NoSQL & NewSQL

Instructors: Peter Baumann

email: p.baumann@jacobs-university.de
tel: -3178
office: room 88, Research 1

With material by Willem Visser

We Don‘t Want No SQL !

- NoSQL movement: SQL considered slow → only access by id („lookup“)
  - Deliberately abandoning relational world: „too complex“, „not scalable“
  - No clear definition, wide range of systems
  - Values considered black boxes (documents, images, ...)
  - simple operations (ex: key/value storage), horizontal scalability for those
  - ACID → CAP, „eventual consistency“
- Systems

  documents | columns | key/values

  - Open source: MongoDB, CouchDB, Cassandra, HBase, Riak, Redis
  - Proprietary: Amazon, Oracle, Google , Oracle NoSQL
- See also: http://glennas.wordpress.com/2011/03/11/introduction-to-nosql-john-nunemaker-presentation-from-june-2010/

Structural Variety in Big Data

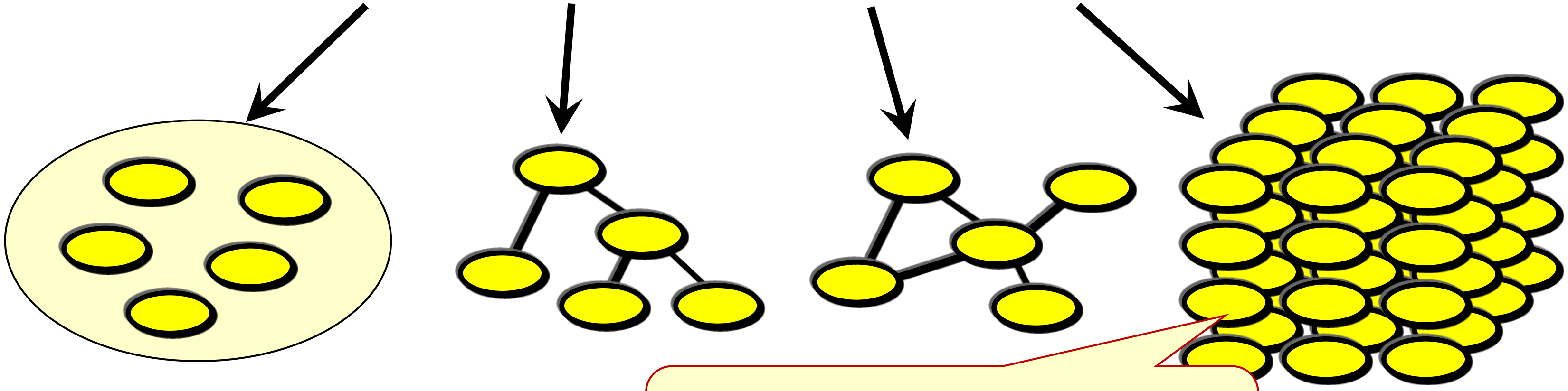
- Stock trading: 1-D sequences (i.e., arrays)
- Social networks: large, homogeneous graphs
- Ontologies: small, heterogeneous graphs
- Climate modelling: 4D/5D arrays
- Satellite imagery: 2D/3D arrays (+irregularity)
- Genome: long string arrays
- Particle physics: sets of events
- Bio taxonomies: hierarchies (such as XML)
- Documents: key/value stores = sets of unique identifiers + whatever
- etc.

Structural Variety in Big Data

- Stock trading: 1-D sequences (i.e., arrays)
- Social networks: large, homogeneous graphs
- Ontologies: small, heterogeneous graphs
- Climate modelling: 4D/5D arrays
- Satellite imagery: 2D/3D arrays (+irregularity)
- Genome: long string arrays
- Particle physics: sets of events
- Bio taxonomies: hierarchies (such as XML)
- Documents: key/value stores = sets of unique identifiers + whatever
- etc.

