

Challenges in the Development and Evolution of Secure Open Architecture Command and Control Systems

Walt Scacchi and Thomas Alspaugh
Institute for Software Research
University of California, Irvine
Irvine, CA 92697-3455 USA

Report Documentation Page				Form Approved OMB No. 0704-0188	
Public reporting burden for the collection of information is estimated to average 1 hour per response, including the time for reviewing instructions, searching existing data sources, gathering and maintaining the data needed, and completing and reviewing the collection of information. Send comments regarding this burden estimate or any other aspect of this collection of information, including suggestions for reducing this burden, to Washington Headquarters Services, Directorate for Information Operations and Reports, 1215 Jefferson Davis Highway, Suite 1204, Arlington VA 22202-4302. Respondents should be aware that notwithstanding any other provision of law, no person shall be subject to a penalty for failing to comply with a collection of information if it does not display a currently valid OMB control number.					
1. REPORT DATE JUN 2013		2. REPORT TYPE		3. DATES COVERED 00-00-2013 to 00-00-2013	
4. TITLE AND SUBTITLE Challenges in the Development and Evolution of Secure Open Architecture Command and Control Systems (BRIEFING CHARTS)				5a. CONTRACT NUMBER	
				5b. GRANT NUMBER	
				5c. PROGRAM ELEMENT NUMBER	
6. AUTHOR(S)				5d. PROJECT NUMBER	
				5e. TASK NUMBER	
				5f. WORK UNIT NUMBER	
7. PERFORMING ORGANIZATION NAME(S) AND ADDRESS(ES) University of California, Irvine, Institute for Software Research, Irvine, CA, 92697-3455				8. PERFORMING ORGANIZATION REPORT NUMBER	
9. SPONSORING/MONITORING AGENCY NAME(S) AND ADDRESS(ES)				10. SPONSOR/MONITOR'S ACRONYM(S)	
				11. SPONSOR/MONITOR'S REPORT NUMBER(S)	
12. DISTRIBUTION/AVAILABILITY STATEMENT Approved for public release; distribution unlimited					
13. SUPPLEMENTARY NOTES Presented at the 18th International Command & Control Research & Technology Symposium (ICCRTS) held 19-21 June, 2013 in Alexandria, VA. U.S. Government or Federal Rights License					
14. ABSTRACT					
15. SUBJECT TERMS					
16. SECURITY CLASSIFICATION OF:			17. LIMITATION OF ABSTRACT Same as Report (SAR)	18. NUMBER OF PAGES 25	19a. NAME OF RESPONSIBLE PERSON
a. REPORT unclassified	b. ABSTRACT unclassified	c. THIS PAGE unclassified			

Overview

- Challenges of securing open architecture (OA) systems
- Specifying security requirements for software systems
- *Case study*: Securing the development and evolution of an OA C2 system within an agile, adaptive software ecosystem
- Discussion and conclusions

Challenges of securing open architecture (OA) C2 systems

Scacchi, W., Brown, C. and Nies, K. (2012). Understanding the Potential of Virtual Worlds for Decentralized Command and Control, *Proc. 17th. Intern. Command and Control Research and Technology Symposium (ICCRTS)*, Paper-096, Fairfax, VA, June 2012.

Scacchi, W., Brown, C. and Nies, K. (2012). Understanding the Potential of Computer Games for Decentralized Command and Control, *Proc. 17th. Intern. Command and Control Research and Technology Symposium (ICCRTS)*, Paper-104, Fairfax, VA, June 2012.

Virtual world for experimental studies in decentralized command and control centers using open source software components



Security challenges

- Security threats to software systems are increasingly multi-modal and distributed across system components.
- Physically isolated systems are vulnerable to external security attacks.
- What makes an OA C2 system secure changes over time, as new threats emerge and systems evolve.
- Need an approach *to continuously assure the security of evolving OA C2 systems* that is practical, scalable, robust, tractable, and adaptable.

Current security approaches

- Mandatory access control lists, firewalls;
- Multi-level security;
- Authentication (including certificate authority and passwords);
- Cryptographic support (including public key certificates);
- Encapsulation (including virtualization), hardware confinement (memory, storage, and external device isolation), and type enforcement capabilities;
- Secure programming practices;
- Data content or control signal flow logging/auditing;
- Honey-pots, traps, sink-holes;
- Security technical information guides for configuring the security parameters for applications and operating systems;
- Functionally equivalent but diverse multi-variant software executables.

Software systems/components *evolve*: what to do about security?

