AN INTRODUCTION TO BIG DATA

Master Program in Computer Science
University of Calabria

Prof. F. Ricca

Let’s start from few citations

Data creation is exploding.
With all the selfies and useless files people refuse to delete on the cloud. . . . The world’s data storage capacity will be overtaken. . . . Data shortages, data rationing, data black markets . . . data-geddon!

Gavin Belson, HBOs Silicon Valley

Data is the new oil

Clive Humby, UK Mathemetician, 2006

…

Qi Lu, the chief of Microsoft’s Applications and Services, 2016

Let’s start from few citations

Information is the oil of the 21st century, and analytics is the combustion engine

Peter Sondergaard, Gartner Research, 2011

A relational database is like a garage that forces you to take your car apart and store the pieces in little drawers

Anonymous

An finally SQL is back!

Anonymous

Is there a definition of bigdata?

- *No unique definition!*
- *A buzzword often misused*

Ste 5 parole sono importanti

"Data whose volume, variety, velocity of production, and complexity require new architectures, techniques, algorithms and analytics to manage it and extract value and hidden knowledge from it"

Aggiungi qui il testo

The famous “Vs”

- The Vs of Bigdata
  - Volume: scale of data Assai dati
    Data volume is increasing ..up to ZB!
  - Variety: different forms of data
    Text, images, numerical values, etc.

    Tipi diversi di dati. Multiformato
  - Velocity: speed of production and elaboration
    e.g., streaming data, logs

    Non mi è molto chiaro, va un attimo visto onlines
  - Veracity: uncertainty and imprecision of data → Quality in termini di real world quality, non di pulizia dei dati
  - Value: exploit intrinsic value by data
    - to create business advantage, thus
    - *need for strong analytics and reasoning* → *Data Science*

Il valore finale che danno i dati.

WHAT IS BIGDATA?

Let’s try to answer from a historical perspective

Three revolutions in data bases

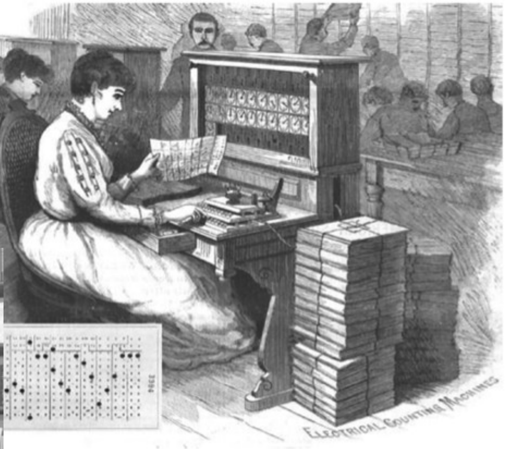
The main factors driving the main changes were:

1. The emergence of the electronic computer
2. The emergence of the relational database
3. The need of global scope and continuous availability

Emergenze come nascita e arrivo

Three revolutions in data bases

- The term database dates back to the late 1960s
- But, *collecting and organizing data has been an integral factor in the development of human civilization and technology*



L'esempio umano di database erano archivi e librerie

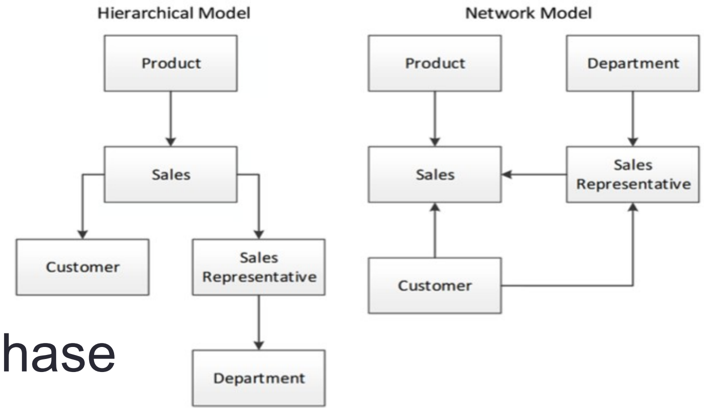
Three revolutions in data bases

- *The emergence of electronic computers following the Second World War ignited the first revolution in databases*
  - Direct high-speed access to individual records became possible in in the mid-1950s
  - ISAM (Index Sequential Access Method) made fast record-oriented access feasible
  - The birth of the first OLTP (On-line Transaction Processing) computer systems
    - These were completely under the control of the application
    - No *Database Management Systems* (*DBMS*)

Non c'erano software. Erano interessate le banche sopratutto

First generation databases

- The first-generation databases ran exclusively on the mainframe computers (largely IBM mainframes)
- Two competing data models emerged in early 70-ties:
  - *Network model* (CODASYL standard)
  - *Hierarchical model* (somewhat simpler approach)
- "Navigational" in nature
  - Navigate using pointers or links
  - Dominated up until the late 1970s
- Extremely inflexible
  - Queries anticipated during the design phase
  - Complex analytic queries required complex coding
- The golden era of Cobol programmers!



