



**Calhoun: The NPS Institutional Archive**

**DSpace Repository**

Reports and Technical Reports | Faculty and Researchers' Publications

2010-05-12

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

Alspaugh, Thomas A.; Scacchi, Walt

Monterey, California. Naval Postgraduate School

https://hdl.handle.net/10945/33511

This publication is a work of the U.S. Government as defined in Title 17, United States Code, Section 101. Copyright protection is not available for this work in the United States.

*Downloaded from NPS Archive: Calhoun*


http://www.nps.edu/library

Calhoun is the Naval Postgraduate School's public access digital repository for research materials and institutional publications created by the NPS community. Calhoun is named for Professor of Mathematics Guy K. Calhoun, NPS's first appointed -- and published -- scholarly author.

Dudley Knox Library / Naval Postgraduate School
411 Dyer Road / 1 University Circle
Monterey, California USA 93943


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

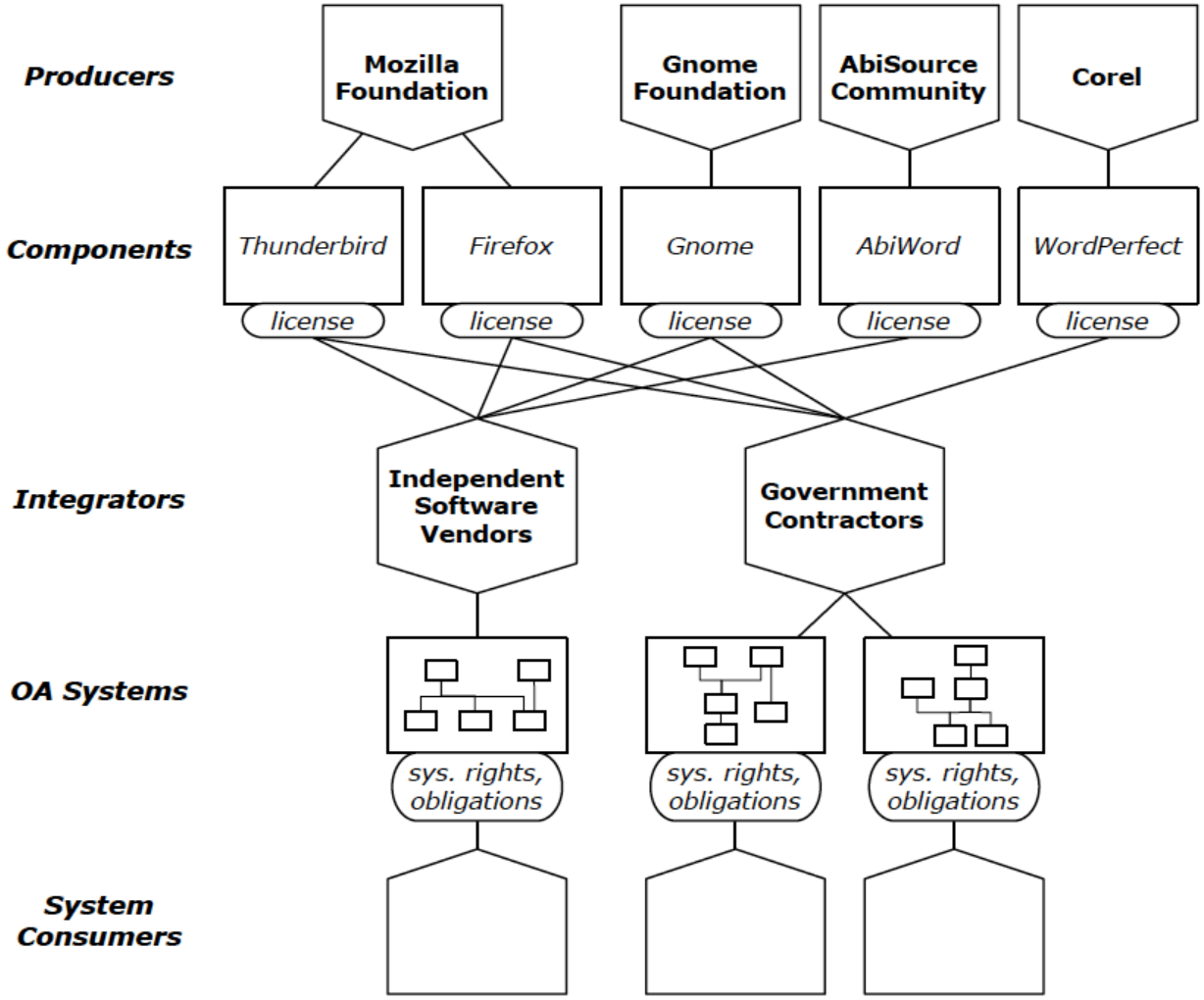
Thomas A. Alspaugh and Walt Scacchi
Institute for Software Research, University of California, Irvine