- Individual components evolve via revisions (e.g., security patches)
- Individual components are updated with functionally enhanced versions;
- Individual components are replaced by alternative components;
- Component interfaces evolve;
- System architecture and configurations evolve;
- System functional and security requirements evolve;
- System security policies, mechanisms, security components, and system configuration parameter settings also change over time.

Specifying the security requirements for OA software systems

Carefully specifying security policy obligations and rights

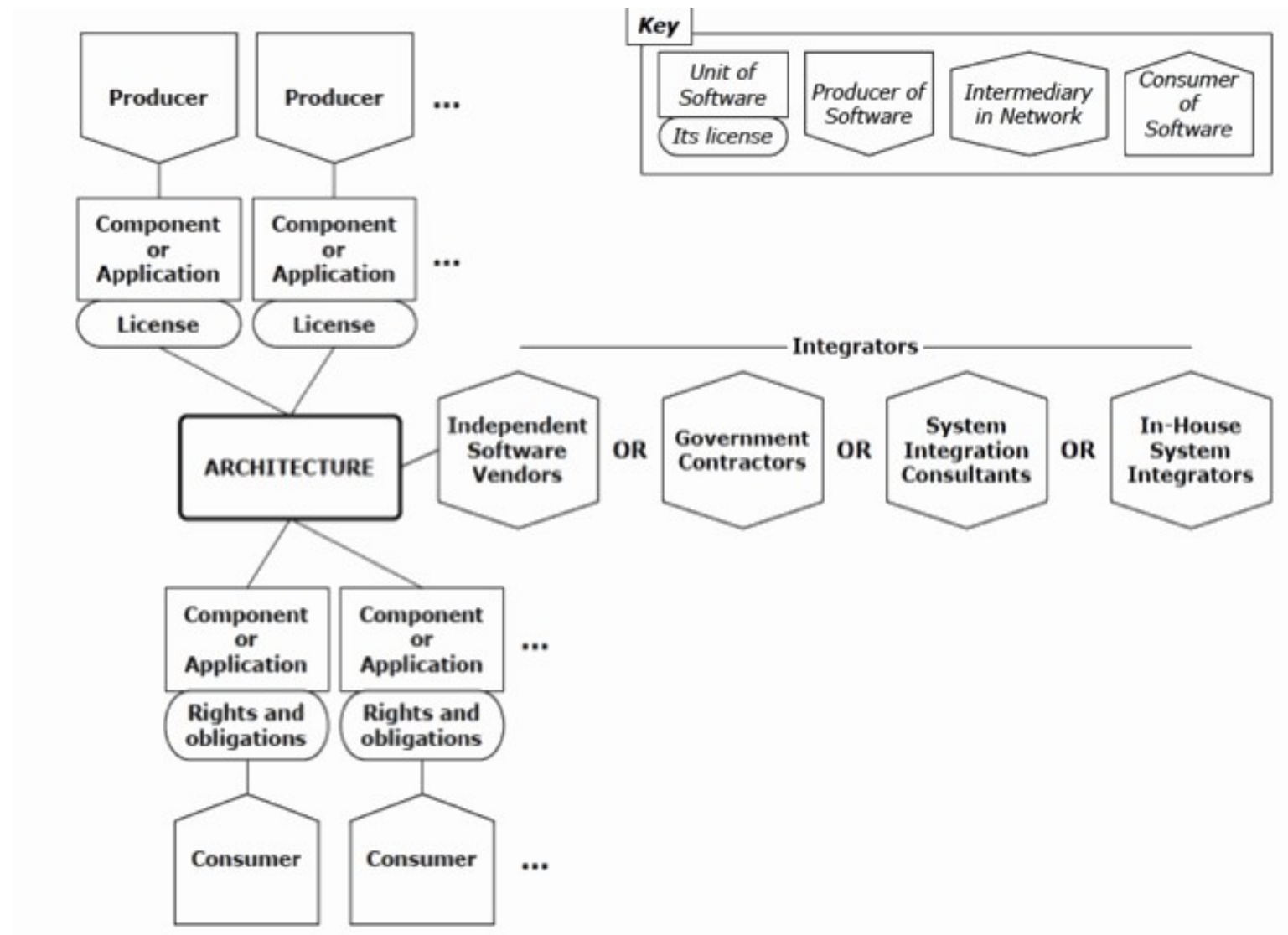
- The obligation for a user to verify his/her authority to see compartment T, by password or other specified authentication process
- The obligation for all components connected to specified component C to grant it the capability to read and update data in compartment T
- The obligation to reconfigure a system in response to detected threats, when given the right to select and include different component versions, or executable component variants.
- The right to read and update data in compartment T using the licensed component
- The right to add, update, replace specified component D in a specified configuration
- The right to add, update, or remove a security mechanism
- The right to update security policy L.

