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

ISR Institute for Software Research
University of California, Irvine

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

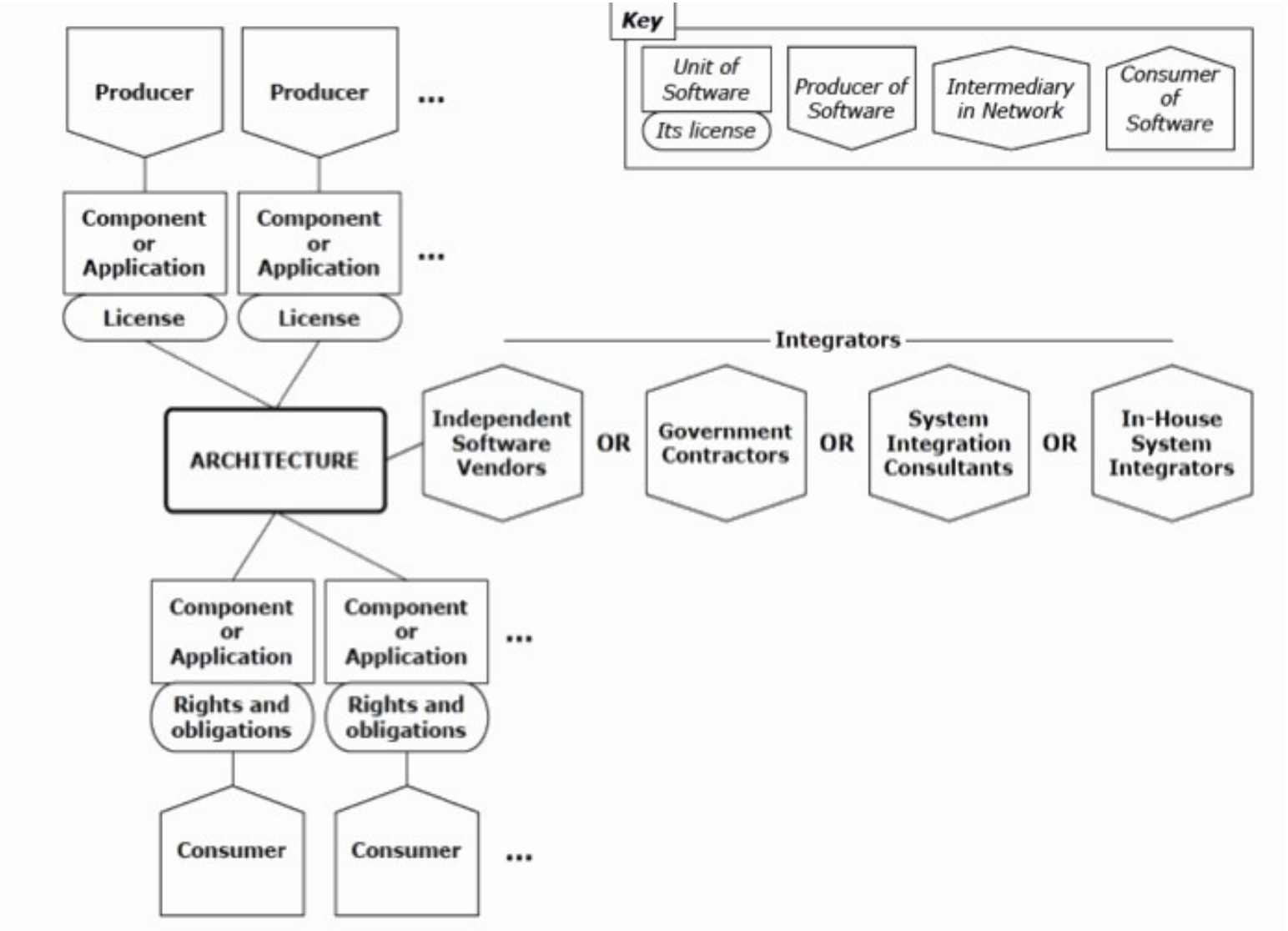
- The obligation for a user to verify his/her authority to see compartment T, by password or other specified authentication process
