
Couchbase

Introduction

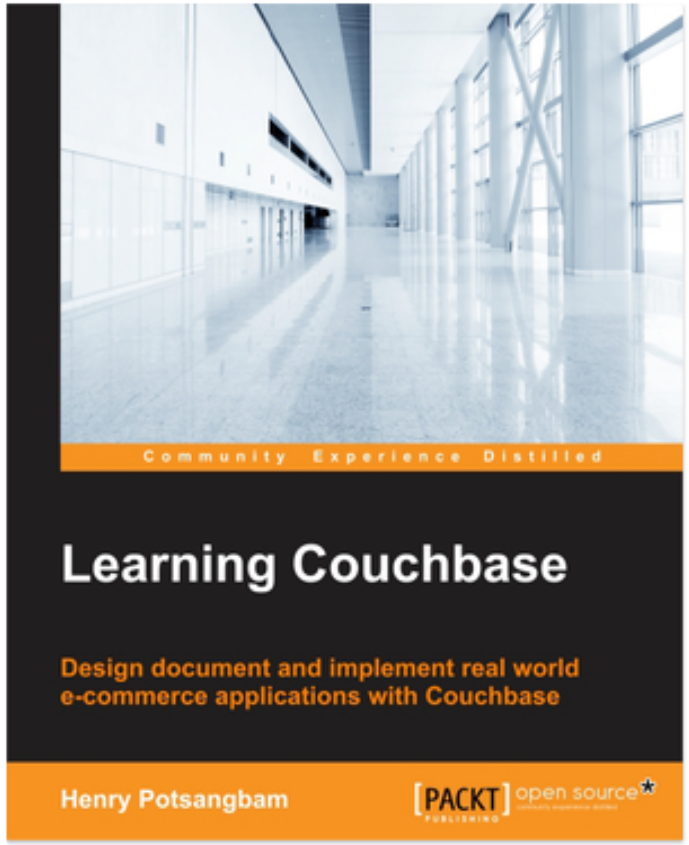
HP

Henry R.P

OSTech
Training & Consulting

Author, IT Architect & Corporate trainer
20 +Year of IT Experience

- **Hadoop Administrator - Mapr**
- **TOGAF**
- **IBM Certified Application & Solution Architect**
- **SAP Certified EP & ABAP Development Consultant .**
- **CIPM – Certificate in Project Management.**



Author

http://opensourcetech.in

HP

Training & Consulting on:

- **NOSQL & Bigdata** –, Couchbase, Cassandra, MongoDB, Hadoop, CDH, Mapr & HDP
- **Predictive Analytics – R & Python**
- **EAI**:- Mule / Fuse ESB /Spring Integration/ JBI / Apache Camel /Talend/ Apache Service Mix
- **Portal**:- Liferay, SAP Netweaver.
- **Application server**:- WAS, Tomcat , WebLogic, Jboss
- **Architecture**: EA, TOGAF, CoBIT etc.
- **JEE Framework**
- **OSGI** - Eclipse PDE/Equinox/Virgo/Spring DM/ Felix / Karaf

Expertise:

Architecture Design
Cloud Architecture
Sizing
NoSQL Data Design
Bug & Fixing Issues
Performance & Tuning

henry@opensourcetech.in

HP

Clientele

HP

Introduce Yourself.

- Name
- Year of Experience.
- Skills Level
  - RDMS
  - NoSql /Couchbase
- Expectation, if any.

HP

Outline

- Overview - NoSQL Basic
- Couchbase Server Architecture
- Couchbase Administration - Webconsole
- Bucket
- Document Database Basic
- Cluster Administration
- Views
- Indexes
- N1QL Security - LDAP
- XDCR
- FTS
- Eventing service(s)
- Analytics service(s)
- Monitoring & Tuning + Back up.

HP

Schedule

Time	
2.00 – 3.30 PM	Session I
3.30 PM to 3.45 PM	Tea Break
3.45 PM to 6.00 PM	Session II
6.00 PM to 6.15 PM	Tea Break
6.15 PM to 7.30 PM	Session III
7.30 PM to 8.30 PM	Dinner
8.30 PM to 10.00 PM	Session IV

