



# Defense Acquisition in Transition

6<sup>TH</sup> ANNUAL ACQUISITION RESEARCH SYMPOSIUM

## Software Licenses, Open Source Components, and Open Architectures

Thomas Alspaugh, Hazeline Asuncion, **Walt Scacchi**

Institute for Software Research

University of California, Irvine

# What is an Open Architecture?

- Open Architecture (OA) systems may include components with open APIs, Open Source Software (OSS) technology or development processes.
- OSS components are subject to different *Intellectual Property* (IP) licenses that impose constraints/conflicts
- Air Force, Army, and Navy each have their own reasons for adoption OA systems.
  - But what happens when there are conflicts across the Defense community regarding what an OA is?
- Therefore, is it clear what an OA is, or how to achieve an OA system with OSS?



# Research Questions

- What license applies to an OA system composed with OSS elements with different licenses?
- How do alternative OSS licenses facilitate or inhibit the development of OA systems?
- How should software license constraints be specified so it is possible to *automatically determine* the overall set of rights and obligations associated with a configured software system architecture?

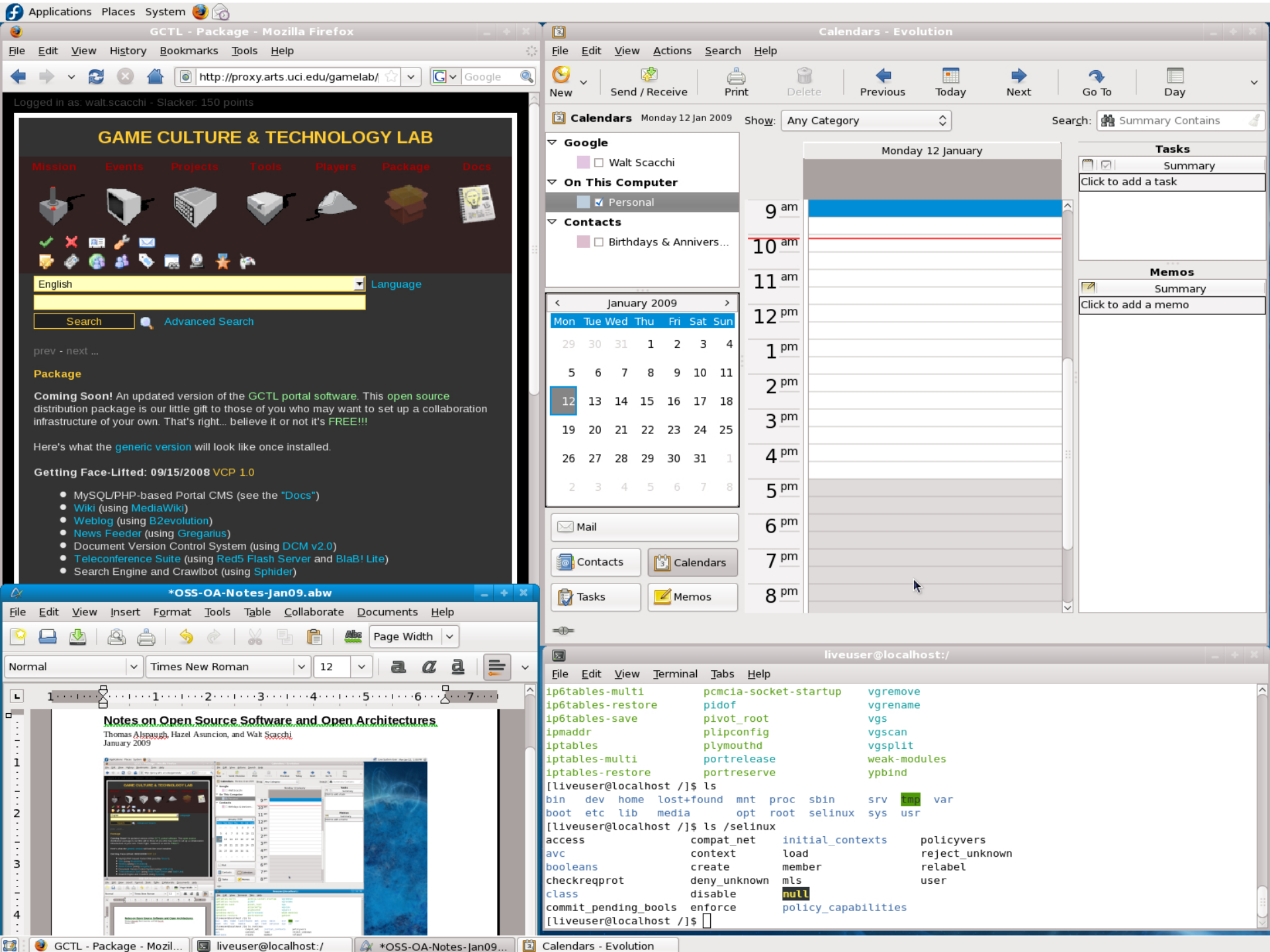


# OA, OSS, and software license analysis

- *Goal*: identify software architecture principles and OSS licenses that mediate OA
- OSS components subject to different IP licenses
- DoD policies and initiatives encouraging OA with OSS elements
- How to determine the *requirements* for how 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.



