NoSQL

Enterprise Architectures for Big Data

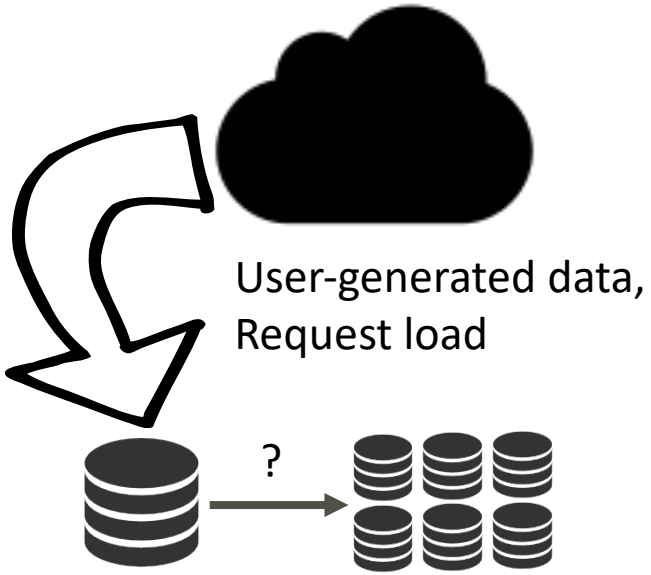
NoSQL

- „NoSQL“ term coined in 2009
- Interpretation: „Not Only SQL“

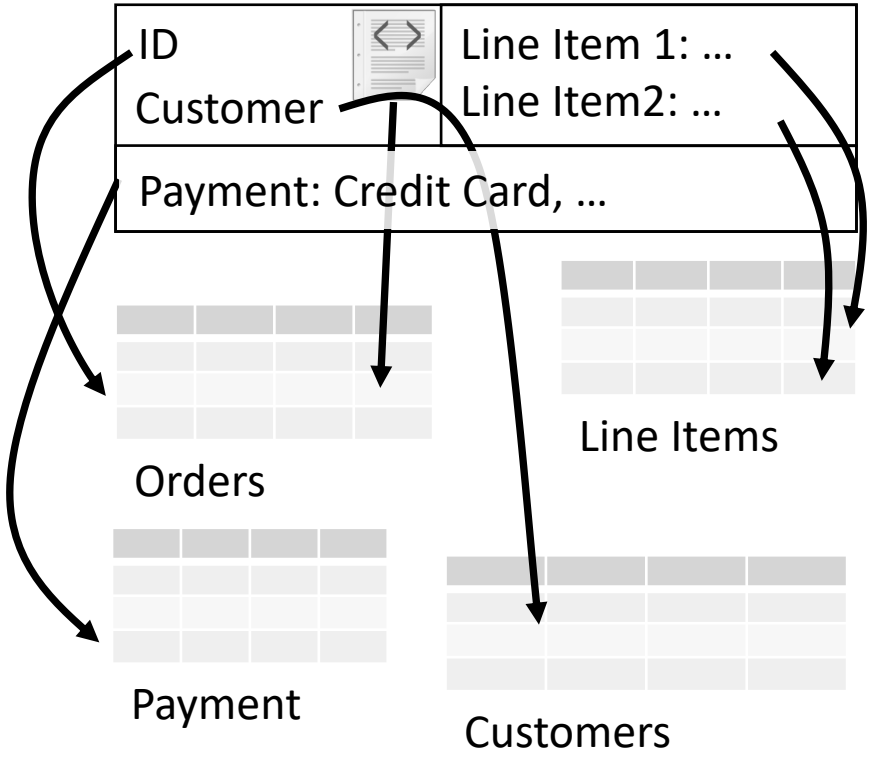
- Typical properties:
  - Non-relational
  - Often Open-Source
  - Schema-less (schema-free)
  - Optimized for distribution (clusters)
  - Tunable consistency

Two main motivations

Scalability



Impedance Mismatch



NoSQL Paradigm Shift
Open Source & Commodity Hardware

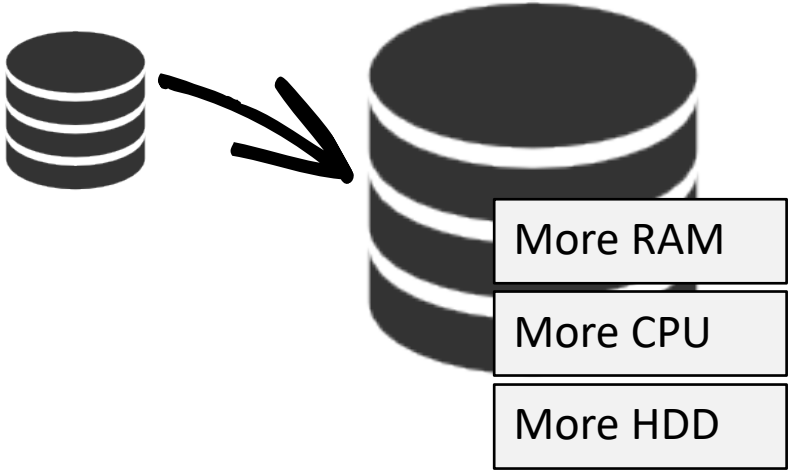
Commercial DBMS	Open-Source DBMS
Specialized DB hardware (Oracle Exadata, etc.)	Commodity hardware
Highly available network (Infiniband, Fabric Path, etc.)	Commodity network (Ethernet, etc.)
Highly Available Storage (SAN, RAID, etc.)	Commodity drives (standard HDDs, JBOD)

Source: Gessert (2016)

Scaling Strategies

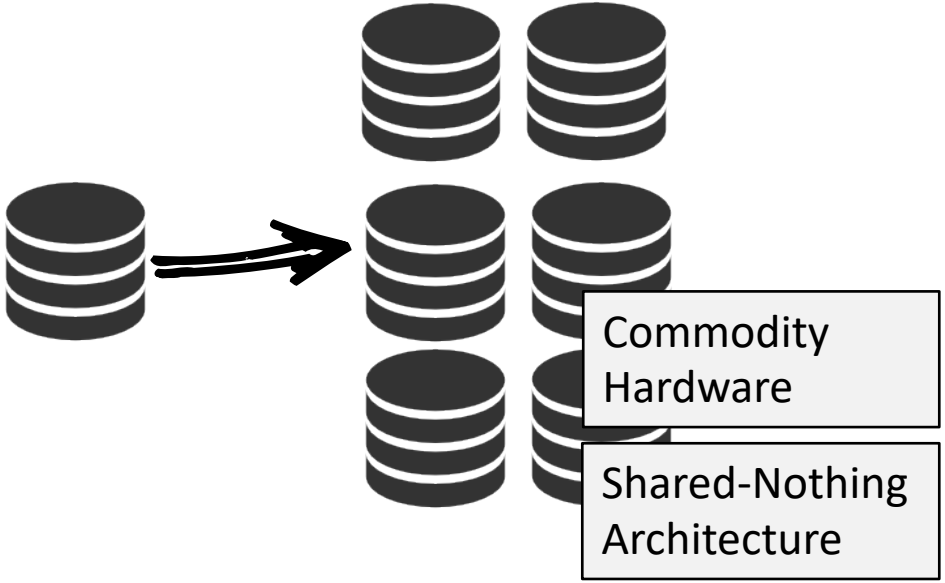
Scale-up

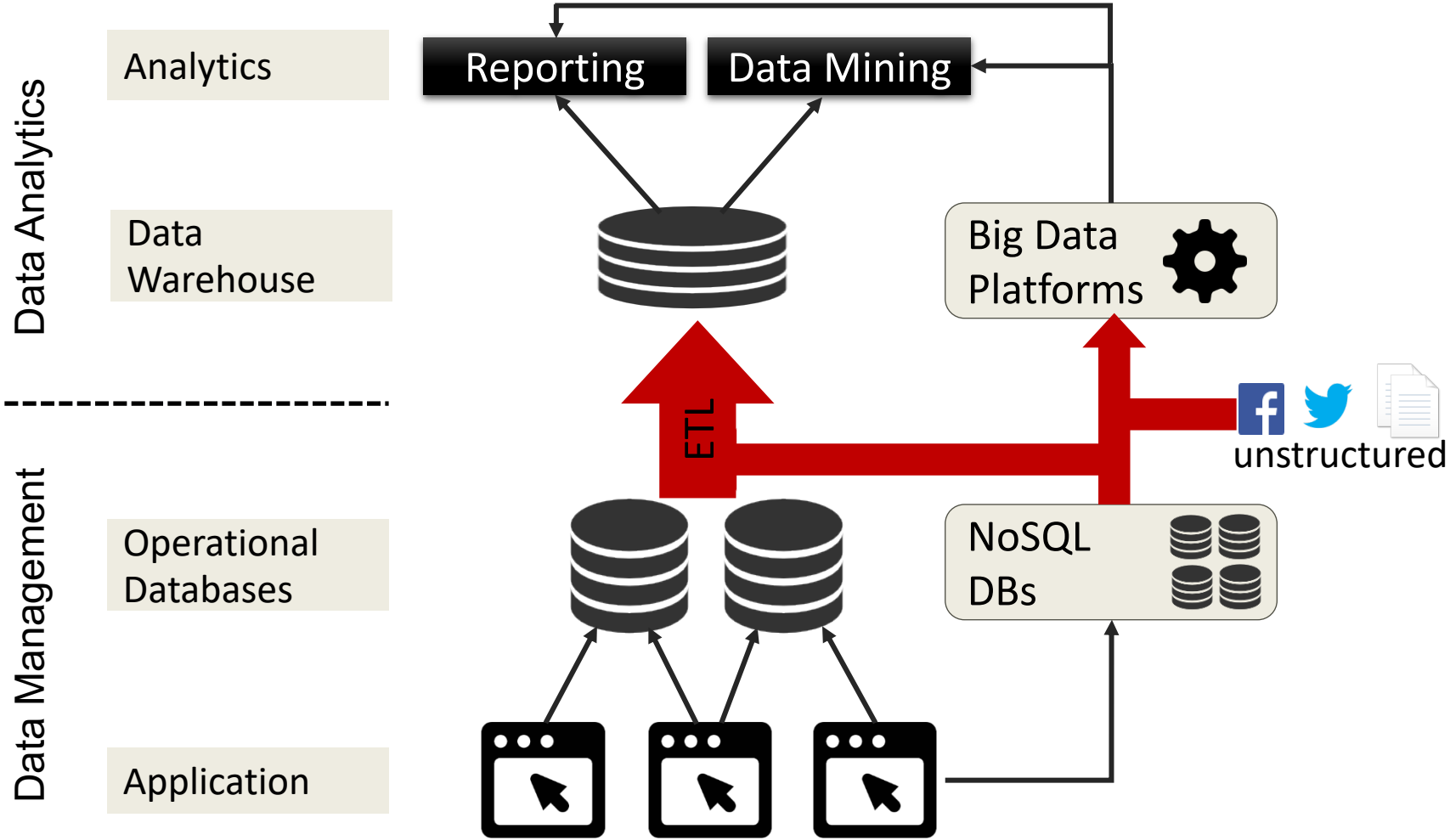
- Vertical Scaling



Scale-out

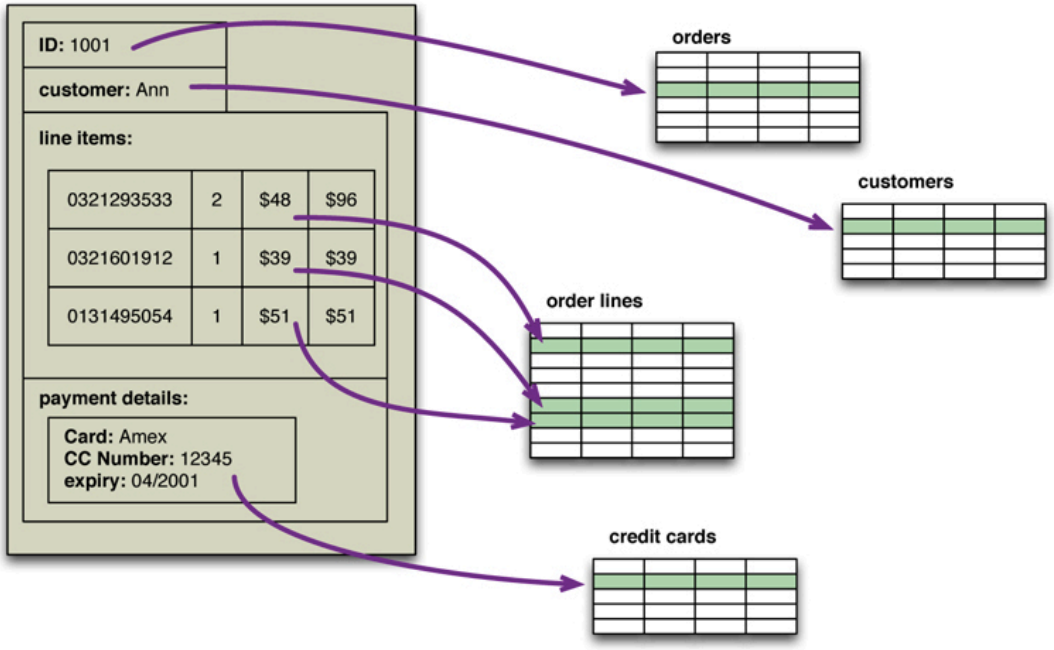
- Horizontal scaling




Data Analytics
Analytics
Reporting
Data Mining
Data Warehouse
Big Data Platforms
ETL
unstructured
Data Management
Operational Databases
NoSQL DBs
Application