Structural Variety in Big Data

sets + hierarchies + graphs + arrays



sensor, image [timeseries], simulation, statistics data

Ex: Key/Value Store

- Conceptual model: key/value store = set of key+value
  - Operations: *Put(key,value), value = Get(key)*
  - → large, distributed hash table
- Needed for:
  - twitter.com: tweet id -> information about tweet
  - kayak.com: Flight number -> information about flight, e.g., availability
  - amazon.com: item number -> information about it
- Ex: Cassandra (Facebook; open source)
  - Myriads of users, like:

Document Stores

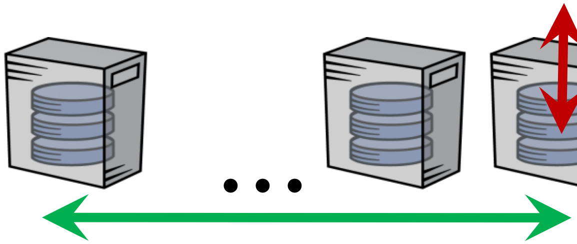
- Like key/value, but value is a complex document
- Added: Search functionality within document
  - Fulltext search: Lucene/Solr, ElasticSearch...
  - Can support this in architecture, eg, full-text index
- Need: content oriented applications
  - Facebook, Amazon, …
- Ex: MongoDB, CouchDB

Performance Comparison

- On > 50 GB data:
- MySQL
  - Writes 300 ms avg
  - Reads 350 ms avg
- Cassandra
  - Writes 0.12 ms avg
  - Reads 15 ms avg

How To Make It Fast: 2 x 2 x 2

- 2 kinds of scalability:
  - horizontal scaling over multiple servers
  - vertical scaling for performance on a single server



- Key features needed to achieve this:
  - For horizontal scaling:
    - partition and replicate
    - automatic failure recovery, database evolution w/o downtime
  - For vertical scaling:
    - RAM, avoid random disk I/O
    - minimize overhead for locking & latching, minimize network calls between servers

[Rick Cattell, http://www.cattell.net/datastores/ScalabilityRequirements.html]

Giving Up ACID

- RDBMS provide ACID
- Cassandra provides BASE
  - Basically Available Soft-state Eventual Consistency
  - Prefers availability over consistency

CAP Theorem

- Proposed by Eric Brewer, UCB; subsequently proved by Gilbert & Lynch
- In a distributed system you can satisfy at most 2 out of the 3 guarantees
  - Consistency: all nodes have same data at any time
  - Availability: system allows operations all the time
  - Partition-tolerance: system continues to work in spite of network partitions
- Traditional RDBMSs
  - Strong consistency over availability under a partition
- Cassandra
  - Eventual (weak) consistency, Availability, Partition-tolerance

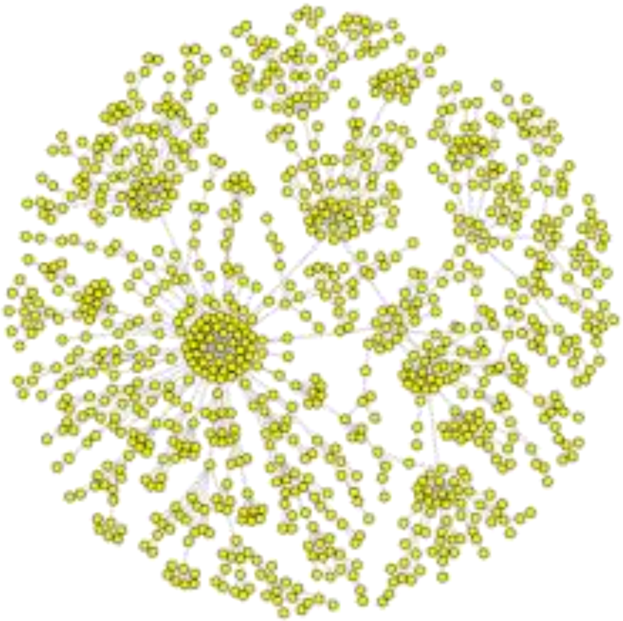
NoSQL

- Previous „young radicals“ approaches subsumed under „NoSQL“
- = we want „no SQL“
- Well...„not only SQL“
  - After all, a QL is quite handy
  - So, QLs coming into play again (and 2-phase commits = ACID!)
- Ex: MongoDB: „tuple“ = JSON structure

```
db.inventory.find(
  {   type: 'food',
      $or: [ { qty: { $gt: 100 } }, { price: { $lt: 9.95 } } ]
  }   )
```

