UCSC 2016

NOSQL DATABASES

Database Systems II

THEJAKA USGODAARACHCHI

Software Architect

- ❑ MSc in Computer Science
- ❑ BSc in Physical Science
- ❑ Currently working in Intel Software as a Software Architect
- ❑ Keen on research and development on IOT
- ❑ Reading for MSc in Applied Physics

thejaka@gmail.com

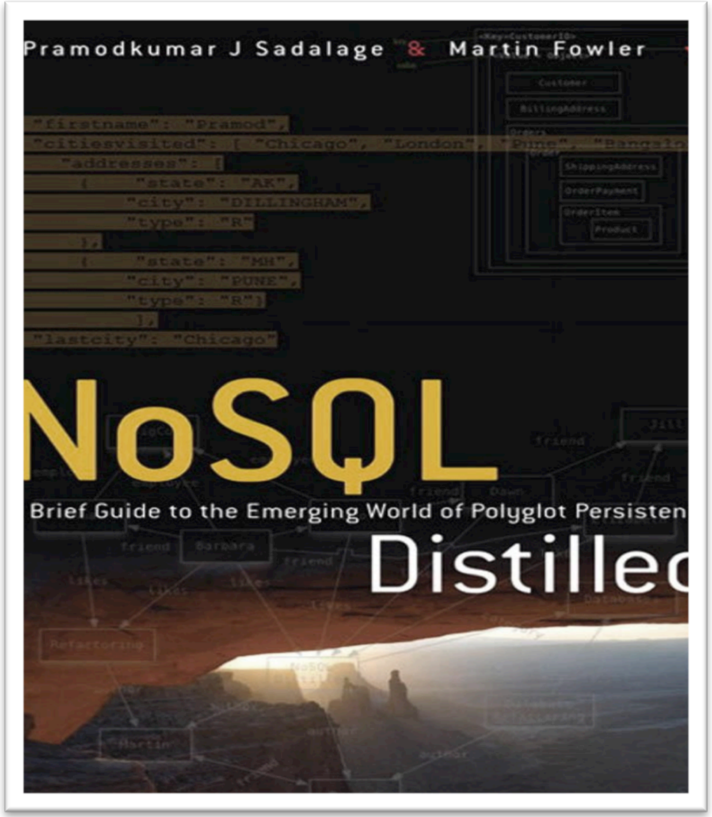
THANK YOU KASUN DILUNIKA FOR PROVIDING SOME SLIDES

http://www.dilunika.com

kasundilunika@gmail.com

RDBMS TO NOSQL

Lecture 1

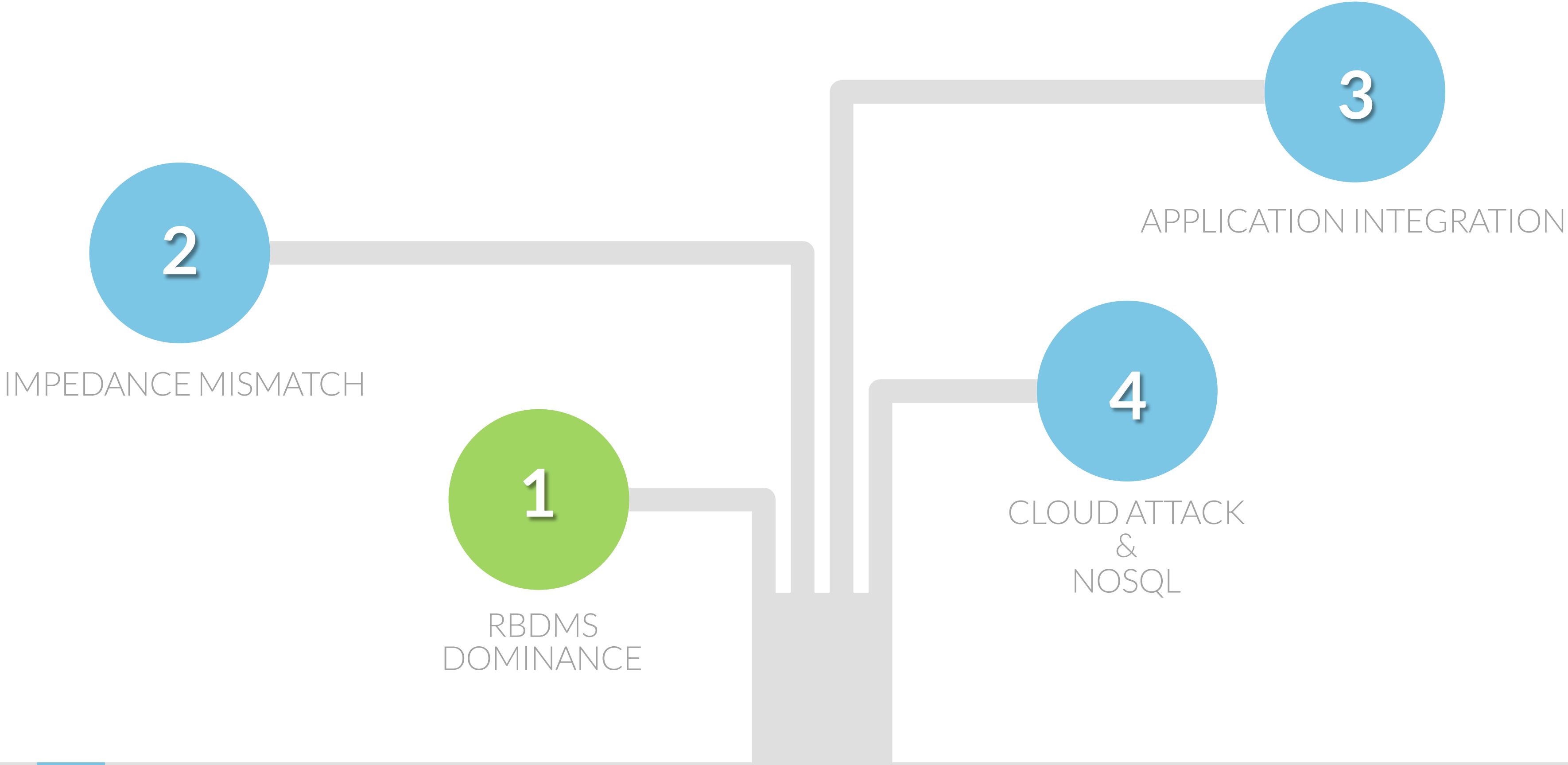

Pramodkumar J Sadalage & Martin Fowler
NoSQL
Brief Guide to the Emerging World of Polyglot Persisten
Distilled

MAIN REFERENCE

NoSQL Distilled

- A book by Pramod J. Sadalage and Martin Fowler
- A Brief Guide to the Emerging World of Polyglot Persistence
- Very concise and small book.
- A must read to all the students.

ROAD MAP


3
APPLICATION INTEGRATION
2
IMPEDANCE MISMATCH
4
CLOUD ATTACK
&
NOSQL
1
RBDMS
DOMINANCE

RELATIONAL DATABASES WERE **DOMINATING**

in persistent world

PERSISTENT DATA

Probably the most obvious value of a database is keeping large amounts of persistent data

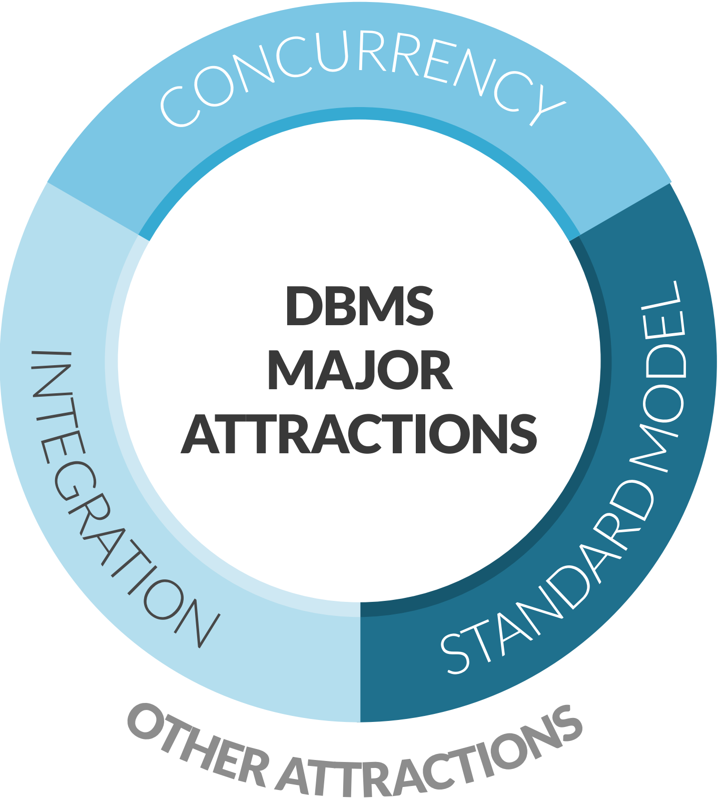
- ❑ Two areas of memory:
  - – Fast Volatile Main memory
  - – Slow Backing Store
- ❑ Backing Store can be,
  - – File System in productivity applications (word processors, ..)
  - – Database in Enterprise Application

DATABASES OVER FLAT FILES

The database allows more flexibility than a file system in storing large amounts of data in a way that allows an application program to get at small bits of that information quickly and easily.

