Using Open Source Software in Ground Systems

Walt Scacchi
Institute for Software Research
University of California, Irvine
2012 Ground Systems Architecture Workshop
27 February 2012

Overview

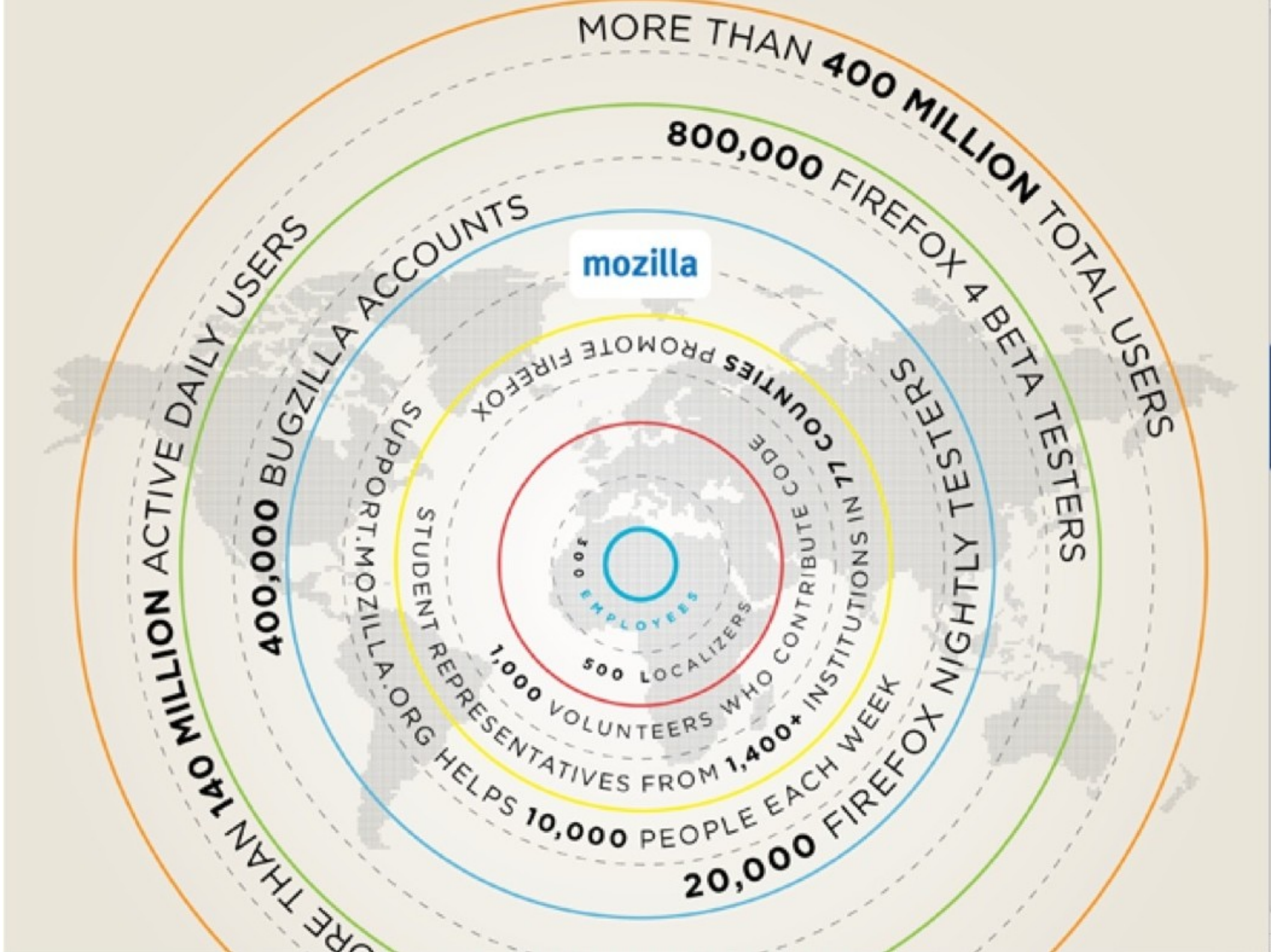
- Background
- Empirical findings from OSSD
- Open architectures and OSS ground systems
- OSSD corporate strategies for ground systems
- Discussion and limitations
- R&D opportunities and issues

Background

- What is Free/Open Source Software Development?
- FOSSD project characteristics
- FOSS development models
- What is an Open Architecture?
- FOSS and DoD

What is free/open source software development?

- Free (as in “freedom” or liberty) vs. open source
  - Freedom to access, browse/view, study, modify and redistribute the source code
  - Free is always open, but open source is not always free
- FOSSD is not “software engineering”
  - *Different*: FOSSD can be faster, better, and cheaper than SE in some circumstances
  - FOSSD teams use 10-500+ OSSD tools (versions) and communications applications to support their development work


MORE THAN 400 MILLION TOTAL USERS
800,000 FIREFOX 4 BETA TESTERS
20,000 FIREFOX NIGHTLY TESTERS
MORE THAN 140 MILLION ACTIVE DAILY USERS
400,000 BUGZILLA ACCOUNTS
mozilla
SUPPORT.MOZILLA.ORG HELPS 10,000 PEOPLE EACH WEEK
STUDENT REPRESENTATIVES FROM 1,400+ INSTITUTIONS IN 77 COUNTRIES
1,000 VOLUNTEERS WHO CONTRIBUTE CODE
500 LOCALIZERS
PROMOTE FIREFOX
300 EMPLOYEES

Categories

Mobile	1,093
Internet	33,111
Text Editors	4,368
Religion and Philosophy	557
Scientific/Engineering	21,273
Social sciences	665
Other/Nonlisted Topic	5,413
Formats and Protocols	5,606
Database	9,564
Security	5,476
Printing	793
Terminals	944
Office/Business	14,205
System	26,242
Education	8,718
Games/Entertainment	24,590
Desktop Environment	5,258
Software Development	37,643
Communications	20,779
Multimedia	19,483

Platform

Windows
Mac
Linux
Symbian

Dev Status

Inactive	6,067
Mature	3,478
Production/Stable	38,311
Beta	44,465
Alpha	32,597
Pre-Alpha	30,994
Planning	41,790

Showing 25 of **304,442** results

Sort by: Downloads

Ares Galaxy Updated 2010-10-27

Filesharing-Bittorrent p2p client connected to TCP supernode/leaf network and UDP DHT network. Ares features a built-in directshow media player, a powerful library manager, shoutcast radio support and can be used to host p2p Chatrooms.

Download AresRegular217_102710.

116936 recommendations

10,495,575 this week

7-Zip Updated 2011-04-18

7-Zip is a file archiver with the high compression ratio. The program supports 7z, XZ, BZIP2, GZIP, TAR, ZIP, WIM, ARJ, CAB, CHM, CPIO, CramFS, DEB, DMG, FAT, HFS, ISO, LZH, LZMA, MBR, MSI, NSIS, NTFS, RAR, RPM, SquashFS, UDF, VHD, WIM, XAR, Z.

Download 7z922.tar.bz2

24726 recommendations

1,606,945 this week

eMule Updated 2010-04-27

eMule is a filesharing client which is based on the eDonkey2000 network but offers more features than the standard client

Download vlc-1.0.5_partfile_plugin_v

46424 recommendations

1,486,206 this week

Smart package of Microsoft's core fonts Updated 2006-05-07

So far this project consists of a source rpm that can be used to easily create a binary rpm package that, when installed, gives access to Microsoft's TrueType core fonts for the Web.

73 recommendations

1,150,581 this week

PortableApps.com: Portable Software/USB Updated 2011-08-12

PortableApps.com allows you to carry your favorite computer programs and all of your bookmarks, settings, email and more with you on a portable device (USB flash drive, iPod, portable hard drive, CD, etc) and use them on any Windows computer.

Download PortableApps.com_Platfon

11177 recommendations

1,020,379 this week

MinGW - Minimalist GNU for Windows Updated 2011-08-05

MinGW: A native Windows port of the GNU Compiler Collection (GCC), with freely distributable import libraries and header files for building native Windows applications; includes extensions to the MSVC runtime to support C99 functionality.

Download x86-mingw32-build-1.0-sh

679 recommendations

916,767 this week

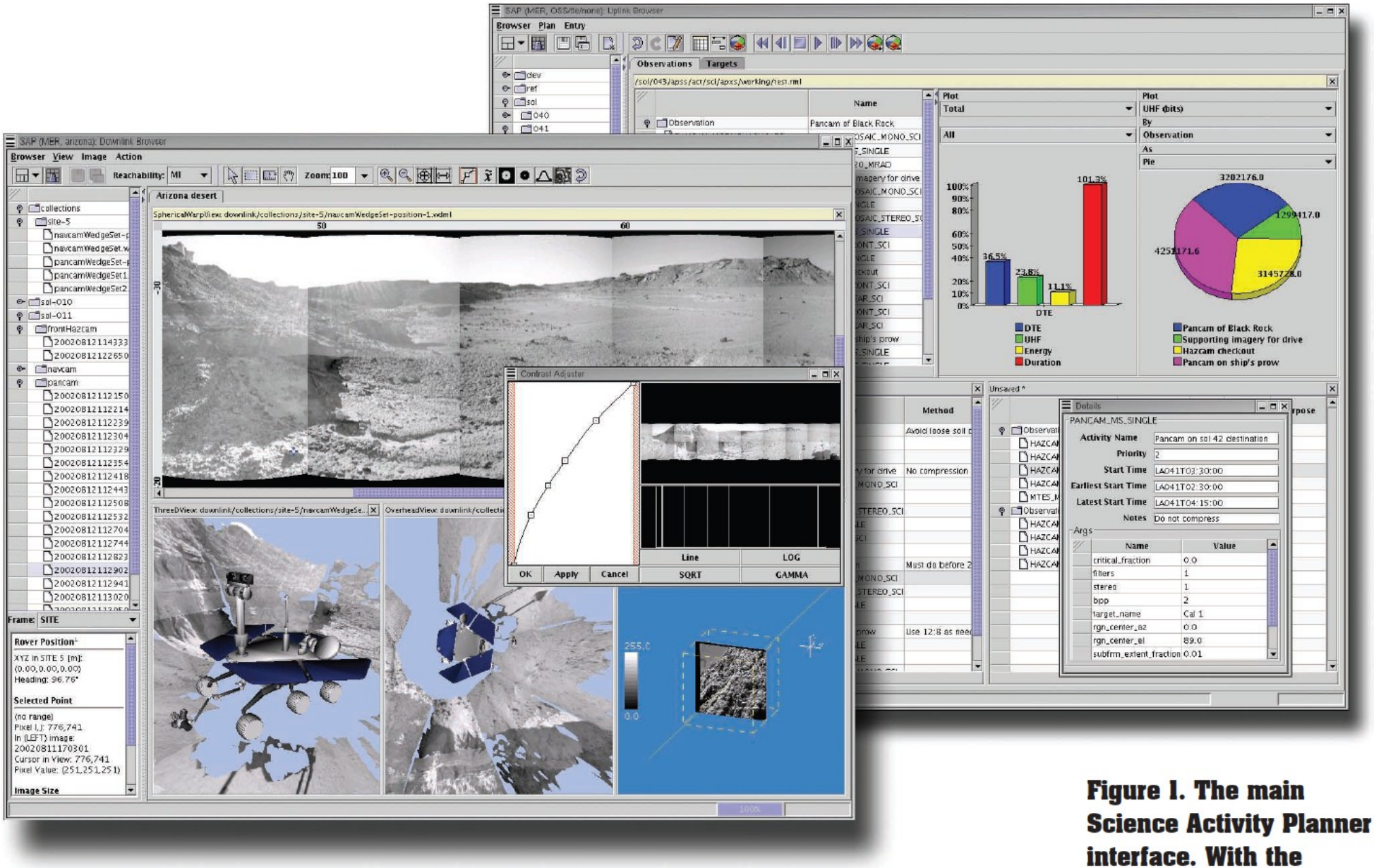


Figure 1. The main Science Activity Planner interface. With the downlink browser

Source: J.S. Norris, Mission Critical Development of Open Source Software: Lessons Learned, *IEEE Software*, 21(1), 42-49, Jan 2004.

FOSSD Project Characteristics

- FOSS developers are typically users of what they build, while FOSS users (~1%) are also FOSS developers
- Requires "*critical mass*" of contributors and FOSS components connected through socio-technical interaction networks
- FOSSD projects can emerge/evolve via *bricolage*
  - Unanticipated architectural (de)compositions
  - Multi-project component integrations

FOSS Development Models

- Free Software (GPL)
- Permissive Open Source (BSD/MIT, FreeBSD)
- Corporate/Inner Source (Hewlett-Packard)
- Consortium/Alliance (OSDL, SugarCRM)
- Non-profit foundations (Apache, Mozilla, Gnome, Perl)
- Corporate-Sponsored (Google, HP, IBM, Microsoft, Nokia, Oracle)
- Modding: OSS extensions to Closed Source (many game companies)
- (Membership) Community Source (Sakai, Westwood)

---------- not OSSD models below ----------

- Shared Source with Non-Disclosure (Microsoft)
- Open Systems (open APIs, closed source components)

What is an Open Architecture?

- DoD has announced policies and initiatives that commit to the acquisition of software-intensive systems that require or utilize an *Open Architecture* and *Open Technology*.*
- OA systems may include components with open APIs, OSS technology or development processes.
- Air Force, Army, and Navy each have their own reasons for adoption OA systems.
  - But what happens when there are conflicts across the services regarding what an OA is?
- Therefore, is it clear what an OA is?

* *Open Technology Deployment (OTD): Lessons Learned and Best Practices for Military Software* DoD CIO, November 2011.

FOSS and DoD

Home | About | Site Map | Communities of Interest | C2 Research Program | Links | Search

View From The Front Office

Teri M Takai

Vision / Mission

U.S. Code

DoDD 5144.01

"Information is our greatest strategic asset."
Teri M Takai, DoD CIO

Media

Net Centricity - Overview
Net Centricity - Summary
Blue Force Tracking

Free Open Source Software (FOSS)

- Clarifying Guidance Regarding Open Source Software (OSS)
  Memorandum by David M. Wennergren, 16 October 2009
- Open Technology Development (OTD): Lessons Learned & Best Practices for Military Software
  OSD Report, May 2011
- DoD Open Source Software Frequently Asked Questions
- Use of Free and Open-Source Software (FOSS) in the U.S. Department of Defense
  2003 Study by MITRE Corporation performed for DoD

DoD FOSS Guidance

- Policy (2009)

  http://dodcio.defense.gov/sites/oss/2009OSS.pdf

- Lessons Learned (2011)
  http://dodcio.defense.gov/sites/oss/OTD-lessons-learned-military-signed.pdf

- Frequently Asked Questions and Answers (2010)
  http://dodcio.defense.gov/sites/oss/Open_Source_Software_(OSS)_FAQ.htm

- Foundations (2003)

  http://dodcio.defense.gov/sites/oss/2003Survey/dodfoss_pdf.pdf

FOSS Policy at DoD:

Wennergren 2009 – http://dodcio.defense.gov/sites/oss/2009OSS.pdf

- "OSS is software for which the source code is 'open.'"
- "There have been misconceptions and misinterpretations of the existing laws, policies and regulations that deal with software and apply to OSS, that have hampered effective DoD use and development of OSS."
- "In almost all cases, OSS meets the definition of 'commercial computer software'"
  - OSS *is* COTS!

FOSS Benefits (Wennergren 2009)

- Continuous and broad peer-review enabled by publicly available source code supports software reliability and security efforts
- Ability to modify software source code enables the DoD to respond more rapidly to changing situations, missions, and threats
- Reliance on a particular software developer or vendor due to proprietary restrictions may be reduced by the use of OSS
- OSS provides a net-centric licensing model that enables rapid provisioning of both known and unanticipated users

FOSS Benefits (Wennergren 2009)

- OSS can provide a cost advantage and can mitigate risk of cost growth by avoiding unit licensing costs
- By sharing the responsibility for maintenance of OSS with other users, the DoD can benefit by reducing the total cost of ownership for software
- OSS is particularly suitable for rapid prototyping and experimentation
- Ultimately, the software that best meets the needs and mission of the DoD should be used, regardless of whether the software is open source.

OSS components in Ground Systems?