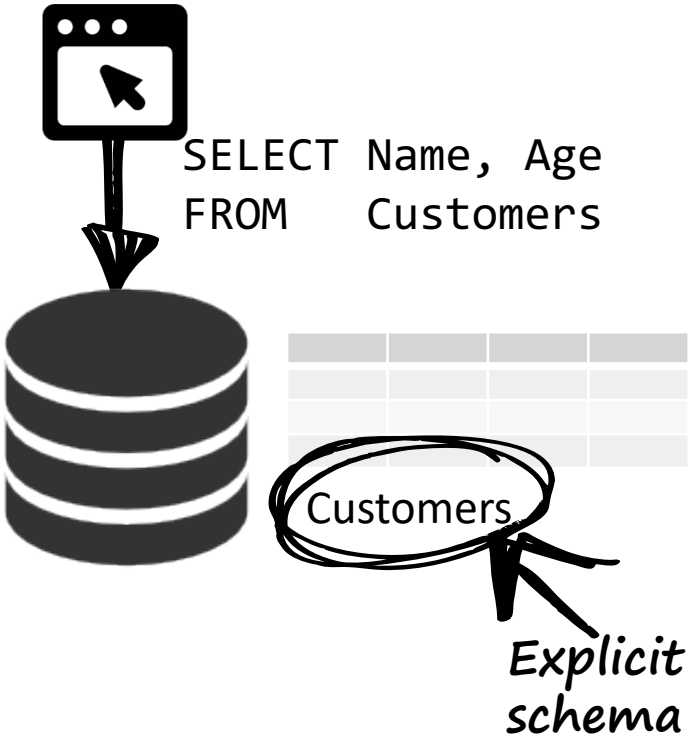
Impedance Mismatch

- An order are many tables in a relational database

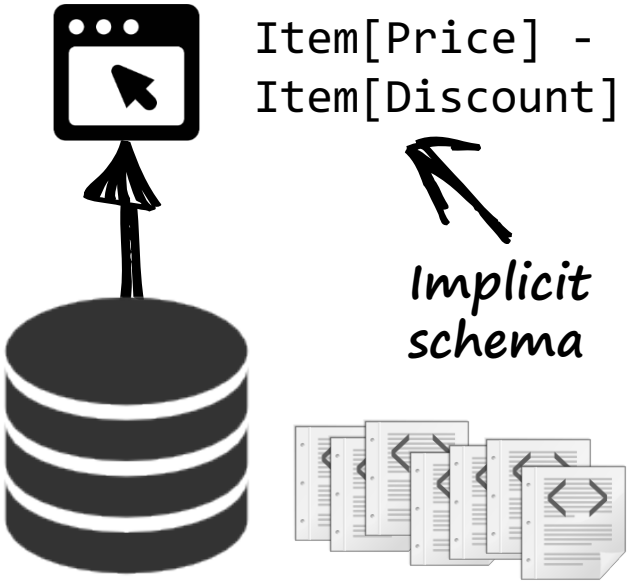


Schema Free Data Modeling

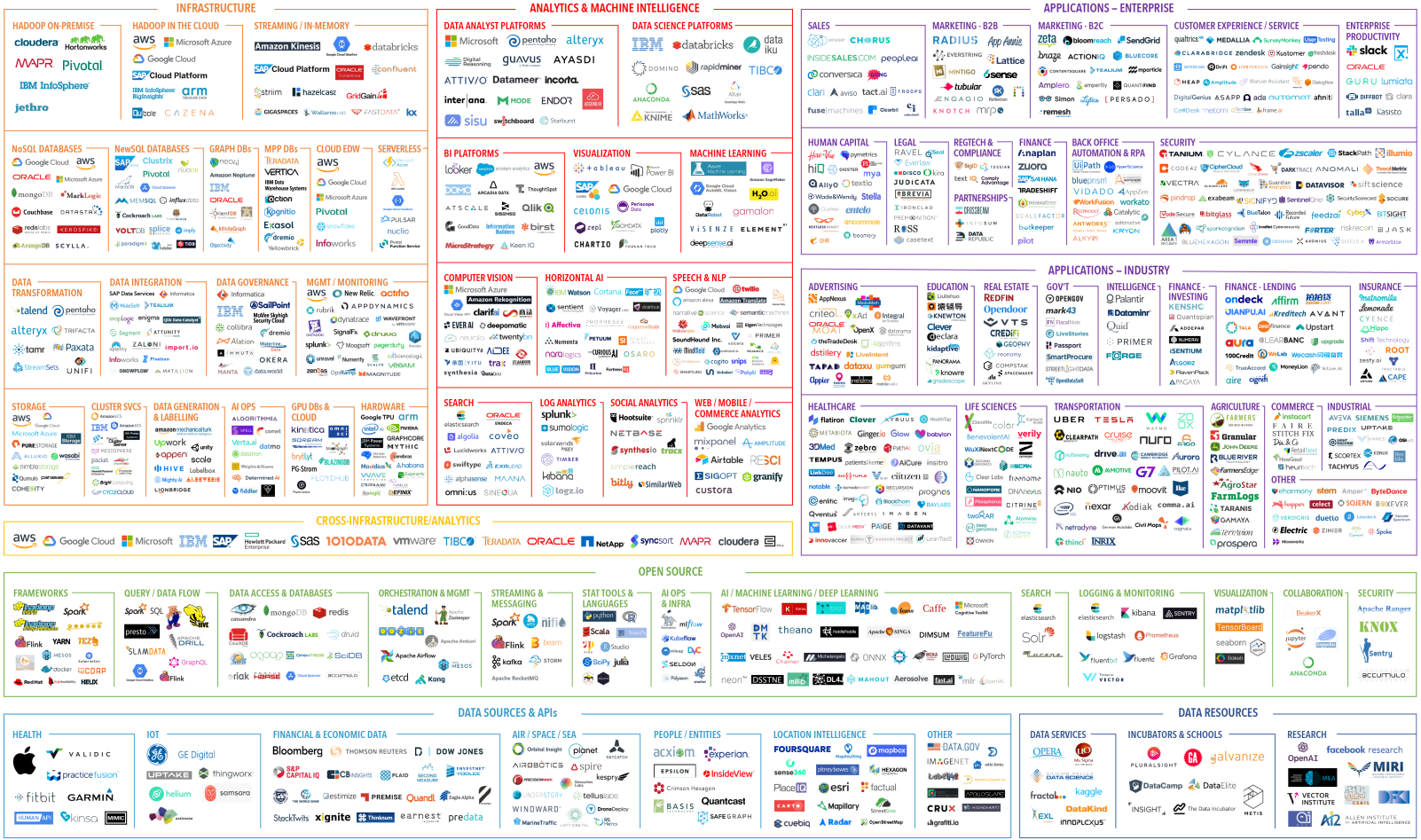
RDBMS



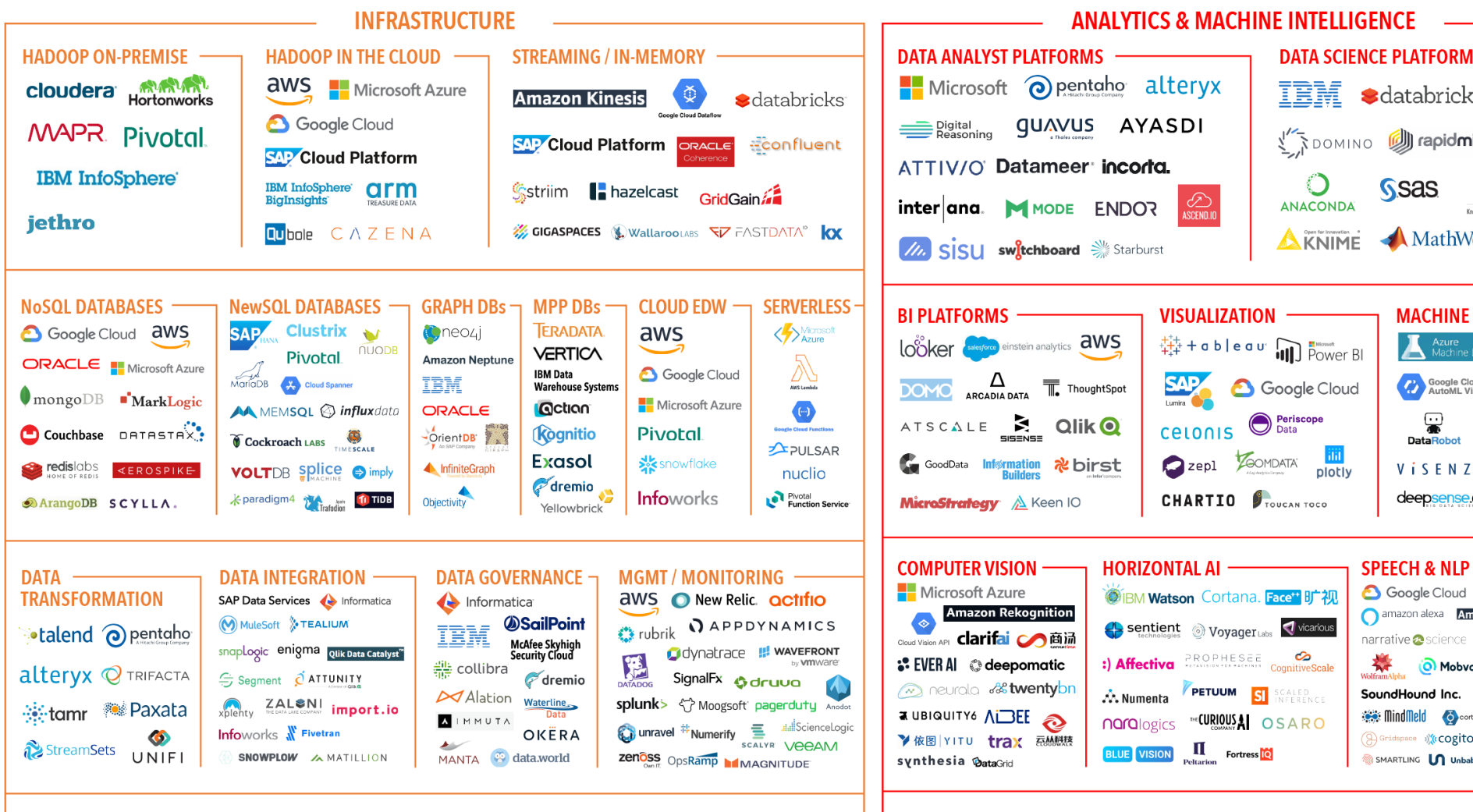
NoSQL DB

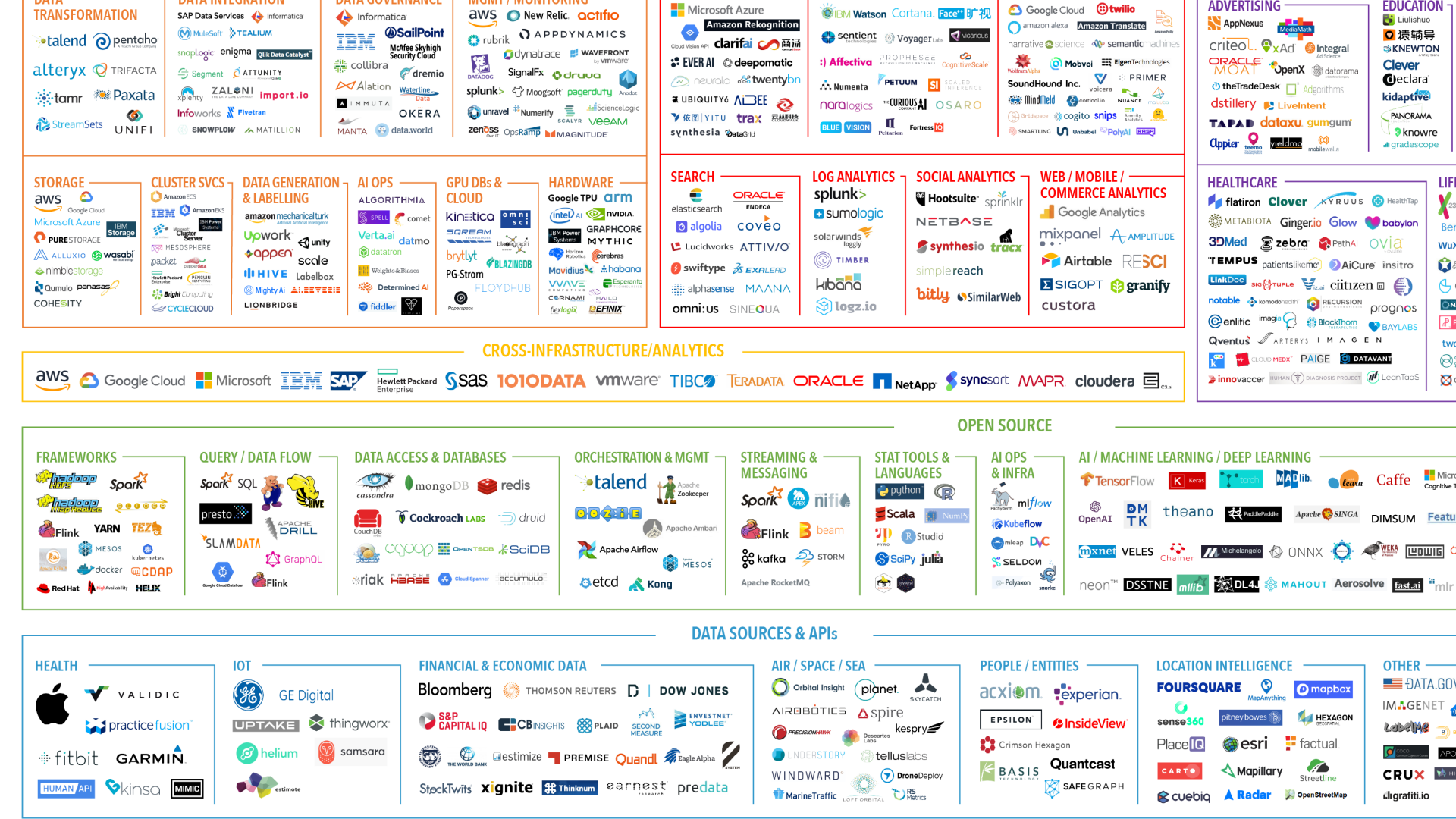


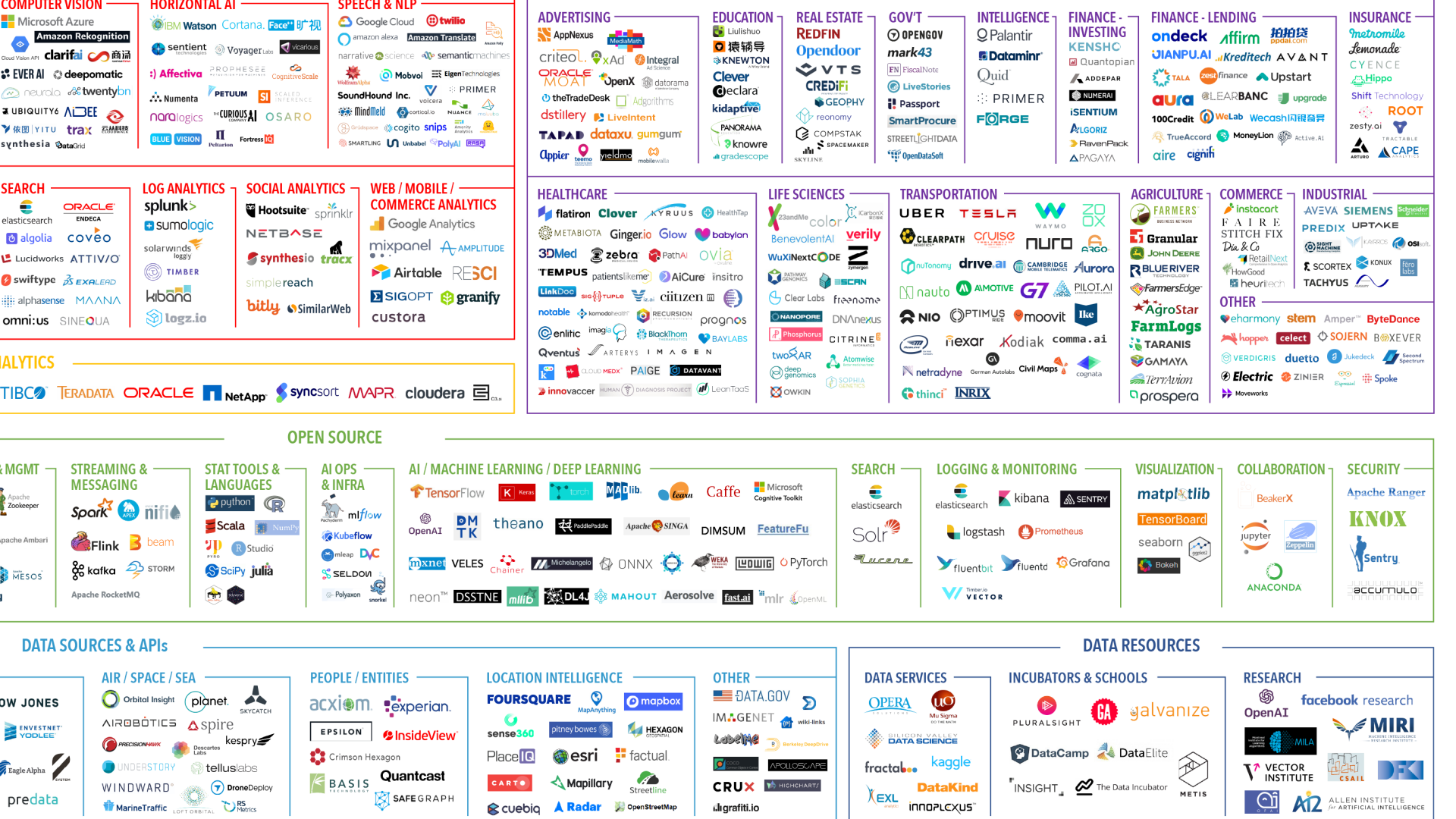
DATA & AI LANDSCAPE 2019



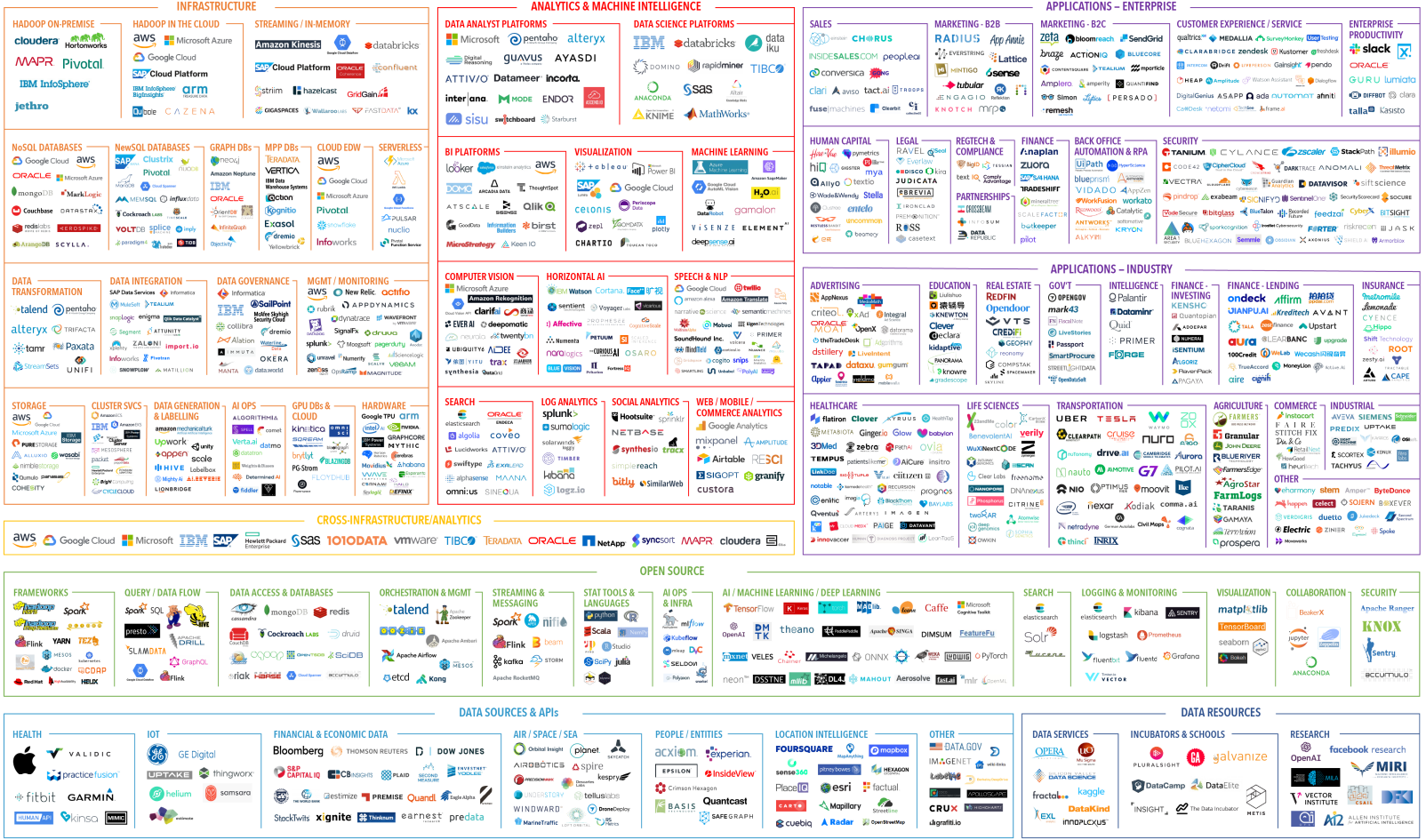
© Matt Turck (@mattturck), Lisa Xu (@lisaxu92), & FirstMark (@firstmarkcap)


INFRASTRUCTURE
HADOOP ON-PREMISE
cloudera
Hortonworks
MAPR
Pivotal
IBM InfoSphere
jethro
HADOOP IN THE CLOUD
aws
Microsoft Azure
Google Cloud
SAP Cloud Platform
IBM InfoSphere BigInsights
arm TREASURE DATA
Qubole
CAZENA
STREAMING / IN-MEMORY
Amazon Kinesis
Google Cloud Dataflow
databricks
SAP Cloud Platform
ORACLE Coherence
confluent
striim
hazelcast
GridGain
GIGASPACES
WallarooLABS
FASTDATA°
kx
NoSQL DATABASES
Google Cloud
aws
ORACLE
Microsoft Azure
mongoDB
MarkLogic
Couchbase
DATASTAX
redislabs HOME OF REDIS
AEROSPIKE
ArangoDB
SCYLLA.
NewSQL DATABASES
SAP HANA
Clustrix
NUODB
Pivotal
MariaDB
Cloud Spanner
MEMSQL
influxdata
Cockroach LABS
TIMESCALE
VOLTDB
splice MACHINE
imply
paradigm4
Trafodion
TIDB
GRAPH DBs
neo4j
Amazon Neptune
IBM
ORACLE
OrientDB
InfiniteGraph
Objectivity
MPP DBs
TERADATA
VERTICA
IBM Data Warehouse Systems
Actian
Kognitio
Exasol
dremio
Yellowbrick
CLOUD EDW
aws
Google Cloud
Microsoft Azure
Pivotal
snowflake
Infoworks
SERVERLESS
Microsoft Azure
AWS Lambda
Google Cloud Functions
PULSAR
nuclio
Pivotal Function Service
DATA TRANSFORMATION
talend
pentaho
alteryx
TRIFACTA
tamr
Paxata
StreamSets
UNIFI
DATA INTEGRATION
SAP Data Services
Informatica
MuleSoft
TEALIUM
snapLogic
enigma
Qlik Data Catalyst
Segment
ATTUNITY
xplenty
ZALONI
import.io
Infoworks
Fivetran
SNOWPLOW
MATILLION
DATA GOVERNANCE
Informatica
IBM
SailPoint
McAfee Skyhigh Security Cloud
collibra
dremio
Alation
Waterline Data
IMMUTA
OKERA
MANTA
data.world
MGMT / MONITORING
aws
New Relic.
actifio
rubrik
APPDYNAMICS
dynatrace
WAVEFRONT by vmware
DATADOG
SignalFx
druva
splunk>
Moogsoft
pagerduty
Anodot
unravel
Numerify
SCALYR
ScienceLogic
zenoss
OpsRamp
MAGNITUDE
VEEAM
ANALYTICS & MACHINE INTELLIGENCE
DATA ANALYST PLATFORMS
Microsoft
pentaho
alteryx
Digital Reasoning
GUAVUS
AYASDI
ATTIVIO
Datameer
incorta.
interana
MODE
ENDOR
ASCEND.IO
sisu
switchboard
Starburst
DATA SCIENCE PLATFORM
IBM
databricks
DOMINO
rapidm
ANACONDA
SAS
KNIME
MathW
BI PLATFORMS
looker
salesforce einstein analytics
aws
DOMO
ARCADIA DATA
ThoughtSpot
ATSCALE
SISENSE
Qlik
GoodData
Information Builders
birst
MicroStrategy
Keen IO
VISUALIZATION
+ableau
Power BI
SAP Lumira
Google Cloud
celonis
Periscope Data
zepl
GOOMDATA
plotly
CHARTIO
TOUCAN TOCO
MACHINE
Azure Machine
Google Cloud AutoML
DataRobot
ViSENZ
deepsense
COMPUTER VISION
Microsoft Azure
Amazon Rekognition
Cloud Vision API
clarifai
商汤
EVER AI
deepomatic
neurala
twenty bn
UBIQUITY6
AIDEE
依图 | YITU
trax
synthesia
DataGrid
HORIZONTAL AI
IBM Watson
Cortana.
Face++
旷视
sentient technologies
Voyager Labs
vicarious
:) Affectiva
PROPHESEE
CognitiveScale
Numenta
PETUUM
SCALED INFERENCE
narrative science
naralogics
THE CURIOUS AI COMPANY
OSARO
BLUE VISION
Peltarion
Fortress IQ
SPEECH & NLP
Google Cloud
amazon alexa
narrative science
WolframAlpha
Mobvo
SoundHound Inc.
MindMeld
Gridspace
cogito
SMARTLING
Unbab

