



Acquisition Research Program:  
Creating Synergy for Informed Change

# The Challenge of Heterogeneously Licensed Systems in Open Architecture Software Ecosystems

Thomas A. Alspaugh and Walt Scacchi  
Institute for Software Research, University of California, Irvine

# Overview

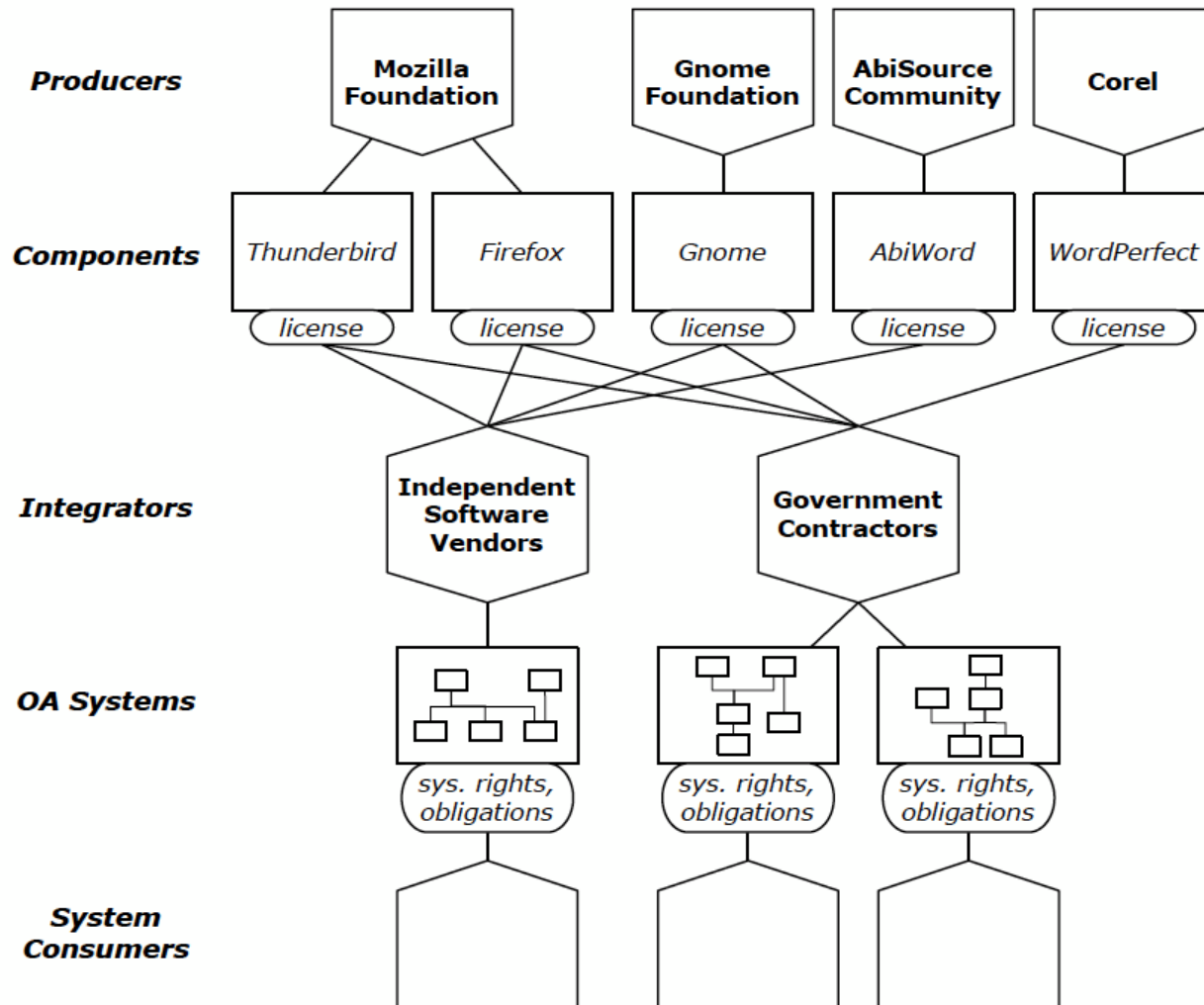
- Background
- Developing Open Architectures (OAs)
- Software licenses, architectures, and analysis
- Discussion
- Conclusions



