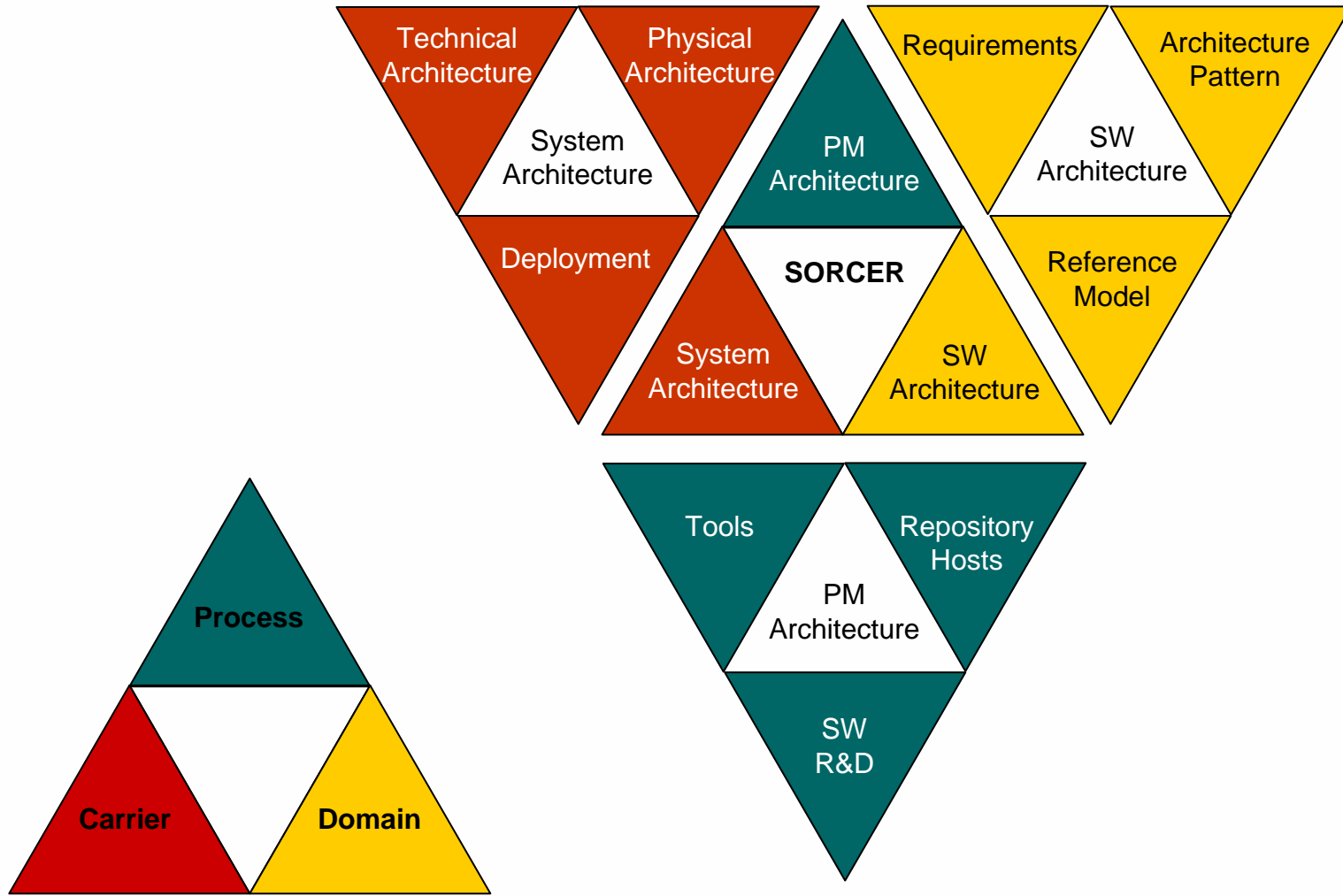


SORCER Project Management Architecture



Michael Sobolewski
sobol@cs.ttu.edu

SORCER & Architecture Views



- CVS
- JDK 5 (Java 5 SE), Java EE 5 SDK
- Jini 2.1
- Eclipse/Emacs/vi
- JUnit
- Apache Ant and Tomcat
- Mailman, the GNU Mailing List Manager
- Anthill (for nightly builds)
- Bugzilla
- Web browsers
- Service browsers: Inca X
- OSs: UNIX, Linux, Mac OS, Windows
- Developer's Corner
<http://sorcer.cs.ttu.edu/resources/dev-corner.html>