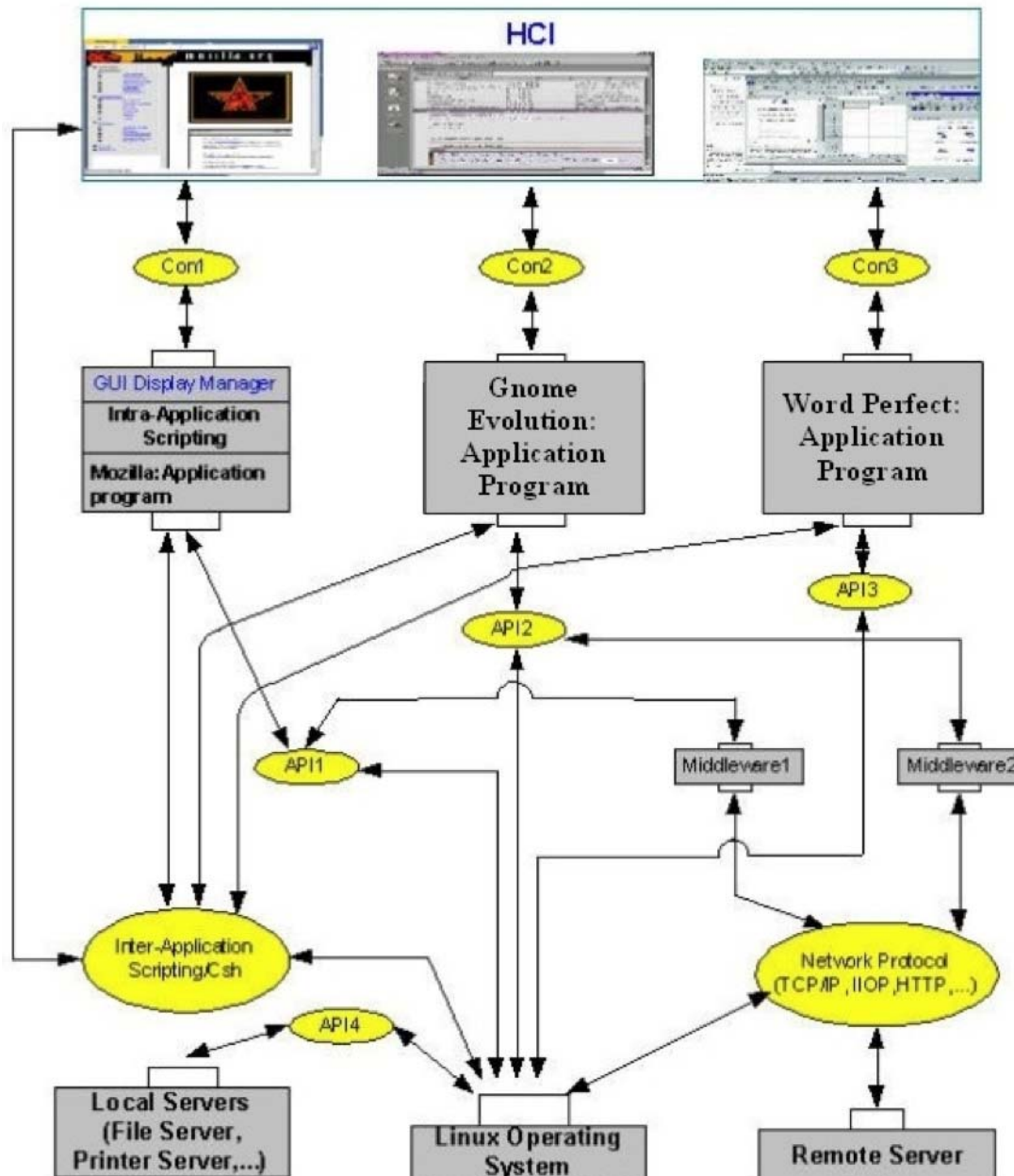




# Open Software Architecture Elements

- Software source code components
  - Standalone programs
  - Libraries, frameworks, or middleware
  - Inter-application script code (e.g., for building subsystems)
  - 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*;  
yellow ellipses are *connectors*;  
white boxes are *code interfaces*;  
arrows are data or control *flow paths*;  
complete figure is architectural design *configuration*,  
denote different kinds of elements.



# OSS elements subject to different IP licenses

- Intellectual Property licenses stipulate *rights* (“can/not do”) and *obligations* (“must/not do”) regarding IP usage
- GPL (Gnu Public License) stipulates right to access, study, modify, and *reciprocal* obligation to redistribute modified source code
- Mozilla now offers a *tri-license* for its software like Firefox Web Browser:
  - GPL, MPL (lightweight), or Restricted (accommodating proprietary services)
- Other OSS covered by different license rights and obligations, thus unclear how to check or enforce!





# OSS elements subject to different IP licenses

- How to determine which rights and obligations will apply or conflict within a configured system?



# OSS elements subject to different IP licenses

- How to determine which rights and obligations will apply or conflict within a configured system?
  - At *design-time* (maximum flexibility—can employ “license firewalls” to mitigate license constraints)
  - At *build-time* (may/not be able to redistribute components at hand)
  - At *run-time* (software release may/not need to install/link-to components from other sites or repositories)



# OSS elements subject to different IP licenses

- Different license constraints may apply or conflict at different times
- Different license constraints imply overall system may/not be OA at different times
- We need to know at all times what license constraints apply, and whether/not we have an OA
- License analysis with large, complex systems may be intractable for developers, lawyers, PEOs, etc.
  - Further exacerbated by different time constraints



# OSS elements subject to different IP licenses

## Practical requirements:

- Must formally specify software license rights and obligations
- Must specify and model software system architectures
- Must analyze software architectures in terms of license obligations and rights for multi-component systems or sub-systems
  - At architectural design-time
  - At build-time
  - At run-time (including integration with legacy systems)
  - Analysis same at each time, but results may differ!
- Should *provide automated support* to help meet these needs

