"text":"quelosepash","indices":[58,70]}], "symbols":[],
"urls":[],

"user_mentions":[]}, "favorited":false, "retweeted":false, "filter_level":"medium",

"lang":"es"
}
```

Views on Tweets in four dimensions



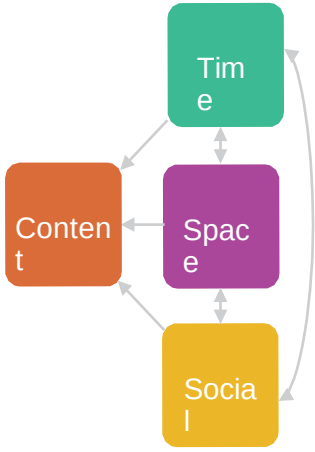
when ⇐
where+what+who

"*Time series*
"*Timelines*



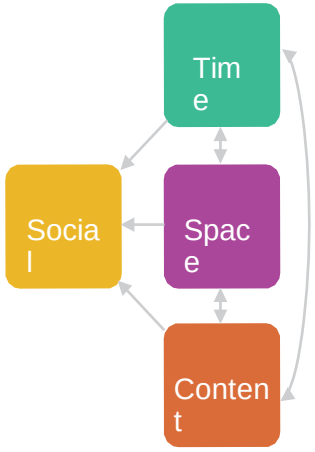
where ⇐
when+what+who

"*Geo maps*
"*Density plots*



what ⇐
when+where+who

"*Word clouds*
"*Topic trends*



who ⇐
when+where+what

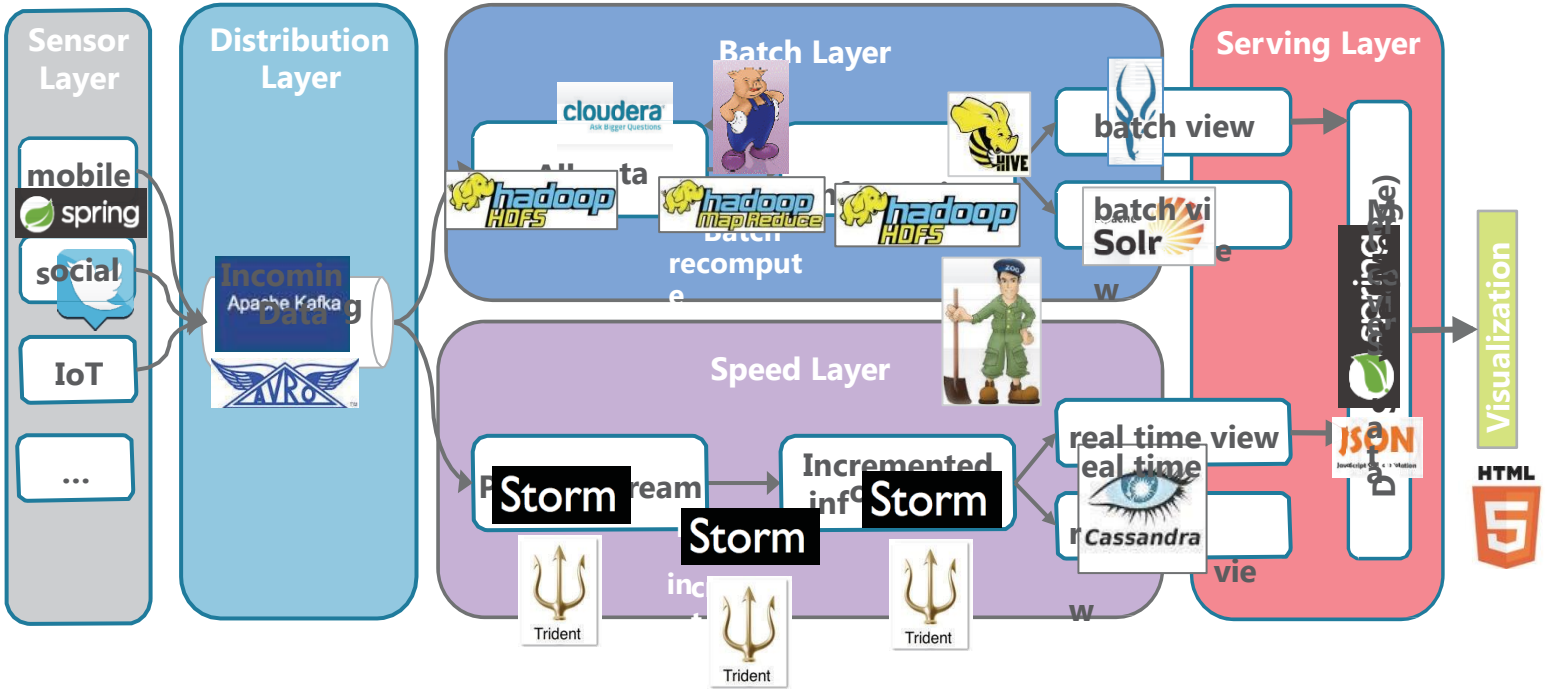
"*Social network graphs*
"*Activity graphs*

Accessing Twitter

Quelle	Limitierungen	Zugang