Prima si pensava alla query e poi si faceva lo schema per fare in modo che la query fosse veloce

The Second Database Revolution

The main issues:

1. Existing databases were too hard to use
   - only for people with specialized programming skills
2. Lacked a solid mathematical foundation
   - no logical consistency, nor ability to deal with missing information
3. Mixed logical and physical implementations
   - physical storage incomprehensible to nontechnical users

- In 1970 Codd published

  "A Relational Model of Data for Large Shared Data Banks"

  - defined the *relational database model*
  - the most significant—almost universal—model for database systems for a generation

assurdo

The Second Database Revolution

- The relational model by Codd
  - Clearly presents data to the user
  - Does not mention how it should be stored on disk or in memory
  - Solid mathematical foundations
  - Levels of conformance via "normal forms"

Separazione logical view e physical

enforce data quality

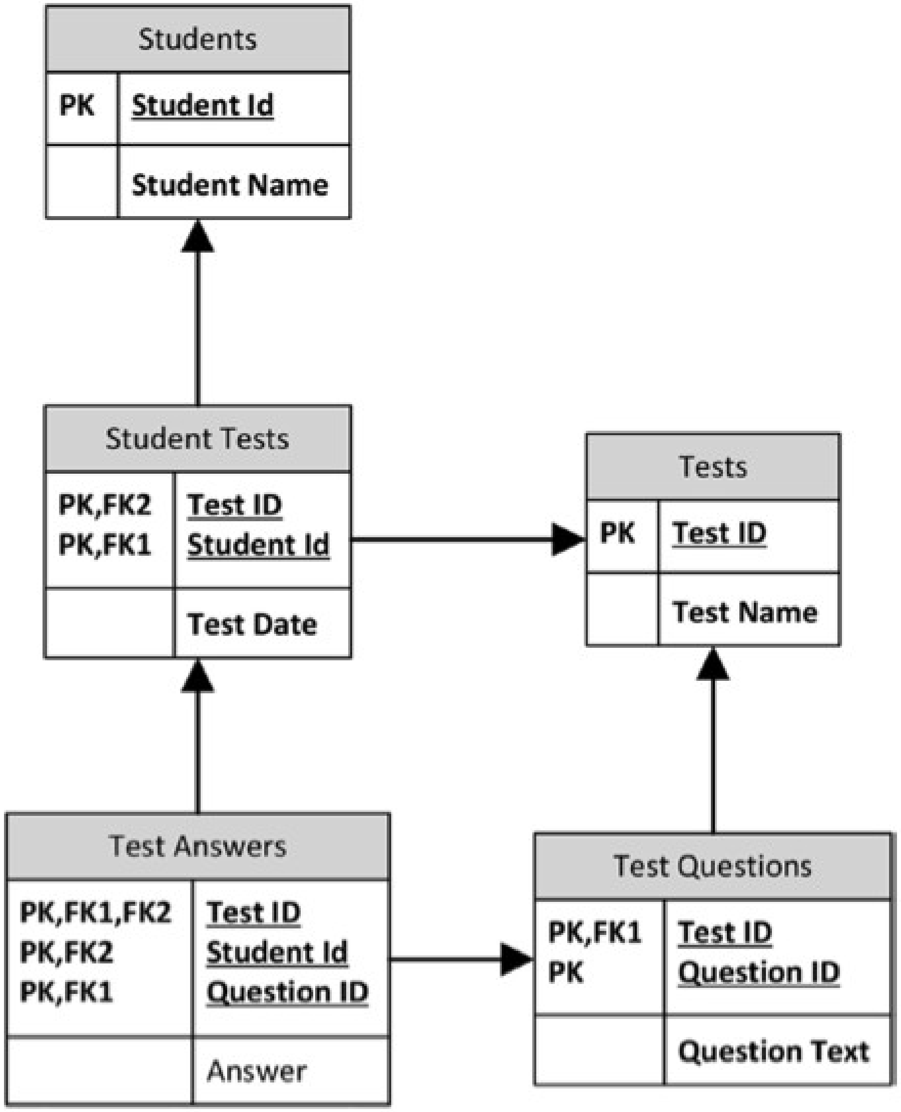
- Concurrent data change requests → transactions
  - ensure consistency and integrity of data
  - *A transaction is a transformation of state which has the properties of atomicity (all or nothing), durability (effects survive failures) and consistency (a correct transformation).*
  - ACID transactions: Atomic, Consistent, Isolated, and Durable.
    - Model defined by Jim Gray in the late 1970s
  - the standard for all serious database implementations

Per evitare problemi di race condition

Un-normalized data

Test scores	
	Student Name Test Name Test Date Answer 1 Answer 2 Answer 3 Answer 4 Answer 5 Answer 6 Answer N

Normalized data



Students	
PK	**Student Id**
	Student Name

Student Tests	
PK,FK2 PK,FK1	**Test ID** **Student Id**
	Test Date

Tests	
PK	**Test ID**
	Test Name

Test Answers	
PK,FK1,FK2 PK,FK2 PK,FK1	**Test ID** **Student Id** **Question ID**
	Answer

Test Questions	
PK,FK1 PK	**Test ID** **Question ID**
	Question Text

Relational DBMSs

- Initially vendors including IBM did not like the idea
  - Can a relational DB deliver adequate performance?
- IBM initiate proved it in 1974 with System R
  - it pioneered the *SQL language*

Minicomputer è inteso come server, al tempo

- On the hardware side minicomputers replaced mainframes in the 80-ties

Importante brother

- In 10 about years many new DBMSs were introduced
  - E.g., Oracle, Sybase, SQL Server, Informix, MySQL, and DB2
- Today a Relational DBMS means:
  - Relational data model + ACID transactions + SQL

