basho

Navigating Unstructured Data

Matt Brender
Developer Advocate

```
{
  "text": ”Woot! #GOTOChgo",
  "entities": {
    "hashtags": [“#GOTOChgo”],
    "symbols": [],
    "urls": [],
    "user_mentions": [{
      "screen_name": ”mjbrender",
      "name": ”Matt Brender",
      "id": 4948123,
      "id_str": ”42424242",
      "indices": [81, 92]
    }, {
      "screen_name": ”mjbrender",
      "name": ”Matt Brender",
      "id": 376825877,
      "id_str": "376825877",
      "indices": [121, 132]
    }]
  }
}
```

screen_name	text	hashtags	id	id_str
mjbrender	…	#GOTOChgo	376825877	4242342


25
%
SMS

Big Data


Big Data


Big Data

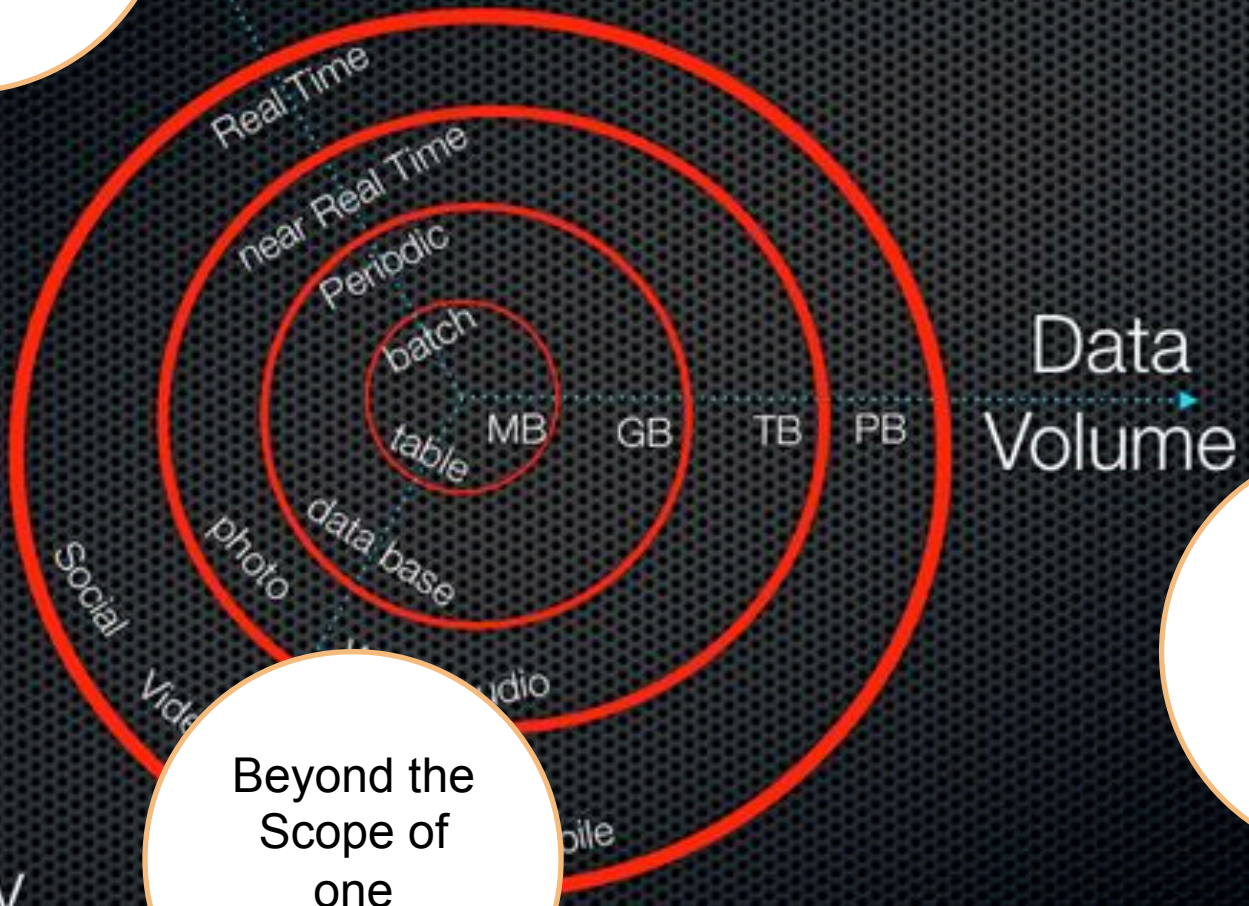
Big Data is

Big Data: Expanding on 3 fronts at an increasing rate.

And it's a distributed systems problem

Beyond the Scope of one tool

Big Data: Expanding on 3 fronts at an increasing rate.