Twitter's Search API	3200 / user 5000 / keyword 180 Anfragen / 15 Minuten	gratis
Twitter's Streaming API	1%-40% des Volumens	gratis
DataSift	keine	0.15 -0.20$ / unit
Gnip	keine	Auf Anfrage

Lambda Architecture

Open Source Frameworks for implementing a Lambda Architecture



Lambda Architecture in Action

Twitter Horsebird Client (hbc) Distribution

- Twitter Java API over Streaming API

Spring Framework

- Popular Java Framework used to modularize part of the logic (sensor and serving layer)

Apache Kafka

- Simple messaging framework based on file system to distribute information to both batch and speed layer

Apache Avro

- Serialization system for efficient cross-language RPC and persistent data storage

JSON

- open standard format that uses human-readable text to transmit data objects consisting of attribute–value pairs.

Cloudera

Distribution of Apache Hadoop: HDFS, MapReduce, Hive, Flume, Pig, Impala

Cloudera Impala

- distributed query execution engine that runs against data stored in HDFS and HBase

Apache Zookeeper

- Distributed, highly available coordination service. Provides primitives such as distributed locks

Apache Storm & Trident

- distributed, fault-tolerant realtime computation system

Apache Cassandra

- distributed database management system designed to handle large amounts of data across many commodity servers, providing high availability with no single point of failure

Facts & Figures

14 active twitter feeds

- ~ 14 million tweets/day (> 5 billion tweets/year)
- ~ 8 GB/day raw data, compressed (2 DVDs)
- 66 GB storage capacity / day (replication & views/results included)

Cluster of 10 nodes

- ~100 processors
- ~40 TB HD capacity in total; 46% used
- >500 GB RAM

Currently in total

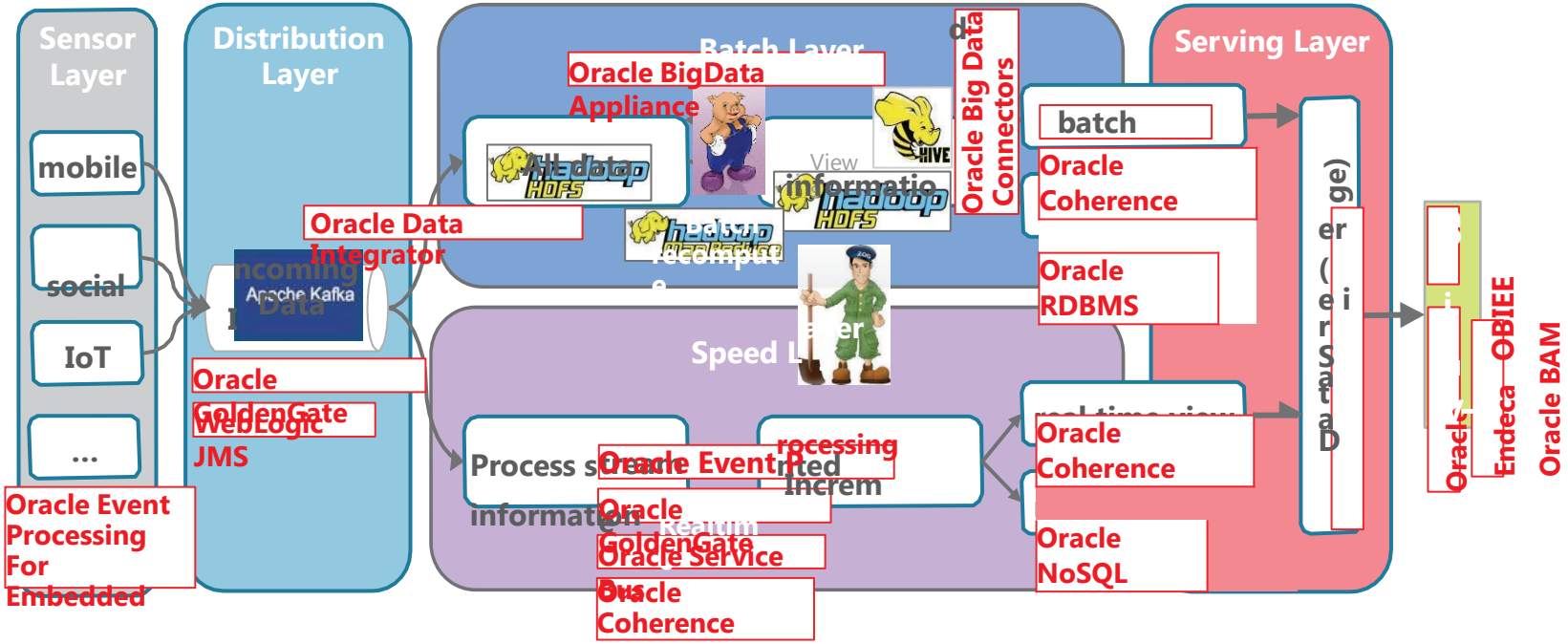
- 2.7 TB Raw Data
- 1.1 TB Pre-Processed data in Impala