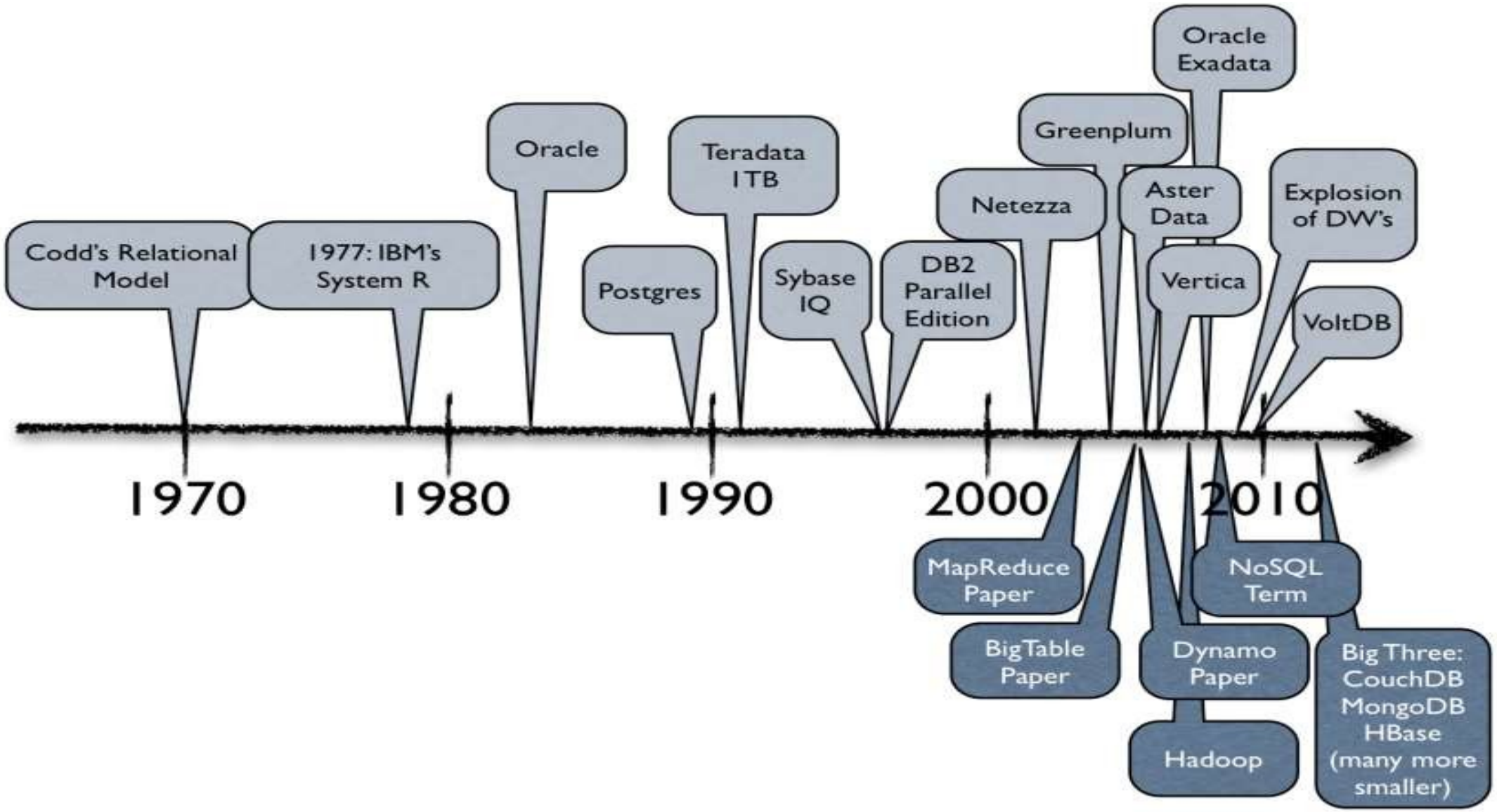
HP

An introduction to

NoSQL databases

HP

A brief history of databases



HP

Relational databases

- Benefits of Relational databases:
  - Designed for all purposes
  - ACID
  - Strong consistancy, concurrency, recovery
  - Mathematical background
  - Standard Query language(SQL)
  - Lots of tools to use with i.e: Reporting services, entity frameworks, ...
  - Vertical scaling (upscaling)

Object / Object-relational databases were not practical. Mainly because of Impedance mismatch

HP

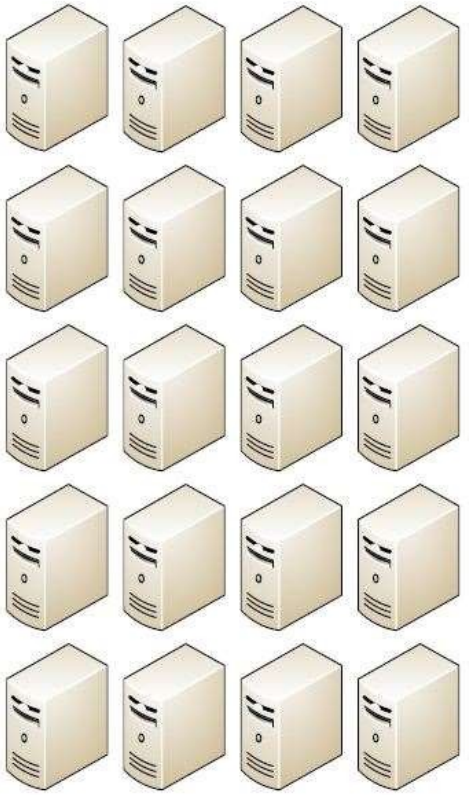
Era of Distributed Computing

But…

- Relational databases were not built for **distributed applications.**

Because…

- Joins are expensive
- Hard to scale horizontally
- Impedance mismatch occurs
- Expensive (product cost, hardware, Maintenance)



HP

Era of Distributed Computing

But…

- Relational databases were not built for **distributed applications.**

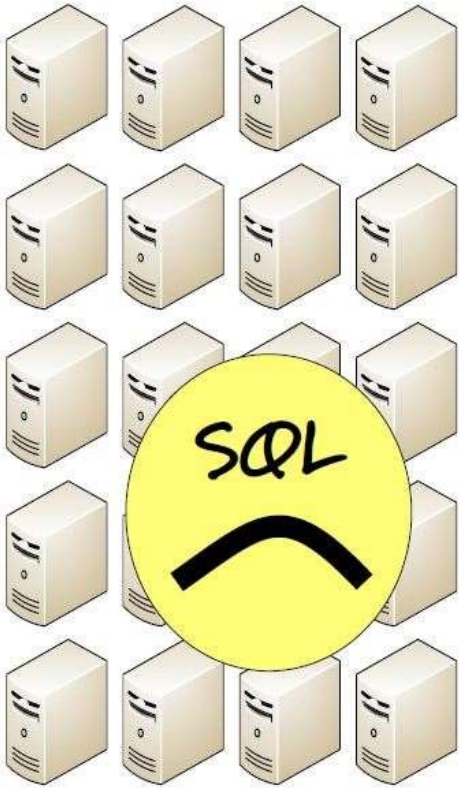
Because…

- Joins are expensive
- **Hard to scale horizontally**
- Impedance mismatch occurs
- Expensive (product cost, hardware, Maintenance)

And….