La combinazione di questi è quello che oggi si intende con DBMS

- Dominating technology, unchallenged until the latter half of the 2000s

Relational DBMS

- More than 30 years of commercial dominance!
- A triumph of computer science and software engineering
  - Strong theoretical foundations
  - Data independent of the physical storage implementation
  - ACID transaction model
  - Flexible query mechanisms that do not require sophisticated programming skills

non serve
Cobol

The *unlucky* OODBMS

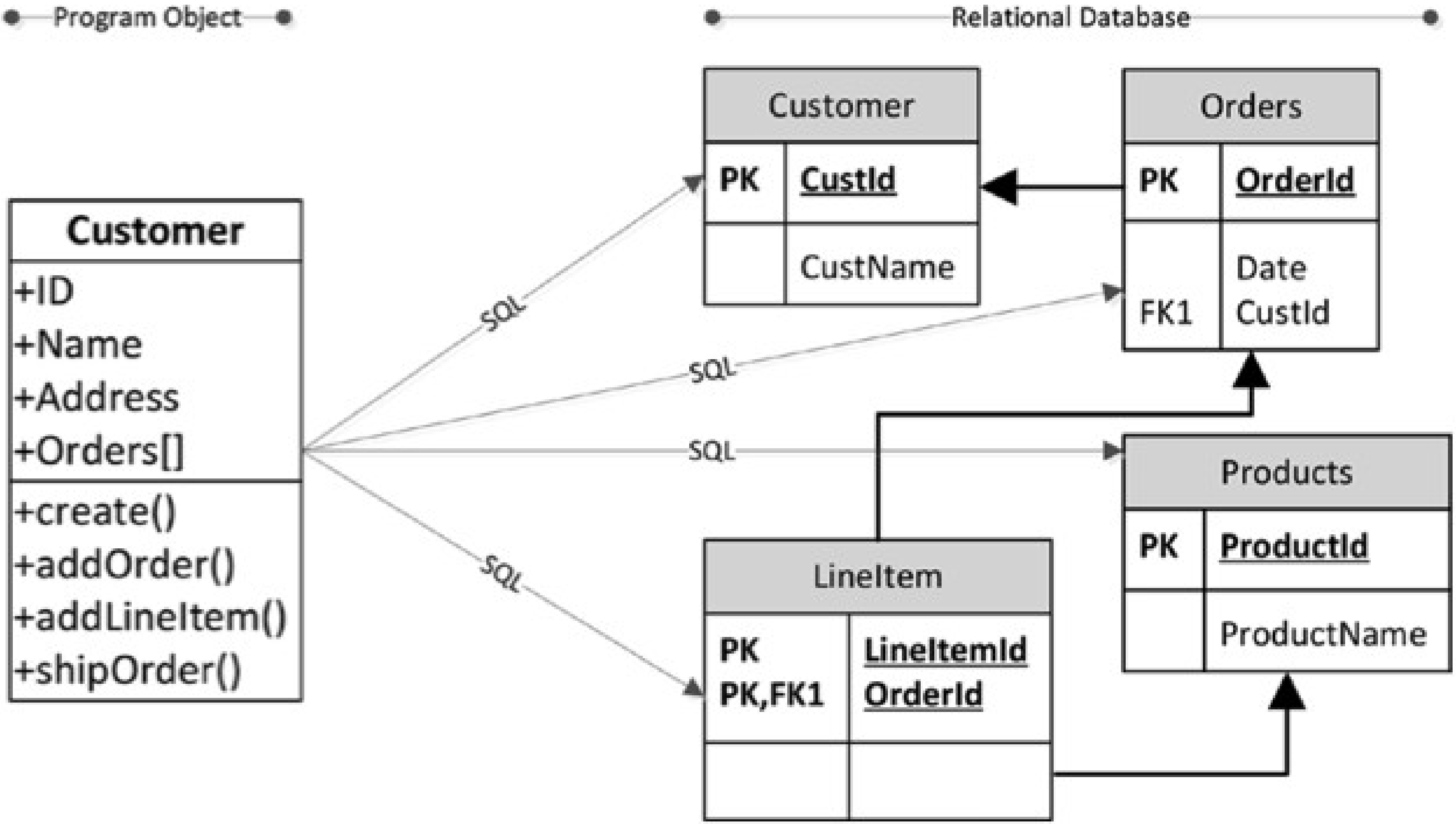
mai funzionato lol

C'è da dire che prima nei 90s c'erano solo OO languages

- Another significant paradigm shift impacted mainstream application-development languages:
  - *Object-oriented (OO) programming*
    - Encapsulation: An object class encapsulates both data and actions (methods) that may be performed on that data
    - Inheritance: Object classes can inherit the characteristics of a parent class
- The “impedance mismatch”

Impedance Mismatch: non comunicano per niente bene tra di loro

  - The first serious challenge to the relational database
  - Various differences in the models
    - From identity, to navigation, to association maintenance, etc.
  - Did not match current technology
  - Alleviated by Object Relational Matching (ORM)



Program Object

Relational Database

Customer
+ID +Name +Address +Orders[]

+create()
+addOrder()
+addLineItem()
+shipOrder()

SQL

SQL

SQL

SQL

Customer	
PK	**CustId**
	CustName

Orders	
PK	**OrderId**
 FK1	Date CustId

Products	
PK	**ProductId**
	ProductName

LineItem	
PK **PK,FK1**	**LineItemId** **OrderId**

The Third Database Revolution

then something just snapped....