Why Database win?

Basically databases had three major attractions to stick their users.



CONCURRENCY

Something which is notoriously difficult to get right

- SAME DATA, SAME TIME, MANY ACCESS
- AVOID CONCURRENT ACCESS PROBLEMS
- RDBMS USES TRANSTIONS TO HANDLE THOSE

Relational databases help handle concurrent issues by controlling all access to their data through transactions.

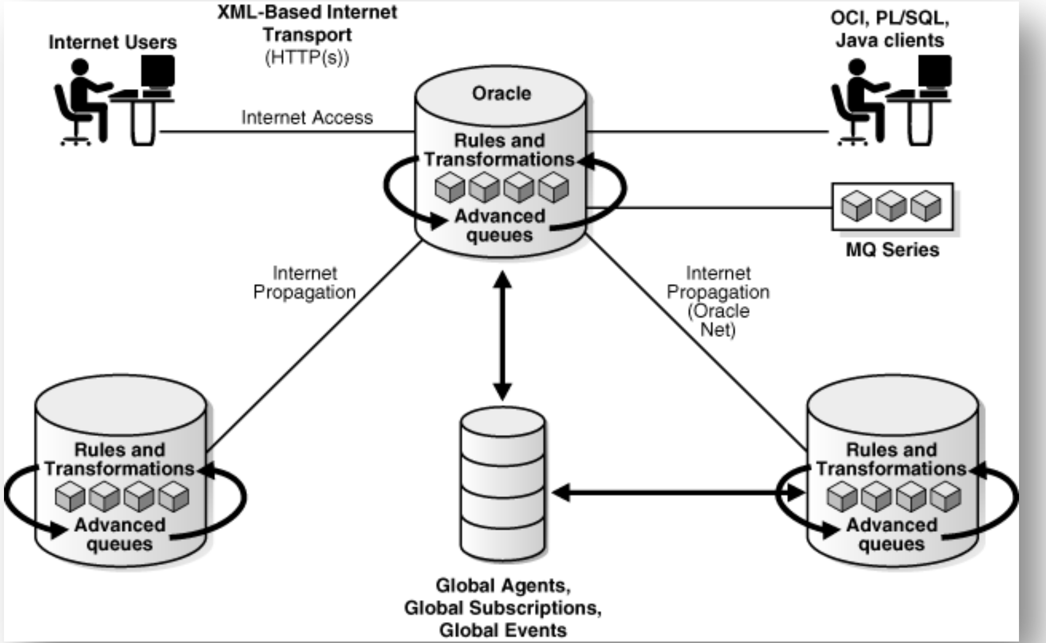
While this isn’t a cure all, the transactional mechanism has worked well to contain the complexity of concurrency.

.

APPLICATION INTEGRATION

Pushing the human organizational boundaries

- ❑ Applications use same data
- ❑ Updates made through one application visible to others
- ❑ DB's concurrency controls system handles concurrency problems



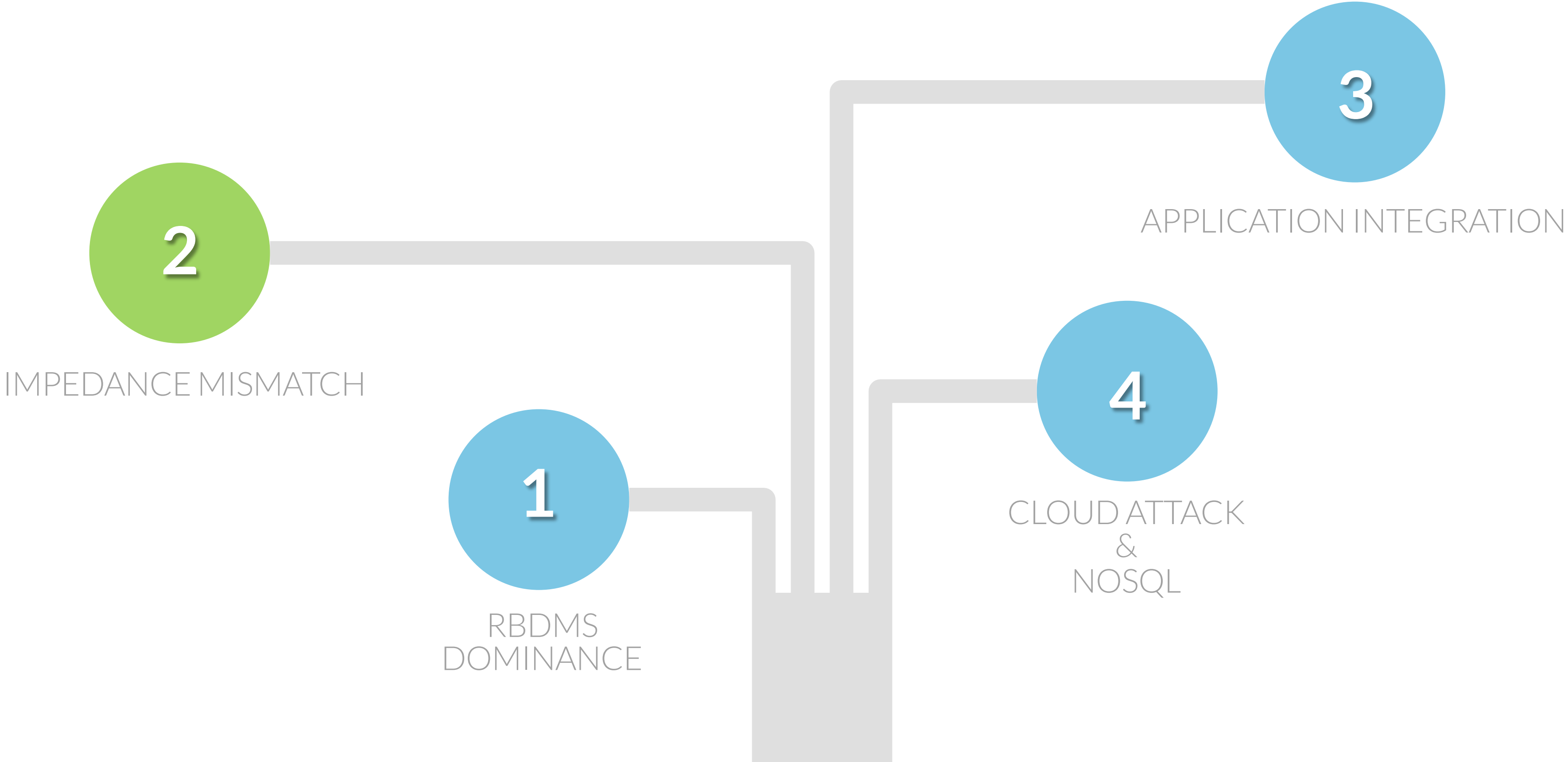
Internet Users
XML-Based Internet Transport (HTTP(s))
OCI, PL/SQL, Java clients
Oracle
Internet Access
Rules and Transformations
Advanced queues
MQ Series
Internet Propagation
Internet Propagation (Oracle Net)
Rules and Transformations
Advanced queues
Rules and Transformations
Advanced queues
Global Agents, Global Subscriptions, Global Events

STANDARD MODEL

More reasonable to say “mostly a standard model”

- ❑ Developers and database professionals
  - – Learn the basic relational model
  - – Apply it in many projects
- ❑ Different vendors, different DBs but,
  - – SQL dialects are similar,
  - – Transactions operate in mostly the same way.