It's weak in:

- Speed (performance)
- High availability
- Partition tolerance

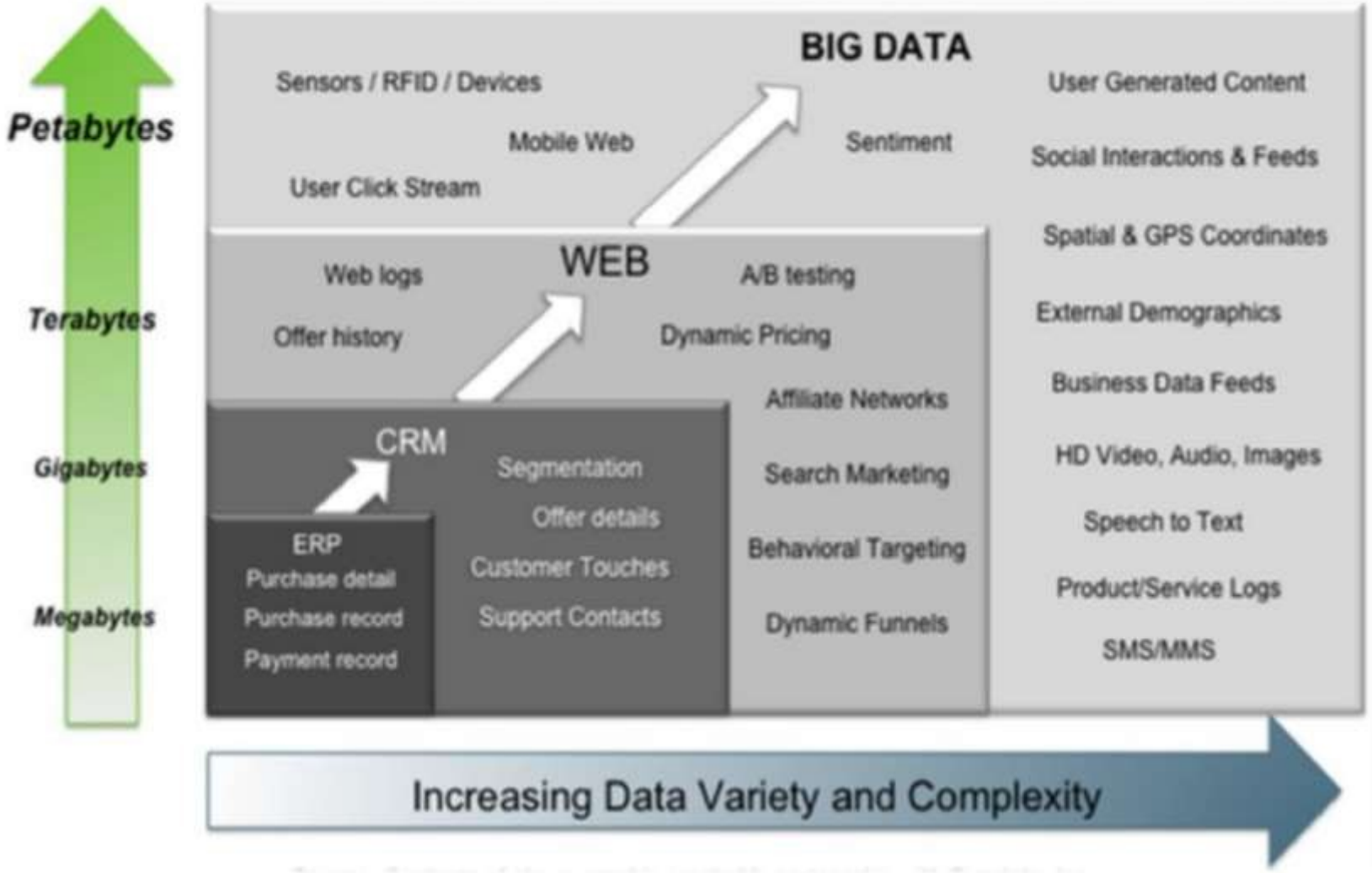


HP

Rise of Bigdata

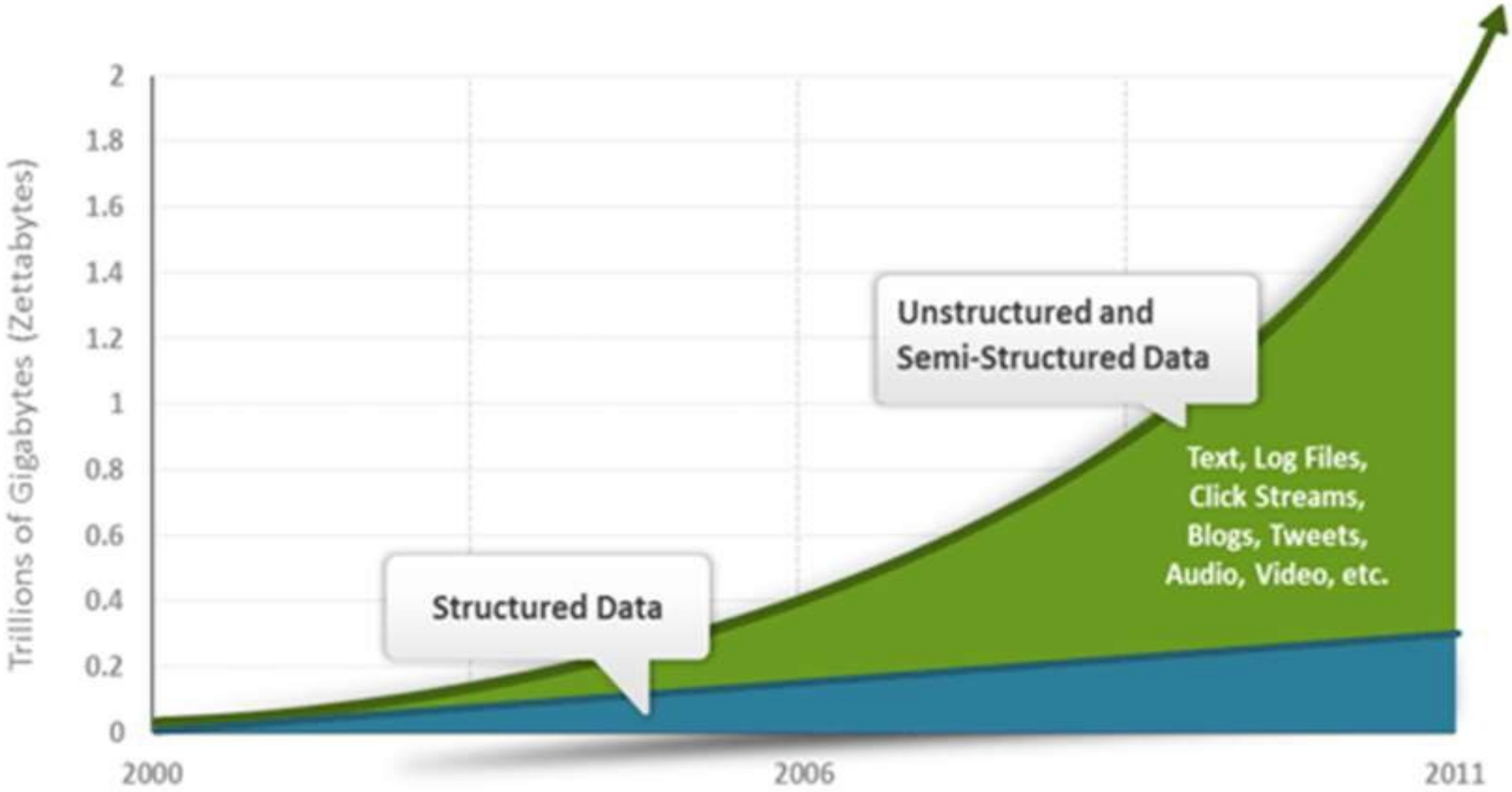
Three V(s) of Bigdata:

