# CAP

* **C**onsistency, **A**vailability, **P**artitioning tolerance (== Clustering)
* See on [Wikipedia](http://en.wikipedia.org/wiki/CAP_theorem)
* Proven: you can only have 2 at once
* Recommendation: don’t cluster, if there is no real need
* HTTP-Session and EJB-Session is not consistent by default in Application servers
* Nearly no distributed system (e.g. Amazon, Facebook, Banks, others) are consistent – instead they are “eventually consistent”

# FLP

* Asynchronous communication is not always the answer
  + Error handling too complex
  + When waiting for an answer – how long do you wait?
* See on [Wikipedia](http://en.wikipedia.org/wiki/Consensus_(computer_science)#Solvability_results_for_some_agreement_problems)

# Hardware costs

* Hardware does not cost so much any longer
* Use virtualization combined with automization
* New possibilities regarding deployment and testing
* No modularization effort in Java necessary any longer (just deploy everything – but keep deployments small)
* In one word: **private clouds**

# 24x7

* Check if it is really needed
* Reason: reaching real 24x7 greatly involves complexity

# BASE vs. ACID

* **A**tomicity, **C**onsistency, **I**solation, **D**urability
  + CA in CAP Theorem
  + RDBMS cannot scale
* **B**asically **A**vailable **S**oft-state services with **E**ventual-consistency
  + AP in CAP Theorem
  + NoSQL can scale
* Notes on NoSQL databases
  + Writes are fast – but data is distributed async
  + You can achieve consistent reads, if you have time
  + Consistency level can by set by operation (fast or consistent)

# Clean code