- The obligation for all components connected to specified component C to grant it the capability to read and update data in compartment T
- The obligation to reconfigure 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

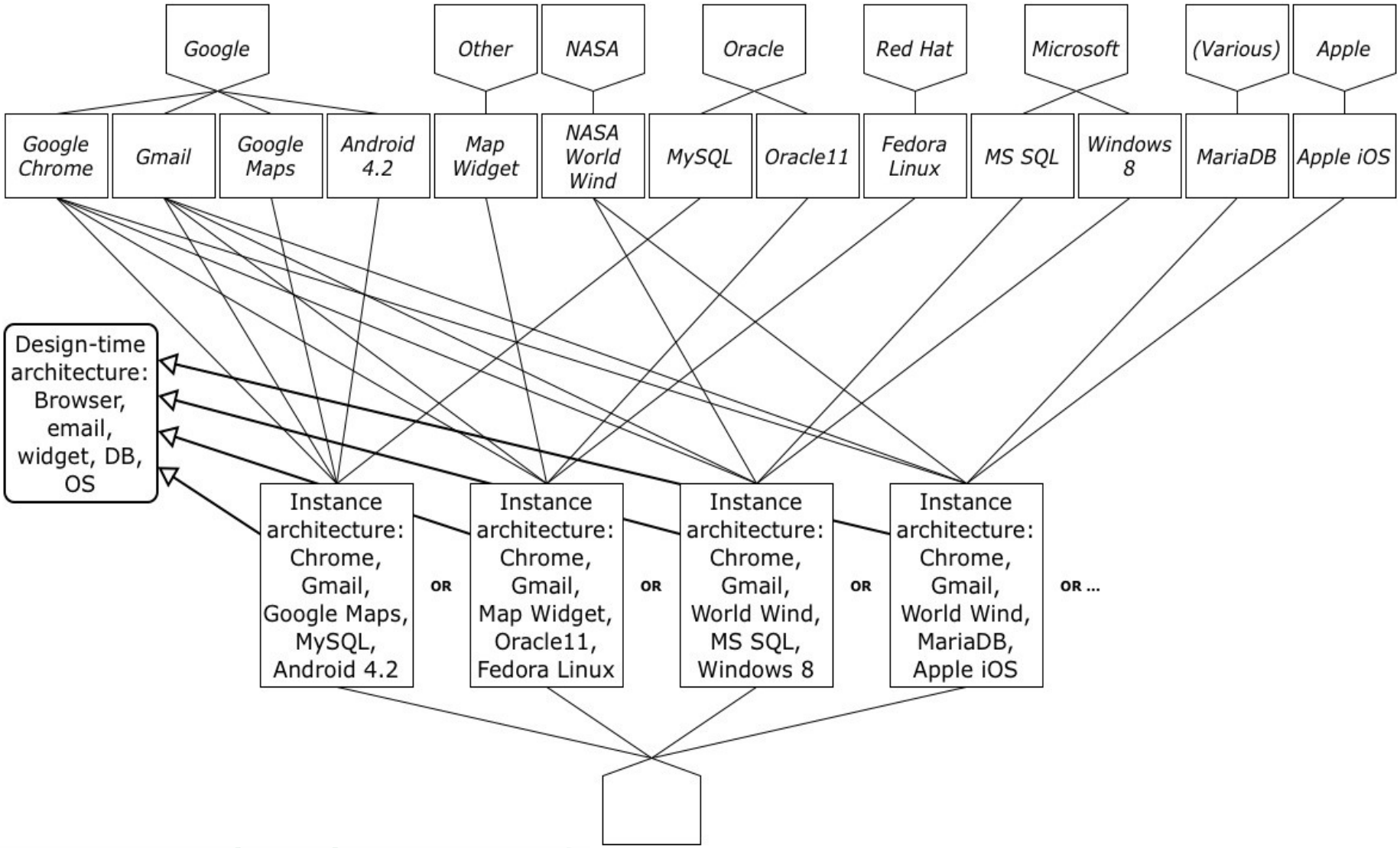
Software product lines?

- 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.

