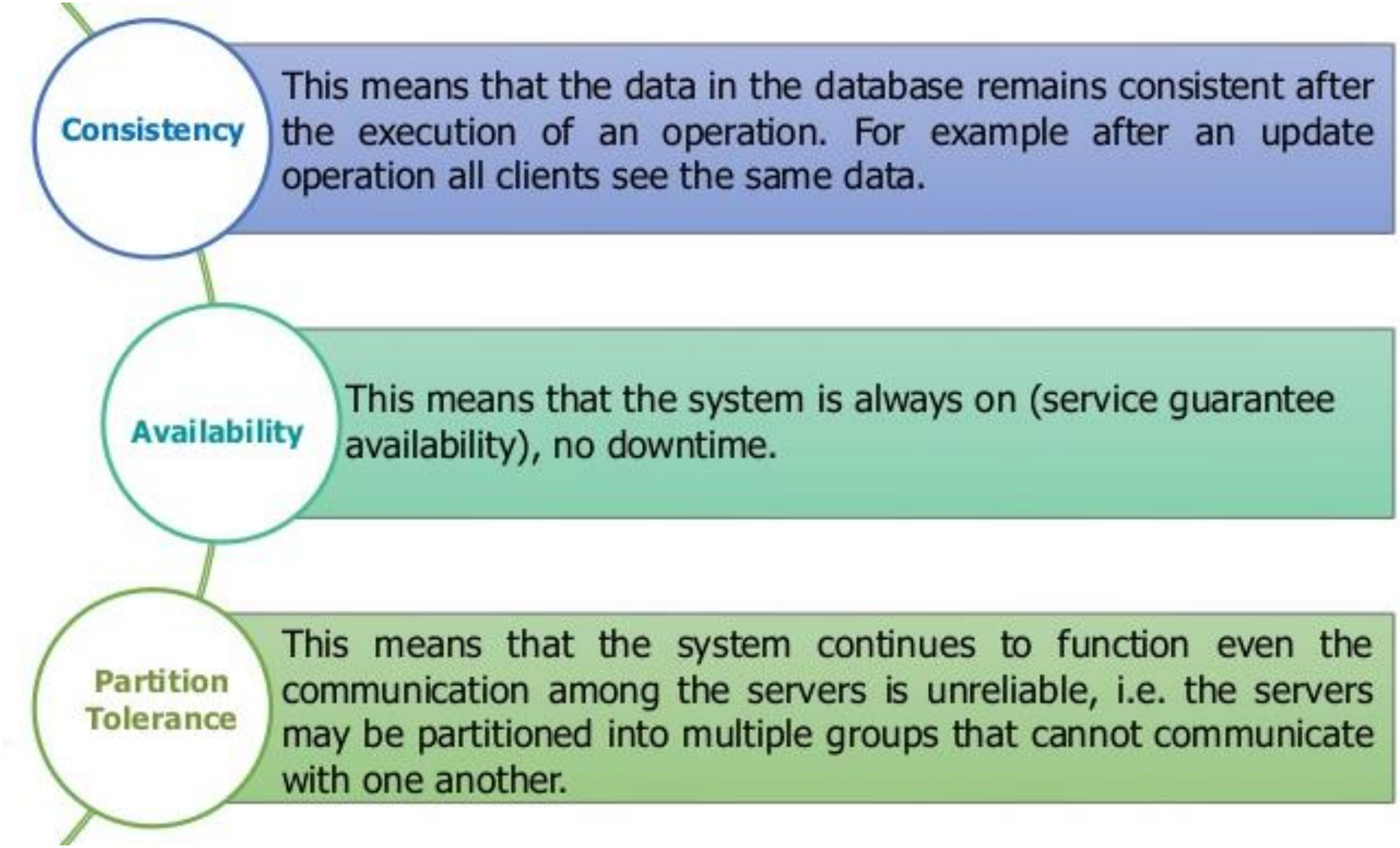
NoSQL Databases: HBase

Data Architecture

- No standard solution that fits all
- Business and data defines the architecture
- Multiple databases, different types depending on the characteristics of each data subset

CAP



Consistency

This means that the data in the database remains consistent after the execution of an operation. For example after an update operation all clients see the same data.

Availability

This means that the system is always on (service guarantee availability), no downtime.

Partition Tolerance

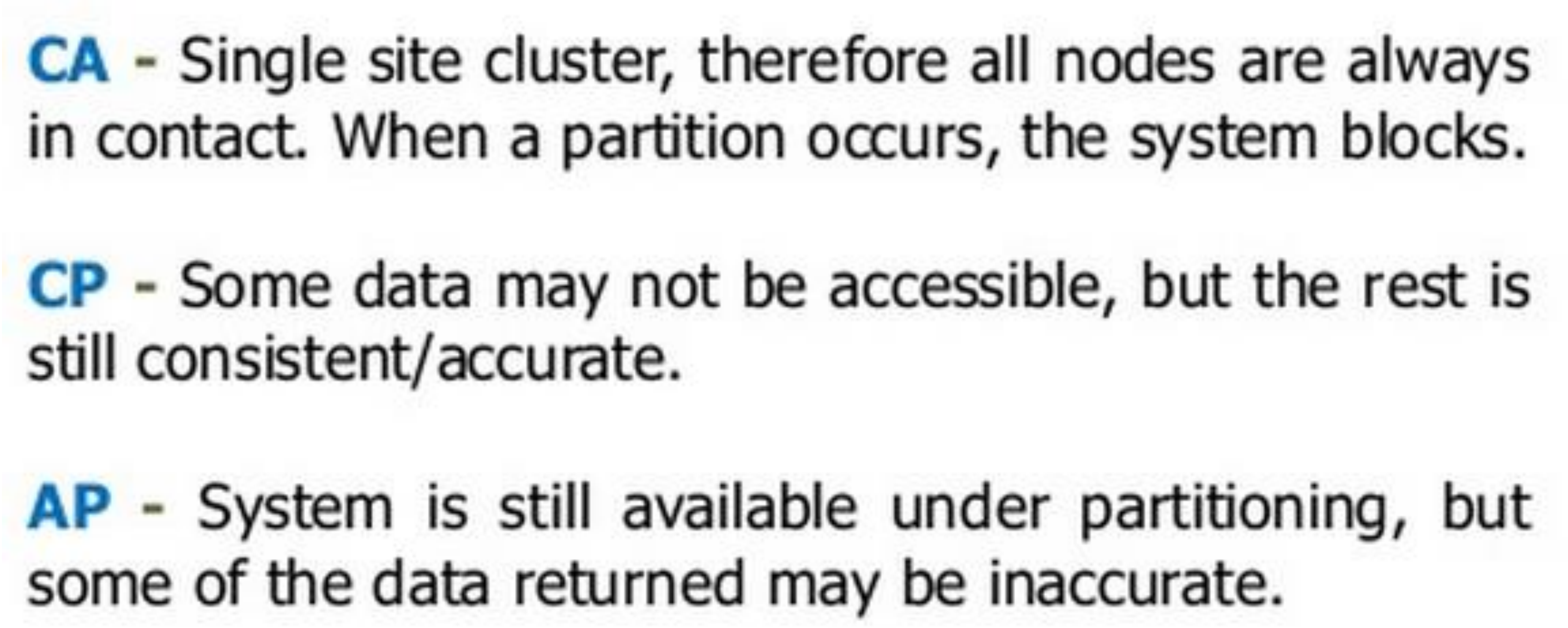
This means that the system continues to function even the communication among the servers is unreliable, i.e. the servers may be partitioned into multiple groups that cannot communicate with one another.

