



Experiences in the Development of a Service-Based Scientific Application Framework

Mark L. Green¹, Catherine L. Ruby¹, Stephen D. Miller²

¹Tech-X Corporation, Buffalo Office, Systems Integration
Group

²Oak Ridge National Laboratory, Spallation Neutron Source



Scientific Application Experiences Outline



- Experiences
 - Sustainability
 - Supporting a software project, and its people, over an extended period of time
 - Commonality and Interoperability
 - To what degree technologies and methodologies must be “homogenized” and where integration approaches can be more effective: i.e. when square pegs *must* fit into round holes
 - Enabling diverse organizations with different technologies to work together in an effective manner
 - Graphical User Interfaces and Service Interfaces
 - Providing intuitive and robust interfaces that are both agile and reusable
 - Facilitating interoperability and sophisticated capabilities through modular, well-defined and standards-based services.
 - Flexibility
 - Agile component-based development allowing organic growth in an ever-changing environment
 - Enabling both user-side and developer-side statistics for a better understanding of the overall system
 - “Thing build right” versus “Right thing built”





Scientific Application Experiences Outline (continued)

- Show of Hands
- Systems Integration Approach
- Questions



Scientists vote with the feet



TECH-X CORPORATION



Sustainability Experiences

- Providing dedicated funds for long term software projects is extremely difficult!
 - Can facilities provide 10s of FTEs for application development?
 - Can facilities collaborate at a committed level of 10s FTE over the long-term?
 - History tells us that this is not sustainable and these large scale software projects collapse into a very low level maintenance mode with feature freeze. What will happen with DANSE?
 - Application based closed source or restricted open source are hard to sustain. GPL vs. BSD?
 - Organic growth is required over the long-term to sustain a software project which requires buy-in from the user base. Open Science Grid?
 - Infrastructure development is more sustainable than application development. Amazon, Google, Yahoo, etc.



Commonality and Interoperability Experiences

- Should every square peg fit in a round hole? Why should one size fit all?
- Loosely coupled components with well-defined interfaces are required.
 - How do you develop an application that is flexible enough to meet the needs of a diverse and evolving user base?
 - Stove piped application are common place. Is APS working?
 - Is selling the next new technology going to met these needs? Possibly if there is a pluggable infrastructure? Best of breed?
 - Is it okay to be different? Are Amazon, Google, and Yahoo identical? Mashups are enabled by the infrastructures and standards.



Graphical User Interfaces and Service Interface Experiences

- Is anyone ever really happy with someone else's user interface?
 - All components should be coded in their language of choice and presented through well-defined interfaces.
 - Interfaces lead to well-defined components which lead to flexibility, sustainability, and code re-use.
 - De-coupling user stove pipe application from forced GUI choices or technology adoption.



Flexibility Experiences

- Complex systems are not simplified by making them more rigid!
 - Does one framework meet the needs of all facilities and applications now and into the future? How would this be possible?
 - Should you code to a framework? Or do you code around a framework? To stop this there is only one solution here, multiple frameworks that better meet the needs of the user base.
 - Multiple frameworks can use standardize interfaces to interoperate.
 - Can systems integration be performed at the framework level with a flexible infrastructure? Eclipse Rich Client Platform Integrated Development Environment.