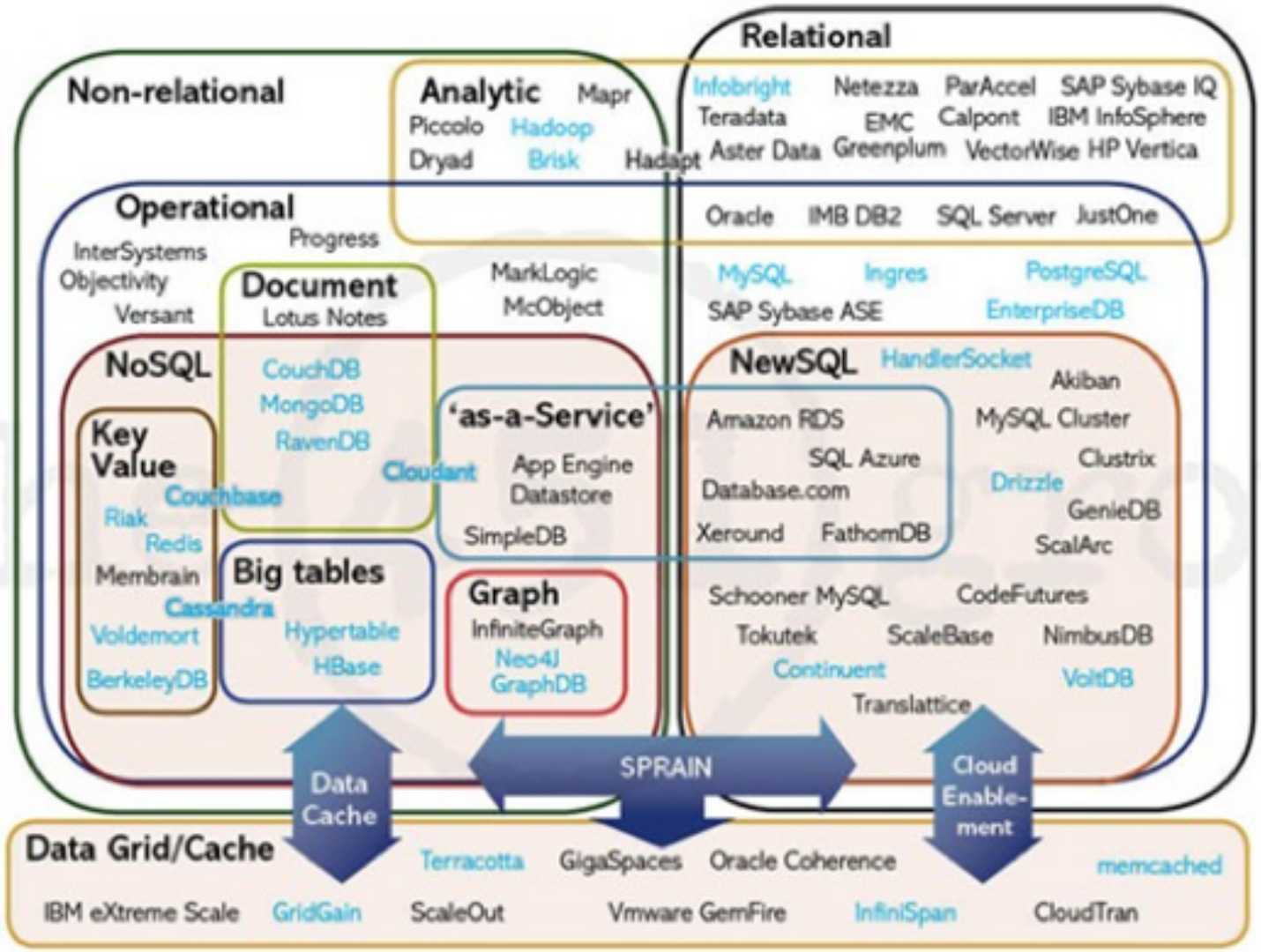


Beyond the Scope of one database

Beyond the Scope of one file system

Ergo, NoSQL

NoSQL is


Relational
Non-relational
Analytic
Mapr
Piccolo
Hadoop
Dryad
Brisk
Hadapt
Infobright
Netezza
ParAccel
SAP Sybase IQ
Teradata
EMC
Calpont
IBM InfoSphere
Aster Data
Greenplum
VectorWise
HP Vertica
Operational
Oracle
IMB DB2
SQL Server
JustOne
Progress
InterSystems
Objectivity
Versant
Document
Lotus Notes
MarkLogic
McObject
MySQL
Ingres
PostgreSQL
SAP Sybase ASE
EnterpriseDB
NoSQL
CouchDB
MongoDB
RavenDB
Cloudant
Key Value
Couchbase
Riak
Redis
Membrain
Cassandra
Voldemort
BerkeleyDB
Big tables
Hypertable
HBase
'as-a-Service'
App Engine Datastore
SimpleDB
Graph
InfiniteGraph
Neo4J
GraphDB
NewSQL
HandlerSocket
Akiban
Amazon RDS
MySQL Cluster
SQL Azure
Clustrix
Drizzle
Database.com
GenieDB
Xeround
FathomDB
ScalArc
Schooner MySQL
CodeFutures
Tokutek
ScaleBase
NimbusDB
Continuent
VoltDB
Translattice
Data Cache
SPRAIN
Cloud Enable-ment
Data Grid/Cache
Terracotta
GigaSpaces
Oracle Coherence
memcached
IBM eXtreme Scale
GridGain
ScaleOut
Vmware GemFire
InfiniSpan
CloudTran