datasciencedojo
unleash the data scientist in you

CAP

- **Consistency** - A read is guaranteed to return the most recent write for a given client.
- **Availability** - A non-failing node will return a reasonable response within a reasonable amount of time (no error or timeout).
- **Partition Tolerance** - The system will continue to function when network partitions occur.

CAP Theorem

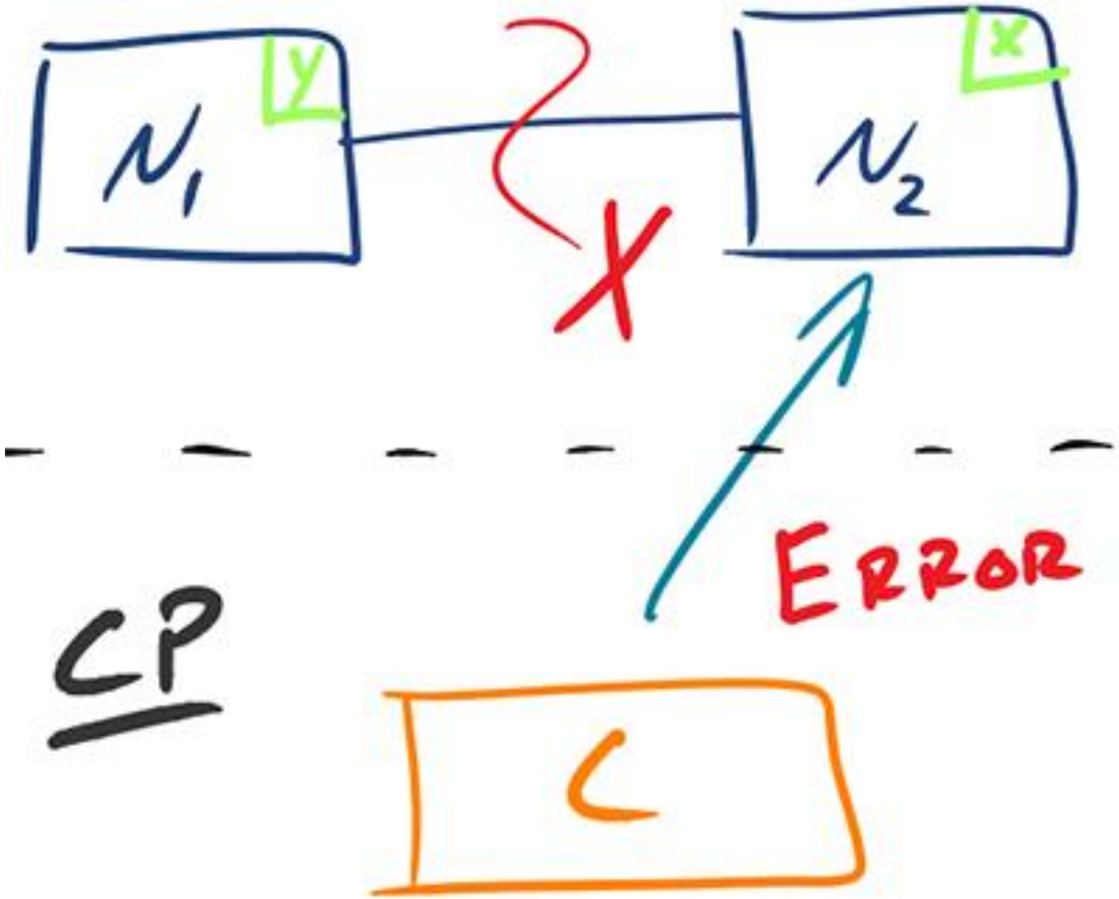


CA - Single site cluster, therefore all nodes are always in contact. When a partition occurs, the system blocks.