ROAD MAP



1

RBDMS DOMINANCE

2

IMPEDANCE MISMATCH

3

APPLICATION INTEGRATION

4

CLOUD ATTACK
&
NOSQL

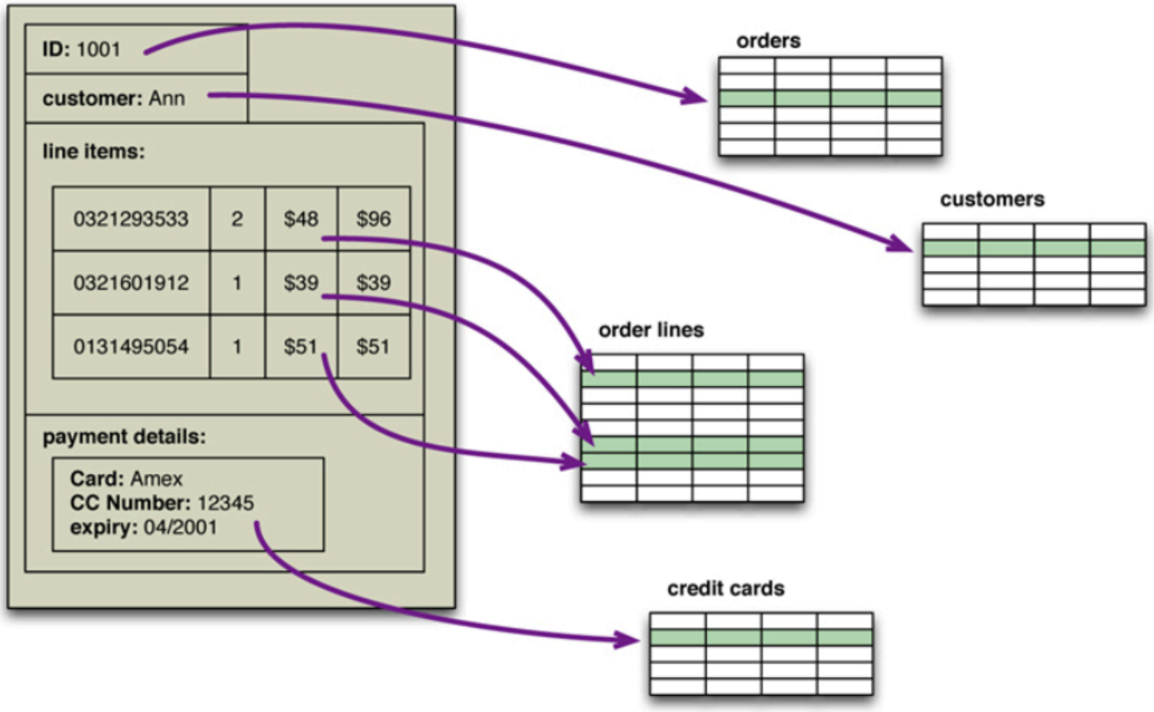
IMPEDANCE MISMATCH

for developers, this is a really boring, painful task

Impedance Mismatch

The difference between the relational model and the in-memory data structures.

Developer’s Biggest Frustration



- The relational data model organizes data into a structure of tables and rows, or more properly, relations and tuples. x

- If you want to use a richer in-memory data structure, you have to translate it to a relational representation to store it on disk.

90's – THE GOLDEN DAYS

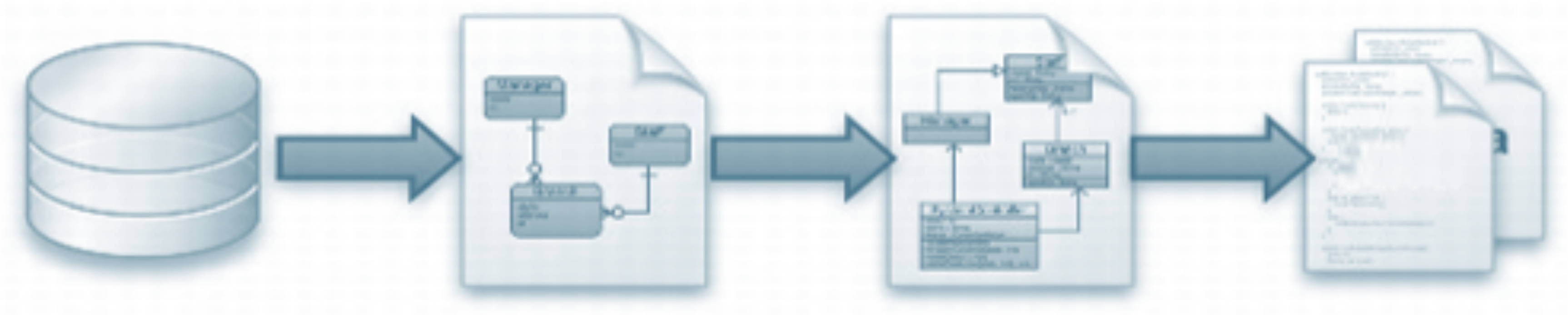
Rock Stars – Object Oriented Programming & Object Oriented Databases

❑ In 90s many people believed that, relational databases being replaced with databases that replicate the in-memory data structures to disk. (Object Oriented Databases)

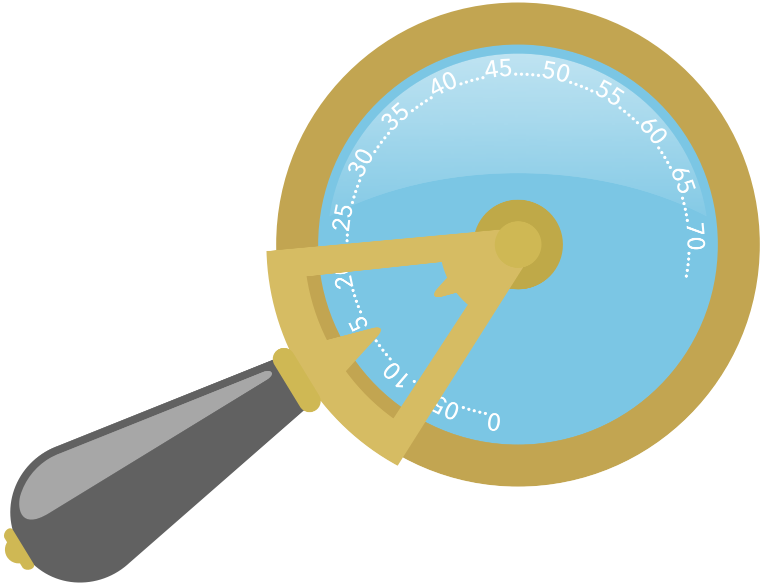
❑ However, while object-oriented languages succeeded in becoming the major force in programming, object-oriented databases faded into obscurity.

THE SAVIOR – ORM FRAMEWORKS

Object Relational Mapping Frameworks

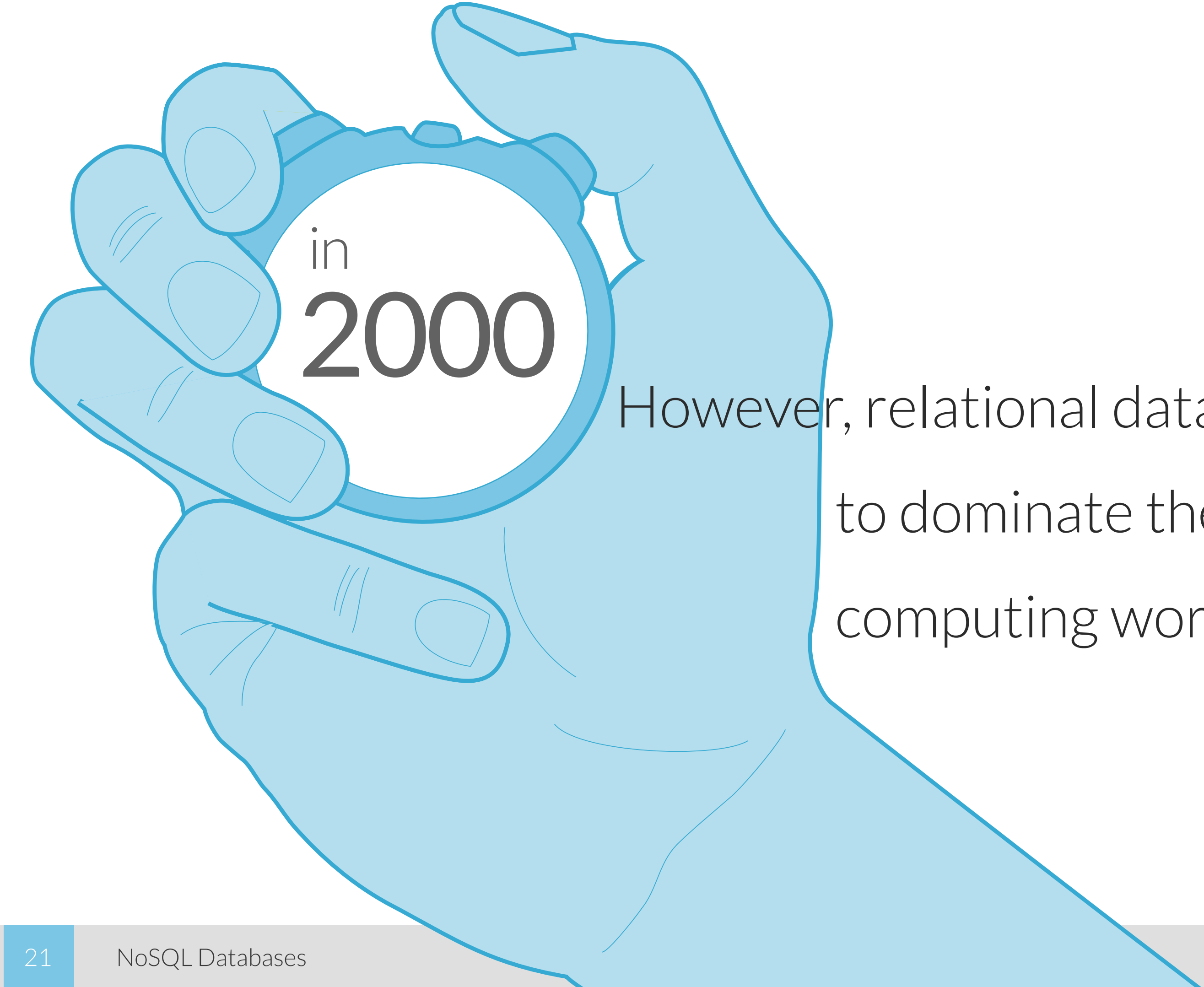


- ❑ Impedance mismatch has been made much easier to deal with by the wide availability of Object-Relational Mapping frameworks (ORM). (Hibernate, iBATIS)



Slow Savior

ORM frameworks remove a lot of boilerplate work, but can become a problem of their own when people try too hard to ignore the database and query performance suffers.