# Background



# OA software ecosystem – mapping rights and obligations from producers to consumers



# A heterogeneous software license for the *Unity 3D* software system

1. The Mono Class Library, Copyright 2005-2008 Novell, Inc.
2. The Mono Runtime Libraries, Copyright 2005-2008 Novell, Inc.
3. Boo, Copyright 2003-2008 Rodrigo B. Oliveira
4. UnityScript, Copyright 2005-2008 Rodrigo B. Oliveira
5. OpenAL cross platform audio library, Copyright 1999-2006 by authors.
6. PhysX physics library. Copyright 2003-2008 by Ageia Technologies, Inc.
7. libvorbis. Copyright (c) 2002-2007 Xiph.org Foundation
8. libtheora. Copyright (c) 2002-2007 Xiph.org Foundation
9. zlib general purpose compression library. Copyright (c) 1995-2005 Jean-loup Gailly and Mark Adler
10. libpng PNG reference library
11. jpeglib JPEG library. Copyright (C) 1991-1998, Thomas G. Lane.
12. Twilight Prophecy SDK, a multi-platform development system for virtual reality and multimedia. Copyright 1997-2003 Twilight 3D Finland Oy Ltd
13. dynamic bitset, Copyright Chuck Allison and Jeremy Siek 2001-2002.
14. The Mono C# Compiler and Tools, Copyright 2005-2008 Novell, Inc.
15. libcurl. Copyright (c) 1996-2008, Daniel Stenberg <[daniel@haxx.se](mailto:daniel@haxx.se)>.
16. PostgreSQL Database Management System
17. FreeType. Copyright (c) 2007 The FreeType Project ([www.freetype.org](http://www.freetype.org)).
18. NVIDIA Cg. Copyright (c) 2002-2008 NVIDIA Corp.



# Supporting OA with heterogeneously licensed system components

- Must account for design-time, build-time, and run-time architectures
- Must distinguish architect constructs relevant to software licenses, and license effects
- Must define license architecture
- Must provide automated environment for managing system and license architectures
- Must automate calculations of system license rights, obligations, architectures as they evolve



# Developing OA systems with heterogeneously licensed software components



# A composed multi-component OA system

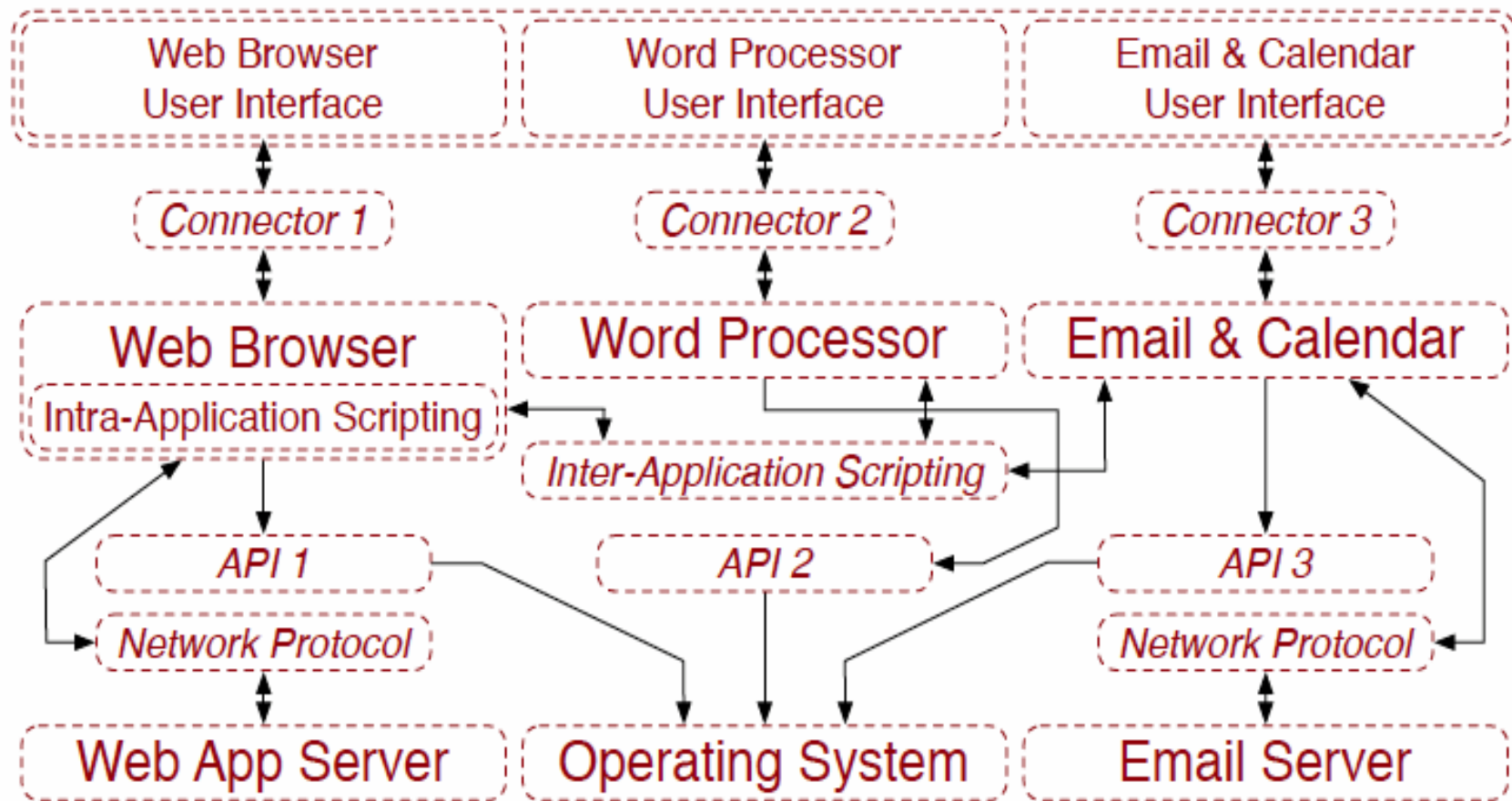
The screenshot displays a multi-component Open Architecture (OA) system running on a desktop environment. The system is composed of several interconnected components:

- Web Browser (Mozilla Firefox):** Displays the "GAME CULTURE & TECHNOLOGY LAB" website. The page includes a navigation menu (Mission, Events, Projects, Tools, Players, Package), a search bar, and a "Mission" section. The mission statement reads: "The mission of the Game Culture & Technology Lab is to play with how game metaphors, design principles, and technologies can be utilized for alternative content and context delivery. The focus is on the next generation Internet and beyond." It also describes the approach and methods used, and mentions the lab's history since its inception in 1999.
- Calendar (Evolution):** Shows a calendar for Monday, April 26, 2010. Key events include a "Proposal review meeting" at 1:00pm and "Work on JSS paper draft" at 3:00pm. The calendar also displays a monthly view for April 2010.
- Document Editor (LibreOffice Writer):** Displays a document titled "JSS-Figure4-draft.abw". The document content is "A Composed Open Architecture Software System at Run-Time". The editor interface includes a menu bar, a toolbar, and a status bar.
- Terminal Window (Terminal):** Shows the command prompt for the user "liveuser@localhost". The user has executed the command `cd ../selinux` and is viewing the contents of the `selinux` directory. The output is a long list of files and directories, including `dmraid.static`, `ifconfig`, `lvchange`, `modinfo`, `pppoe-sniff`, `sfdisk`, `vgremove`, `dmsetup`, `ifdown`, `lvconvert`, `modprobe`, `pppoe-start`, `shutdown`, `vgrename`, `dosfstck`, `lvsave`, `lvcreate`, `mount.fuse`, `pppoe-status`, `slattach`, `vgs`, `dosfslabel`, `ifrename`, `lvdisplay`, `pppoe-stop`, `sln`, `vgscan`, `dump`, `ifup`, `lvextend`, `ppp-watch`, `start`, `vgsplit`, `dump2fs`, `init`, `lvm`, `pvchange`, `start_udev`, `weak-modules`, `e2fsck`, `initctl`, `lvchange`, `mount.nfs`, `prck`, `status`, `ypbind`, `e2image`, `initlog`, `lvmdiskscan`, `mount.nfs-3g`, `pvcreate`, `stop`, `ypbind`, `bootfs`, `initlog`, `mount.nfs-fuse`, `pvcreate`, `stop`, `ypbind`.

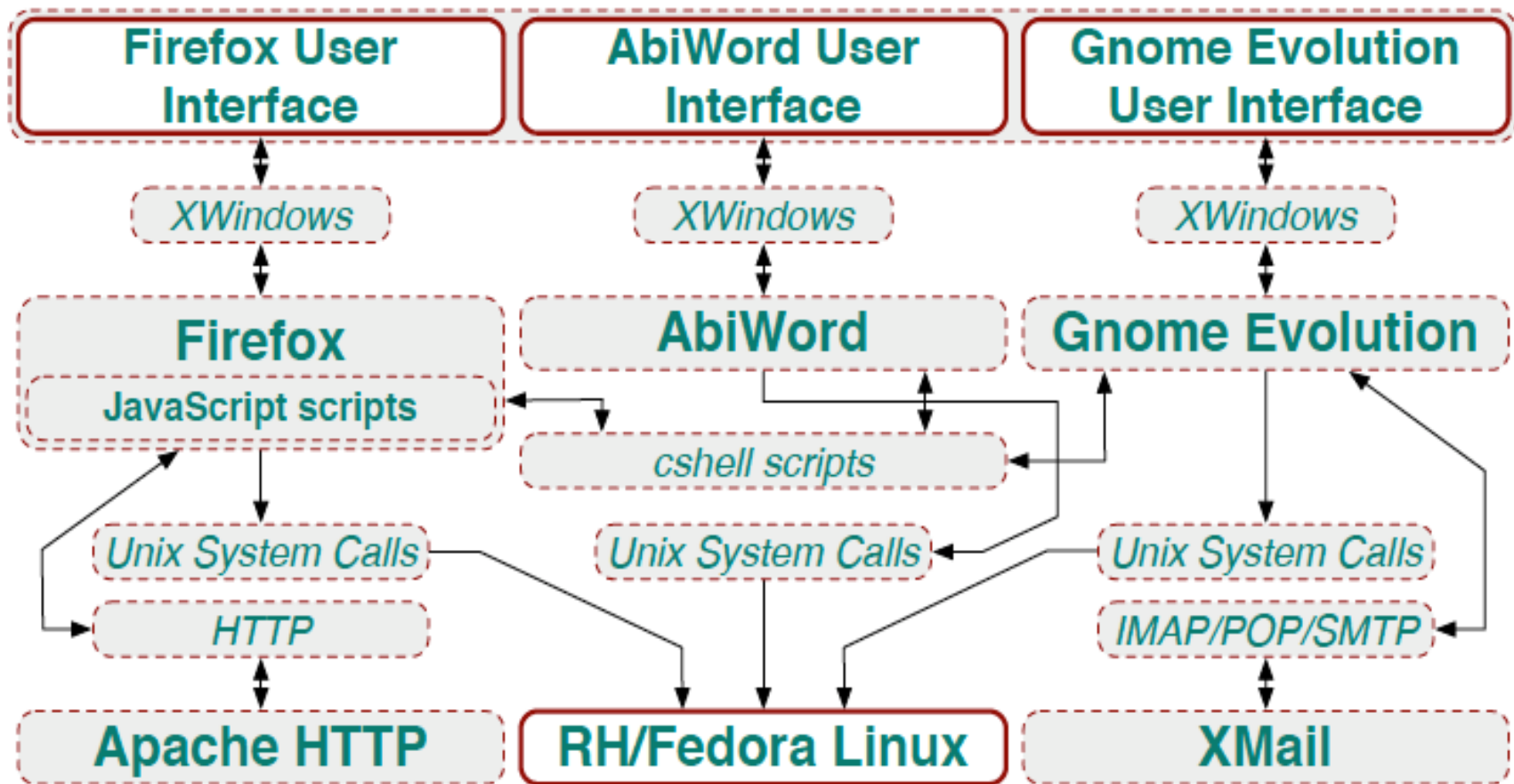




# Design-time architecture



# Build-time architecture



# Run-time architecture (user view)

The screenshot displays a desktop environment with several windows open. The top-left window is a Mozilla Firefox browser showing the 'GAME CULTURE & TECHNOLOGY LAB' website. The top-right window is a document editor titled '\*JSS-Figure4-draft.abw' containing the text 'A Composed Open Architecture Software System at Run-Time'. The bottom-left window is a calendar application for Monday, April 26, 2010, showing a schedule with a 1:00pm meeting and a 3:00pm work session. The bottom-right window is a terminal window showing the output of the 'ls' command in the /sbin directory.

```
liveuser@localhost:~$ ls
[liveuser@localhost sbin]$ pwd
/sbin
[liveuser@localhost sbin]$ cd ../selinux
[liveuser@localhost selinux]$ ls
access      checkreqprot  compat_net  deny_unknown  initial contexts  nls          policyvers  user
avc         class          context     disable        load              null         reject_unknown  relabel
boolean     commit_pending_boots  create      enforce        load             member      policy_capabilities
```



# Evolutionary changes in OA Systems

- Component evolution
- Component replacement
- Architecture evolution
- Component license evolution
- Change in desired rights or acceptable obligations
  
- Evolutionary changes reconfigure a system's software ecosystem!



# Component replacement and architecture evolution

The screenshot displays a desktop environment with three main windows:

- GCTL - Mission - Mozilla Firefox:** A website titled "GAME CULTURE & TECHNOLOGY LAB". It features a navigation menu with "Mission", "Events", "Projects", "Tasks", "Projects", and "Package". Below the menu is a search bar with "English" selected and a "Language" dropdown. The main content area includes a "Mission" section with text: "The mission of the Game Culture & Technology Lab is to play with how game metaphors, design principles, and technologies can be utilized for alternative contexts and context delivery. The focus is on the next generation Internet and beyond." It also mentions "The approach combines theory and practice, art and science, education and entertainment, to create an environment that supports diverse forms of expression in a wide range of applications." and "The methods include sampling, misuse, hacking, appropriation, reverse engineering, and custom creation in the interest of open-source innovation and critical intervention." A final paragraph states: "Since its inception in 1999, the Game Lab has been physically housed in the Claire Trevor School of the Arts (SOTA). In 2002 we established a co-located facility in The California Institute for Telecommunication and Information Technology (Calit2)." There are also "prev" and "next" links.
- A Composed Open Architecture Software... - Google Docs - Mozilla Firefox:** A Google Docs document titled "A Composed Open Architecture Software System at Run-Time". The document content is partially obscured by other windows but shows a screenshot of the GCTL website and a terminal window. The terminal window shows the output of the command `ls` in the `/sbin` directory, listing various system binaries and SELinux modules.
- Google Calendar - Mozilla Firefox:** A Google Calendar window showing the date "Monday, Apr 26, 2010". The calendar view shows a grid for April 2010, with the 26th highlighted. A task list on the right shows a "2:30p Proposal review meeting" and a "9:30p work on jcs paper draft".

The terminal window in the Google Docs document shows the following output:

```
liveuser@localhost:/sbin$ ls
dmraid_static  ifconfig          lvchange          modinfo           pppoe-sniff      sfdisk            vgrename
dmssetup       ifdown            lvconvert         modprobe          pppoe-start      shutdown         vgrname
dosfsck        ifenslave         lvcreate          mount.fuse        pppoe-status     slattach         vgs
dosfslabel     ifrename          lvdisplay         mount.nfs         pppoe-stop       sln              vgscan
dump           ifup              lvextend          mount.nfs4         ppp-watch        start            vgsplit
dumpe2fs       init              lvm               mount.ntfs         pvck              start_udev       weak-modules
e2fsck         initctl           lvchange          mount.ntfs-3g     pvcreate          status            ypbind
e2image        initlog           lvm2diskscan     mount.ntfs-fuse   pvcree           stop              ypbind

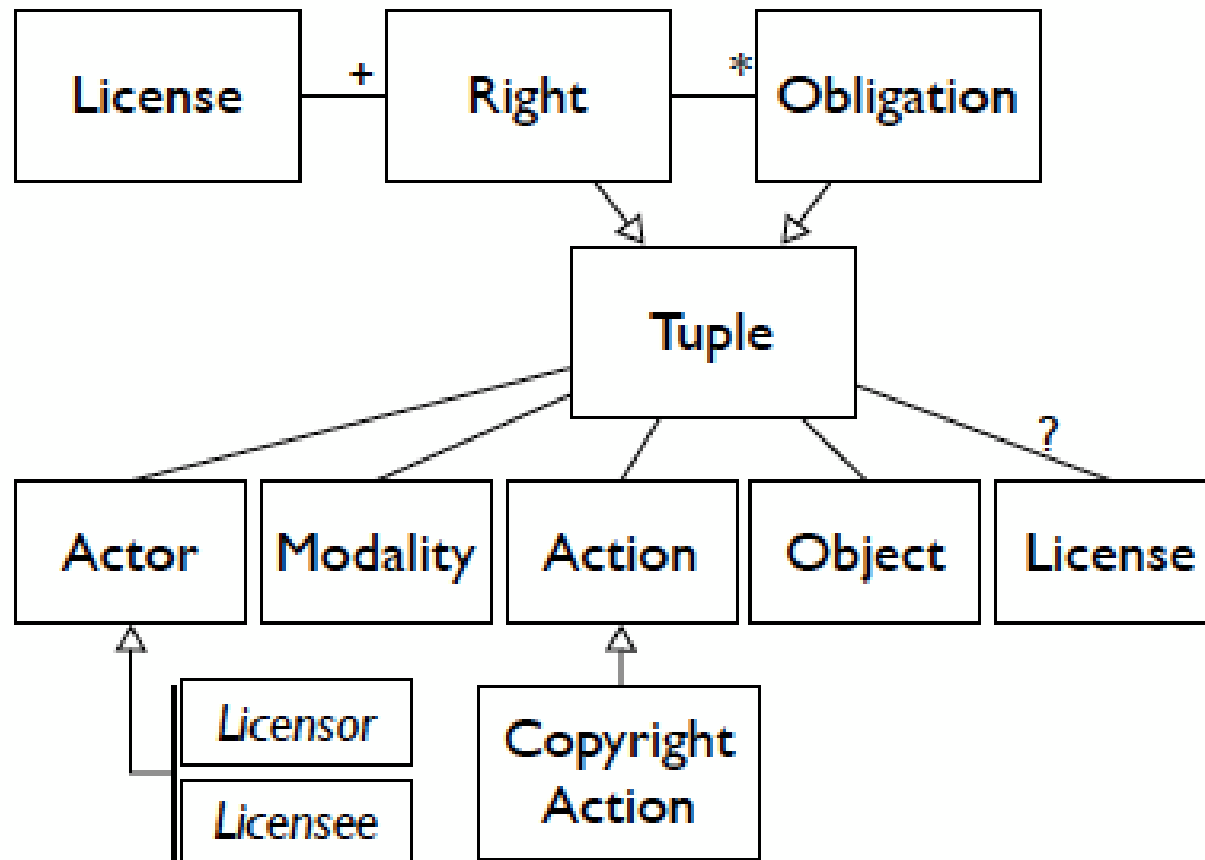
liveuser@localhost/sbin$ pwd
/sbin
liveuser@localhost/sbin$ cd ../selinux
liveuser@localhost/selinux$ ls
access  checkreqprot  compat_net  deny_unknown  initial_contexts  mls  policyvers  user
avc     class          context     disable        load               null  reject_unknown
booleans  commit_pending_boots  create      enforce        member             policy_capabilities  relabel
```



# Software licenses, architectures, and analysis



# Software license meta-model



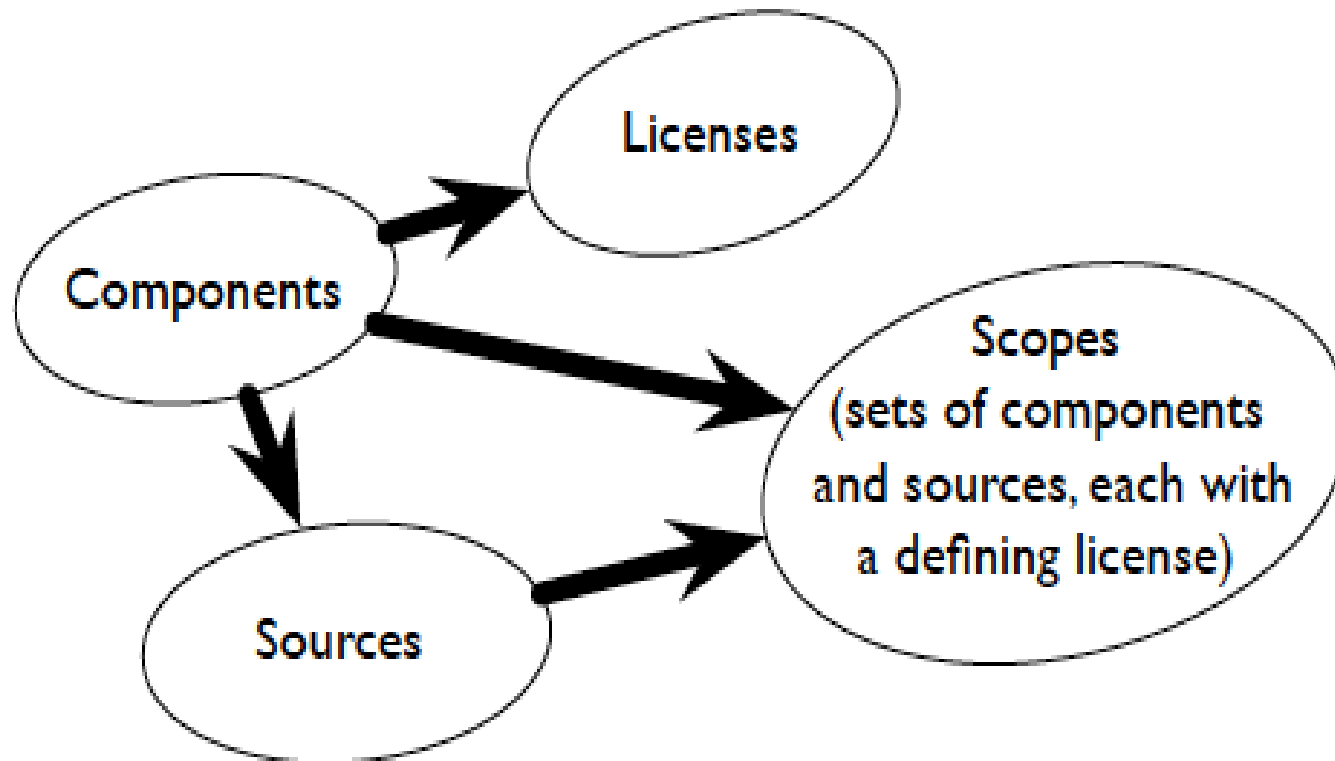
# Logical modality and objects of software license rights and obligations

	Modality	Object	License (optional)
Abstract Right	<i>May or Need Not</i>	<i>Any Under This License</i>	<i>This License or Object's License</i>
		<i>Any Source Under This License</i>	
		<i>Any Component Under This License</i>	
Concrete Right		Concrete Object	Concrete License
Concrete Obligation			
Abstract Obligation	<i>Must or Must Not</i>	<i>Right's Object</i>	<i>Concrete License or Right's License</i>
		<i>All Sources Of Right's Object</i>	
		<i>X Scope Sources</i>	
		<i>X Scope Components</i>	