- *No significant new databases were introduced for about 10 years (1995–2005)*

E da qua nascono i social networks che cambiano il paradigma

- The era of massive web-scale applications begins
  - RDBMS demonstrate scalability limits, and high costs
    - Scale-up vs scale-out

Non bastavano le infrastrutture del tempo

Scale out: significa che metti insieme più Computer per gestire grandi carichi di dati e interazioni

  - Google had to invent new hardware and software architectures
  - Amazon needed transactional processing capability that could operate at massive scale

Con ACID: 1 solo può scrivere. Non va bene.

  - MySpace and eventually Facebook faced similar challenges in scaling their infrastructure from thousands to millions of users

  - *Oracle could not provide sufficient scalability*

New database designs emerge

Sharding = dividere i dati in più fonti e avere un layer sopra che gestisce questa divisione. Utile in caso di problemi.

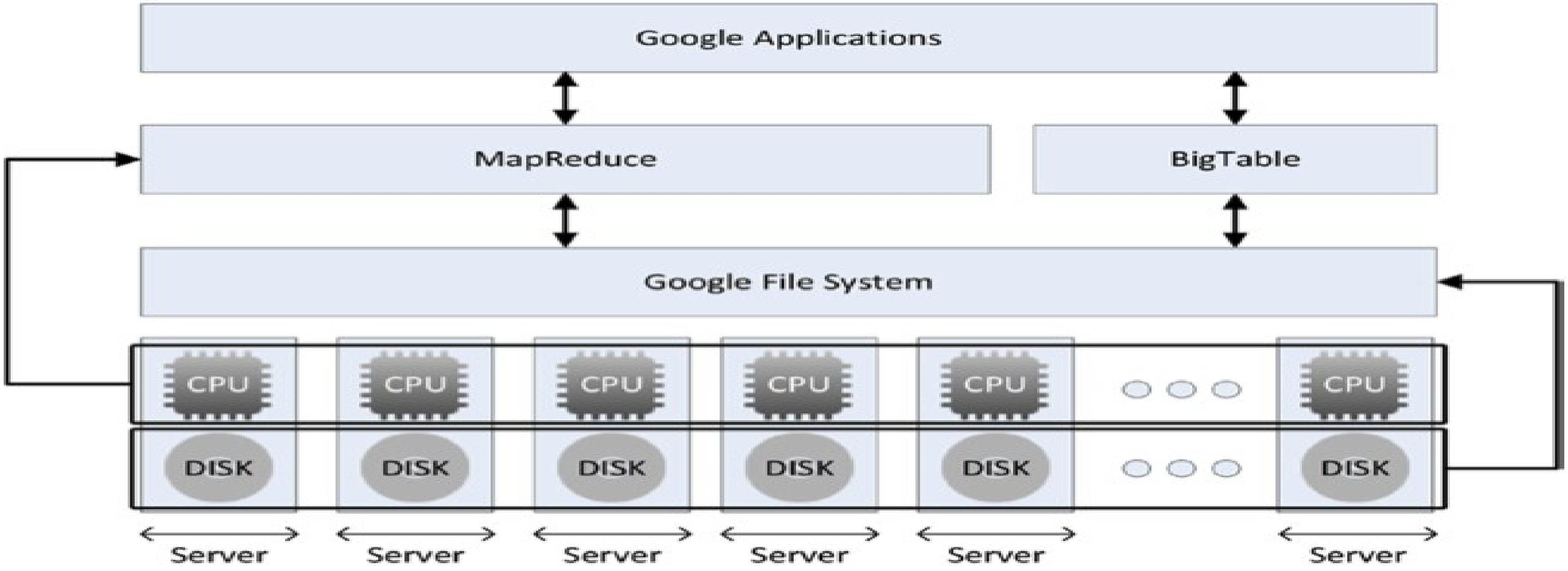
- Sharding (Facebook/Twitter) involves partitioning the data across multiple databases → ACID transactions are lost
- Amazon developed an alternative to strict ACID → key-value store (DynamoDB)
- Programmers unhappy with the impedance mismatch → Document databases (CouchBase and MongoDB)
  - Enabled by AJAX and diffusion of JSON
- NoSQL and NewSQL *“the Nonrelational Explosion”*
  - H-Store described a pure in-memory distributed database
  - C-Store specified a design for a columnar database

Google solutions

- In 2003, Google revealed
  - the distributed file system GFS
  - the parallel processing algorithm MapReduce
  - its BigTable distributed structured database