- How to evaluate OSS components?
  - Same as COTS, but with community support
- Risks with OSS components?
  - Same as COTS, but with community costs and benefits
- Liability and ownership of OSS components?
  - Same as COTS, but through the community
- Foreign contributors to OSS components?
  - Same as COTS
- Security?
  - Better transparency than COTS, and better when community provides continuous, fast cycle improvements

Empirical research findings about FOSSD

(*its not about the source code*)

Individual participation in FOSSD projects: motives and consequences

- FOSS developers want to:
  - learn about new tools, techniques, skills, etc.
  - have fun building software
  - exercise their technical skill
  - try out new kinds of systems to develop
  - interconnect multiple FOSSD projects
- FOSS developers frequently:
  - build trust and reputation with one another
  - achieve “geek fame” (for project leaders)
  - spend more time reading online documents and communicating with one another than writing code

FOSSD *informalisms* for decentralized knowledge sharing and collaboration

Email lists	Discussion forum	News postings	Project digests
IM/Internet Relay Chat	Scenarios of usage	How-to guides	Screenshots
FAQs; to-do lists: item lists	Project Wikis	System documentation	External publications
Copyright licenses	Architecture diagrams	Intra-app scripting	Plug-ins
Code from other projects	Project Web site	Multi-project portals	Project source code
Project repositories	Software bug reports	Issue tracking databases	Blogs, videos, photos, etc.

Introduction

This newsletter mainly covers the the #gnuenterprise IRC channel, with occasional coverage of the three main mailing lists (gnue-announce, gnue and gnue-dev) for the GNU Enterprise project.

1. Further trouble-shooting with the wx 2.6 drivers

20 Jun - 21 Jun Archive Link: "[IRC] 20 Jun 2006"
Summary By Peter Sullivan
Topics: Forms, Common
People: Reinhard Müller, James Thompson, Johannes Vetter, Peter Sullivan

Further to Issue #117, Section #2 (**22 May :** Layout in GNUe Forms with wx 2.6 driver) , Reinhard Müller (reinhard) suggested to James Thompson (jamest) **"if you are bored, you can try again the wx26 uidriver"** , as Johannes Vetter (johannesV) had done **"some massive changes and it might be that your issues with fscking up the boxes are solved"** . James said that, although he was busy, **"i really need to get that tested, as the dropdown box issues in 2.4 are preventing some selections from being allowed"** . So he was keen to have a version of GNUe Forms that worked with the user interface driver for wx 2.6 as soon as possible.

Trying Johannes' new code for GNUe Forms with his existing GNUe Forms Definitions, James found problems - **"none of which are due to anything wrong with what you've done - it's all in my forms"** , where he had been relying on 'features' (such as overlapping text boxes) that Johannes had treated as 'bugs' and now fixed. Johannes confirmed that **"overlaping is now being checked ... not only for boxes but for all widgets"** . He added, **"if you click the detail-button you'll see the offending line in your XML-File - this makes debuging"** a GNUe Form Definition (gfd) **"a lot easier"** . James reported that all five of his existing GNUe Form Definitions were not working with the new code - but **"i would still imagine it's something funky I'm doing in the form"** rather than a problem with Johannes' code. He noted that, on the last one, the problem that he had been having with the dropdown menu had been fixed, but the form now **"aborts on query"** .

(ed. [Peter Sullivan] Note that the lack of any guarantees on backward compatability, even with 'features'/'bugs' is one of the reasons why GNUe Forms remains at a version number below 1.0 as of time of writing, as discussed further in Issue #112, Section #4 (**13 Apr :** Forms approaching version 1.0?) .)

Cooperation, coordination, and control in FOSSD projects: *project governance*

A *meritocratic* role hierarchy and role migration paths for FOSSD

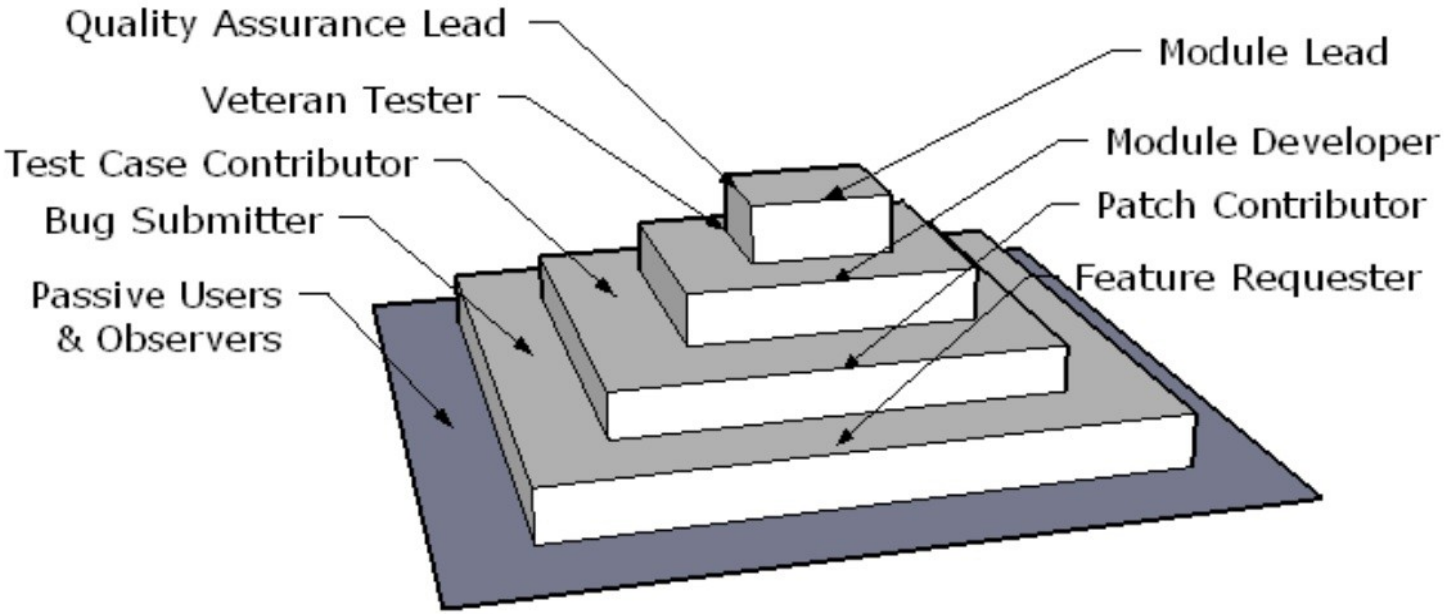


Figure 2. An "onion" pyramid representation of a generic OSSD project organizational hierarchy with multiple role-sets and advancement tracks.

Source: C. Jensen and W. Scacchi, Role Migration and Advancement Processes in OSSD Projects: A Comparative Case Study, in *Proc. 29th. Intern. Conf. Software Engineering,* Minneapolis, MN, May 2007, 364-374.

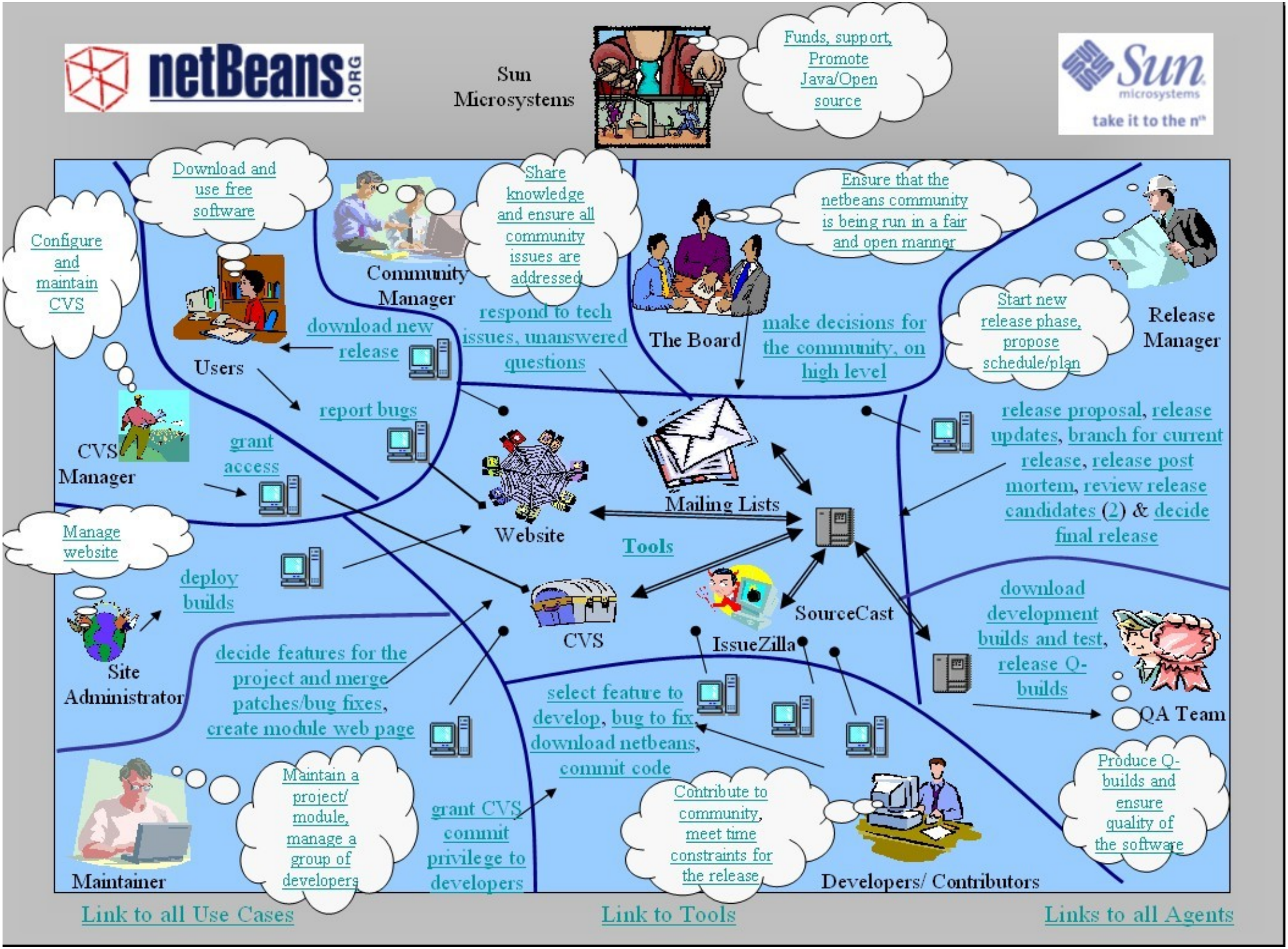
Software version control

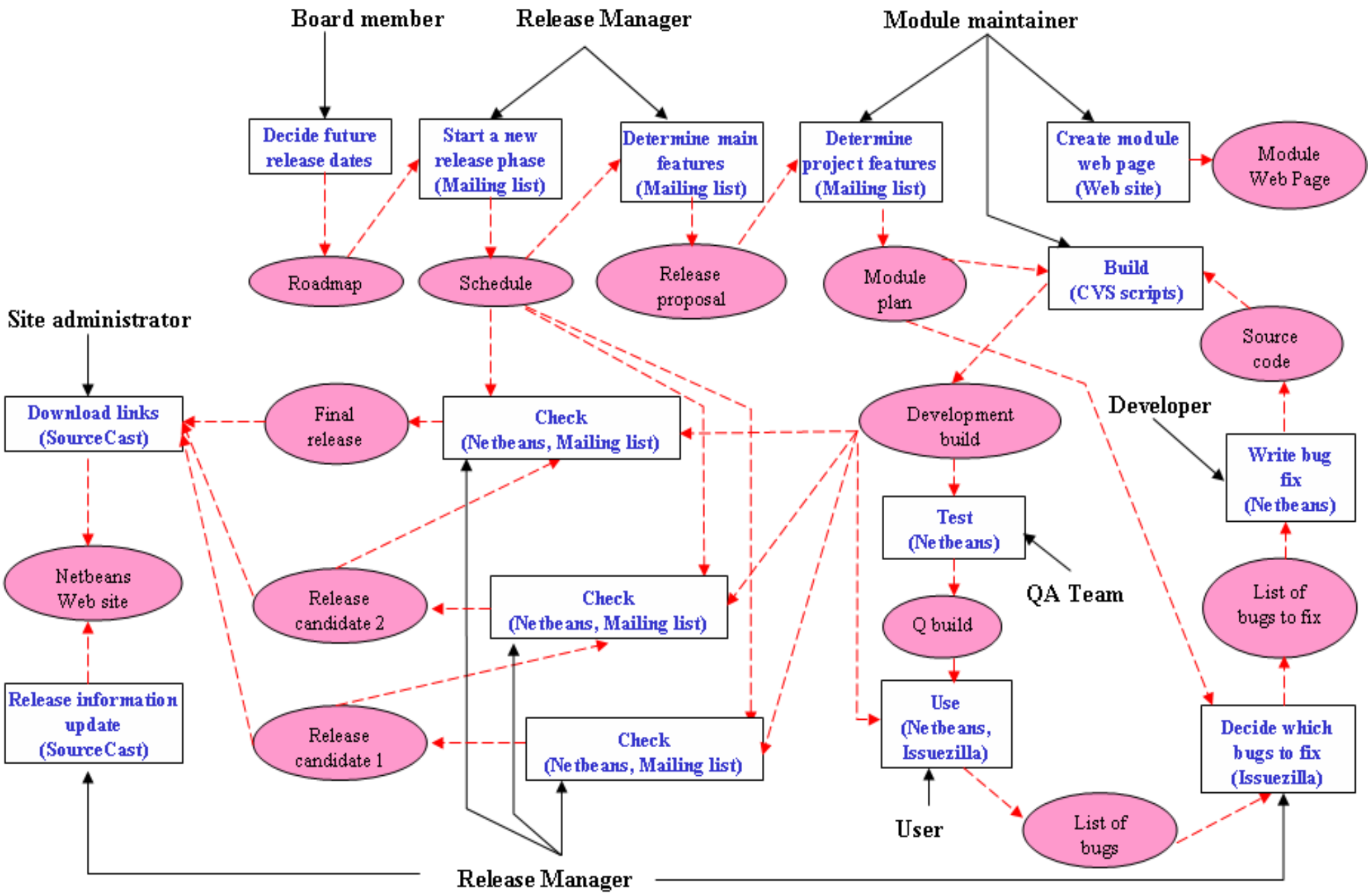
- Enables stabilization and synchronization of dispersed, “invisible” FOSSD work
- SVC tools (CVS, SVN, Git, etc.) used as:
  - Central mechanism coordinating development
  - Online venue for mediators (e.g., “code sheriffs”, “buildmeister”) to control what changes will be accommodated
    - Mediators “throttle” version update and release rates, to allow for “long term support,” “candidate,” and “daily patch” (continuous integration) versions
  - Gentle but sufficient social control mechanism that constrains overall project complexity
  - SVC supports collaboration practices through repositories.

