# Build

* WAR is the best archive
* Make POMs as small as possible
* Tweak builds with –T flag of Maven (concurrent builds)
* Use CI
* Have a good Jenkins machine
* Check-in very often
* Quest every dependency (keep the WAR small)
* Versioning: no build number – use the POM version
* (Check [www.semver.org](http://www.semver.org))
* Configure Jenkins in Master-Slave mode
* Jenkins with Git => Hook possible that informs Jenkins about the change
* Move integration test logic to CI server

# Versioning

* Use Git
* SVN is slow / Git is fast
* Better concepts in background
* Less problems
* It’s THE new VCS (everybody moves away from SVN)

# RESTful architectures

* SOAP is not object-oriented
* SOAP calls static methods
* REST is about objects == „resources“
* REST allows to read/update the state of resources
* HTTP operations perfectly map to typical object operations
  + GET == Select
  + POST == Insert
  + PUT == Update
  + DELETE == delete
* HTTP status (e.g. 201, 202) allows well-defined return states
* Practical approaches
  + Search operation
    - Define a new resource for searches
    - E.g. GET xmas/v1/searches/[your search]
  + Think in terms (== substantives), NOT operations
    - Make a resource out of every concept
    - E.g.: searches, transactions, …
  + If you need validation – use Bean Validation (no need for XSD validation)
* Examples:
  + Kenai (Sun Cloud API)
  + HornetQ REST API