- CVS SW Repository
 - CVS physically on yieldddb
 - mounted as /research/sorcer/sorcer.cvsrep
 - CVS for cs4311 in /sp/cvs/sorcer.cvsrep
- File Storage
 - NFS File System



1. use ***tcs*** as your default shell
 2. Configure
 1. `cp -r /projects/sorcer/configs ~`
 2. `cp ~/configs/.cshrc ~`
 3. `cp ~/configs/.login ~`
 4. `mkdir IGRID_HOME` for example
`/projects/users/<login-name>/projects/iGrid`
 5. set IGRID_HOME in `~/configs/.iGrid_config` accordingly
 3. Get iGrid sources
 1. `cd <IGRID_HOME>`
 2. `cp -r ~/configs .`
 3. `source ~/.cshrc`
 4. `cd ..`
 5. `cvs co iGrid`
 4. Get an individual package
 1. `cd $IGRID_HOME`
 2. `cd ..`
 3. `cvs co iGrid/modules/examples/src/sorcer/provider/arithmetic`
- In `.cshrc`:
`source ~/.iGrid_config` – aliases and CVS repository
`source $IGRID_HOME/configs/.iGrid_env` – sandbox configuration
 - In Eclipse workspace, configuration files are in `iGrid/configs`
<http://sorcer.cs.ttu.edu/resources/sorcer/sorcer-eclipse.pdf>



- Wintel desktops (in 20B) – development
- bamboo and yieldddb – login servers
- yucca and yield – remote desktop servers
- Sun Ray 170 – VLAN931 Red Hat
- Sun Ray 170 – VLAN932 Solaris
- pipal – databases and documents
- Servers in 17B, access per justified request
(contact mike.sobolewski@cs.ttu.edu)
- See the list of developments hosts at:
<http://sorcer.cs.ttu.edu/resources/dev-hosts.html>



- /research
 - feasibility studies, course projects
- /projects
 - Masters, Ph.D. projects
- /integra
 - integrated projects with the SORCER environment
- /archive
 - archived SORCER SW and documents



- /research
 - studies – poppy:/Volumes/Projects
 - sorcer – neem:/export/research/sorcer (research - R:)
 - fiper – neem:/export/research/fiper
 - jgapp – redwood:/research/disk2/jgapp (gapp - G:)
- /projects
 - sorcer – redwood:/home/staff1/sorcer (with portal) (sorcer - S:)
 - dm - redwood:/home/faculty2/search
 - users – neem:/export/projects/users (sandbox - W:)
 - ce2004s - neem:/export/projects/users/ce2004
 - ce2004 – redwood:/home/faculty2/ce2004 (with confdm)
 - ce2005 – redwood:/home/courses1/ce2005
- /integra
 - sorcer – yew:/local/home/sorcer (with portal)
 - ce2004 - yew:/local/home/ce2004
 - ce2005 - yew:/local/home/ce2005
 - demos - yew:/export/projects
- /archive - yew:/export/archive
- /backup
 - projects – poppy:/Volumes/Backup (cron -> sorcer)
 - cvs – pipal:/export/backup (cron -> sorcer)
 - database – pipal:/home/oracle/backups (cron -> oracle)



CS File System – development

/research/disk2/jgapp

/home/staff1/sorcer

/home/faculty2/ce2004

/home/courses1/ce2005

/home/undergrad3/gisoa

/home/undergrad1/gisob

/home/faculty2/search

/home/courses1/java, cs3392, cs5376, cs5331

/research/disk2/compuse

Websites (spring)

/local/home/java, cs4392, cs4311, cs5376, cs5331



Basic Volumes in SORCER

neem	/export/research – 34G /export/projects – 34G /usr/integra in /usr (34G) /usr/archive
yew	/local/home - production only, not exported /export/projects – 33G (for new projects) /export/archive – 33G
poppy	/Volumes/Backup – 57G /Volumes/Projects – 20G
pipal	/ora-data – 46G /export/backup – 7.7G /home/oracle/backups (cron)
teak	/export/share