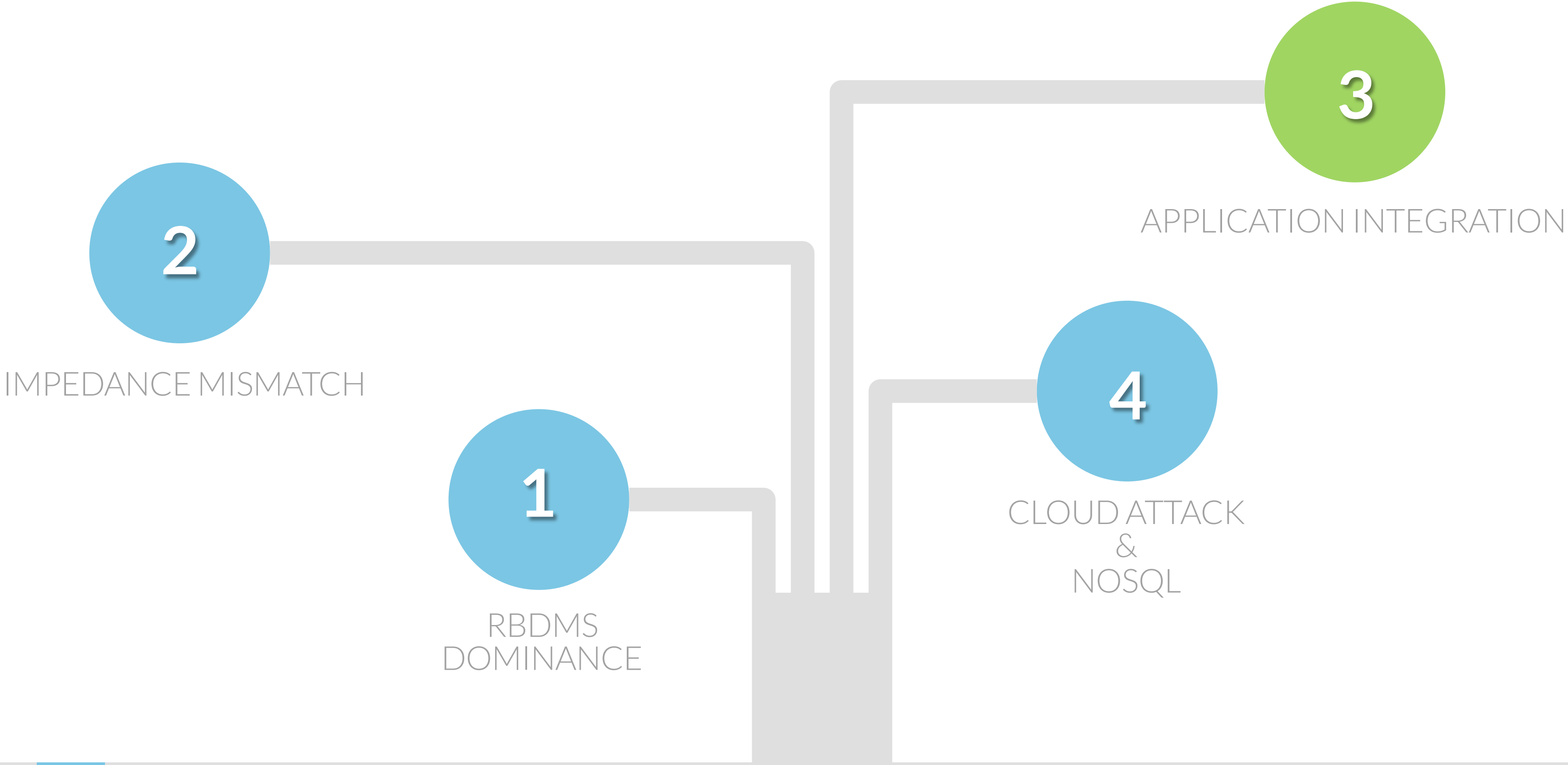


However, relational databases continued to dominate the enterprise computing world.

BUT FEW THREATS BEGAN TO CHALLENGE RDMS

The vital factor for a change in data storage was the need to support large volumes of data by running on clusters. Are Relational Databases capable on this?

ROAD MAP



3
APPLICATION INTEGRATION
2
IMPEDANCE MISMATCH
4
1
CLOUD ATTACK
&
NOSQL
RBDMS
DOMINANCE

LOSE OF RDBMS INTEGRATION USAGE

One of the main factors for RDBMS dominance over was role of

SQL as an integration mechanism between applications

RDBMS INTEGRATION USAGE

Database acts as an integration database

MULTIPLE APPLICATIONS

A structure that's designed to integrate many applications ends up being more complex.

DEVELOPED BY SEPARATE TEAMS

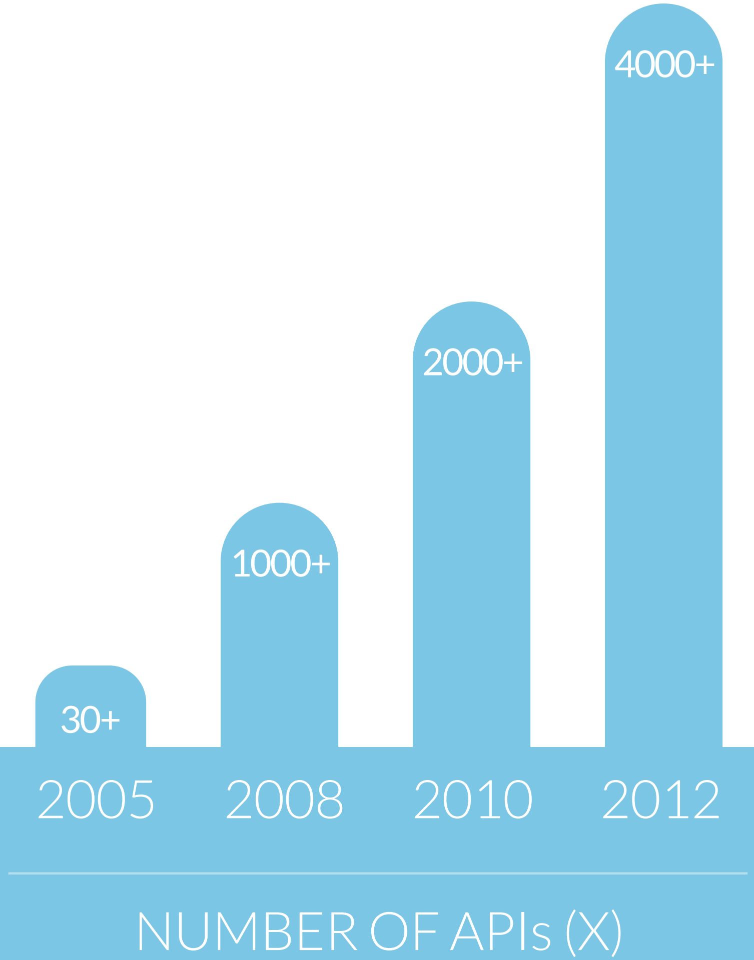
If an application want to make changes to its data storage, it needs to coordinate with all the other applications using the database.

STORING COMMON DATA

Different applications have different structural and performance needs, optimizing one, may cause to other's performance.

Rise of Web Services

During the 2000s, web services enabled a new form of a widely used communication mechanism.