# The software license architecture meta-model

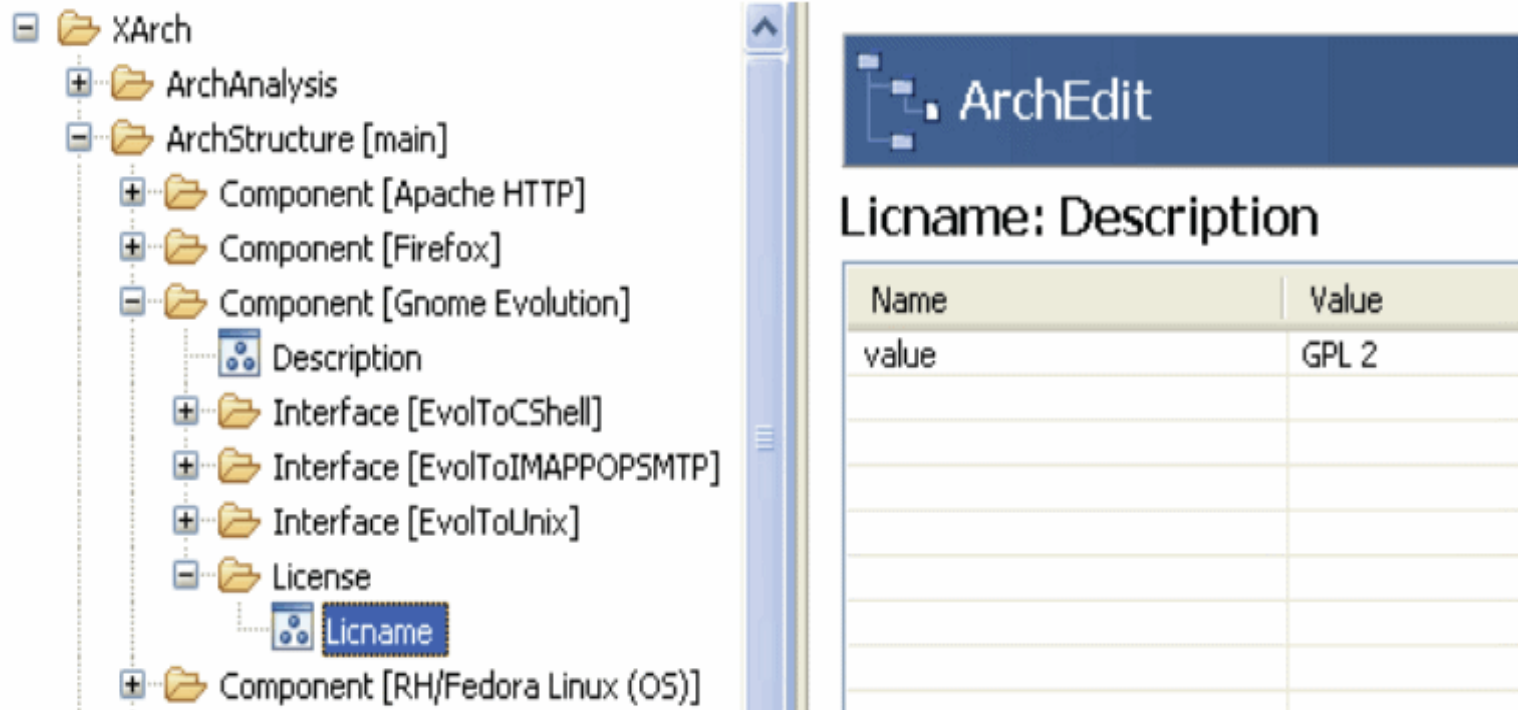


# Software license analysis

- License types:
  - Strongly reciprocal (GPL), weakly reciprocal (LGPL), academic (BSD), Terms of Service, Proprietary
- Propagation of reciprocal obligations
- Conflicting obligations
- Calculating obligations and rights



# Component license annotation prior to analysis



ArchEdit

Licname: Description

Name	Value
value	GPL 2



# License review during license analysis

The screenshot displays the ArchStudio 4 IDE interface. The main window shows a dependency graph with nodes for Firefox, WordProcessor, and GnomeEvolution, all connected to a central Cshell scripts node. Below this, three Unix System nodes are shown. The right-hand side of the IDE features a 'View License Information' panel with fields for License (GPL 2), Code (WordProcessor), and revision numbers. Below this is an 'Inference Tree' and 'Options' section.

The Mozilla Firefox browser window displays the GNU General Public License Version 2, June 1991. The page includes a navigation menu, a source link, and the following text:

Copyright (C) 1989, 1991 Free Software Foundation, Inc.  
59 Temple Place, Suite 330, Boston, MA 02111-1307 USA

Everyone is permitted to copy and distribute verbatim copies of this license document, but changing it is not allowed.