Implicit project management

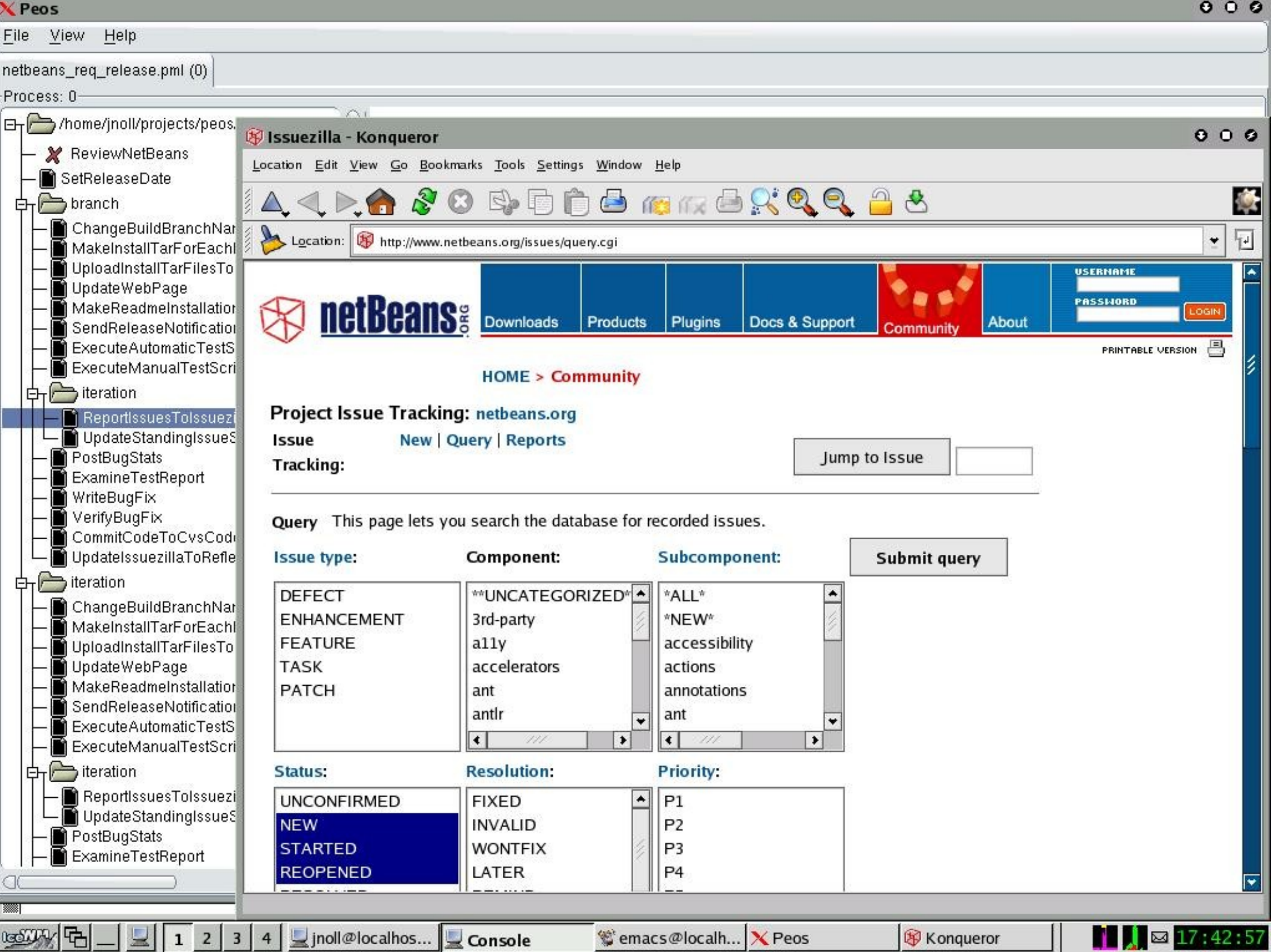
- FOSSD projects self-organize into a *meritocractic role-hierarchy* that enables *virtual project management*
  - Meritocracies embrace incremental innovations over radical innovations
  - VPM requires people to act in leadership roles based on skill, availability, and belief in project community
- Reliance on evolving web of software informalism content constrains collective action within FOSSD project via traceable and searchable information/content legacy

Decentralized OSSD *processes* guided by collaborative tools and informalism coordination repositories

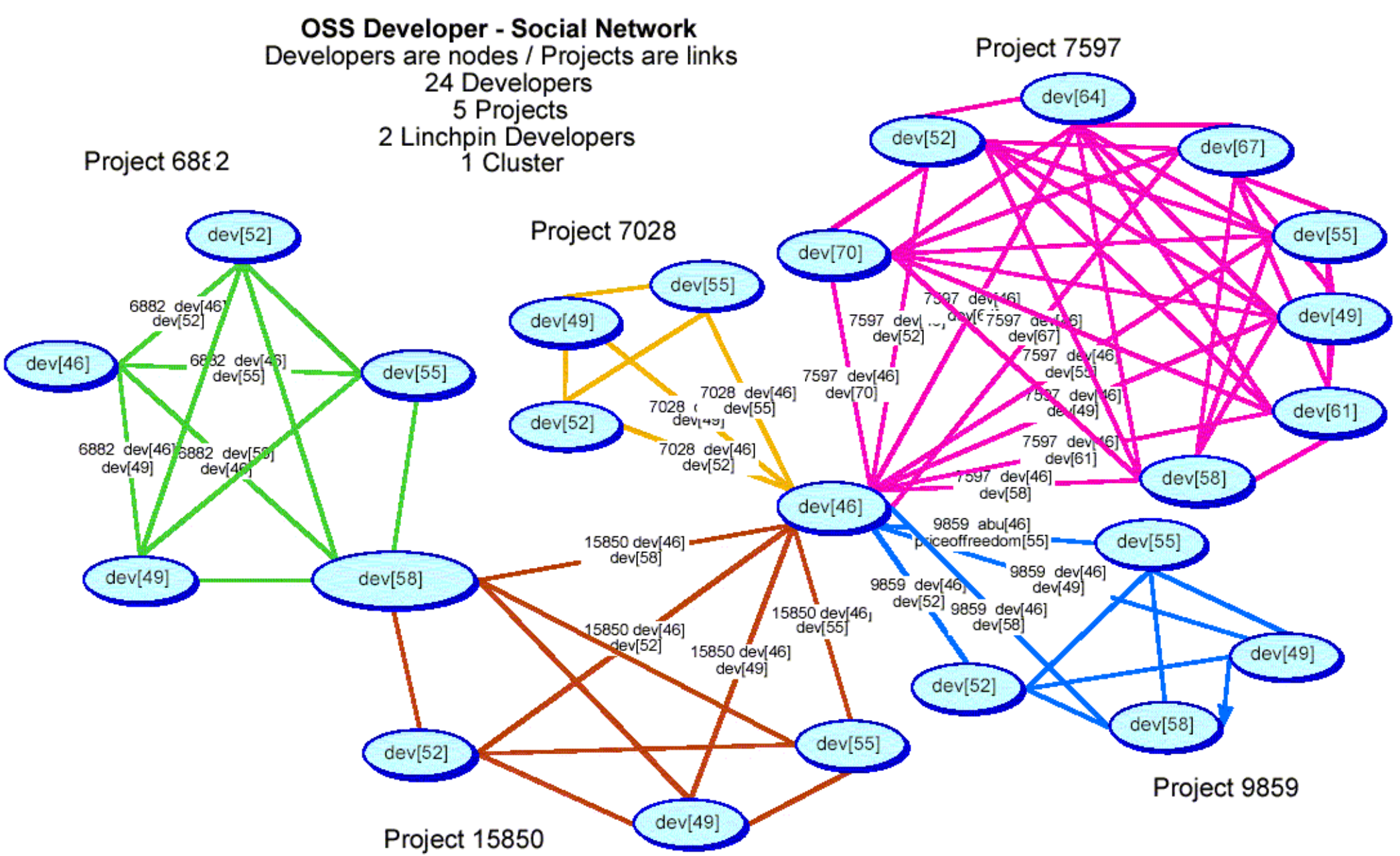

netBeans.ORG
Sun Microsystems
Funds, support, Promote Java/Open source
Sun microsystems
take it to the n
Download and use free software
Configure and maintain CVS
Community Manager
Share knowledge and ensure all community issues are addressed
Ensure that the netbeans community is being run in a fair and open manner
Release Manager
Start new release phase, propose schedule/plan
Users
download new release
respond to tech issues, unanswered questions
The Board
make decisions for the community, on high level
CVS Manager
report bugs
grant access
Website
Mailing Lists
release proposal, release updates, branch for current release, release post mortem, review release candidates (2) & decide final release
Tools
Manage website
deploy builds
Site Administrator
CVS
SourceCast
IssueZilla
download development builds and test, release Q-builds
QA Team
decide features for the project and merge patches/bug fixes, create module web page
select feature to develop, bug to fix, download netbeans, commit code
Maintainer
Maintain a project/ module, manage a group of developers
grant CVS commit privilege to developers
Contribute to community, meet time constraints for the release
Developers/ Contributors
Produce Q-builds and ensure quality of the software
Link to all Use Cases
Link to Tools
Links to all Agents



Legend: Boxes are *activities* (using *informalisms*); Ellipses are *resources* required or provided; Actor *roles* in boldface; *flow dependencies* as arrows.

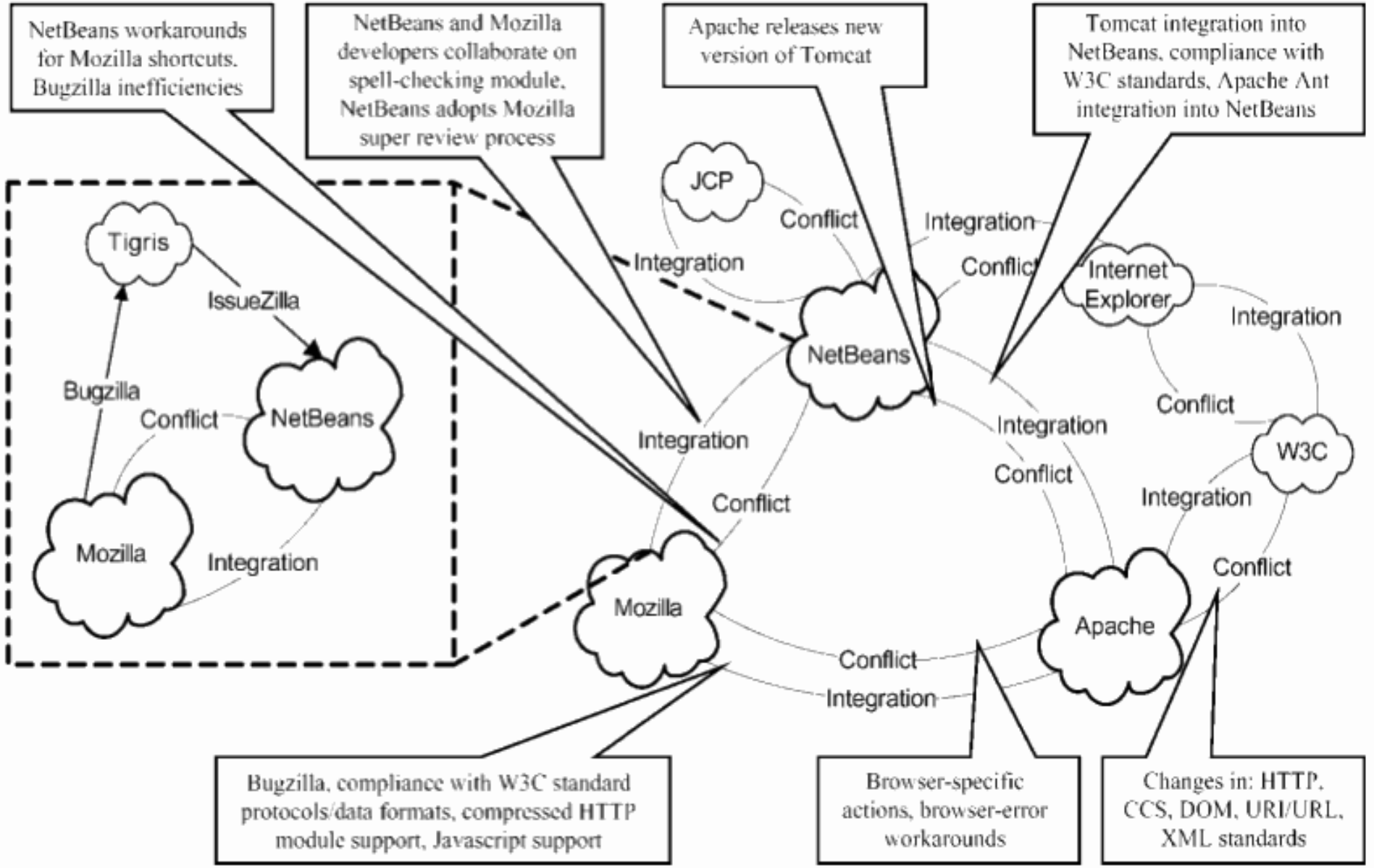

Peos
File View Help
netbeans_req_release.pml (0)
Process: 0
/home/jnoll/projects/peos.
ReviewNetBeans
SetReleaseDate
branch
ChangeBuildBranchNar
MakeInstallTarForEachI
UploadInstallTarFilesTo
UpdateWebPage
MakeReadmeInstallatior
SendReleaseNotificatio
ExecuteAutomaticTestS
ExecuteManualTestScri
iteration
ReportIssuesToIssuezi
UpdateStandingIssueS
PostBugStats
ExamineTestReport
WriteBugFix
VerifyBugFix
CommitCodeToCvsCod
UpdateIssuezillaToRefle
iteration
ChangeBuildBranchNar
MakeInstallTarForEachI
UploadInstallTarFilesTo
UpdateWebPage
MakeReadmeInstallatior
SendReleaseNotificatio
ExecuteAutomaticTestS
ExecuteManualTestScri
iteration
ReportIssuesToIssuezi
UpdateStandingIssueS
PostBugStats
ExamineTestReport
Issuezilla - Konqueror
Location Edit View Go Bookmarks Tools Settings Window Help
Location: http://www.netbeans.org/issues/query.cgi
netBeans.org
Downloads Products Plugins Docs & Support Community About
USERNAME
PASSWORD
LOGIN
PRINTABLE VERSION
HOME > Community
Project Issue Tracking: netbeans.org
Issue Tracking:
New | Query | Reports
Jump to Issue
Query This page lets you search the database for recorded issues.
Issue type:
DEFECT
ENHANCEMENT
FEATURE
TASK
PATCH
Component:
**UNCATEGORIZED*
3rd-party
a11y
accelerators
ant
antlr
Subcomponent:
ALL
NEW
accessibility
actions
annotations
ant
Submit query
Status:
UNCONFIRMED
NEW
STARTED
REOPENED
Resolution:
FIXED
INVALID
WONTFIX
LATER
Priority:
P1
P2
P3
P4
1 2 3 4
jnoll@localhos...
Console
emacs@localh...
Peos
Konqueror
17:42:57

Alliances, social networking, and community development



Source: G. Madey, *et al.*, 2005

FOSS as multi-project software ecosystems



Source: C. Jensen and W. Scacchi, Process Modeling Across the Web Information Infrastructure, *Software Process--Improvement and Practice*, 10(3), 255-272, July-September 2005.

FOSS ecosystems develop around software forges (portals)

FLOSSmole

Collaborative collection and analysis of free/libre/open source project data

About | Getting Data | Using Data | Donating Data | Blog

Navigation

- Recent posts

User login

Username: *

Password: *

Log in

Log in using OpenID

- Request new password

Like 6 likes. Sign Up to see what your friends like.

Search

Search this site:

Search

Getting Data

- Code Forges Study
- Database schema
- Data collection details
- Direct database access
- Download data (at Google Code)

Using Data

Home

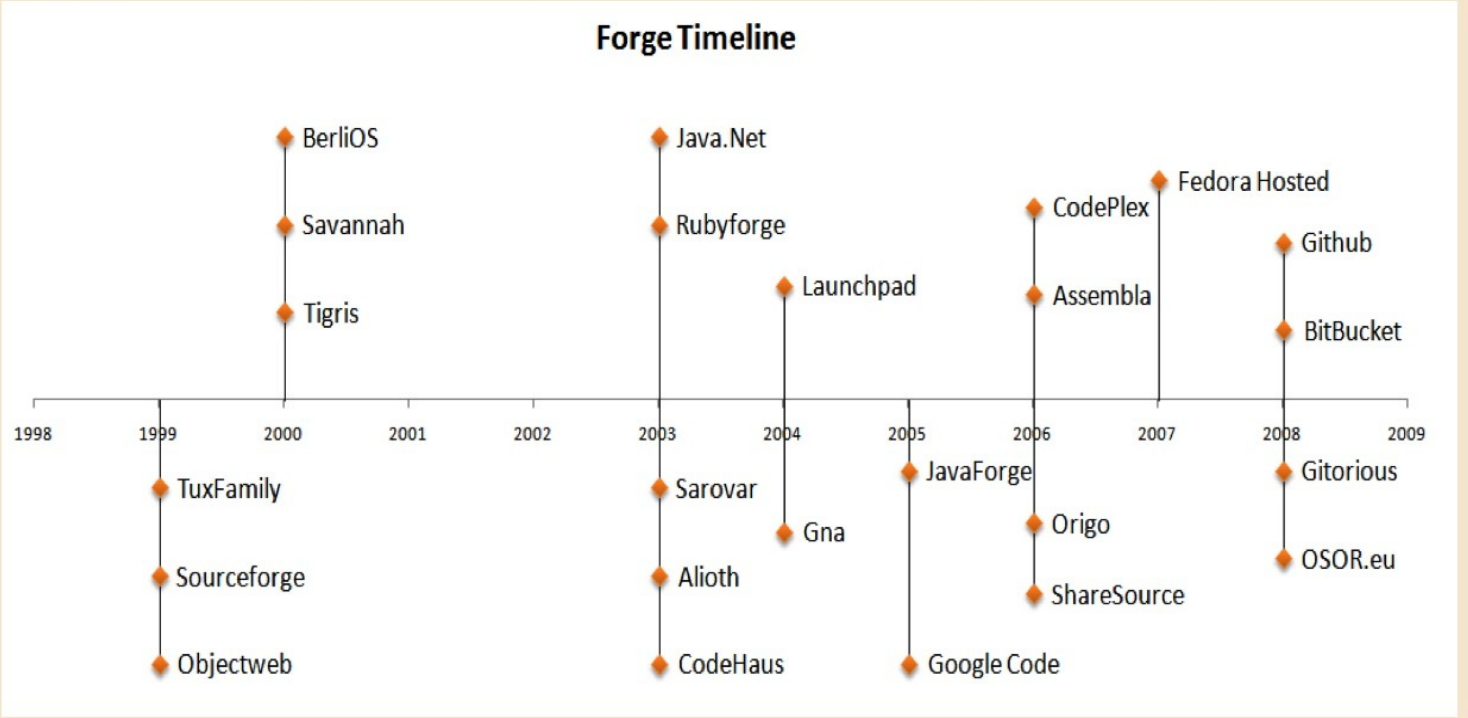
When were the forges established?