Case Study:
Securing the development and
evolution of an OA C2 system within
an agile, adaptive software
ecosystem

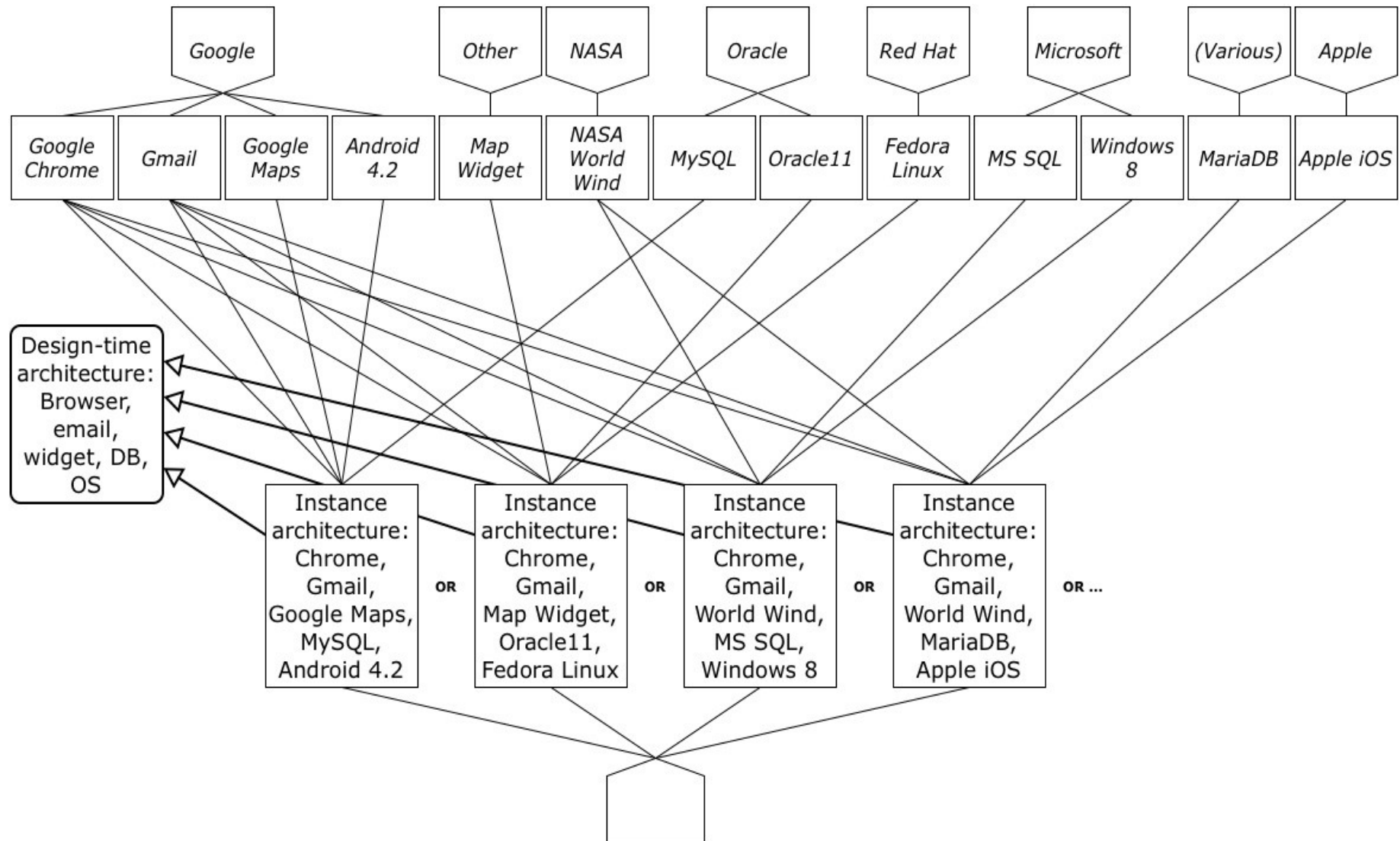
Software product lines?

- When functionally similar software components, connectors, or configurations exist,
- Such that equivalent alternatives, versions, or variants may be substituted for one another, then
- We have a strong relationship among these OA system elements that is called a *software product line*.
- Software product lines for OA systems enable support from agile, adaptive software (component) ecosystems
 - Reed, H., Benito, P., Collens, J. and Stein, F. (2012). Supporting Agile C2 with an Agile and Adaptive IT Ecosystem, *Proc. 17th Intern. Command and Control Research and Technology Symposium (ICCRTS)*, Paper-044, Fairfax, VA, June 2012.