- Volume
- Velocity
- Variety



HP

Rise of Bigdata



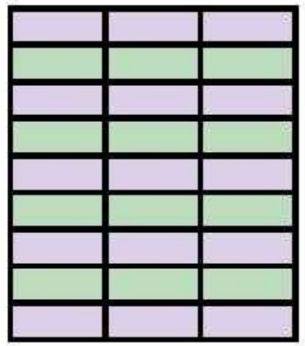
HP

Rise of Bigdata

- Wallmart: 1 million transactions per hour
- Facebook: 40 billion photos
- People are talking about petabytes today

Data Sources

were



will be

text

image

video

connections

HP

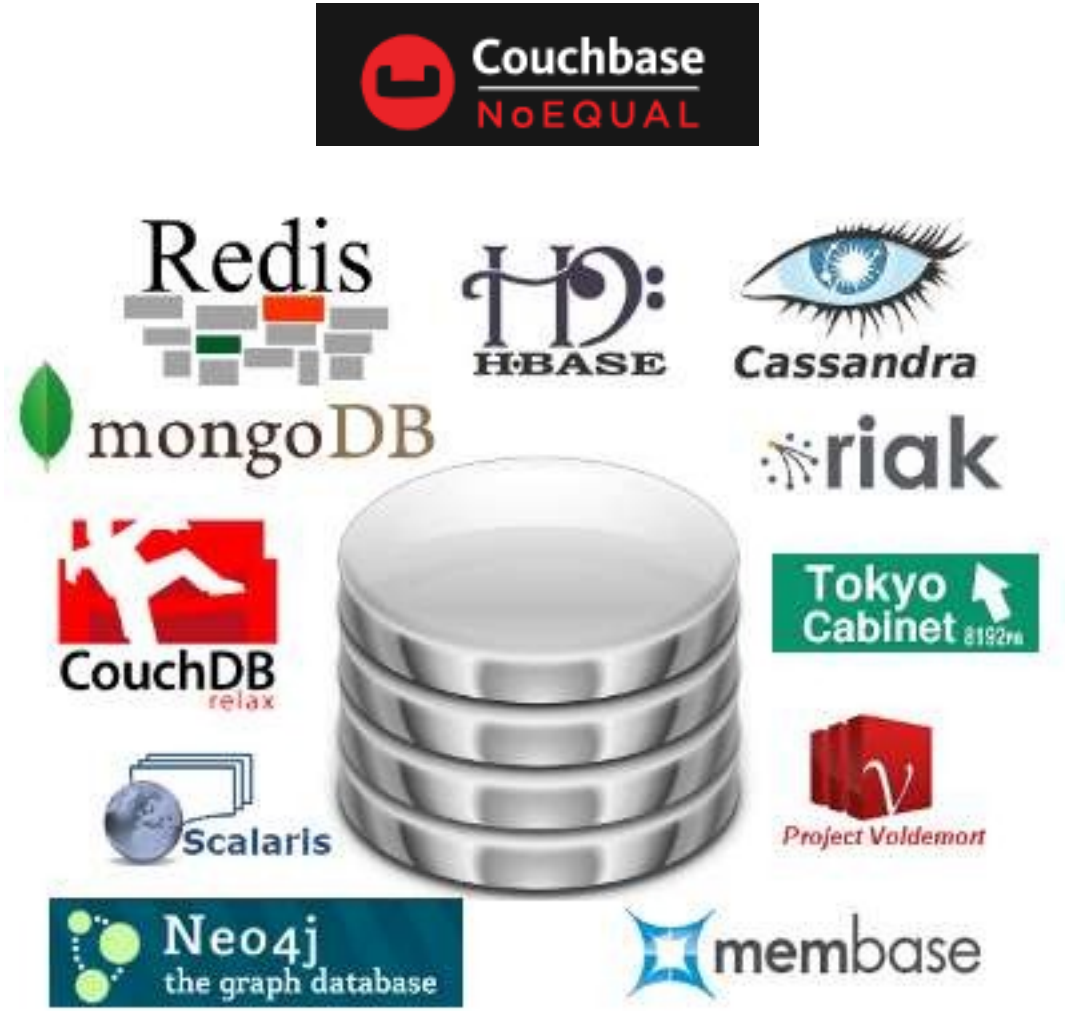
NoSQL why, what and when?