Submitted by **dwilliams** on Fri, 06/03/2011 - 11:19 Examples

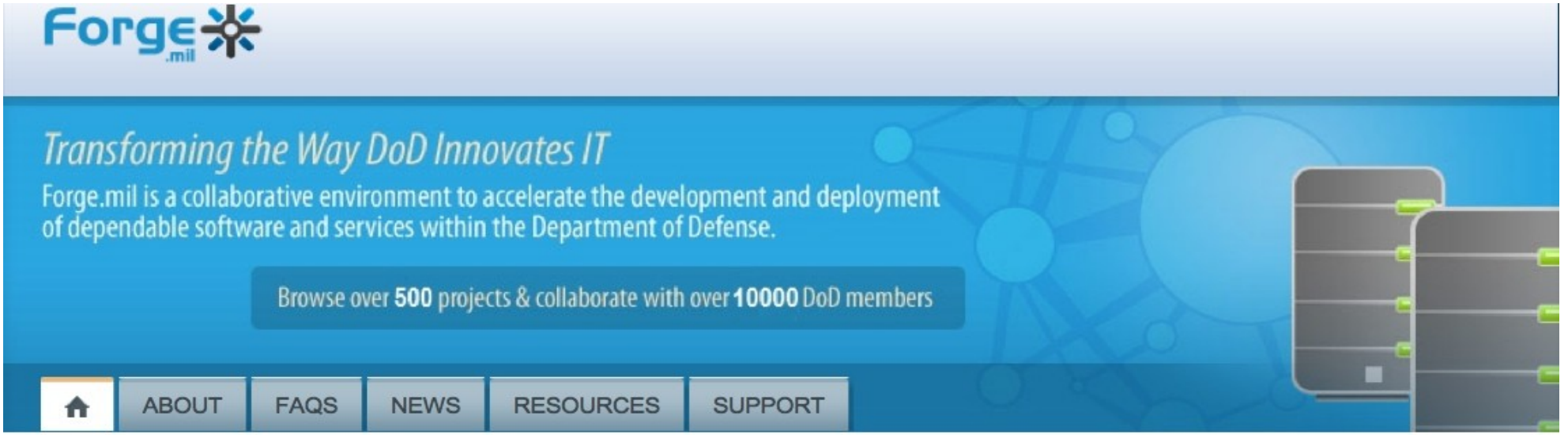
Description

This timeline illustrates when the forges were officially established.

Visualization



DoD Forge.mil



Top 10 FAQs

- What is the Forge.mil Program?
- What is the Forge.mil Community?
- What is SoftwareForge?
- What is ProjectForge?
- What is the difference between SoftwareForge and ProjectForge?
- What capabilities are available in the system?
- What are the guidelines for participating?

The Forge.mil Program

Forge.mil is a family of services provided to support the DoD's technology development community. The system enables the collaborative development and use of open source and DoD community source software. For programs and projects that require greater access control, the system supports private collaborative development with an on demand, fee for service offering. These initial software development capabilities are growing to support the full system life-cycle and enable continuous collaboration among all stakeholders including Project Managers, developers, testers, certifiers, operators, and users.

Tools & Resources

Forge.mil Brochure & ProjectForge Info Sheet

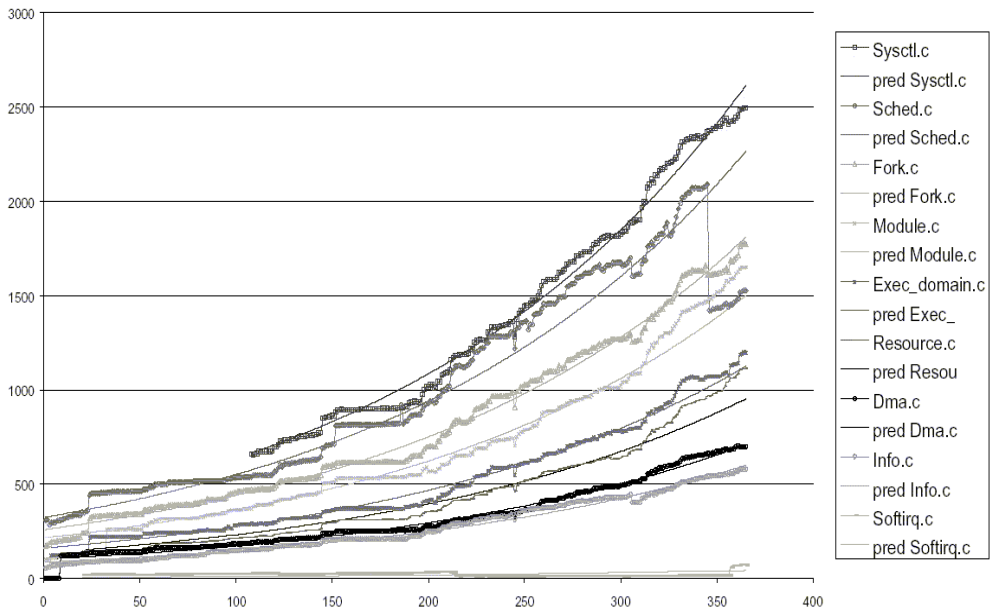
Join Community

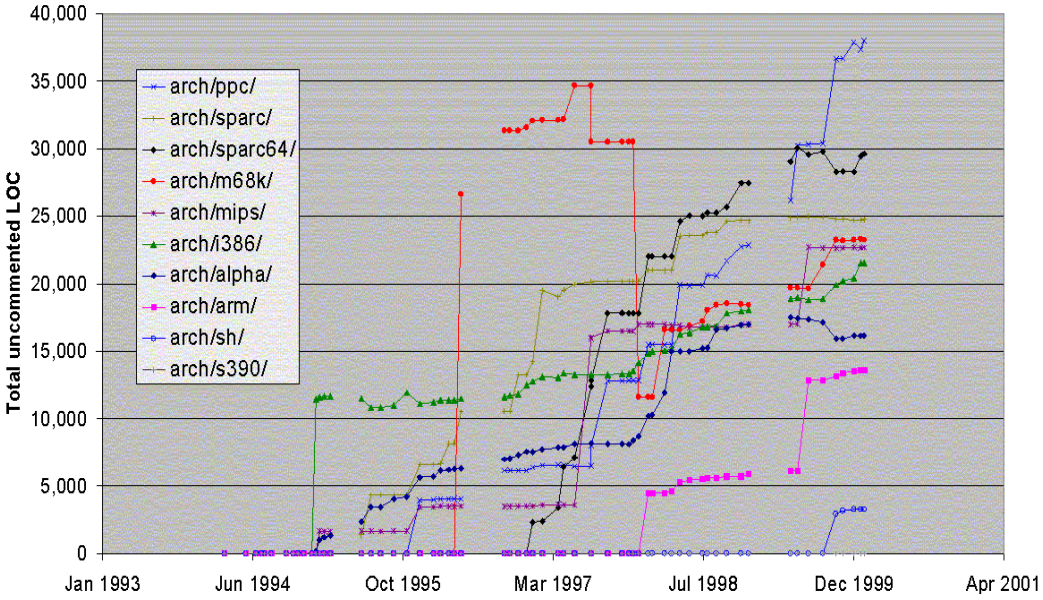
Latest News

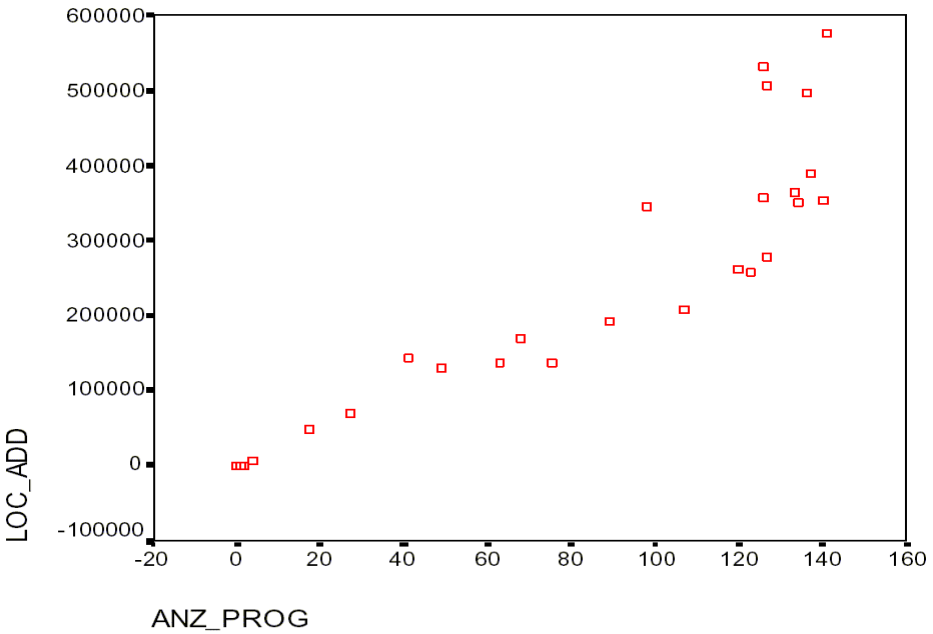
- Forge.mil: A DoD Innovation for the Enterprise
- Social Networking Comes to Forge.mil
- DISA Launches the Forge.mil Community Site
- Hanscom ESC Embraces New

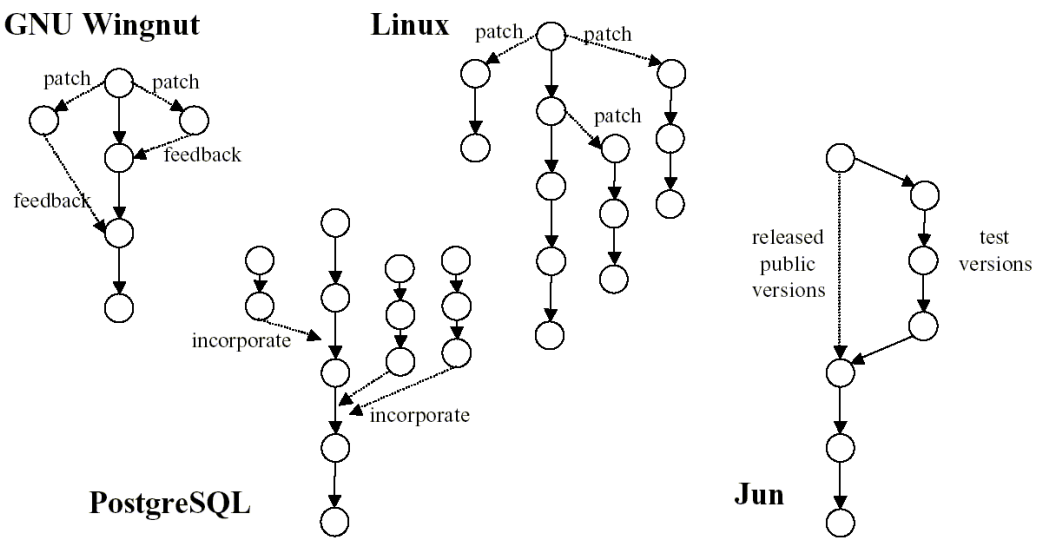
Multi-project software ecosystem

- Mutually dependent FOSS development and evolution propagate functional software cliches/idioms, cloned code, architectural styles, dependencies, and vulnerabilities
- *Architectural bricolage* arises when autonomous FOSSD projects, artifacts, tools, and systems co-mingle or merge
  - Enables discontinuous or exponential growth of FOSS source code, functionality, complexity, contributions


3000
2500
2000
1500
1000
500
0
0
50
100
150
200
250
300
350
400
Sysctl.c
pred Sysctl.c
Sched.c
pred Sched.c
Fork.c
pred Fork.c
Module.c
pred Module.c
Exec_domain.c
pred Exec_
Resource.c
pred Resou
Dma.c
pred Dma.c
Info.c
pred Info.c
Softirq.c
pred Softirq.c


Total uncommented LOC
40,000
35,000
30,000
25,000
20,000
15,000
10,000
5,000
0
Jan 1993
Jun 1994
Oct 1995
Mar 1997
Jul 1998
Dec 1999
Apr 2001
arch/ppc/
arch/sparc/
arch/sparc64/
arch/m68k/
arch/mips/
arch/i386/
arch/alpha/
arch/arm/
arch/sh/
arch/s390/


LOC_ADD
600000
500000
400000
300000
200000
100000
0
-100000
-20
0
20
40
60
80
100
120
140
160
ANZ_PROG


GNU Wingnut
patch
patch
feedback
feedback
Linux
patch
patch
patch
incorporate
incorporate
PostgreSQL
released
public
versions
test
versions
Jun

Evolutionary redevelopment, reinvention, and redistribution

- Overall evolutionary dynamic of many FOSSD projects is *reinvention and redevelopment*
  - Reinvention enables continuous improvement and collective learning
- FOSS evolve through minor mutations
  - Expressed, recombined, redistributed via incremental releases
- FOSS systems *co-evolve* with their development community
  - Success of one depends on the success of the other

Open Architectures and OSS

Open Architectures and OSS

- Air Force sees that with its software-intensive systems, there is increasing complexity of the software (code) itself, they may be "held hostage" to proprietary legacy components, they seek more timely delivery of new solutions, and that acquisitions and requirements take too long [Riechers 2007].
- Army seeks to move away from closed source software, expensive software upgrades, vendor lock-in, and broadly exploited security weaknesses [Justice 2007a,b, Starrett 2007]
- Navy seeks to mitigate the spiraling costs of weapon systems through adoption of OA [Navy 2006], as well as the adoption of open business models for the acquisition and spiral development of new systems [Guertin 2007].

Matching OSS Alternatives to Tactical Warfighter Needs

Classic PEO C3T Acquisition Constraints

Dynamic Wartime Environment

Aggressive Unit Set Fielding Pace

Today's Reality

OSS Spectrum of Alternatives

- Take advantage of custom code & leverage economies of scale of COTS
- Can modify in-house or outsource to vendor

Pure COTS
or unmodified OSS

"Modifiable COTS"
or OSS that relies on short-term modifications, yet attempts to re-merge with newly released OSS updates

Custom Code
or thoroughly modified OSS (owner-maintained)

BUY ⟷ **BUILD**

- Cheaper to acquire
- Need to determine suitability/functionality
- Subject to licensing restrictions
- Subject to maintenance schedule
- Authors maintain control

- More expensive to acquire
- Function according to specification
- Need more labor
- Sometimes difficult to support

Source: MITRE

Can PMs use OSS to deliver viable warfighter capabilities that balance traditional project demands with today's operational reality?

Open Architectures and OSS

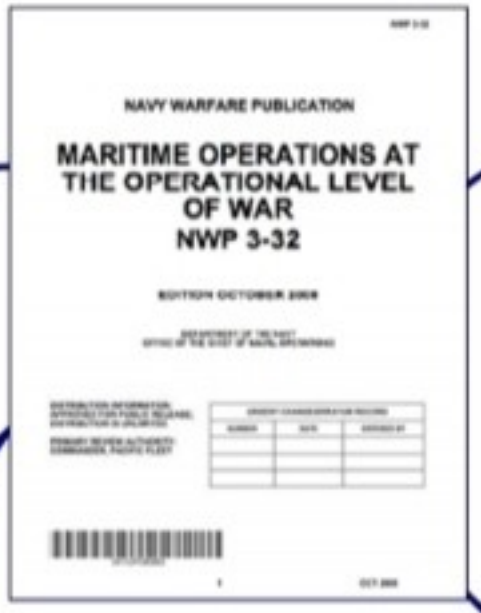
- DoD policies and initiatives encouraging OA with OSS elements
- *Goal*: identify software architecture principles (and OSS licenses) that mediate OA
  - OSS components subject to different IP licenses
- How to determine the requirements needed to realize OA strategies with OSS?