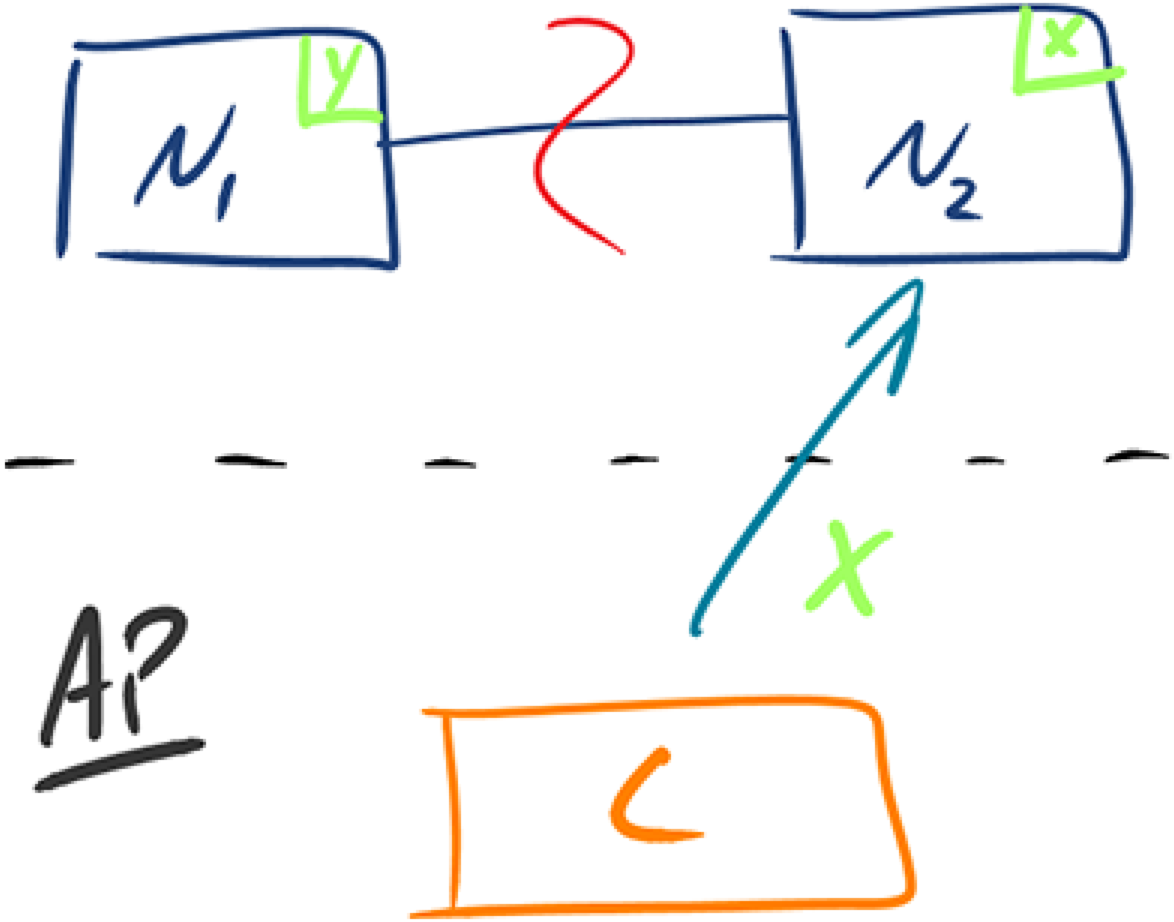
CP - Some data may not be accessible, but the rest is still consistent/accurate.

AP - System is still available under partitioning, but some of the data returned may be inaccurate.