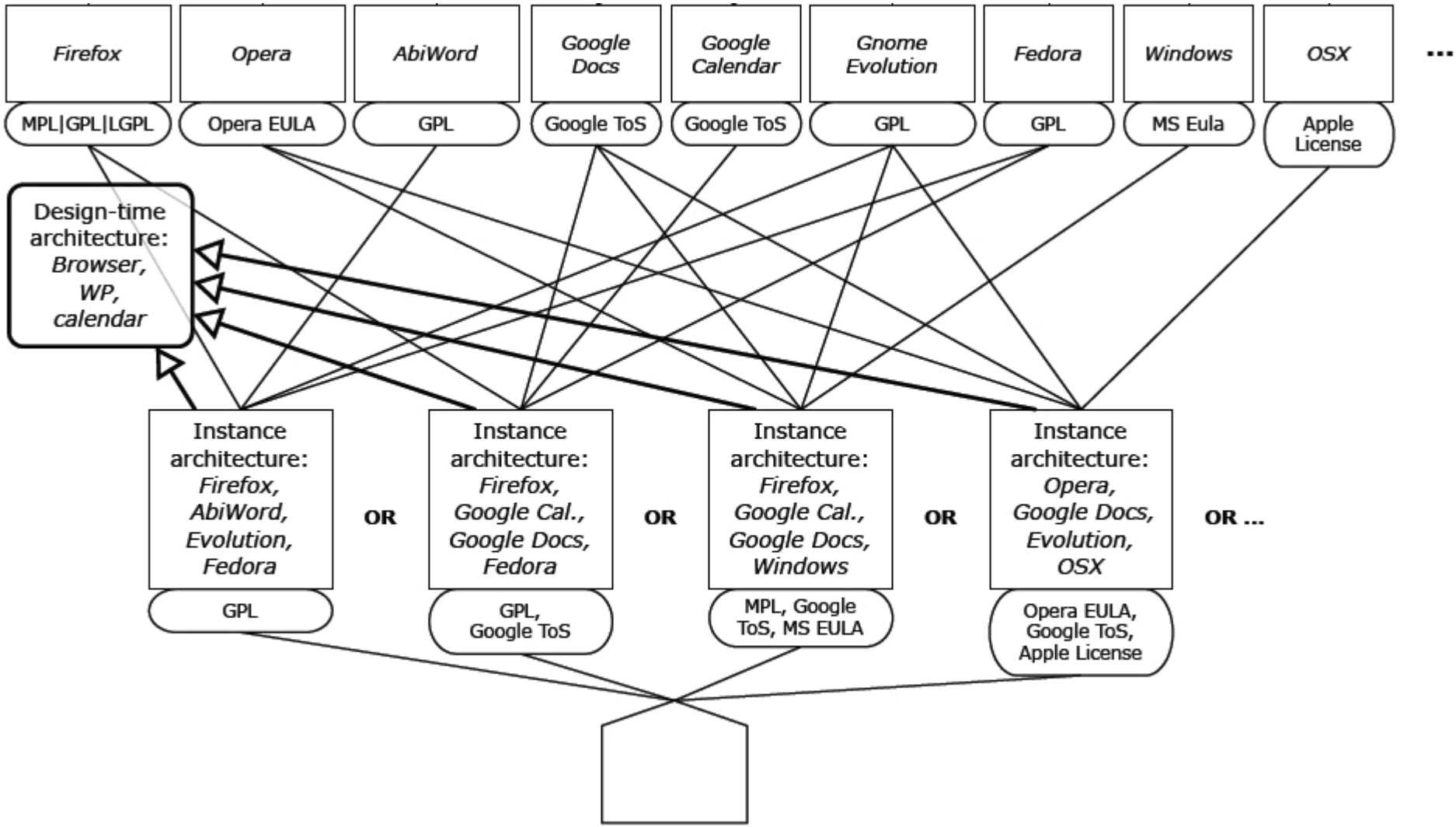
Software ecosystem of producers and the software components or application widgets for an enterprise system



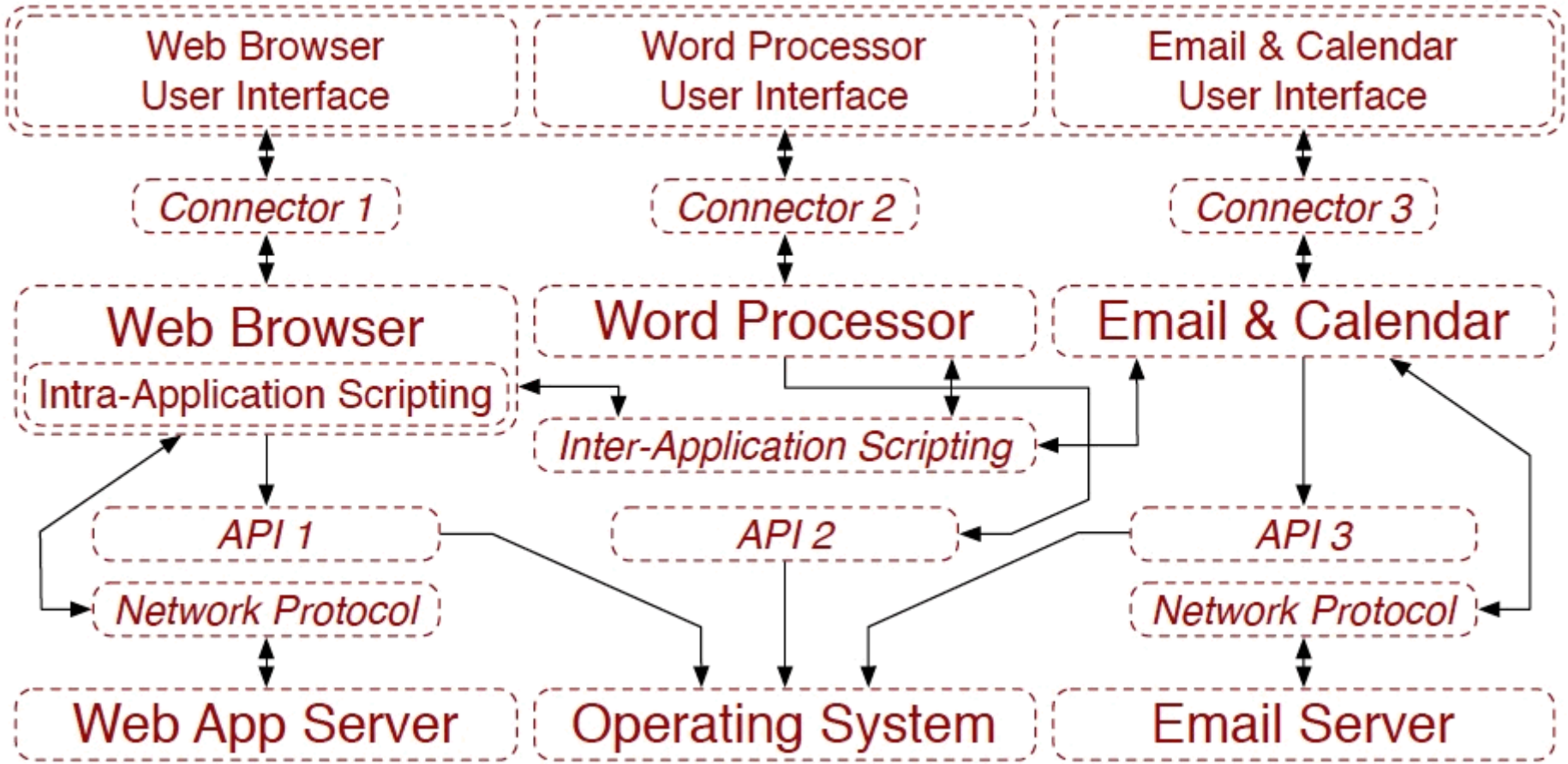