- Google & Amazon bulit their own databases (Big table & Dynamo)
- Facebook invented Cassandra and is using thousands of them
- #NoSQL was a twitter hashtag for a conference in 2009
- The name doesn't indicate its characteristics
- There is no strict defenition for NoSQL databases
- There are more than 150 NoSQL databases (nosql-database.org)

HP

Characteristics of NoSQL databases

- Non relational
- Cluster friendly
- Schema-less
- 21 century web
- Open-source



HP

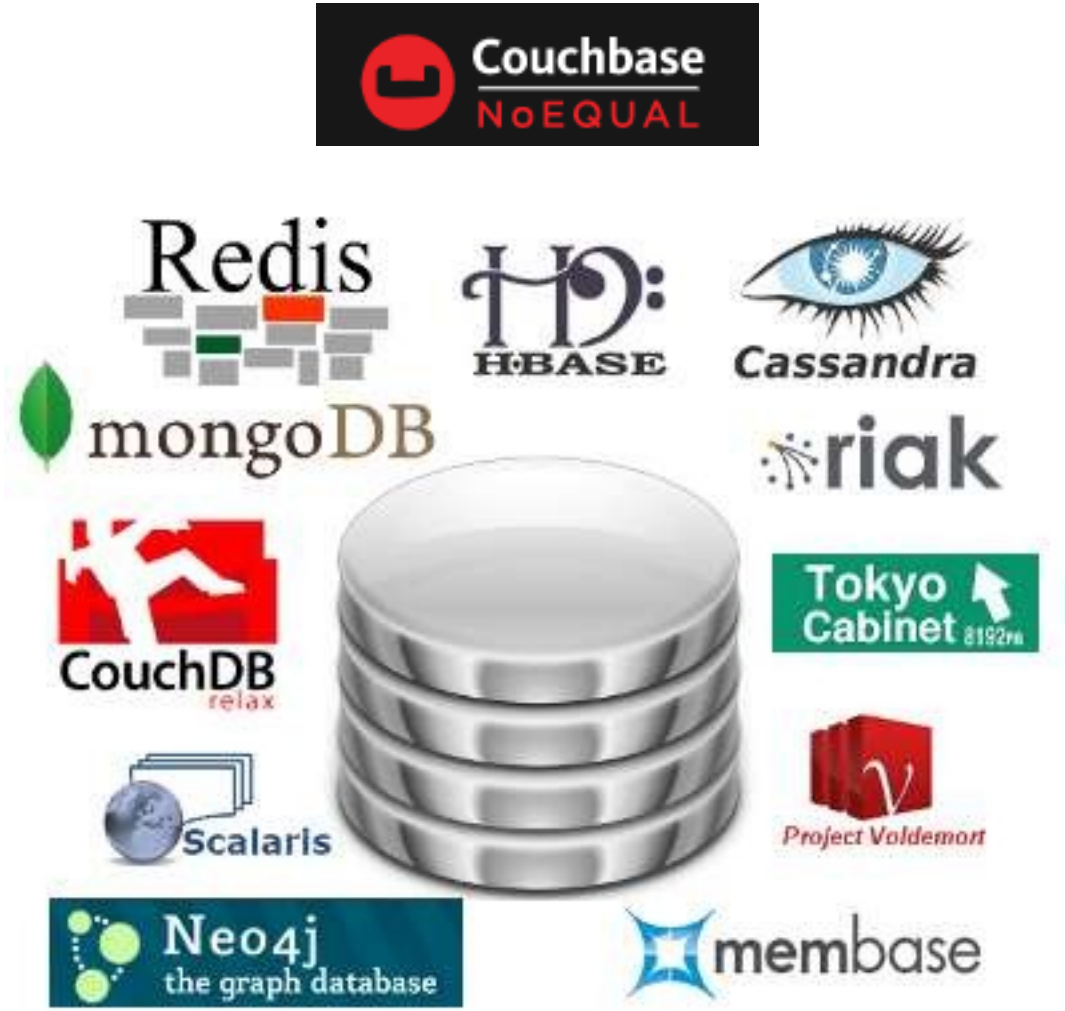
Characteristics of NoSQLdatabases

NoSQL avoids:

- Overhead of ACID transactions
- Complexity of SQLquery
- Burden of up-front schema design
- DBA presence
- Transactions (Itshould be handled at application layer)

Provides:

- Easy and frequent changes to DB
- Horizontal scaling (scaling out)
- Solution to Impedance mismatch
- Fast development



HP

Aggregate Data Models

NoSQL databases are classified in four major datamodels:

- Key-value
- Document
- Column family

- Graph

Each DB has its own query language

HP

Key-value data model

- The main idea is the use of a hash table
- Access data (values) by strings called keys
- Data has no required format – data may have any format
- Data model: (key, value) pairs
- Basic Operations:

  Insert(key,value), Fetch(key),Update(key), Delete(key)

Car	
Key	Attributes
1	Make: Nissan Model: Pathfinder Color: Green Year: 2003
2	Make: Nissan Model: Pathfinder Color: Blue Color: Green Year:2005 Transmission: Auto

HP

Key-value data model

- "Value" is stored as a "blob"
  - Without caring or knowing what is inside
  - Application is responsible for understanding the data

- Main observation from Amazon (using **Dynamo**)
  - "There are many services on Amazon's platform that only need primary-key access to a data store."

  E.g. Best seller lists, shopping carts, customer preferences, session management, sales rank, product catalog

HP

Column family data model

- The column is lowest/smallest instance of data.
- It is a tuple that contains a name, a value and a timestamp

ColumnFamily: Authors	
Key	Value
"Eric Long"	Columns: Name = Value; "email" = "eric (at) long.com"; "country" = "United Kingdom"; "registeredSince" = "01/01/2002"
"John Steward"	Columns: Name = Value; "email" = "john.steward (at) somedomain.com"; "country" = "Australia"; "registeredSince" = "01/01/2009"
"Ronald Mathies"	Columns: Name = Value; "email" = "ronald (at) sodeso.nl"; "country" = "Netherlands, The"; "registeredSince" = "01/01/2010"

HP

Column family data model

Some statistics about Facebook Search (using **Cassandra**)

- MySQL >50 GB Data
  - Writes Average : ~300 ms
  - Reads Average : ~350 ms
- Rewritten with Cassandra >50 GB Data
  - Writes Average : 0.12 ms
  - Reads Average : 15 ms

HP

Graph data model

- Based on Graph Theory.
- Scale vertically, no clustering.
- You can use graph algorithms easily
- Transactions
- ACID

Neo4j
the world's
leading graph database

HP

Document-based datamodel

- Usually JSON like interchange model.
- Query Model: JavaScript-like or custom.
- Aggregations: **Map/Reduce**
- Indexes are done via B-Trees.
- unlike simple key-value stores, both keys and values are fully searchable in document databases.

```
{
  person: {
    first_name: "Peter",
    last_name: "Peterson",
    addresses: [
      {street: "123 Peter St"},
      {street: "504 Not Peter St"}
    ],
  }
}
```

Couchbase NoEQUAL

mongoDB

HP

What we need?