CP

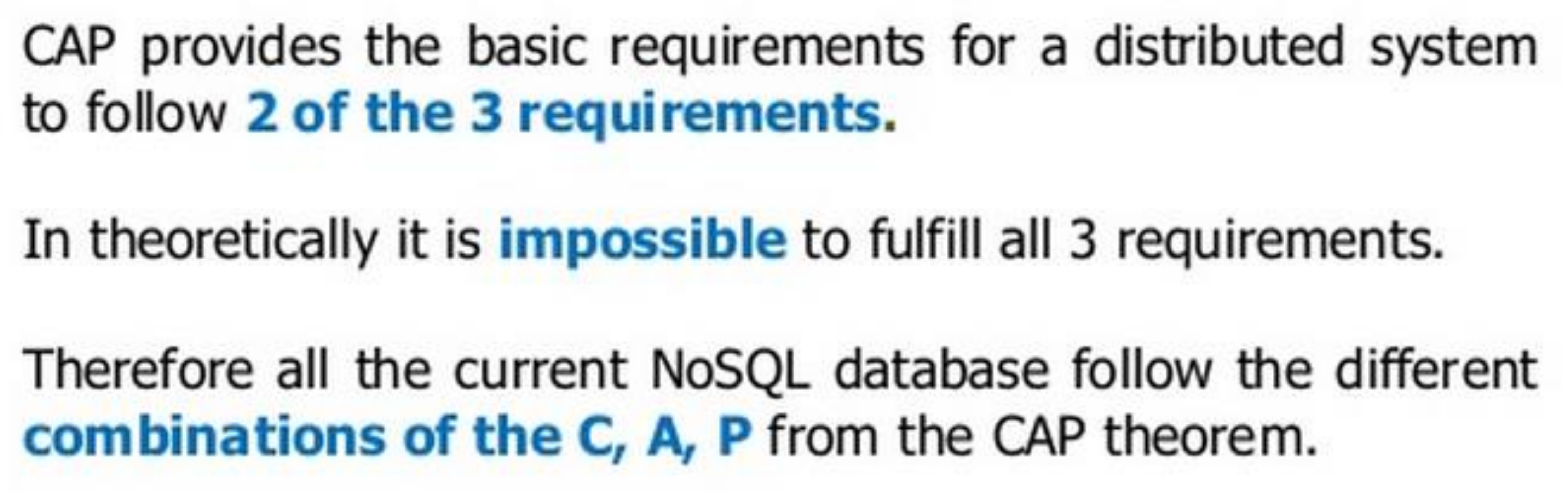


AP



datasciencedojo
unleash the data scientist in you

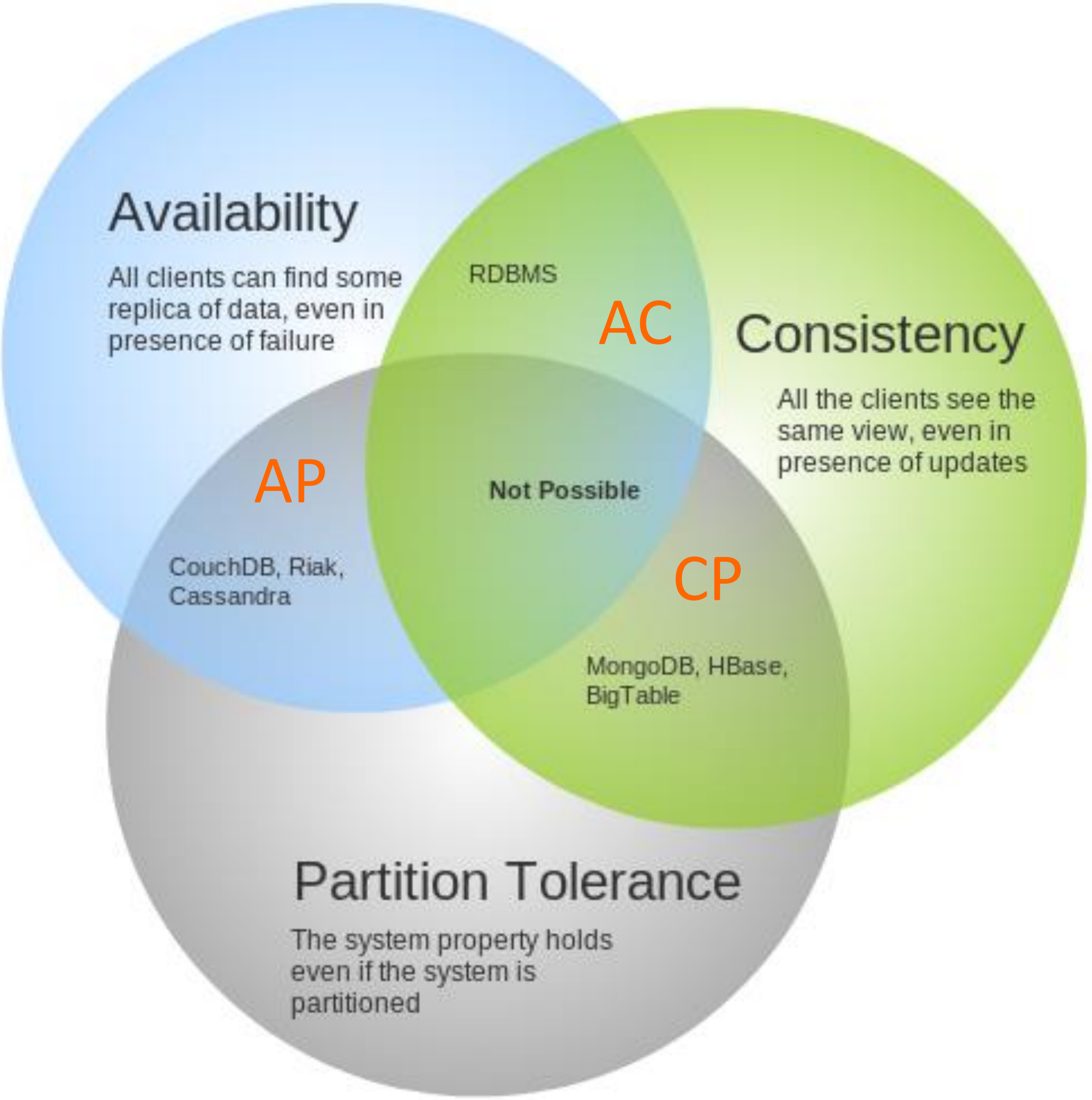
CAP Theorem



CAP provides the basic requirements for a distributed system to follow **2 of the 3 requirements.**

In theoretically it is **impossible** to fulfill all 3 requirements.

Therefore all the current NoSQL database follow the different **combinations of the C, A, P** from the CAP theorem.