Software ecosystem of components or application widgets for an OA system



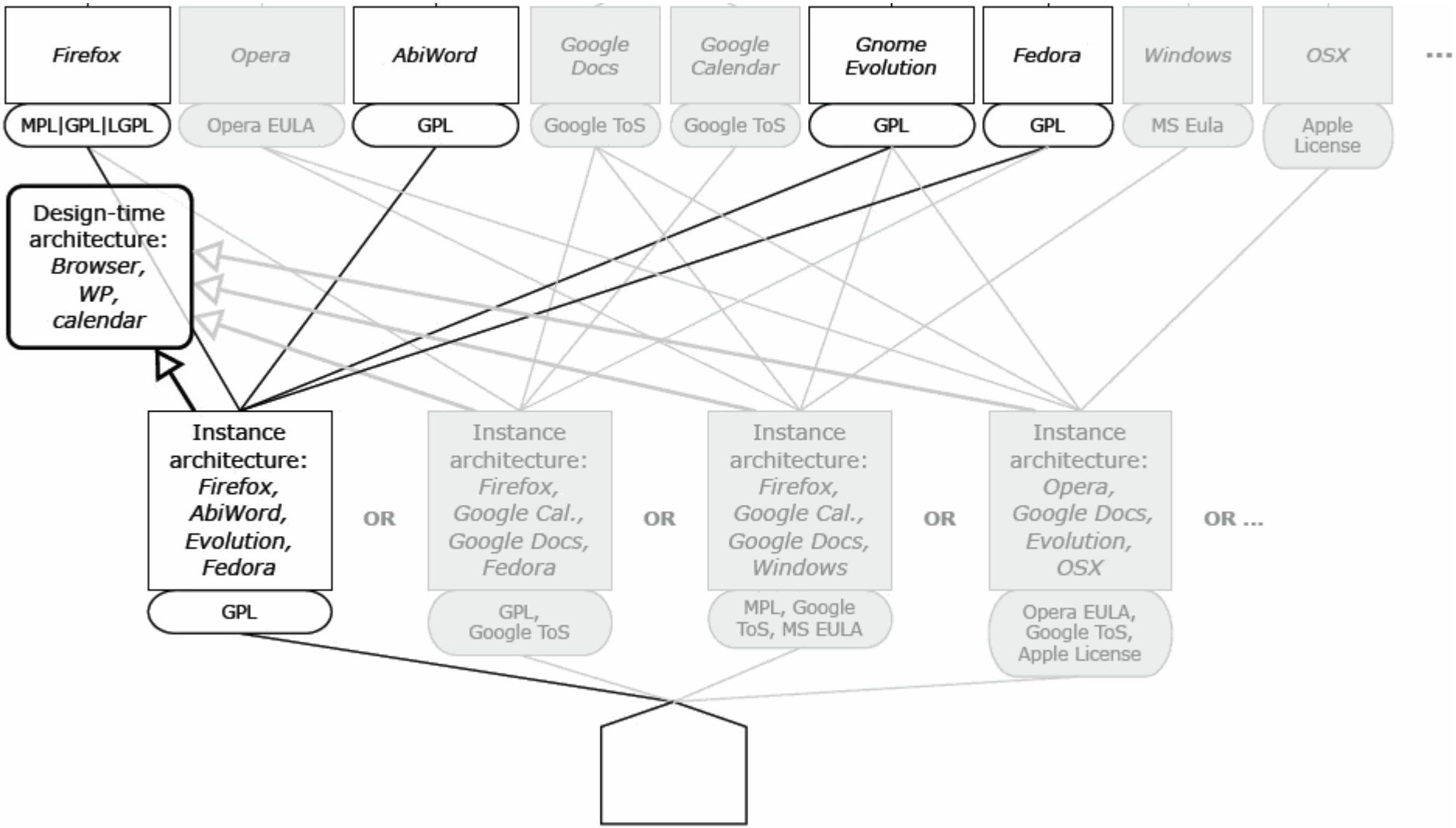
Software product line that provides functionally similar components or applications compatible with an OA system design