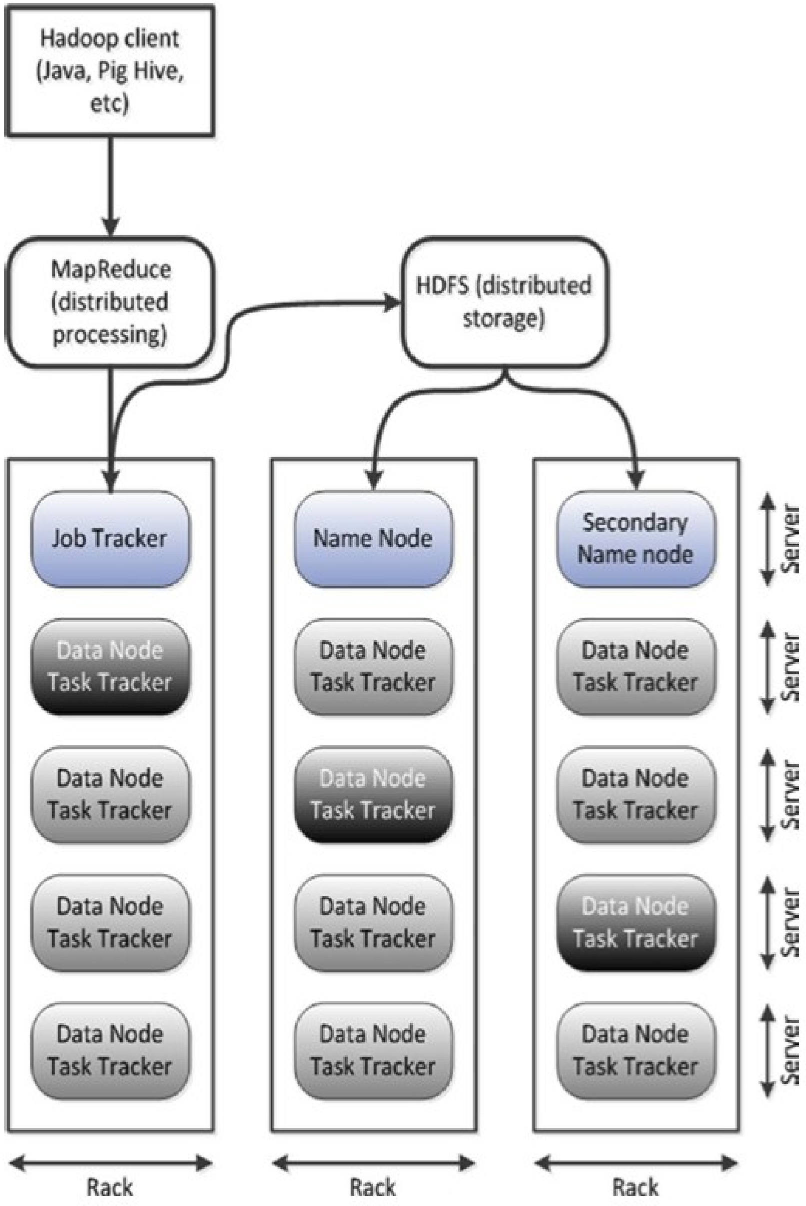
GFS: google file systems

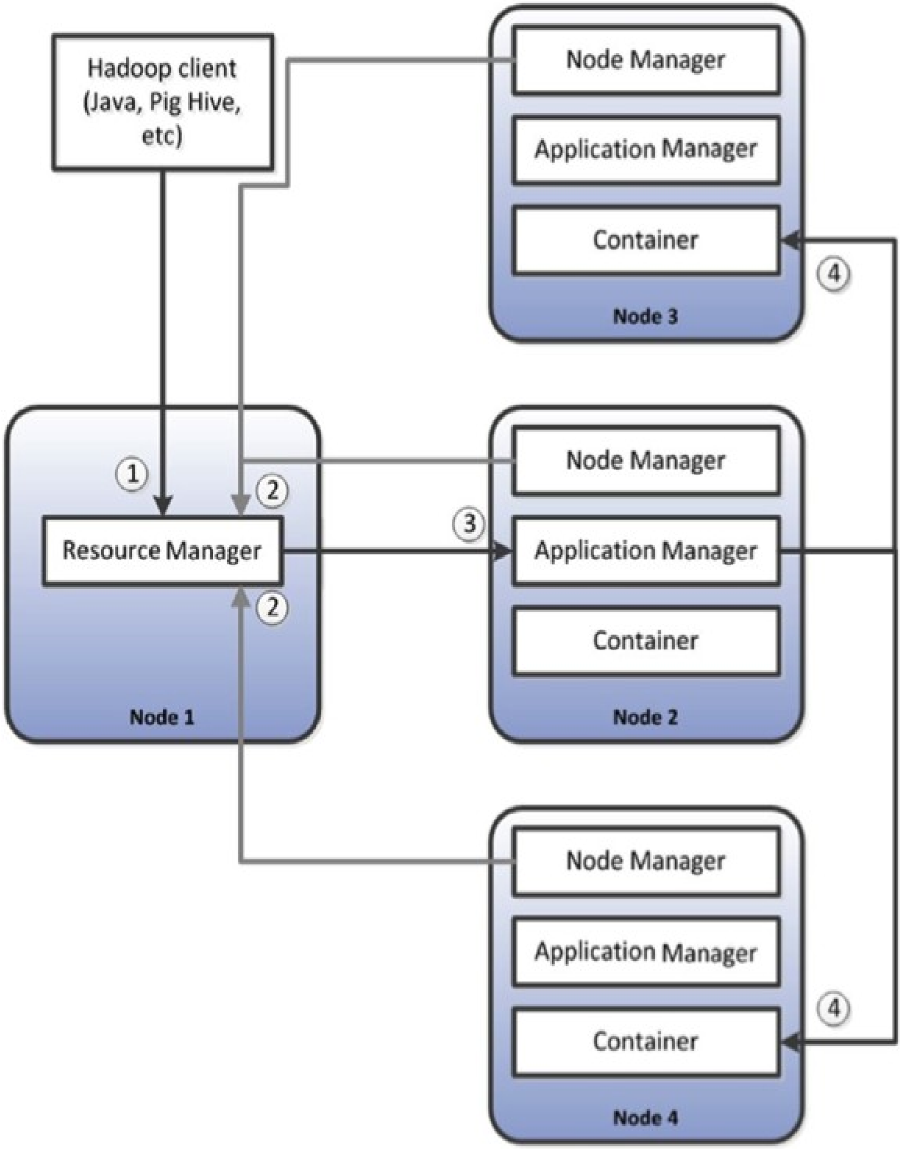
BigTable è NON RELAZIONALE

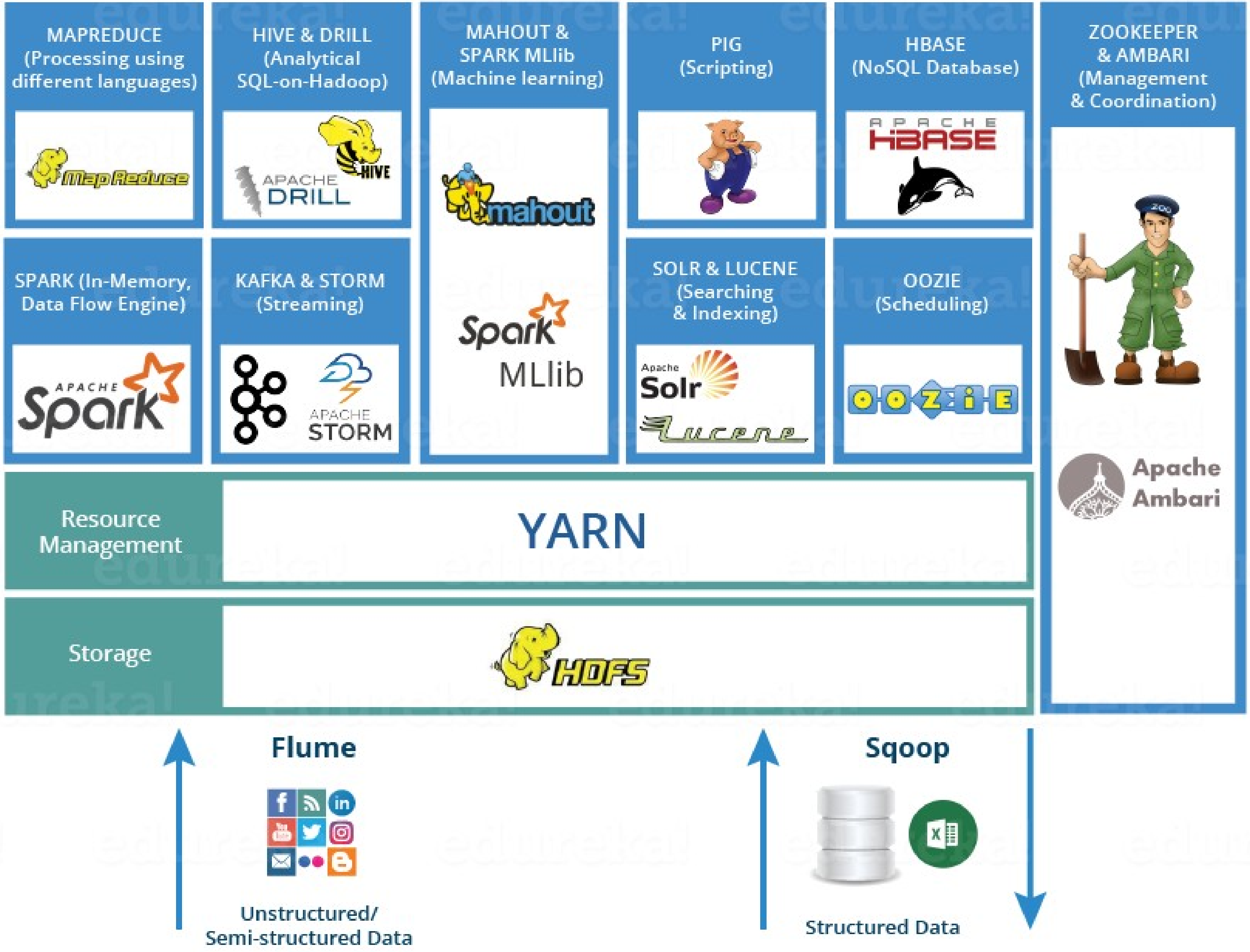


Hadoop ecosystem

- The Hadoop project (by Yahoo!)
  - Opens source inspired to Google solutions
  - a rapid uptake from 2007 on become an ecosystem
  - a technology enabler for Big Data
  - the de facto solution for massive unstructured data
- Hadoop
  - *Hadoop Distributed File System* (*HDFS*)
  - *YARN* (*Yet Another Resource Negotiator)*
- *Hadoop ecosystem*
  - Hbase, Hive, Pig, Sqoop, Spark, Mahout, etc.
- Spark
  - in-memory, distributed, fault-tolerant processing framework
  - Implemented in Scala, higher-level than MapReduce, no IO bottlenecks