- 1 TB Solr indices for full text search

Cloudera 4.7.0 with Hadoop, Pig, Hive, Impala and Solr

Kafka 0.7, Storm 0.9, DataStax Enterprise Edition

Lambda Architecture with Oracle Product Stack

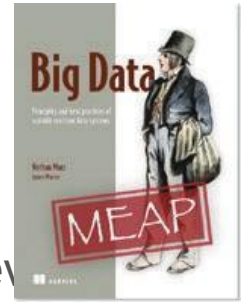
Possible implementation with Oracle Product stack



AGENDA

1. Big Data and Fast Data, what is it?
2. Architecting (Big) Data Systems
3. The Lambda Architecture
4. Use Case and the Implementation
5. **Summary and Outlook**

Summary – The lambda architecture

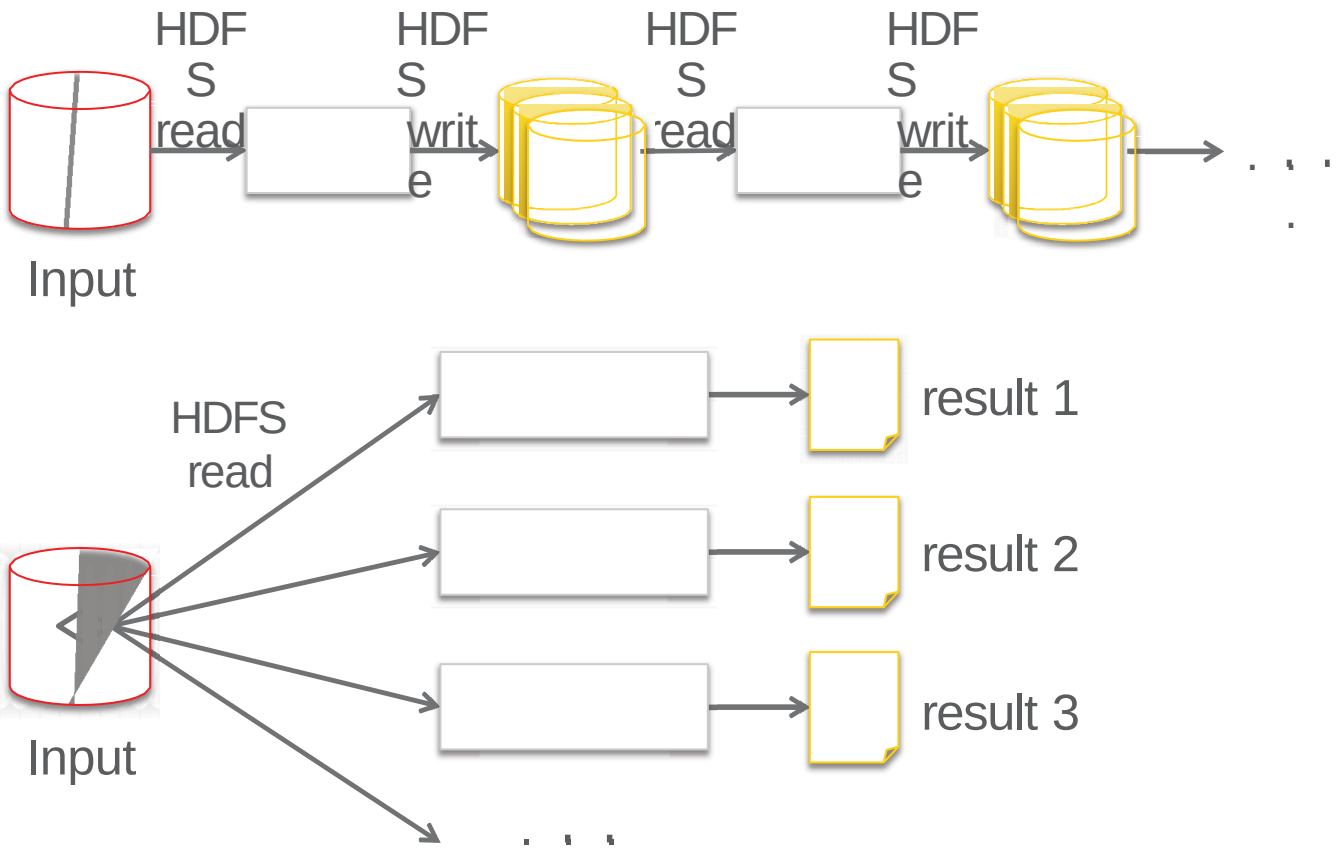


- Can discard batch views and real-time views and recreate ev from scratch