- We need a distributed database system having such features:
  - **Fault tolerance**
  - **High availability**
  - **Consistency**
  - **Scalability**

Which is impossible!!!

According to CAP theorem

HP

Should we...?

- In some cases getting an answer quickly is more important than getting a correct answer
- By giving up ACID properties, one can achieve higher performance and scalability.
- Any data store can achieve Atomicity, Isolation and Durability but do you always need **consistency**?
- Maybe we should implement Asynchronous Inserts and updates and should not wait for confirmation?

HP

BASE

Almost the opposite of ACID.

- Basically available: Nodes in the a distributed environment can go down, but the whole system shouldn't be affected.
- Soft State (scalable): The state of the system and data changesover time.
- Eventual Consistency: Given enough time, data will be consistent across the distributed system.

HP

BASE vs ACID

ACID:

- Strong consistency.
- Less availability.
- Pessimistic concurrency.
- Complex.

BASE:

- Availability is the most important thing. Willing to sacrifice for this (CAP).
- Weaker consistency (Eventual).
- Best effort.
- Simple and fast.
- Optimistic.

HP

CAP theorem

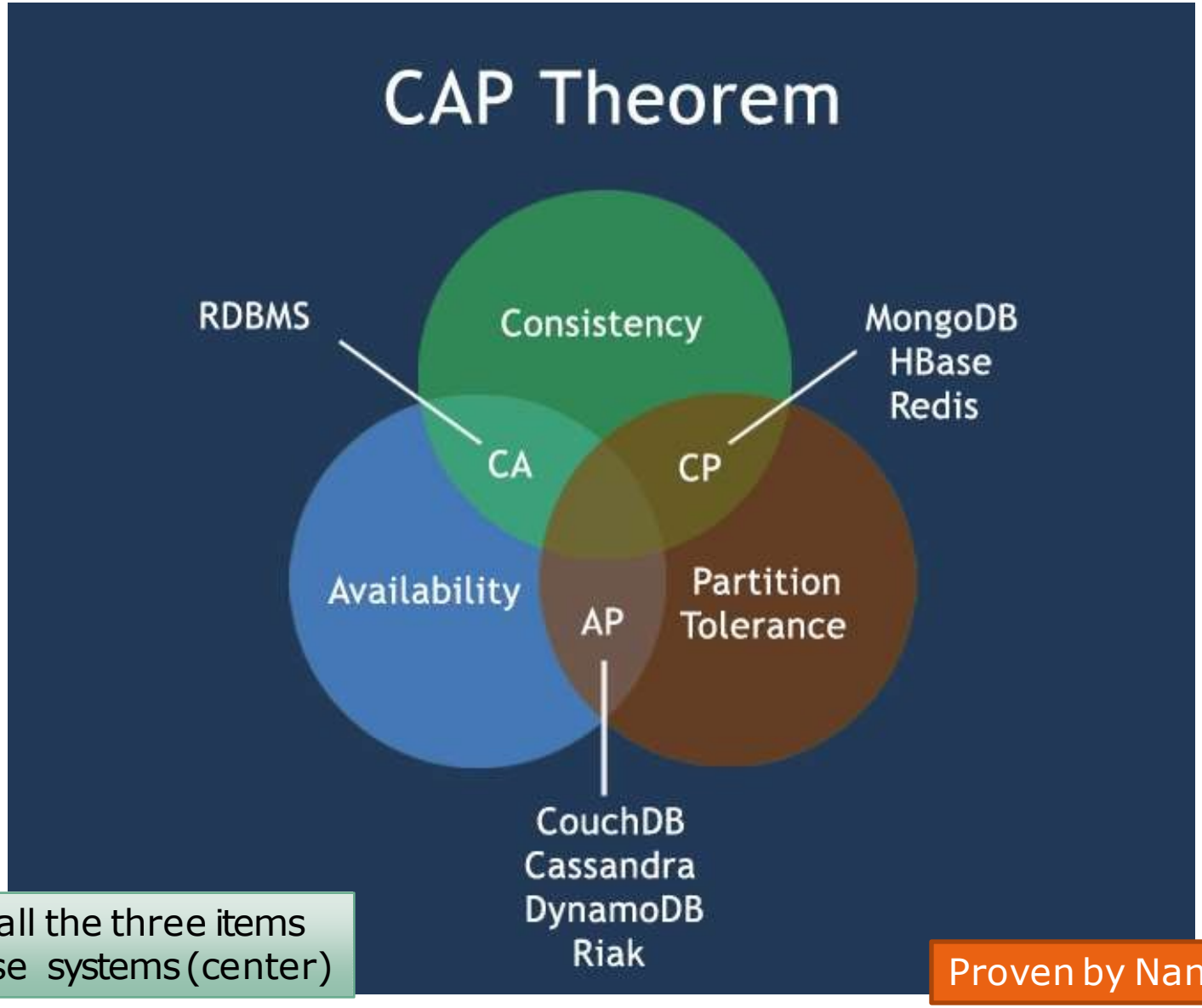
- **C**onsistency: **Clients should read the same data**. There are many levels of consistency.
  - Strict Consistency –RDBMS.
  - **Tunable Consistency – Cassandra.**
  - **Eventual Consistency – Mongodb.**
- **Availability**: Data to be available.
- **P**artial Tolerance: Data to be partitioned across network segments due to network failures.