Hadoop client (Java, Pig Hive, etc)
MapReduce (distributed processing)
HDFS (distributed storage)
Job Tracker
Name Node
Secondary Name node
Data Node Task Tracker
Data Node Task Tracker
Data Node Task Tracker
Data Node Task Tracker
Data Node Task Tracker
Data Node Task Tracker
Data Node Task Tracker
Data Node Task Tracker
Data Node Task Tracker
Data Node Task Tracker
Data Node Task Tracker
Data Node Task Tracker
Server
Server
Server
Server
Server
Rack
Rack
Rack


Hadoop client (Java, Pig Hive, etc)
Node Manager
Application Manager
Container
Node 3
4
1
2
Resource Manager
2
Node 1
Node Manager
3
Application Manager
Container
Node 2
Node Manager
Application Manager
4
Container
Node 4



MAPREDUCE (Processing using different languages)

HIVE & DRILL (Analytical SQL-on-Hadoop)

MAHOUT & SPARK MLlib (Machine learning)

PIG (Scripting)

HBASE (NoSQL Database)

ZOOKEEPER & AMBARI (Management & Coordination)

SPARK (In-Memory, Data Flow Engine)

KAFKA & STORM (Streaming)

SOLR & LUCENE (Searching & Indexing)

OOZIE (Scheduling)