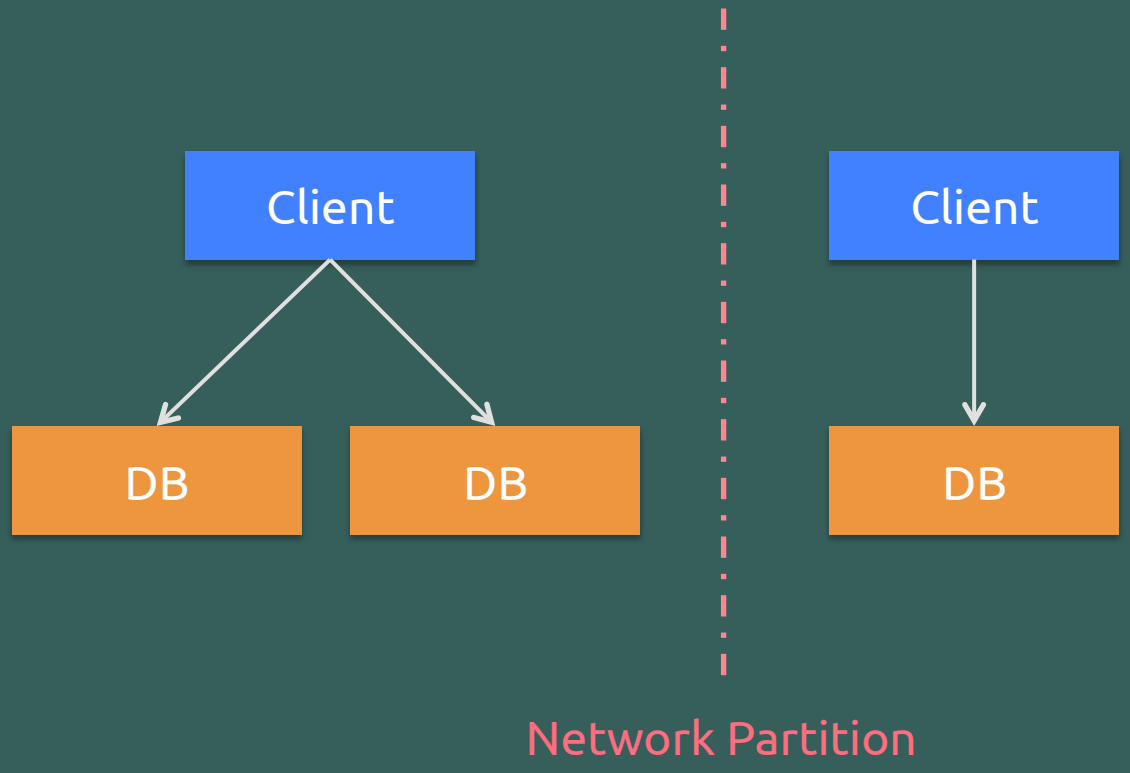
For Good Reason

1. Consistency Level
2. Conflict Resolution
3. Partitioning Strategy

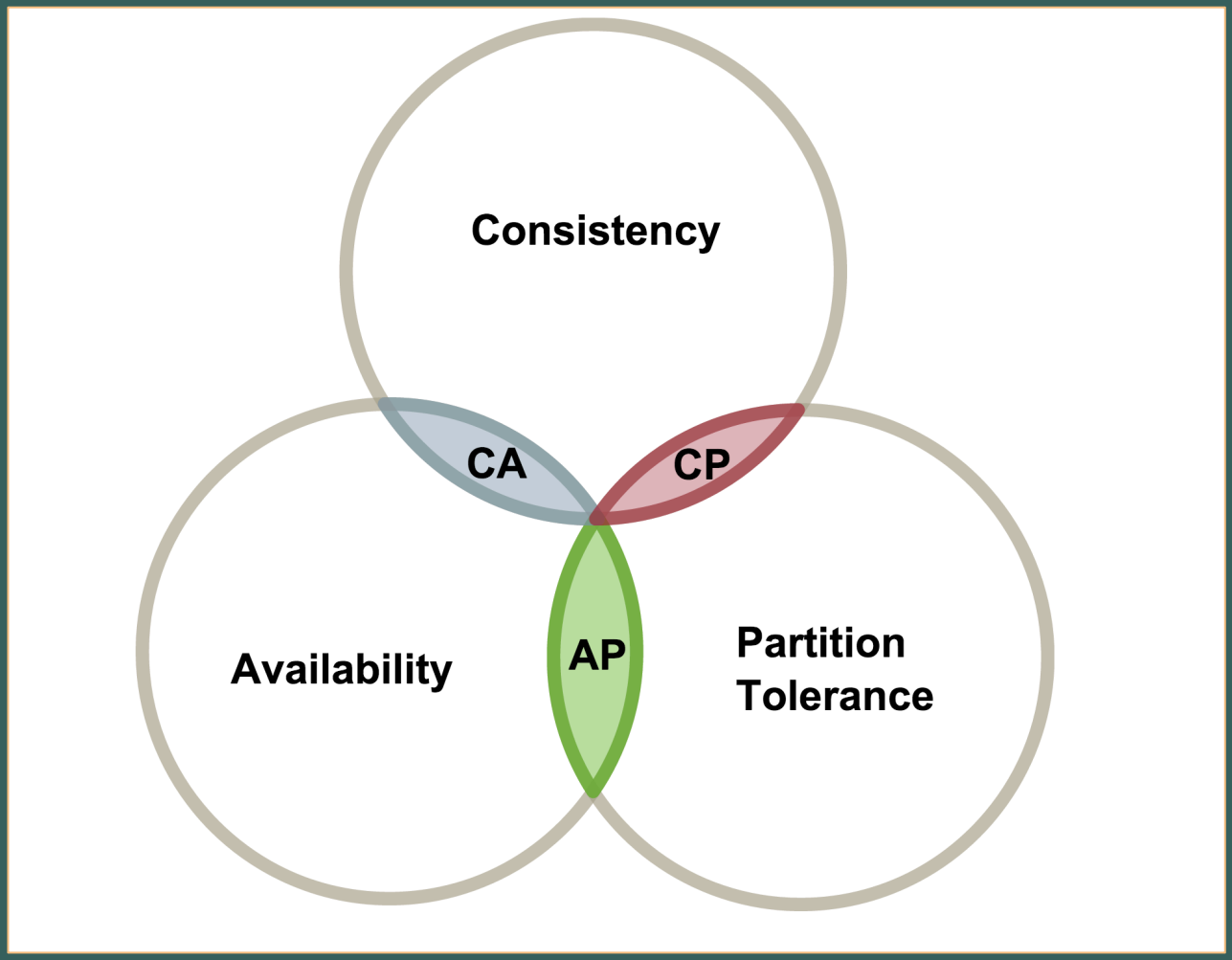
Brewer's Conjecture

Cap theorem states that a distributed system can at most support 2 out of these 3 properties

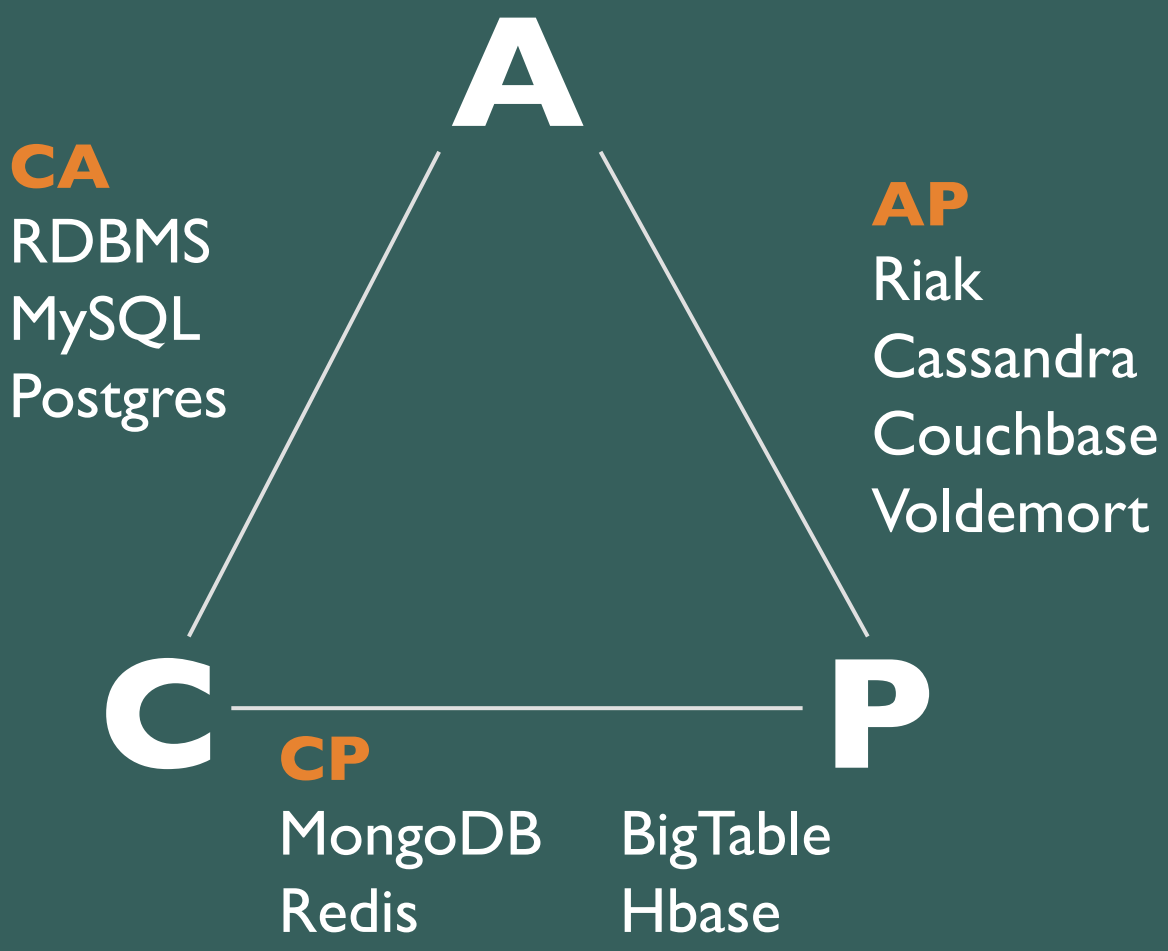
C = Consistency
A = Availability
P = Partition Tolerance