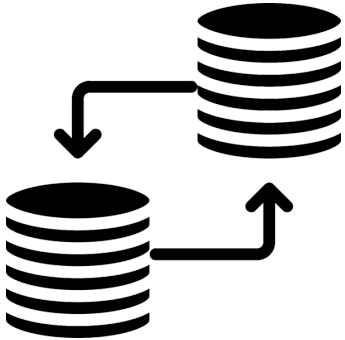

DATA TRANSFORMATION
talend
pentaho
alteryx
TRIFACTA
tamr
Paxata
StreamSets
UNIFI
DATA INTEGRATION
SAP Data Services
Informatica
MuleSoft
TEALIUM
snapLogic
enigma
Qlik Data Catalyst
Segment
ATTUNITY
xplenty
ZALONI
import.io
Infoworks
Fivetran
SNOWPLOW
MATILLION
DATA GOVERNANCE
Informatica
IBM
SailPoint
McAfee Skyhigh Security Cloud
collibra
dremio
Alation
Waterline Data
IMMUTA
OKERA
MANTA
data.world
MGMT / MONITORING
aws
New Relic
actifio
rubrik
APPDYNAMICS
dynatrace
WAVEFRONT by vmware
DATADOG
SignalFx
druva
splunk>
Moogsoft
pagerduty
unravel
Numerify
SCALYR
ScienceLogic
zenoss
OpsRamp
MAGNITUDE
Microsoft Azure
Amazon Rekognition
Cloud Vision API
clarifai
商汤
EVER AI
deepomatic
neurala
twenty bn
UBIQUITY6
ADEE
依图 | YITU
trax
CLARIFAI
synthesia
DataGrid
IBM Watson
Cortana
Face++ 旷视
sentient technologies
Voyager Labs
vicarious
:) Affectiva
PROPHESEE
CognitiveScale
Numenta
PETUUM
SCALED INFERENCE
narrative science
semantic machines
narrologics
THE CURIOUS AI COMPANY
OSARO
BLUE VISION
Peltarion
Fortress IQ
Google Cloud
twilio
amazon alexa
Amazon Translate
Amazon Polly
narrative science
semantic machines
Wolfram|Alpha
Mobvoi
EigenTechnologies
SoundHound Inc.
voicera
PRIMER
MindMeld
cortical.io
NUANCE
mocha
Gridspace
cogito
snips
Amplify Analytics
SMARTLING
Unbabel
PolyAI
ADVERTISING
AppNexus
MediaMath
criteo
xAd
Integral Ad Science
ORACLE MOAT
OpenX
datarama
theTradeDesk
Adgorithms
dstillery
LiveIntent
TAPAD
dataxu
gumgum
Appier
teemo
yieldmo
mobilewalla
EDUCATION
Liulishuo
猿辅导
KNEWTON
Clever
eclara
kidaptive
PANORAMA
knowre
gradescope
STORAGE
aws
Google Cloud
Microsoft Azure
IBM Storage
PURESTORAGE
ALLUXIO
wasabi
nimblestorage
Qumulo
panasas
COHESITY
CLUSTER SVCS
AmazonECS
IBM
Amazon EKS
Cluster Server
IBM Power Systems
MESOSPHERE
packet
Hewlett Packard Enterprise
PENGUIN COMPUTING
Bright Computing
CYCLECLOUD
DATA GENERATION & LABELLING
amazon mechanical turk
Artificial Artificial Intelligence
Upwork
unity
appen
scale
HIVE
Labelbox
Mighty AI
LIONBRIDGE
AI OPS
ALGORITHMIA
SPELL
comet
Verta.ai
datmo
datatron
Weights & Biases
Determined AI
fiddler
GPU DBs & CLOUD
kinetica
omnisci
SQREAM
blazegraph
brytlyt
BLAZINGDB
PG-Strom
FLOYDHUB
Paperspace
HARDWARE
Google TPU
arm
intel AI
NVIDIA
IBM Power Systems
GRAPHCORE
MYTHIC
Horizon Robotics
cerebras
Movidius
habana
WAVE COMPUTING
CORNAMI
Esperanto Technologies
HAILO
flexlogix
EFINIX
SEARCH
elasticsearch
ORACLE ENDECA
algolia
coveo
Lucidworks
ATTIVIO
swiftype
EXALEAD
alphasense
MAANA
omni:us
SINEQUA
LOG ANALYTICS
splunk>
sumologic
solarwinds loggly
TIMBER
kibana
logz.io
SOCIAL ANALYTICS
Hootsuite
sprinklr
NETBASE
synthesio
tracx
simplereach
bitly
SimilarWeb
WEB / MOBILE / COMMERCE ANALYTICS
Google Analytics
mixpanel
AMPLITUDE
Airtable
RESCI
SIGOPT
granify
custora
HEALTHCARE
flatiron
Clover
KYRUUS
HealthTap
METABIOTA
Ginger.io
Glow
babylon
3DMed
zebra
PathAI
ovia
TEMPUS
patientslikeme
AiCure
insitro
LinkDoc
SIG TUPLE
Viz.ai
citizen
notable
komodoHealth
RECURSION
prognos
enlitic
imagia
BlackThorn
BAYLABS
Qventus
ARTERYS
IMAGEN
CLOUD MEDX
PAIGE
DATAVANT
innoviccer
DIAGNOSIS PROJECT
LeanTaaS
CROSS-INFRASTRUCTURE/ANALYTICS
aws
Google Cloud
Microsoft
IBM
SAP
Hewlett Packard Enterprise
sas
1010DATA
vmware
TIBCO
TERADATA
ORACLE
NetApp
syncsort
MAPR
cloudera
OPEN SOURCE
FRAMEWORKS
hadoop
Spark
Flink
YARN
TEZ
MESOS
kubernetes
docker
CDAP
Red Hat
HELIX
QUERY / DATA FLOW
Spark
SQL
HIVE
presto
APACHE DRILL
SLAMDATA
GraphQL
Flink
DATA ACCESS & DATABASES
cassandra
mongoDB
redis
CouchDB
Cockroach LABS
druid
OpenTSDB
SciDB
riak
HBASE
Cloud Spanner
ACCUMULO
ORCHESTRATION & MGMT
talend
Apache Zookeeper
OOZIE
Apache Ambari
Apache Airflow
MESOS
etcd
Kong
STREAMING & MESSAGING
Spark
APEX
nifi
Flink
beam
kafka
STORM
Apache RocketMQ
STAT TOOLS & LANGUAGES
python
R
Scala
NumPy
PYRO
Studio
SciPy
julia
AI OPS & INFRA
Pachyderm
mlflow
Kubeflow
mleap
DVC
SELDON
Polyaxon
snorkel
AI / MACHINE LEARNING / DEEP LEARNING
TensorFlow
Keras
torch
MAGMA
mlib
learn
Caffe
Microsoft Cognitive Toolkit
OpenAI
DMTK
theano
PaddlePaddle
Apache SINGA
DIMSUM
Featu
mxnet
VELES
Chainer
Michelangelo
ONNX
WEKA
LOWIG
neon™
DSSTNE
mlib
DL4J
MAHOUT
Aerosolve
fast.ai
mlr
DATA SOURCES & APIs
HEALTH
VALIDIC
practicefusion
fitbit
GARMIN
HUMAN API
kinsa
MIMIC
IOT
GE Digital
UPTAKE
thingworx
helium
samsara
estimote
FINANCIAL & ECONOMIC DATA
Bloomberg
THOMSON REUTERS
DOW JONES
S&P CAPITAL IQ
CB INSIGHTS
PLAID
SECOND MEASURE
ENVESTNET YODLEE
THE WORLD BANK
estimize
PREMISE
Quandl
Eagle Alpha
StockTwits
xignite
Thinknum
earnest
predata
AIR / SPACE / SEA
Orbital Insight
planet.
SKYCATCH
AIROBOTICS
spire
kespry
PRECISIONHAWK
Descartes Labs
UNDERSTORY
tellusLabs
WINDWARD
DroneDeploy
MarineTraffic
LOFT ORBITAL
RS Metrics
PEOPLE / ENTITIES
acxiom
experian
EPSILON
InsideView
Crimson Hexagon
BASIS TECHNOLOGY
Quantcast
SAFEGRAPH
LOCATION INTELLIGENCE
FOURSQUARE
MapAnything
mapbox
sense360
pitney bowes
HEXAGON GEOSPATIAL
PlaceIQ
esri
factual
CARTO
Mapillary
Streetline
cuebiq
Radar
OpenStreetMap
OTHER
DATA.GOV
IMAGENET
CRUX
grafiti.io


COMPUTER VISION
HORIZONTAL AI
SPEECH & NLP
ADVERTISING
EDUCATION
REAL ESTATE
GOV'T
INTELLIGENCE
FINANCE - INVESTING
FINANCE - LENDING
INSURANCE
SEARCH
LOG ANALYTICS
SOCIAL ANALYTICS
WEB / MOBILE / COMMERCE ANALYTICS
HEALTHCARE
LIFE SCIENCES
TRANSPORTATION
AGRICULTURE
COMMERCE
INDUSTRIAL
OTHER
ANALYTICS
OPEN SOURCE
& MGMT
STREAMING & MESSAGING
STAT TOOLS & LANGUAGES
AI OPS & INFRA
AI / MACHINE LEARNING / DEEP LEARNING
SEARCH
LOGGING & MONITORING
VISUALIZATION
COLLABORATION
SECURITY
DATA SOURCES & APIs
AIR / SPACE / SEA
PEOPLE / ENTITIES
LOCATION INTELLIGENCE
OTHER
DATA RESOURCES
DATA SERVICES
INCUBATORS & SCHOOLS
RESEARCH

DATA & AI LANDSCAPE 2019

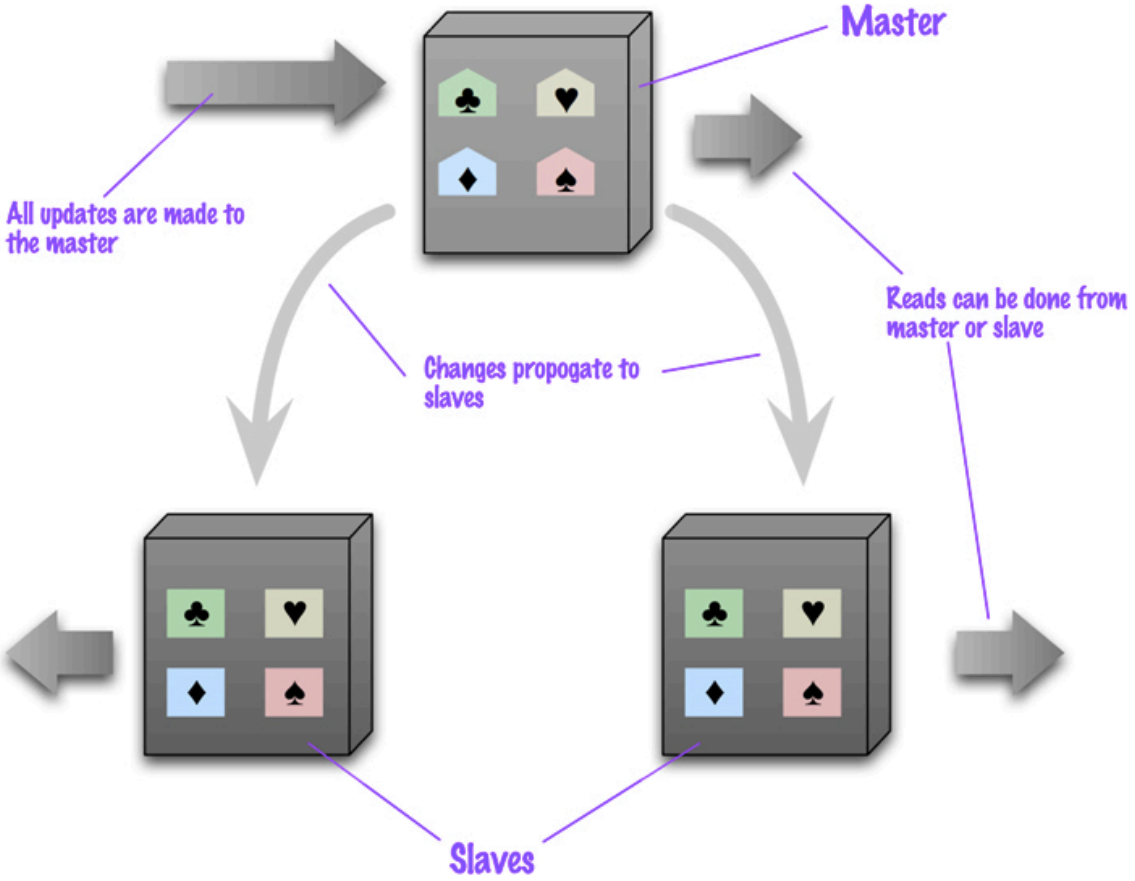


© Matt Turck (@mattturck), Lisa Xu (@lisaxu92), & FirstMark (@firstmarkcap)



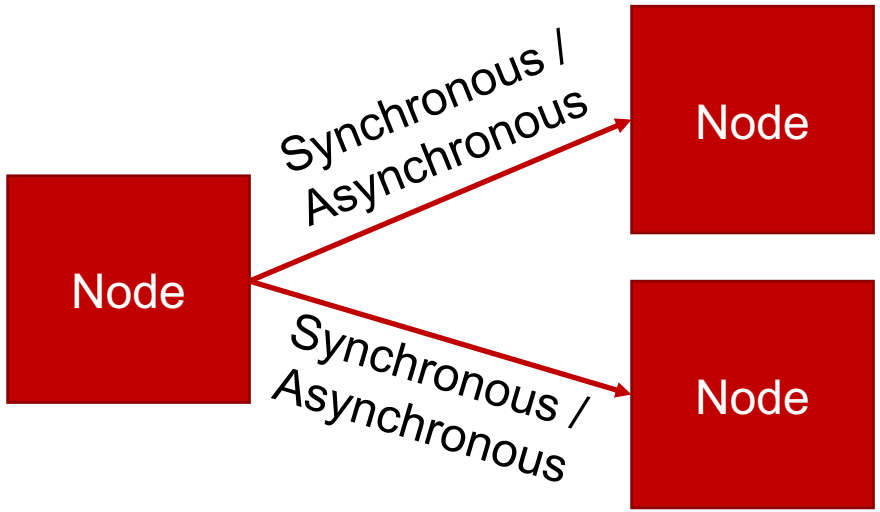
Replication

Replication

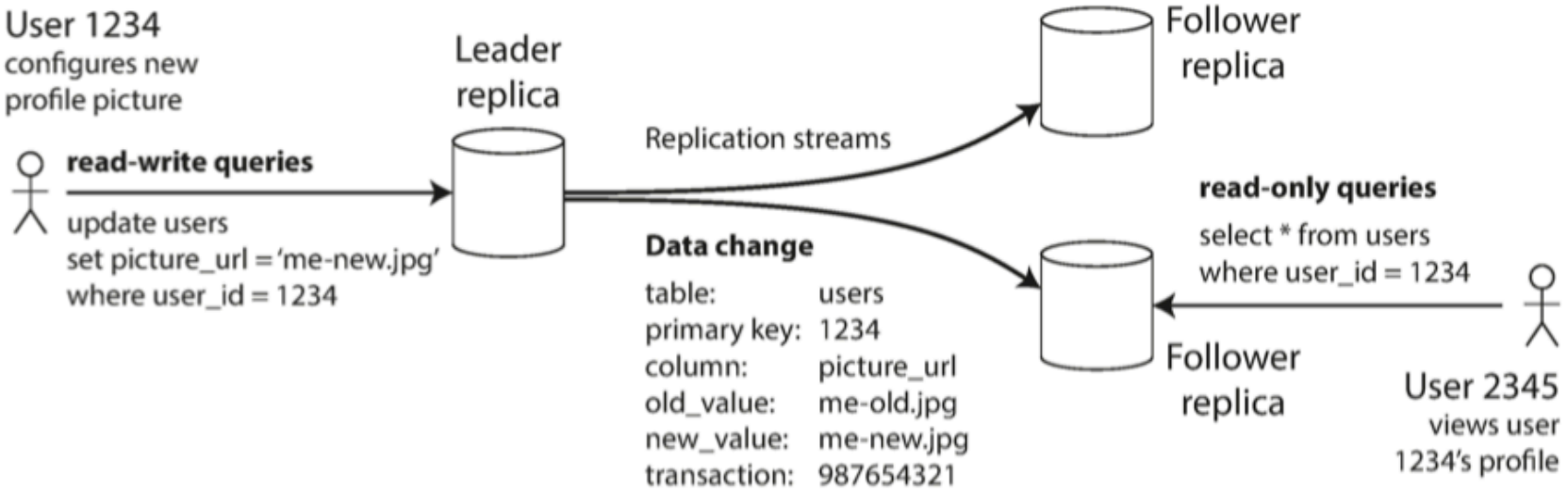


Replication

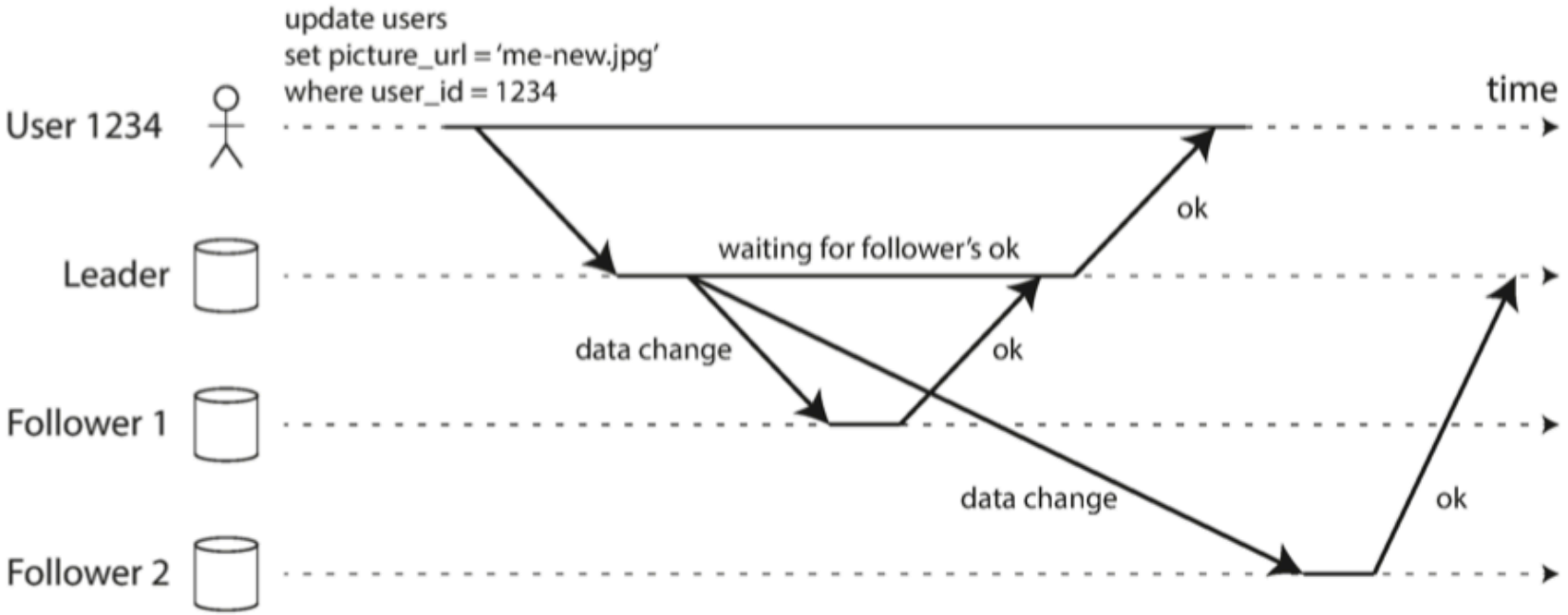
- Stores N copies of each data item
- Consistency model: synchronous vs. asynchronous
- Coordination: Single-leader, Multi-Leader, Leaderless