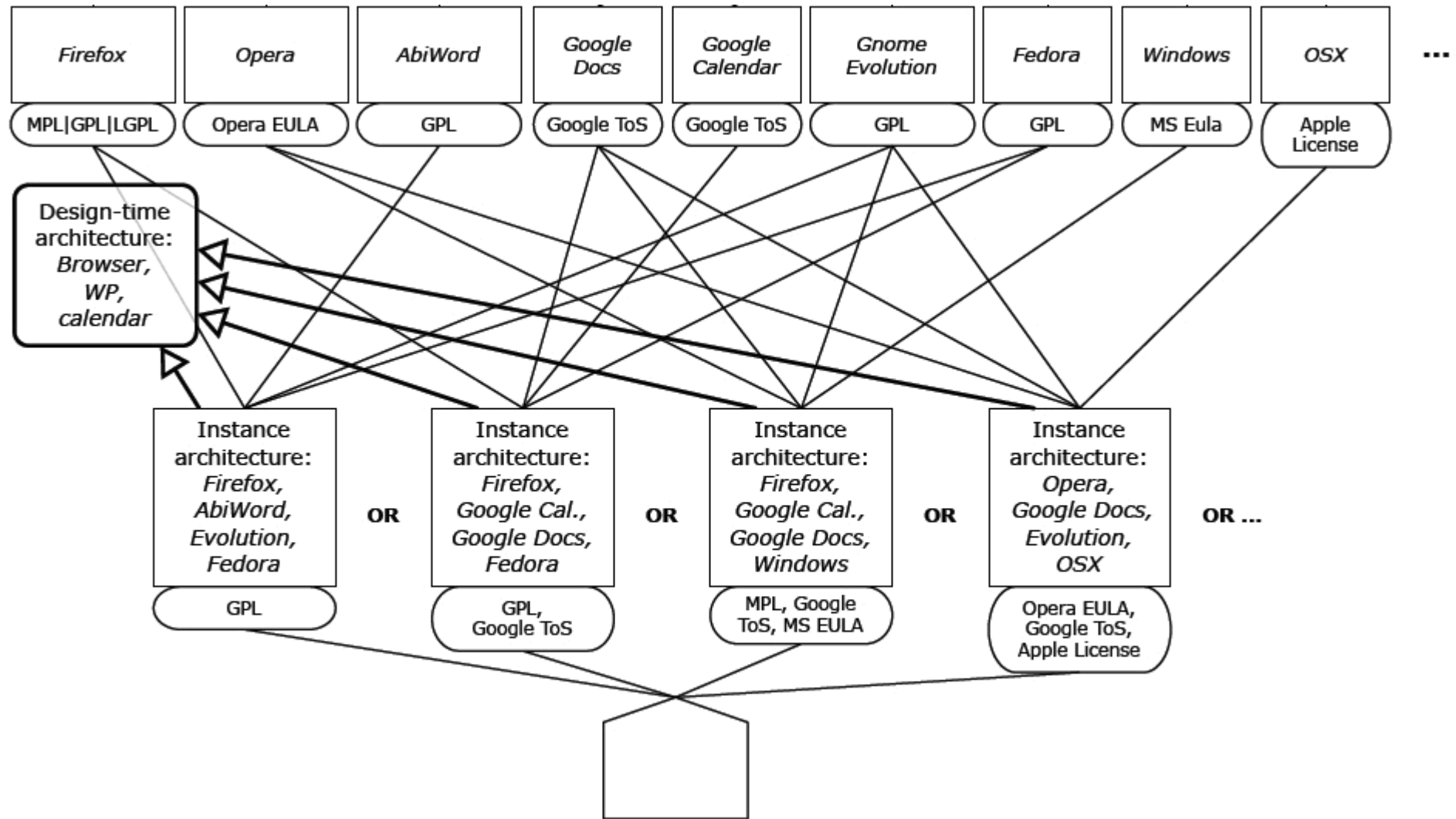
Software ecosystem of producers and the software components or application widgets for an enterprise system