Availability

All clients can find some replica of data, even in presence of failure

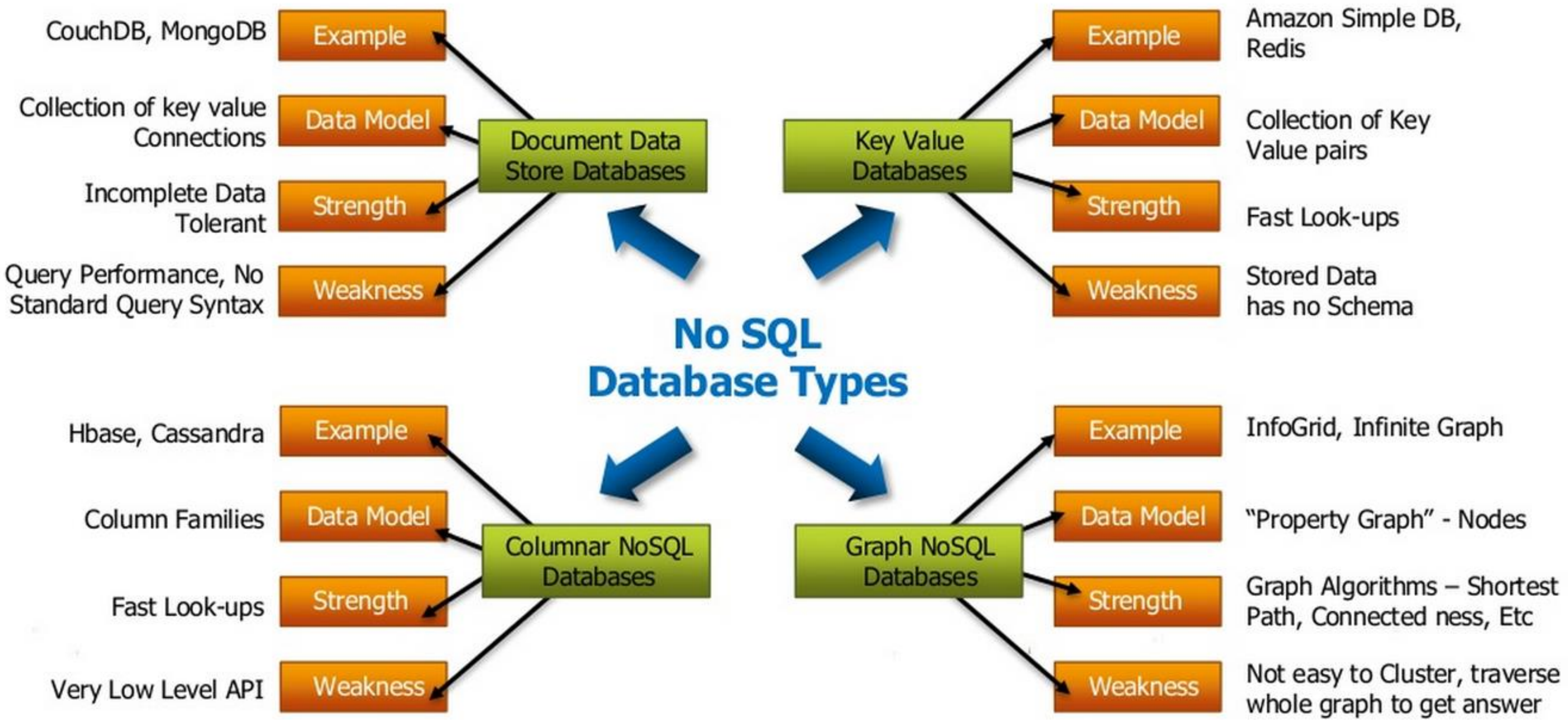
Consistency

All the clients see the same view, even in presence of updates

Partition Tolerance

The system property holds even if the system is partitioned

datasciencedojo
unleash the data scientist in you



No SQL Database Types

Document Data Store Databases

- Example: CouchDB, MongoDB
- Data Model: Collection of key value Connections
- Strength: Incomplete Data Tolerant
- Weakness: Query Performance, No Standard Query Syntax

Key Value Databases

- Example: Amazon Simple DB, Redis
- Data Model: Collection of Key Value pairs
- Strength: Fast Look-ups
- Weakness: Stored Data has no Schema

Columnar NoSQL Databases

- Example: Hbase, Cassandra
- Data Model: Column Families
- Strength: Fast Look-ups
- Weakness: Very Low Level API

Graph NoSQL Databases

- Example: InfoGrid, Infinite Graph
- Data Model: "Property Graph" - Nodes
- Strength: Graph Algorithms – Shortest Path, Connected ness, Etc
- Weakness: Not easy to Cluster, traverse whole graph to get answer

What is HBase

- Distributed, non-relational database
  - Columnar, schema-free data model
  - NoSQL on top of Hadoop
- Large scale
  - Linear scalability
  - Billions of rows X millions of columns
  - Many deployments with 1000+ nodes, PBs of data
- Low latency
  - Real-time random read/writes
- Open Source
  - Modeled after Google's BigTable
  - Started in 2006

FLURRY

Adobe

ebay

salesforce

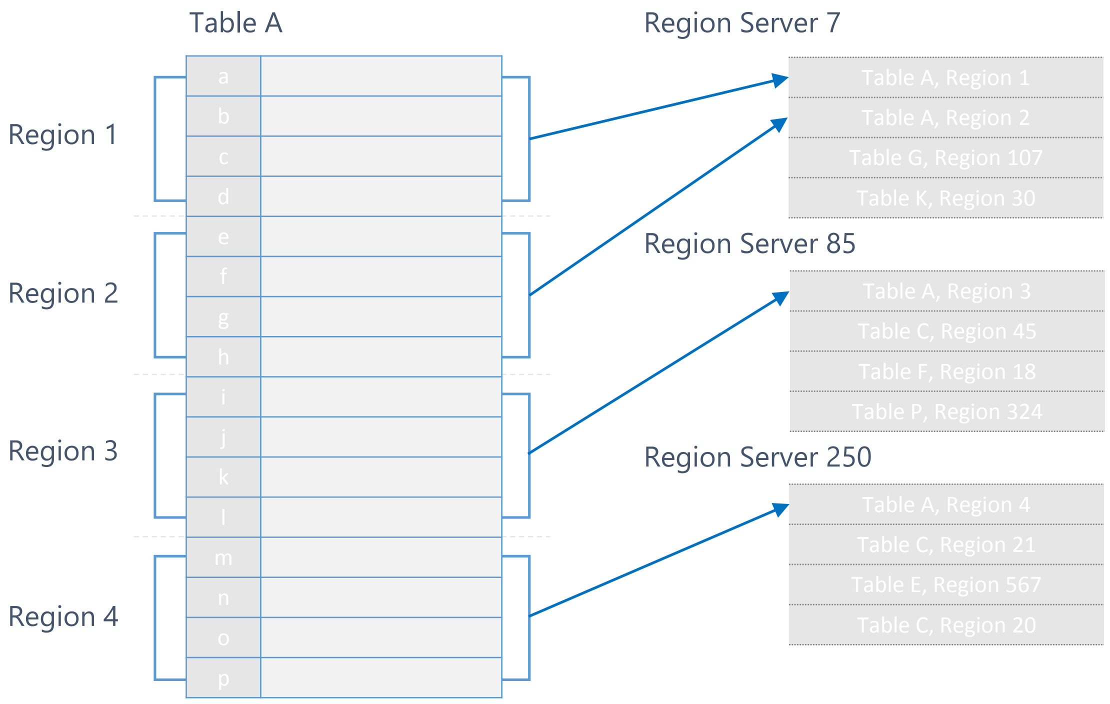
facebook

YAHOO!