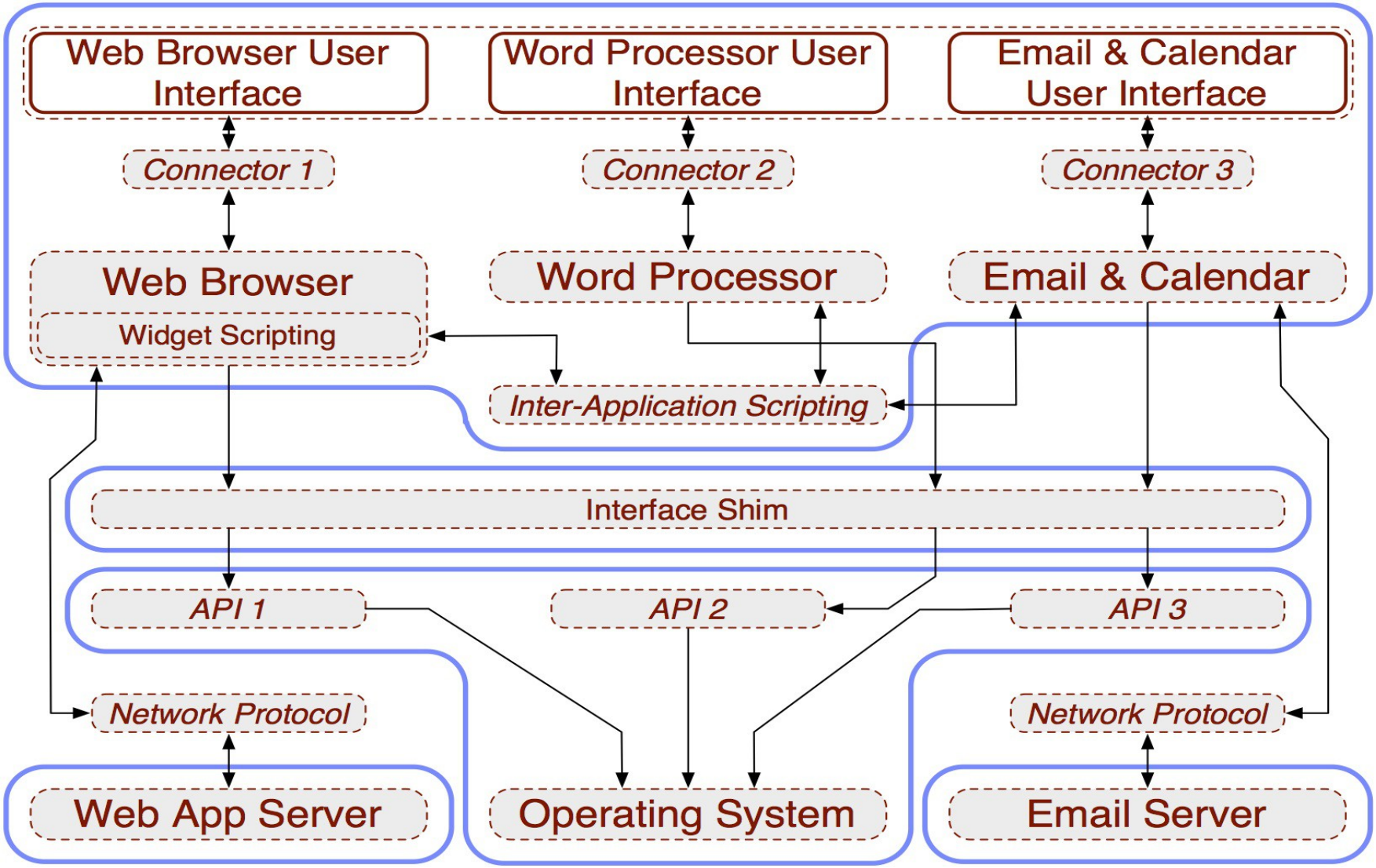
A *design-time* specification of an OA system that accommodates multiple alternative system configurations



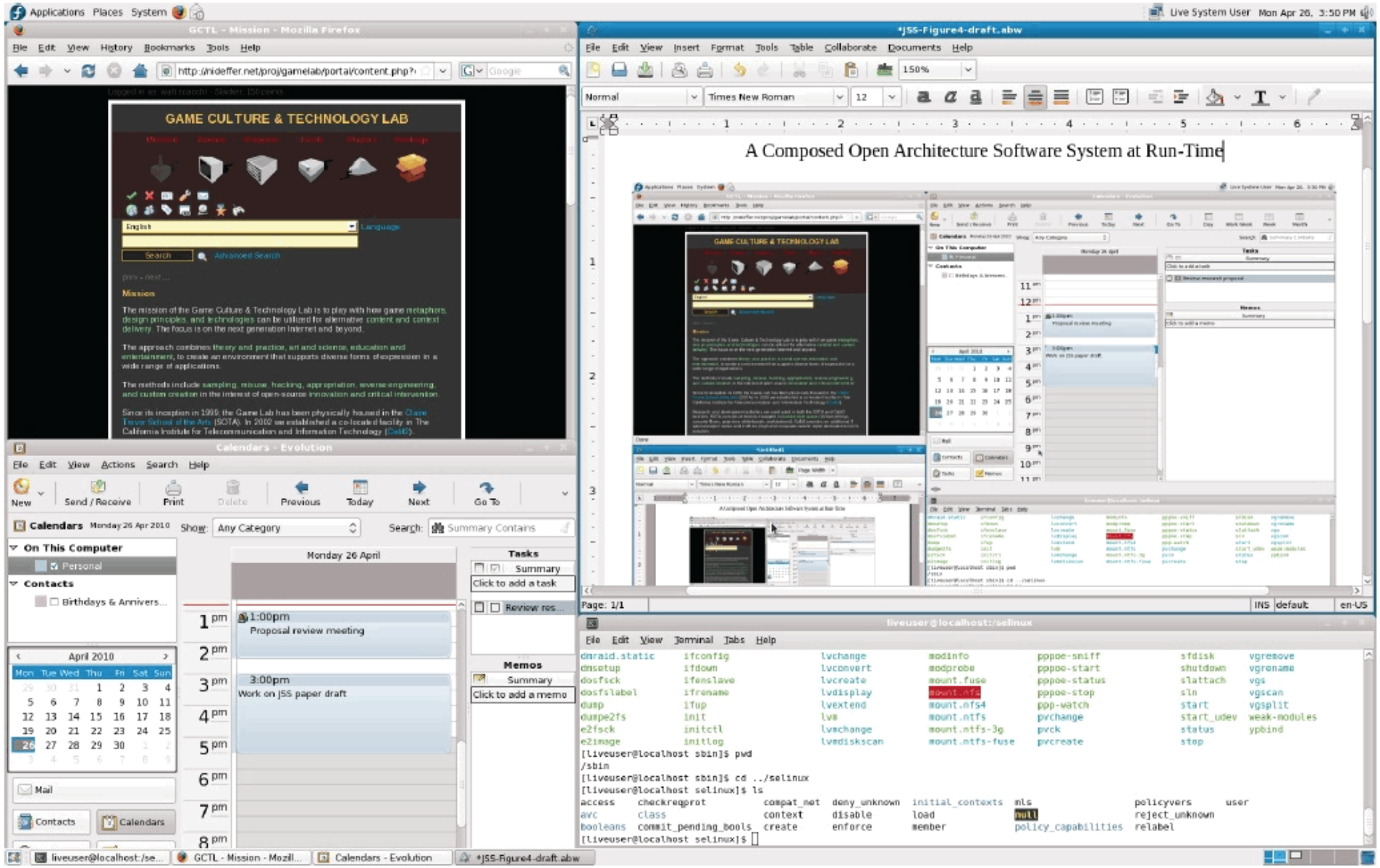