**Preamble**

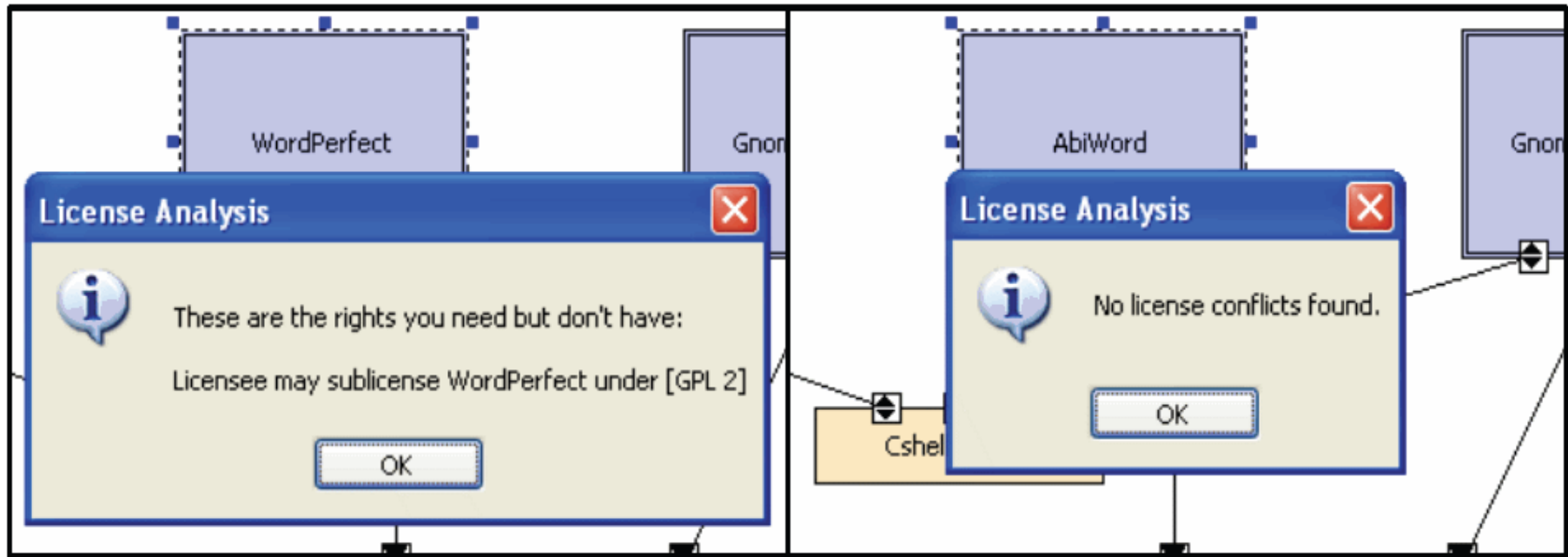
The licenses for most software are designed to take away your freedom to share and change it. By contrast, the GNU General Public License is intended to guarantee your freedom to share and change free software—to make sure the software is free for all its users. This General Public License applies to most of the Free Software Foundation's software and to any other program whose authors commit to using it. (Some other Free Software Foundation software is covered by the GNU Library General Public License instead.) You can apply it to your programs, too.

When we speak of free software, we are referring to freedom, not price. Our General Public Licenses are designed to make sure that you have the freedom to distribute copies of free software (and charge for this service if you wish), that you receive source code or can get it if you want it, that you can change the software or use pieces of it in new free programs; and that you know you can do these things.

To protect your rights, we need to make restrictions that forbid anyone to deny you these rights or to ask you to surrender the rights. These restrictions translate to certain responsibilities for you if you distribute copies of the software, or if you modify it.



# Results from license analyses with system component replacement



# Discussion



# Software product lines (SPLs) and OA systems

- An SPL may or may not be an OA system
- If SPL subject to single vendor/proprietary license, then lock-in is possible
- If OA system has design-time reference architecture and instantiated build-time architecture, then OA conforms to an SPL
- If SPL is based on OA with heterogeneously licensed components, then OA conforms to a *virtual SPL*, and works with our approach.



# Specifying and analyzing system security requirements as “licenses”

- Security capabilities can correspond to “rights and obligations” in licenses
- Should be possible to specify and analyze system *security architecture* that conform to a *security meta-model*, much like we do for software licenses
- Should be possible to develop computational tools and development environments that can analyze security at design-time, build-time, and run-time, as well as when the system evolves





# Conclusions

- Software component licenses and heterogeneously licensed systems becoming more widespread as we move to OA software ecosystems
- Our approach and tools demonstrate the ability to specify, model, and analyze such systems as they evolve, and are subject to diverse licenses
- Our approach is compatible with SPLs and can be extended to support system security licenses



# Acknowledgements

This presentation was developed under work supported by the Naval Postgraduate School Acquisition Research Program Assistance Agreement No. #N00244-10-1-0038 awarded by the U.S. Fleet Industrial Supply Center, San Diego (FISC-SD). It has not been formally reviewed by NPS. The views and conclusions contained in this document are those of the authors and should not be interpreted as necessarily representing the official policies, either expressed or implied, of the U.S. FISC-SD and NPS. The FISC-SD and NPS do not endorse any products or commercial services mentioned in this publication.

Funding also from grant #0808783 from the National Science Foundation. No review, approval, or endorsement implied.