Source: W. Scacchi and T. Alspaugh, Emerging Issues in the Acquisition of Open Source Software within the U.S. Department of Defense, *Proc. 5th Annual Acquisition Research Symposium*, Vol. 1, 230-244, NPS-AM-08-036, Naval Postgraduate School, Monterey, CA, 2008.

C2BMC
MIPS

(i.e. BMD)

Planning



Monitoring (SA)

GCCS (COP)

JFMCC Supporting Plan

To JTF Counterstrike
PLAN ALFA

PowerPoint slides & Word Documents

Directing

J/ADOCS (Fires)

Execution

Outlook .PST files

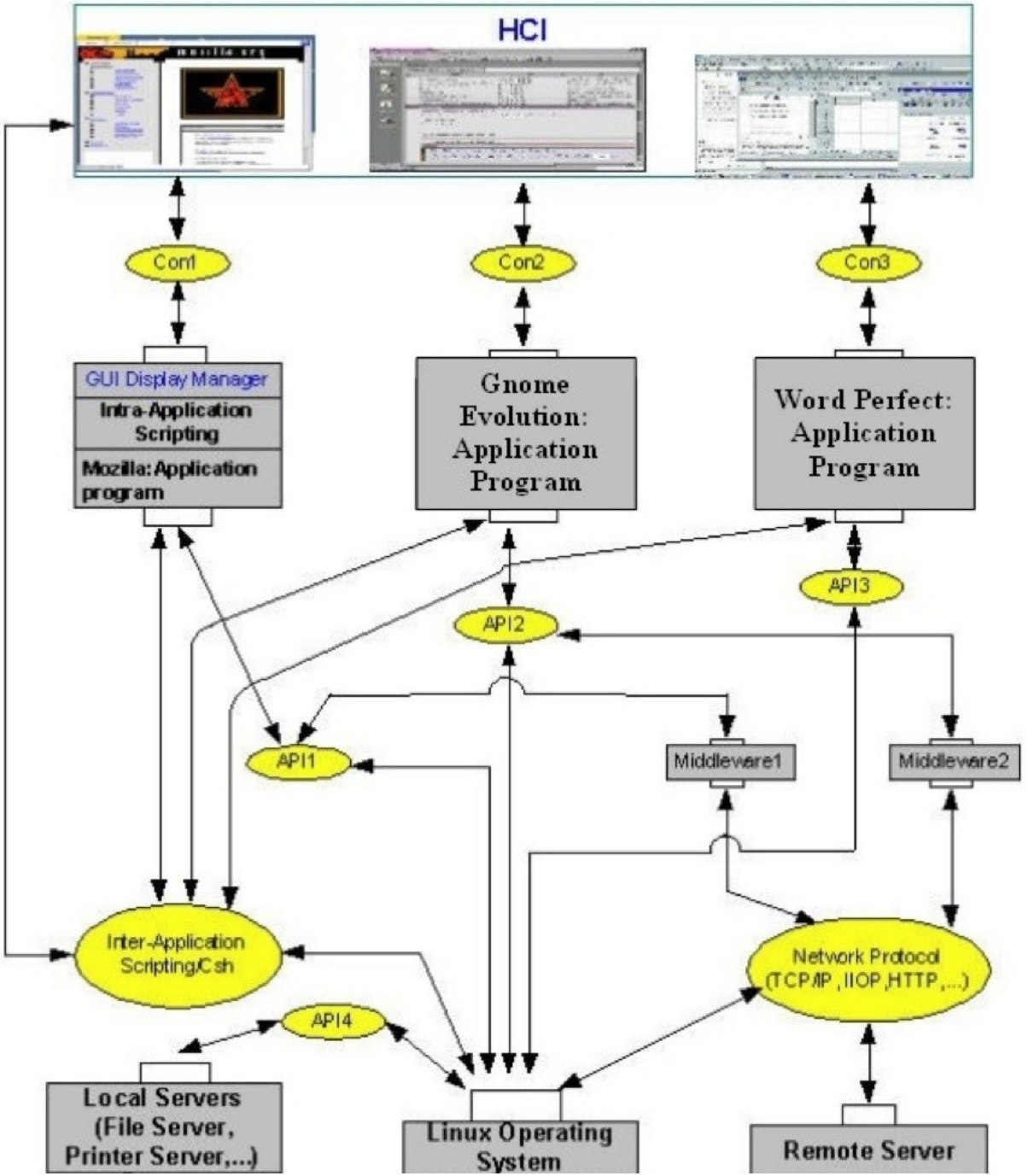
Chat Windows

Excel Spreadsheets

TBMCS (ATO)

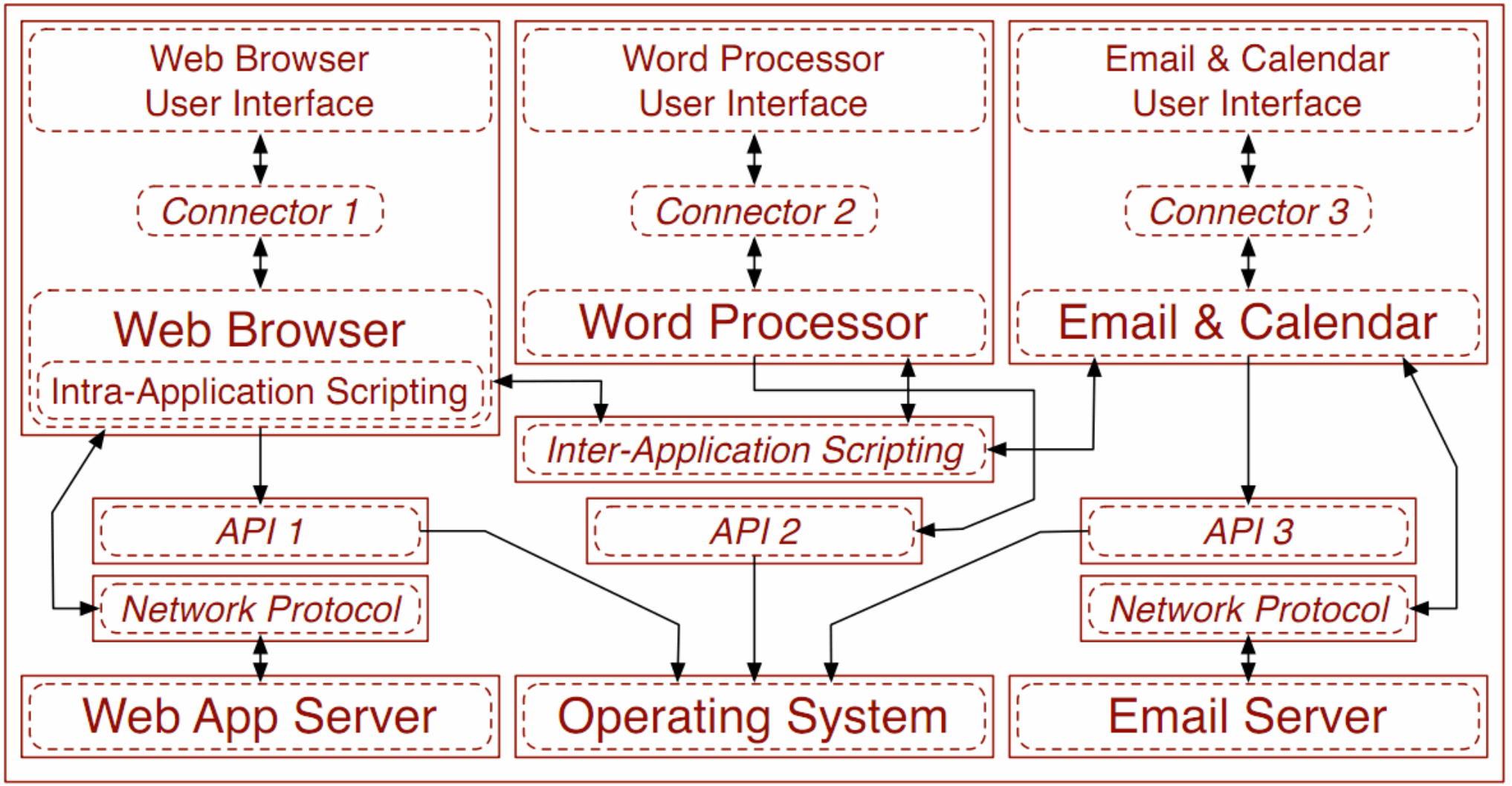
Open Software Architecture Concepts

- Software source code components
  - Standalone programs
  - Libraries, frameworks, or middleware
  - Inter-application script code (e.g., for mash-ups)
  - Intra-application script code (e.g., for Rich Internet Apps.)
- Executable software components (binaries)
- Application program interfaces (APIs)
- Software connectors
- Configured sub-system or system



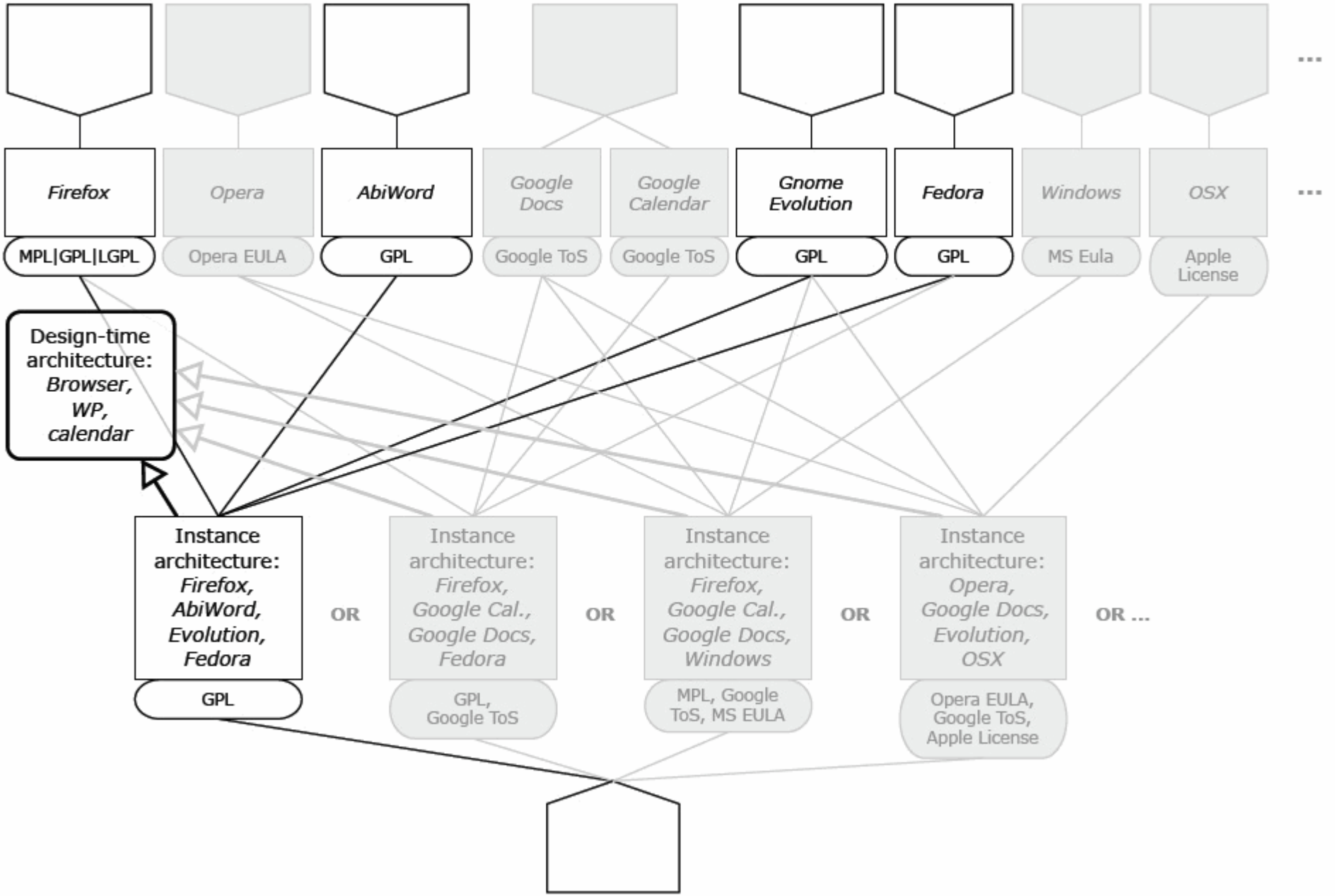
Legend: Grey boxes are *components;* ellipses are *connectors;* white boxes are *interfaces*; arrows are data or control *flow paths*; complete figure is architectural design *configuration*