# 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

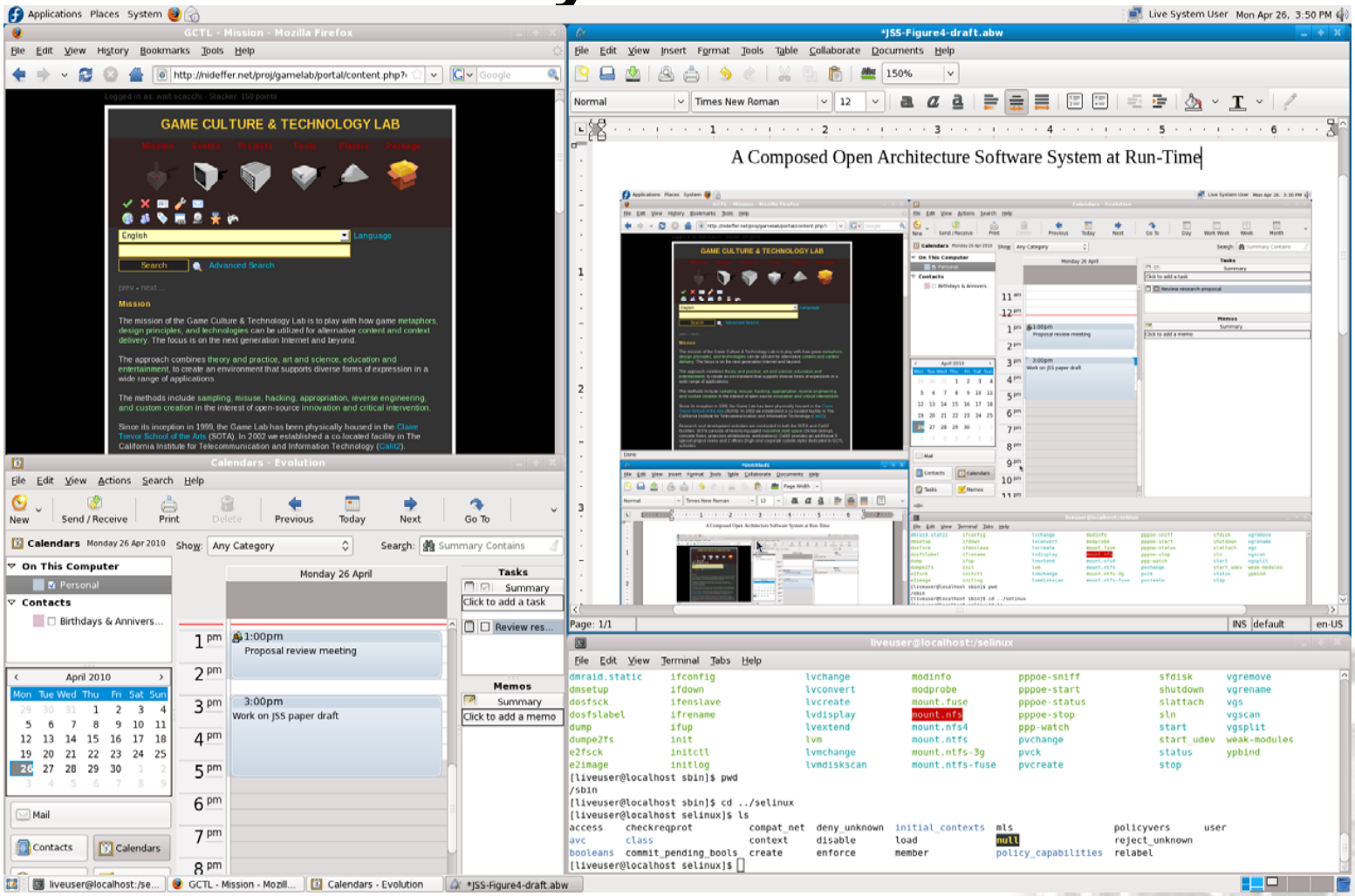
1. The Mono Class Library, Copyright 2005-2008 Novell, Inc.
2. The Mono Runtime Libraries, Copyright 2005-2008 Novell, Inc.
3. Boo, Copyright 2003-2008 Rodrigo B. Oliveira
4. UnityScript, Copyright 2005-2008 Rodrigo B. Oliveira
5. OpenAL cross platform audio library, Copyright 1999-2006 by authors.
6. 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>.
16. PostgreSQL Database Management System
17. FreeType. Copyright (c) 2007 The FreeType Project (www.freetype.org).
18. NVIDIA Cg. Copyright (c) 2002-2008 NVIDIA Corp.