Consistency Level



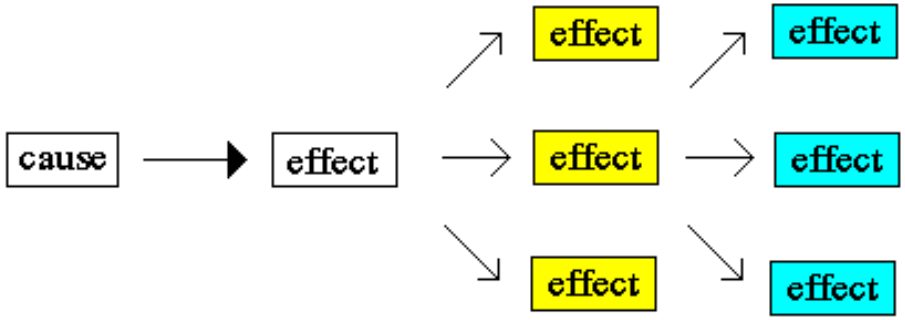
Consistency Level



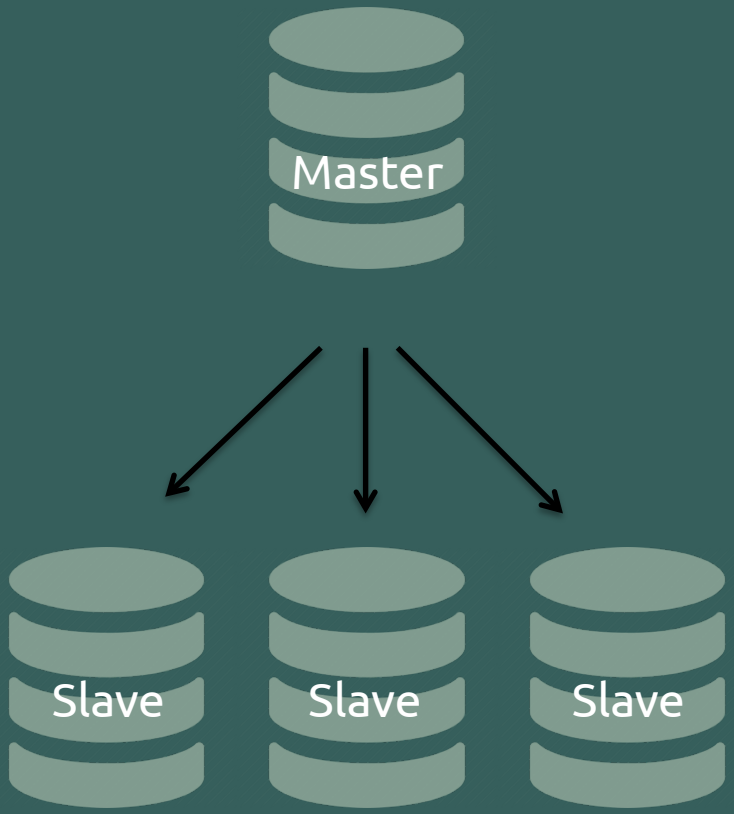
Conflict Resolution

Last Write Wins
vs.
Causal Context

Conflict Resolution



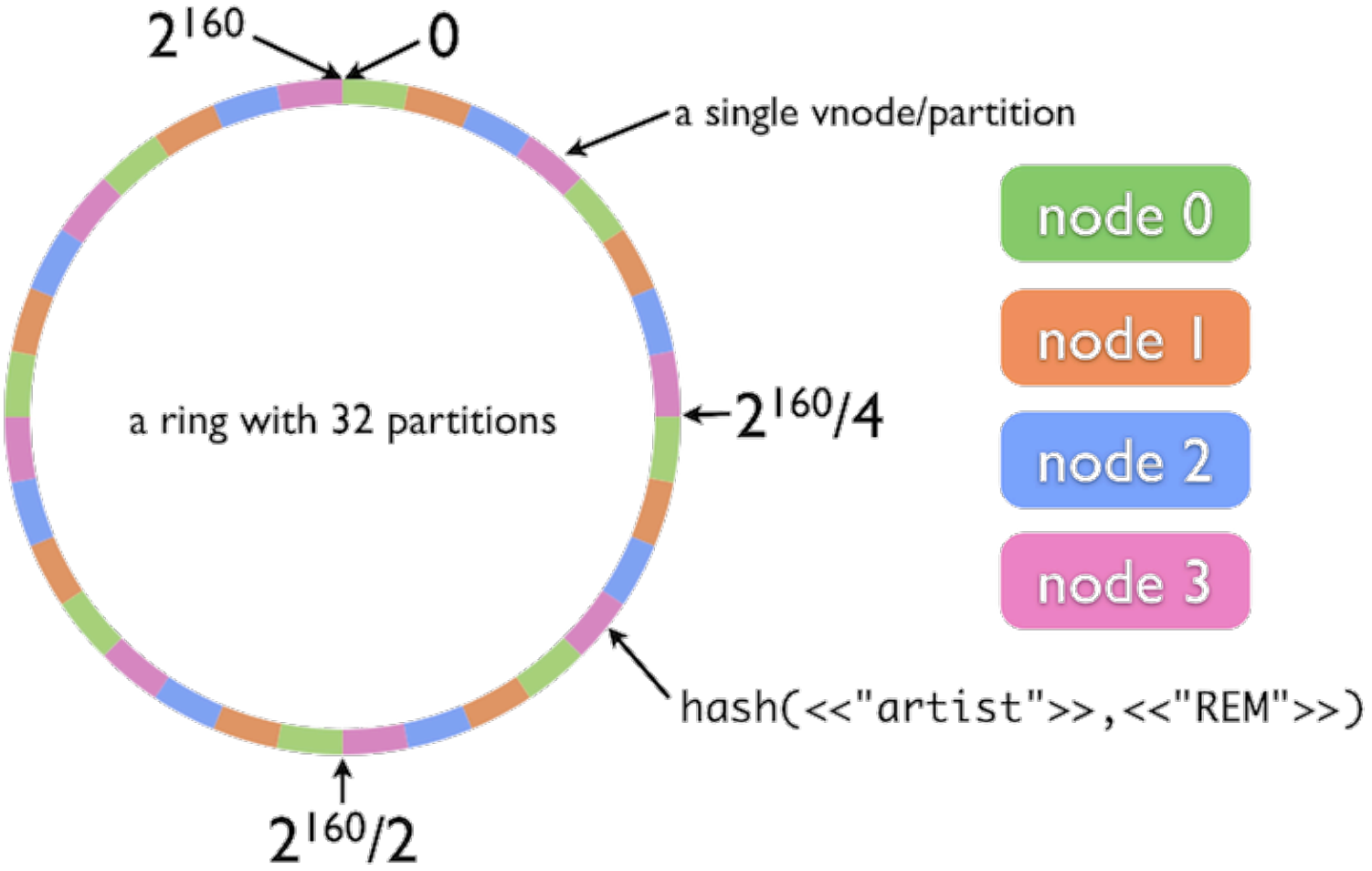
Partition Strategy



OR

Node 1 Node 2 Node 3