Ex 1: Graph Store

- Conceptual model: Labeled, directed, attributed multi-graph
  - Multi-graph = multiple edges between nodes
- Needed by: social networks



[blog.revolutionanalytics.com]

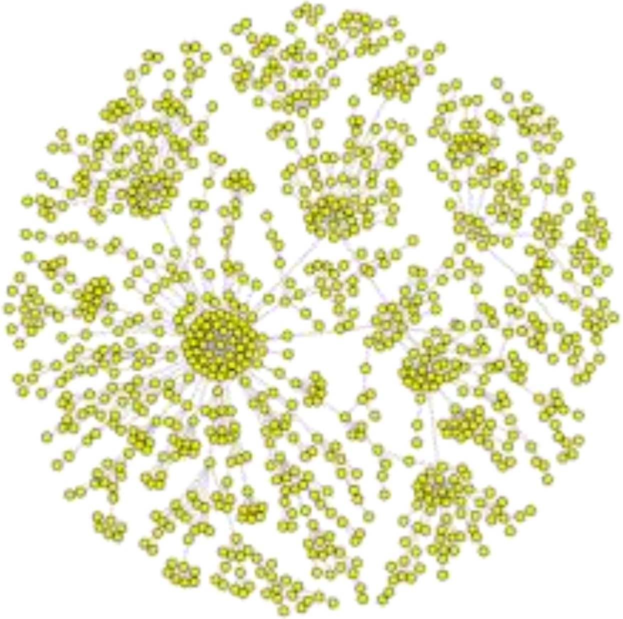
Ex 1: Graph Store

facebook
December 2010

[blog.revolutionanalytics.com]

Ex 1: Graph Store

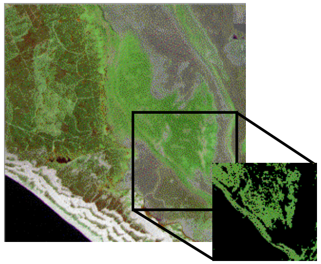
- Conceptual model: Labeled, directed, attributed multi-graph
  - Multi-graph = multiple edges between nodes
- Needed by: social networks
  - My friends, who has no / many followers, closed communities, new agglomerations, new themes, ...
- Sample system: Neo4j
- Why not relational DB? can model graphs!
  - but “endpoints of an edge” already requires (expensive) join
  - No support for global ops like transitive hull



Ex 2: Array Databases

- Array DBMSs for declarative queries on massive n-D arrays
  - Ex: rasdaman = Array DBMS for massive n-D arrays

```
select img.green[x0:x1,y0:y1] > 130
from   LandsatArchive as img
```



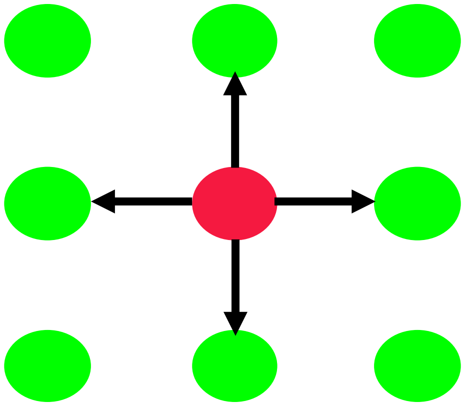
- Array DBMSs can be 200x RDBMS [Cudre-Maroux]
- Demo at http://standards.rasdaman.org