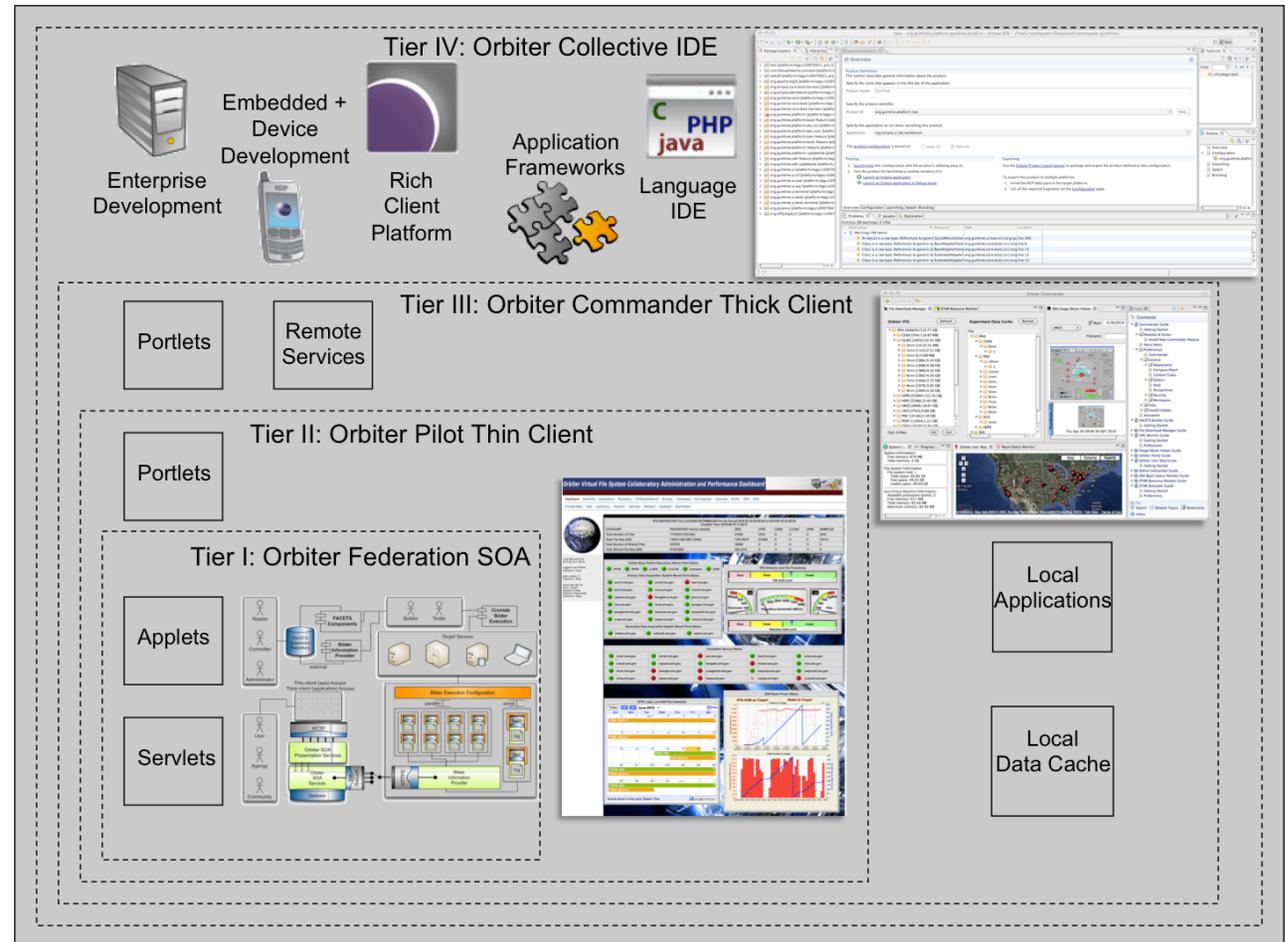
Show of Hands

- Who is using off-site HPC resources for their applications?
- Who is using only facility-owned computing resources for their applications?
- How many would consider their software development budget as:
 - too high?
 - sufficient?
 - extremely under funded?
- Who is delivering their application via web only?
- Who is delivering their applications by desktop/laptop installations?



Orbiter Multitier Portal Architecture (MPA)

- Framework for delivering capabilities to thin- and thick-clients using the Orbiter RESTful SOA
- Flexible and re-usable architecture for developing capabilities for thin web clients and thick local clients
- Comprised of four tiers: Orbiter Federation, Orbiter Pilot, Orbiter Commander, and Orbiter Collective