- Unix/Linux workspace
/projects/users/<username>
mount as \\yieldddb\users
- Wintel workspace, yucca and yield -
USERS (D:)
D:\<username>
- Shared research related code
/reserach/sorcer
mount as \\yieldddb\research



iGrid (SORCER_HOME)

- bin
- configs
- policy
- modules
 - jgapp
 - sorcer
 - sorcer dm
 - sorcerportal
 - providers
 - wservices
 - mobility
 - examples
 - buildtools
 - utilities
- classes
- common (off the shelf libraries)
- lib (iGrid libraries)
- docs
- logs



- iGrid (SORCER_HOME)
 - bin
 - configs
 - policy
 - modules
 - jgapp
 - sorcer
 - providers
 - mobility
 - cdc
 - » pp
 - cldc
 - » midp
 - examples
 - arithmetic
 - raytrace
 - classes
 - common (off the shelf libraries)
 - lib (iGrid libraries)
 - docs



iGrid (SORCER_HOME)

iGrid-build.xml – built in terms of other modules

- bin
- configs
- logs
- policy
- modules
- sorcer

sorcer-build.xml – build a module

- src
 - sorcer
 - core
 - provider
 - jobber

bin

jobber-run-build.xml

configs

logs

policy

jobber-build.xml – build submodule (package)

common-build.xml

common-run.xml

- classes
- common
- lib
- docs



<https://starterkit.dev.java.net/downloads/index.html>

sorcer.DEV
persimmon, willow
/space/iGrid/jini2_1
yew
/usr/jini2_1

If you want to run own Jin, then change the Jini default service group to **socer.DEV** by replacing the folowing line:

```
initialLookupGroups = new String[] { System.getProperty("user.name") + "InstallVerifyGroup" };  
by  
initialLookupGroups = new String[] { "sorcer.DEV" };
```

in all *config files jini2_1/installverify/support then execute jini2_1/installverify/Launch-All

Jini 2.1 install available on: persimmon, willow, yucca, hemp, ivy, spring, yew, poppy



- Class server (webster) environment variables
 - IGRID_WEBSTER
 - IGRID_WEBSTER_PORT
- Scripts and webster-run.xml in iGrid/bin/webster
- Provide all lib, common *.jar files and iGrid/classes
- Use most recent SW versions (included in iGrid sandbox) from:
 - /research/sorcer/jini
 - /research/sorcer/rio
 - /research/sorcer/ant
 - /research/sorcer/tomcat
 - /research/sorcer/incax
 - /research/sorcer/java



- In iGrid/modules/<module-name>/<package-name>/ and/or <package-name>:
 - src
 - bin
 - *secure-persitent-activatable-<protocol>-<provider name>-prv.sh*
 - *secure-persitent-activatable-<protocol>-<provider name>-prv.bat*
 - *secure-persitent-activatable-<protocol>-<provider name>-req.sh*
 - *secure-persitent-activatable-<protocol>-<provider name>-req.bat*
 - *secure-persitent-activatable-<protocol>-<provider name>-test.sh*
 - *secure-persitent-activatable-<protocol>-<provider name>-test.bat*
 - lib
 - configs
 - policy
 - logs

- <protocol> - *jrmp, jeri, jsse, http, https*
- If not secure, not persistent, not activatable– drop the word



- In iGrid/modules/<module-name>/<package-name>/ and/or <package-name>:
 - src
 - bin
 - *secure-persitent-activatable-<protocol>-<provider name>-prv-run.xml*
 - *secure-persitent-activatable-<protocol>-<provider name>-req-run.xml*
 - *secure-persitent-activatable-<protocol>-<provider name>-test-run.xml*
 - lib
 - configs
 - policy
 - Logs
 - *secure-persitent-activatable-<protocol>-<provider name>-prv-run.xml*
 - *secure-persitent-activatable-<protocol>-<provider name>-req-run.xml*
 - *secure-persitent-activatable-<protocol>-<provider name>-test-run.xml*
- <protocol> - *jrmp, jeri, jsse, http, https*
- If not secure, not persistent, not activatable – drop the word



