# Biz Drivers

* Takes too long to get information
* Data formats not accessible (not structured or unknown structure, too high variety)
* Data volume grows too rapidly

# When to use noSQL Databases

* Scalability and clustering support (Sharding)
* Data is too big to fit on one machine (within lifetime of developed product)
* Data acquisition performance (too fast for one machine)
* Data analysis performance (Map-Reduce brings algorithm to data, not data to algorithm)  
  🡪 large data to transport over net OR complex operation on each document
* Unstructured data (or unknown structure) 🡪 may be dangerous to maintain!
* License costs (often none due to open source)
* Special algorithms to be used, e.g. graph algos using long traces of same node types (Person 🡪 Person 🡪 Person 🡪 Person 🡪 … 🡪 Something )
* Processing of data is typically a-sync

# When NOT to use noSQL Databases

* “Typical” CRUD-Application
* Know Data-Structure and Data small enough
* Data coherence / integrity must be guaranteed   
  (dependencies between **aggregates**, Foreign Key Constraints)  
  🡪 or graph database
* Benefit from well-established frameworks (JEE / Spring container, OR-Mappers, …)

# Things to consider

* **First Quantify, then decide (Requirements especially for Scalability)**
* Well control distribution of data across cluster nodes
* Costs of communication during data analysis (Map-Reduce on nodes, sql-Functions, …)
* Possible Loss of “ACID”
* Possible more complex mapping (no OR-Mapper support)
* Sophistication of framework support (container managed transactions … )
* Redundancy in document stores, e.g. keys in JSON or enumeration values
* How to address structural changes / typos by architecture, e.g. { name : … } vs. { nAmE: … }
* Avoidance of duplicate data within one document
* Tooling (SQL Workbenches, Analysis tools …)
* It may be an option to use a noSQL DB and a sqlDB in parallel
* Data in silos 🡪 how to address?
* Audit / Traceability capabilities? How to prove?
* Extensibility
* Legal / company constraints (Location of physical data storage)
* Garbage in 🡪 Garbage out
* Test environment

# Various

* Provisioning with ambari (and Vagrant, Puppet) nice, but does not stand a reboot … not clear why?