Partition Strategy

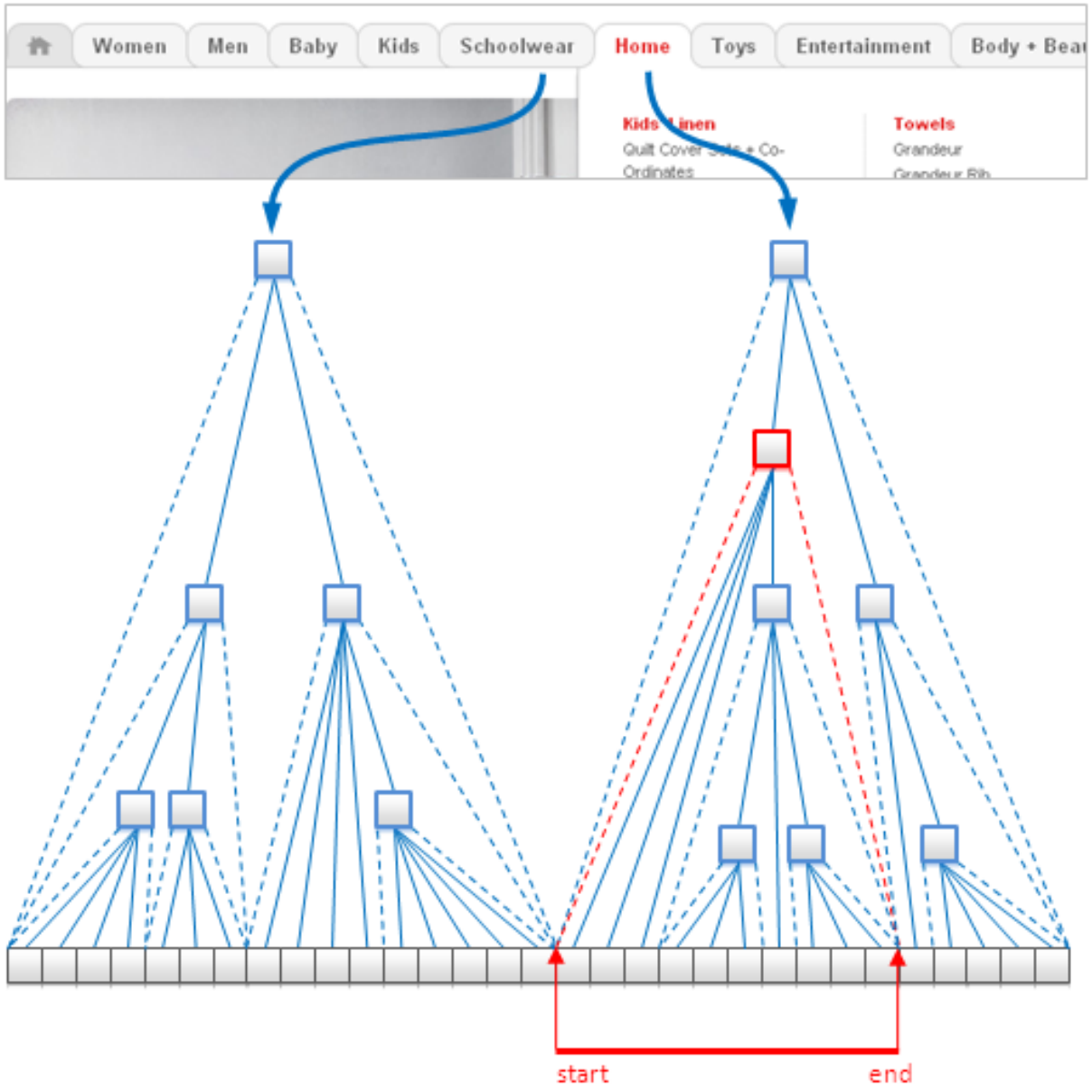


A few tactics

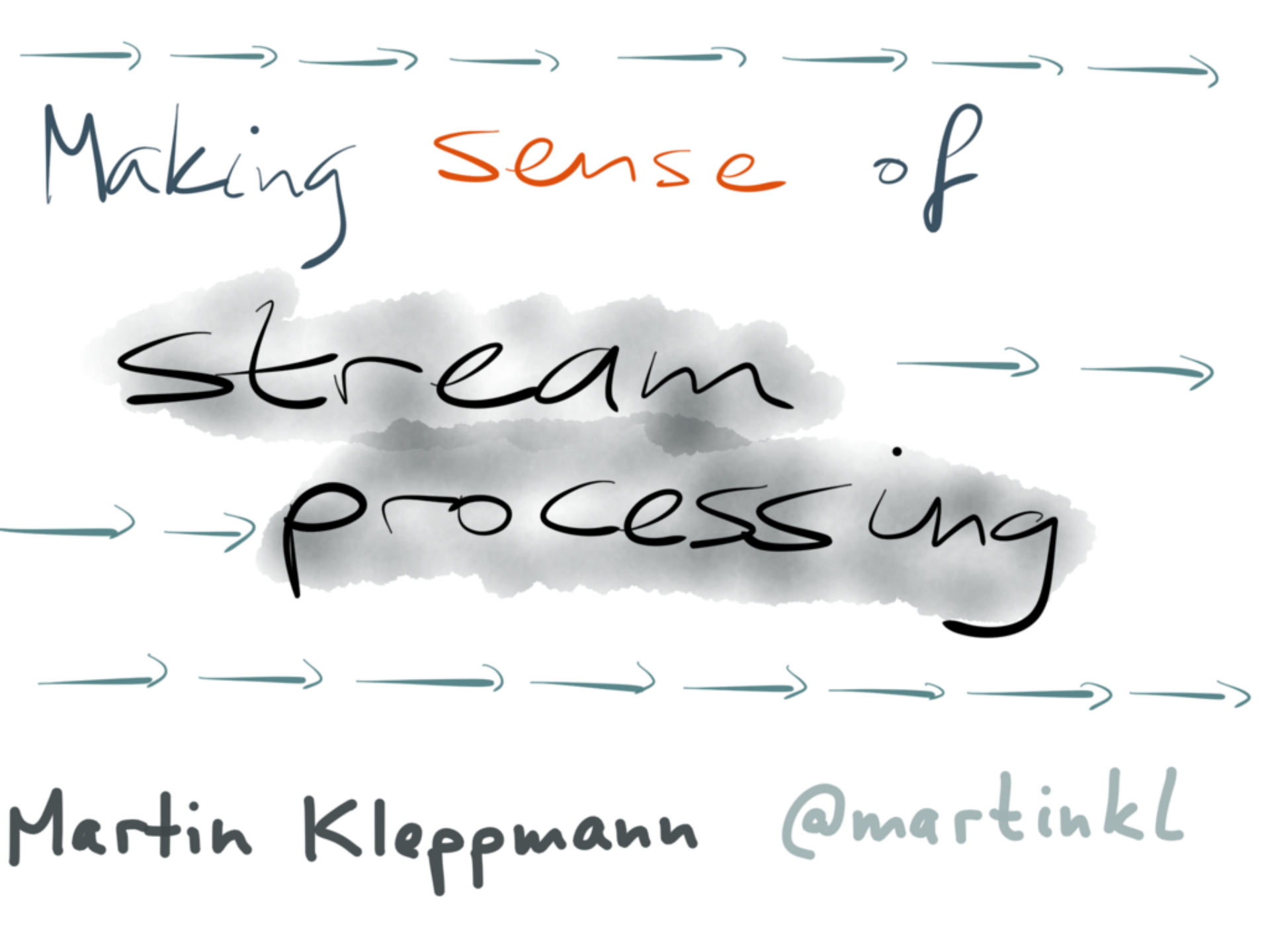
basho

What kind of questions do you need to ask?


GOSSE —
MEXICANS


Women
Men
Baby
Kids
Schoolwear
Home
Toys
Entertainment
Towels
Grandeur
start
end

No Updates.
Only Truth.