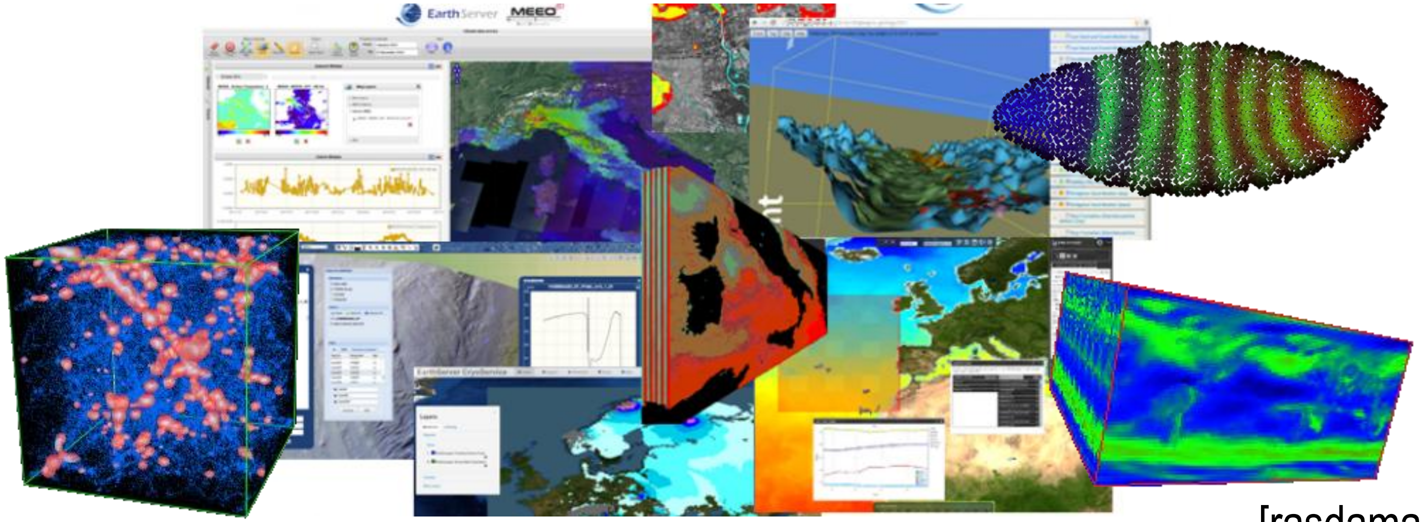
The Earth Monitoring Competition
Copernicus masters
WINNER
T-SYSTEMS BIG DATA CHALLENGE
2014
T · · Systems ·

Array Analytics

- **Array Analytics** :=
  Efficient analysis on multi-dimensional arrays of a size several orders of magnitude above the evaluation engine‘s main memory

- Essential property: *n*-D Euclidean neighborhood





[rasdaman]