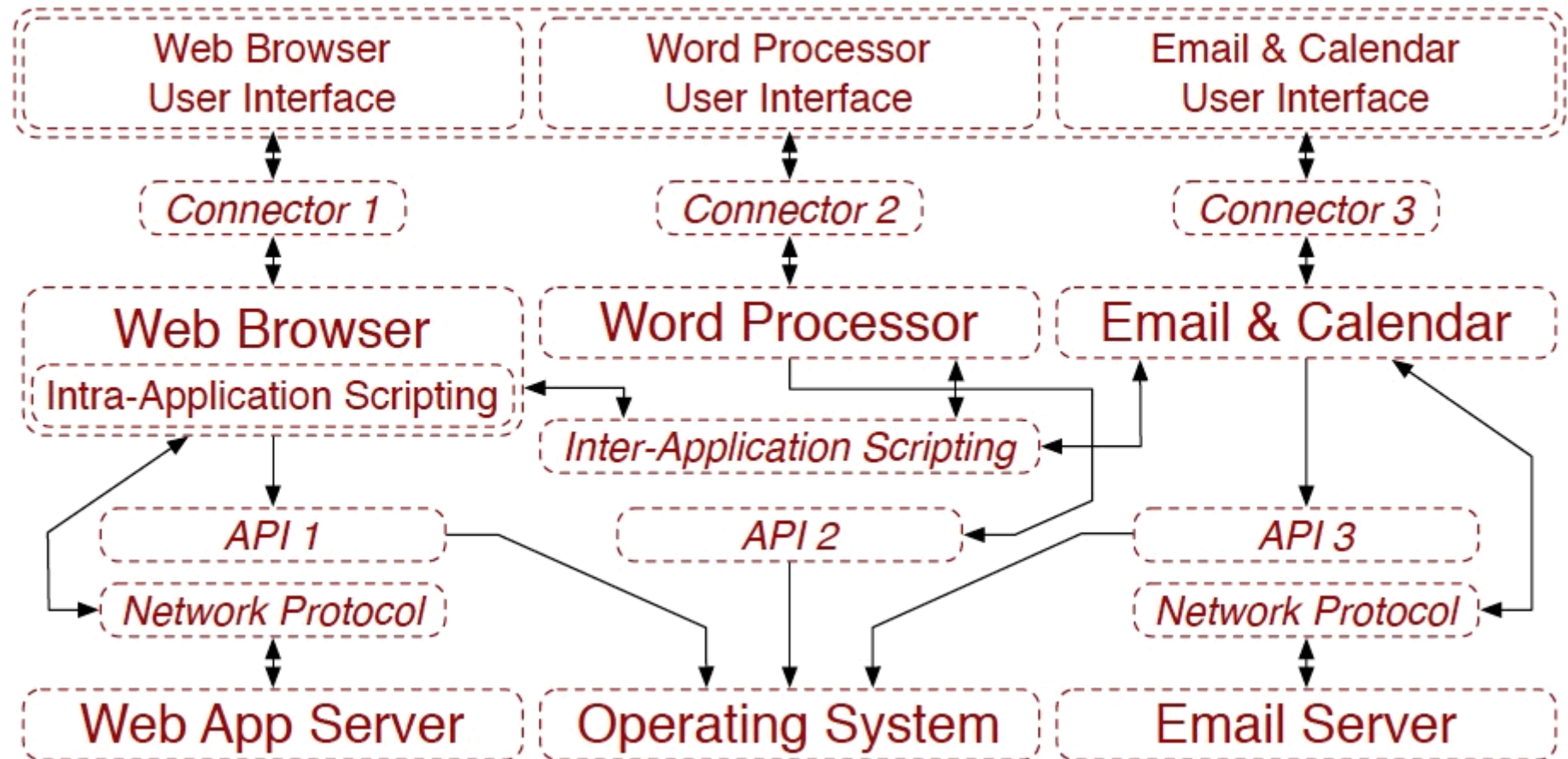
Software ecosystem of components or application widgets for an OA system