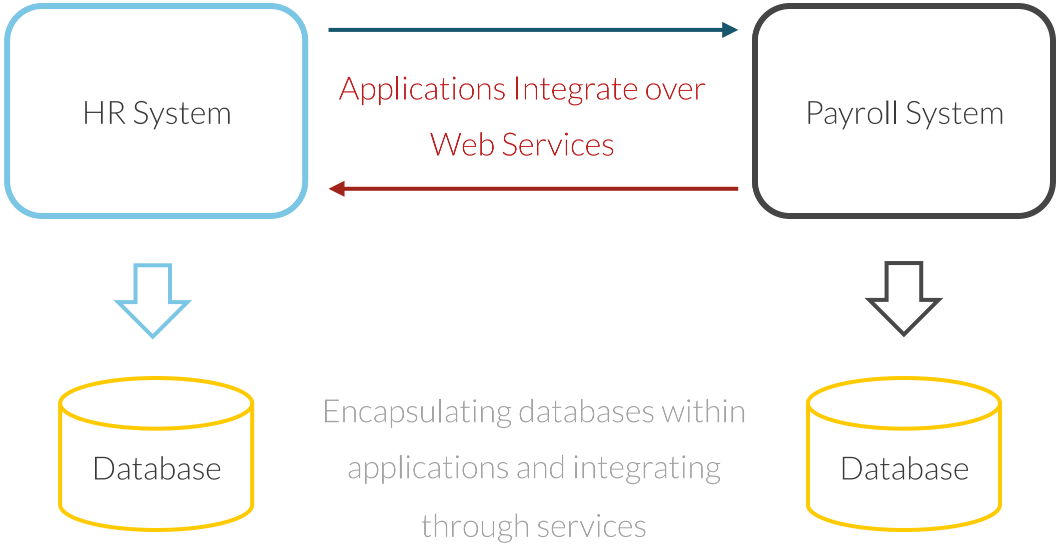
CHALLENGER TO USING THE SQL WITH SHARED RDMS

Integration mechanisms changed from mammoth integration SQLs to simple Remote Method Invocations.

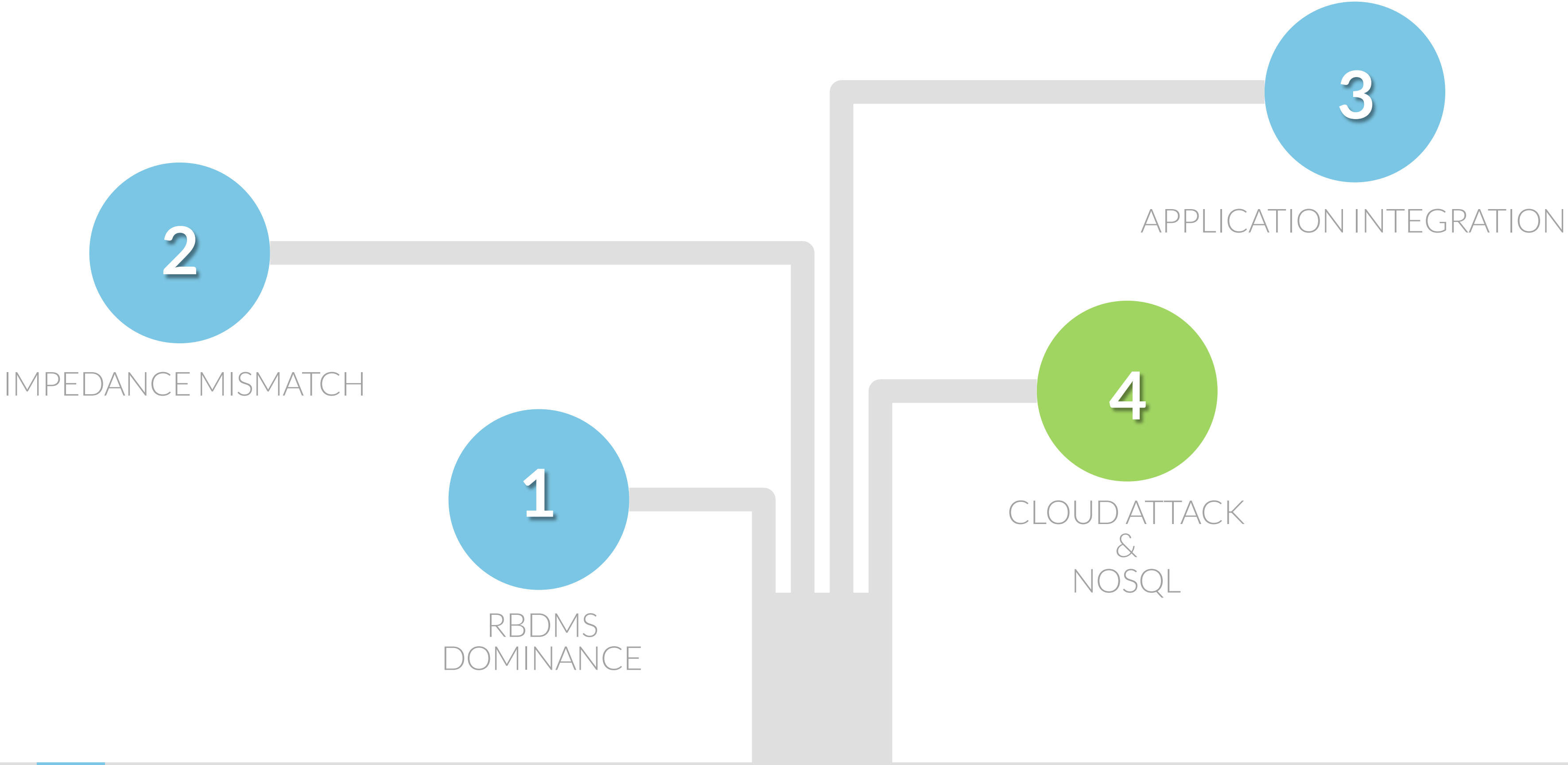
RDBMS AFTER WEB SERVICES

INTEGRATION : NO, PERSISTENT : YES

- ❑ Internal DB and Services talk to outside world are decoupled.
- ❑ Outside world doesn't have to care how you store your data.


HR System
Applications Integrate over Web Services
Payroll System
Database
Encapsulating databases within applications and integrating through services
Database