2000 Prof. Eric Brewer, PoDC Conference Keynote

2002 Seth Gilbert and Nancy Lynch, ACM SIGACT News 33(2)

> *Of three properties of shared-data systems - data **C**onsistency, system **A**vailability and tolerance to network **P**artitions - only two can be achieved at any given moment in time.*

CAP theorem in different SQL/NoSQL databases

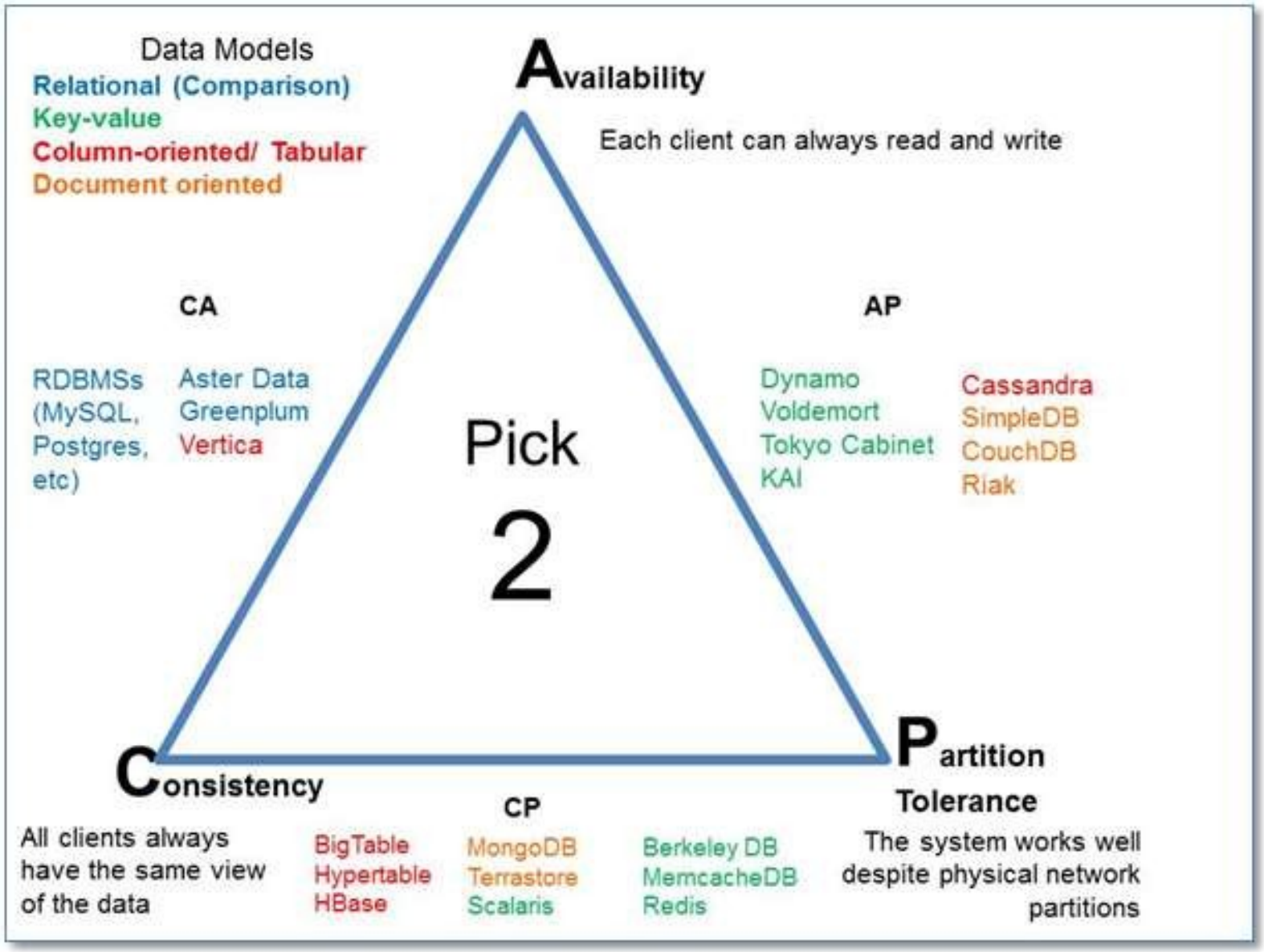


We can not achieve all the three items In distributed database systems (center)

Proven by Nancy Lynch et al. MIT labs.