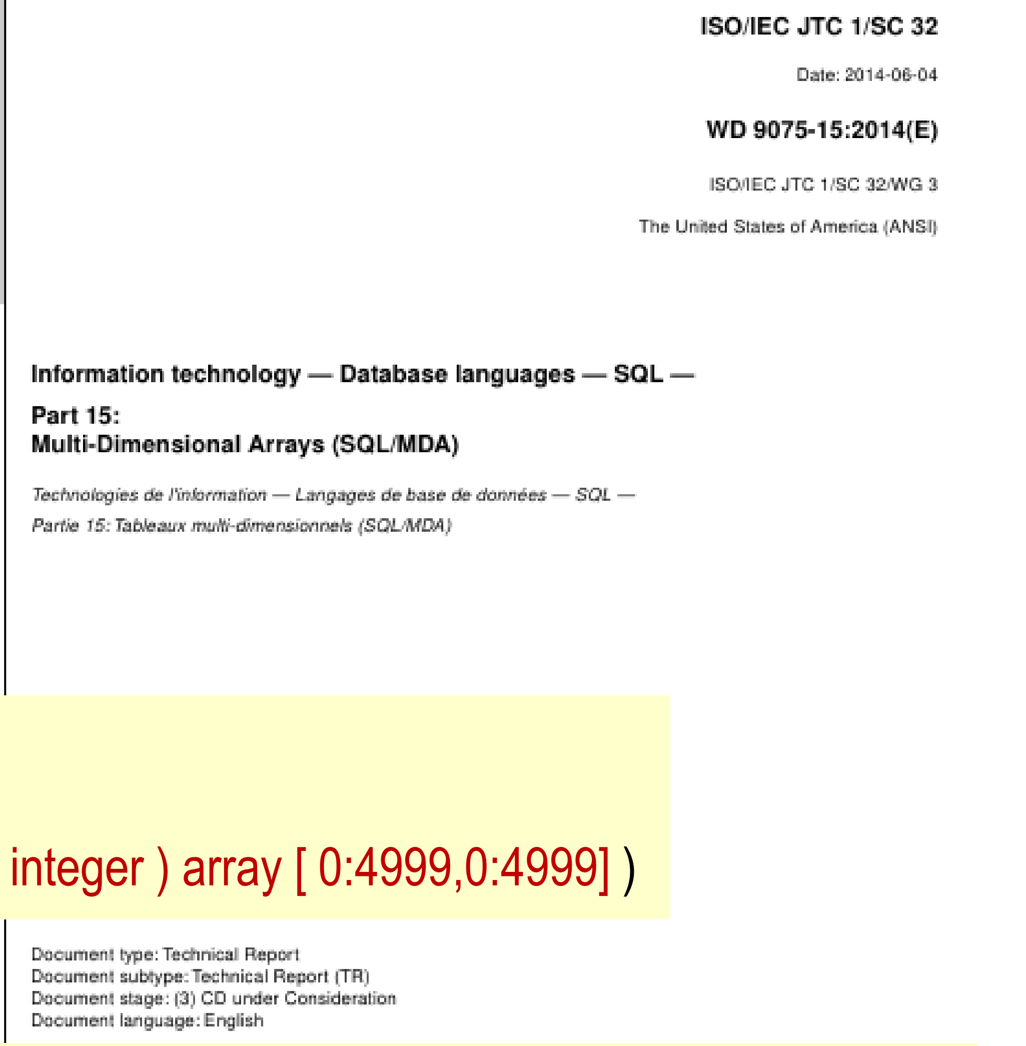
ISO Array SQL

- ISO 9075 Part 15: SQL/MDA
  - resolved by ISO SQL WG in June 2014
- n-D arrays as attributes

```
create table LandsatScenes(
   id: integer not null, acquired: date,
   scene: row( band1: integer, ..., band7: integer ) array [ 0:4999,0:4999] )
```

- declarative array operations

```
select  id, encode(scene.band1-scene.band2)/(scene.nband1+scene.band2)), „image/tiff“ )
from    LandsatScenes
where acquired between „1990-06-01“ and „1990-06-30“ and
          avg( scene.band3-scene.band4)/(scene.band3+scene.band4)) > 0
```



ISO/IEC JTC 1/SC 32

Date: 2014-06-04

WD 9075-15:2014(E)

ISO/IEC JTC 1/SC 32/WG 3

The United States of America (ANSI)

Information technology — Database languages — SQL —

Part 15:
Multi-Dimensional Arrays (SQL/MDA)

Technologies de l'information — Langages de base de données — SQL —
Partie 15: Tableaux multi-dimensionnels (SQL/MDA)

Document type: Technical Report
Document subtype: Technical Report (TR)
Document stage: (3) CD under Consideration
Document language: English

Ed.)