Design-time OA example

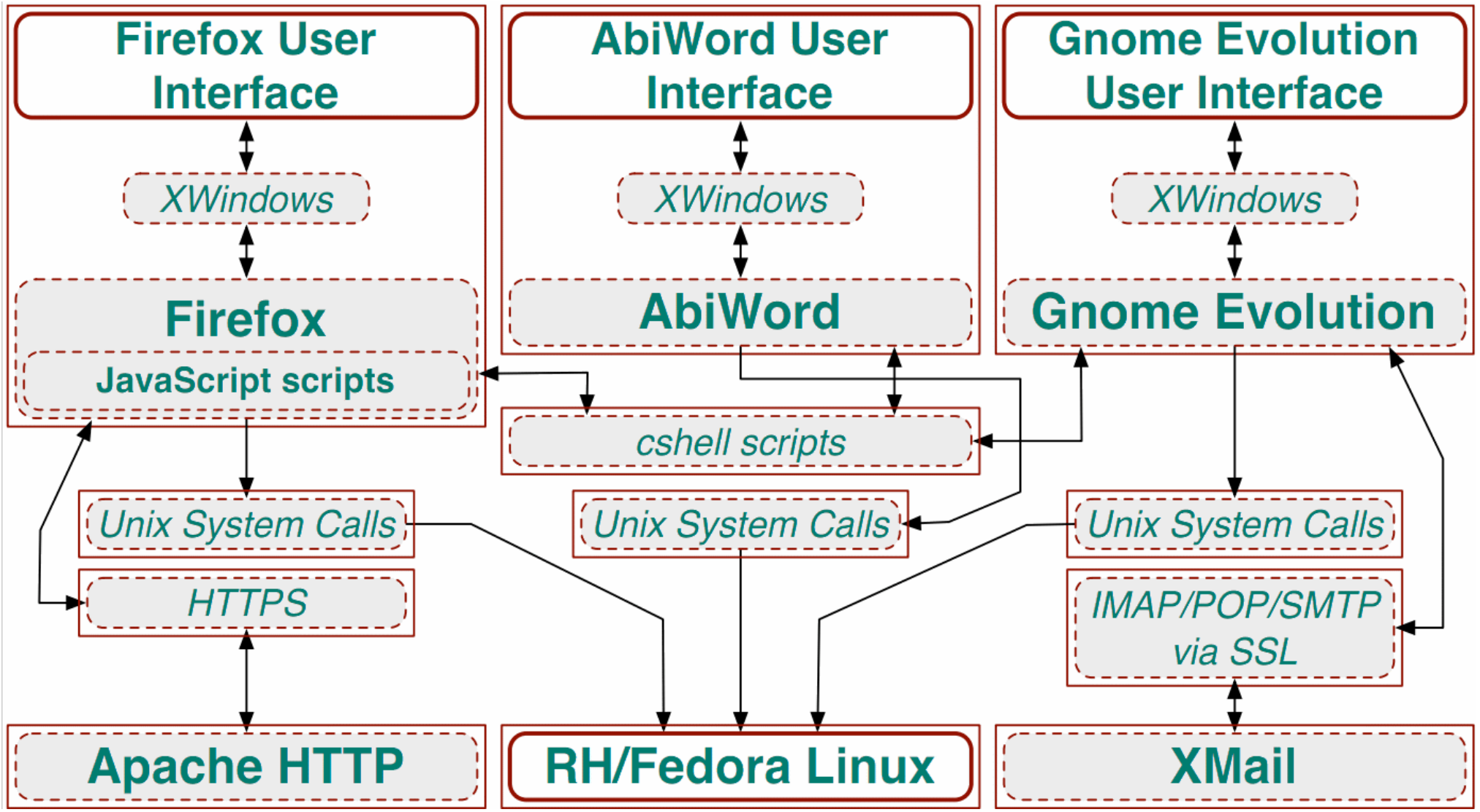


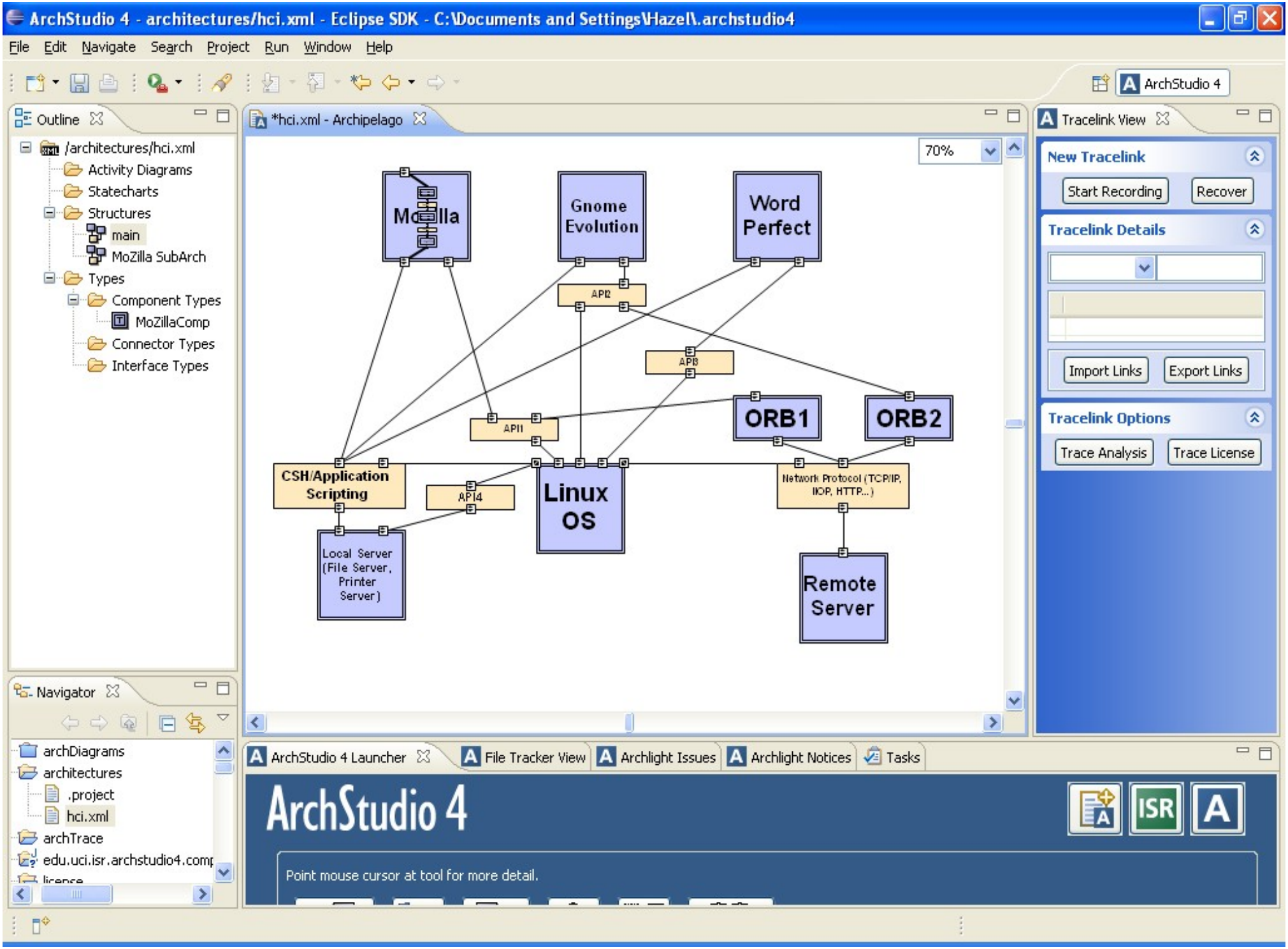
Key: Containment Vessel | Architecture Element

OA design decisions for implementation



Build-time OA example

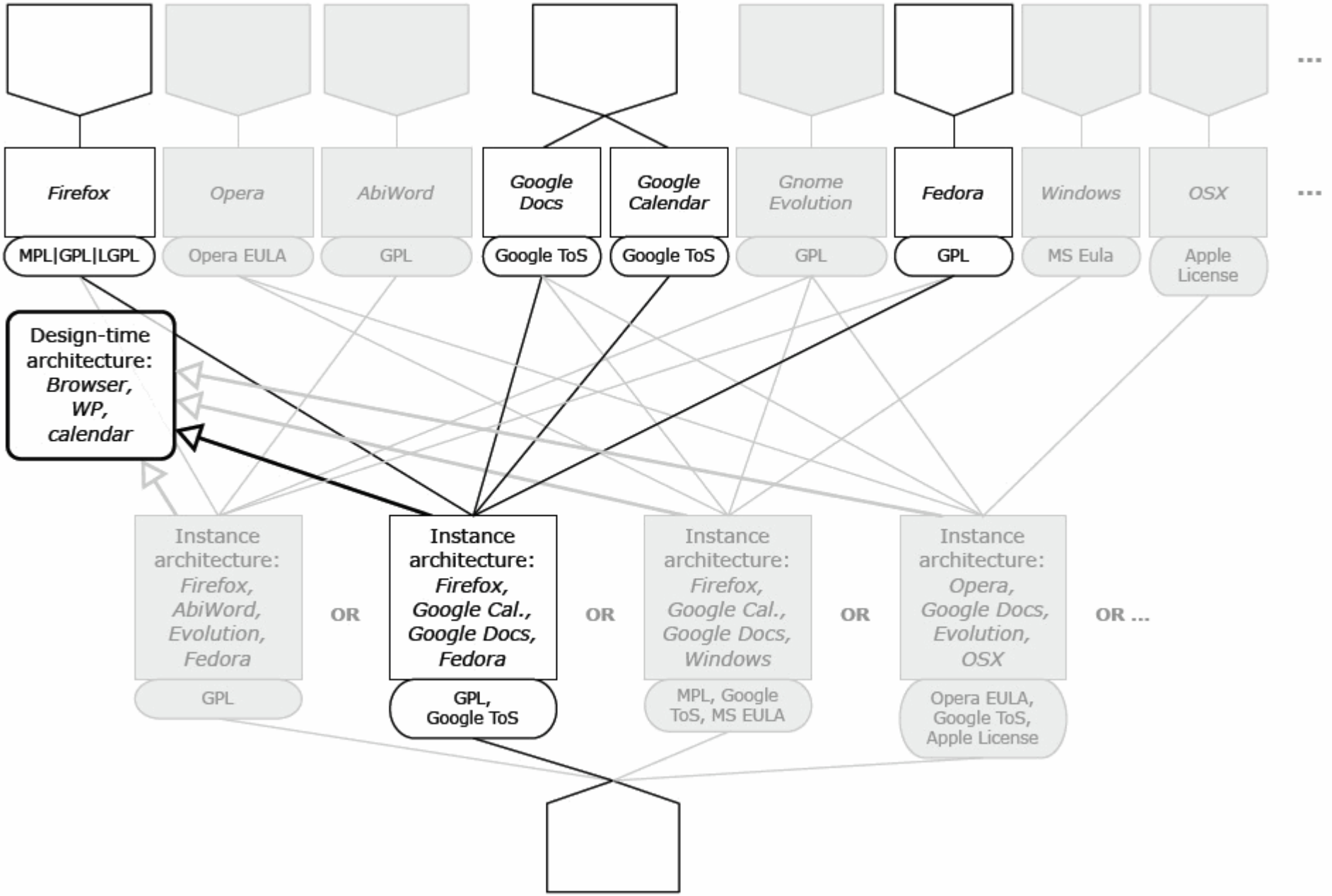



ArchStudio 4 - architectures/hci.xml - Eclipse SDK - C:\Documents and Settings\Hazel\.archstudio4
File Edit Navigate Search Project Run Window Help
ArchStudio 4
Outline
/architectures/hci.xml
Activity Diagrams
Statecharts
Structures
main
MoZilla SubArch
Types
Component Types
MoZillaComp
Connector Types
Interface Types
*hci.xml - Archipelago
70%
Mozilla
Gnome Evolution
Word Perfect
API2
API3
API1
ORB1
ORB2
CSH/Application Scripting
API4
Linux OS
Network Protocol (TCP/IP, IIOP, HTTP...)
Local Server (File Server, Printer Server)
Remote Server
Tracelink View
New Tracelink
Start Recording
Recover
Tracelink Details
Import Links
Export Links
Tracelink Options
Trace Analysis
Trace License
Navigator
archDiagrams
architectures
.project
hci.xml
archTrace
edu.uci.isr.archstudio4.comp
ArchStudio 4 Launcher
File Tracker View
Archlight Issues
Archlight Notices
Tasks
ArchStudio 4
ISR
A
Point mouse cursor at tool for more detail.

OSS elements subject to different IP licenses

- Intellectual Property licenses stipulate rights and obligations regarding use of the IP
  - GPL (Gnu Public License) stipulate right to access, study, modify, and *reciprocal* obligation to redistribute modified source
  - Mozilla now offers a "tri-license" for its software like Firefox:
    - GPL, MPL (lightweight), or Restricted (accommodating proprietary services)
  - Other OSS covered by different rights and obligations
- How to determine which rights and obligations will apply to a configured system?
  - At *design-time* (maximum flexibility)
  - At *build-time (*may/not be able to redistribute components at hand)
  - At *run-time* (may/not need to install/link-to components from other sources)
- Use architectural meta-models to define OA whose component license interactions and architecture can be formally specified, analyzed, and realized with automated OA design environment
  - T. Alspaugh, W. Scacchi, and H. Asuncion, Software Licenses in Context: The Challenge of Heterogeneously Licensed Systems, *Journal of the Association for Information Systems*, 11(11), 730-755, November 2010.

Alternative OA design decision



OSSD corporate strategies

OSSD corporate strategies: case studies

- AT&T/Lucent Technologies
- Eclipse (IBM)
- European SpaceAgency
- Hewlett-Packard
- Nokia

OSSD Strategy Summary

- AT&T/Lucent Technologies demonstrated how to start up OSSD projects by creating a small, research focused OSSD project that built from open, international standards to eventually involve dozens of contributors from their customers, competitors, and end-users.
- European Space Agency provided examples for how to use international standards to drive OSSD on different multi-vendor, systems integration projects.
- HP pioneered the concept and practice of OSS and OSSD projects behind the corporate firewall, its major (hardware) products now rely on OSSD projects, and it contributes resources and intellectual property to the support of dozens of external OSSD projects, and thousands of internal OSSD projects.
- Nokia and IBM actively support the OSSD community internal and external to their firms, has acquired software product competitors and open sourced their source code, encourages industry participations with its customers, competitors and end-users, and supports independent OSSD projects.
- Contractors/market leaders will compete on providing integrated systems and/or support services for OSS-based ground systems

OSSD and Ground Systems

(including *decentralized command and control systems* -- DC2)

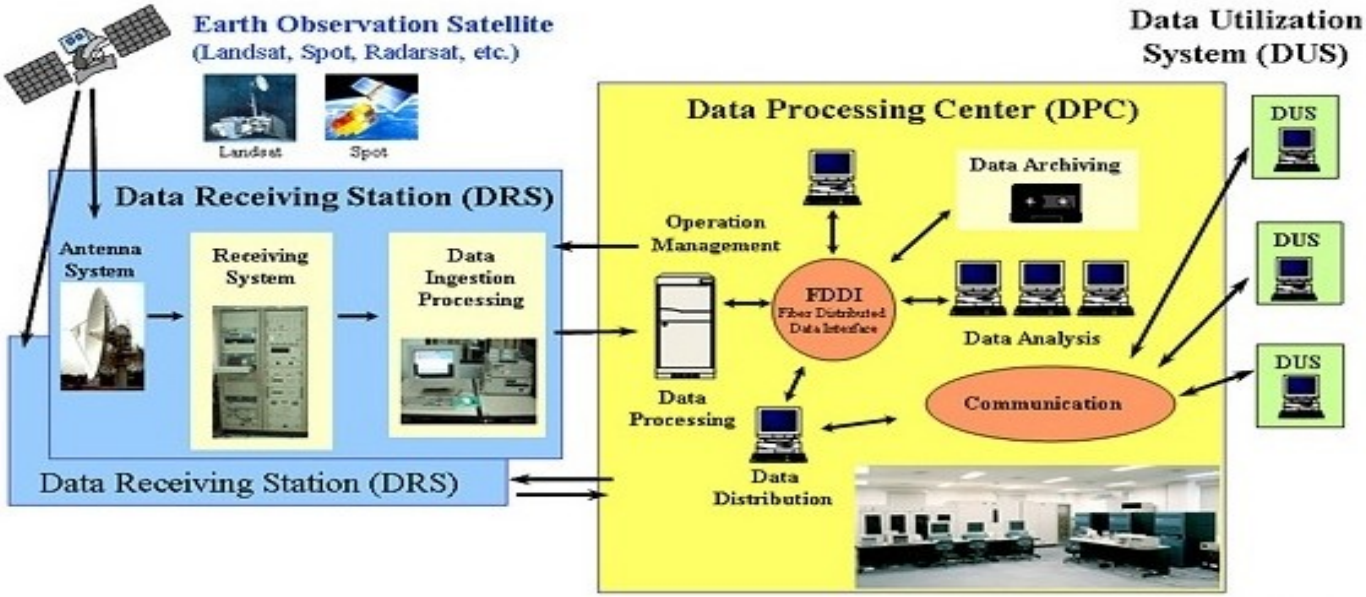
Ground systems?