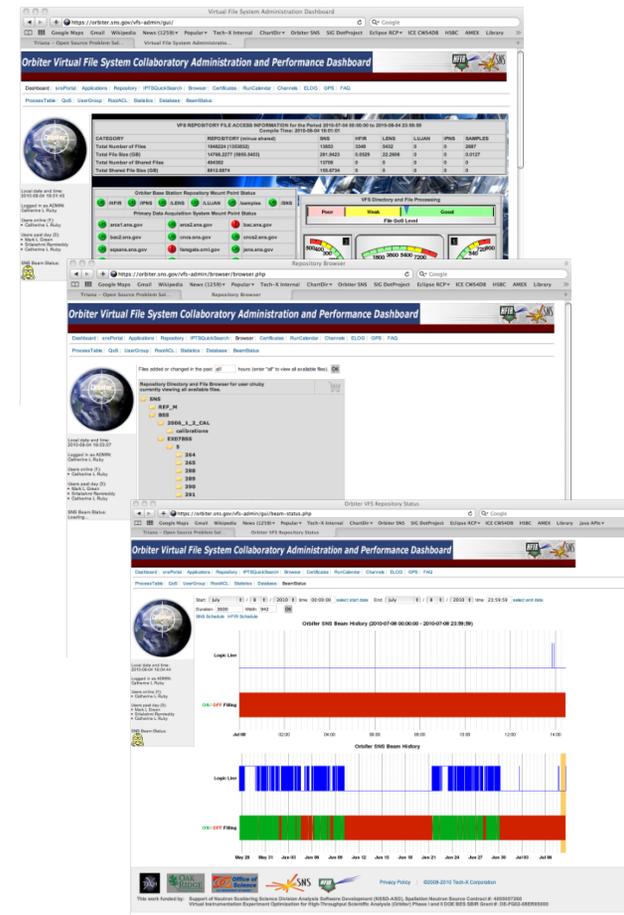
Orbiter Federation – Data, Logic, and Presentation

- Bridges the gap between instrument data and rich user interfaces
- Implemented as fast and secure RESTful services, delivering diverse capabilities
- SSL protocol and RSA PKI ensure service request privacy and integrity
- Master/slave database replication ensures data integrity
- Easy-to-use service APIs make capabilities accessible to a wide range of users and applications
- Standards-based schema and WSDL define easily reusable service interfaces



Orbiter Pilot – Web-Accessible Thin Client

- Tier II of the Orbiter Multitier Portal Architecture
- Accessible to users with accounts and internet access (via a web browser)
- Build upon the services provided by the Orbiter SOA infrastructure
- Rich, dynamic statistics and QoS metrics
- Live monitoring and information are easily accessible
- Virtual File System (VFS) browsing and download capabilities
- XCAMS/UCAMS authentication provides role-based authorization
- Public and administrative interfaces provide active control
- Rich search interface on repository files and metadata
- Flexible, integrated, and interactive tables, charts, and maps
- Instant online access for Firefox, Safari, Internet Explorer, and mobile devices





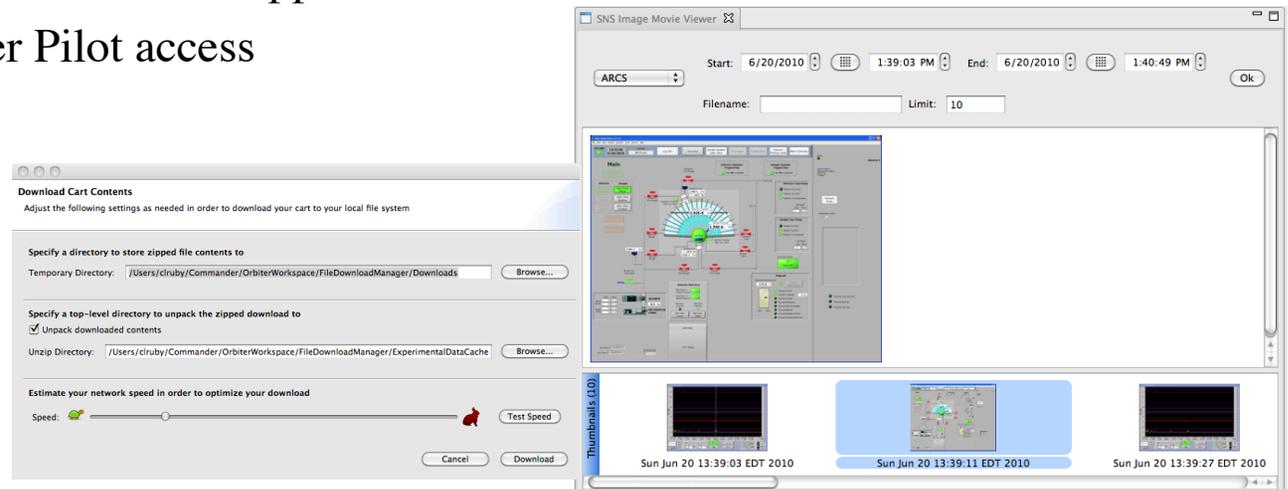
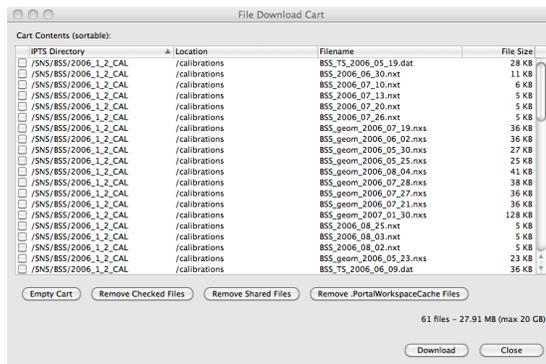
Orbiter Commander – Customizable Client