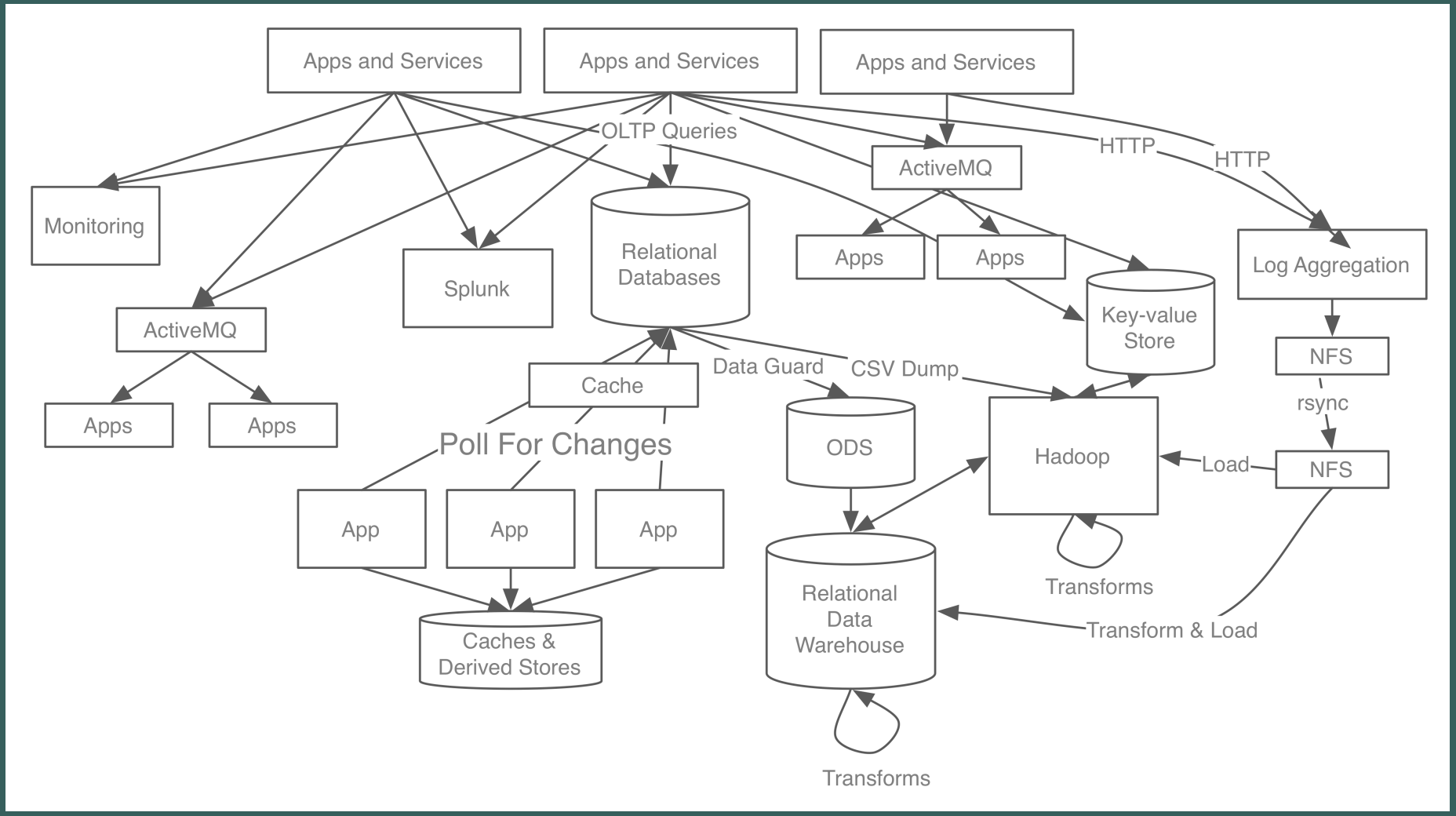

Making sense of
stream
processing
Martin Kleppmann @martinkl

Immutable commands

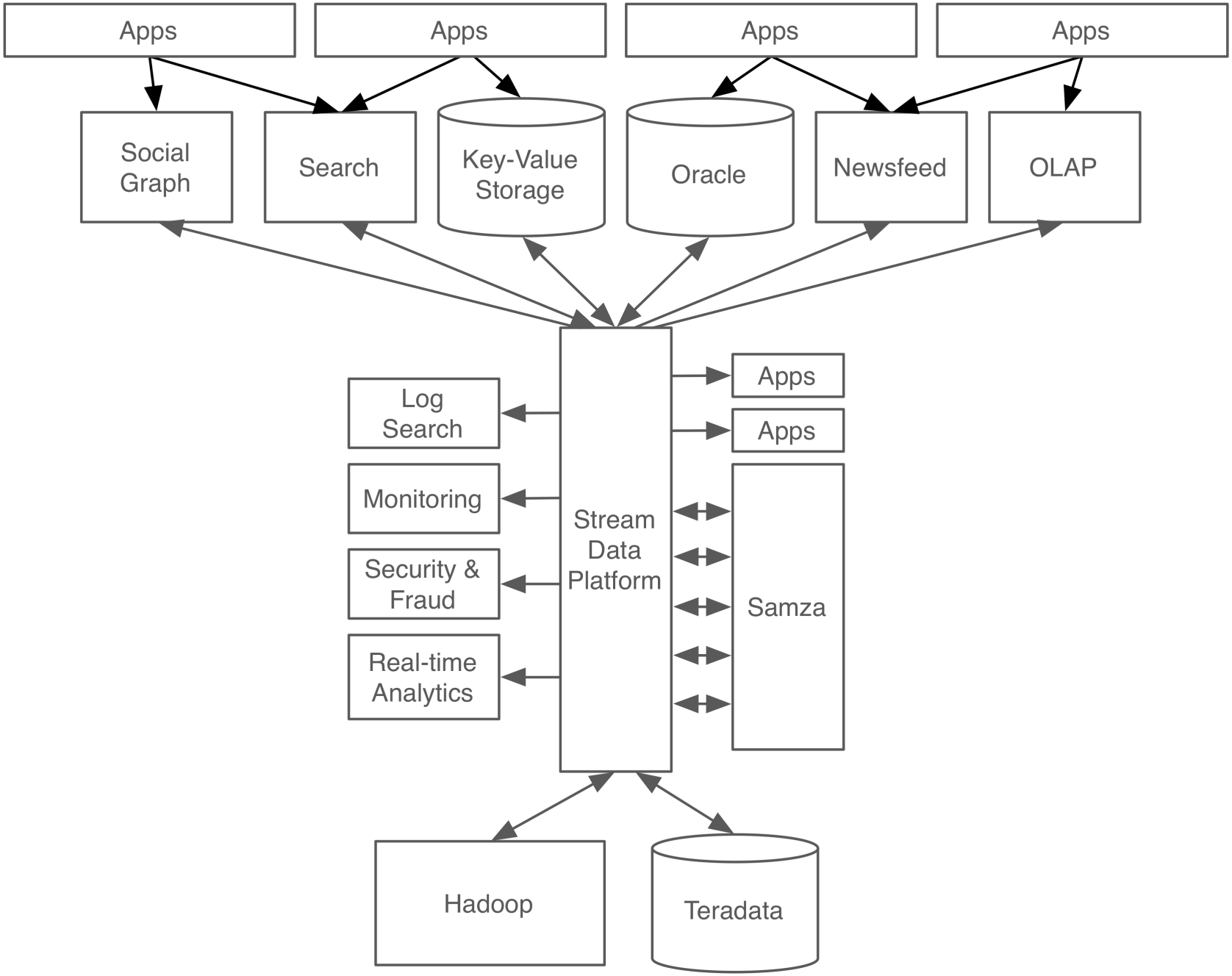
```
AddToCart(cust=123,
    prod=888, quantity=1)
UpdateCartQuantity(cust=123,
    prod=888, quantity=3)
UpdateCartQuantity(cust=123,
    prod=888, quantity=2)
CheckoutCart(cust=123)
```