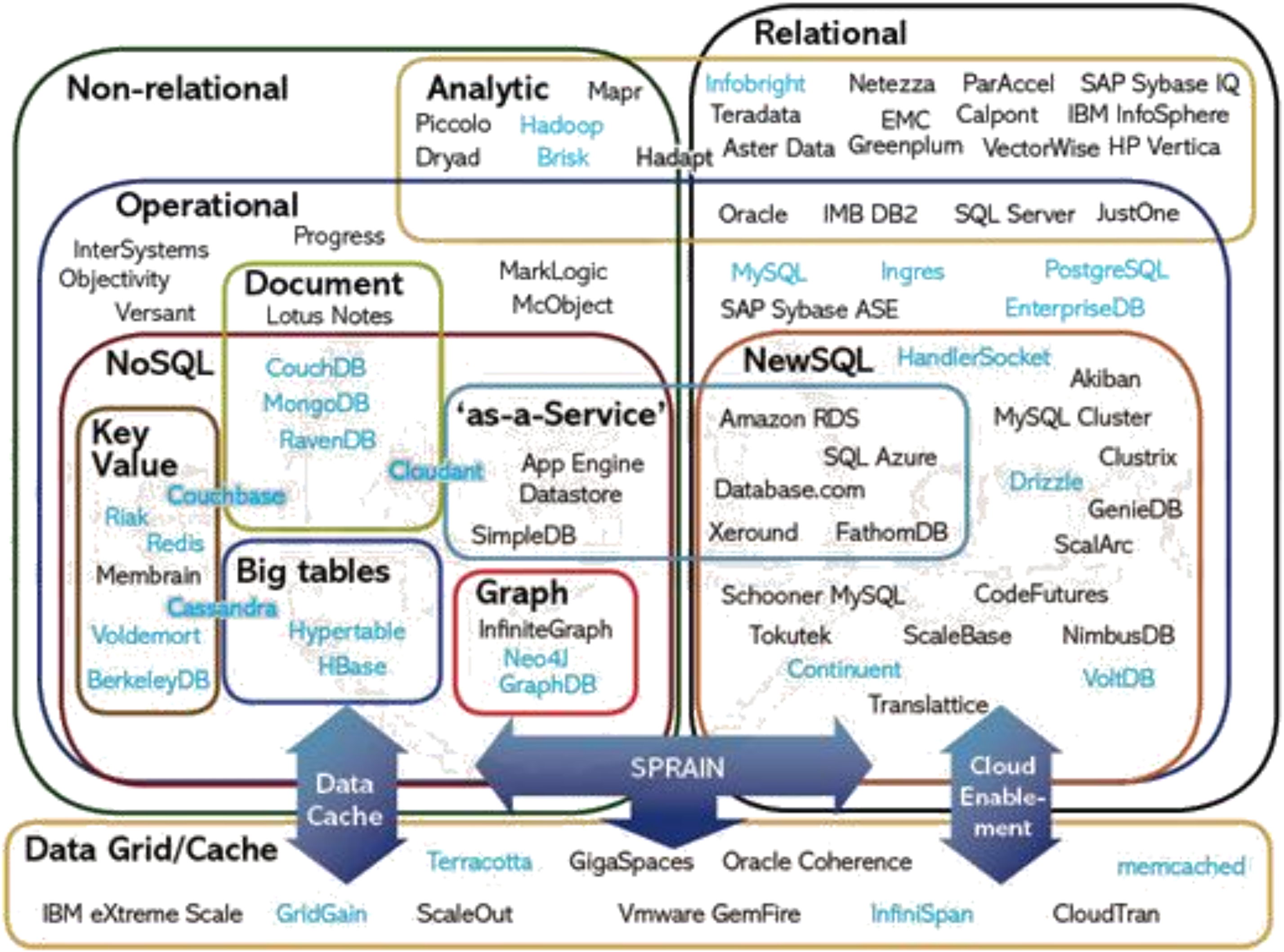
NewSQL

- Michael Stonebraker: „no one size fits all“
- NoSQL: sacrifice functionality for performance – no QL, only key access
- Swinging back from NoSQL:
  declarative QLs considered good, but SQL often inadequate
- Definition 1: NewSQL = SQL with enhanced performance architectures
- Definition 2: NewSQL = SQL enhanced with, eg, new data types
  - Some call this NoSQL

Column-Store Databases

- *The Relational Empire strikes back*
- Observation: fetching long tuples overhead when few attributes needed
- Brute-force decomposition: one value (plus key)
  - Ex: Id+SNLRH → Id+S, Id+N, Id+L, Id+R, Id+H
  - Column-oriented storage: each binary table separate file
- Observation: with clever architecture, reassembly of tuple pays off
- Sample system: MonetDB
  - All major vendors say they have one, but caveat