Leader-based (Master–Slave) Replication



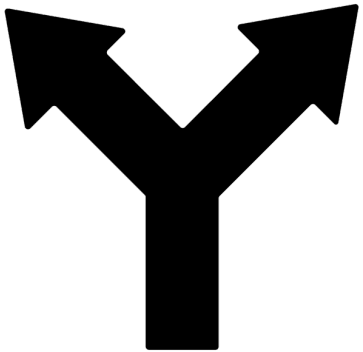
Leader-based replication with one synchronous and one asynchronous follower



Three main approaches to replication

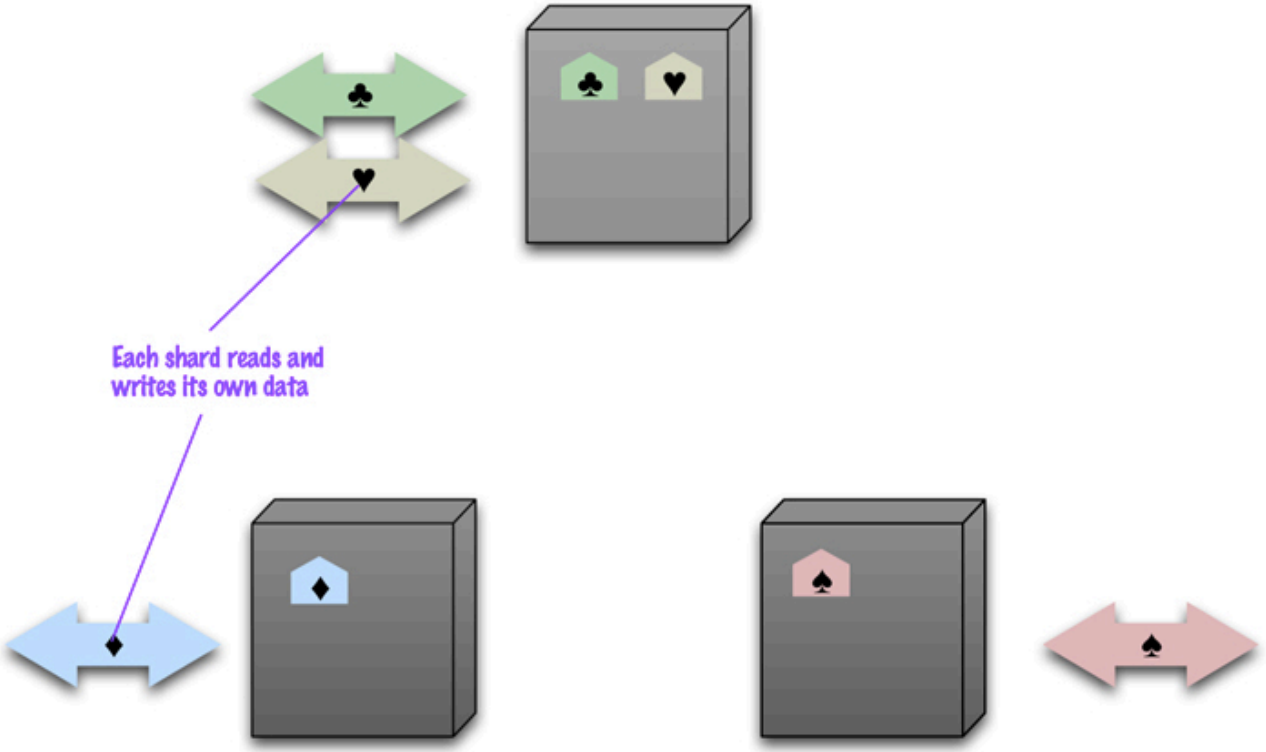
- **Single-leader replication**
  - Clients send all **writes** to a **single node** (the **leader**), which sends a stream of data change events to the other replicas (**followers**).
  - **Reads** can be performed on any replica, but **reads** from followers might be **stale**.
- **Multi-leader replication**
  - Clients send each **write** to one of **several leader nodes**, any of which can accept writes.
  - The leaders send streams of data change events to each other and to any follower nodes.
- **Leaderless replication**
  - Clients send each **write** to **several nodes**, and **read** from **several nodes** in parallel in order to detect and correct nodes with stale data.

Source: Kleppmann (2017), p. 192



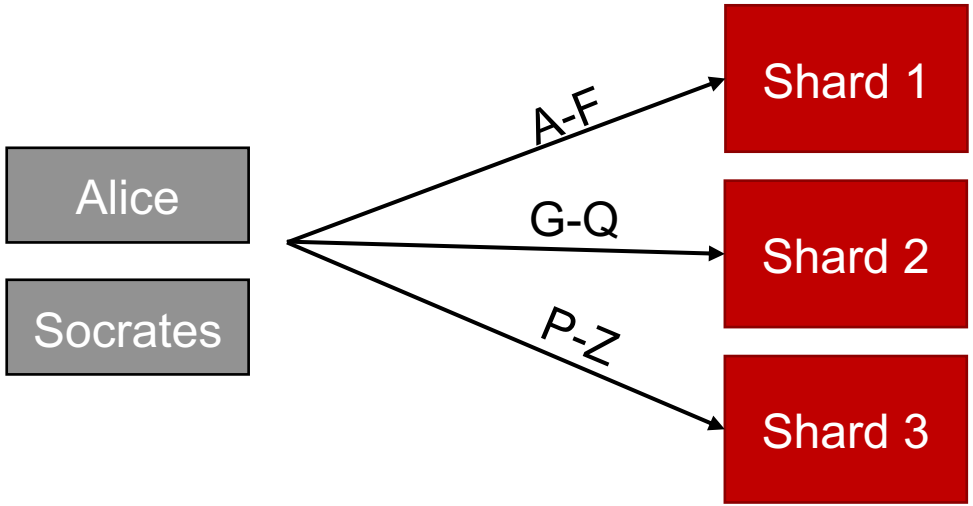
Partitioning

Partitioning (Sharding)



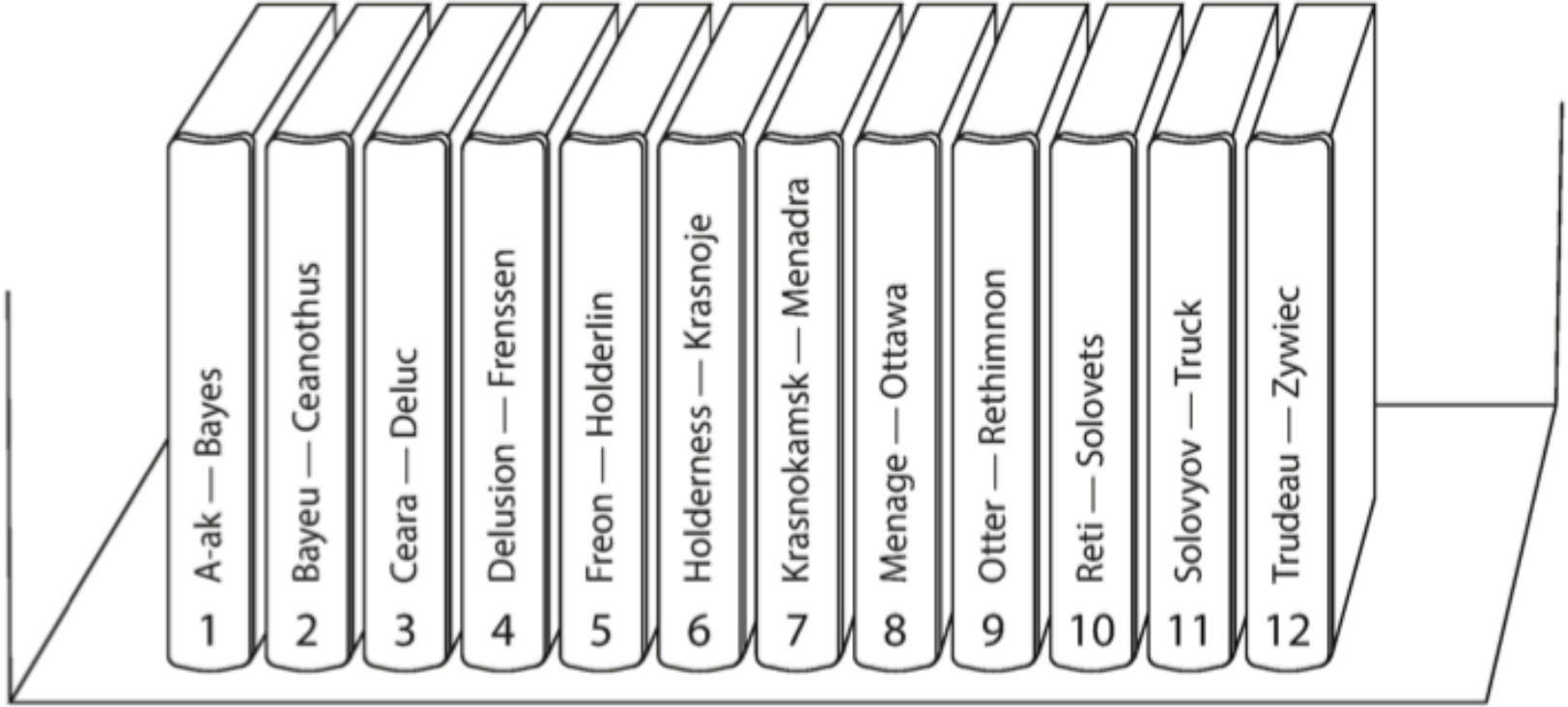
Partitioning (Sharding)

- Horizontal distribution of data over nodes
- Partitioning strategies: Hash-based vs. Range-based
- Difficulty: Multi-Shard-Operations (join, aggregation)



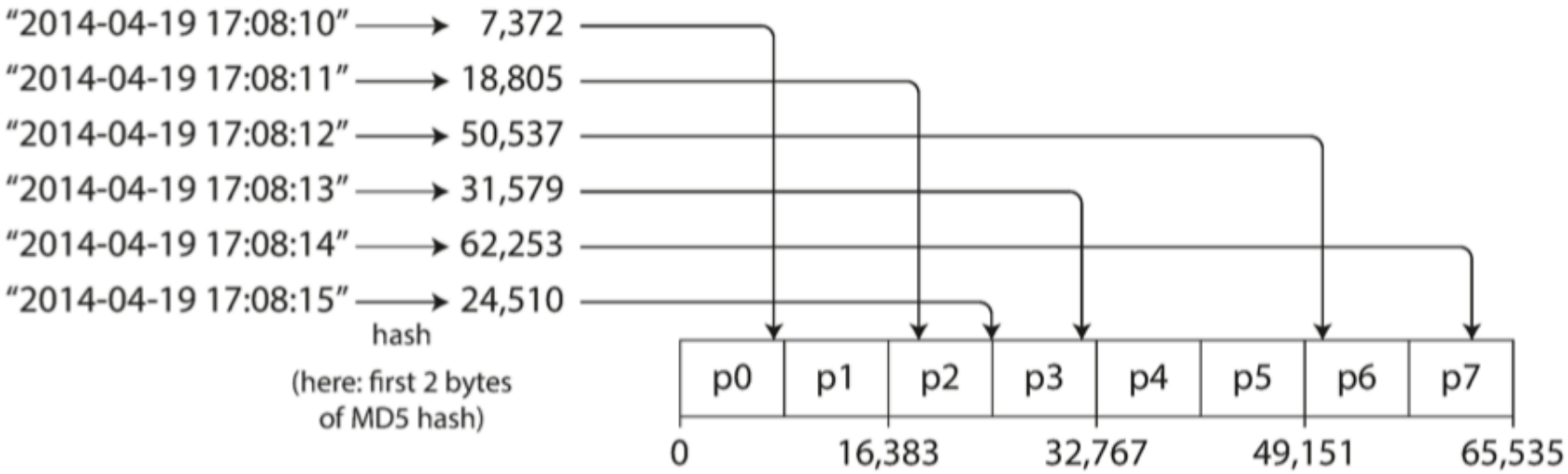
Source: Gessert (2016)

A print encyclopedia is partitioned by key range

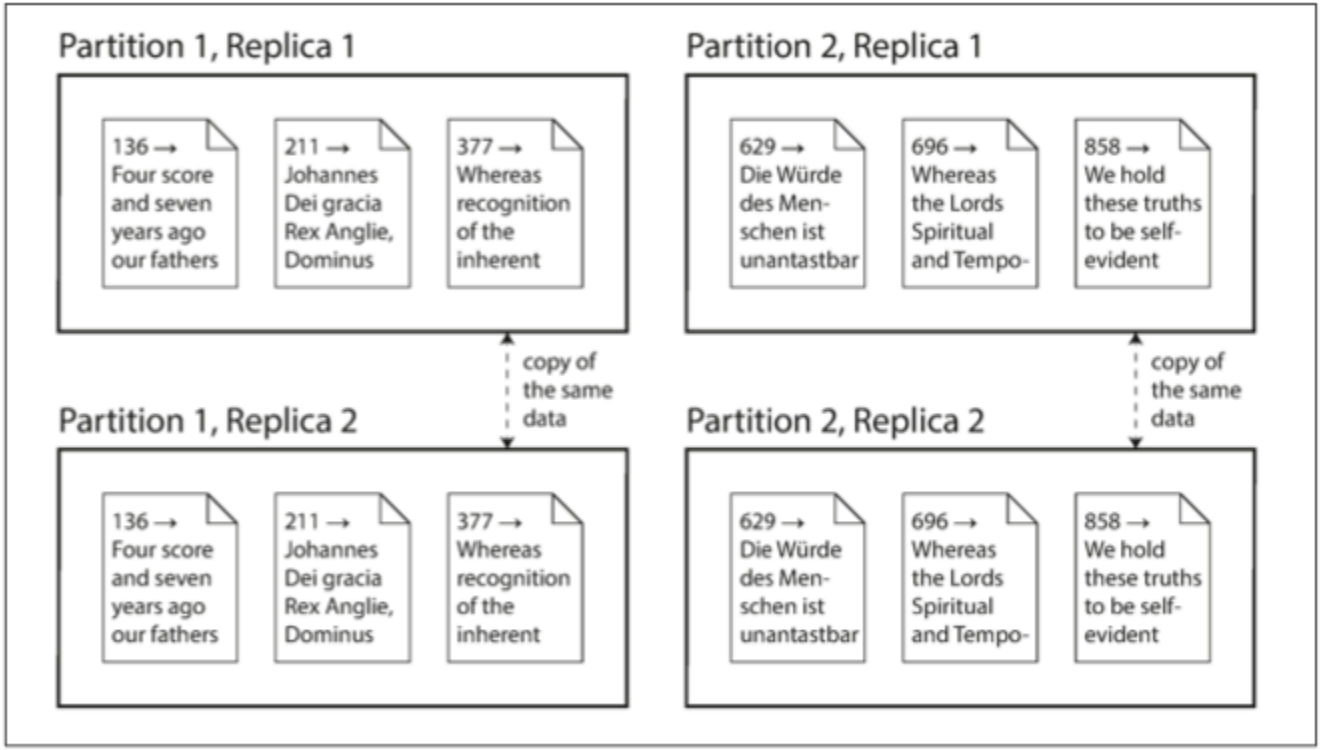


Source: Kleppmann (2017), p. 191

Partitioning by hash of key



Combining Replication and Partitioning



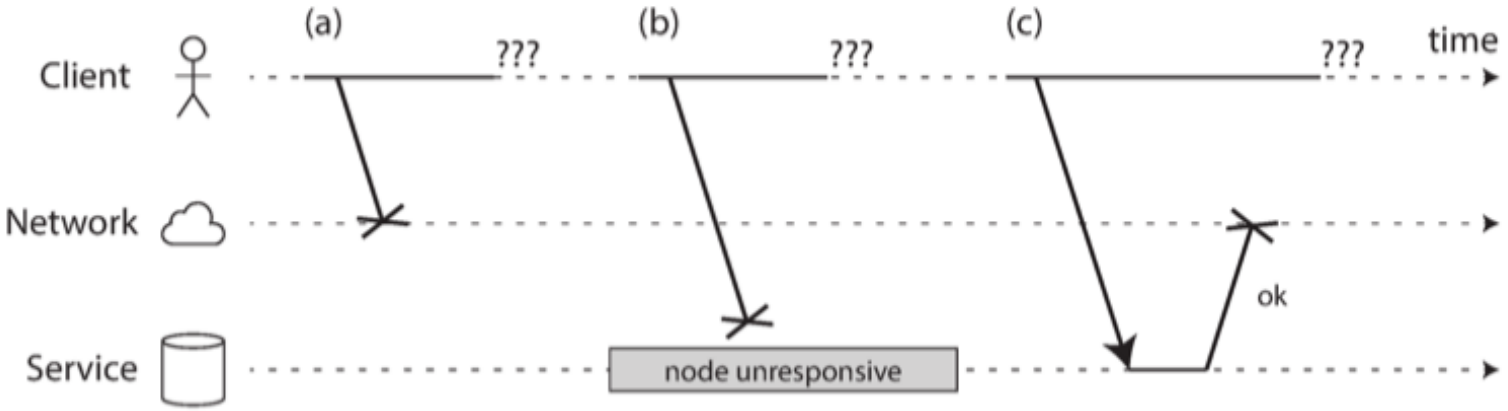
Impact of Replication and Partitioning on

- Reliability
  (Fault tolerance / high availability)
  - Keeping the system running, even when one machine (or several machines, or an entire datacenter) goes down
  - Disconnected operation
  - Allowing an application to continue working when there is a network interruption
- Scalability
  - Being able to handle a higher volume of reads than a single machine could handle, by performing reads on replicas or partitions
  - Storing more data than possible on a single machine by partitioning the data to more nodes
- Latency
  - Placing data geographically close to users, so that users can interact with it faster
- Maintainability
  - Difficult to operate
  - More complicated
  - Sometimes automatic failover possible
  - Problems of Distributed Consistency

Problems of Distributed Systems

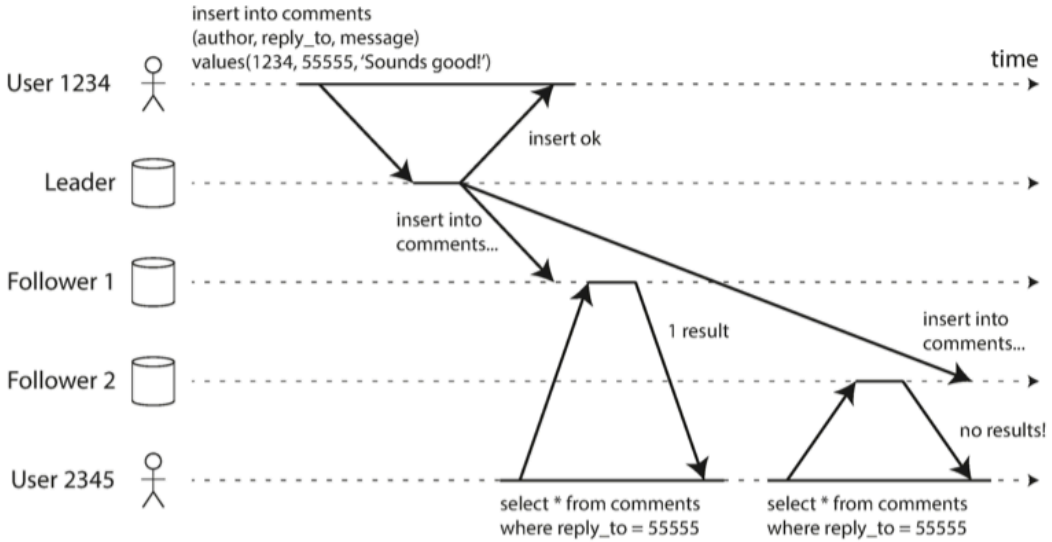
Problems of Distributed Systems

If you send a request and don't get a response, it's not possible to distinguish whether
(a) the request was lost,
(b) the remote node is down, or
(c) the response was lost.