Bloomberg

Data Model

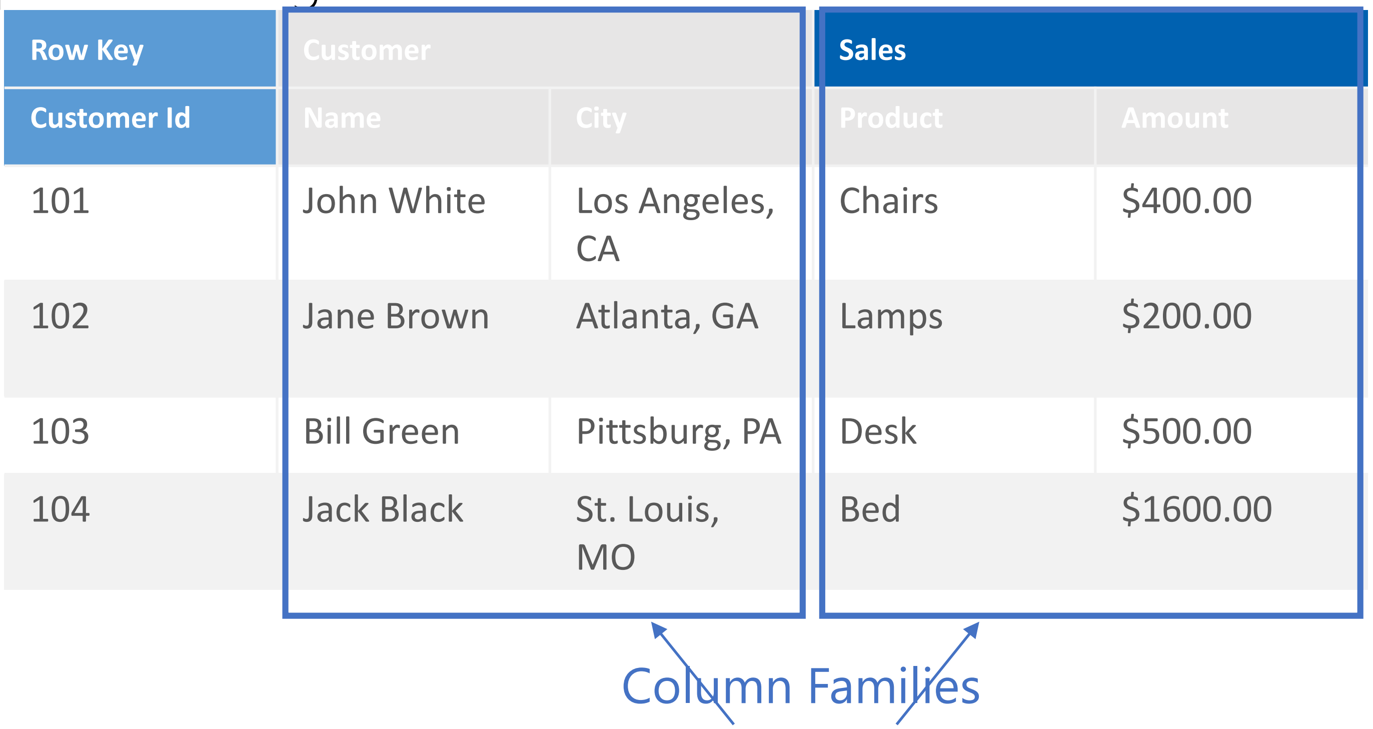
- Scale-out architecture
  - Automatic sharding of tables
  - Automatic failover
  - Strong consistency for reads and writes
- APIs
  - Get/Put
  - Scan
  - Coprocessors



datasciencedojo
unleash the data scientist in you

Performance Features

- Column Families
- In-memory caching
- High throughput streaming writes



Row Key	Customer		Sales	
Customer Id	Name	City	Product	Amount
101	John White	Los Angeles, CA	Chairs	$400.00
102	Jane Brown	Atlanta, GA	Lamps	$200.00
103	Bill Green	Pittsburg, PA	Desk	$500.00
104	Jack Black	St. Louis, MO	Bed	$1600.00

Sharding

- Holding rows of database on different partitions
- Same table divided onto different servers, even different geographies
- Reduces index size

Sharding

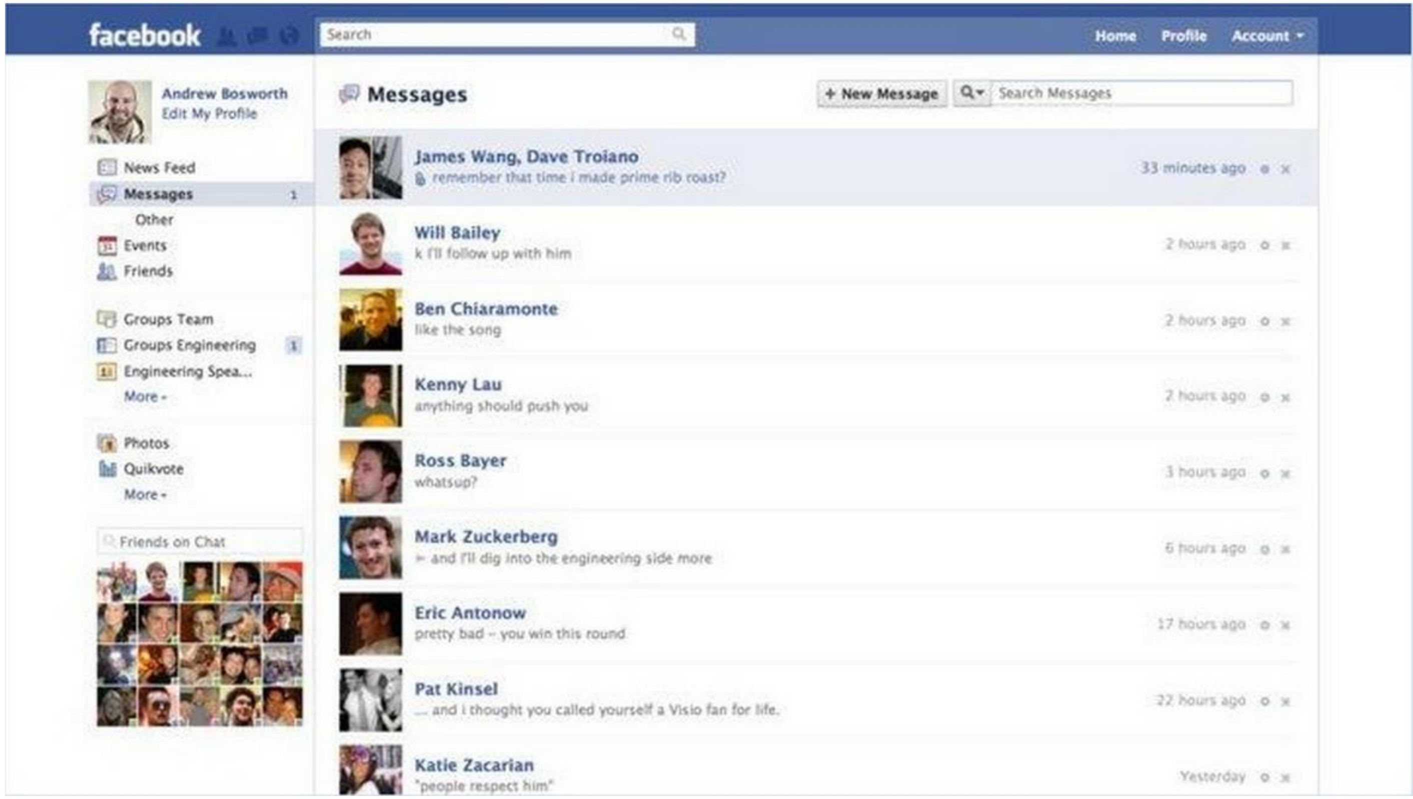
- More reliance on interconnection between servers
- Increased latency in querying when more than one shard must be searched
  - Some searches are fast, others are slow
- Often no guarantees about cross shard consistency