- Mistakes corrected via re-computation
- Scalability through platform and distribution
- Data storage layer optimized independently from query resolution layer
- Still in a early stage !. But a very interesting idea!
  - Today a zoo of technologies are needed => Infrastructure group might not like it
  - Better with so-called Hadoop distributions and Hadoop V2 (YARN)

Alternative Approaches – Motivation

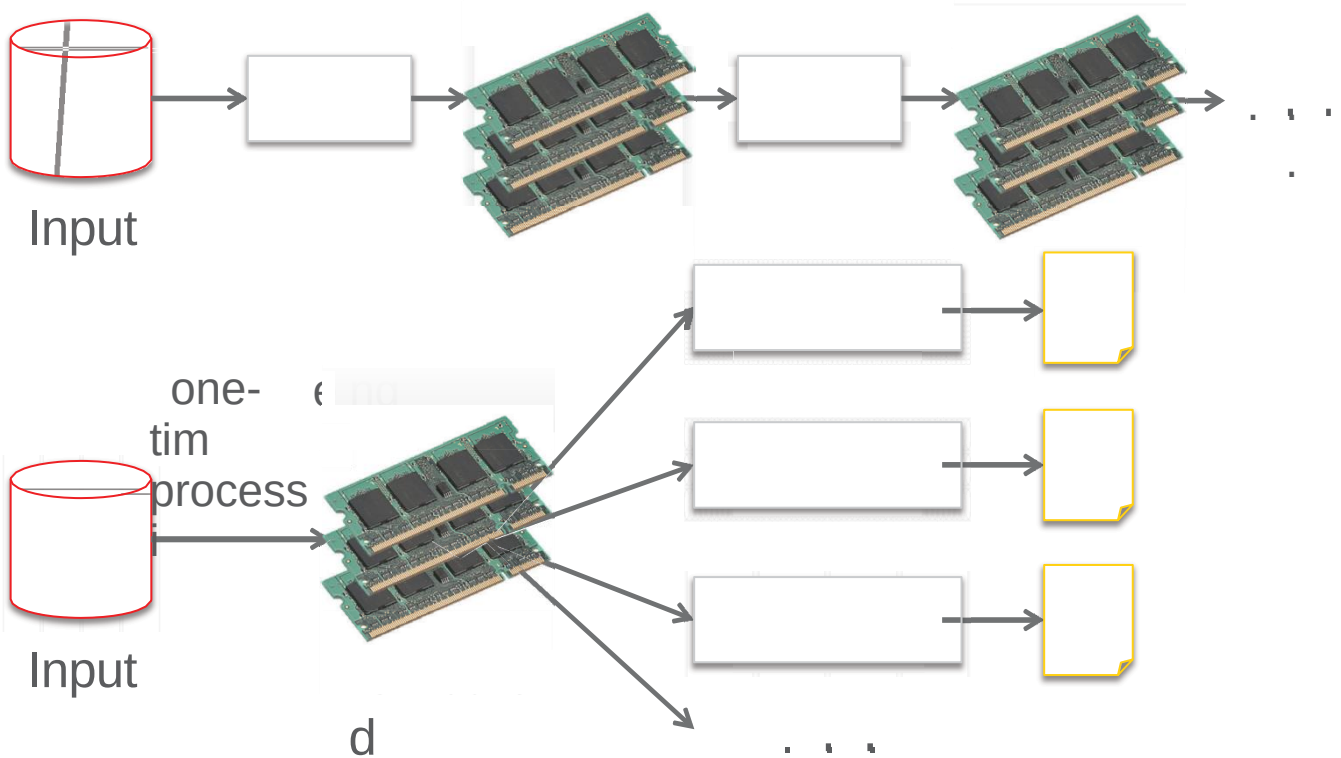
Data Sharing in Map Reduce

!

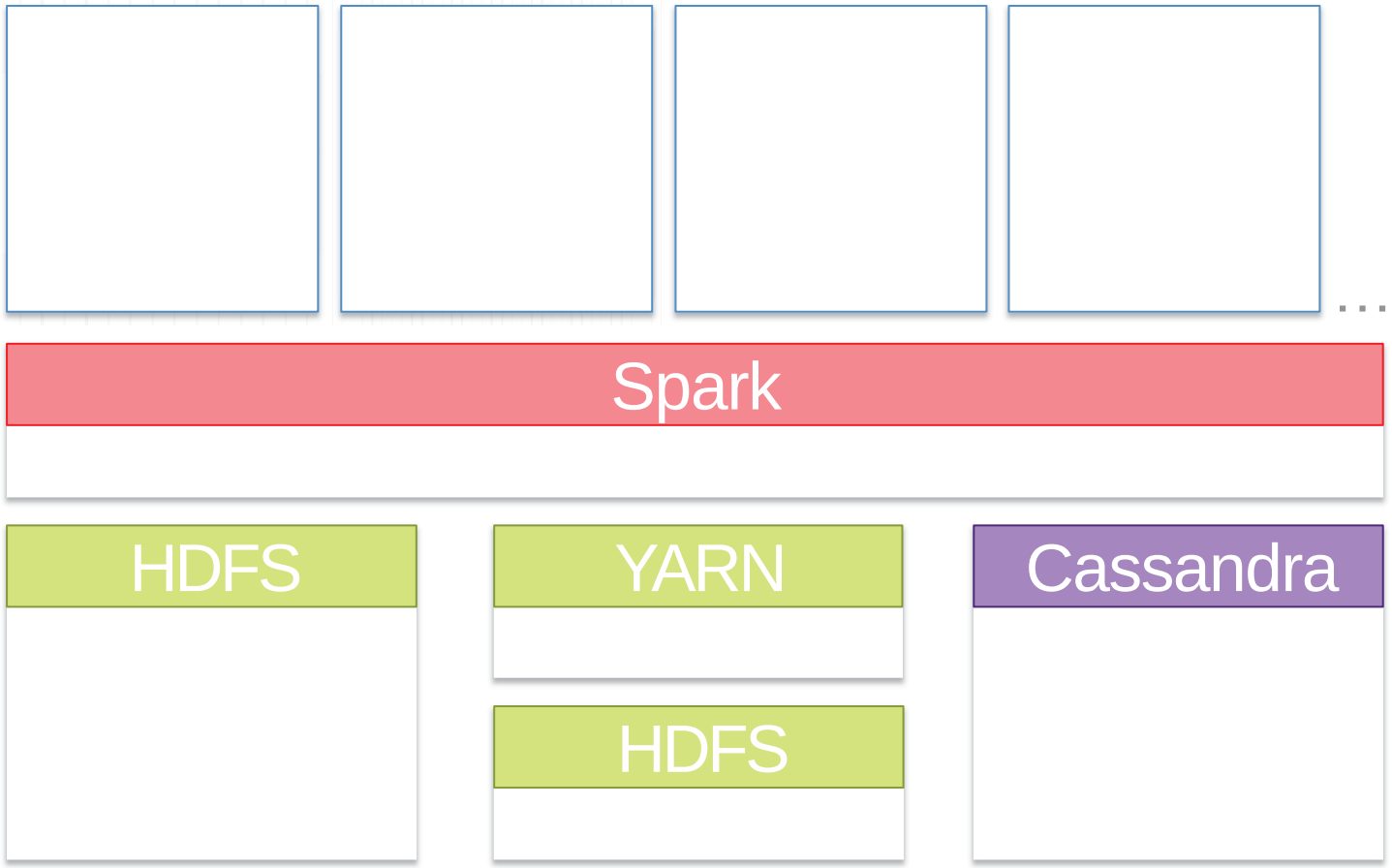


Alternative Approaches – Motivation

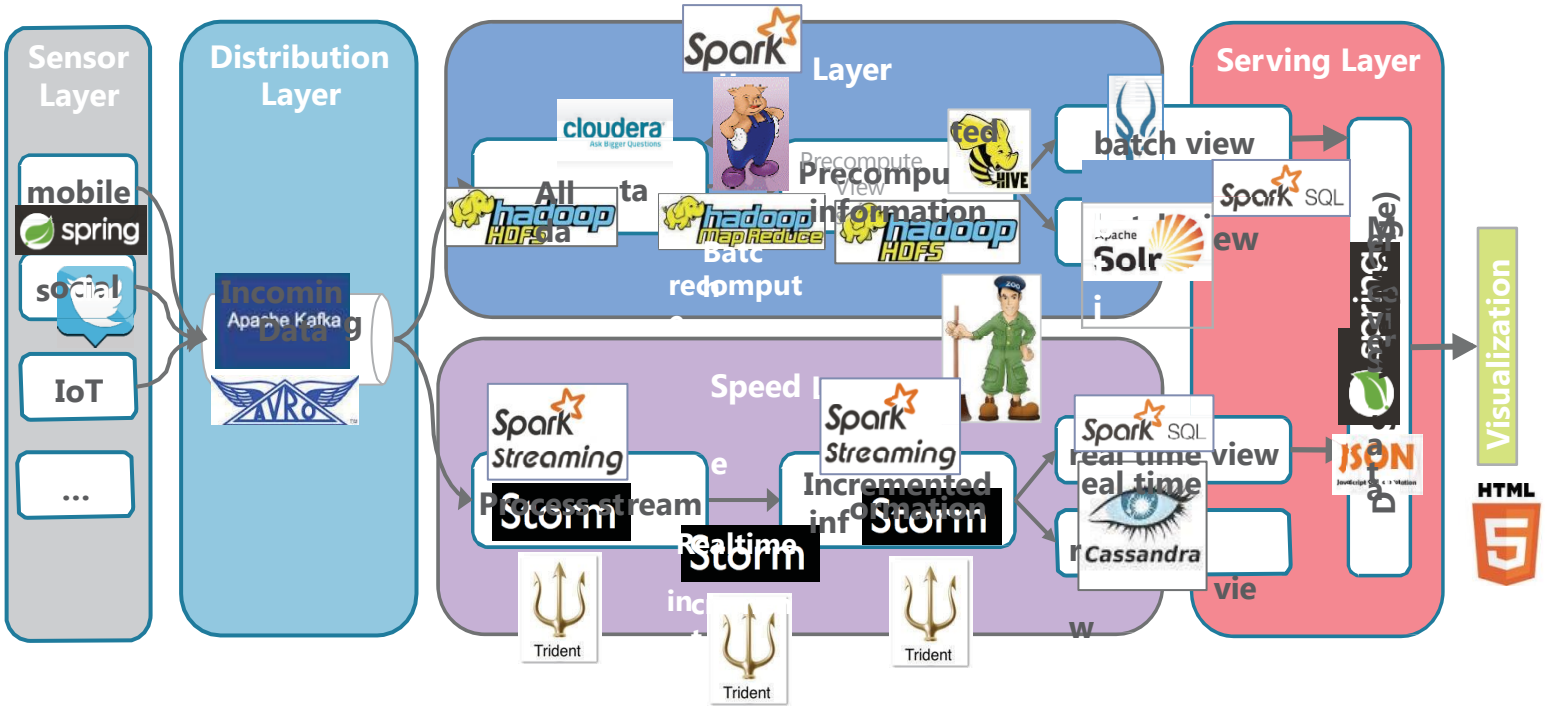
What we would like
!