Problems of Distributed Systems

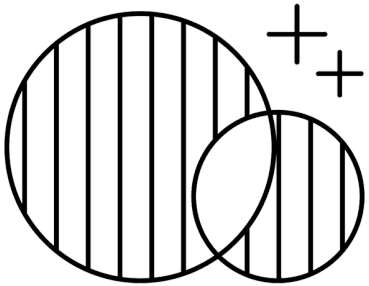
- A user first reads from a fresh replica, then from a stale replica.
- Time appears to go backward.
- To prevent this anomaly, we need monotonic reads.



Distributed Consistency Models

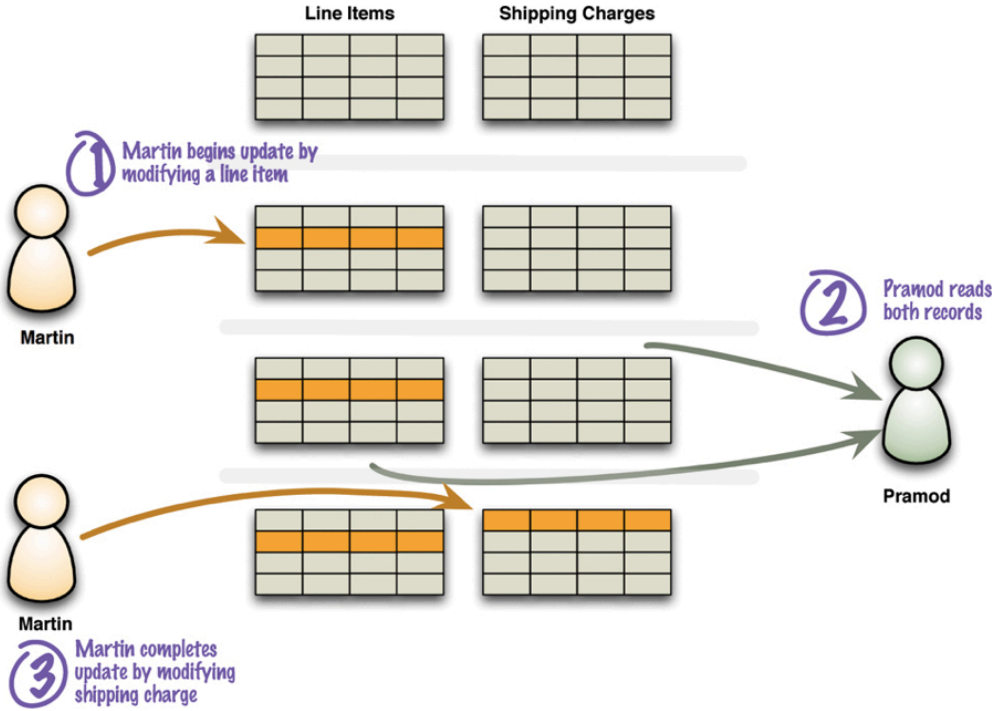
- **Read-after-write consistency**
  - Users should always see data that they submitted themselves.
- **Monotonic reads**
  - After users have seen the data at one point in time, they shouldn't later see the data from some earlier point in time.
- **Consistent prefix reads**
  - Users should see the data in a state that makes causal sense: for example, seeing a question and its reply in the correct order.
- **Linearizability**
  - Every user gets the same (consistent) view on the data independent with which node in a cluster he/she is connected

Source: Kleppmann (2017), p. 165



CAP Theorem

Consistency
(Also known as Linearizability)



Not the same consistency as in ACID!

Source: Fowler (2016)

Different Meaning of Consistency (Linearizability)

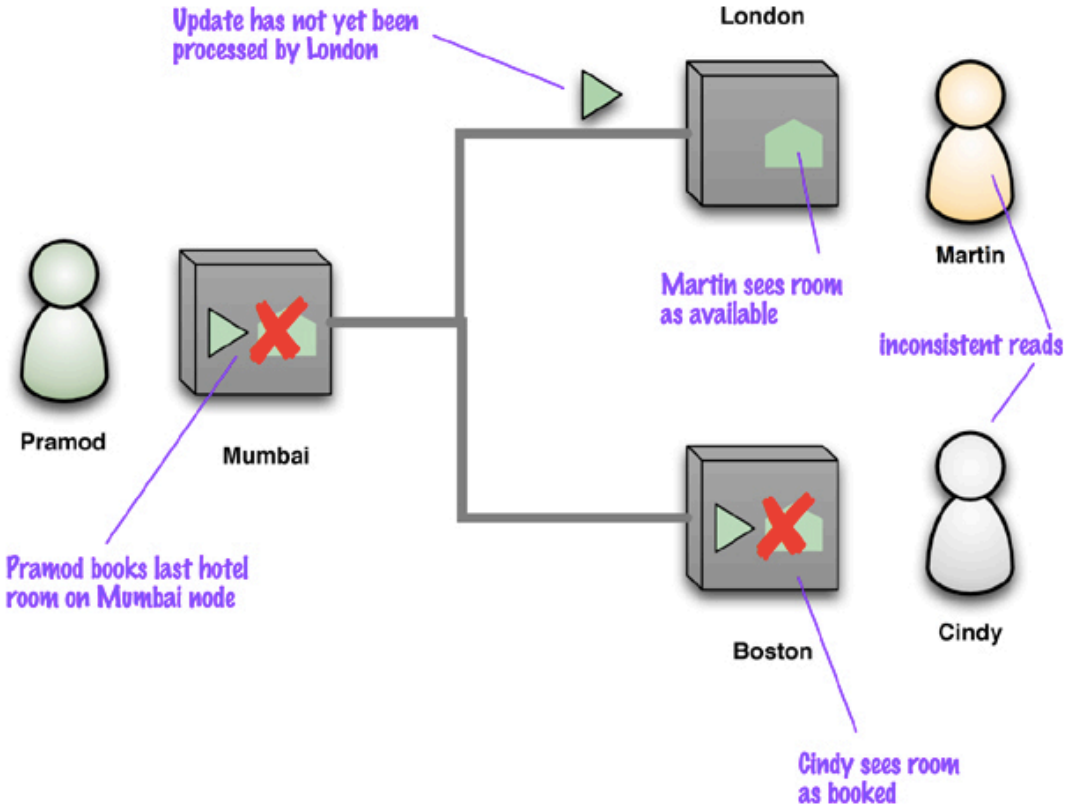
ACID

- Any transaction will bring the database from one valid state to another
- Valid according to all defined rules and constraints

CAP

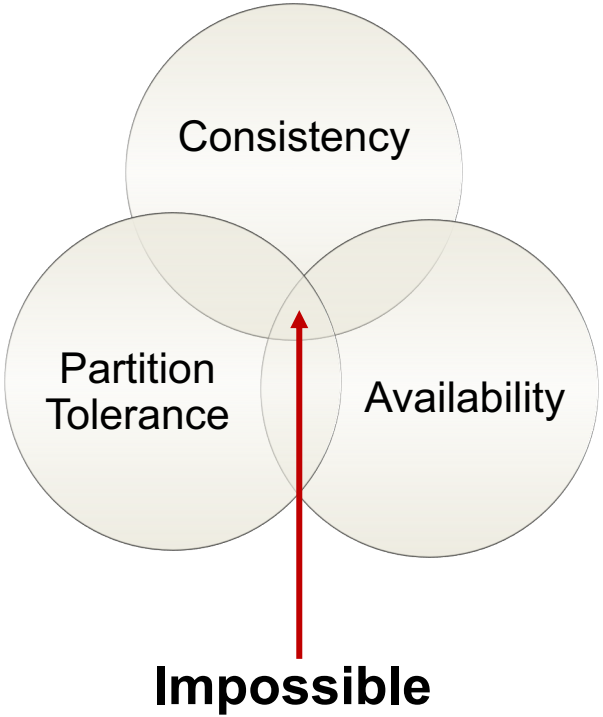
- Every user gets the same (consistent) view on the data independent with which node in a cluster he/she is connected

Inconsistency Problems because of Replication



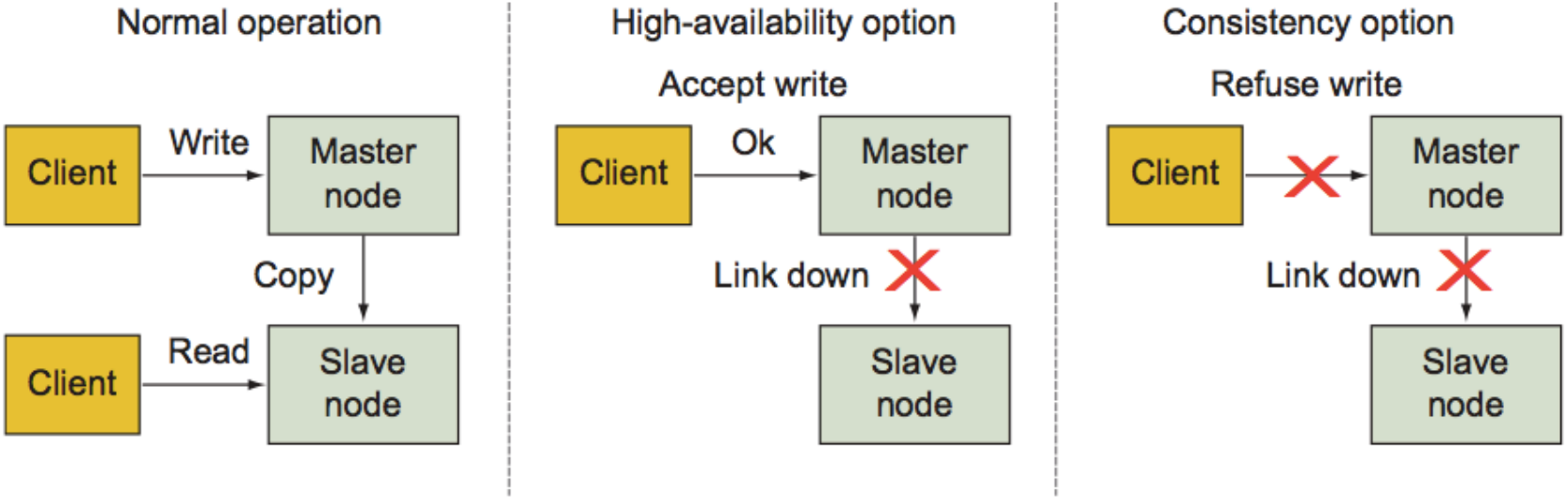
Source: Fowler (2016)

CAP-Theorem



- Only 2 out of 3 properties are achievable at a time:
  - **Consistency:** all clients have the same view on the data
  - **Availability**: every request to a non-failed node most result in correct response
  - **Partition tolerance**: the system has to continue working, even under arbitrary network partitions

CAP-Theorem


Normal operation
Client
Write
Master node
Copy
Client
Read
Slave node
High-availability option
Accept write
Client
Ok
Master node
Link down
Slave node
Consistency option
Refuse write
Client
Master node
Link down
Slave node

CAP-Theorem

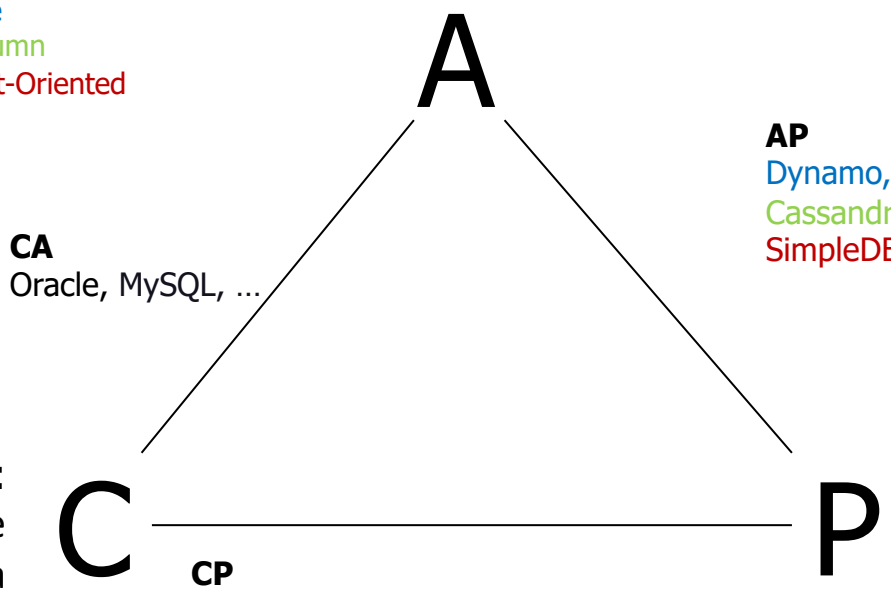
If you have…	then you get…
A single processor or many processors on a working network	Consistency AND availability
Many processors and network failures	Each transaction can select between consistency OR availability depending on the context

Source: McCreary and Kelly (2013)

CAP Triangle

Availability: Every client can always read and write

Data models
- Relational
- Key-Value
- Wide-Column
- Document-Oriented



AP
Dynamo, Redis, Riak, Voldemort
Cassandra
SimpleDB

CA
Oracle, MySQL, …

Consistency:
All clients share the same view on the data

CP
Postgres, MySQL Cluster, Oracle RAC
BigTable, HBase, Accumulo, Azure Tables
MongoDB, RethinkDB, DocumentsDB

Partition tolerance:
All nodes continue working under network partitions

Source: Gessert (2017)

Problematic and Unrealistic Definition of CAP Theorem

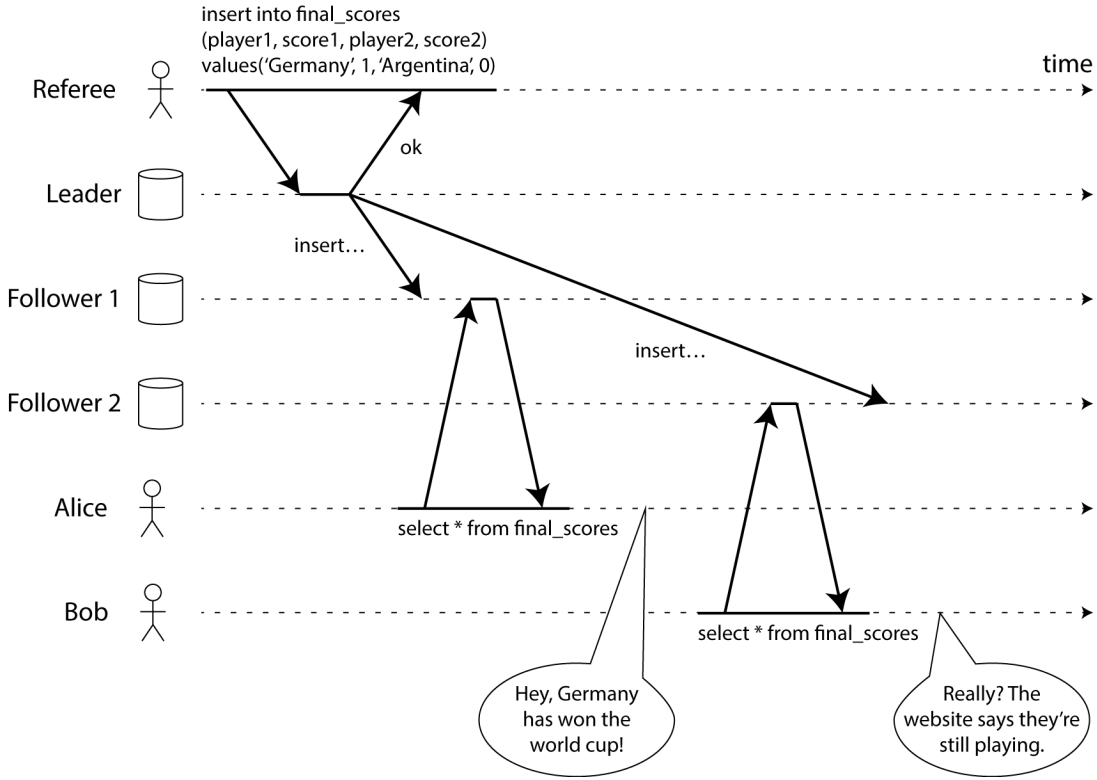
- Availability
  - Instance available. No chance of of waiting for self-repair or delay
- Network Partition
  - Could be arbitrary long!
- Consistency
  - Very strong type of consistency ➔ Linearizability

- What about Node failure?
- What about Latency?

Consistency (Linearizability)

- „If operation B started after operation A successfully completed,
- then operation B must see the system in the same state as it was on completion of operation A, or a newer state.”