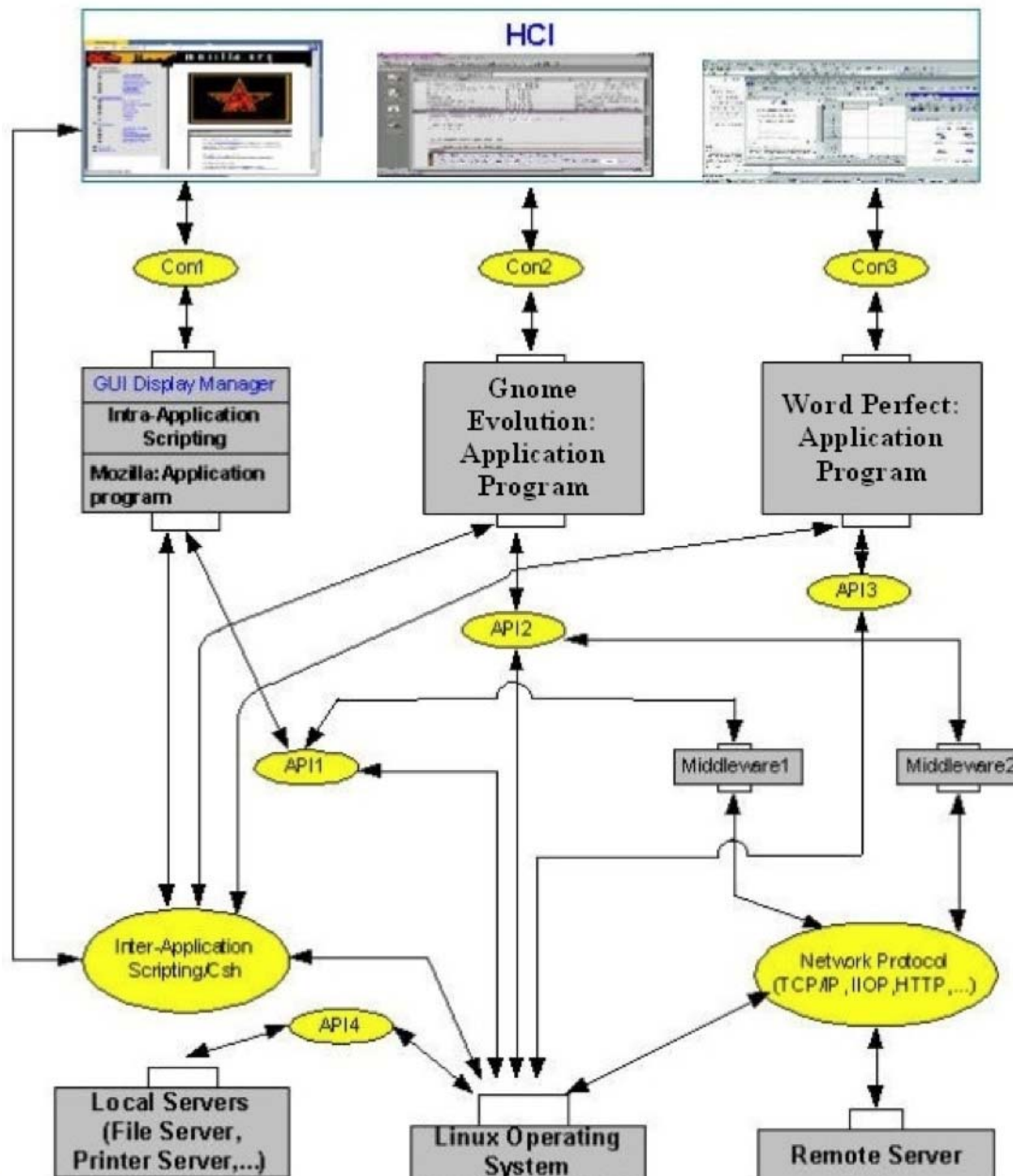


# ***Proposed Solution:*** **Automated OA Design and License Analysis Environment**

- Developed an operational prototype OA design and license environment
- Demonstrates the ability to satisfy the all of the practical requirements
- Developed as an extension to the *ArchStudio* Architecture Design Environment from UC Irvine
  - <http://www.isr.uci.edu/projects/archstudio/>