TT&C (Telemetry, Tracking & Command)

ARABSAT TTC&M Station13m DIA. Antenna
(Riyadah, Saudi Arabia)

JAXA TT&C Station
(Kiruna, Sweden)

Data Processing

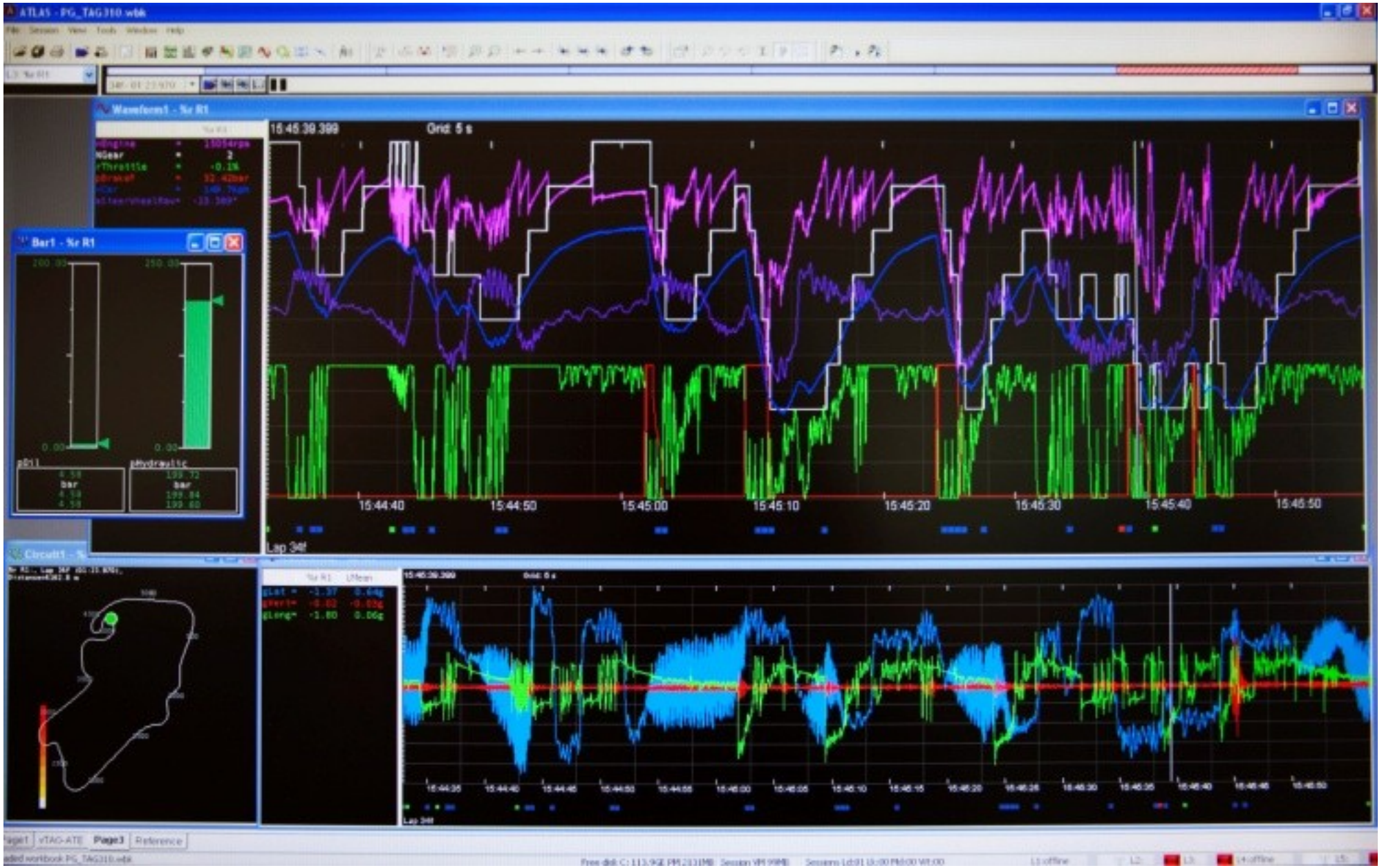


Conventional C2 facility for mission management

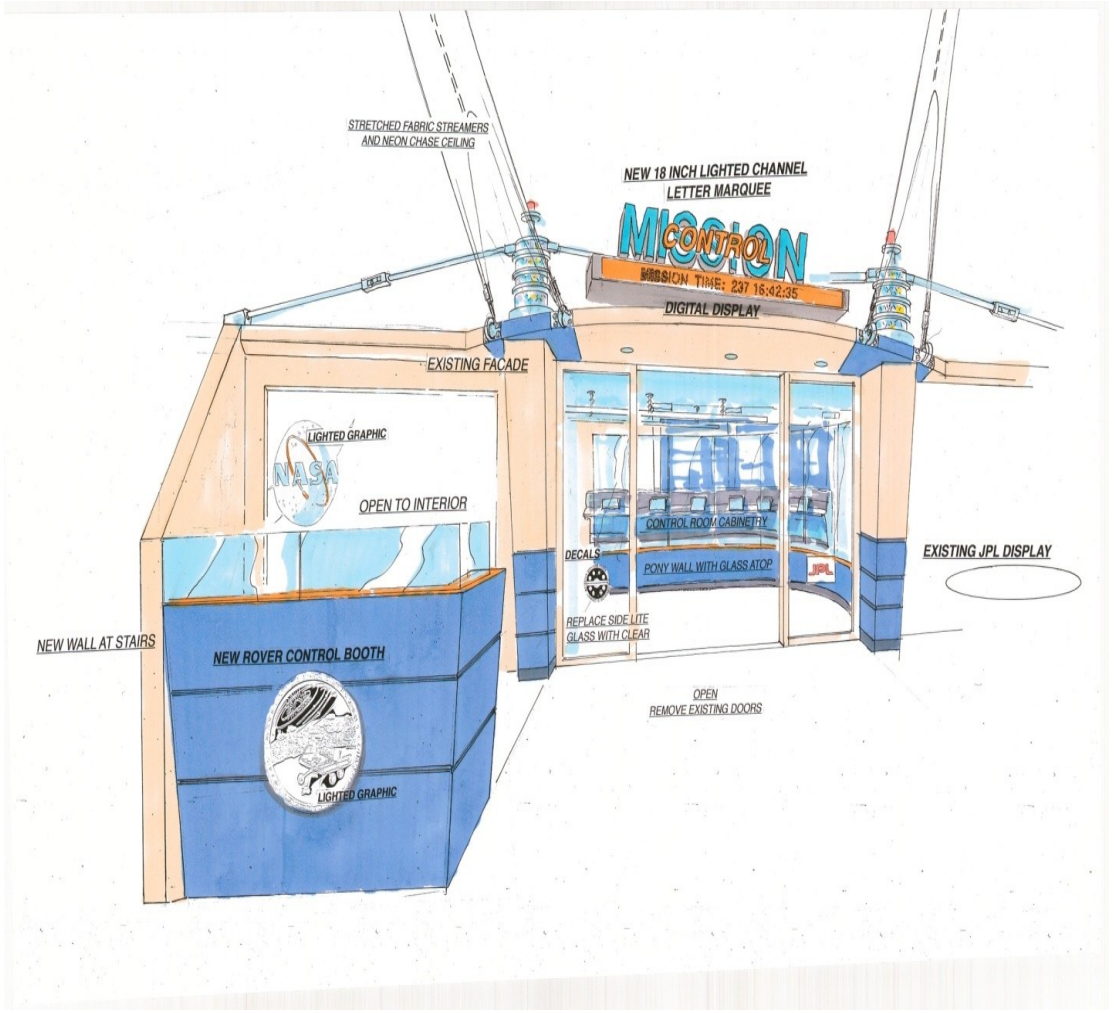
Conventional C2 facility: *Formula 1* race control

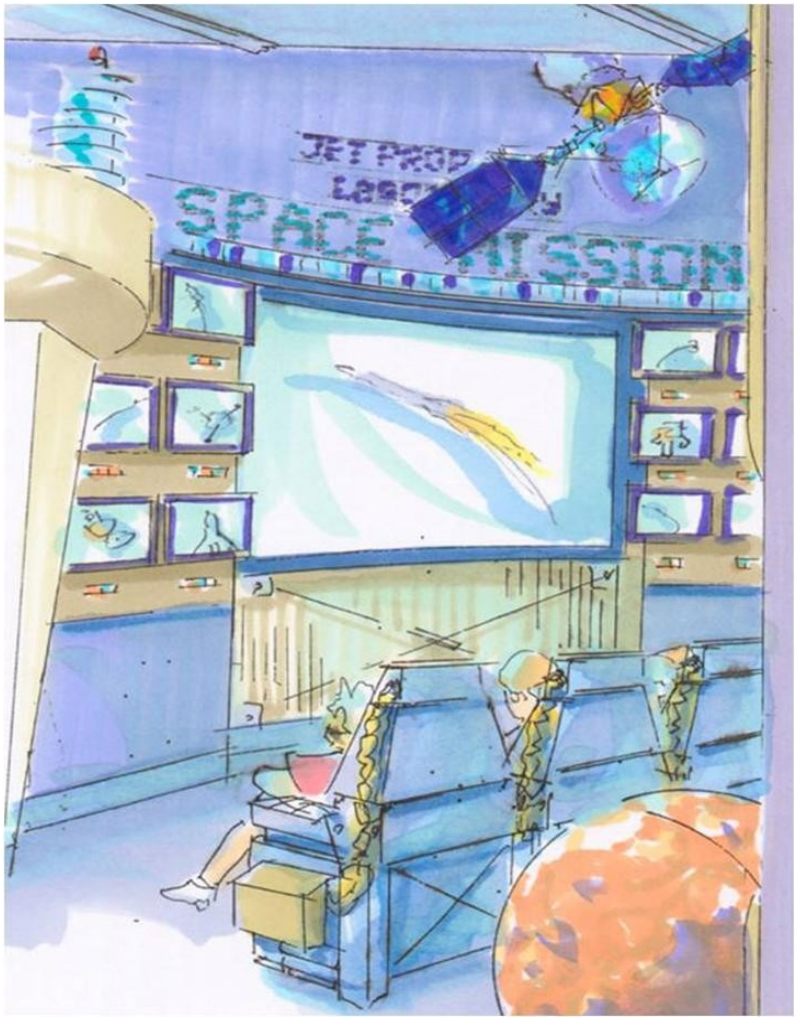
F1 race team ground systems: trackside vehicle telemetry, data processing trailer, satellite link to factory

F1 race control: multiple sensors, real-time event telemetry streams, real-time information fusion coordinated across team sites



Mission Control Room at the *Discovery Science Center*: Child-friendly ground systems?

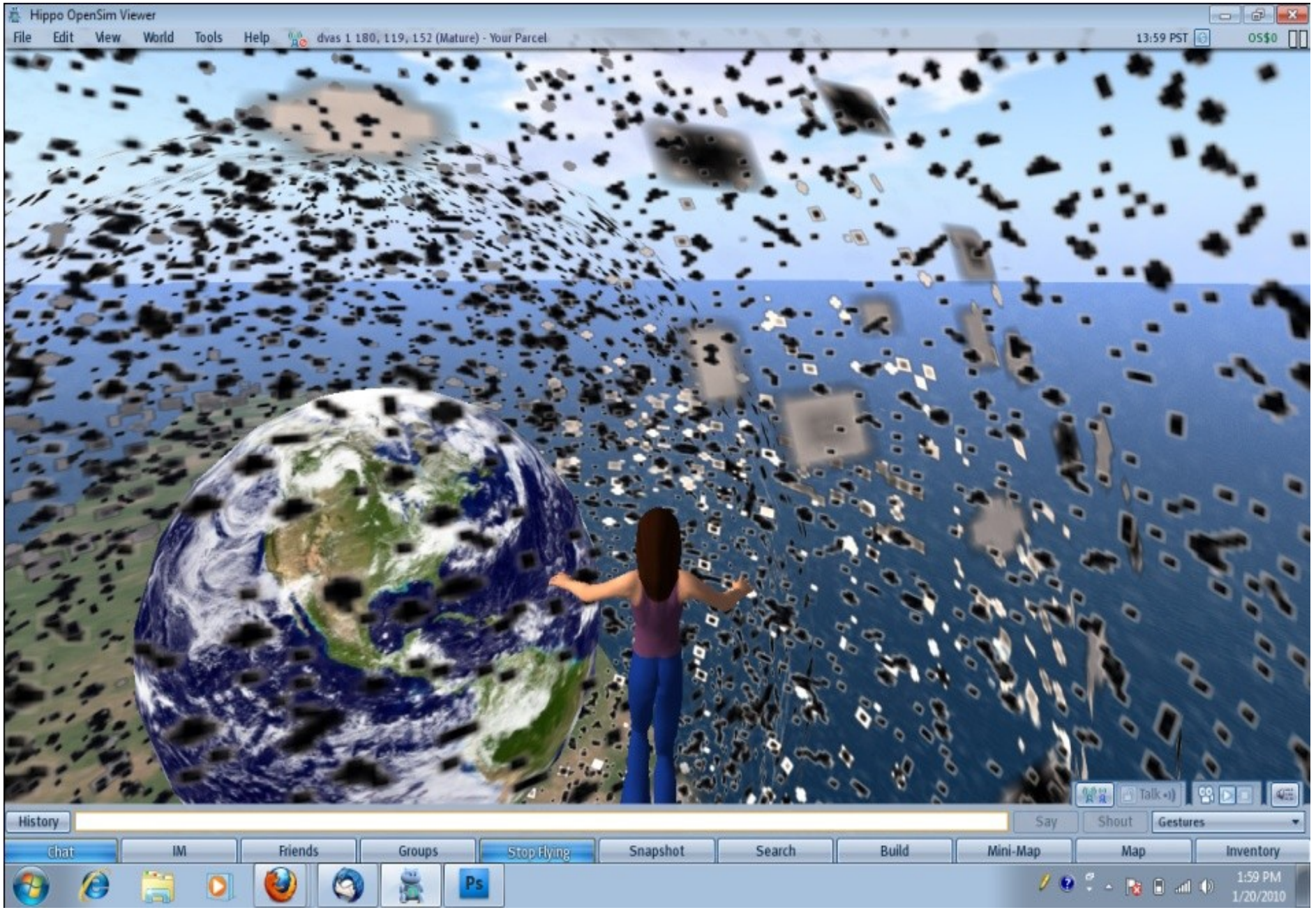




Mission Control Environment at DSC: Status (Winter 2012)

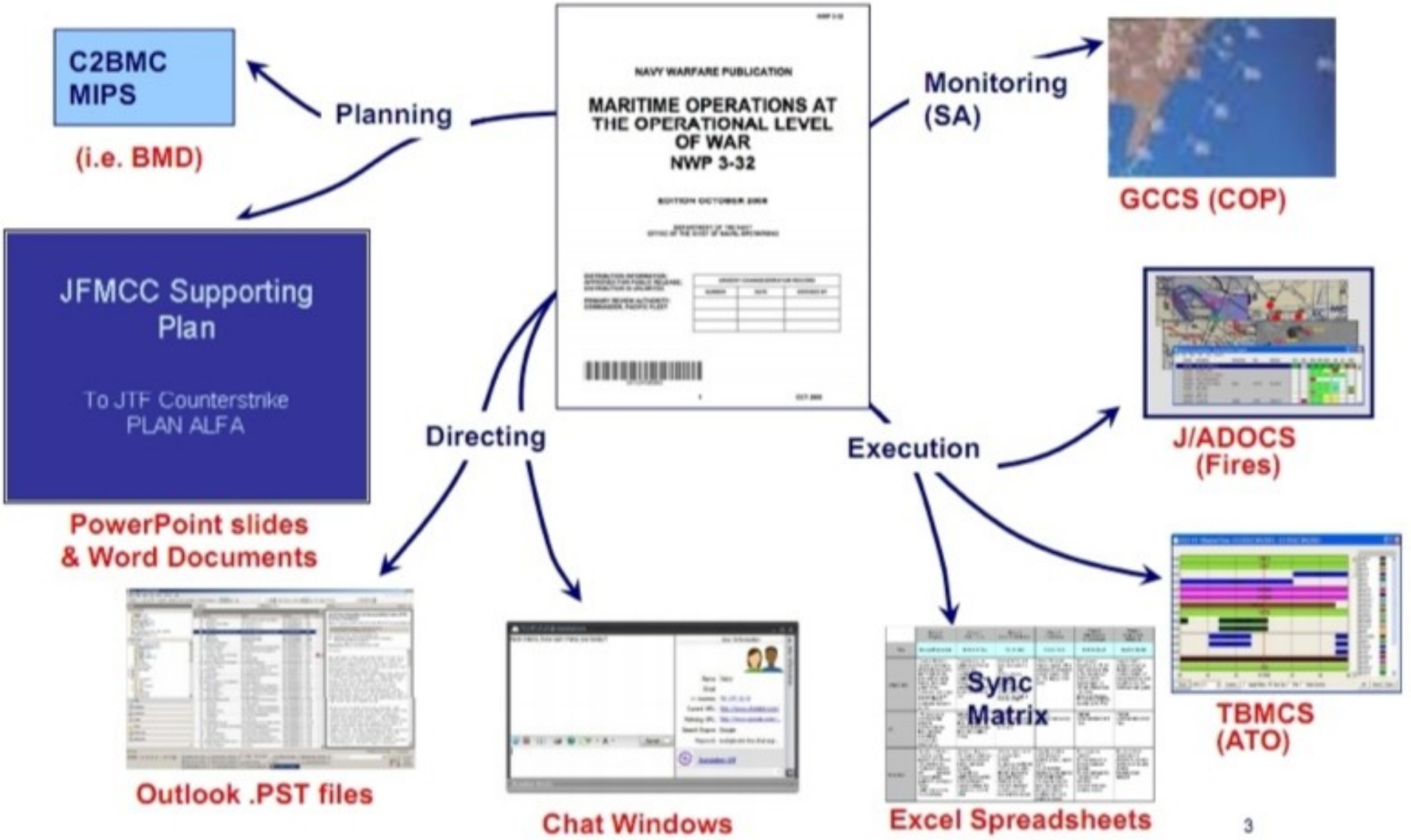
- Space Science Hall at DSC, elements in place:
  - Boeing Rocket Lab
  - Planetary Research Station
  - NASA Exploration Vehicle installation
- Mission Control Room (in development)
  - Will complete the onsite Space Science Hall at DSC
  - Partially funded (external matching fund challenge grant)
  - DSC needs to bring in resources to complete the Mission Control Room, and Mission Control game, as the onsite exhibit
    - DSC is collaborating with UCI to develop and deploy MC game
    - UCI and DSC have proposed expanding MC game to create comprehensive MCE game environment

Games/Applications for MCE: space debris tracking, fly-through, clean-up, mirrored objects trajectories, etc. using OSS (*OpenSim*) and synthetic data



Play testing mission management game mockup: incremental resource allocation with shifting uncertainty and outcomes

Ground systems (or C2 systems) should embed heterogeneous software system applications within extensible *open architectures*



C2RPC (2010). Command and Control Rapid Deployment Continuum Overview, http://www.afcea-sd.org/wp-content/uploads/2010/12/YoungAFCEA_C2RPC.pdf

Decentralized C2 (DC2) Systems coming soon

"To support high-tempo distributed operations..., this concept envisions decentralized command and control to the extend possible in both planning and execution."