Consistency (Linearizability)



PACELC – Alternative for CAP

Partition: Availability vs. Consistency, Else: Latency vs. Consistency

if there is a partition (P)

- how does the system trade off availability and consistency (A and C)

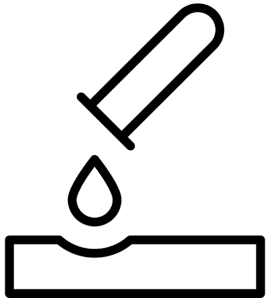
else (E)

- when the system is running normally in the absence of partitions, how does the system trade off latency (L) and consistency (C)

Classification of distributed storage systems:

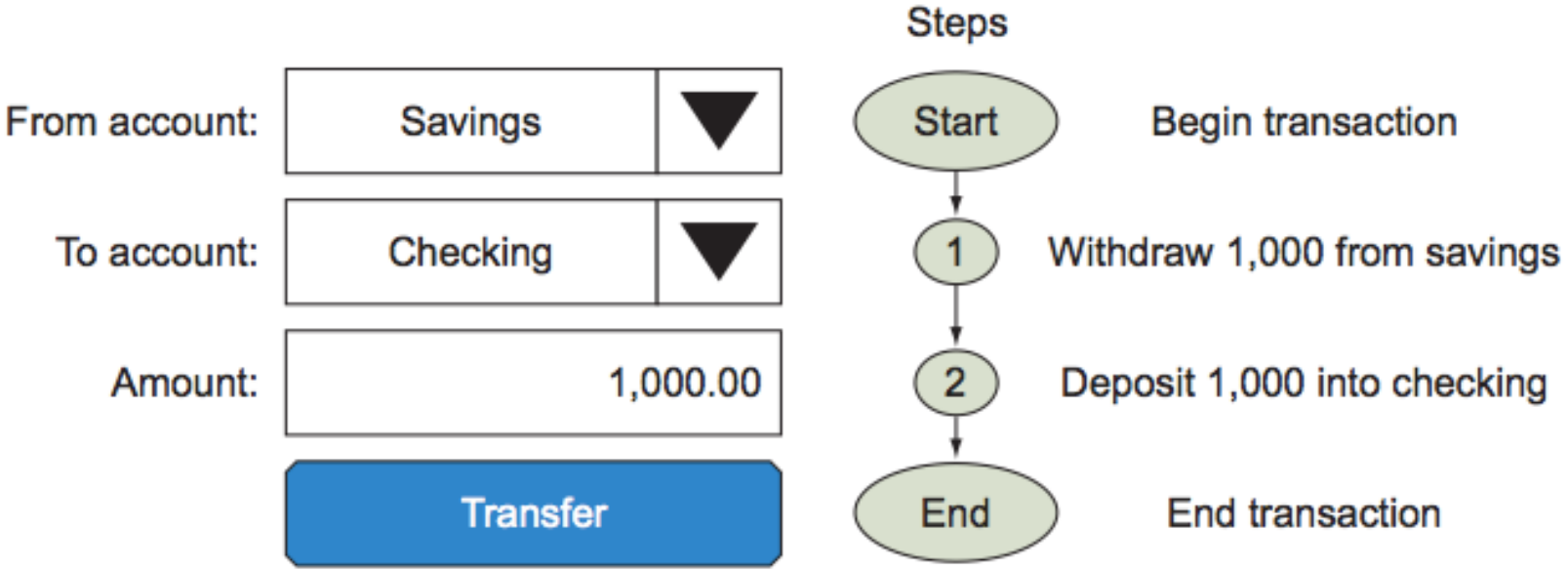
- PA/EL: Dynamo, Cassandra, Riak
- PC/EC: ACID Databases (like PostgreSQL), Big Table, Hbase
- PA/EC: MongoDB
- PC/EL: PNUTS

Source: Abadi (2012)



ACID vs. BASE

Transaction



ACID Properties of a Transaction

- Atomicity
  - each transaction is "all or nothing": if one part of the transaction fails, the entire transaction fails, and the database state is left unchanged.
- Consistency
  - any transaction will bring the database from one valid state to another
  - valid according to all defined rules and constraints
- Isolation
  - concurrent execution of transactions results in a system state that would be obtained if transactions were executed serially, i.e., one after the other
- Durability
  - once a transaction has been committed, it will remain so, even in the event of power loss, crashes, or errors

Source: Wikipedia https://en.wikipedia.org/wiki/ACID

BASE

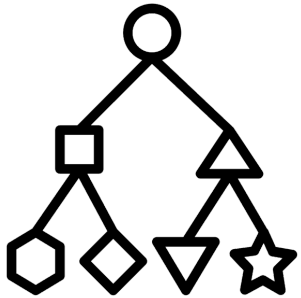
- Basic Availability
- Soft-state
- Eventual consistency

ACID vs. BASE

Vs.

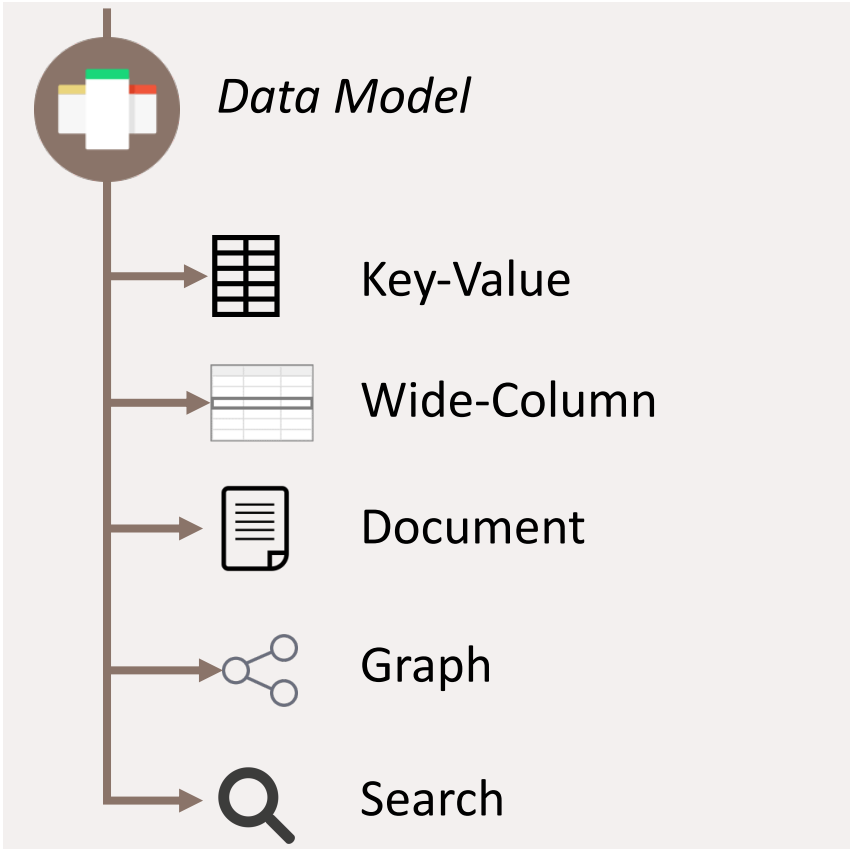
Acid	Base
• Get transaction details right • Block any reports while you are working • Be pessimistic: anything might go wrong! • Detailed testing and failure mode analysis • Lots of locks and unlocks	• Never block a write • Focus on throughput, not consistency • Be optimistic: if one service fails it will eventually get caught up • Some reports may be inconsistent for a while, but don't worry • Keep things simple and avoid locks

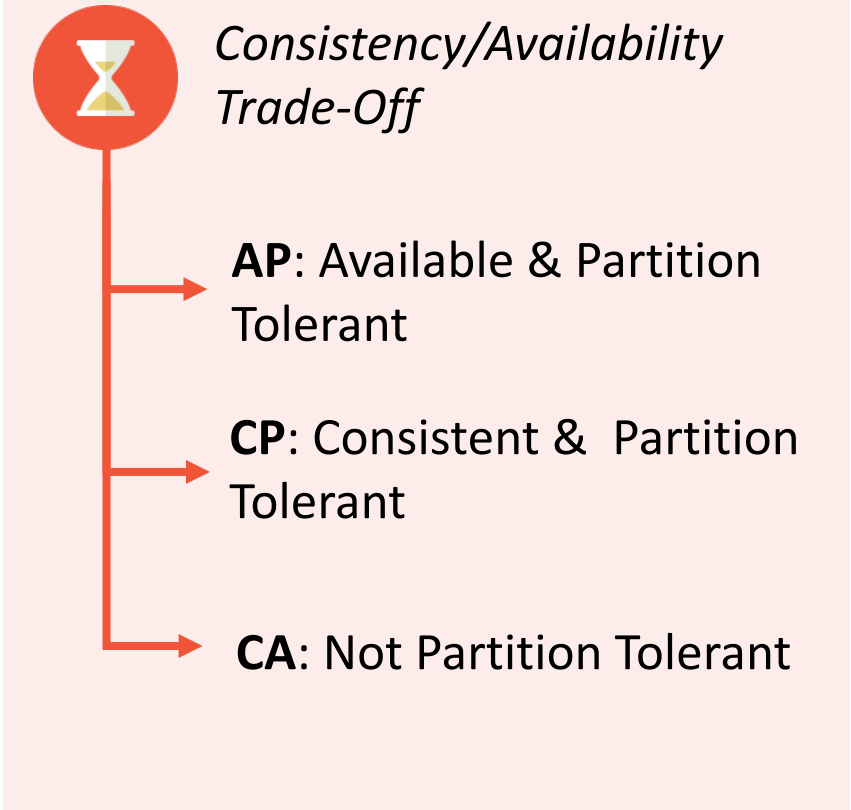
Source: McCreary and Kelly (2013)



NoSQL Classification

NoSQL System Classification

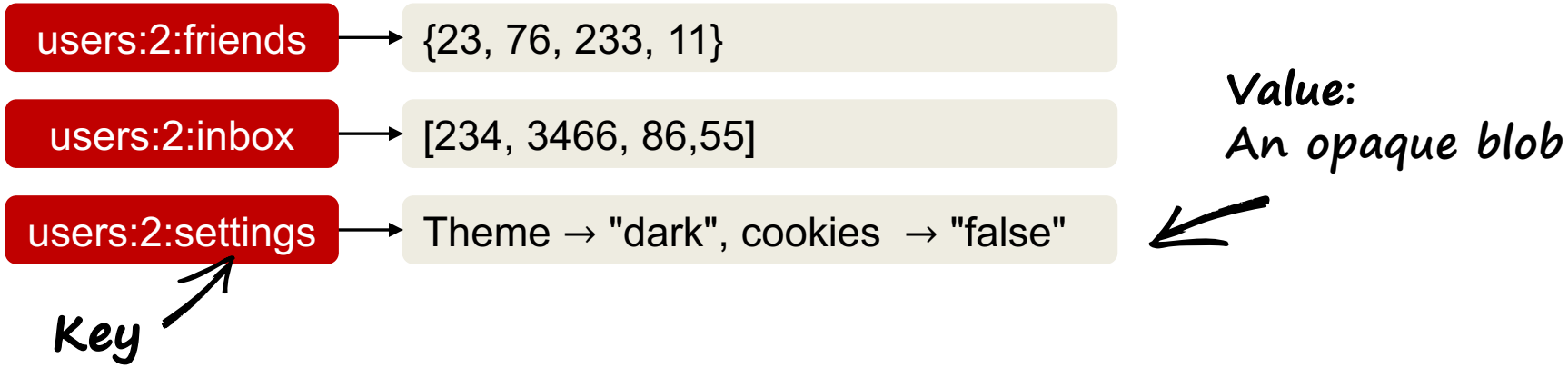




Source: Gessert (2017)

Key-Value Stores

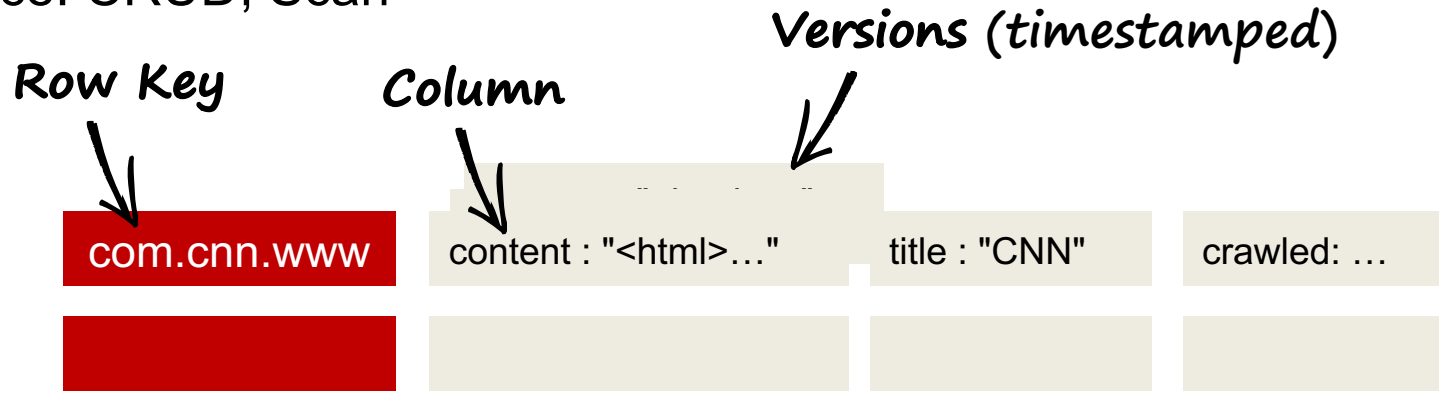
- Data model: (key) -> value
- Interface: CRUD (Create, Read, Update, Delete)



- Examples: Redis (CP), Amazon Dynamo (AP), Riak (AP)

Wide-Column Stores

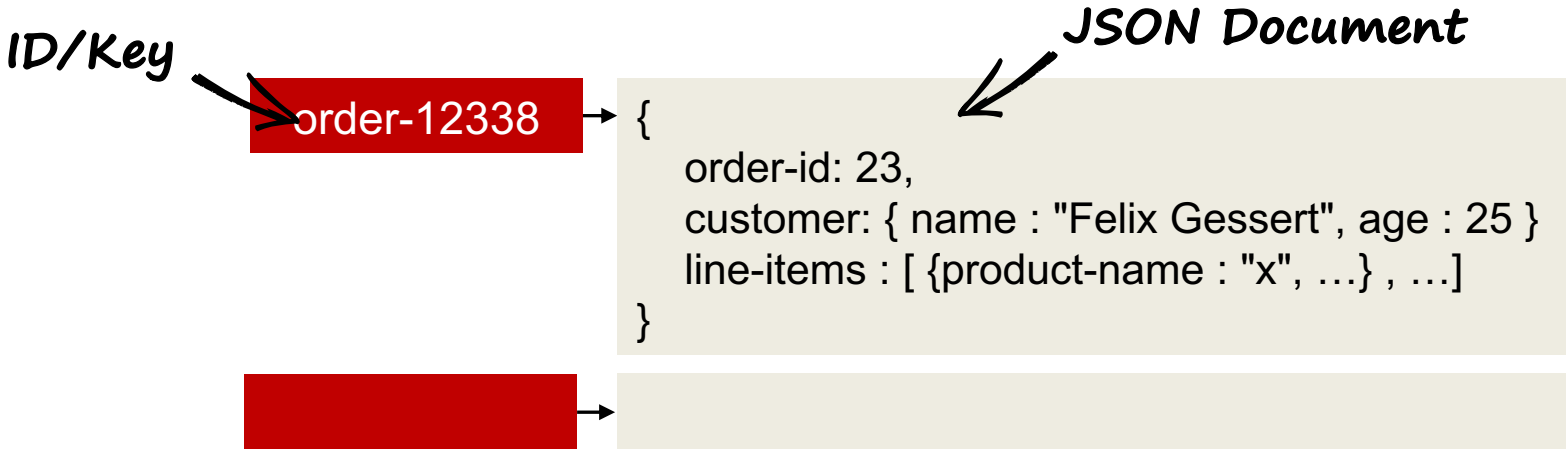
- Data model: (rowkey, column, timestamp) -> value
- Interface: CRUD, Scan



- Examples: Cassandra (AP), Google BigTable (CP), HBase (CP)
- Two-dimensional key-value store
- NOT equal to column-oriented storage like in SAP HANA!

Document Stores

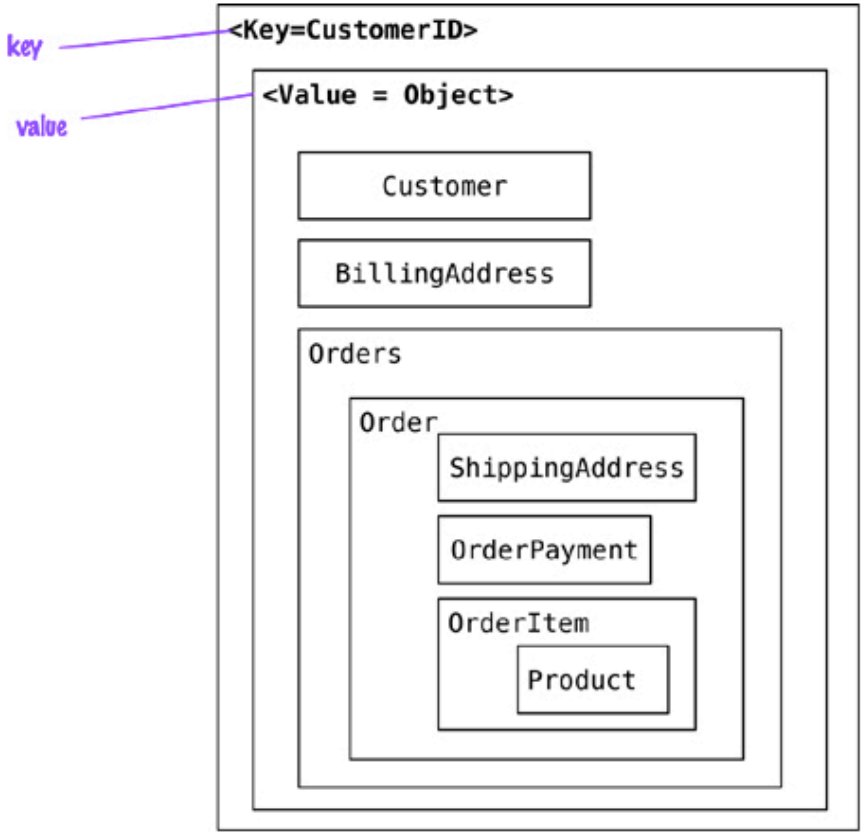
- Data model: (collection, key) -> document
- Interface: CRUD, Querys, Map-Reduce



- Examples: MongoDB (CP), CouchDB (AP), RethinkDB (CP)

Source: Gessert (2017)

Document Stores



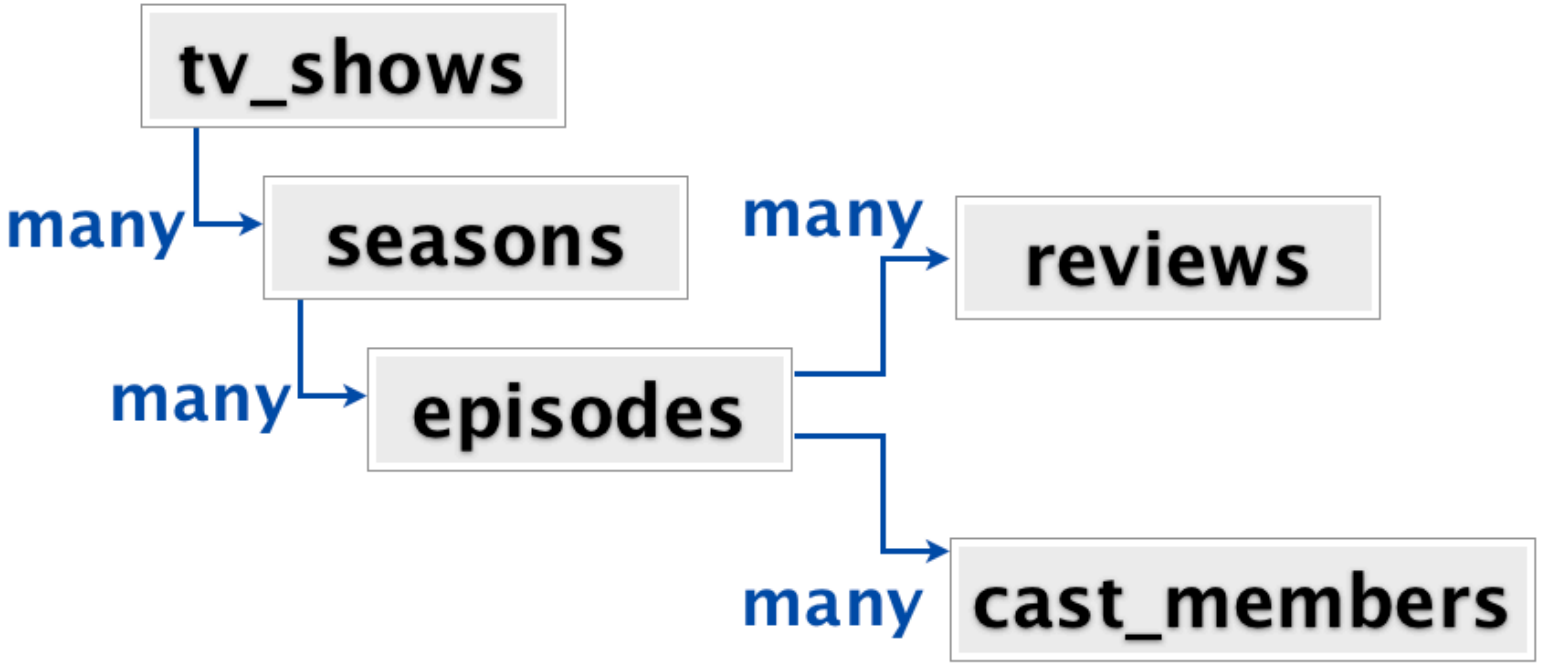
JSON

```
# Customer object
{
"customerId": 1,
"customer": {
  "name": "Martin",
  "billingAddress": [{"city": "Chicago"}],
  "payment": [{"type": "debit","ccinfo": "1000-1000-1000-1000"}],
  "orders":[{"orderId":99}]
 }
}

# Order object
{
"customerId": 1,
"orderId": 99,
"order":{
  "orderDate":"Nov-20-2011",
  "orderItems":[{"productId":27, "price": 32.45}],
   "orderPayment":[{"ccinfo":"1000-1000-1000-1000",
           "txnId":"abelif879rft"}],
   "shippingAddress":{"city":"Chicago"}
   }
}
```