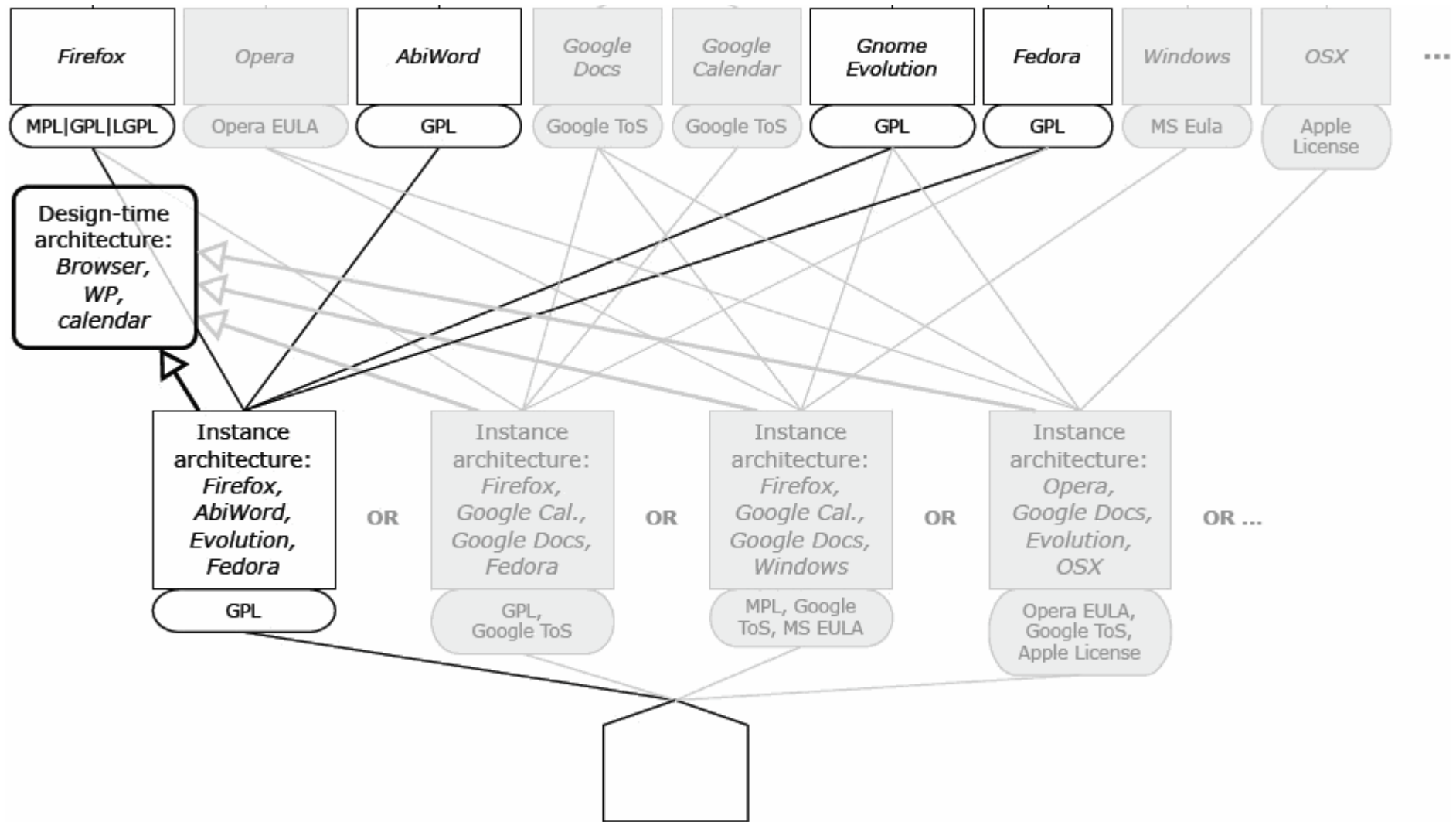
Software product line that provides functionally similar components or applications compatible with an OA system design