Department of Defense, *Joint Operation Access Concept* (JOAC), Version 1.0, 17 January 2012.

Also see, W. Scacchi, C. Brown, and K. Nies, *Investigating the Use of Computer Games and Virtual Worlds for Decentralized Command and Control*, Final Report, Grant #N00244-10-1-0064, Institute for Software Research, University of California, Irvine, Irvine, CA, Fall 2011.

What is *Decentralized Command and Control (DC2)*?

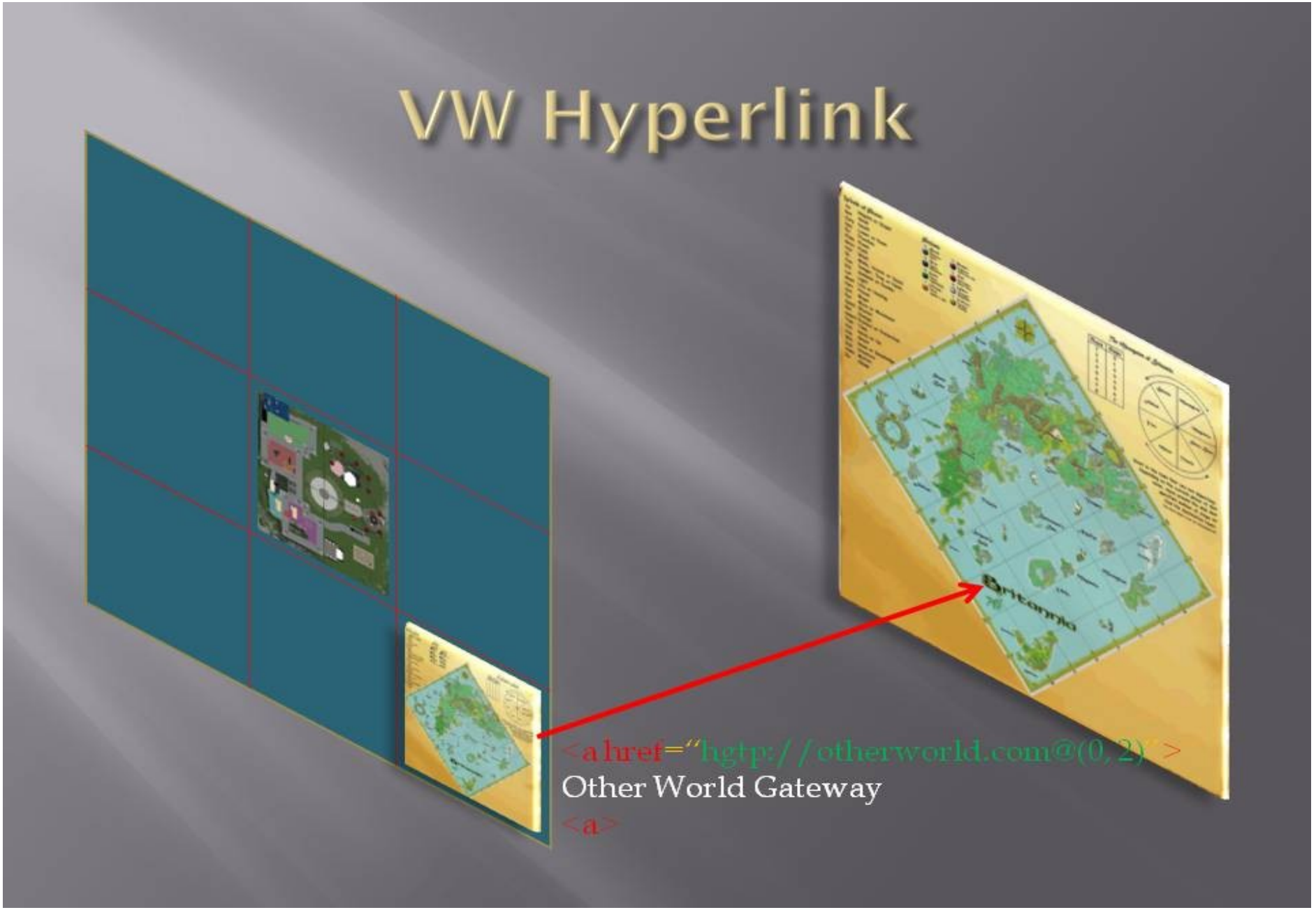
DC2 systems are C2 systems that can be:

- logically centralized and physically dispersed
- operated as virtual enterprises
- used at “edge” of an multi-site organization
- engage agents (avatars) and human actors
- engage actors in different locations that are virtually collocated in a DC2 virtual world
- “cloud” friendly

Virtualized ground system/DC2 facility using *OpenSim*

Virtualized DC2 actors/avatars: where are their users located?

DC2 virtual worlds can be linked into interoperable *hypergrids*



C. Lopes, (2011). Hypergrid: Architecture and Protocol for Virtual World Interoperability, *IEEE Internet Computing*, 15(5), 22-29, Sept-Oct 2011.

DC2 worlds can be linked into *hypergrids* across heterogeneous platforms

Military Open Simulator Enterprise Strategy: http://fvwc.army.mil/moses/

New DC2 platforms: personal, mobile, built into other devices/platforms

DC2 diverse clients (multi-version implementations, OSS and closed source software) accessing common world

Lessons to learn

- What is *Decentralized Command and Control (DC2)*?
  - Something to master; invest in now
- OSS Computer games and virtual worlds for DC2
  - Enable new ways and means for experimenting with new DC2 concepts, techniques, workforce
- Applications for OSS DC2
  - Potential for DC2 product line architectures and (domain-specific) application generators
- Scaling DC2 systems: capability versus cost
  - Finding the sweet-spot in scalable capability/cost.
- OSS DC2 challenges and opportunities for Cyber-security
  - Great R&D opportunities; visualize security

Discussion and limitations

OSSD and Validation

- OSSD is a decentralized approach to software development
  - OSSD processes can also be open source!
- OSS is amenable to decentralized validation
- OSS may be better suited for independent validation and continuous improvement across enterprises than closed source software
- OSSD for ground systems may work best when conceived as transcending the efforts of a single enterprise.

FOSSD research limitations

- Individual participation
  - Some form of reciprocity and intrinsic, self-serving motivation is necessary
- Cooperation, coordination, and control
  - Negotiation and conflict management are part of the cost FOSS developers incur in order to have their believes fulfilled
  - Time, effort, and attention are spent negotiating socio-technical dependencies

Research opportunities

- Results from study of cooperation, coordination and control in FOSSD
  - Virtual project management (VPM) and role migration can provide a lightweight approach to SE project management
  - Unclear whether proprietary software projects willing to embrace VPM

FOSSD research limitations

- Alliances and community development
  - FOSSD projects give rise to *new kinds of requirements* for community building, community software, and community information sharing systems
  - Alliances and community require attention to sustain their effectiveness, and to prevent them from becoming self-serving and bureaucratic

Research opportunities

- Alliance formation and social networking results suggest SE projects operate at a disadvantage compared to FOSSD projects
  - SE projects tend to produce systems whose growth/evolution is limited
  - FOSSD projects can produce systems capable of sustained exponential growth/evolution of both software and developer-user community

FOSSD research limitations

- Empirical studies of FOSSD are expanding the scope of what we can observe, discover, analyze, and learn about large software systems.
  - Mining software repositories
  - Multi-modal modeling and analysis of socio-technical processes and networks found in sustained FOSSD projects

Research opportunities

Research opportunities

- FOSSD is poised to alter the calculus of empirical SE
  - Software process discovery, modeling, and simulation
  - Repository mining can support software visualization, refactoring/redesign studies
  - Compare SE versus FOSSD approaches to software inspection and peer review
  - OSS components and processes for future ground systems?

FOSS systems R&D areas

- Development processes, practices, and project forms
- Collaboration
- Ecosystems
- Evolution
- Instrumentation and infrastructure

Development processes, practices, and project forms

- What are the
  - development processes,
  - work practices,
  - alternative project organizational forms

  that give rise to successful FOSS systems?
- What works where, when, why and how, and for whom?

Collaboration

- How does the practice of developing large or very large scale software systems depend on the collaborative work practices and communities of practice found in successful FOSS system projects?
- How do software licenses facilitate or inhibit collaboration among global software developers?

Ecosystems

- How do FOSS systems emerge within a complex, decentralized web of people, artifacts, practices, and other infrastructural resources while most FOSS projects fail to take root and thrive?
- How do those few that do succeed become widespread and transform industry, government, or science practices?

Evolution

- How can successful FOSS systems continue to grow, develop (within releases), and evolve (across releases) across ever larger communities of developer-users at sustained exponential rates?
  - Via evolution/replacement of components, architecture, component licenses, project forms, communities, tools, practices, etc.
- To what end, and following what processes?

Instrumentation and infrastructure

- *Scalability*:
  - Research studies range from small-scale studies of individual FOSS projects to very-large populations of FOSS projects
- *Repositories and Meta-Repositories:*
  - Each FOSS project is an ecology of mostly informal online artifacts
  - Source code and meta-data are formal
- *Data analysis tools and analytics*
  - Supporting text and software data mining, (process) knowledge discovery, data visualization, provenance, and archiving of “executable research papers”
  - See FLOSSmole, FLOSShub, FLOSSmetrics.

FLOSSmole view of FOSS Forges

FLOSSmole

Collaborative collection and analysis of free/libre/open source project data

About | Getting Data | Using Data | Donating Data | Blog

Navigation

- Recent posts

User login

Username: *

Password: *

Log in

Log in using OpenID

- Request new password

Like 6 likes. Sign Up to see what your friends like.

Search

Search this site:

Search

Getting Data

- Code Forges Study
- Database schema
- Data collection details
- Direct database access
- Download data (at Google Code)

Using Data

Home

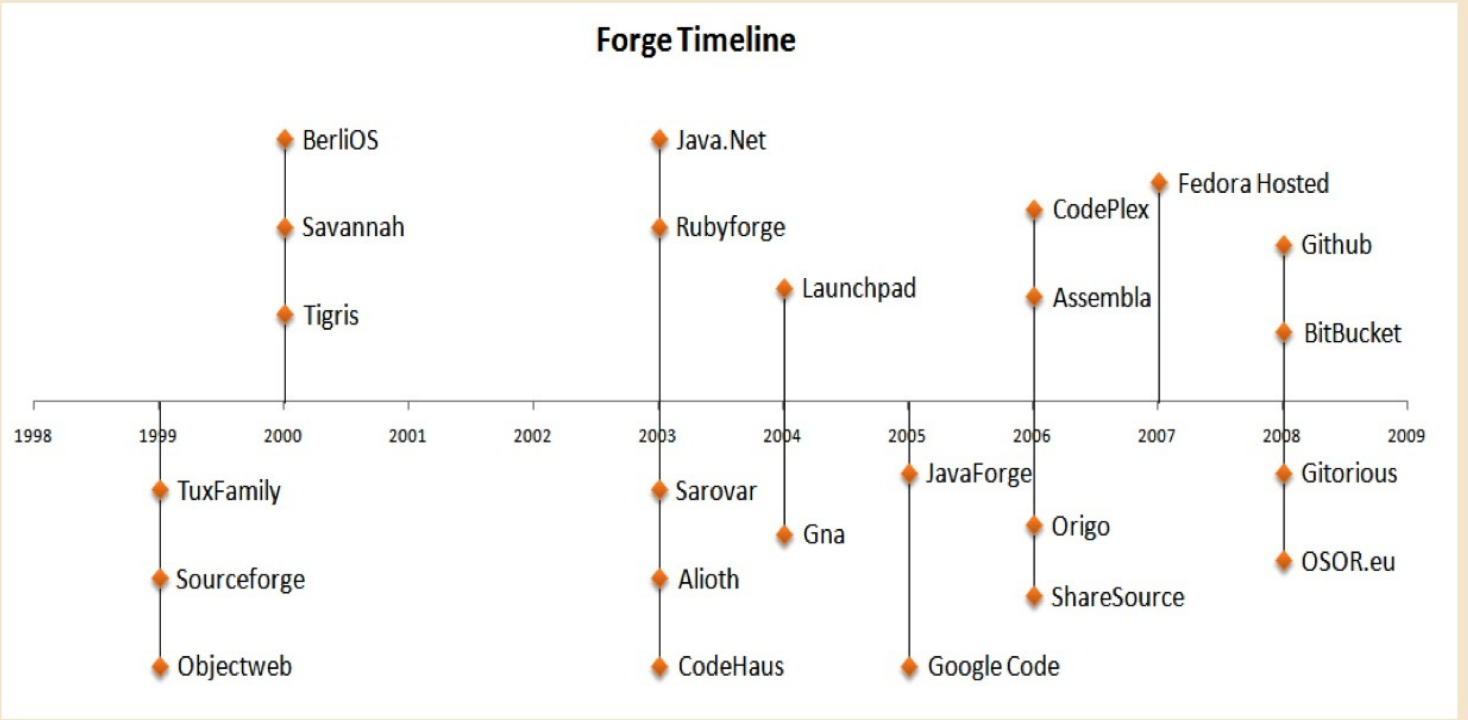
When were the forges established?

Submitted by **dwilliams** on Fri, 06/03/2011 - 11:19 Examples

Description

This timeline illustrates when the forges were officially established.

Visualization



Further readings on FOSS

- Scacchi, W. (2007). Free/Open Source Software Development: Recent Research Results and Emerging Opportunities. *Proc. 6th. ESEC/FSE*, 459–468. Also see, Free/Open Source Software Development: Recent Research Results and Methods, in M.V. Zelkowitz (ed.), *Advances in Computers*, 69, 243-295, 2007.
- Gasser, L. and Scacchi, W. (2008). Towards a Global Research Infrastructure for Multidisciplinary Study of Free/Open Source Software Development, in *Open Source Development, Community and Quality*; B. Russo, E. Damiani, S. Hissan, B. Lundell, and G. Succi (Eds.), IFIP Vol. 275, Springer, Boston, MA. 143-158.
- Hauge, O., Ayala, C. and Conradi, R. (2010). Adoption of Open Source Software in Software-Intensive Organizations - A Systematic Literature Review. *Information and Software Technology,* 52(11), 1133-1154.
- Aksulu, A. and Wade, M.R. (2010). A Comprehensive Review and Synthesis of Open Source Research, *J. Assoc. Info. Systems*, 11(11), 576-656.
- Scacchi, W., Crowston, K., Jensen, C., Madey, G., Squire, M., and others (2010). *Towards a Science of Open Source Systems*, Final Report, Computing Community Consortium, November 2010. http://foss2010.isr.uci.edu/content/foss-2010-reports/
- Höst, M., Oručević-Alagić, A. (2011). A Systematic Review of Open Source Software in Commercial Software Product Development, *Information and Software Technology*, 53(6), June, 616-624.
- Crowston, K., Wei, K., Howison, J., and Wiggins, A. (2011). Free/libre open source software development: what we know and what we do not know. *ACM Computing Surveys*, (to appear).

Acknowledgements

- *Recent collaborators*:
  - Kevin Crowston, SyracuseU; Les Gasser, UIllinois, Urbana-Champaign; James Howison, UTexas, Austin; Chris Jensen, Google; Greg Madey, Notre Dame University; John Noll, LERO (Ireland); Megan Squire, Elon University
  - Thomas Alspaugh and others at ISR
  - UCI Video Game Developers Club (students)
- *Research funding support (**No endorsement implied**)*:
  - National Science Foundation: Grants #0083075, #0205679, #0205724, #0350754, #0534771, #0749353, #0808783, #1041918.
  - Acquisition Research Program, Naval Postgraduate School (2007-2012)
  - Center for the Edge, Naval Postgraduate School, Grant #N00244-10-1-0064
  - Computing Community Consortium (2009-2011)
  - Northrop Grumman Information Systems

Thank you!