ROAD MAP

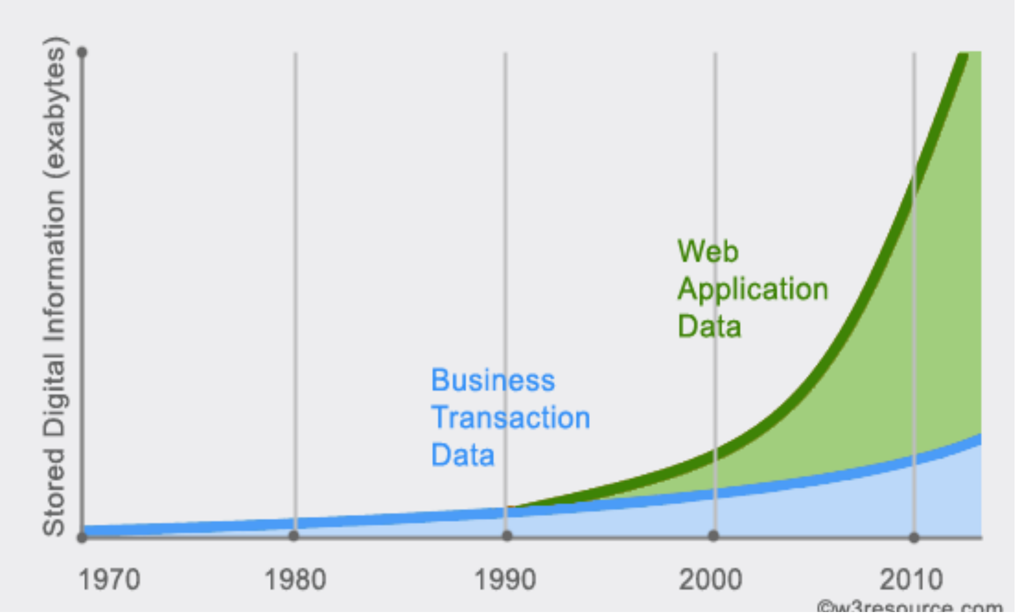

3
APPLICATION INTEGRATION
2
IMPEDANCE MISMATCH
4
CLOUD ATTACK
&
NOSQL
1
RBDMS
DOMINANCE

SQL WAS NOT **CLUSTER** READY

Opened a real crack in the relational hegemony

WWW IS GROWING…

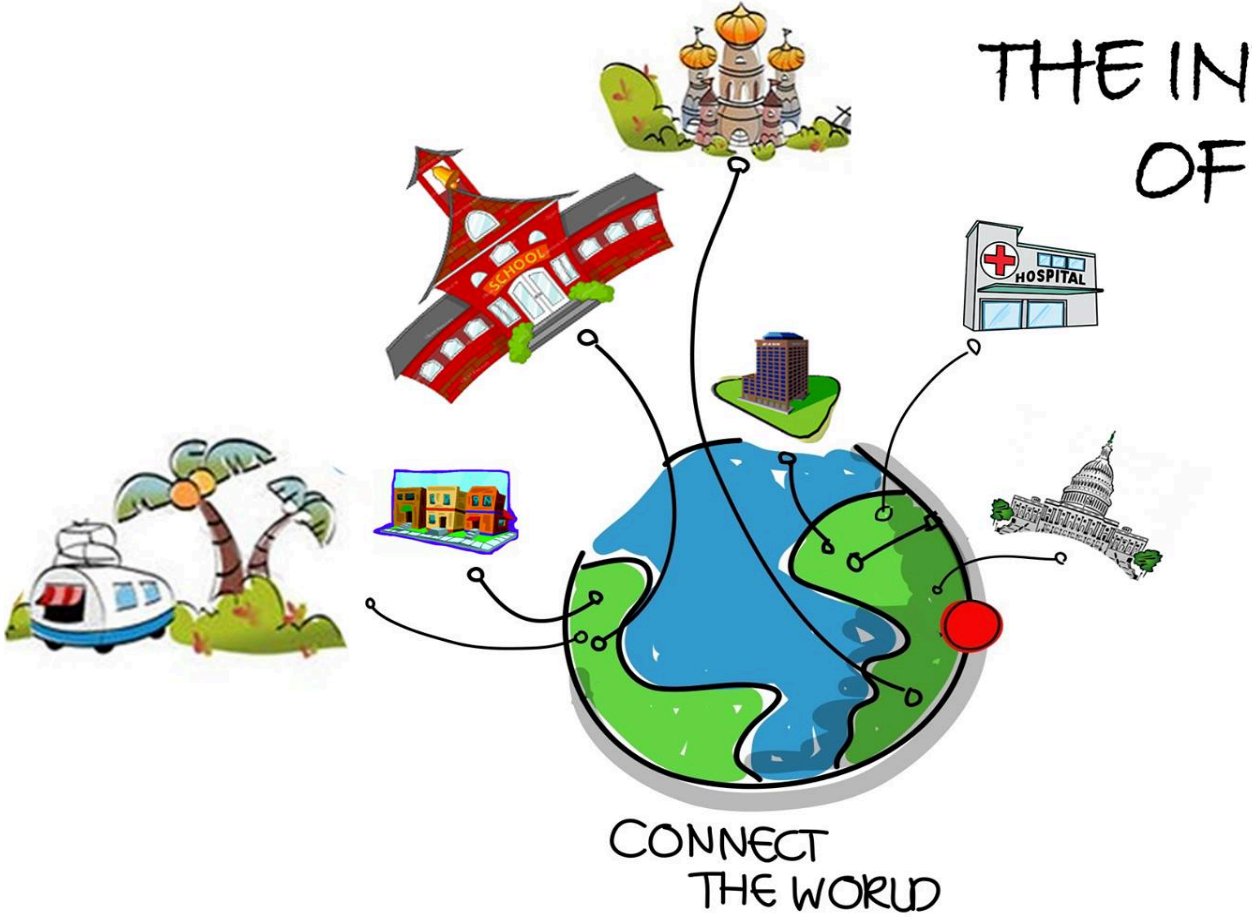
In 2000s, several large web properties dramatically increase in scale

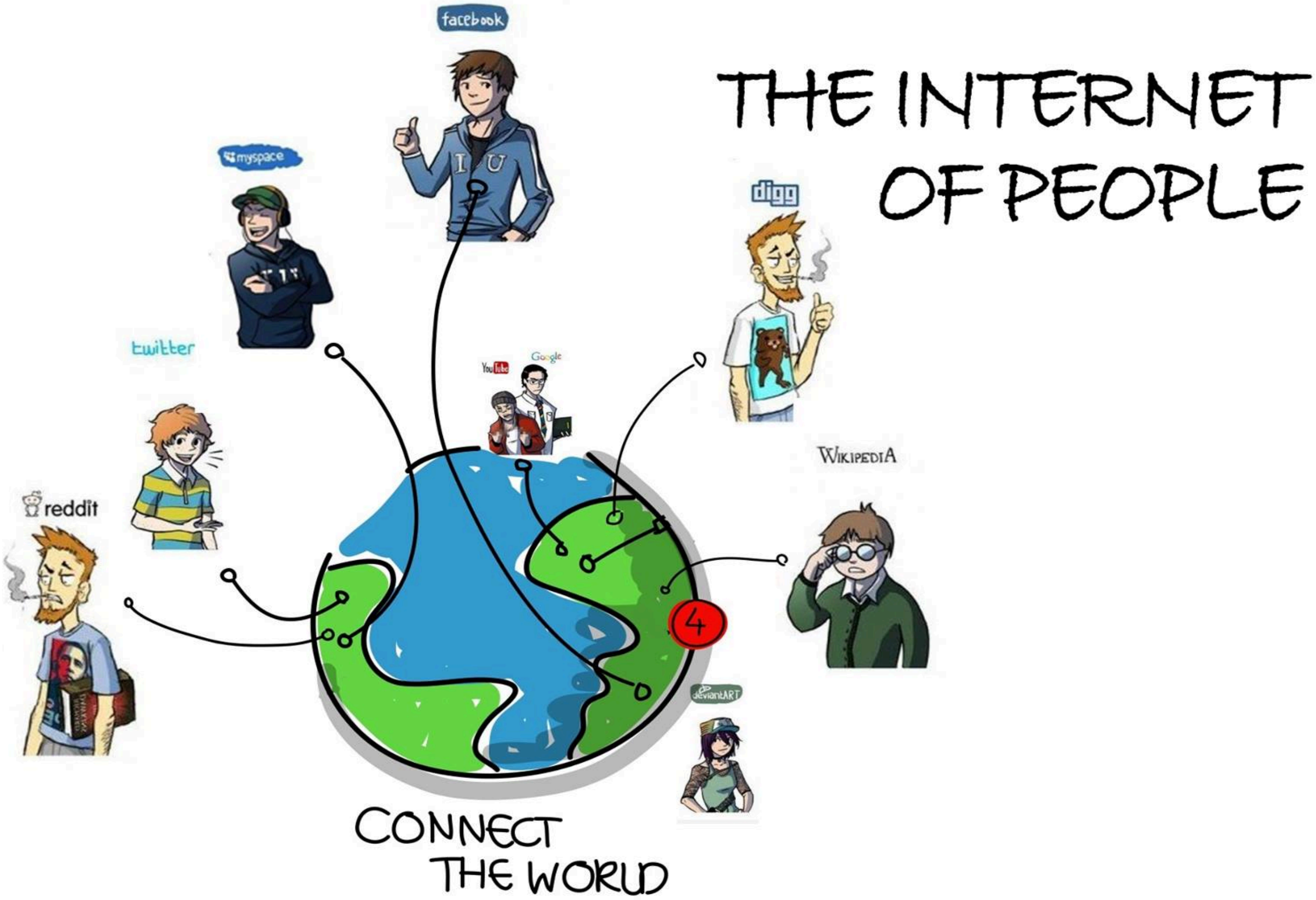


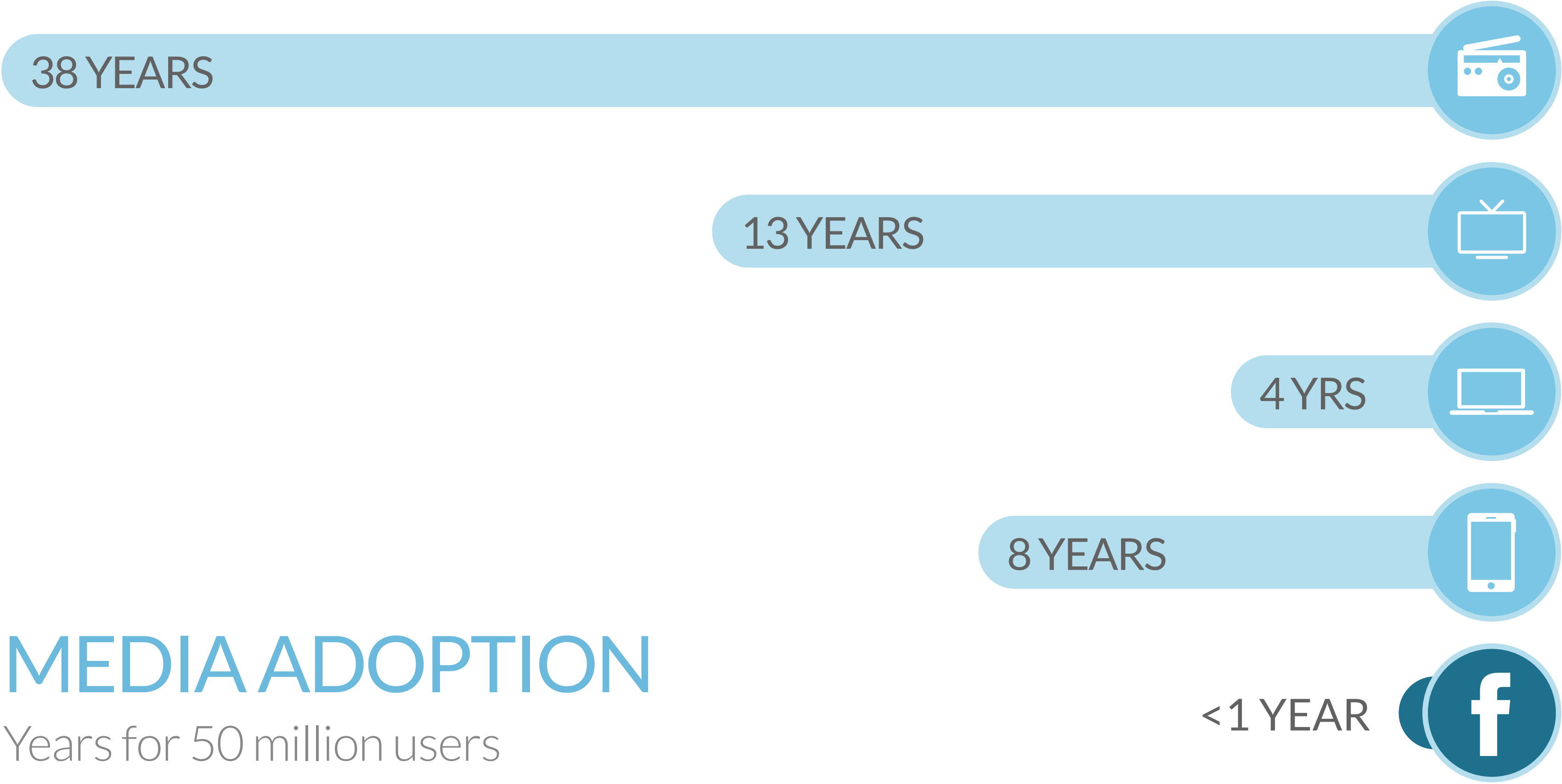
- ❑ Websites started tracking activity and structure in a very detailed way
- ❑ Large sets of data appeared: links, social networks, activity in logs, mapping data