Source: Fowler (2016)

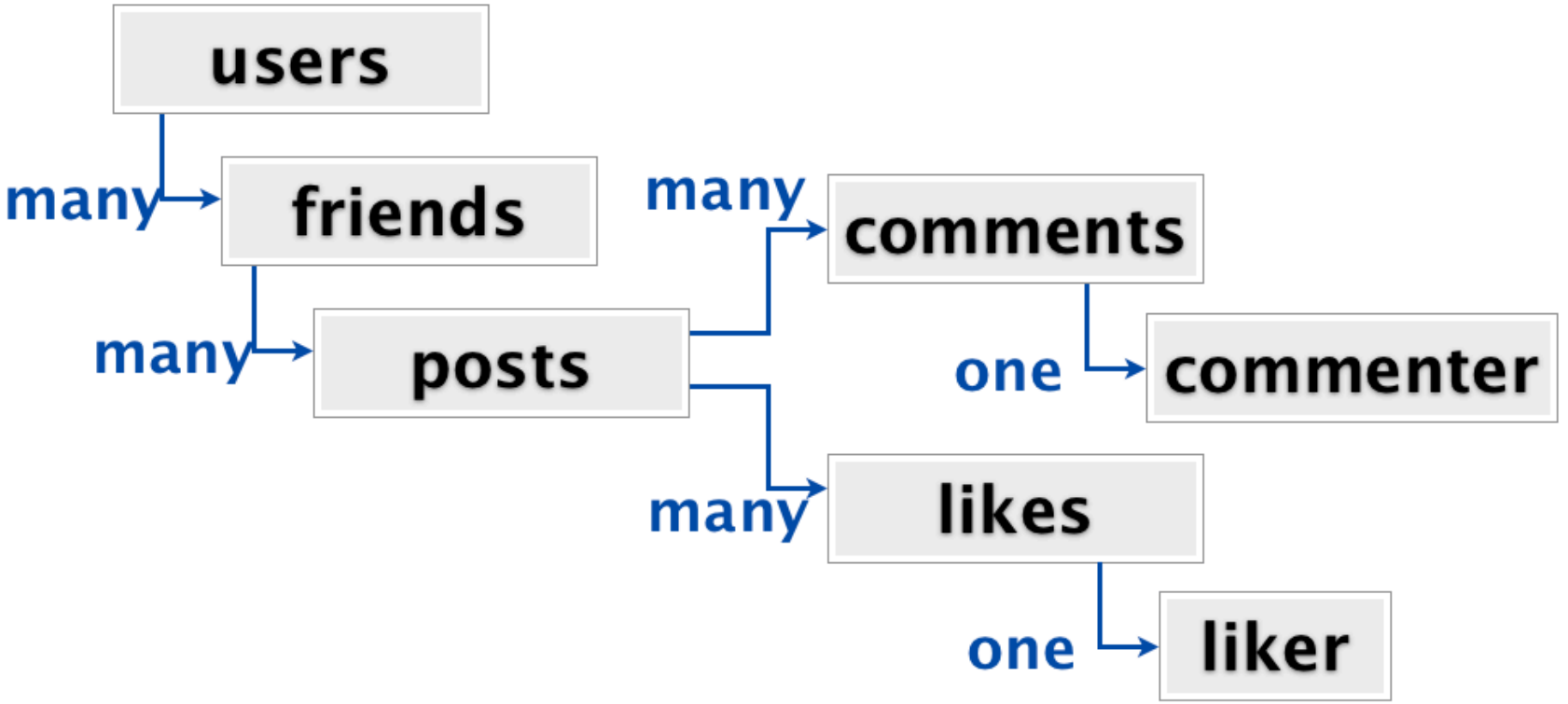
Not all Tree Structures should be saved in Document Databases



Not all Tree Structures should be saved in Document Databases

```
{title: 'Babylon 5',
 seasons: [
  {season_number: '1',
    episodes: [
     {ordinal_within_season: '1',
      title: 'Midnight on the Firing Line'
      reviews: [{...}],
      cast_members: [{...}]
     }
    ]
  }
 ]
}
```

Not all Tree Structures should be saved in Document Databases



MongoDB (CP)

- From humongous ≅ gigantic
- Schema-free document database with tunable consistency
- Allows complex queries and indexing
- Sharding (either range- or hash-based)
- Replication (either synchronous or asynchronous)
- Storage Management:
  - Write-ahead logging for redos (journaling)
  - Storage Engines: memory-mapped files, in-memory, Log-structured merge trees (WiredTiger), …

Basics of MongoDB

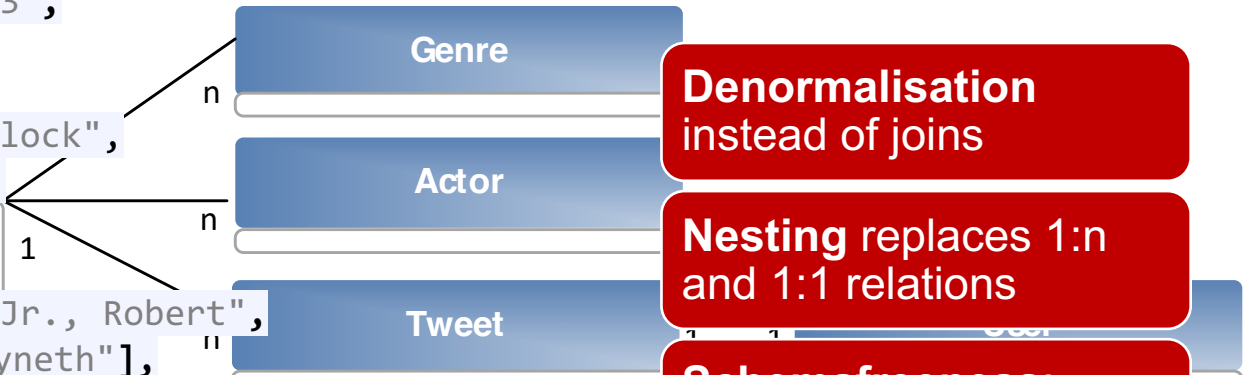
```
> mongod &
> mongo imdb
MongoDB shell version: 2.4.3
connecting to: imdb
> show collections
movies
tweets
> db.movies.findOne({title : "Iron Man 3"})
{
    title : "Iron Man 3",
    year : 2013 ,
    genre : [
         "Action",
         "Adventure",
         "Sci -Fi"],
    actors : [
         "Downey Jr., Robert",
         "Paltrow , Gwyneth",]
}
```

Properties

Arrays, Nesting allowed

Data Modelling of MongoDB

```
{
    "_id" : ObjectId("51a5d316d70beffe74ecc940")
    title : "Iron Man 3",
    year : 2013,
    rating : 7.6,
    director: "Shane Block",
    genre : ["Action",
         "Adventure",
         "Sci -Fi"],
    actors : ["Downey Jr., Robert",
         "Paltrow , Gwyneth"],
    tweets : [ {
            "user" : "Franz Kafka",
            "text" : "#nowwatching Iron Man 3",
            "retweet" : false,
            "date" : ISODate("2013-05-29T13:15:51Z")
    }]
}
```



Denormalisation instead of joins

Nesting replaces 1:n and 1:1 relations

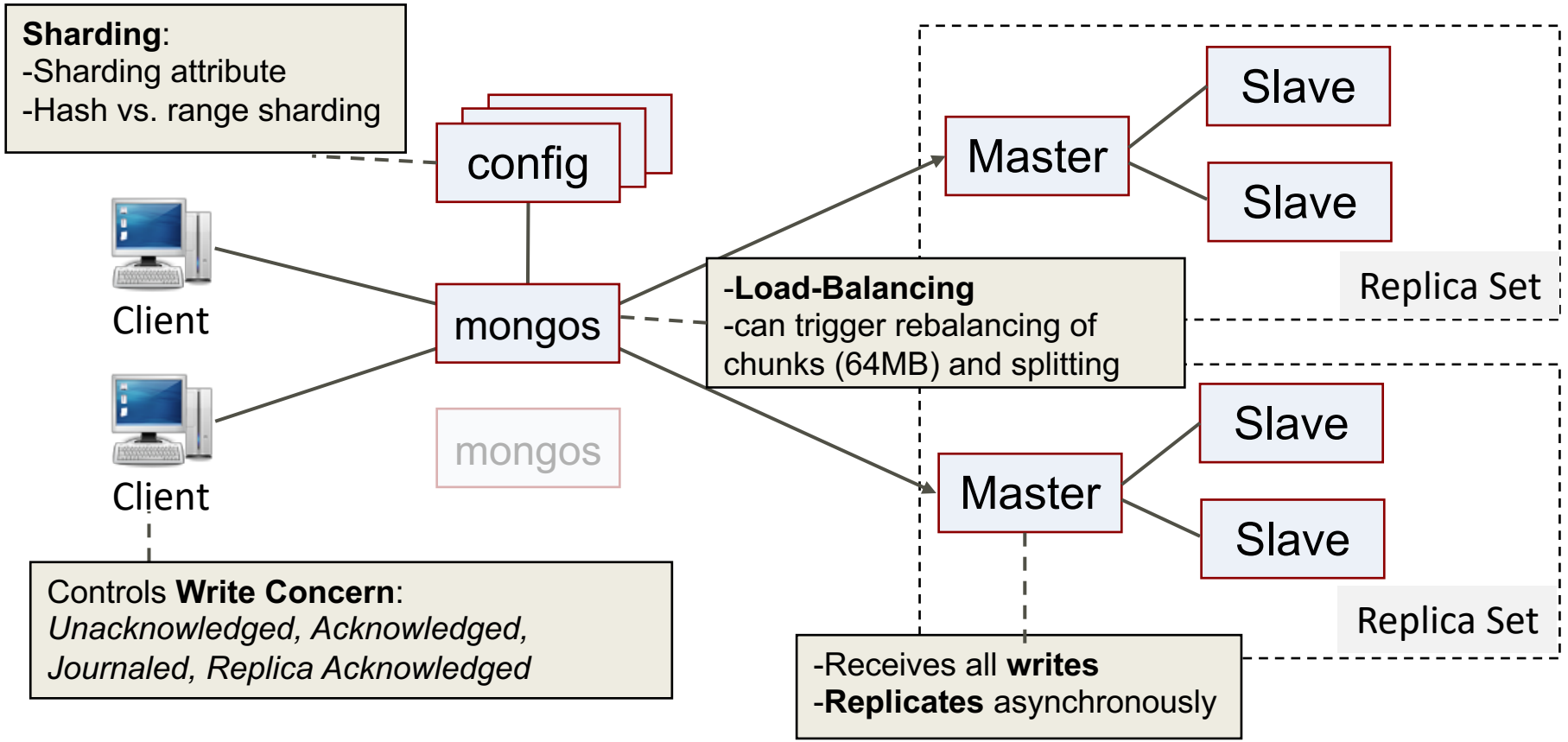
Schemafreeness: Attributes per document

Unit of atomicity: document

Movie
Document

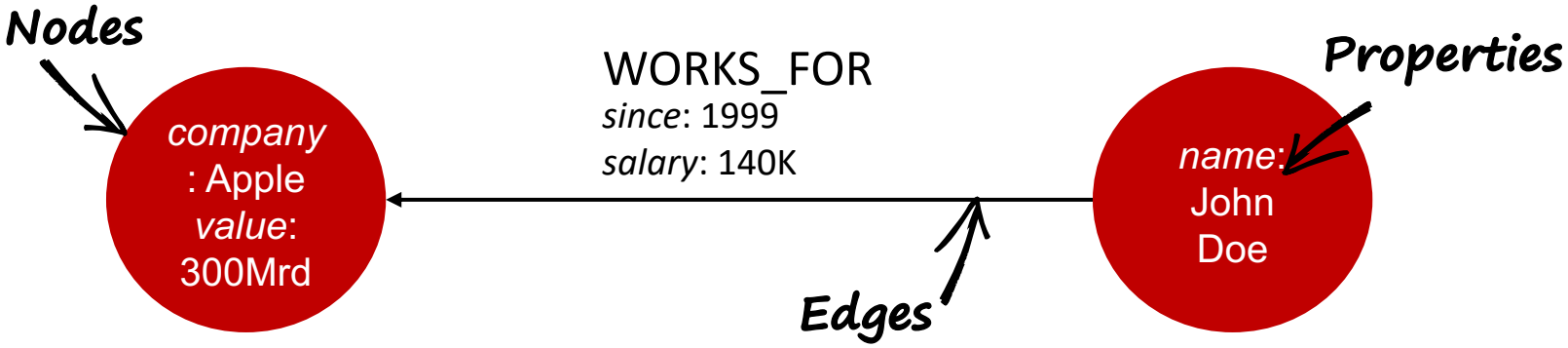
Principles

Sharding and Replication of MongoDB



Graph Databases

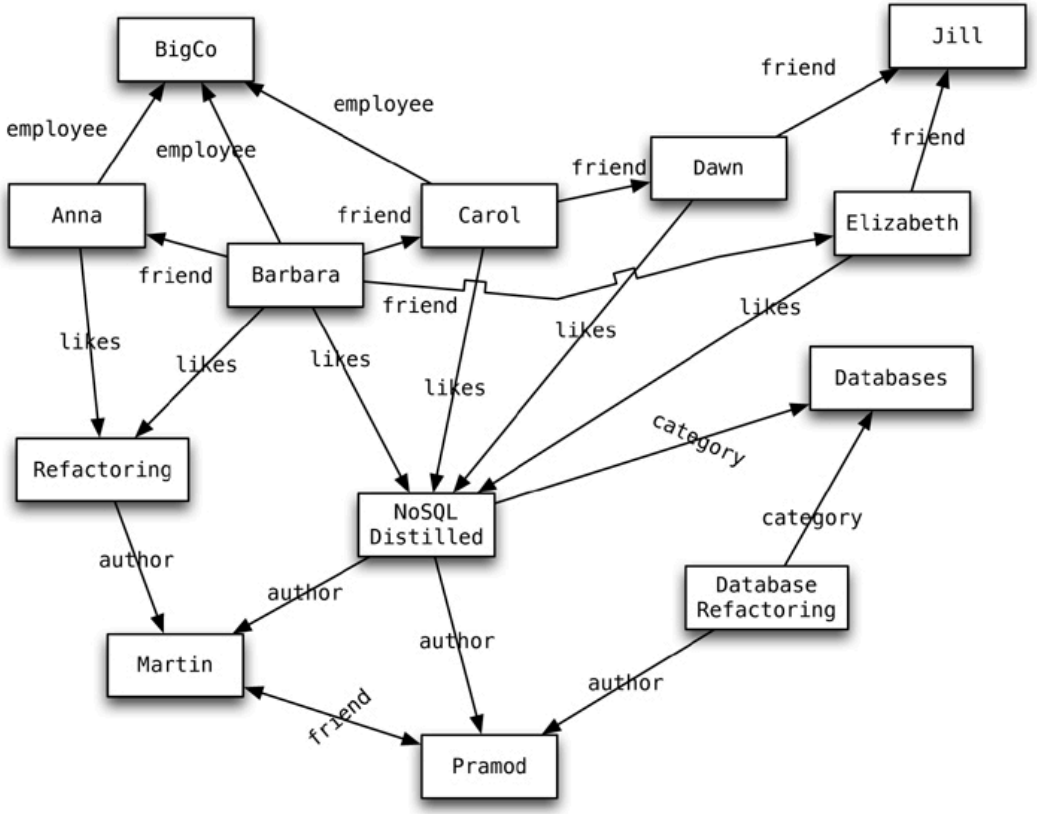
- Data model: G = (V, E): Graph-Property Modell
- Interface: Traversal algorithms, querys, transactions



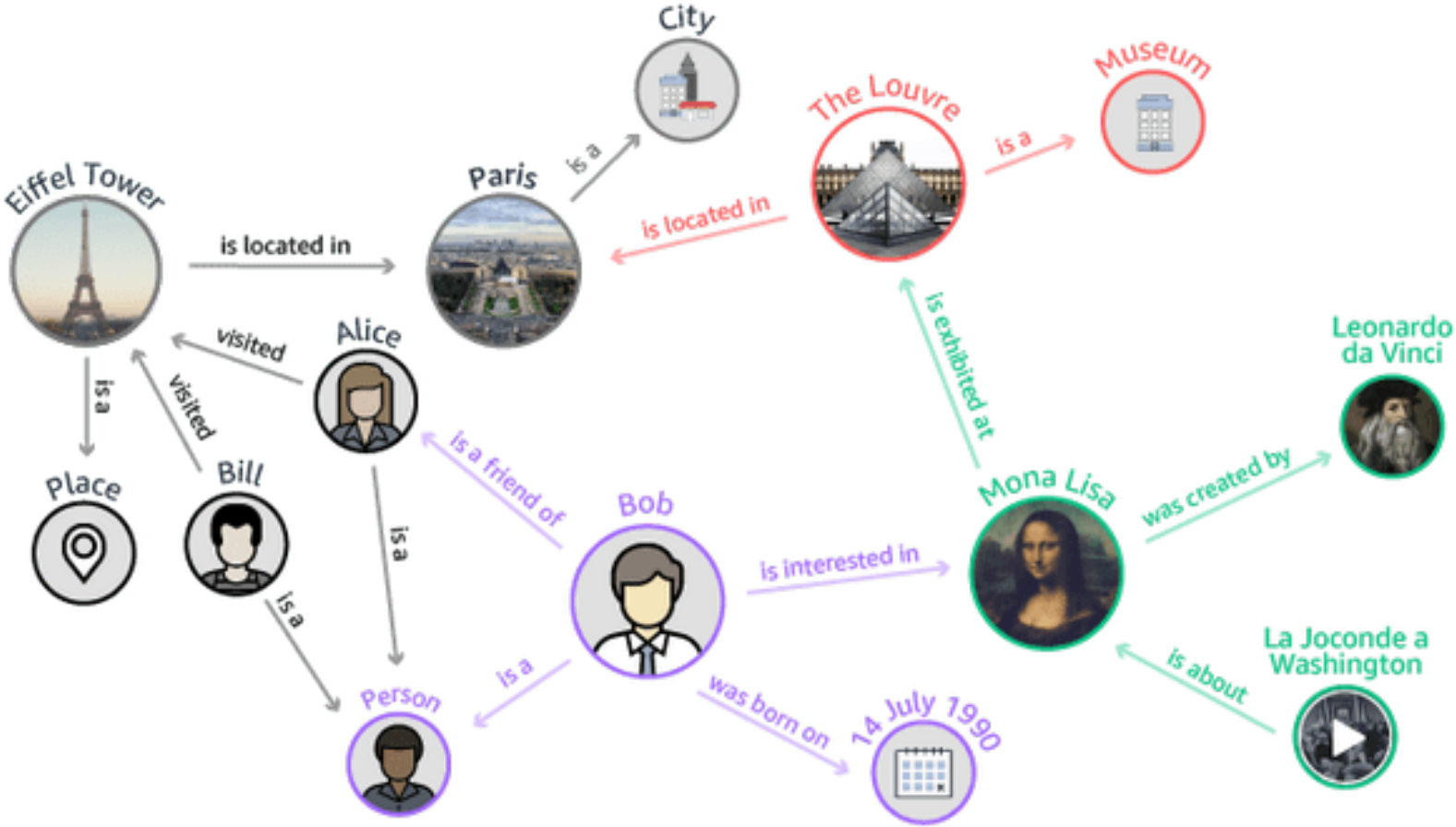
- Examples: Neo4j (CA), InfiniteGraph (CA), OrientDB (CA)