A *build-time deployment selection* among alternative components that produce an integrated enterprise system within the product line



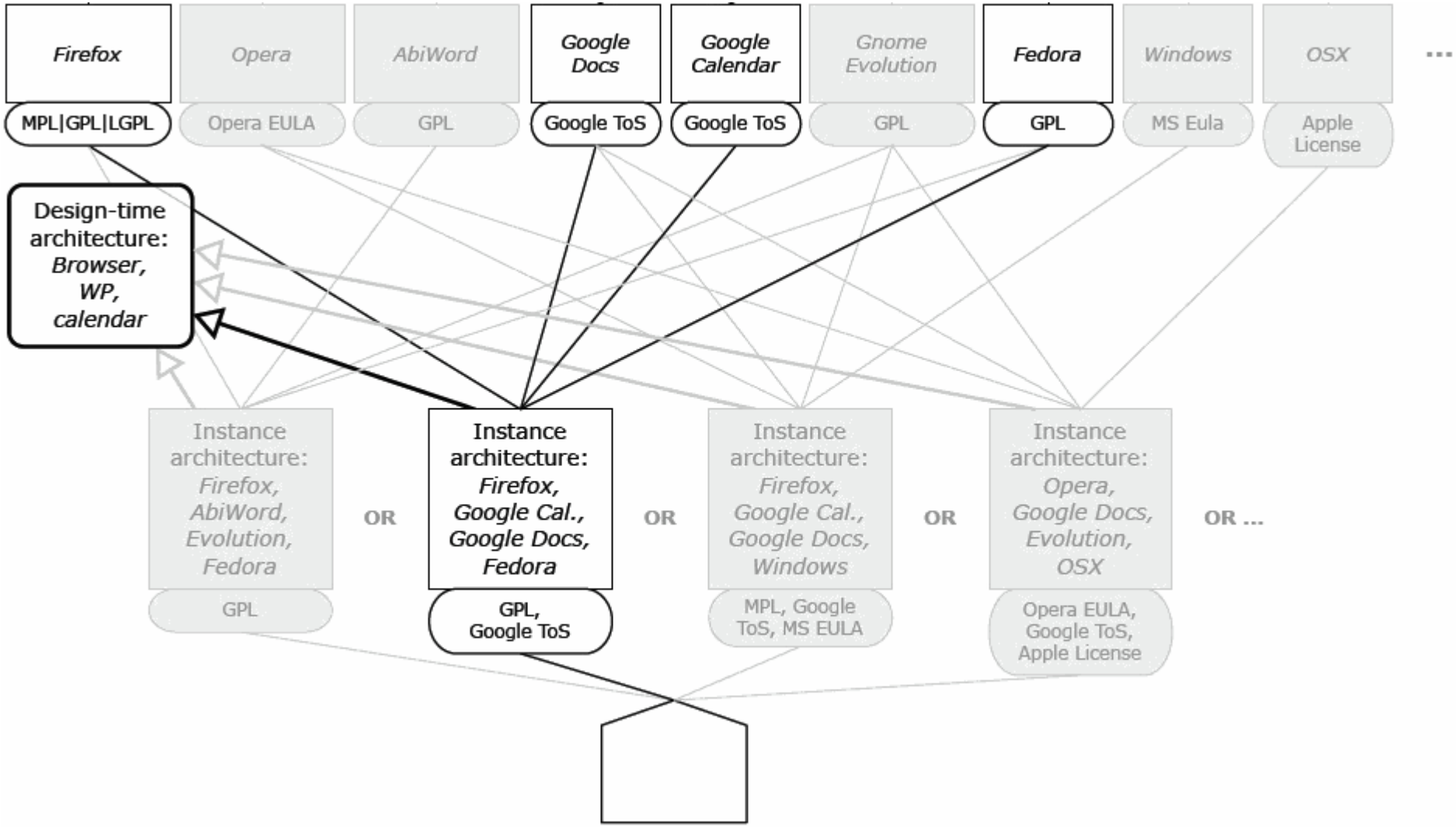
A security capability specification encapsulating