- In your module:
 - src
 - bin
 - lib
 - configs
 - secure-persitent-activatable-<protocol>-<provider name>-priv.config
 - secure-persitent-activatable-<protocol>-<provider name>-req.config
 - secure-persitent-activatable-<protocol>-<provider name>-test.config
 - start-secure-persitent-activatable-<protocol>-<provider name>-priv.config
 - start-secure-persitent-activatable-<protocol>-<provider name>-req.config
 - start-secure-persitent-activatable-<protocol>-<provider name>-test.config
 - <provider name>-priv.properties
 - <provider name>-req.properties
 - secure-persitent-activatable-<protocol>-<provider name>-priv.logging
 - secure-persitent-activatable-<protocol>-<provider name>-req.logging
 - sorcer.logging
 - policy
 - logs
 - <protocol> - *jrmpp, jeri, jsse, http, https*
 - If not secure, not persistent, not activatable – drop the word



- In your module:
 - src
 - bin
 - lib
 - configs
 - policy
 - secure-*<provider name>-prv.policy*
 - secure-*<provider name>-req.policy*
 - secure-*<provider name>-test.policy*
 - logs

- *<protocol>* - *jrmp, jeri, jsse, http, https*
- If not secure – drop the word



- Use `iGrid/configs/sorcer.logging` logging config file
 - D`java.util.logging.config.file=${IGRID_HOME}/configs/sorcer.logging`
- It creates five loggers in `sorcer.core.util.Log` as follows:
`Log.getStarterLog()`, `Log.getSorcerLog()`, `Log.getProviderLog()`,
`Log.getRandomLog()`, `Log.getTestLog()`
- Rotating log files: `sorcer1.log`, `sorcer2.log`, `sorcer3.log`, each 1MB
- Use `java.util.logging.Logger` API with any of five available loggers
- See comments in `sorcer.core.util.Log` for details



- Printers
 - cp309pr or cs3fl - CP 309 (next door to my room)
 - cp20pr is room 20B, SORCER lab printer
 - cp306pr or cs3fl1 - CP 306
- On *yew, neem, poppy*
 - `print myJavaFile.java`



yucca

- /research/sorcer (research – R:)
 - iGrid.cvsrep
 - ant
 - tomact
 - jini
- /projects/sorcer (sorcer – S:)
 - projects/iGrid
 - users (projects – P:)
- /integra
 - sorcer
 - demos

SORCER Web server

- Bookmark URL: Developer's Corner
 - <http://sorcer.cs.ttu.edu/resources/dev-corner.html>



- Apache (SORCER website, CE2004 and CE2005)
 - To start as root: /usr/local/apache2.0.43/bin/apchectl start
- Tomcat-CE2004 -production
 - To start as ce2004 (local):
source ~ce2004/public_html/cedm/conf/.jconfigure
~ce2004/public_html/cedm/bin/tomcat start/restart
- CE2004 dev: ~ce2004/dev/cedm/conf
- CE2004 staging: ~ce2004/public_html/cedm/conf
- Tomcat-CE2005
 - To start as ce2005 (local):
source ~ce2005/public_html/cedm/conf/.jconfigure
~ce2005/public_html/cedm/bin/tomcat start/restart



- /Users/admin/startjini
- uses /research/sorcer/etc/env/poppy
- At startup
 - /Library/StartUpItems/JiniServices
 - /Library/StartUpItems/NFSMounts



- Install JDK 5 and Apache Ant
<http://java.sun.com/javase/downloads/index.jsp> and <http://ant.apache.org/>
- Install Jini 2.1 (runtime Jini services are not required) see page 16
- Install Inca X Service Browser <http://www.incax.com/download.htm>
- Create an iGrid sandbox – see page 5 (**Eclipse CVS-based workspace recommended – at minimum Java 5 needed only**)
- **README**, *-build.xml, and *run.xml files are included in iGrid modules to help you build and learn iGrid and run its services
- Play with *arithmetic* providers and requestors (*jeri* version recommended first in *iGrid/examples/arithmetic/jeri*) to test and learn how to develop your own SORCER services
- ServiceUI <http://www.artima.com/jini/serviceui/index.html>



Q&A



Michael Sobolewski
sobol@cs.ttu.com