Architectures

Report on this


Apps and Services
Apps and Services
Apps and Services
OLTP Queries
HTTP
HTTP
ActiveMQ
Monitoring
Splunk
Relational Databases
Apps
Apps
Log Aggregation
Key-value Store
ActiveMQ
NFS
Data Guard
CSV Dump
Cache
rsync
Apps
Apps
Poll For Changes
ODS
Hadoop
Load
NFS
App
App
App
Transforms
Relational Data Warehouse
Transform & Load
Caches & Derived Stores
Transforms

vs


Apps
Apps
Apps
Apps
Social Graph
Search
Key-Value Storage
Oracle
Newsfeed
OLAP
Log Search
Monitoring
Security & Fraud
Real-time Analytics
Stream Data Platform
Apps
Apps
Samza
Hadoop
Teradata

Error Analysis?
NoSQL + Solr

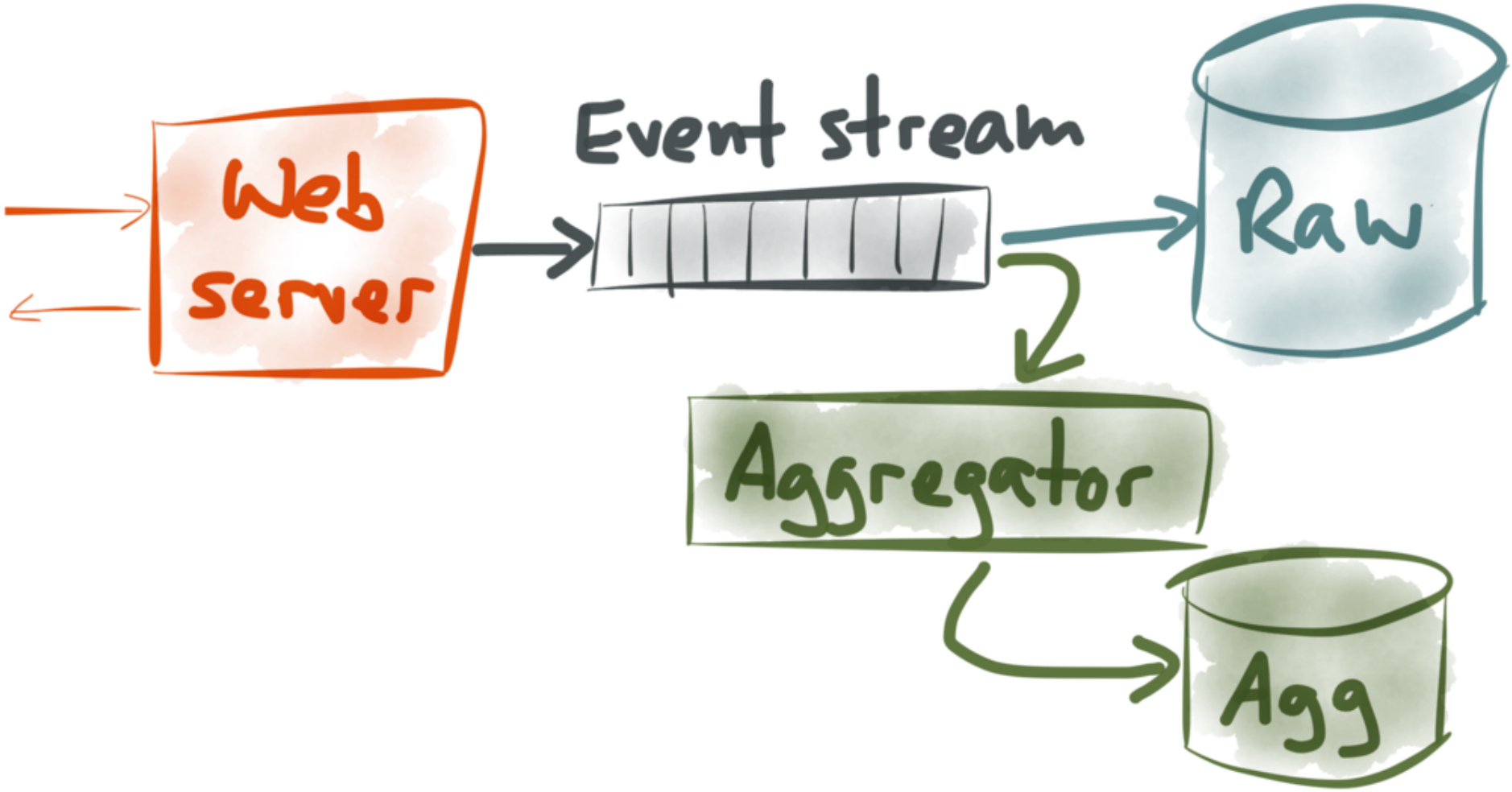
Patterns?
Multi-client writes to NoSQL & HDFS