Source: Gessert (2017)

Graph Structures

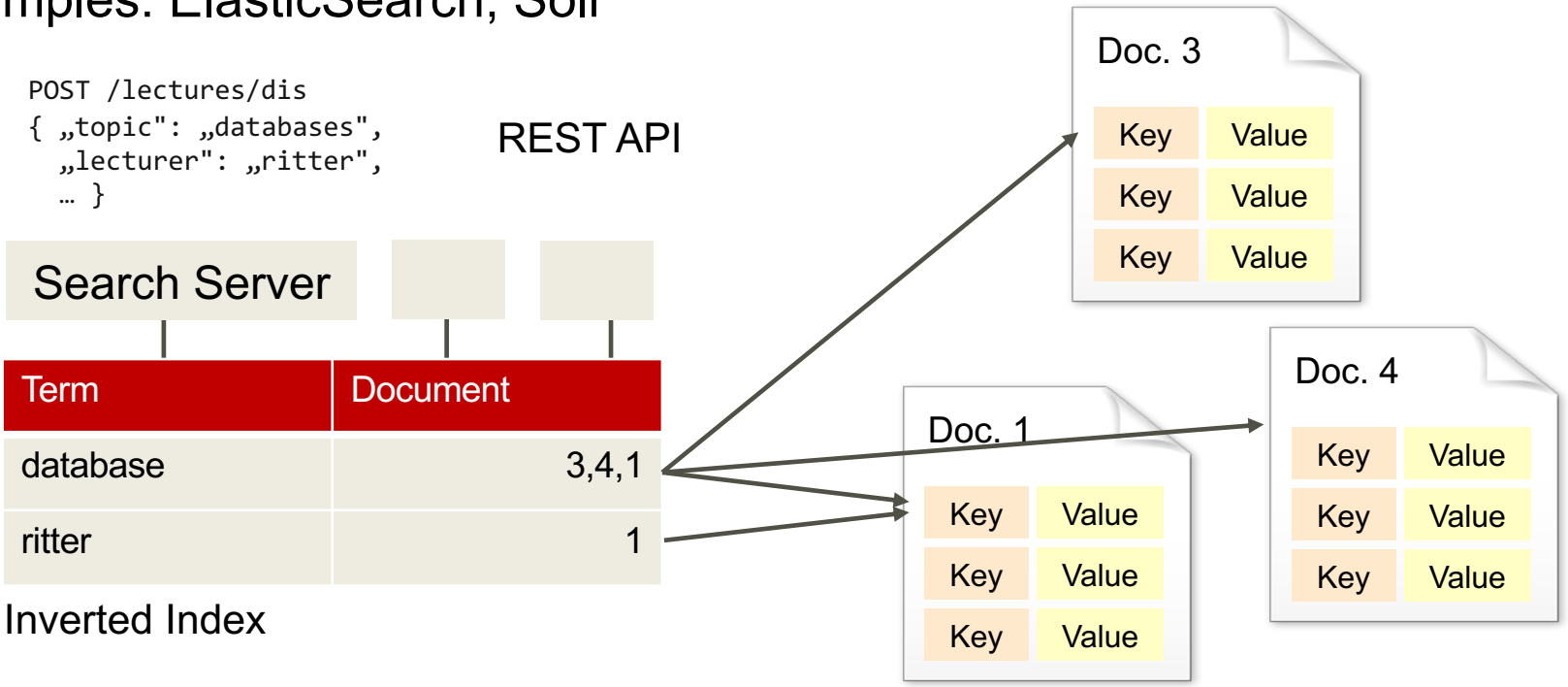


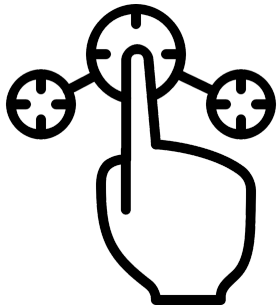
Knowledge Graph



Search Platforms

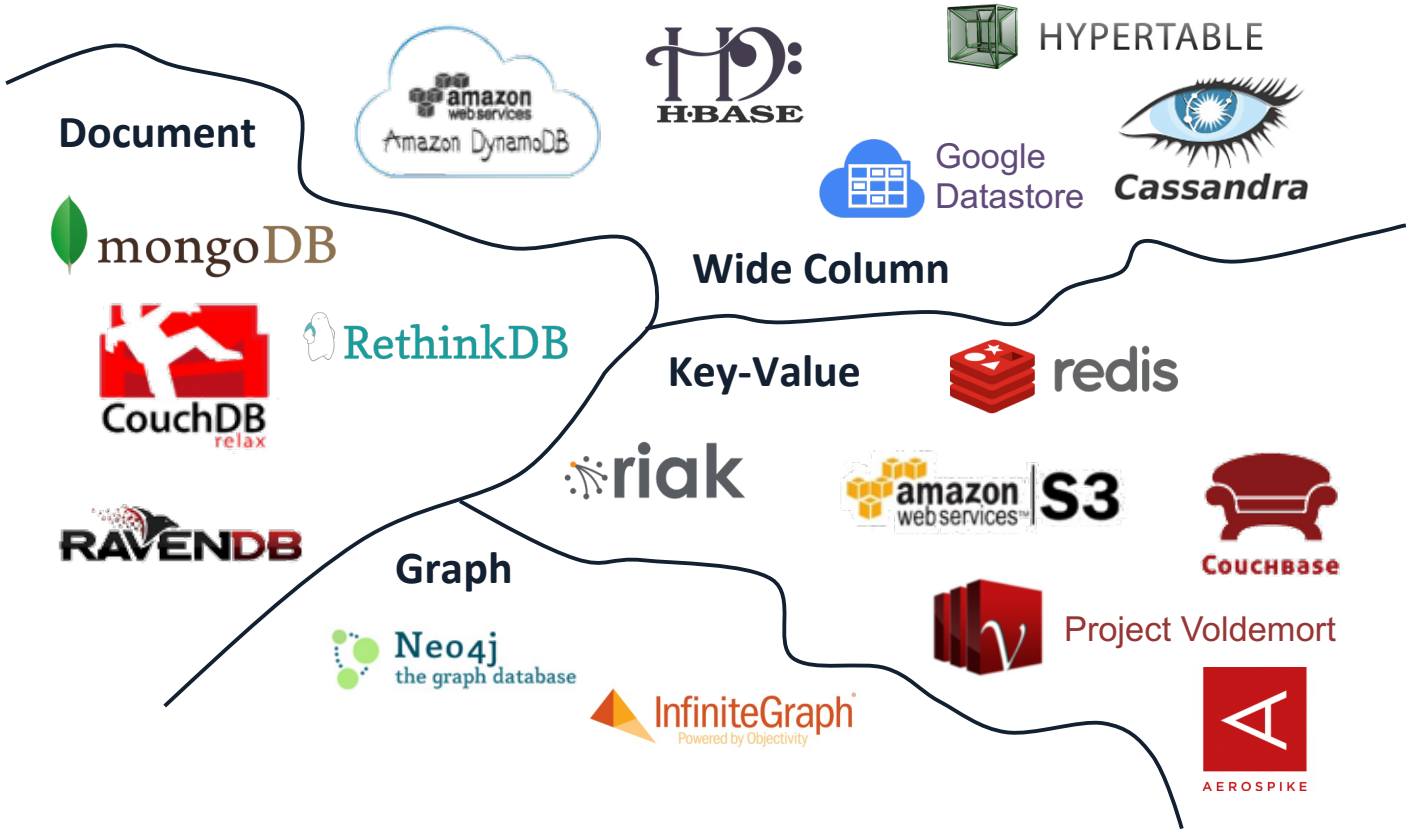
- Data model: vectorspace model, docs + metadata
- Examples: ElasticSearch, Solr





NoSQL Selection

NoSQL Landscape



Database Popularity May 2020

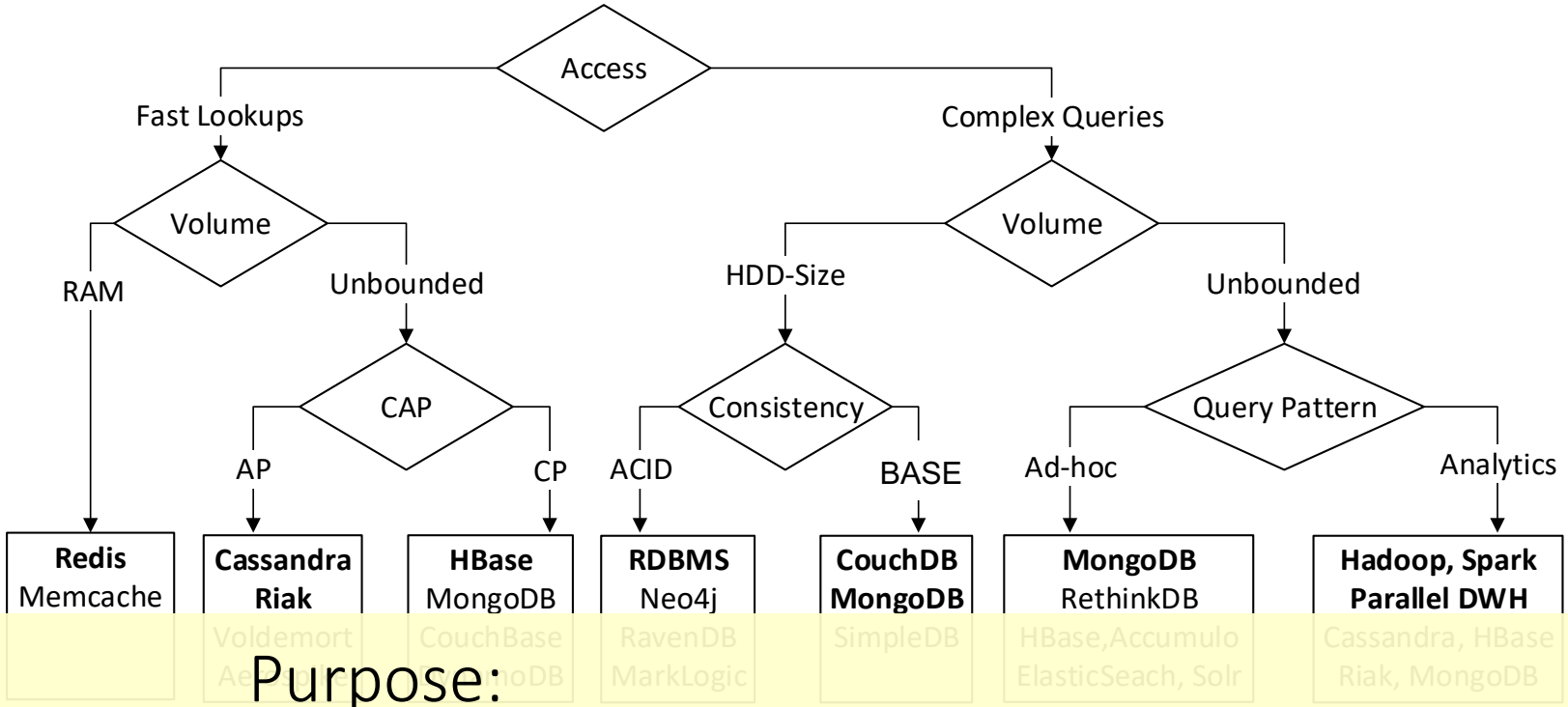
#	System	Model	Score
1.	Oracle	Relational DBMS	1345
2.	MySQL	Relational DBMS	1282
3.	MS SQL Server	Relational DBMS	1078
4.	PostgreSQL	Relational DBMS	514
5.	**MongoDB**	**Document store**	439
6.	DB2	Relational DBMS	162
7.	**Elasticsearch**	**Search engine**	149
8.	**Redis**	**Key-value store**	143
9.	SQLite	Relational DBMS	123
10.	MS Access	Relational DBMS	120
11.	**Cassandra**	**Wide column store**	119
12.	MariaDB	Relational DBMS	90
13.	**Splunk**	**Search engine**	88
14.	Hive	Relational DBMS	82
15.	Terradata	Relational DBMS	74
16.	**Amazon DynamoDB**	**Multi-modal store**	65
17.	SAP Adaptive Server	Relational DBMS	54
18.	**Solr**	**Search engine**	53
19.	FileMaker	Relational DBMS	51
20.	SAP HANA	Relational DBMS	51
21.	**Neo4j**	**Graph DBMS**	50
22.	**HBase**	**Wide column store**	50
23.	Azure SQL Database	Relational DBMS	43
24.	**Azure Cosmos DB**	**Multi-modal store**	31
25.	**Couchbase**	**Document store**	29

Scoring: Google/Bing results, Google Trends, Stackoverflow, job offers, LinkedIn

Source: https://db-engines.com/en/ranking

Know your Tools

“Having a big toolbox is impressive. But knowing **when to use what tool** and why separates the true expert from the show-off.” *Gregor Hohpe*



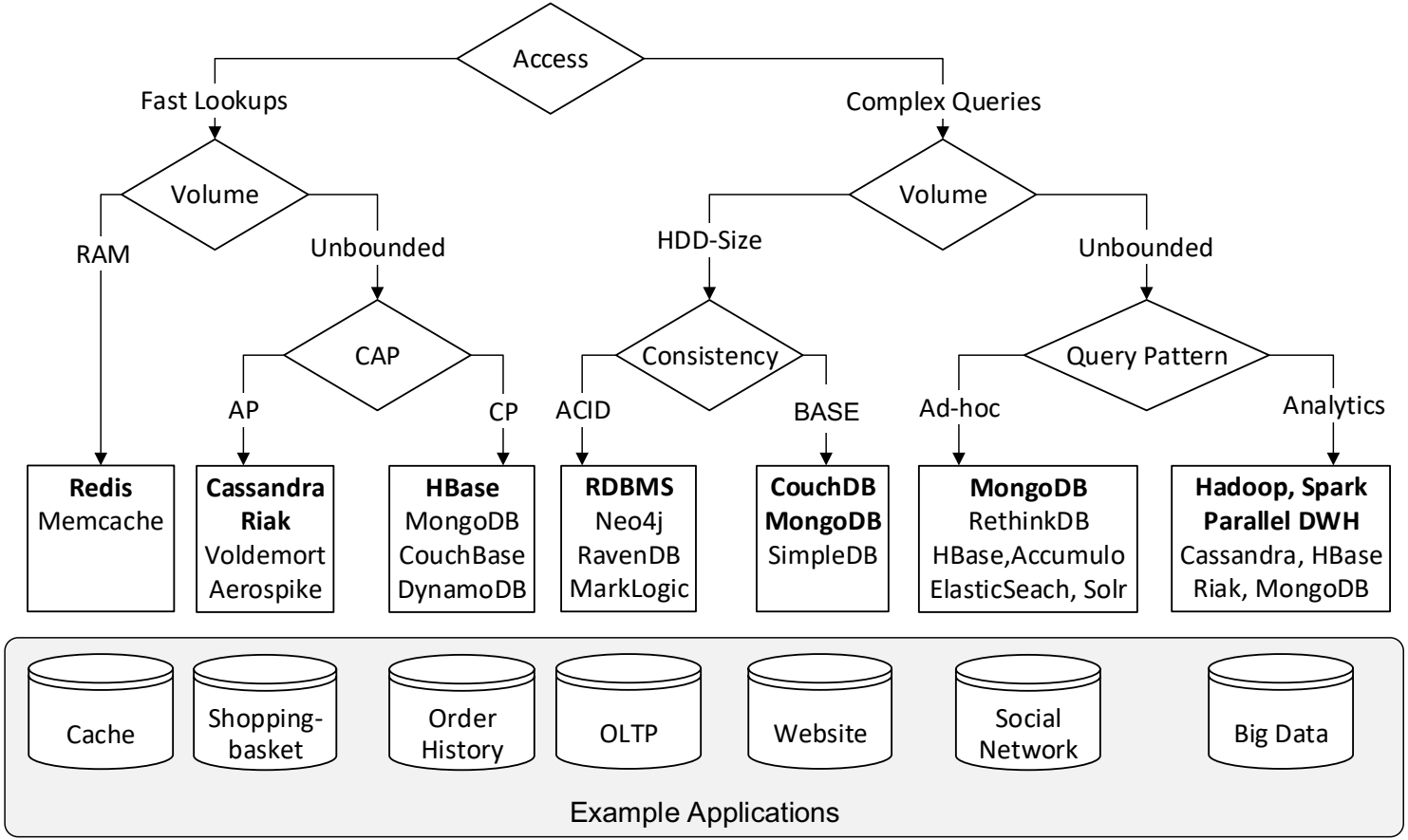
Purpose:

Application Architects: narrowing down the potential system candidates based on requirements

Database Vendors/Researchers: clear communication and design of system trade-offs

Source: Gessert (2017)

NoSQL Decision Tree

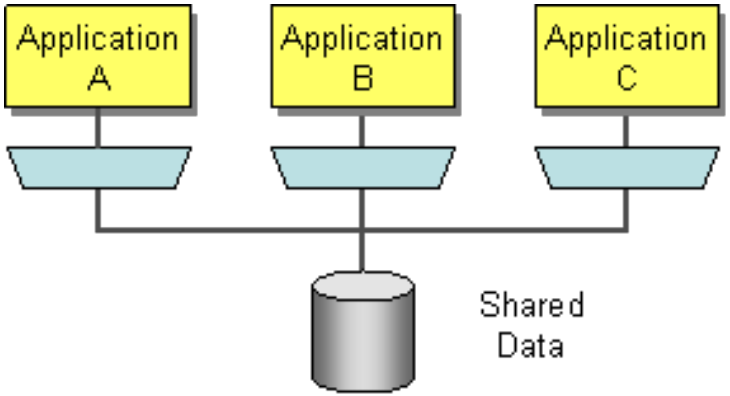


Source: Gessert (2017)

Polyglot Persistence

Integration through a shared database

- **Problem**: How can I integrate multiple distributed applications or services so that they work together and can exchange information?



- **Solution**: Integrate applications by having them store their data in a single **Shared Database**.

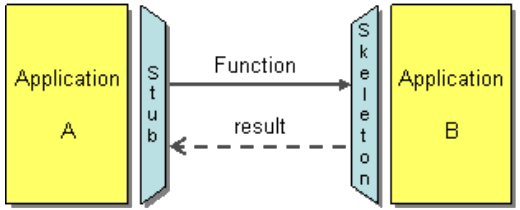
Polyglot Persistence

Source: Fowler (2016)

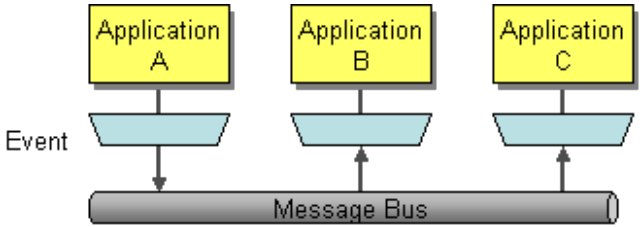
Integration without a shared Database

Service Integration Patterns

Web API



Message Queue



File Transfer Integration