Notable Capabilities

- Integration features
  - Integration with Hadoop MapReduce, Hive, Tez
  - Bulk import of large amounts of data
- Client APIs
  - Java, REST, python, node.js, php, .NET

Use case #1: key value store

- Key value store
  - Message systems
  - Content management systems
- Examples
  - Facebook Messages
  - Twitter-like messages
  - Webtable – web crawler/indexer



James Wang, Dave Troiano
remember that time i made prime rib roast?
33 minutes ago

Will Bailey
k I'll follow up with him
2 hours ago

Ben Chiaramonte
like the song
2 hours ago

Kenny Lau
anything should push you
2 hours ago

Ross Bayer
whatsup?
3 hours ago

Mark Zuckerberg
and I'll dig into the engineering side more
6 hours ago

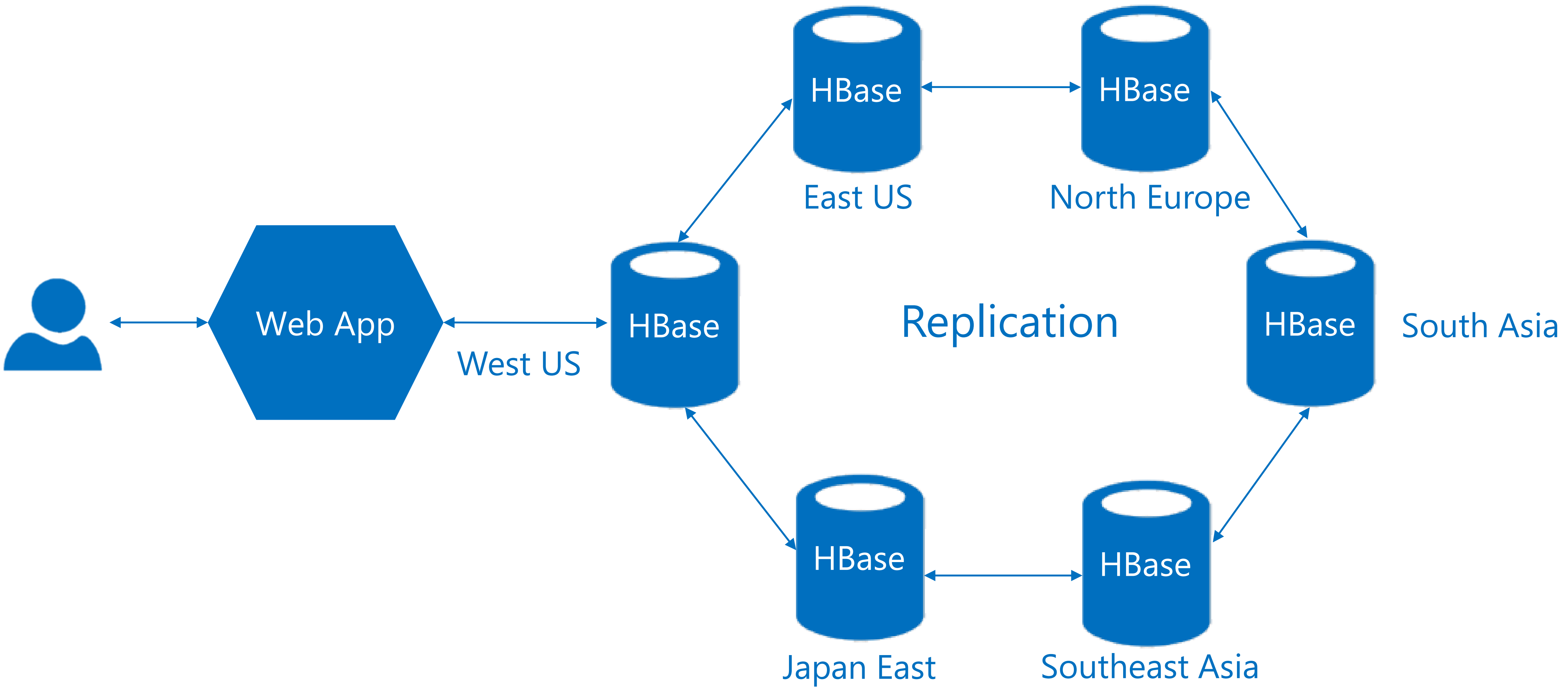
Eric Antonow
pretty bad -- you win this round
17 hours ago

Pat Kinsel
... and i thought you called yourself a Visio fan for life.
22 hours ago

Katie Zacarian
"people respect him"
Yesterday

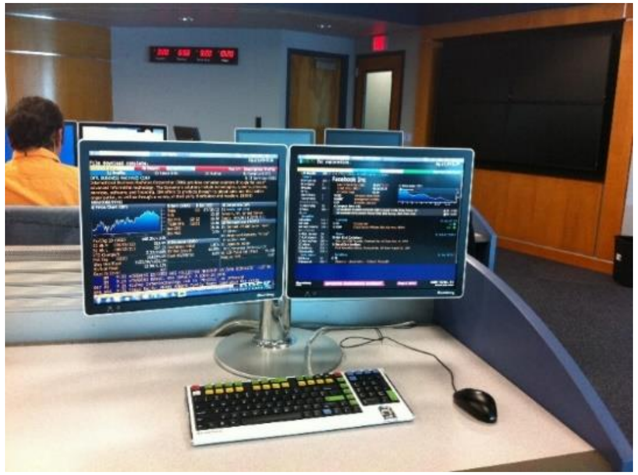
datasciencedojo
unleash the data scientist in you