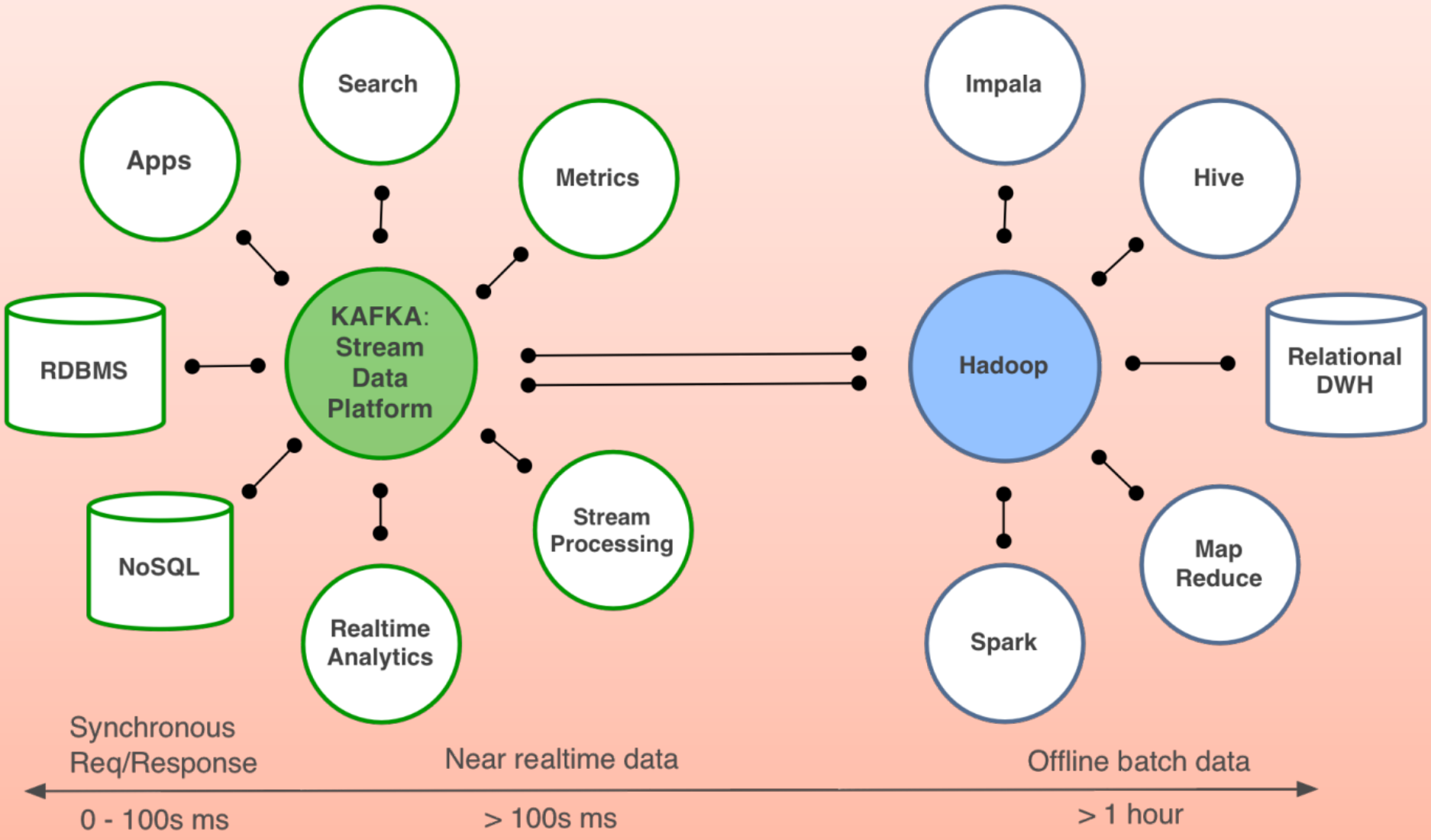
OR

NoSQL + ETL process => Message Queue => Spark and/or Hadoop M/R

OR

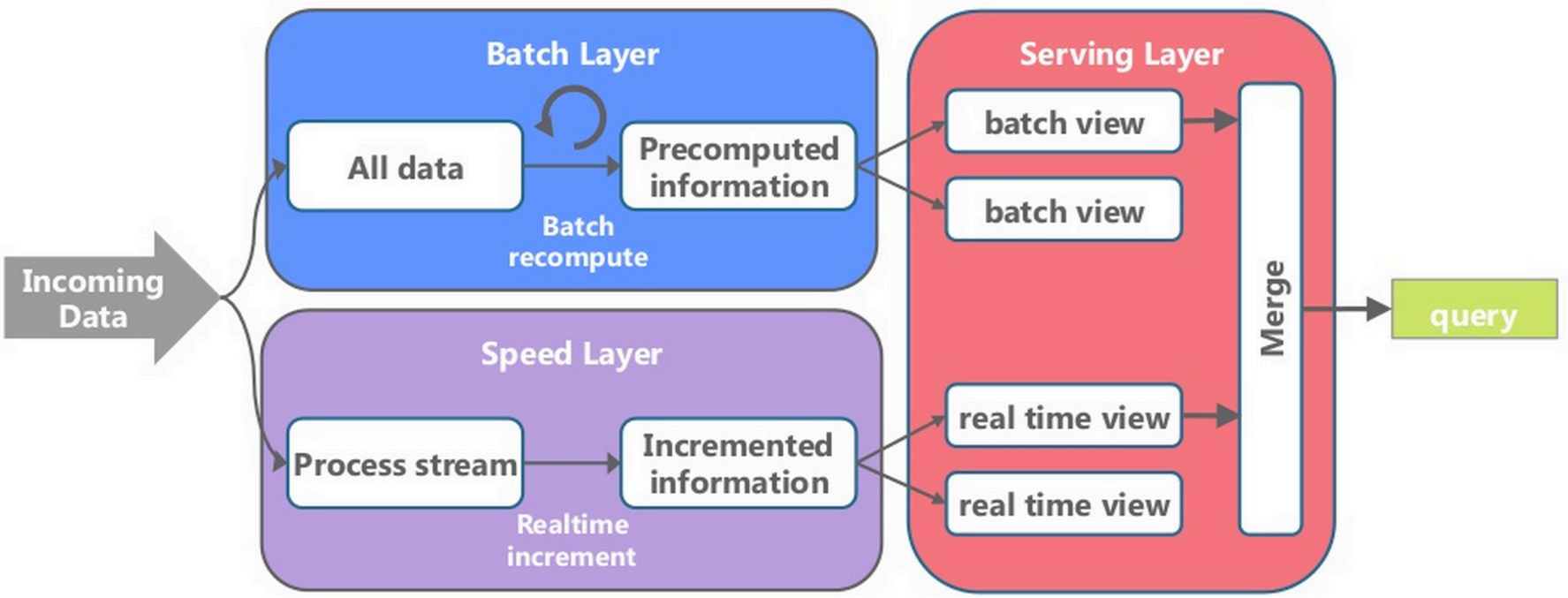
Streaming aggregation




Search
Apps
Metrics
RDBMS
KAFKA:
Stream
Data
Platform
NoSQL
Stream
Processing
Realtime
Analytics
Impala
Hive
Hadoop
Relational
DWH
Map
Reduce
Spark
Synchronous
Req/Response
Near realtime data
Offline batch data
0 - 100s ms
> 100s ms
> 1 hour

OR

LAMDA



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

In Review

You can’t analyze what you don’t have.

And you don’t want an analysis system to be unreliable.

everything works

at small scale

Summary

- **NoSQL**
  A collection of highly scalable, highly available systems that fall within CAP theorem
- **Unstructured Data**
  Data that cannot – or should not – be put into a relational database
- **Immutable Data**
  Modeling data as raw states as oppose to relative updates
- **Distributed System**
  A tough problem to solve
- **Big Data**
  Any computation that requires multiple computers

Summary

- **Lambda Architecture**
  A strategy to provide data processing in batch and real-time simaltaneously
- **Solr**
  Apache project for indexing text for search
- **Kafka**
  Distributed scalable pub/sub messaging queue
- **Hadoop**
  A framework that allows for the distributed processing of large data sets across clusters
- **Spark**
  A fast, general engine for large-scale data processing
- **Storm**
  A distributed real-time computation system

Many thanks for all the concepts I stole:

- Martin Kleppman's post on Streaming
- Confluent's Blog
- Highly Scalable Blog

Thank You!

Matt Brender

@mjbrender