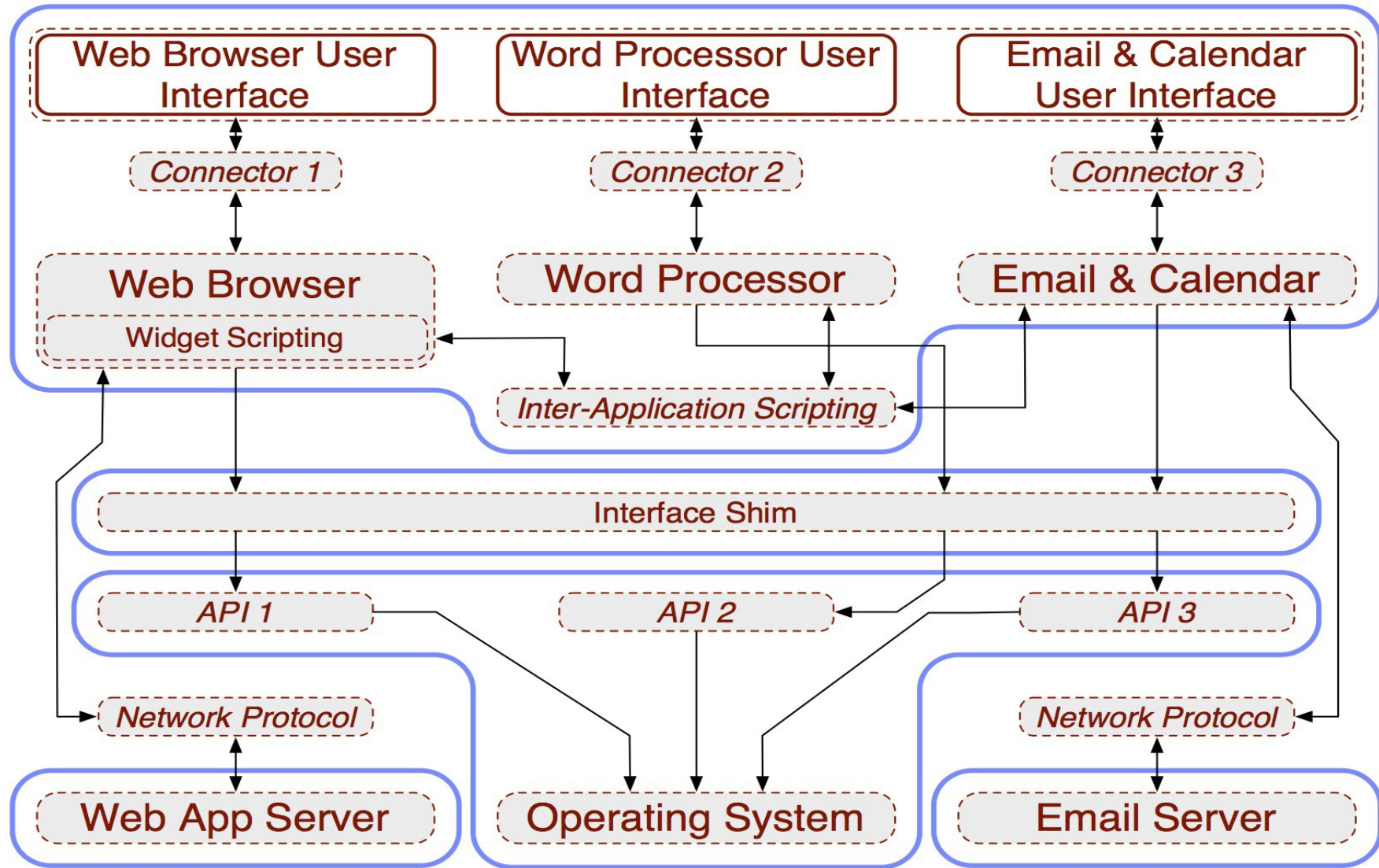
A *design-time* specification of an OA system that accommodates multiple alternative system configurations



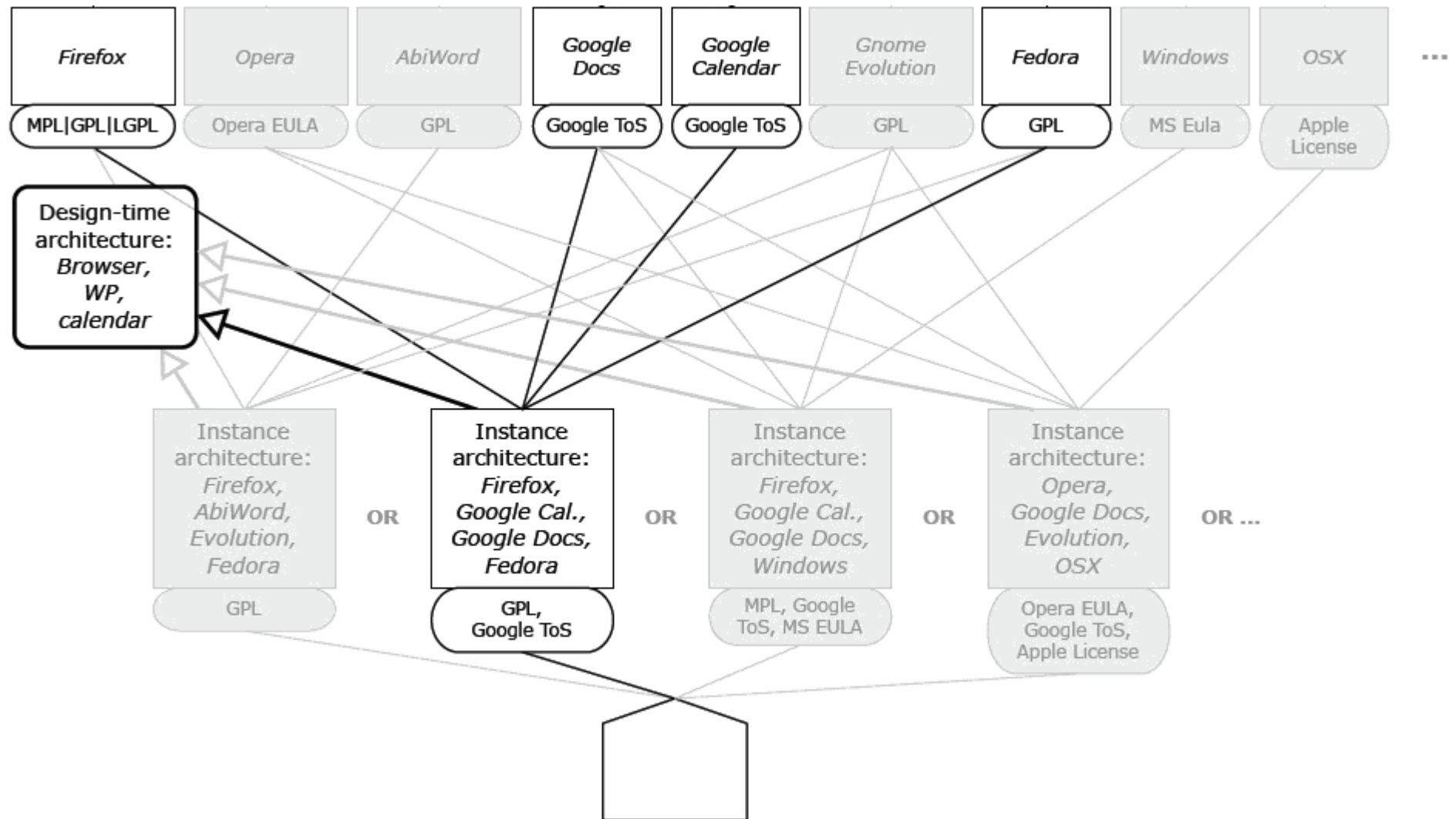
A build-time deployment selection among alternative components that produce an integrated enterprise system within the product line