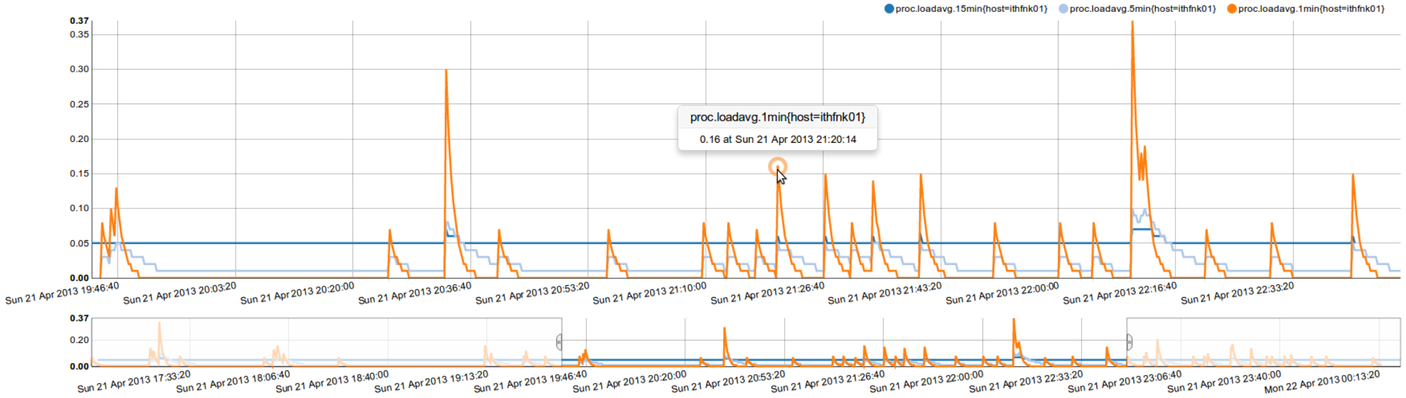
Use case #1: key value store



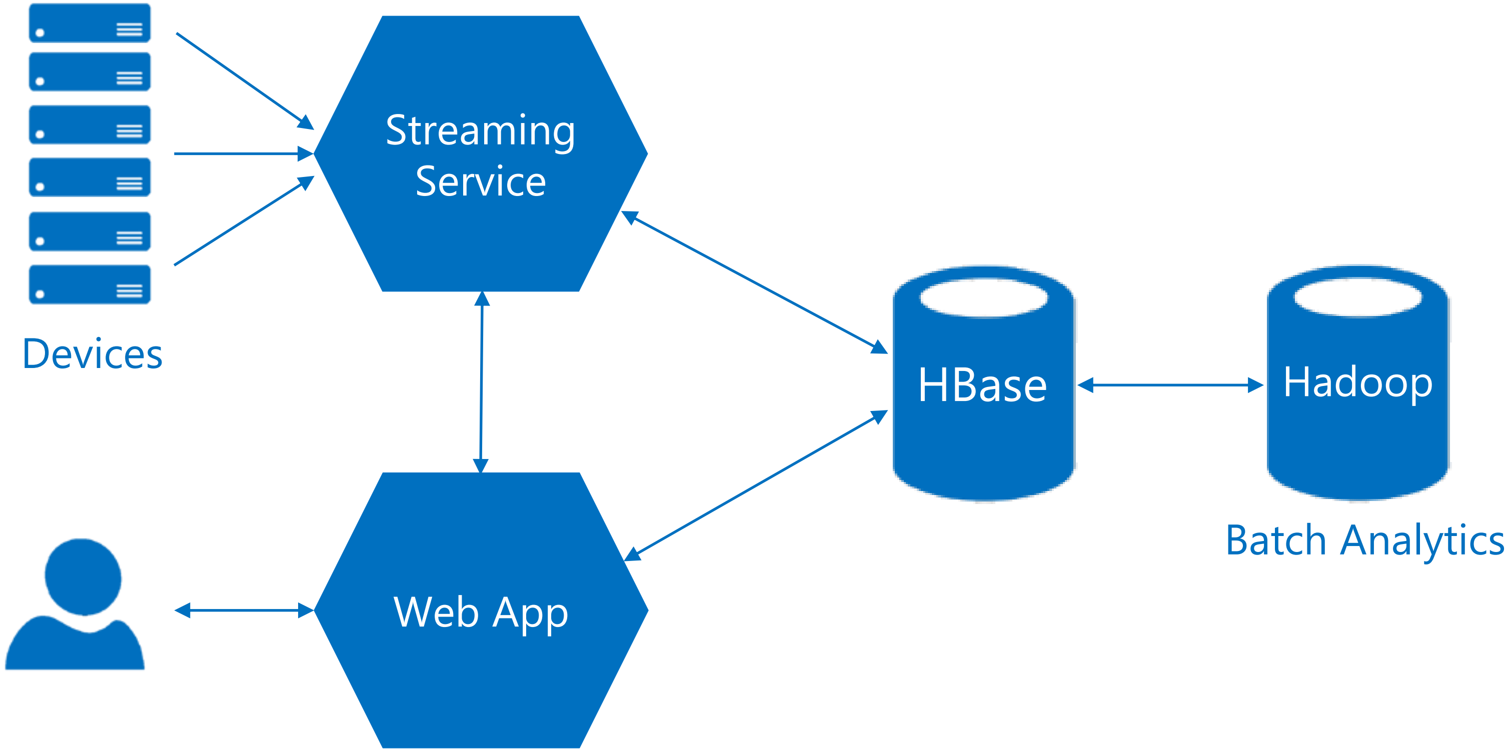
Use case #2: sensor data

- Sensor data
  - Social analytics
  - Time series databases
  - Interactive dashboards with trends, counters, etc
  - Audit log systems





Use case #2: sensor data



HBase as a platform

- Running on top of HBase using it as a datastore:
  - Phoenix
  - Kiji
  - Titan

  OpenTSDB

  Tephra

  Kylin

- Integrated with HBase:
  - Hive
  - Storm
  - Flume
  - Ganglia

  Pig

  Spark

  Solr