Reference : w3resource.com

THE INTERNET OF PLACES




THE INTERNET OF PEOPLE
facebook
myspace
digg
twitter
YouTube
Google
WIKIPEDIA
reddit
4
deviantART
CONNECT THE WORLD



38 YEARS

13 YEARS

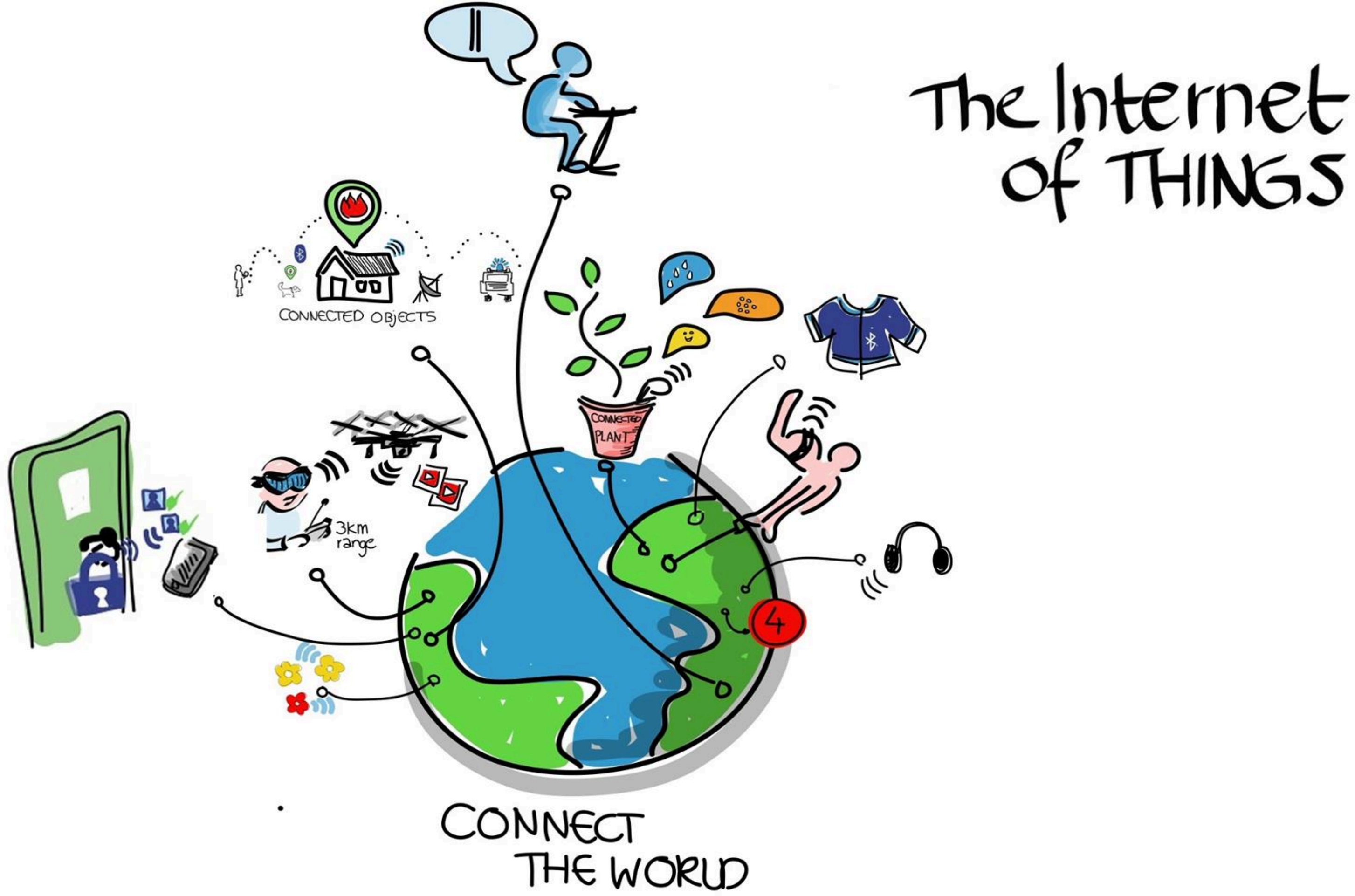
4 YRS

8 YEARS

<1 YEAR

MEDIA ADOPTION

Years for 50 million users


The Internet of THINGS
CONNECTED OBJECTS
CONNECTED PLANT
3km range
4
CONNECT THE WORLD