## Legend:

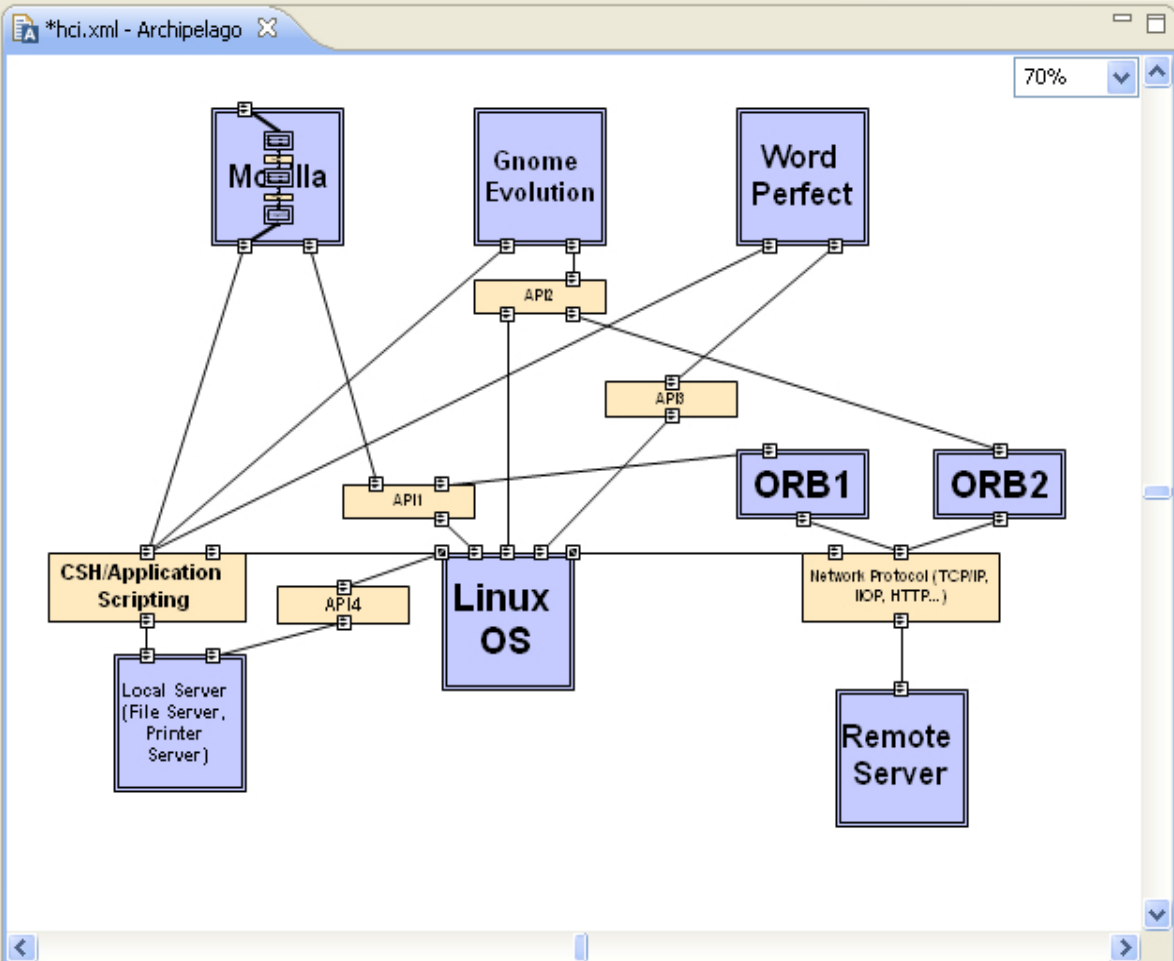
Grey boxes are *components*;  
yellow ellipses are *connectors*;  
white boxes are *code interfaces*;  
arrows are data or control *flow paths*;  
complete figure is architectural design *configuration*,  
denote different kinds of elements.





**Outline**

- /architectures/hci.xml
  - Activity Diagrams
  - Statecharts
  - Structures
    - main
    - MoZilla SubArch
  - Types
    - Component Types
      - MoZillaComp
    - Connector Types
    - Interface Types