the *run-time deployment* configuration via multiple virtual machines



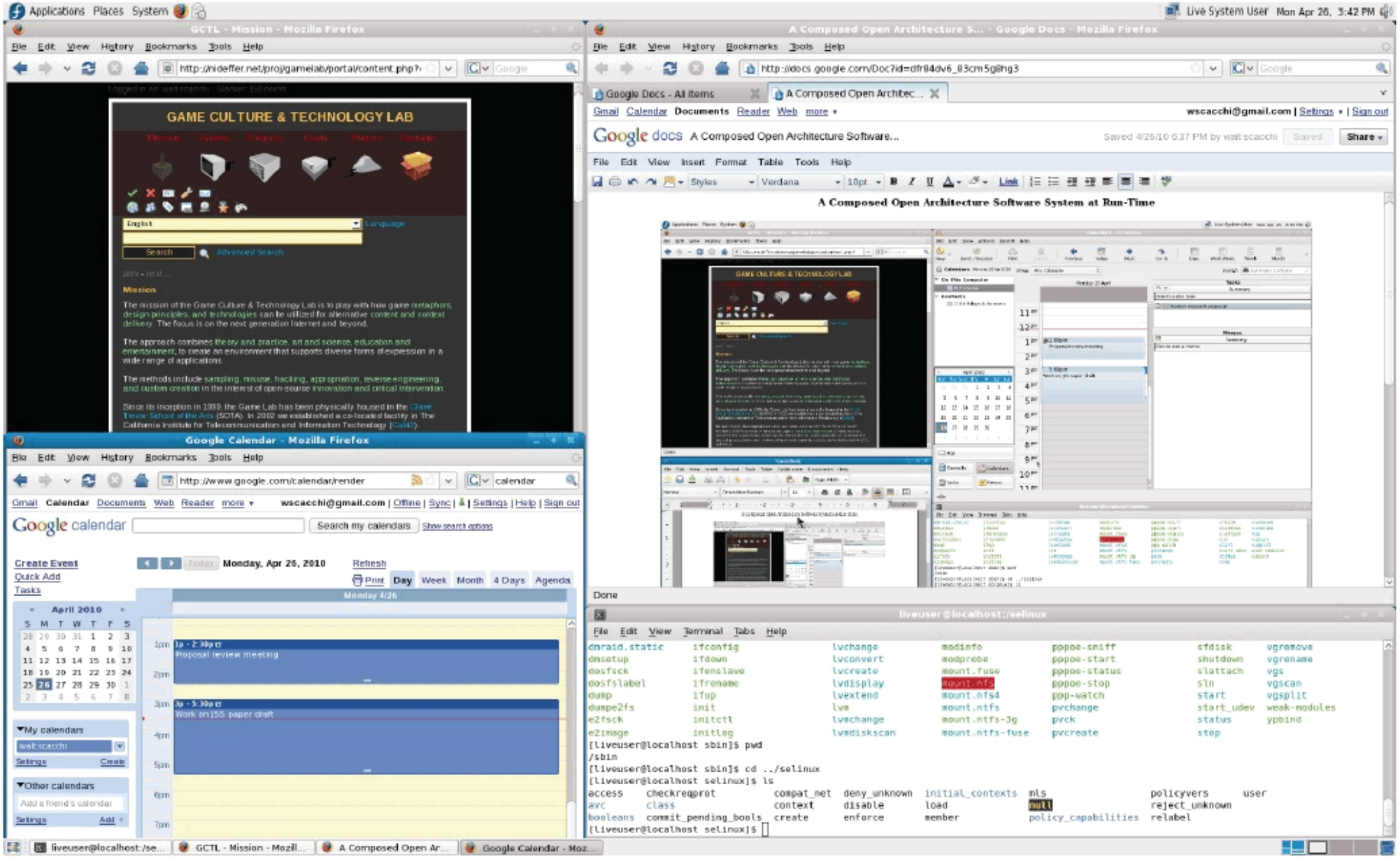
An end-user *run-time deployment version* of selected components within enterprise system product line utilizing security library, **SELinux**, for enforcing mandatory obligations and rights.



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



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.