# Supporting OA with heterogeneously licensed system components

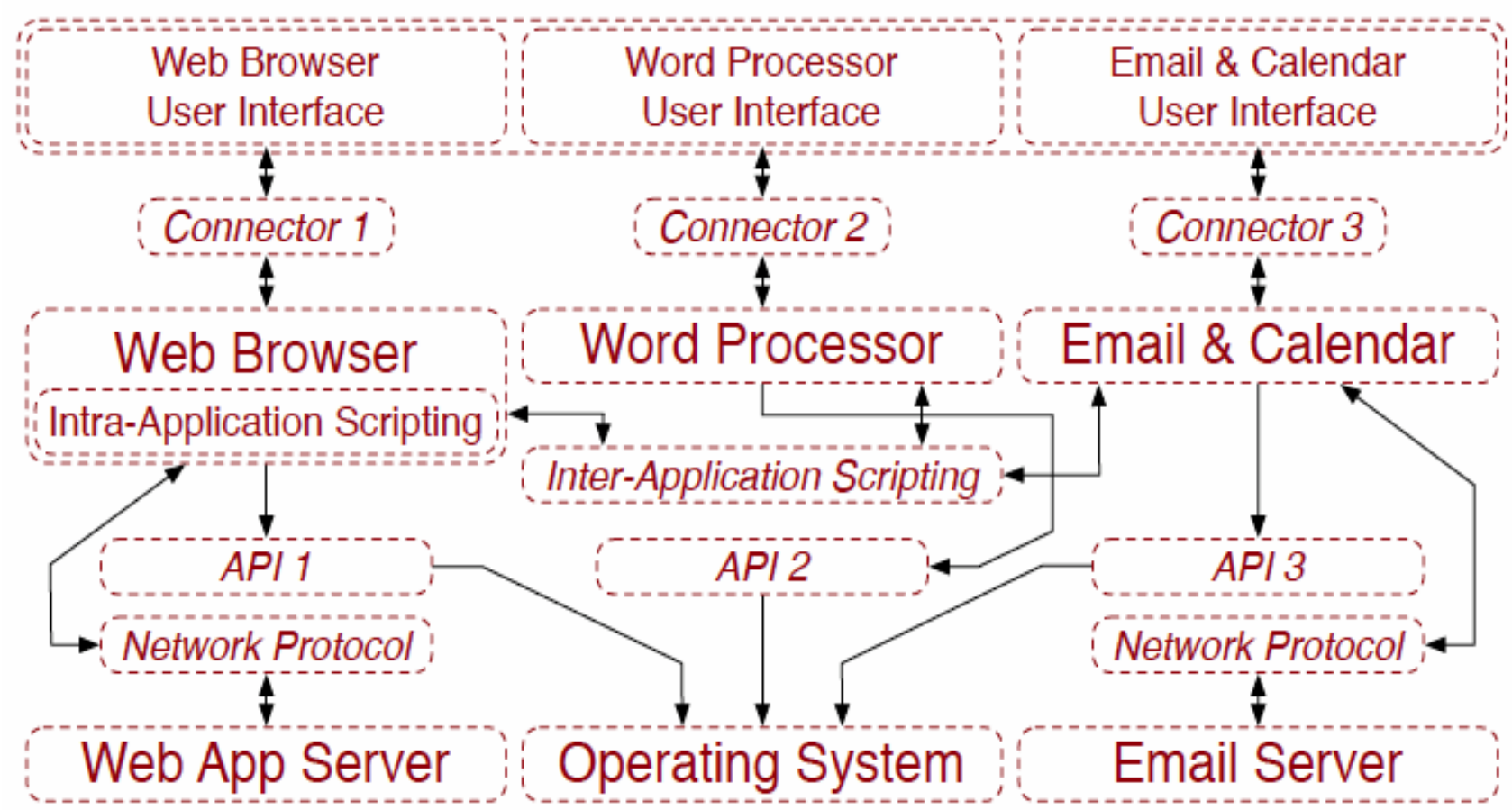
- Must account for design-time, build-time, and run-time architectures
- Must distinguish architect constructs relevant to software licenses, and license effects
- Must define license architecture
- Must provide automated environment for managing system and license architectures
- Must automate calculations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

# Developing OA systems with heterogeneously licensed software components

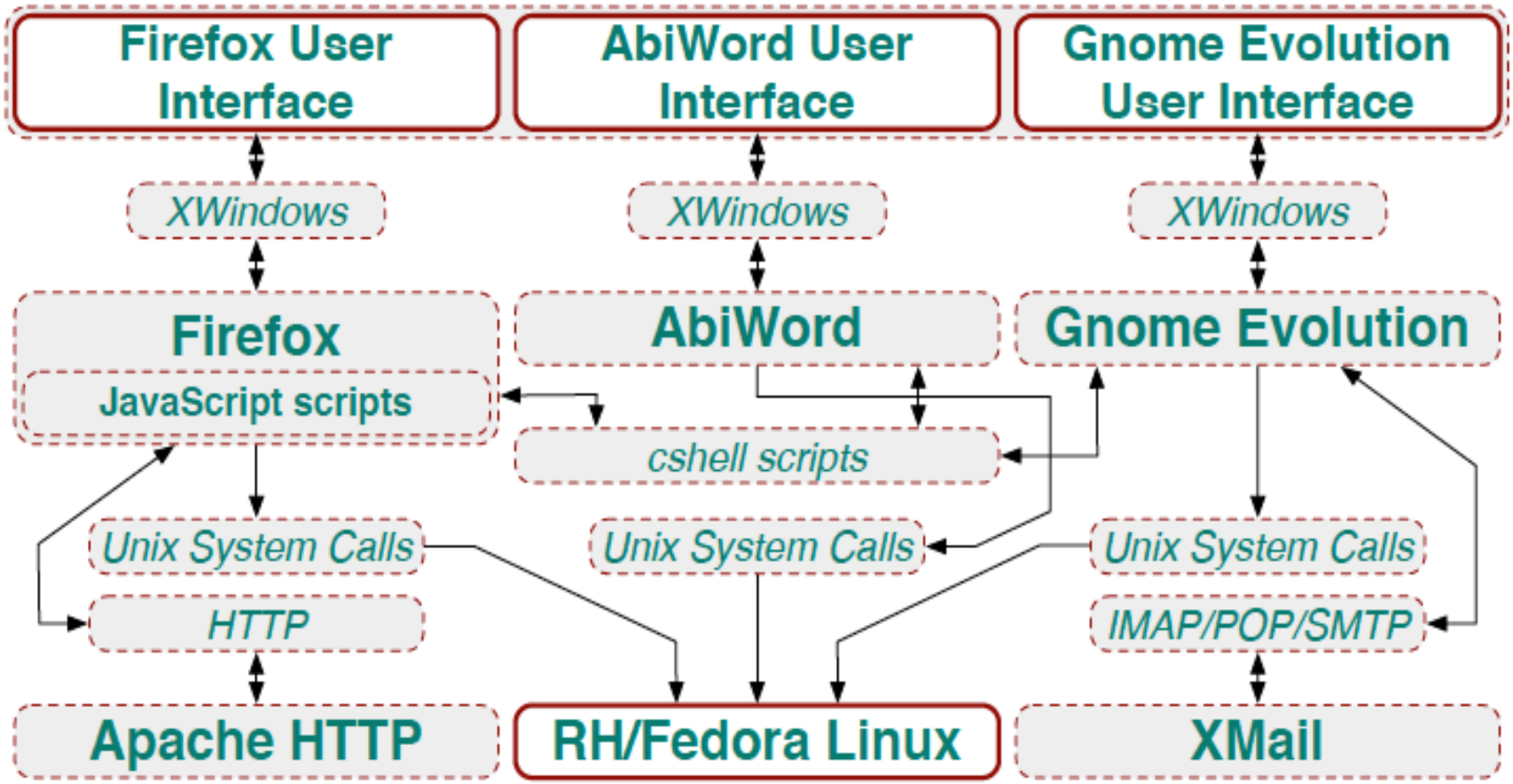
# A composed multi-component OA system



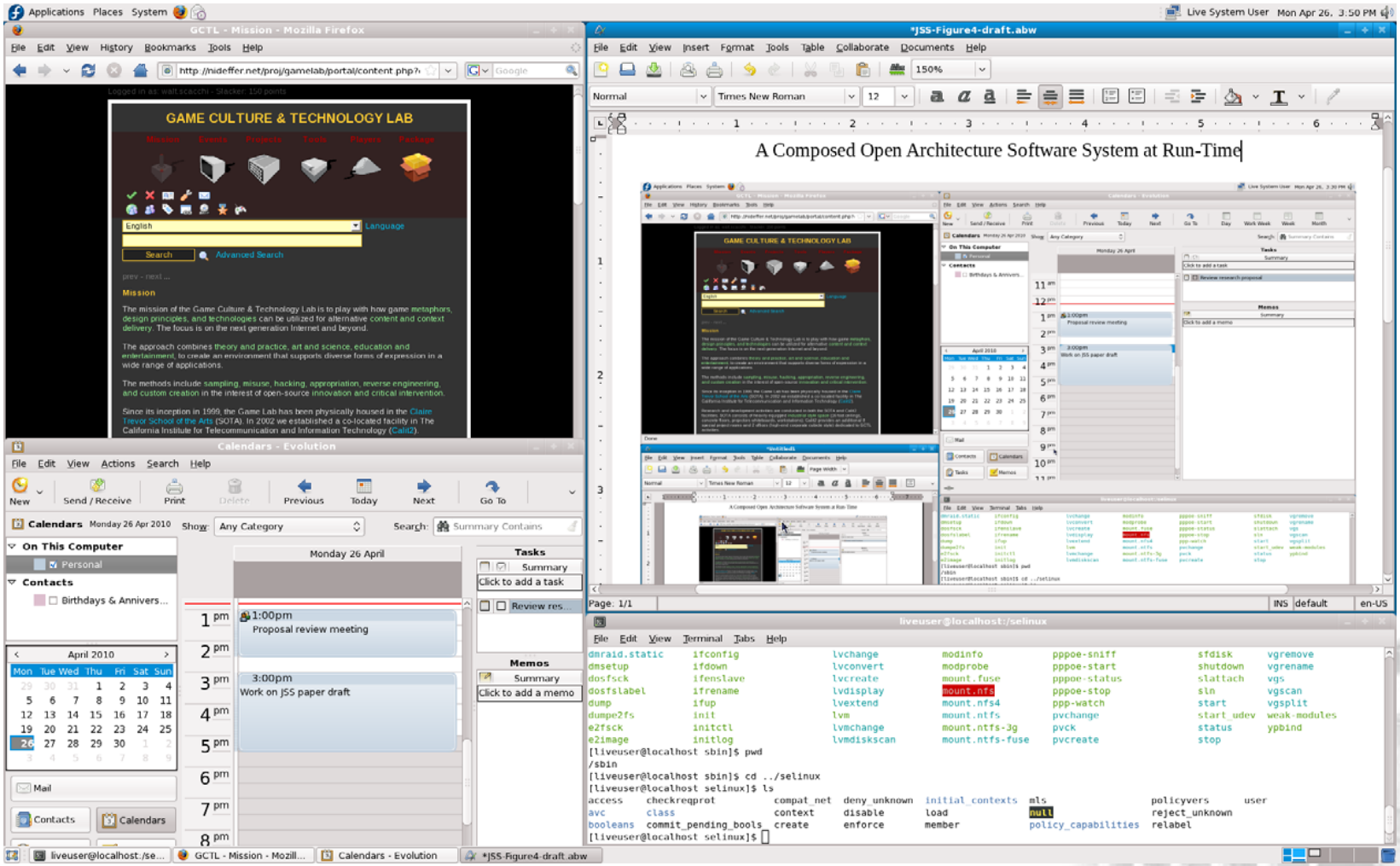
# Design-time architecture



# Build-time architecture



# Run-time architecture (user view)

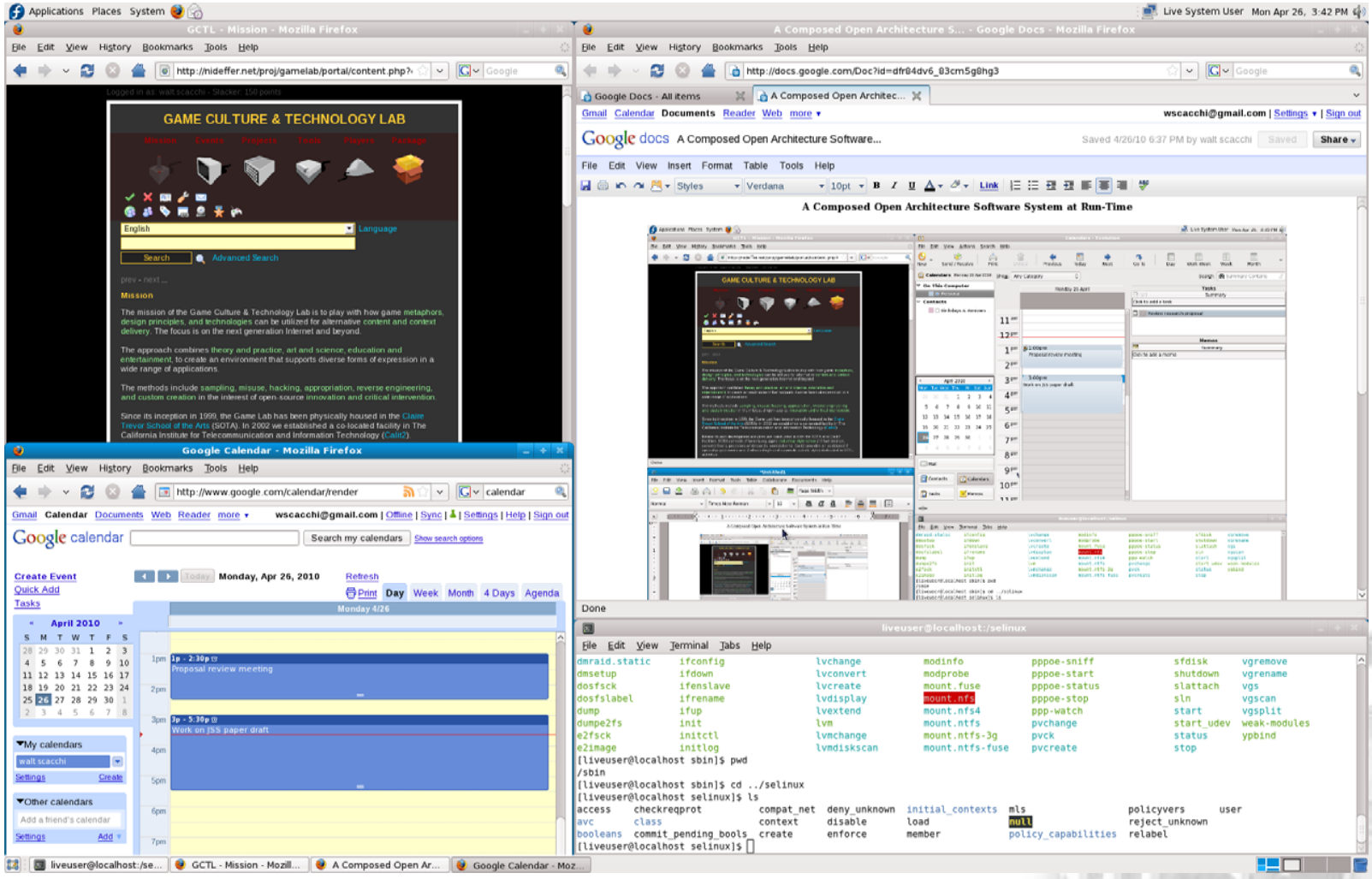


# Evolutionary changes in OA Systems

- Component evolution
- Component replacement
- Architecture evolution
- Component license evolution
- Change in desired rights or acceptable obligations

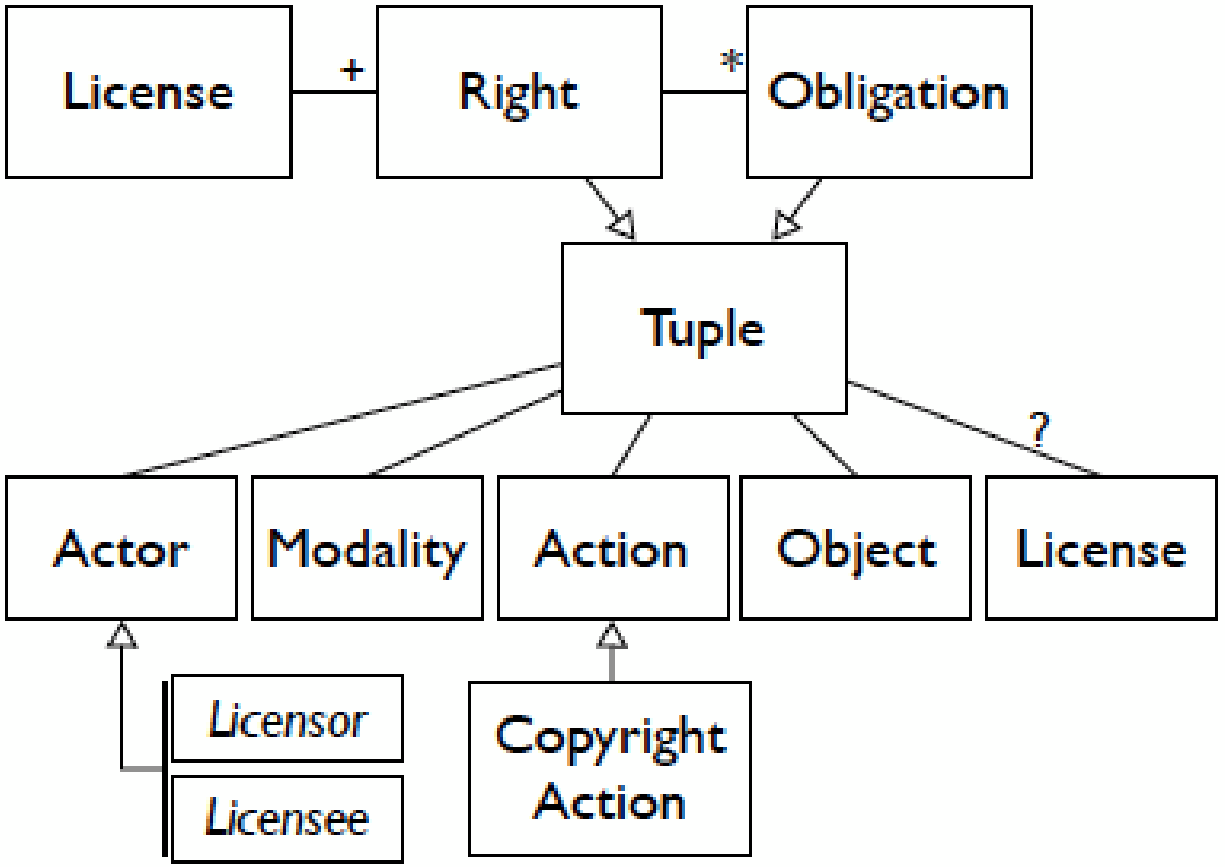
- Evolutionary changes reconfigure a system's software ecosystem!

# Component replacement and architecture evolution



# Software licenses, architectures, and analysis

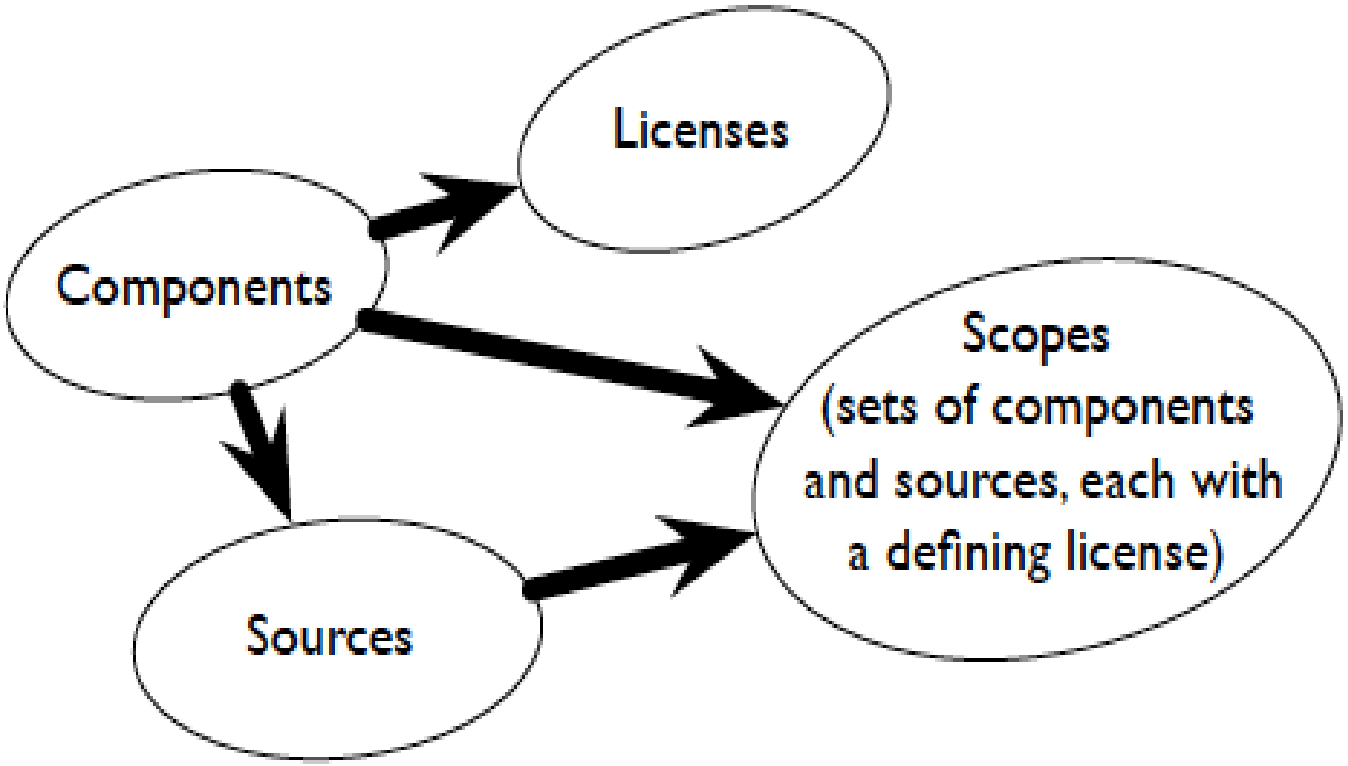
# Software license meta-model



# Logical modality and objects of software license rights and obligations

<table>
<tr><th></th><th>Modality</th><th>Object</th><th>License<br>(optional)</th></tr>
<tr><td rowspan="3">Abstract Right</td><td rowspan="3">May<br>or<br>Need Not</td><td>Any Under This License</td><td rowspan="3">This License<br>or<br>Object's License</td></tr>
<tr><td>Any Source Under This License</td></tr>
<tr><td>Any Component Under This License</td></tr>
<tr><td>Concrete Right</td><td rowspan="2">Concrete Object</td><td rowspan="2">Concrete License</td></tr>
<tr><td>Concrete Obligation</td><td rowspan="4">Must<br>or<br>Must Not</td></tr>
<tr><td rowspan="3">Abstract Obligation</td><td>Right's Object</td><td rowspan="3">Concrete License<br>or<br>Right's License</td></tr>
<tr><td>All Sources Of Right's Object</td></tr>
<tr><td>X Scope Sources</td></tr>
<tr><td></td><td></td><td>X Scope Components</td><td></td></tr>
</table>