HP

Which data model to choose

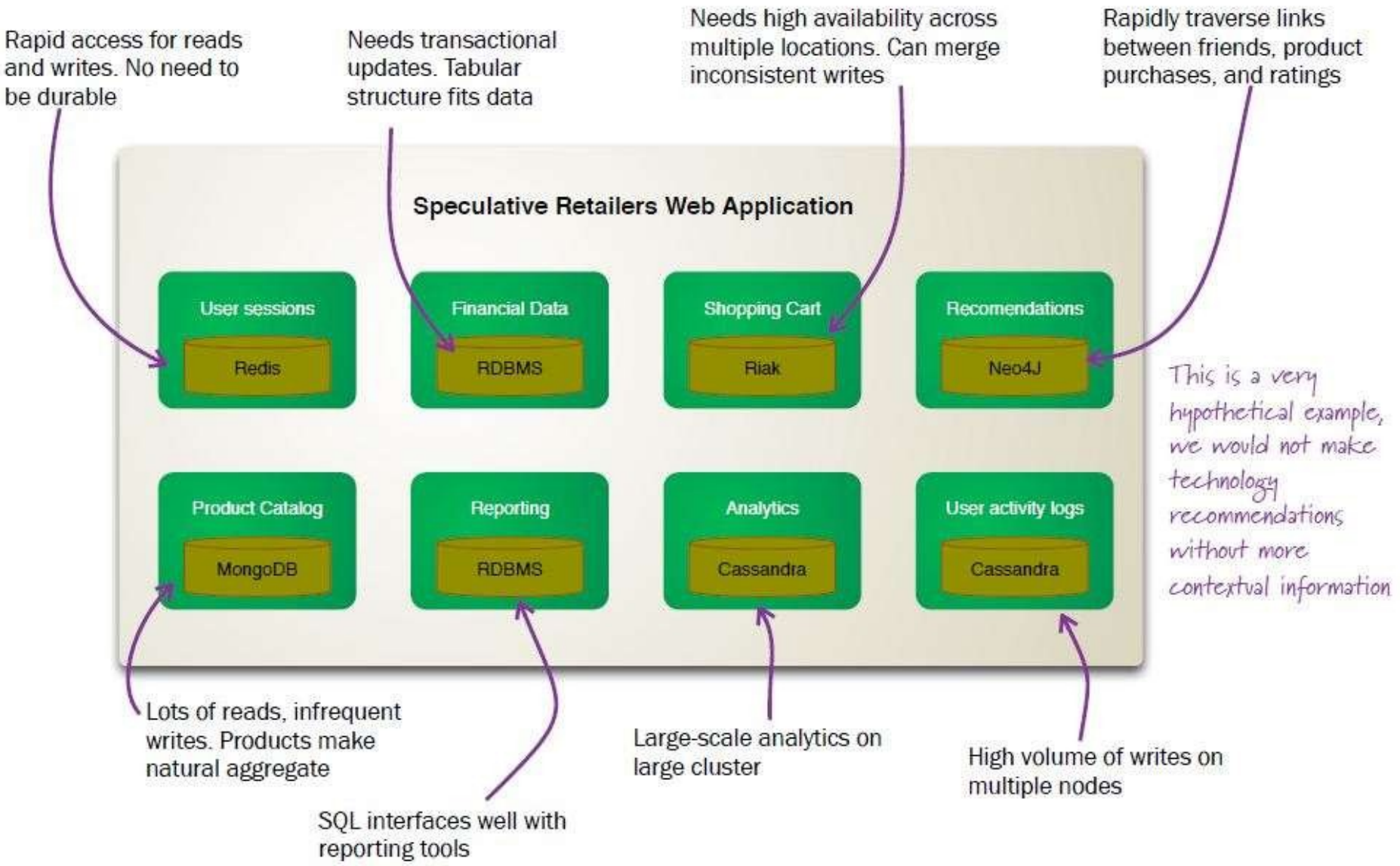


Polyglot persistence : the future of database systems

- Future databases are the combination of SQL & NoSQL
- We still need relational databases

HP

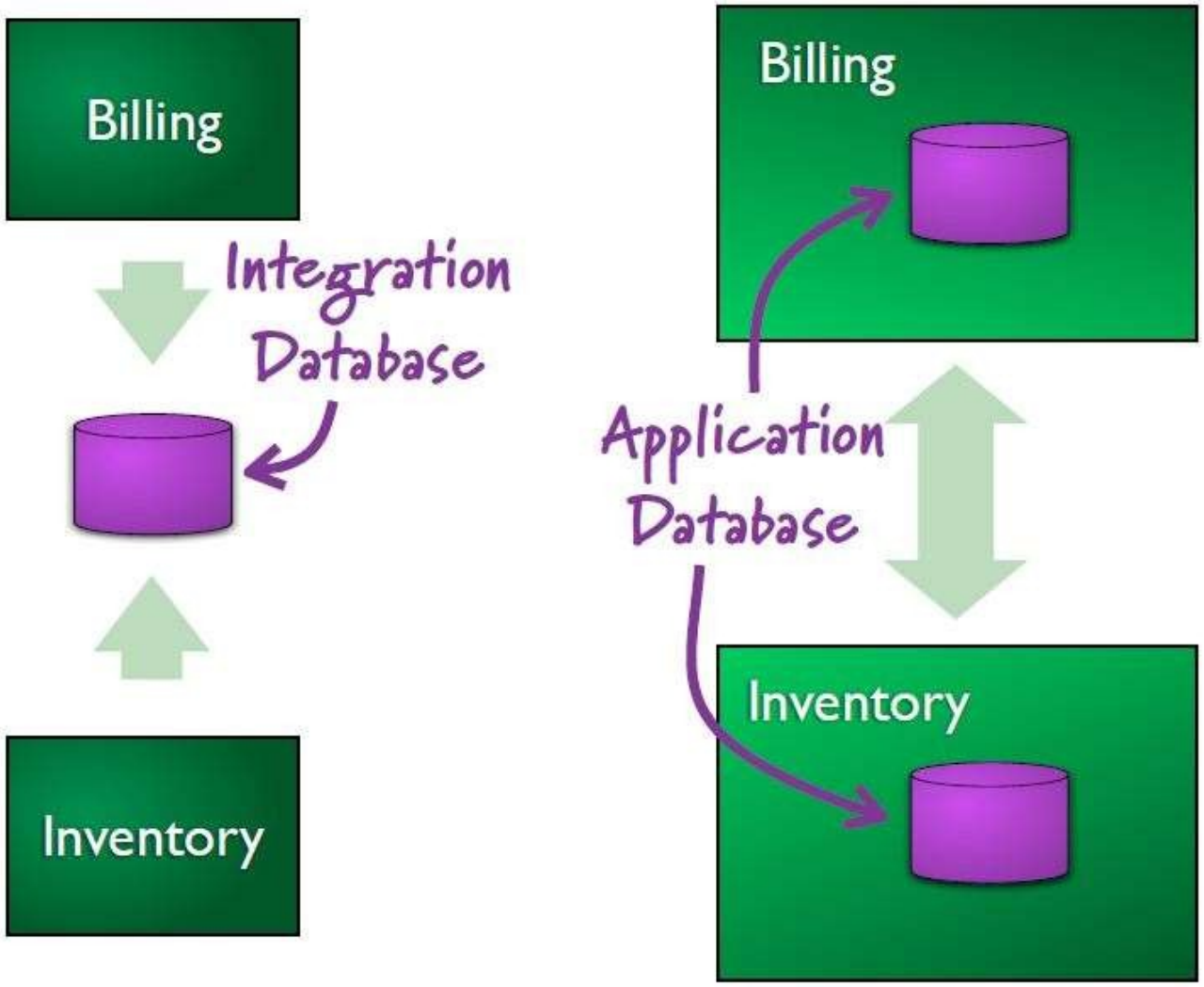
Overview of a polygot db



HP

New approach to database systems:

- Integrated databases has its own advantages and disadvantages
- With the advent of webservices it seems now it's the time to switch to **decentralized data bases**
- Single point of failure, Bottlenecks would be avoided
- Clustering & replication would be much easier



Conclusion:

Before you choose NoSQL as a solution:

Consider these items, ...

- Needs a precise evaluation, Maybe NoSQL is not the right thing
- Needs to read lots of case study papers
- Aggregation is totally a different approach
- NoSQL is still immature
- Needs lots of hours of studing and working to expert in a particular NoSQL db
- There is no standard query language
- Most of controls have to be implemented at the application layer
- Relational databases are still the strongest in transactional environments and provide the best solutions in consistancy and concurrency control

Conclusion

Before you choose NoSQL as a solution: