ADDRESSING DATA MANAGEMENT ON THE CLOUD: *TACKLING THE BIG DATA CHALLENGES*

GENOVEVA VARGAS SOLAR

FRENCH COUNCIL OF SCIENTIFIC RESEARCH, LIG-LAFMIA, FRANCE

Genoveva.Vargas@imag.fr

http://www.vargas-solar.com

"With hardware, networks and software having been commoditized to the point that all are essentially free, it was inevitable that the trajectory toward maximum entropy would bring us the current age of Big Data."

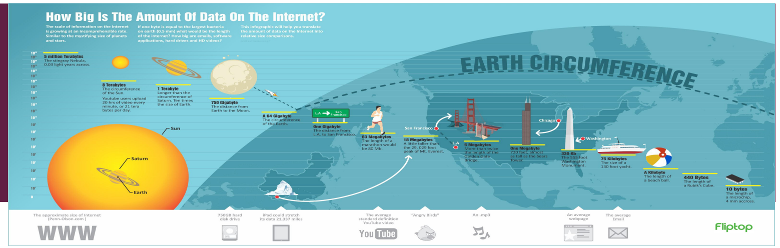
-Shomit Ghose, Big Data, The Only Business Model Tech has Left; CIO Network
http://www.bigdatabytes.com/big-data-is-the-new-tech-economy/

Megabyte(10^6) Gigabyte (10^9) Terabytes (10^{12}), petabytes (10^{15}), exabytes (10^{18}) and zettabytes (10^{21})

DIGITAL INFORMATION SCALE

Unit	Size	Meaning
Bit (b)	1 or 0	Short for binary digit, after the binary code (1 or 0) computers use to store and process data
Byte (B)	8 bits	Enough information to create an English letter or number in computer code. It is the basic unit of computing
Kilobyte (KB)	2^{10} bytes	From "thousand' in Greek. One page of typed text is 2KB
Megabyte (MB)	2^{20} bytes	From "large" in Greek. The complete works of Shakespeare total 5 MB. A typical pop song is 4 MB.
Gigabyte (GB)	2^{30} bytes	From "giant" in Greek. A two-hour film ca be compressed into 1-2 GB.
Terabyte (TB)	2^{40} bytes	From "monster" in Greek. All the catalogued books in America's Library of Congress total 15TB.
Petabyte (PB)	2^{50} bytes	All letters delivered in America's postal service this year will amount ca. 5PB. Google processes 1PB per hour.
Exabyte (EB)	2^{60} bytes	Equivalent to 10 billion copies of The Economist
Zettabyte (ZB)	2^{70} bytes	The total amount of information in existence this year is forecast to be around 1,27ZB
Yottabyte (YB)	2^{80} bytes	Currently too big to imagine

MASSIVE DATA



Data sources

- Information-sensing mobile devices, aerial sensory technologies (remote sensing)
- Software logs, posts to social media sites
- Telescopes, cameras, microphones, digital pictures and videos posted online
- Transaction records of online purchases
- RFID readers, wireless sensor networks (number of sensors increasing by 30% a year)
- Cell phone GPS signals (increasing 20% a year)

Massive data

- Sloan Digital Sky Survey (2000-) its archive contains 140 terabytes. Its successor: Large Synoptic Survey Telescope (2016-), will acquire that quantity of data every five days !
- Facebook hosts 140 billion photos, and will add 70 billion this year (ca. 1 petabyte). Every 2 minutes today we snap as many photos as the whole of humanity took in the 1800s !
- Wal-Mart, handles more than 1 million customer transactions every hour, feeding databases estimated at more than 2.5 petabytes (10^{15})
- The Large Hadron Collider (LHC): nearly 15 million billion bytes per year -15 petabytes (10^{15}). These data require 70,000 processors to be processed!

http://blog.websourcing.fr/infographie-la-vrai-taille-dinternet/

BIG DATA

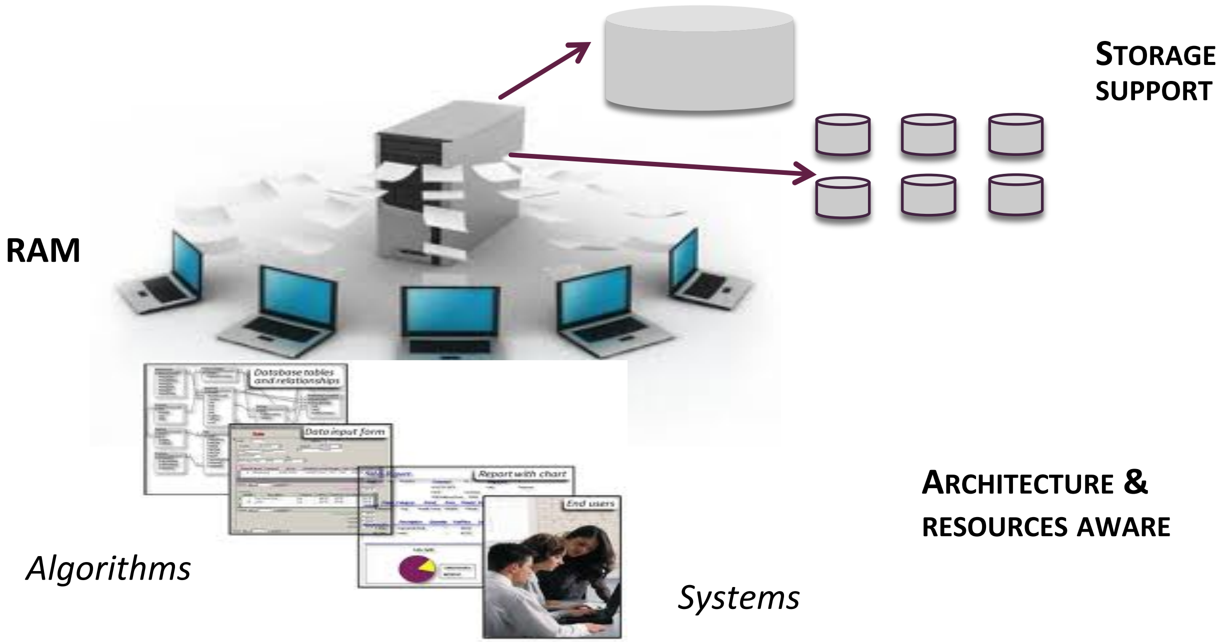
- Collection of data sets so large and complex that it becomes difficult to process using on-hand database management tools or traditional data processing applications.
- Challenges include capture, curation, storage, search, sharing, analysis, and visualization **within a tolerable elapsed time**
- Data growth challenges and opportunities are three-dimensional (**3Vs model**)
  - increasing volume (amount of data)
  - velocity (speed of data in and out)
  - variety (range of data types and sources)

"Big data are high volume, high velocity, and/or high variety information assets that require **new forms of processing** to enable enhanced decision making, insight discovery and process optimization."

AGENDA

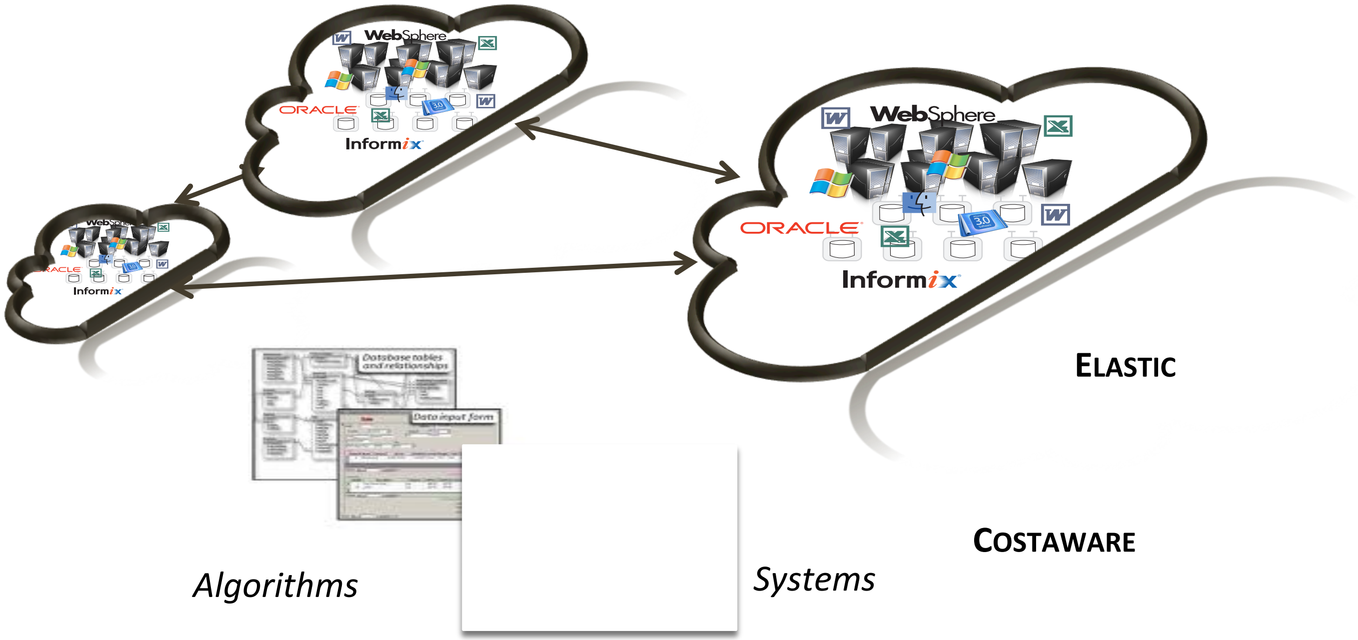
- *Big data all around*
- *Processing and exploiting big data*
  - *Cloud architectures*
  - *Storage vs. Access*
  - *DBMS Architectures*
- *Big data economy*
- *Outlook*

DATA MANAGEMENT WITH RESOURCES CONSTRAINTS



Efficiently manage and exploit data sets according to given specific storage, memory and computation resources

DATA MANAGEMENT WITHOUT RESOURCES CONSTRAINTS



ELASTIC

COSTAWARE

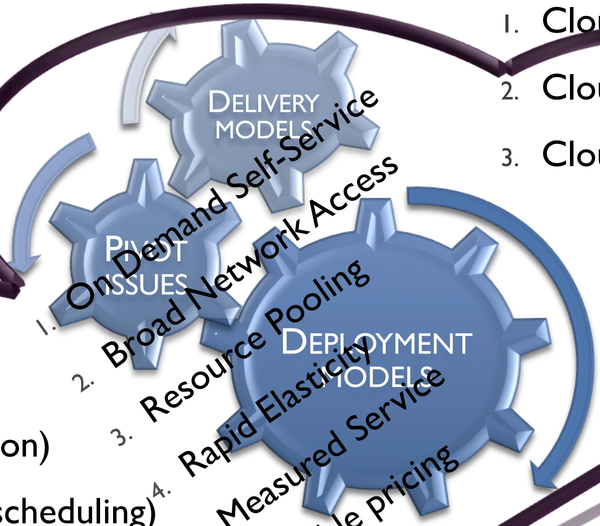
Algorithms

Systems

Costly manage and exploit data sets according to unlimited storage, memory and computation resources

CLOUD IN A NUTSHELL

1. Cloud Software as a Service (SaaS)
2. Cloud Platform as a Service (PaaS)
3. Cloud Infrastructure as a Service (IaaS)



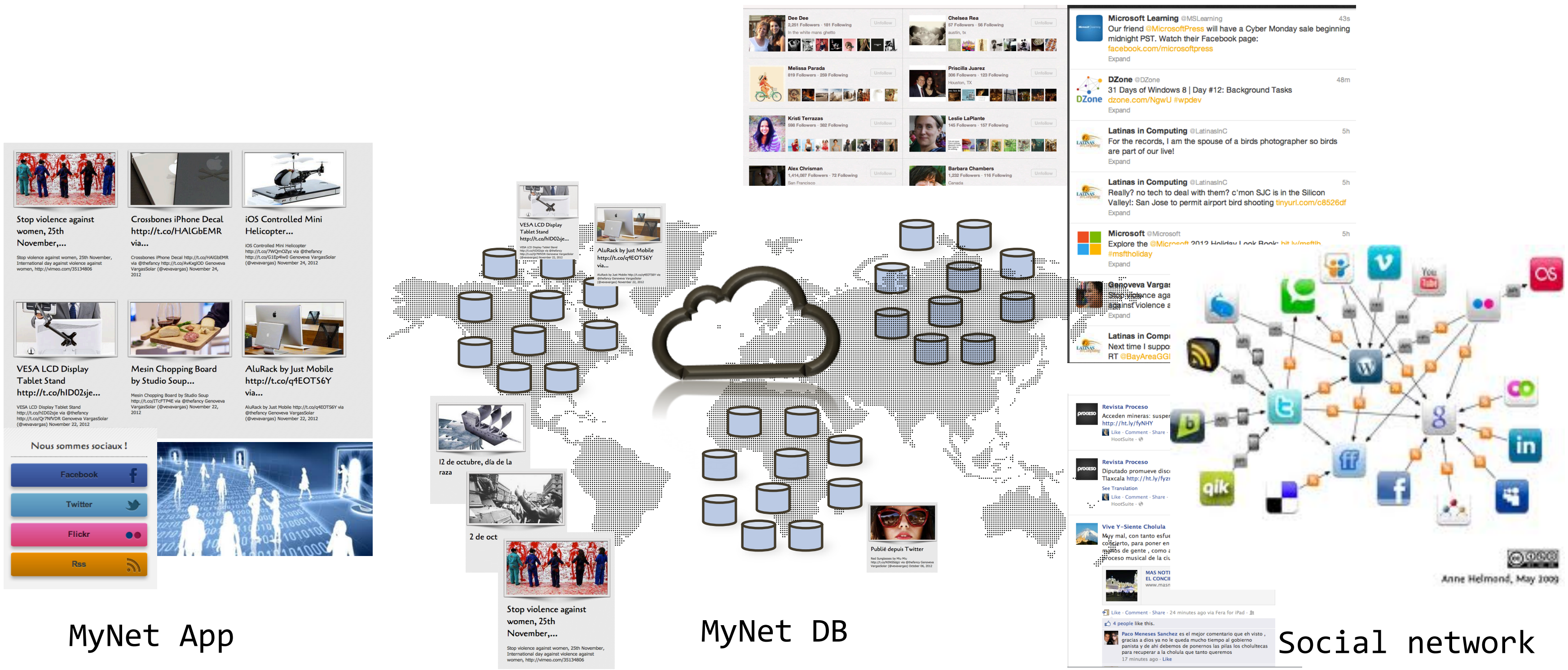
1. On Demand Self-Service
2. Broad Network Access
3. Resource Pooling
4. Rapid Elasticity
5. Measured Service
6. Scalable pricing

1. Virtualization
2. Autonomics (automation)
3. Grid Computing (job scheduling)

1. Private cloud
2. Public cloud
3. Hybrid cloud
4. Community Cloud

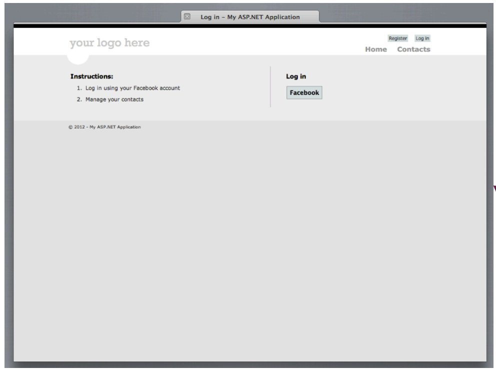
AGENDA

- *Big data all around*
- *Processing and exploiting big data*
  - *Cloud architectures*
  - *Storage vs. Access*
  - *DBMS Architectures*
- *Big data economy*
- *Outlook*

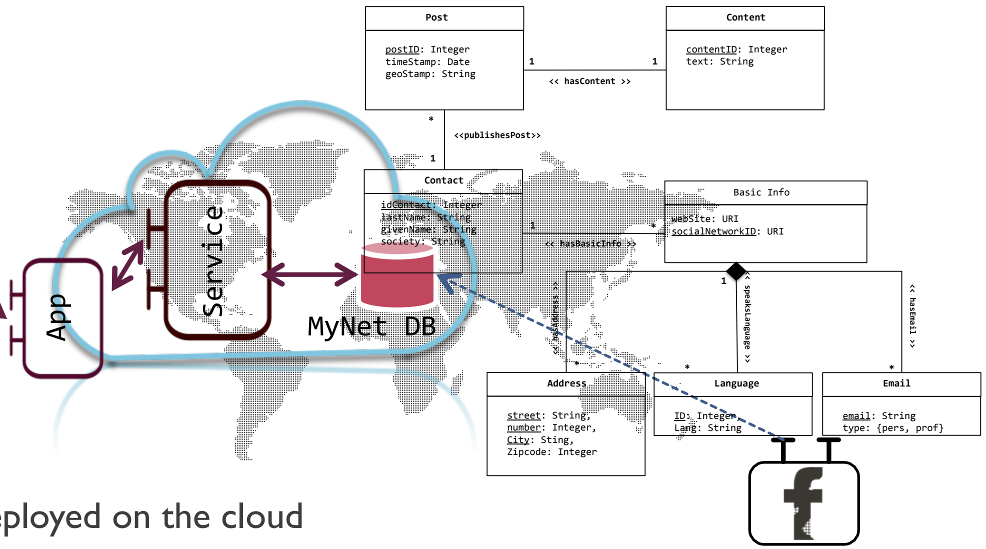


- Build a MyNet app based on a distributed database for building an integrated contact directory and posts from social networks

CLASSIC CLOUD AWARE SOLUTION



MyNet App



A step on the cloud

- creating a database on a relational DBMS deployed on the cloud
- building a simple database application exported as a service
- deploying the service on the cloud and implement an interface

SQL has Ruled for two decades

- **Store persistent data**

  Storing large amounts of data on disk, while allowing applications to grab the bits they need through queries

- **Application Integration**

  Many applications in an enterprise need to share information. By getting all applications to use the database, we ensure all these applications have consistent, up-to-date data

- **Mostly Standard**

  The relational model is widely used and understood. Interaction with the database is done with SQL, which is a (mostly) standard language. This degree of standardization is enough to keep things familiar so people don't need to learn new things

- **Concurrency Control**

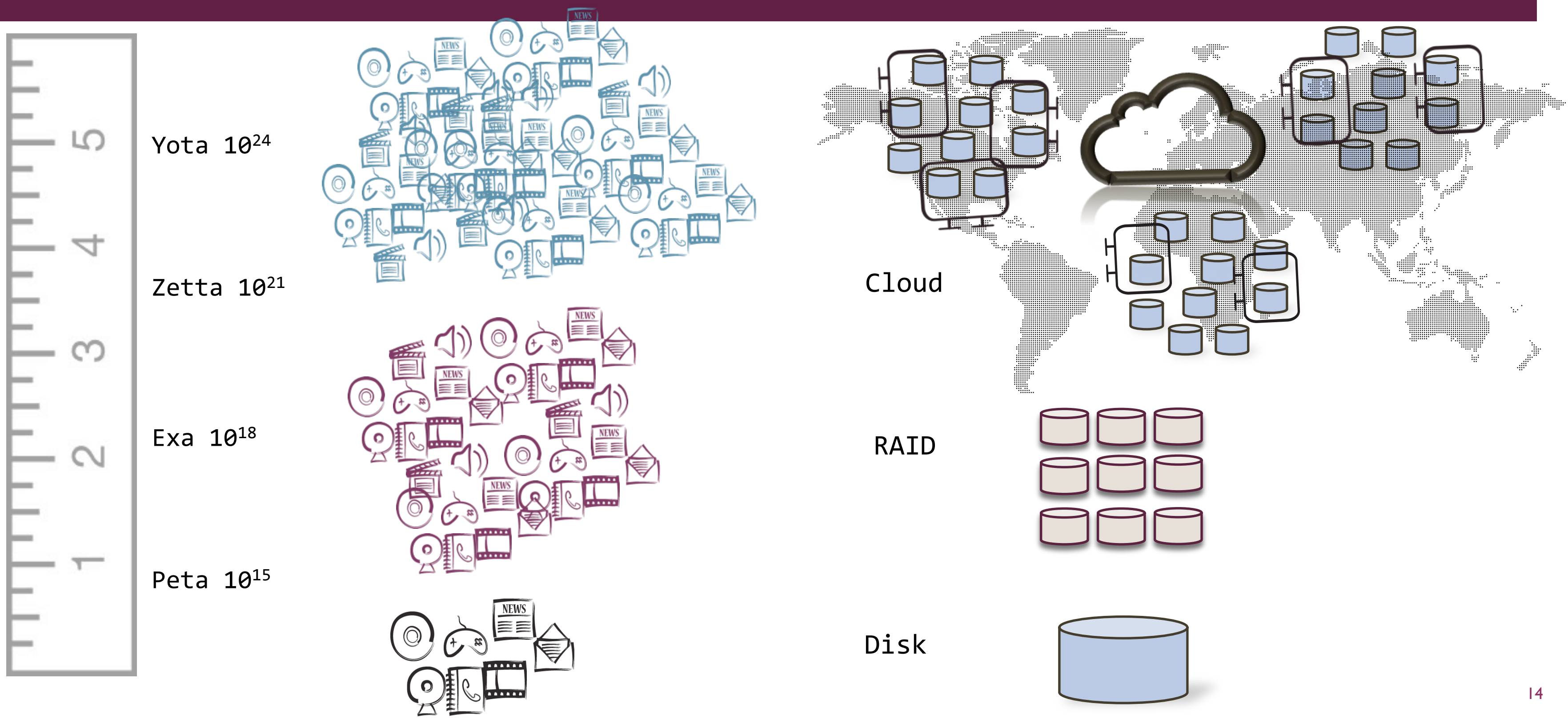
  Many users access the same information at the same time. Handling this concurrency is difficult to program, so databases provide transactions to help ensure consistent interaction.

- **Reporting**

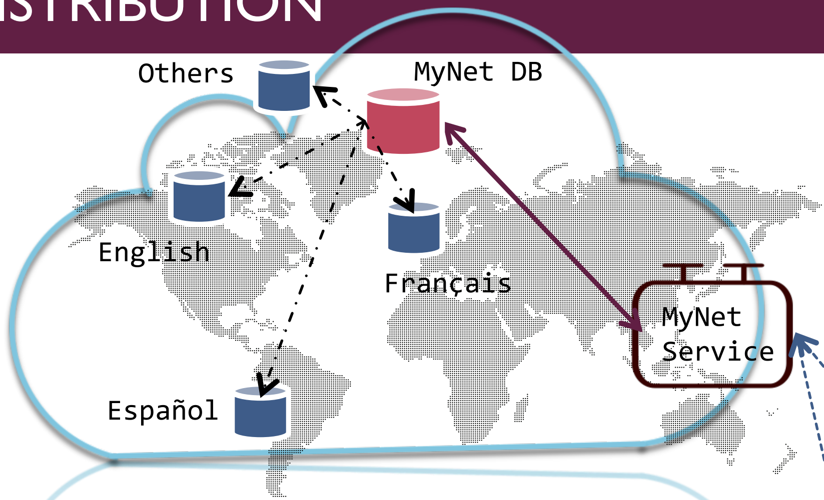
  SQL's simple data model and standardization has made it a foundation for many reporting tools

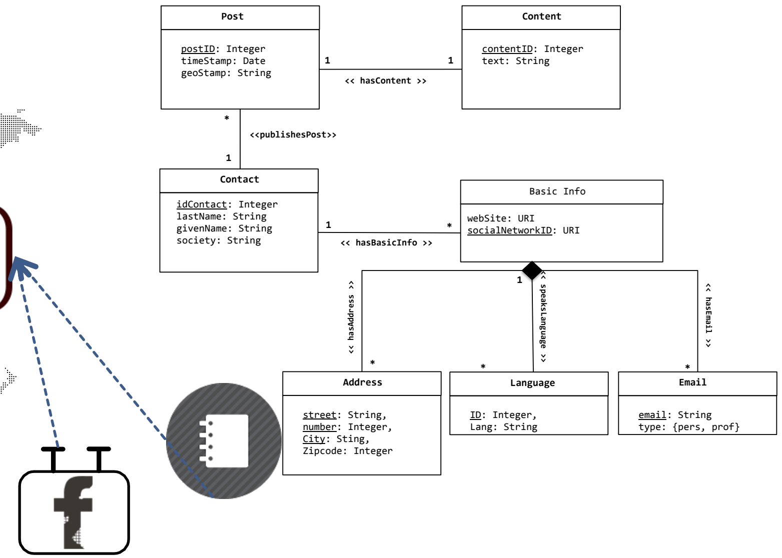
All this supported by Big Database Vendors and the separation of the DBA profession.

DEALING WITH HUGE AMOUNTS OF DATA



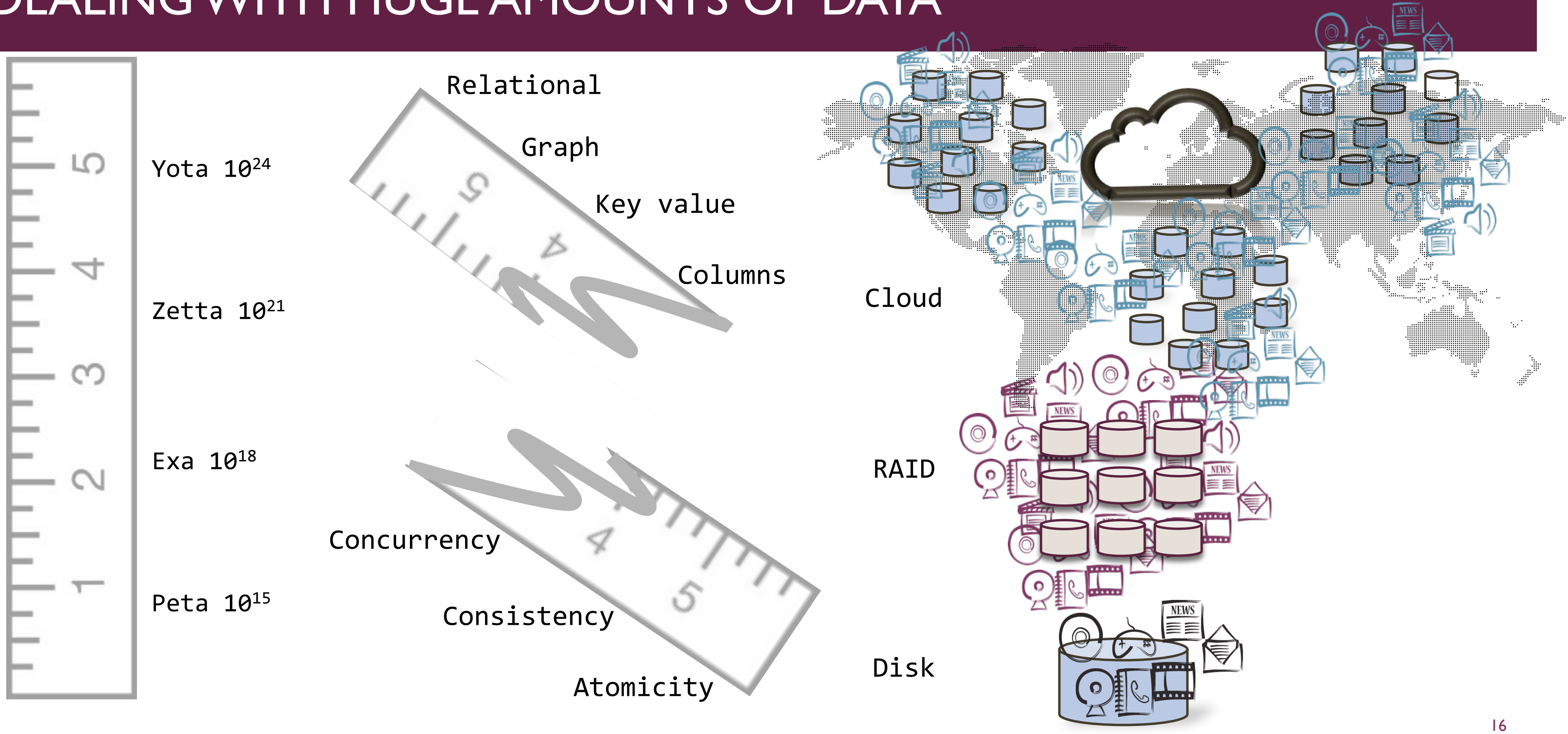
DATA DISTRIBUTION





- sharding a relational DBMS
- dealing with shards synchronization
- deploying the service on the cloud and implement an interface

DEALING WITH HUGE AMOUNTS OF DATA



so now we have NoSQL databases

There is no standard definition of what NoSQL means. The term began with a workshop organized in 2009, but there is much argument about what databases can truly be called NoSQL.

But while there is no formal definition, there are some common characteristics of NoSQL databases

- they don't use the relational data model, and thus don't use the SQL language
- they tend to be designed to run on a cluster
- they tend to be Open Source
- they don't have a fixed schema, allowing you to store any data in any record

examples include

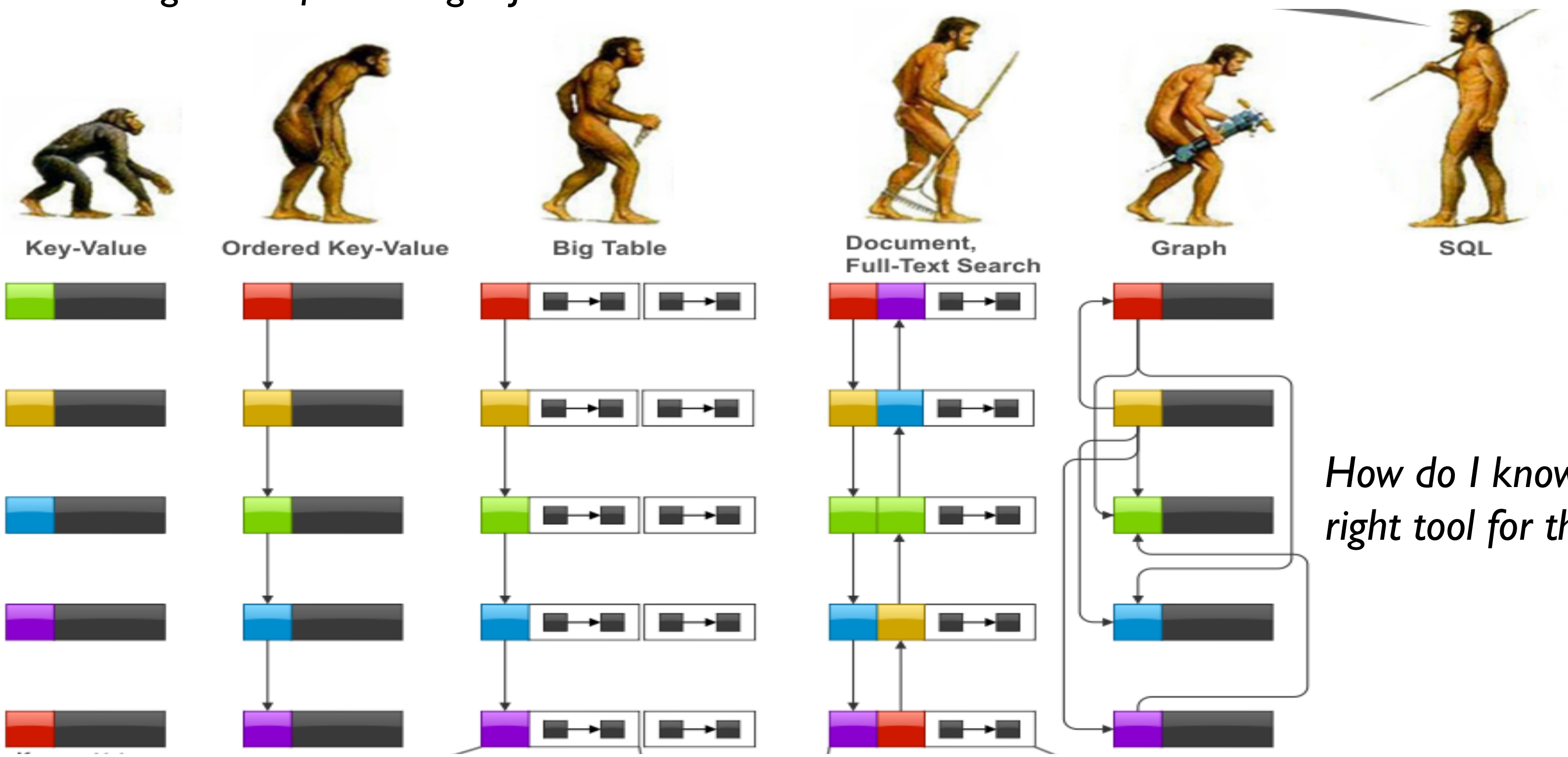
We should also remember Google's Bigtable and Amazon's SimpleDB. While these are tied to their host's cloud service, they certainly fit the general operating characteristics

POLYGLOT PERSISTENCE

- ***Polyglot Programming:*** applications should be written in a mix of languages to take advantage of different languages are suitable for tackling different problems
- ***Polyglot persistence***: any decent sized enterprise will have a variety of different data storage technologies for different kinds of data
  - a new strategic enterprise application should no longer be built assuming a relational persistence support
  - the relational option might be the right one - but you should seriously look at other alternatives

M. Fowler and P. Sadalage. *NoSQL Distilled: A Brief Guide to the Emerging World of Polyglot Persistence*. Pearson Education, Limited, 2012

Use the right tool for the right job…



How do I know which is the right tool for the right job?

(Katsov-2012)

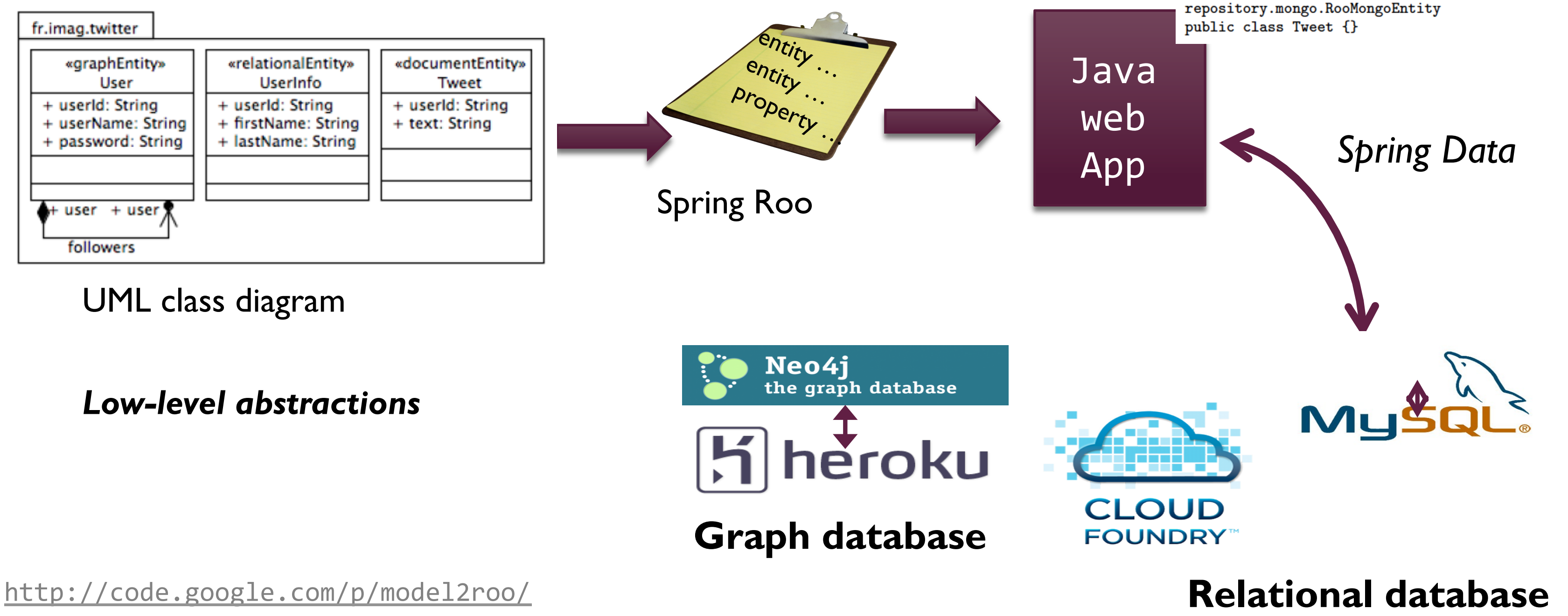
WHEN IS POLYGLOT PERSISTENCE PERTINENT?

- Application essentially composing and serving web pages
  - They only looked up page elements by ID, they had **no need for transactions**, and **no need to share** their database
  - A problem like this is much better suited to a key-value store than the corporate relational hammer they had to use
- **Scaling** to lots of traffic gets harder and harder to do with **vertical scaling**
  - Many NoSQL databases are designed to operate over clusters
  - They can **tackle larger volumes of traffic and data** than is realistic with a single server

GENERATING NOSQL PROGRAMS FROM HIGH LEVEL ABSTRACTIONS

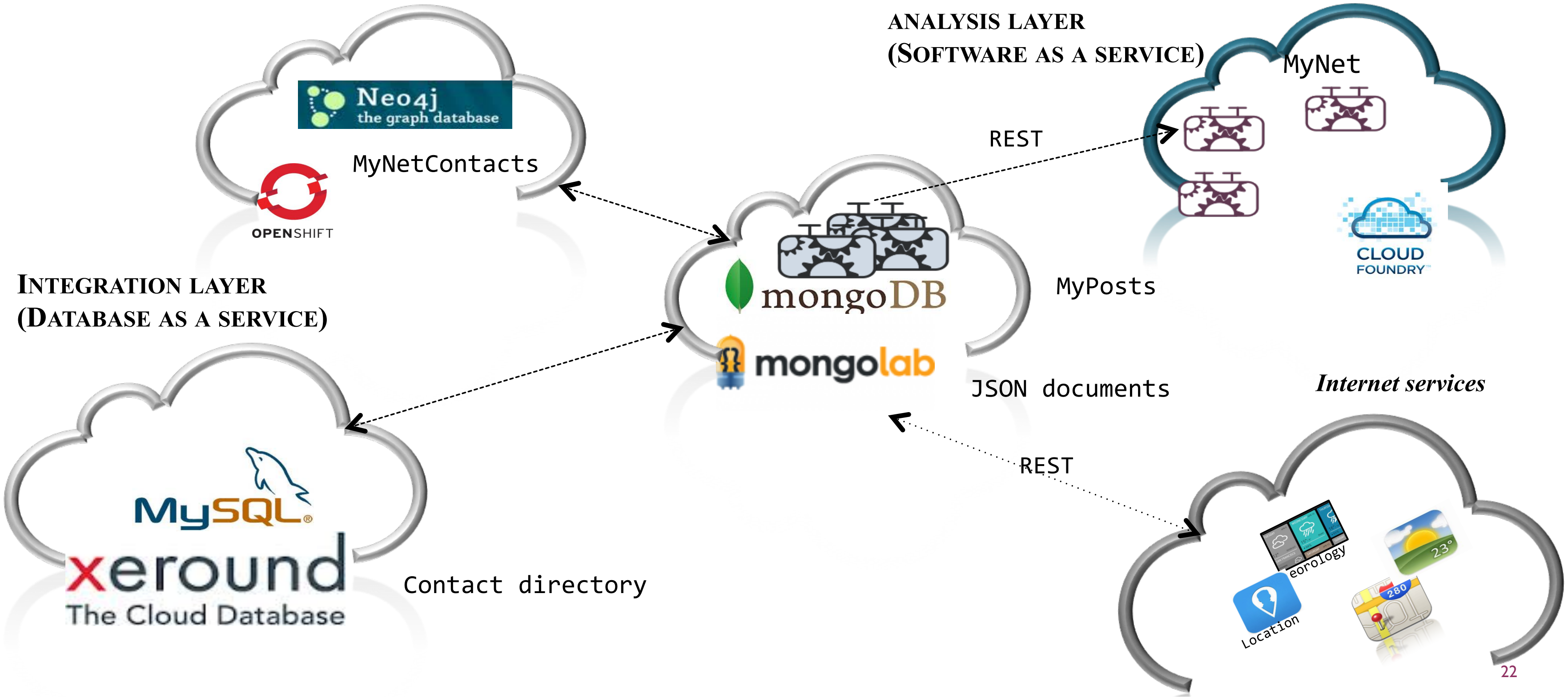
```
@org.springframework.data.neo4j.annotation.NodeEntity
public class User {}

@org.springframework.roo.addon.jpa.
activerecord.RooJpaActiveRecord
public class UserInfo {}

@org.springframework.roo.addon.layers.
repository.mongo.RooMongoEntity
public class Tweet {}
```

High-level abstractions



http://code.google.com/p/model2roo/

POLYGLOT APPROACH TO PERSISTENCE



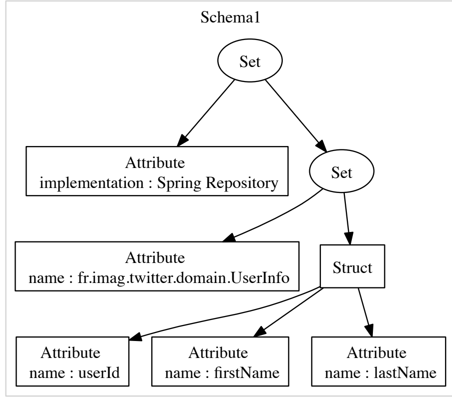
EXTRACTING SCHEMATA FROM NOSQL PROGR

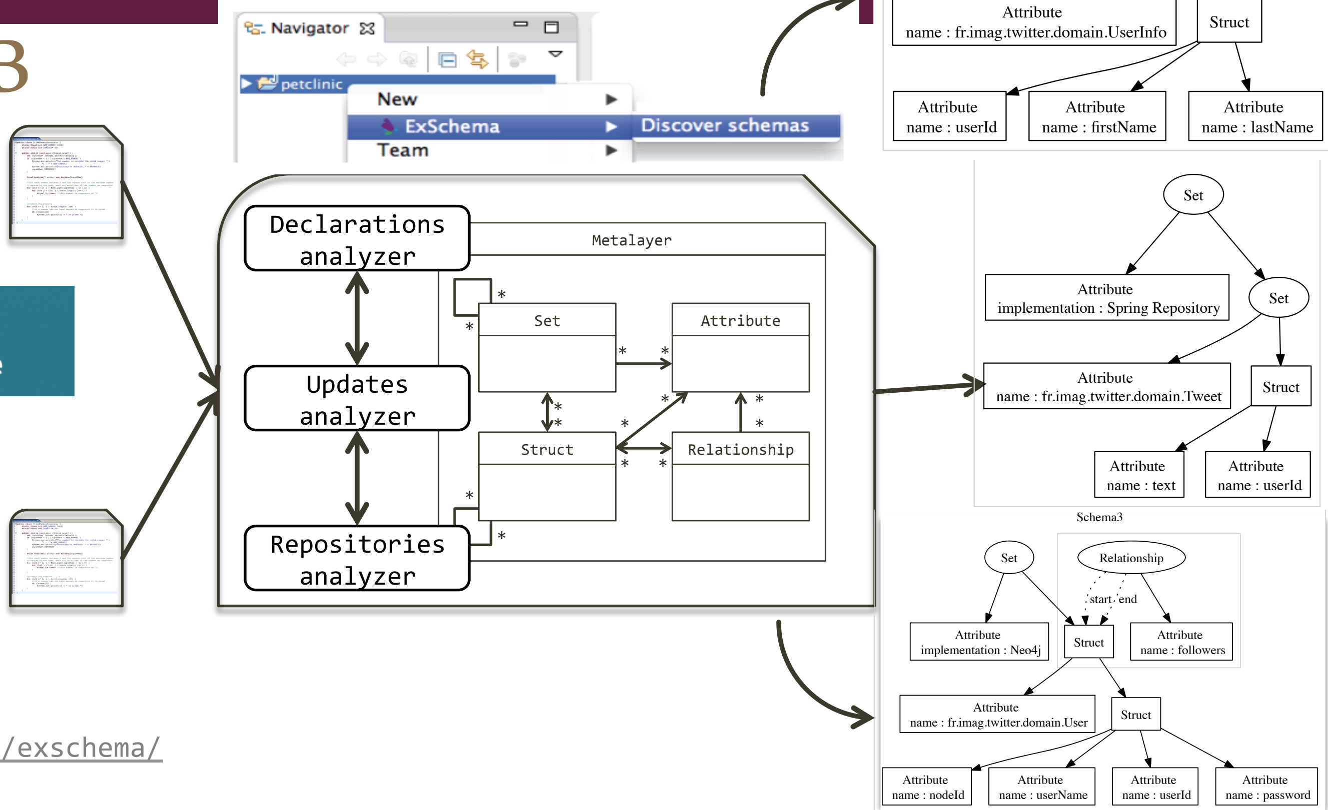
mongoDB

APACHE HBASE

Neo4j the graph database

Spring Data





http://code.google.com/p/exschema/

AGENDA

- *Big data all around*
- *Processing and exploiting big data*
  - *Cloud architectures*
  - *Storage vs. Access*
  - *DBMS Architectures*
- *Big data economy*
- *Outlook*

HOW ARE THIS DATA EXPLOITED?

Domain : Meteorology, genomics, complex physics simulations, biological and environmental research, Internet search, Web mining, Social networks, finance and business informatics, Archaeological, Culture

- Tesco collects 1.5 billion pieces of data every month and uses them to adjust prices and promotions
- Williams-Sonoma uses its knowledge of its 60 million customers (which includes such details as their income and the value of their houses) to produce different iterations of its catalogue
- 30% of Amazon's sales are generated by its recommendation engine ("you may also like")
- Placecast is developing technologies that allow them to track potential consumers and send them enticing offers when they get within a few yards of a Starbucks

“*LIEU COMUN*”

MASSIVE DATA PROCESSING: SAVAGE PARALLELISM

EXECUTION MODEL: MAP-REDUCE

- Programming model for expressing distributed computations on massive amounts of data and an execution framework for large-scale data processing on clusters of commodity servers
- Open-source implementation called Hadoop, whose development was led by Yahoo (now an Apache project)
- Divide and conquer principle: partition a large problem into smaller subproblems
  - To the extent that the sub-problems are independent, they can be tackled in parallel by different workers
  - intermediate results from each individual worker are then combined to yield the final output
- Large-data processing requires bringing data and code together for computation to occur:

→ no small feat for datasets that are terabytes and perhaps petabytes in size!

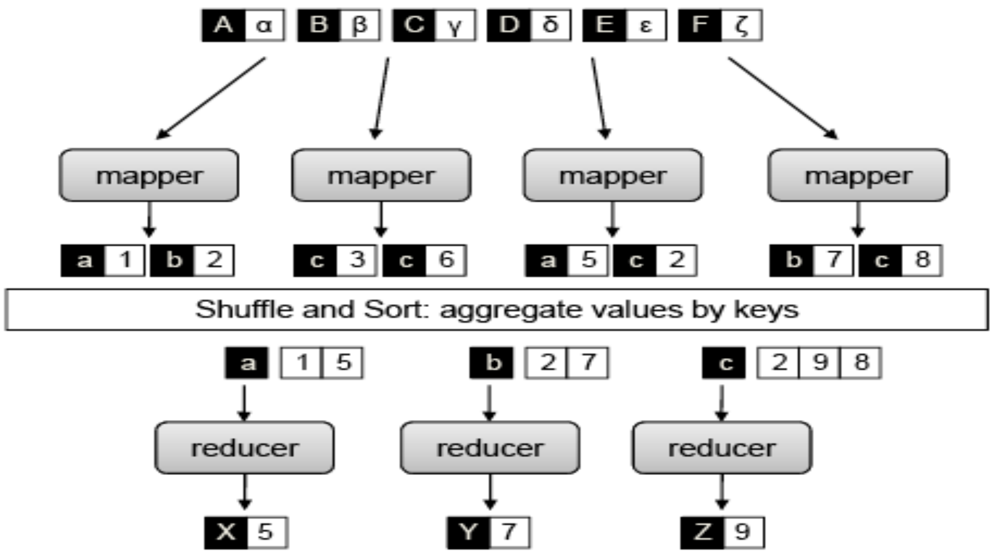
MAP-REDUCE ELEMENTS

$$\text{map: } (k_1, v_1) \longrightarrow [(k_2, v_2)]$$
$$\text{reduce: } (k_2, [v_2]) \longrightarrow [(k_3, v_3)]$$

- **Stage 1**: Apply a user-specified computation over all input records in a dataset.
  - These operations occur in parallel and yield intermediate output (key-value couples)
- **Stage 2**: Aggregate intermediate output by another user-specified computation
  - Recursively applies a function on every pair of the list
- Execution framework coordinates the actual processing
- Implementation of the programming model and the execution framework

MAP REDUCE EXAMPLE

Count the number of occurrences of every word in a text collection



EXECUTION FRAMEWORK

- Important idea behind MapReduce is separating the **what** of distributed processing from the **how**
- A MapReduce program (job) consists of
  - code for mappers and reducers packaged together with
  - configuration parameters (such as where the input lies and where the output should be stored)
- Execution framework responsibilities: scheduling
  - Each MapReduce job is divided into smaller units called tasks
  - In large jobs, the total number of tasks may exceed the number of tasks that can be run on the cluster concurrently → manage tasks queues
  - Coordination among tasks belonging to different jobs

Shrek: For your information, there's a lot more to ogres than people think.
Donkey: Example?
Shrek: Example... uh... ogres are like onions!
(holds up an onion, which Donkey sniffs)
Donkey: They stink?
Shrek: Yes... No!
Donkey: Oh, they make you cry?
Shrek: No!
Donkey: Oh, you leave 'em out in the sun, they get all brown, start sproutin' little white hairs...
Shrek: *(peels an onion)* NO! Layers. Onions have layers. Ogres have layers. Onions have layers. You get it? We both have layers.
(walks off)
Donkey: Oh, you both have LAYERS. Oh. You know, not everybody like onions. What about cake? Everybody loves cake!
Shrek: I don't care what everyone else likes! Ogres are not like cakes.
– *from the 2001 Dreamworks movie "Shrek"* [6]

GOING FOR OGRES, ONIONS OR PARFAITS?

Vinayak borkar, Michael J. Carey, Chen Li, Inside "big data management": ogres, onions, or parfaits?, EDBT, 2012

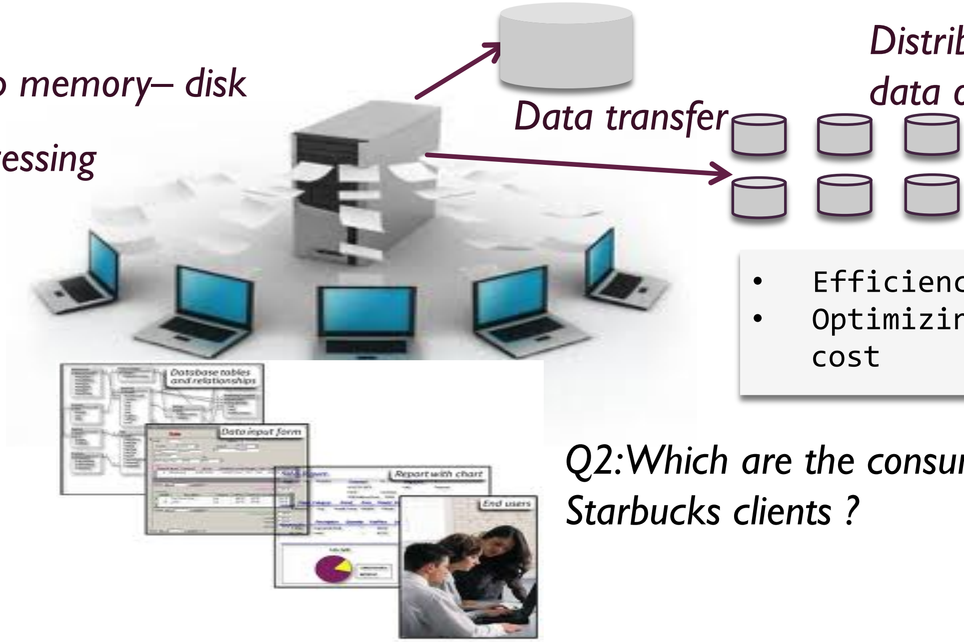
SLICING “*PARFAITS*”

QUERYING ISSUES

… WITH RESOURCES CONSTRAINTS

Query and data processing on server

Swap memory– disk



Distribution and organization of data on disk

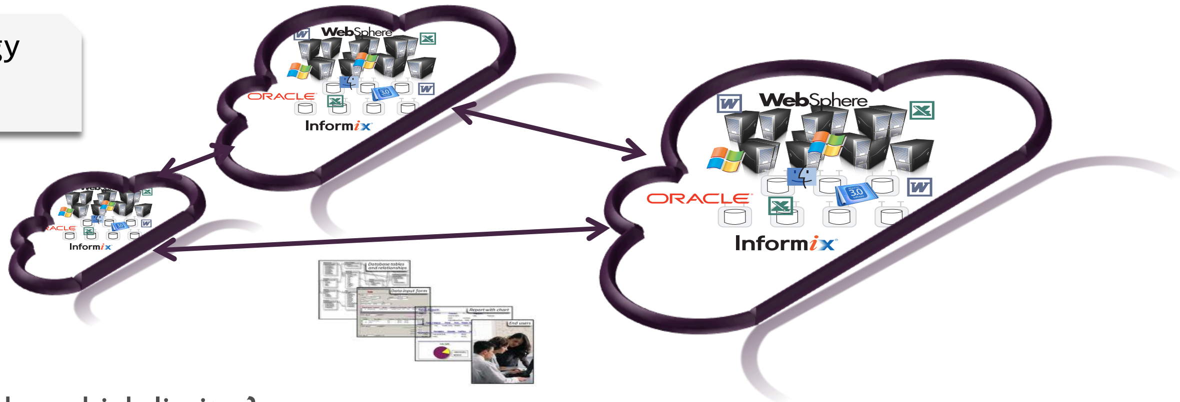
```
•   Efficiency => time cost
•   Optimizing memory and computing
    cost
```

Q2:Which are the consumption rules of Starbucks clients ?

Efficiently manage and exploit data sets according to given specific storage, memory and computation resources

WITHOUT RESOURCES CONSTRAINTS …

Costly => minimizing cost, energy consumption

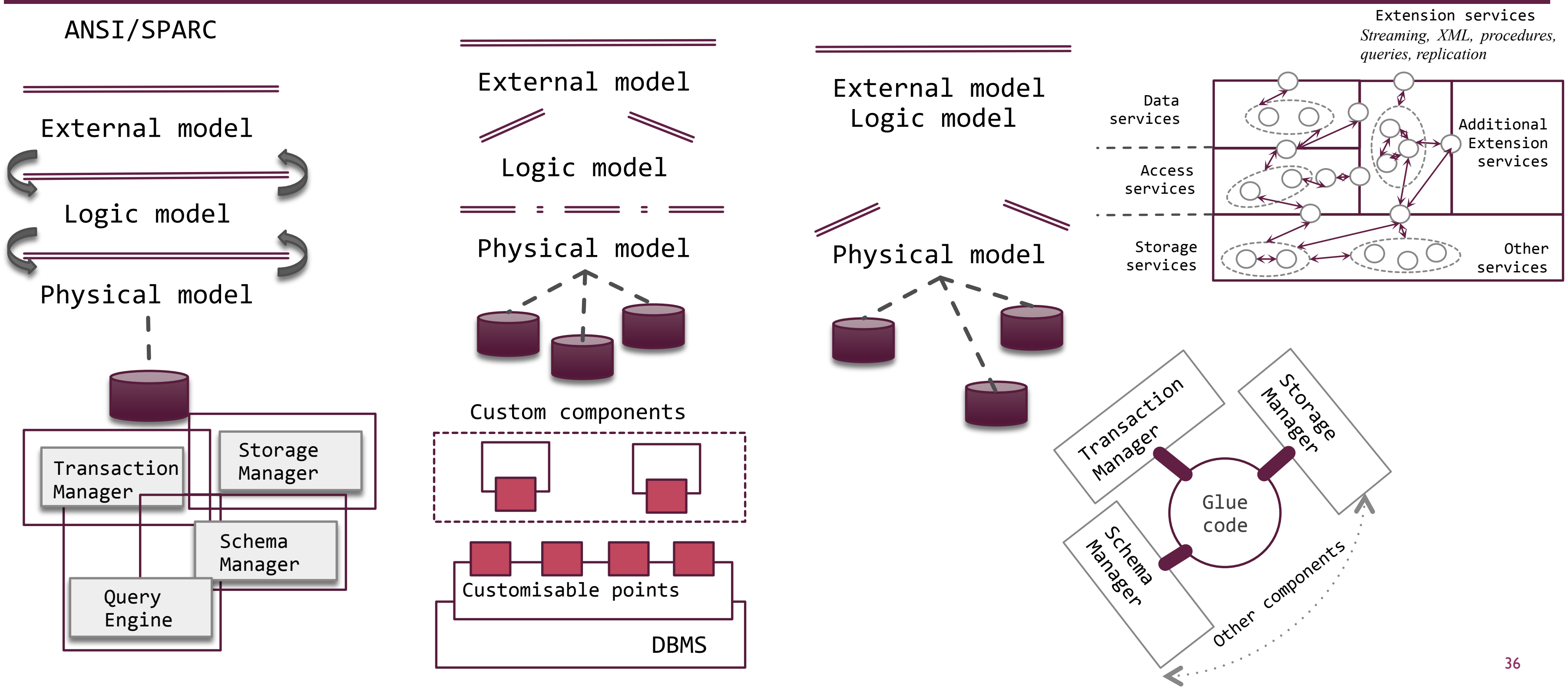


- Query evaluation→ How and under which limits ?
  - Is not longer completely constraint by resources availability: computing, RAM, storage, network services
  - Decision making process determined by resources consumption and consumer requirements
- Data involved in the query, particularly in the result can have different costs: top 5 gratis and the rest available in return to a credit card number
- Results storage and exploitation demands more resources

AGENDA

- *Big data all around*
- *Processing and exploiting big data*
  - *Cloud architectures*
  - *Storage vs. Access*
  - *DBMS Architectures*
- *Big data economy*
- *Outlook*

DATA MANAGEMENT SYSTEMS ARCHITECTURES



B-TREE: PHYSICAL LEVEL

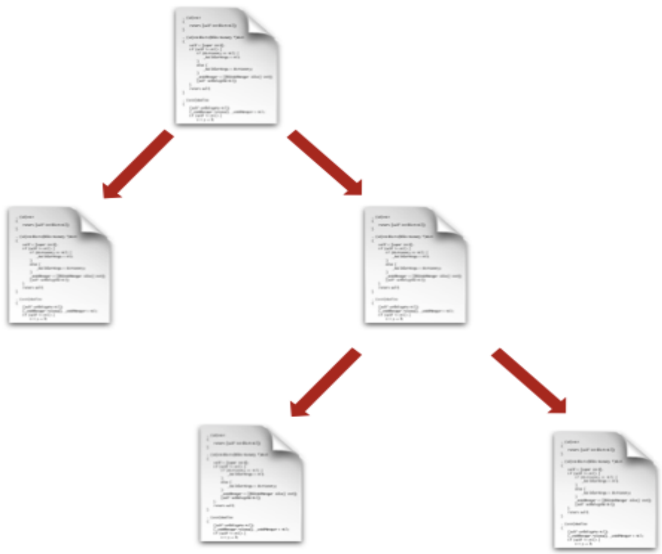
Append only

Concurrency (MVCC)

Crash resistant

Hot backups

Compaction



peritor consulting

«*MEMCACHED*»

- «*memcached*» is a memory management protocol based on a cache:
  - Uses the key-value notion
  - Information is completly stored in RAM

- «*memcached*» protocol for:
  - Creating, retrieving, updating, and deleting information from the database
  - Applications with their own «*memcached*» manager (Google, Facebook, YouTube, FarmVille, Twitter, Wikipedia)

STORAGE ON DISC (1)

- For efficiency reasons, information is stored using the RAM:
  - Work information is in RAM in order to answer to low latency requests
  - Yet, this is not always possible and desirable

➢ **The process of moving data from RAM to disc is called "*eviction*"; this process is configured automatically for every bucket**

STORAGE ON DISC (2)

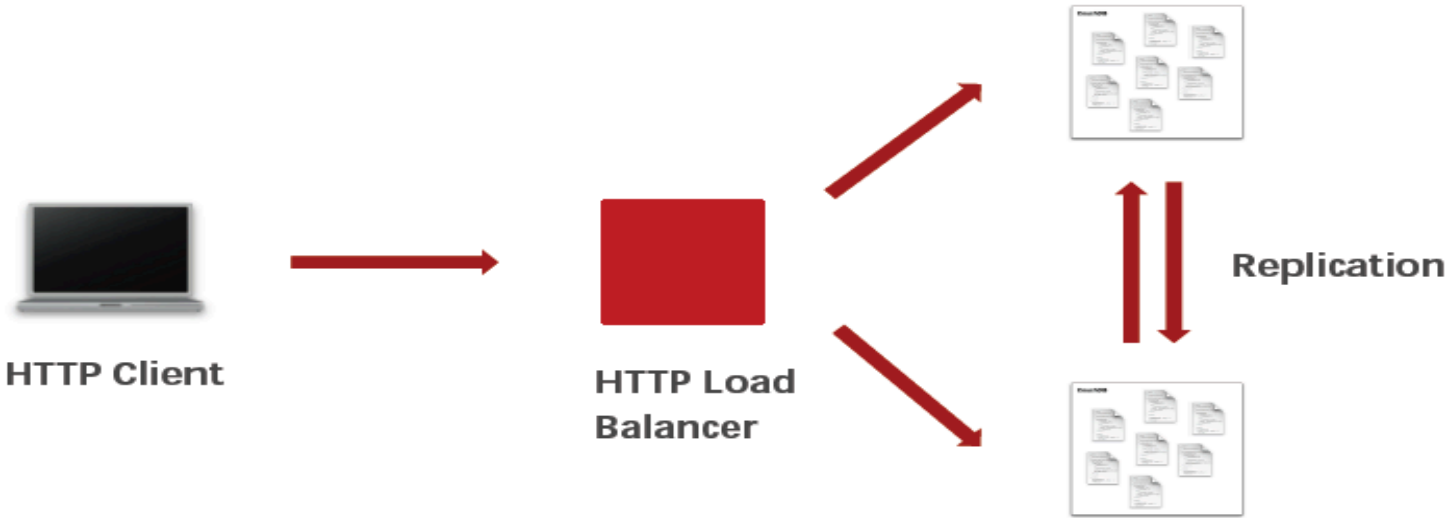
- NoSQL servers support the storage of key-value pairs on disc:
  - **Persistency**–can be executed by loading data, closing and reinitializing it without having to load data from another source
  - **Hot backups**– loaded data are sotred on disc so that it can be reinitialized in case of failures
  - **Storage on disc**– the disc is used when the quantity of data is higher thant the physical size of the RAM, frequently used information is maintained in RAM and the rest es stored on disc

STORAGE ON DISC (3)

- Strategies for ensuring:
  - Each node maintains in RAM information on the key-value pairs it stores. Keys:
    - may not be found, or
    - they can be stored in memory or on disc
  - The process of moving information from RAM to disc is asynchronous:
    - The server can continue processing new requests
    - A queue manages requests to disc

➢ **In periods with a lot of writing requests, clients can be notified that the server is termporaly out of memory until information is evicted**

REPLICATION



AGENDA

- *Big data all around*
- *Processing and exploiting big data*
  - *Cloud architectures*
  - *Storage vs. Access*
  - *DBMS Architectures*
- *Big data economy*
- *Outlook*

BIG DATA TECH ECONOMY

Big data is not a "thing" but instead a dynamic/activity that crosses many IT borders

- Data-driven world guided by a rapid ROD (**Return on Data**)
  - → Generate value by extracting the right information from the digital universe
- The key is how quickly data can be turned in to currency by:
  - Analysing patterns and spotting relationships/trends that enable decisions to be made faster with more precision and confidence
  - Identifying actions and bits of information that are out of compliance with company policies can avoid millions in fines
  - Proactively reducing the amount of data you pay ($18,750/gigabyte to review in eDiscovery) by identifying only the relevant pieces of information
  - Optimizing storage by deleting or offloading non-critical assets to cheaper cloud storage thus saving millions in archive solutions

http://www.emc.com/collateral/analyst-reports/idc-extracting-value-from-chaos-ar.pdf

GIVING VALUE TO DATA: DATA MARKETS

- *Data market*: economically capitalize the effort of going out for hunting data sources and services of different qualities and stemming from different processing processes, curating and delivering them
- The key issues here are:
  - (i) being able to associate a cost model for the data market, i.e., associate a cost to raw data and to processed data according on the amount of data and processing resources used for treating it, for instance
  - (ii) then being able to combine these cost model and the consumer expectations (service level agreement) with processing resources cost required by data processing
  - (iii) providing data management and brokering processing mechanisms under ad hoc business models

GIVING VALUE TO DATA

- Economy oriented data brokering will be tuned by the cost of accessing data markets, and the cost of using resources for brokering data
- Manage the computation of results versus the cost of accessing completely of partially such results according to their
  - *Completeness*: a sample of resulting data that can be completed according to an economic model: guided by predefined fees (1M 10euros, 10M 15 euros), the user specifies whether to buy data to complete results.
  - *Data delivery frequency*: new data can be delivered, while a data market proposes new data. It is up to the user to subscribe according to different fees.
  - *Duration*: volatile/persistent results produced out of series of queries can be used for building a new data market. The owner can require accessing and buying storage services for dealing with her data markets and exporting them as paying services. The associated cost of such service will depend on the QoS and the kind of Service level agreements that the service can honour.
  - *Content quality*: data provenance, data freshness and degree of aggregation.

AGENDA

- *Big data all around*
- *Processing and exploiting big data*
  - *Cloud architectures*
  - *Storage vs. Access*
  - *DBMS Architectures*
- *Big data economy*
- *Outlook*

OUTLOOK

- Big data open issues: we are struggling to capture, storage, search, share, analyze, and visualize it
  - Current technology is not adequate to cope with such large amounts of data (requiring massively parallel software running on tens, hundreds, or even thousands of servers)
  - Expertise in a wide range of topics including
    - Machine architectures (HPC), Service-oriented-architecture
    - Distributed/cloud computing, operating and database systems
    - Software Engineering, algorithmic techniques and networking

"Si vous faîtes attention aux signes, quand donc ferez vous attention à ce qu'ils signifient?"
—François Rabelais

Thanks Merci Gracias

Contact: Genoveva Vargas-Solar, CNRS, LIG-LAFMIA

Genoveva.Vargas@imag.fr

http://www.vargas-solar.com

Open source polyglot persistence tools

http://code.google.com/p/exschema/

http://code.google.com/p/model2roo/

Want to put cloud data management in practice ?

cf. http://www.vargas-solar.com/cloud-data-management/