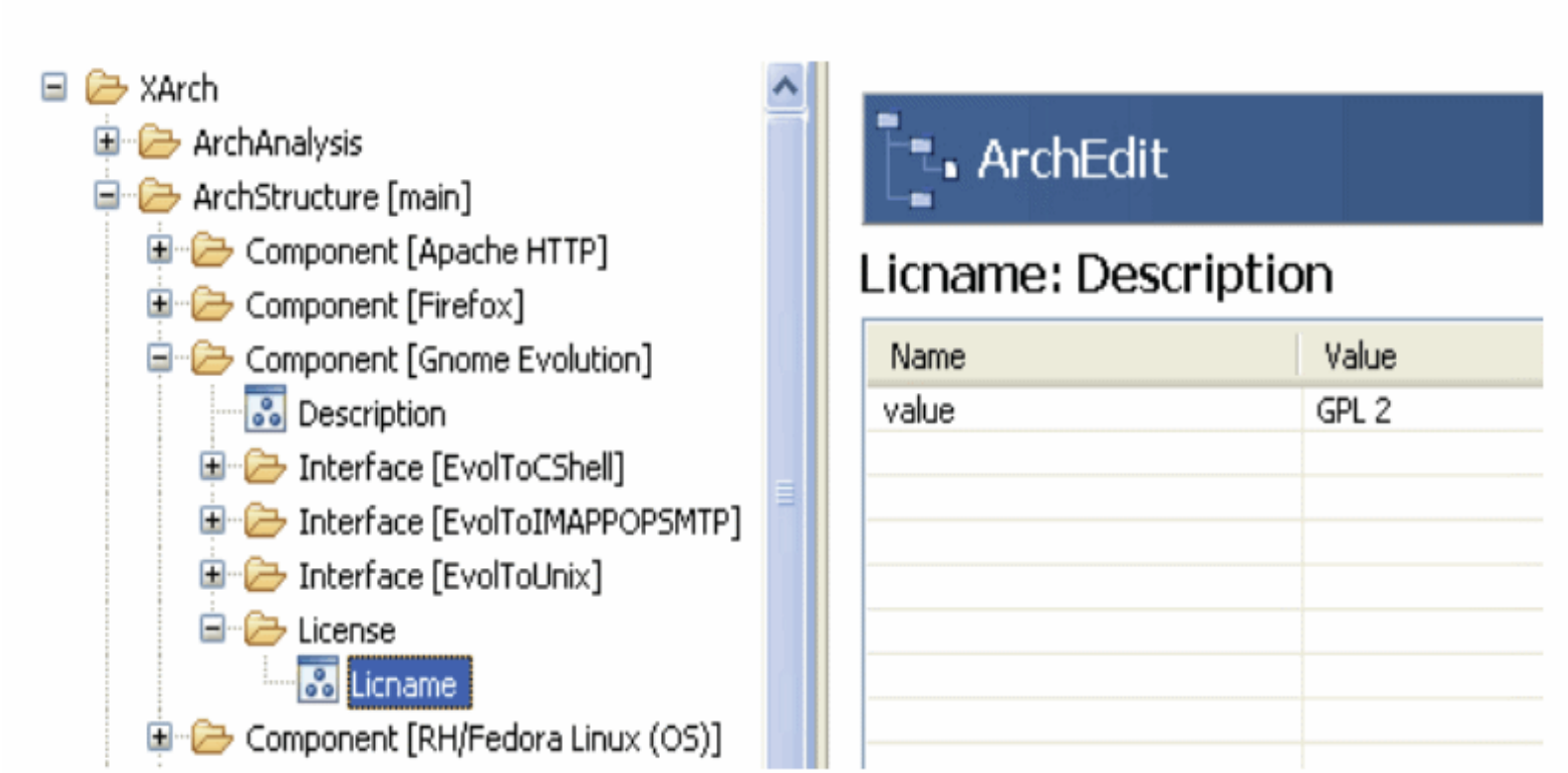
# The software license architecture meta-model