**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.com
- licence





HTML-Kit

File Edit View Tools Tags Actions Workspace Window Help

hci.xml

```
2143 <licenselookup:licenseType xsi:type="licenselookup:LicenseType">
2144   <licenselookup:name xsi:type="instance:Description">MPL</licenselookup:name>
2145   <licenselookup:reference xlink:href="http://www.mozilla.org/MPL/MPL-1.1.html" xlink:type="simple" xsi:type="
2146   <licenselookup:obligation licenselookup:id="MPL2.id" xsi:type="licenselookup:Obligation">
2147     <licenselookup:actor xsi:type="licenselookup:Actor">licensee</licenselookup:actor>
2148     <licenselookup:operation xsi:type="licenselookup:Operation">must not</licenselookup:operation>
2149     <licenselookup:action xsi:type="licenselookup:Action">delete</licenselookup:action>
2150     <licenselookup:object xsi:type="licenselookup:Object">from original code</licenselookup:object>
2151   </licenselookup:obligation>
2152   <licenselookup:obligation licenselookup:id="MPL3.1" xsi:type="licenselookup:Obligation">
2153     <licenselookup:actor xsi:type="licenselookup:Actor">licensee</licenselookup:actor>
2154     <licenselookup:operation xsi:type="licenselookup:Operation">must</licenselookup:operation>
2155     <licenselookup:action xsi:type="licenselookup:Action">retain</licenselookup:action>
2156     <licenselookup:object xsi:type="licenselookup:Object">copyright notice</licenselookup:object>
2157   </licenselookup:obligation>
2158   <licenselookup:obligation licenselookup:id="MPL3.2" xsi:type="licenselookup:Obligation">
2159     <licenselookup:actor xsi:type="licenselookup:Actor">licensee</licenselookup:actor>
2160     <licenselookup:operation xsi:type="licenselookup:Operation">must</licenselookup:operation>
2161     <licenselookup:action xsi:type="licenselookup:Action">redistribute</licenselookup:action>
2162     <licenselookup:object xsi:type="licenselookup:Object">source code</licenselookup:object>
2163   </licenselookup:obligation>
2164   <licenselookup:right licenselookup:id="MPL3.6" xsi:type="licenselookup:Right">
2165     <licenselookup:satisfy xsi:type="licenselookup:Satisfy">
2166       <licenselookup:obligationID xlink:href="#MPL3.1" xlink:type="simple" xsi:type="instance:XMLLink"/>
2167       <licenselookup:obligationID xlink:href="#MPL3.2" xlink:type="simple" xsi:type="instance:XMLLink"/>
2168     </licenselookup:satisfy>
2169     <licenselookup:actor xsi:type="licenselookup:Actor">licensee</licenselookup:actor>
2170     <licenselookup:operation xsi:type="licenselookup:Operation">may</licenselookup:operation>
2171     <licenselookup:action xsi:type="licenselookup:Action">distribute</licenselookup:action>
2172     <licenselookup:object xsi:type="licenselookup:Object">Covered Code in executable form </licenselookup:ob
2173   </licenselookup:right>
  <licenselookup:right licenselookup:id="MPL2.1" xsi:type="licenselookup:Right">
```

Editor Preview Output Split View

hci.xml

2173: 15 INS

```
ArchStudio 4 [Eclipse Application] C:\Program Files\Java\jre1.6.0_05\bin\javaw.exe (Aug 28, 2008 8:16:49 PM)

License Report for the Main Architecture
***Printing all obligations
obligation: MPL2.1d licensee must not delete from original code
obligation: MPL3.1 licensee must retain copyright notice
obligation: MPL3.2 licensee must redistribute source code
obligation: CTL2 licensee must not redistribute source code
obligation: GPL2 licensee must redistribute source code

***Printing all conflicting obligations
obligation: CTL2 licensee must not redistribute source code
obligation: MPL3.2 licensee must redistribute source code
obligation: GPL2 licensee must redistribute source code
obligation: CTL2 licensee must not redistribute source code

***Printing all rights
right: MPL3.6 licensee may distribute Covered Code in executable form
right: MPL2.1 licensee may reproduce original code
right: MPL1 licensee may redistribute executable
right: CTL3 licensee may not reproduce original code
right: CTL3& licensee may not use Licensors name, logo, or trademarks
right: CTL4 licensee may not redistribute executable
right: GPL1 licensee may redistribute executable

***Printing all intersecting rights

License Report for SubArchitecture: Mozilla SubArch
***Printing all obligations
obligation: MPL2.1d licensee must not delete from original code
```



# Discussion

What about *other* OA with OSS problems?

- How to determine what license constraints apply to a delivered software system that may have OSS within it?
- Where is the OSS and is it GPL or not?
  - Requires source code analysis/data mining tools
  - No external architectural design in hand
- Possible to semi-automatically generate build-time architecture specification from source code
- Automated OA license analysis environment, *plus* source code analysis/mining, and architecture (re)generation tools, is best bet for addressing these problems



# Conclusions

- Identified a fundamental challenge to design, build, and release of OA software systems that include OSS components with different IP licenses
- Identified approach for solving the challenge
- Demonstrated a prototype automated environment that solves problem at hand
- Identified other problems that may affect full realization of OA with OSS components



# Acknowledgements

- Research supported by:
  - Acquisition Research Program at the Naval Postgraduate School
  - Grants #0534771 and #0808783 from the National Science Foundation (NSF)
- No endorsement implied.