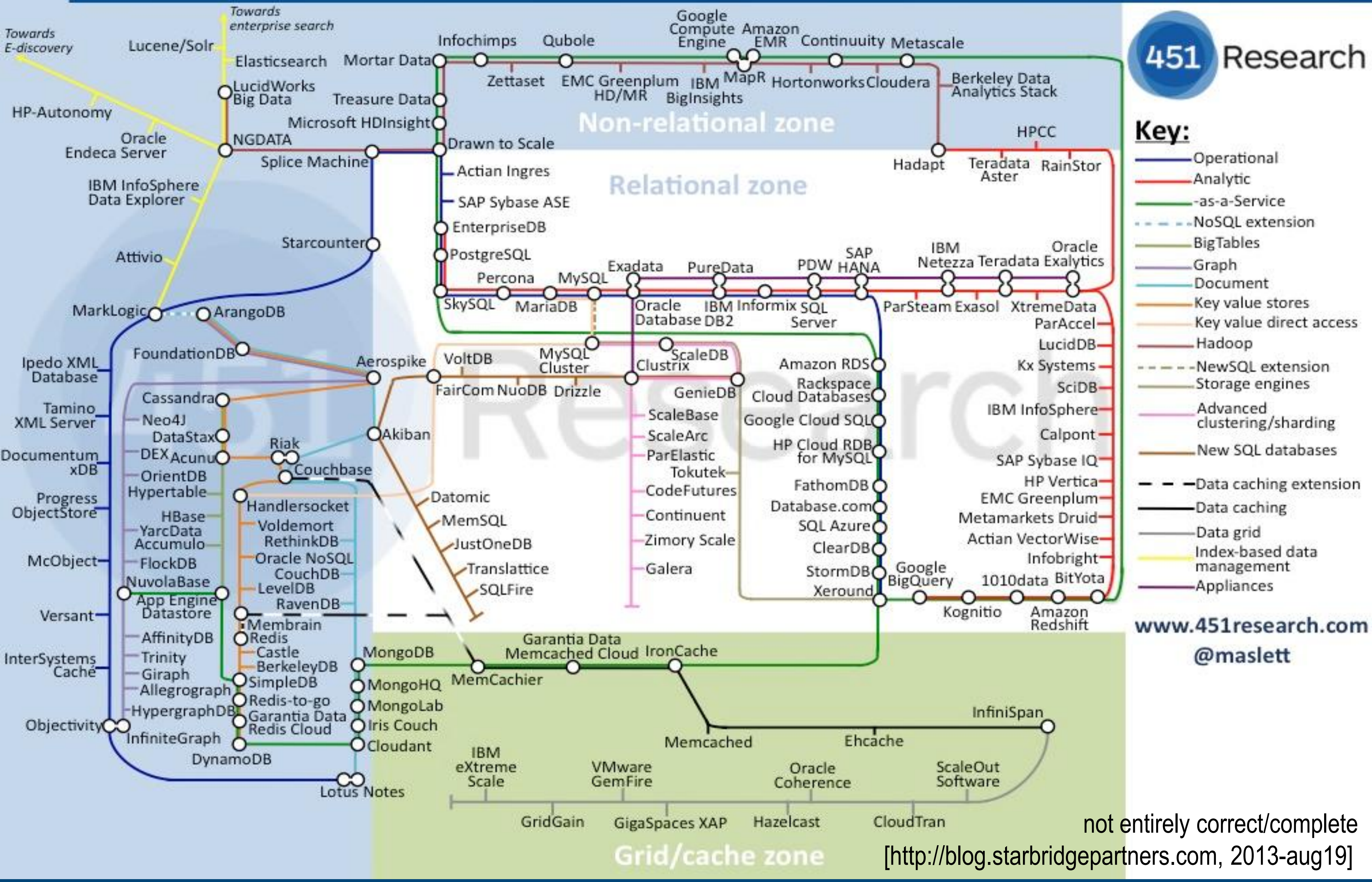
The Explosion of DBMSs



[451 group]

...not entirely correct

Database Landscape Map – December 2012



not entirely correct/complete

[http://blog.starbridgepartners.com, 2013-aug19]

Summary & Outlook

- Fresh approach to scalable data services: NoSQL, NewSQL
  - Diversity of technology → pick best of breed for specific problem
- Avenue 1: Modular data frameworks to coexist
  - Heterogeneous model coupling barely understood - needs research
- Avenue 2: concepts assimilated by relational vendors
  - Like fulltext, object-oriented, SPARQL, ... cf „Oracle NoSQL“
- “SQL-as-a-service”
  - Amazon RDS, Microsoft SQL Azure, Google Cloud SQL
- *More than ever, experts in data management needed !*
  - *More generally: data engineers*