Resource Management

YARN

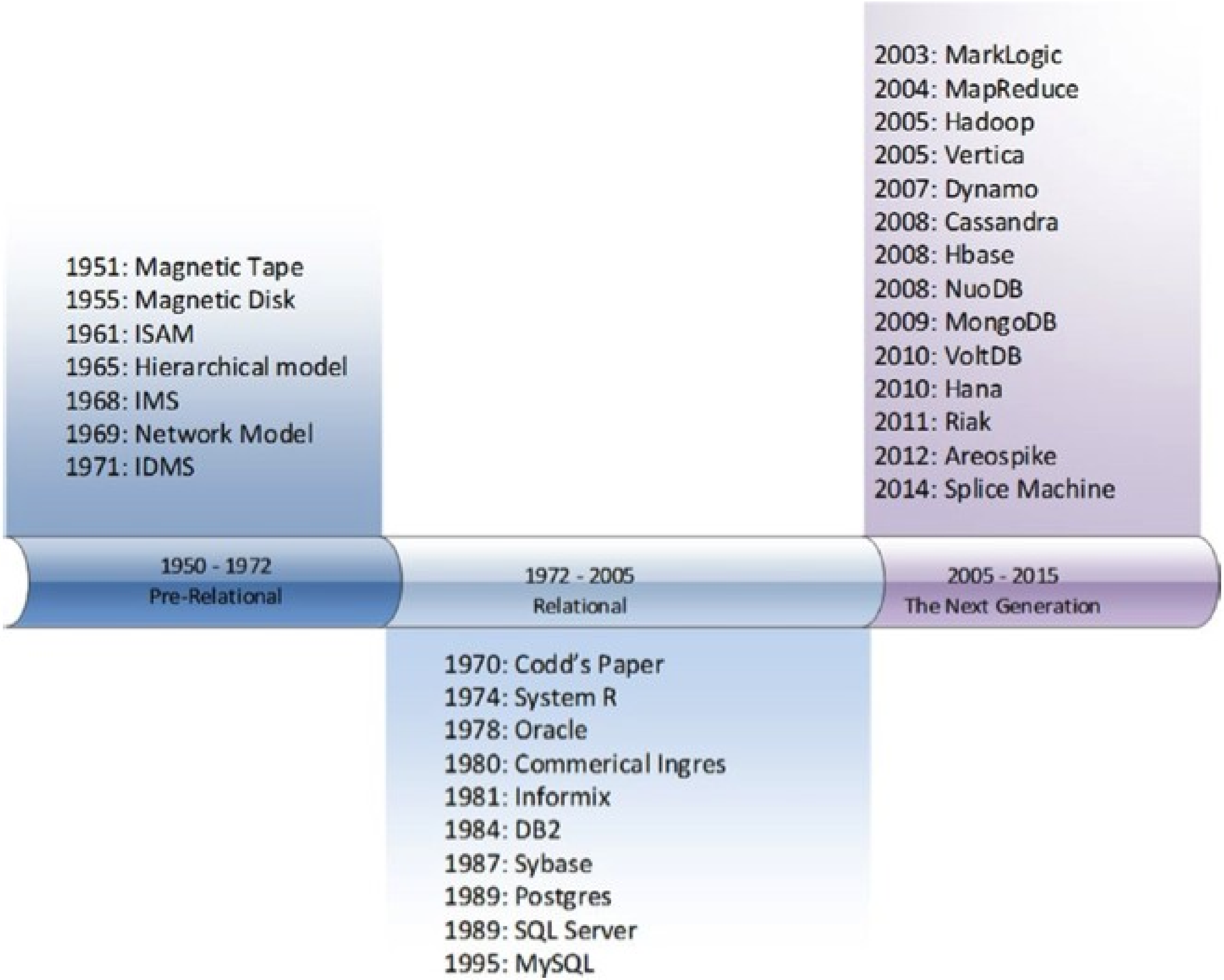
Storage

Flume

Unstructured/ Semi-structured Data

Sqoop

Structured Data



1950 - 1972
Pre-Relational

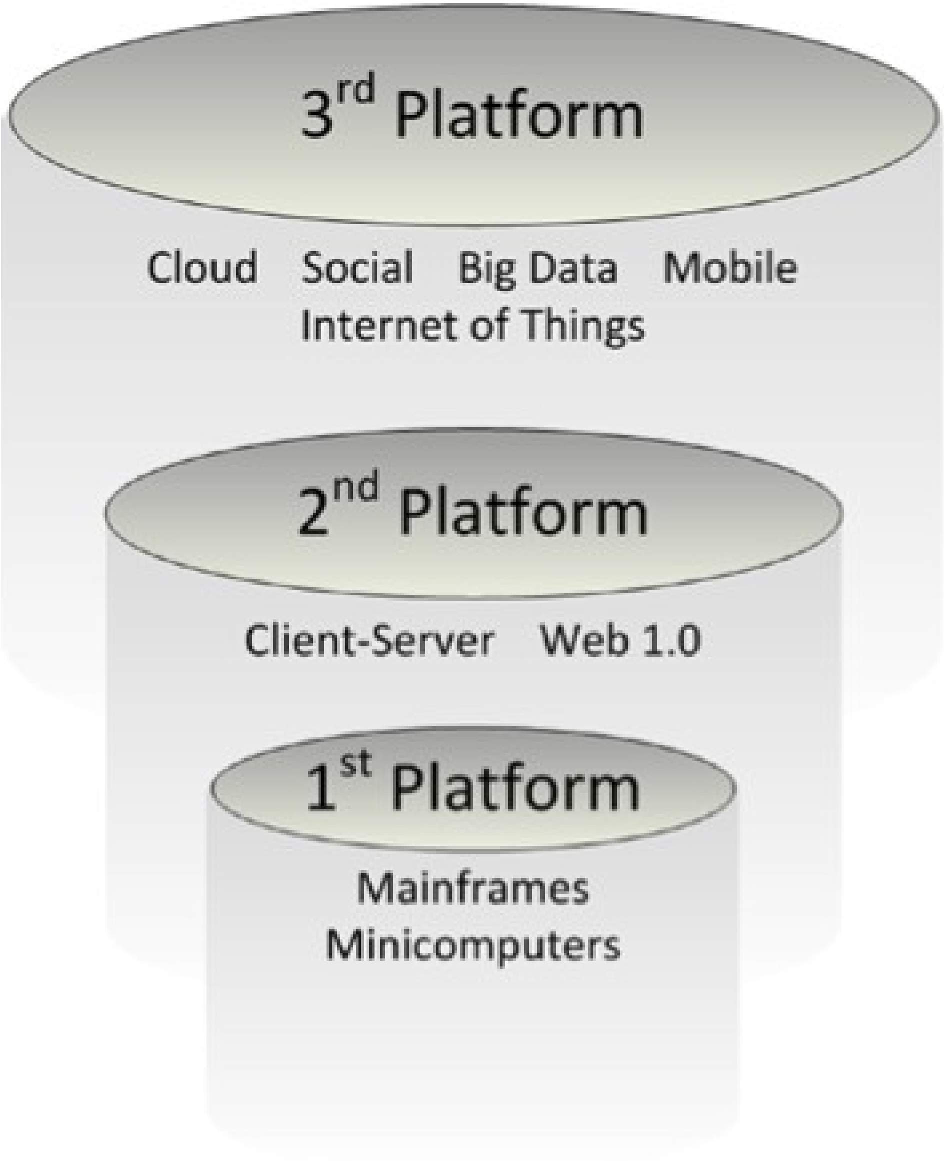
1951: Magnetic Tape
1955: Magnetic Disk
1961: ISAM
1965: Hierarchical model
1968: IMS
1969: Network Model
1971: IDMS

1972 - 2005
Relational

1970: Codd's Paper
1974: System R
1978: Oracle
1980: Commerical Ingres
1981: Informix
1984: DB2
1987: Sybase
1989: Postgres
1989: SQL Server
1995: MySQL

2005 - 2015
The Next Generation

2003: MarkLogic
2004: MapReduce
2005: Hadoop
2005: Vertica
2007: Dynamo
2008: Cassandra
2008: Hbase
2008: NuoDB
2009: MongoDB
2010: VoltDB
2010: Hana
2011: Riak
2012: Areospike
2014: Splice Machine



3rd Platform

Cloud Social Big Data Mobile
Internet of Things

Relational Database
NoSQL
NewSQL
Big Data platforms

2nd Platform

Client-Server Web 1.0

Relational Database

1st Platform

Mainframes
Minicomputers

Hierarchical Database
Network Database
ISAM files

New database designs emerge

- In 2007, Michael Stonebraker
  - *“the hardware assumptions that underlie the consensus relational architecture no longer applied, a single architecture might not be optimal across all workloads”*

- *NoSQL, NewSQL, and Big Data*
  vaguely defined, overhyped, and overloaded terms
  - *NoSQL → reject the constraints of the relational model*
  - *NewSQL→ retain many features of the relational model but new technology*
  - *Big Data systems*
    → generally technologies within the Hadoop ecosystem, increasingly including Spark