A security capability specification encapsulating the *run-time deployment* configuration via multiple virtual machines



Adapting the *post-deployment system configuration*, using alternative but functionally similar components within the product line



An end-user view of the adapted alternative run-time system configuration

The image displays a collage of four screenshots from a desktop environment, illustrating an end-user view of the adapted alternative run-time system configuration.

The top-left screenshot shows a web browser (Mozilla Firefox) displaying the "GAME CULTURE & TECHNOLOGY LAB" website. The page features a navigation menu, a search bar, and a "Mission" section describing the lab's focus on game metaphors, design principles, and technologies for alternative content and context delivery.

The top-right screenshot shows a Google Docs document titled "A Composed Open Architecture Software System at Run-Time". The document is being edited by "wscacchi@gmail.com" and is saved. The interface includes a toolbar with various editing tools and a sidebar showing the document's structure.

The bottom-left screenshot shows a Google Calendar interface for Monday, April 26, 2010. The calendar displays a schedule with events, including a "Proposal review meeting" at 1pm and "Work on ISS paper draft" at 3pm.

The bottom-right screenshot shows a terminal window titled "liveuser@localhost:/sbin". The terminal displays the output of the "ls" command, listing various system binaries and SELinux contexts. The output is organized into columns, showing the file name, its SELinux context, and its permissions.

Discussion and conclusions

Discussion

- Our goal is to demonstrate a new approach to address challenges in the development and evolution of secure component-based OA C2 software systems.
- Future C2 systems require review and approval of security measures employed during the *design, implementation, deployment, and evolution* of OA systems.
- We seek to make this a simpler, more transparent, and more tractable process.

Conclusions (1)

- Our research demonstrates how complex OA systems can be designed, built, deployed, and evolved with alternative components within functionally similar system versions, to realize for overall system security.
- We described a scheme to specify and realize OA system configurations that are compatible with existing security mechanisms.
 - Our scheme does not assume that individual system elements must be secure before inclusion into the secured OA system's configuration.
- Central to our OA scheme is agile, adaptive software ecosystems and product lines integrated with security mechanisms.

Conclusions (2)

Next steps:

- Articulate the *process* how to simply and transparently specify and assess the security of OA C2 systems using streamlined security policy mechanisms.
- Develop and demonstrate a prototype *automated environment* that can support the modeling and analysis of OA system security policies and alternative version OA system configurations, in ways that address the diverse needs of software producers, system integrators and end-users.

Acknowledgements

Research described in this presentation was supported by grant #N00244-12-1-0067 from the Acquisition Research Program at the Naval Postgraduate School, and from grant #1256593 from the National Science Foundation.

No review, approval, or endorsement implied.