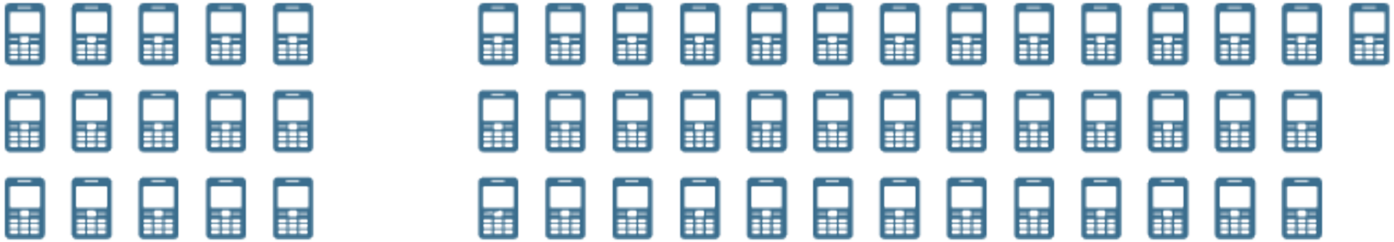
15B

CONNECTED DEVICES IN **2015**

40B

CONNECTED DEVICES BY **2020**

📱 = 1B DEVICES



CONNECTED DEVICES

Next few years

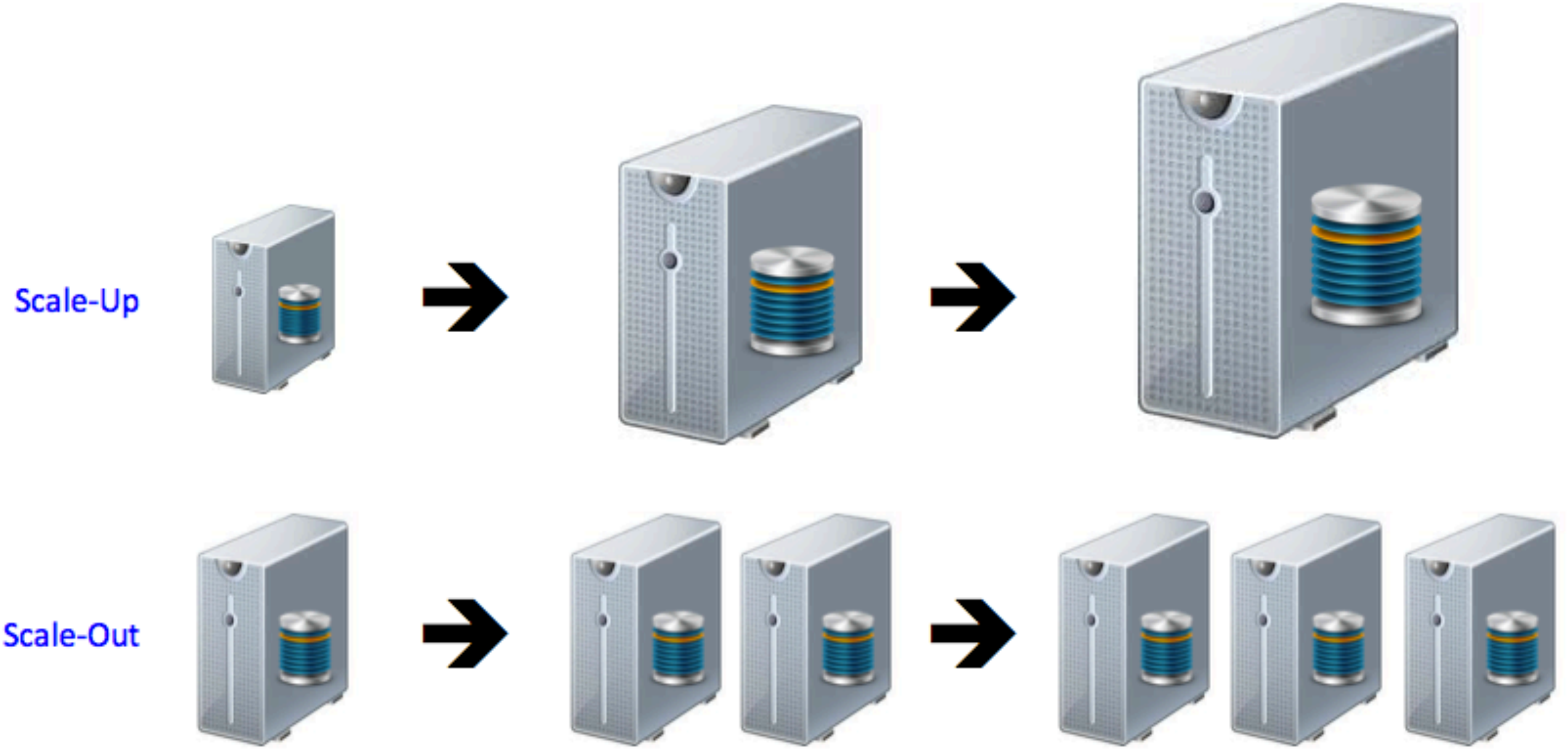
DATA COLLECTED IN TERA BYTES

Two options available for RDBMS

- ❑ By birth, Relational Model was not cluster ready
- ❑ Options were there in high cost, but not satisfied
- ❑ People lead to alternatives

1 Scale Up

2 Scale Out



Scale-Up

Scale-Out

NOSQL WAS **EMERGING** SILENTLY

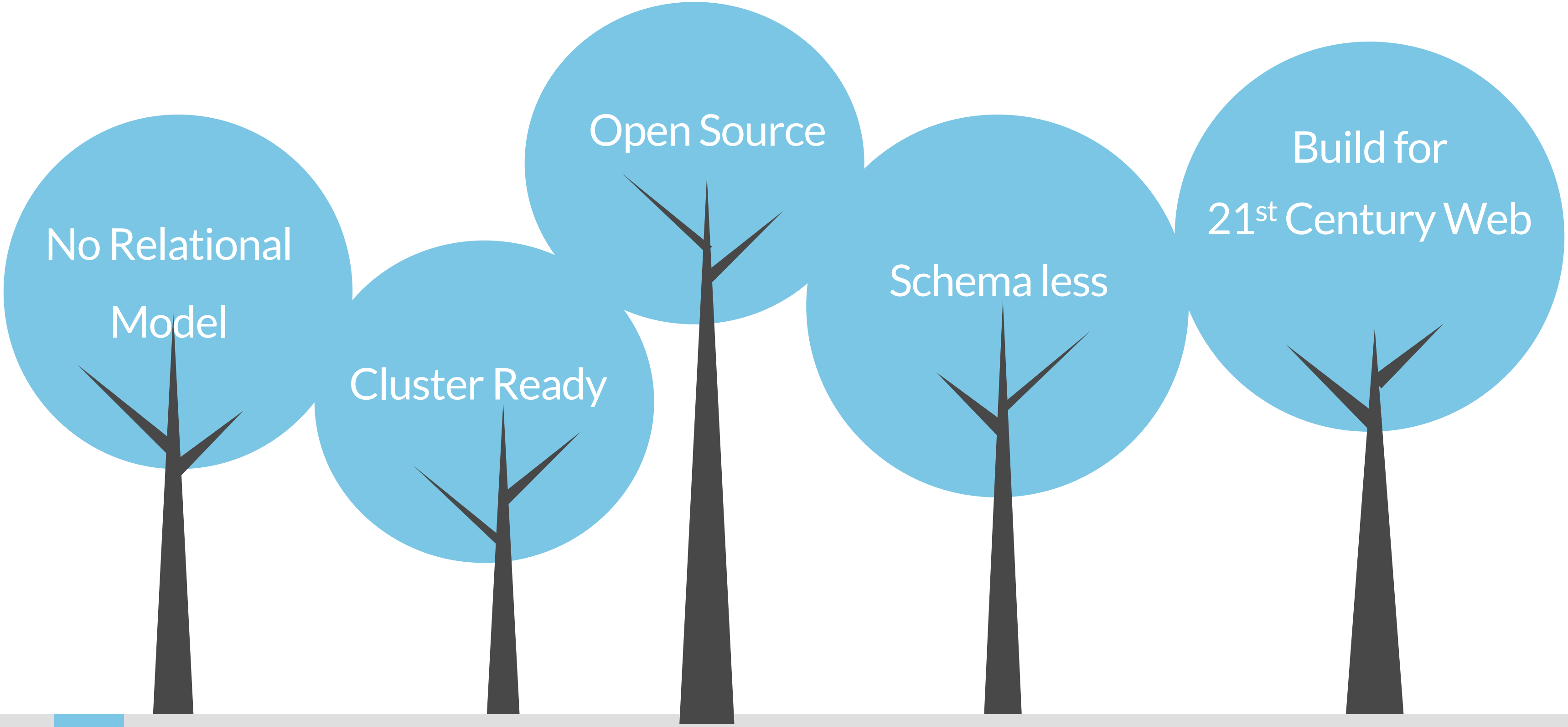
They challenged the dominance of RDBMS

NAME NoSQL IS #TAG

- NoSQL is an accidental neologism. There is no prescriptive definition.
- Johan Oskarsson, organized a meetup in San Francisco to talk about alternative data storage mechanisms.
- Johan wanted a name for the meetup—something that would make a good Twitter hashtag. #tag "NoSQL" was the result.
- The term "NoSQL" caught on like wildfire, but it's never been a term that's had much in the way of a strong definition.

NOSQL HAS NO DEFINITION

But share some common characteristics



No Relational Model

Cluster Ready

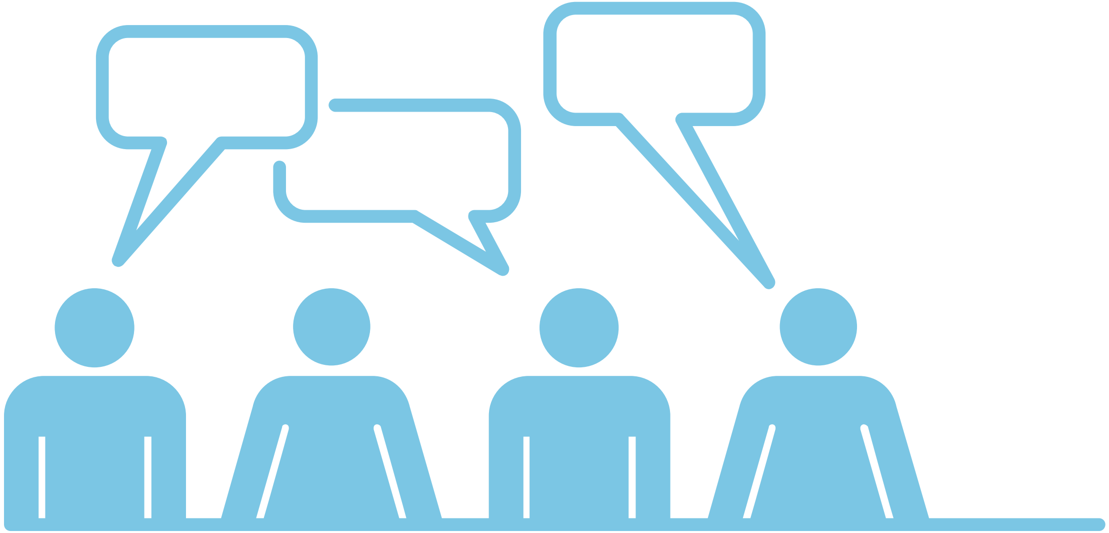
Open Source

Schema less

Build for 21st Century Web

FUTURE IS NOT NOSQL, BUT POLYGLOT PERSISTENCE

The most important result of the rise of NoSQL is Polyglot Persistence.



DISCUSSION