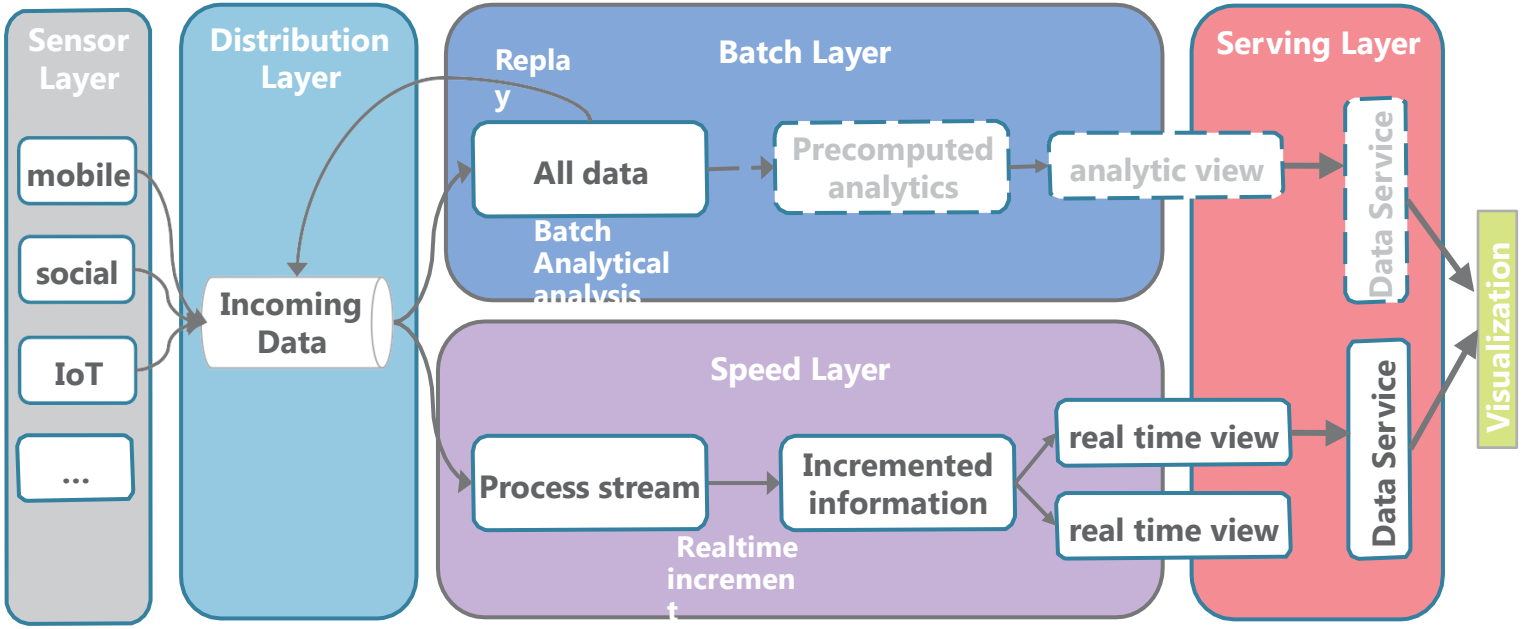
Alternatives – Apache Spark



Alternative Technologies – Apache Spark



“Kappa Architecture”



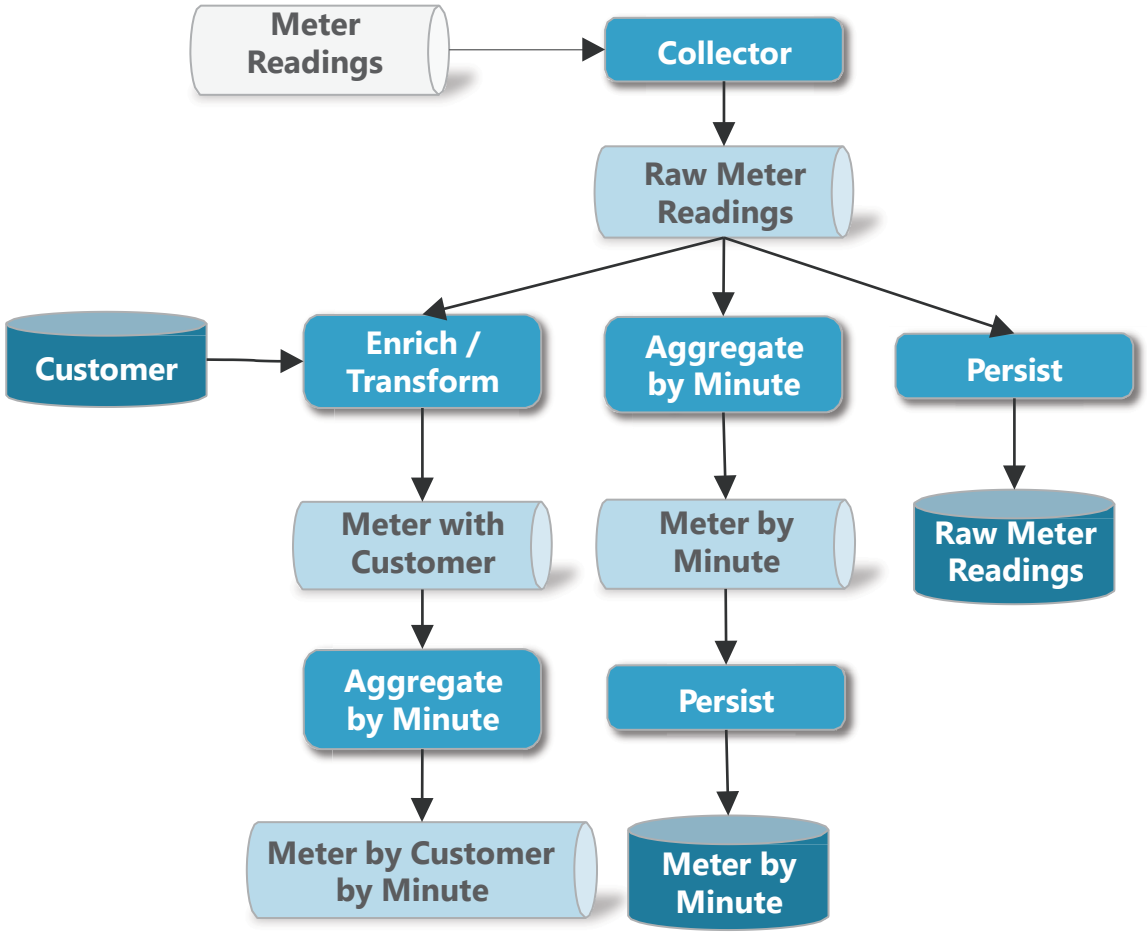
Adapted from: Marz, N. & Warren, J. (2013) Big Data. Manning.

Unified Log Processing Architecture

Stream processing allows
for computing feeds
off of other feeds

Derived feeds are no different
than original feeds
they are computed off

Single deployment of "Unified Log" but logically different feeds



Thank You :)