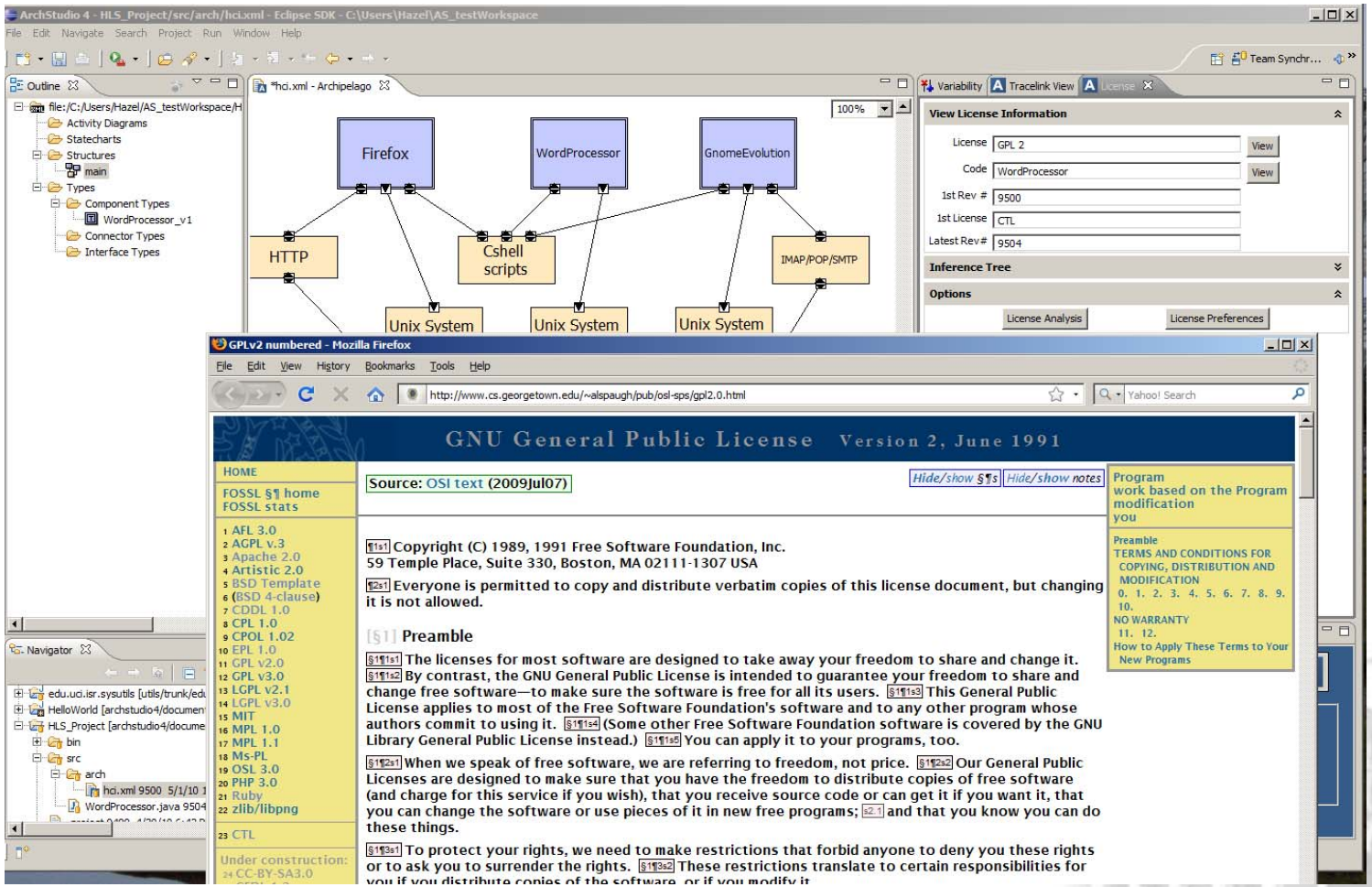
# Software license analysis

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

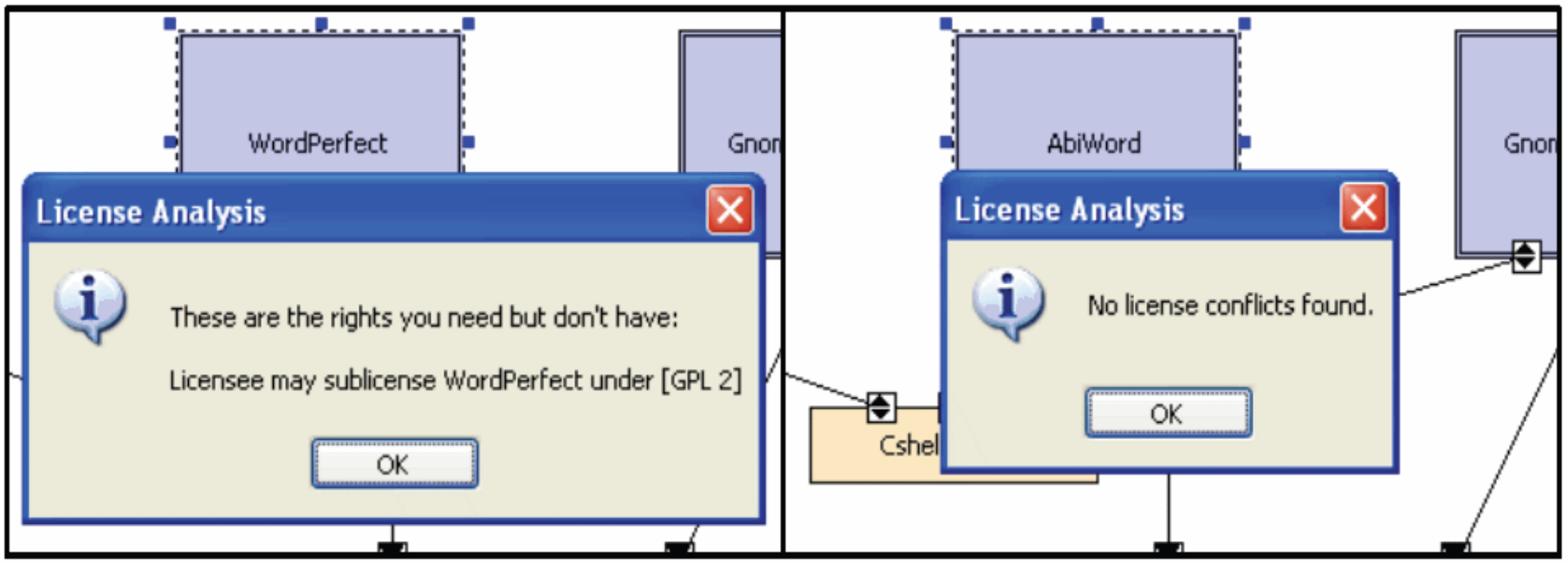
# Component license annotation prior to analysis



# License review during license analysis



# 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.