- Tier III of the Orbiter Multitier Portal Architecture
- Rich Client Platform (RCP) desktop applications for accessing Federation capabilities, run locally on user work stations or personal computers
- Suites and modules deliver diverse functionality
- Customizable interface lets users optimize their workspace
- Plug-and-play framework allows new modules to be added to the application seamlessly
- Rapid multi-threaded download for optimized access to Orbiter VFS files
- Integrated help offers on-the-spot support
- Seamless integration between Federation Services and local desktop resources
- Cross-Platform compatibility with Windows, Mac OS X, Linux, AIX, and HP-UX
- Build upon the services provided by the Orbiter SOA infrastructure
- Allows users to run complex simulations or computationally-intensive tasks on their local machines, relieving QoS concerns on web service providers



Orbiter Commander – Customizable Client (continued)

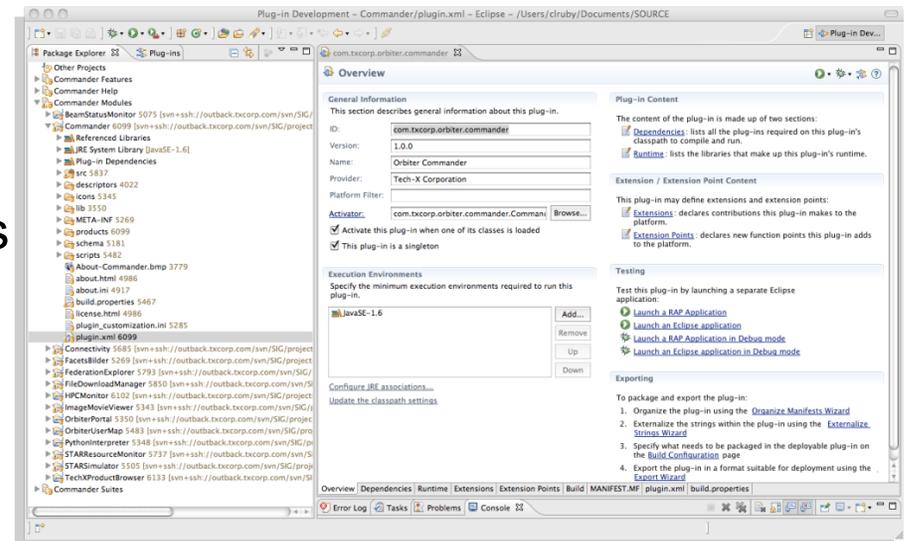
- Cross-Platform compatibility with Windows, Mac OS X, Linux, AIX, and HP-UX
- Build upon the services provided by the Orbiter SOA infrastructure
- Allows users to run complex simulations or computationally-intensive tasks on their local machines, relieving QoS concerns on web service providers
- Atomic capabilities are provided as *modules* that can be installed as needed from a central module repository
- The Orbiter RESTful SOA provides robust access to diverse capabilities, such as:
 - Multi-threaded streaming downloads of repository files
 - Live status monitoring of the SNS Beam
 - Slideshows of instrument application screenshots
 - Direct Orbiter Pilot access





Orbiter Collective – Module Development Environment

- Tier IV of the Orbiter Multitier Portal Architecture
- Development environment for Commander capabilities facilitating widespread community adoption and collaboration
- Eclipse RCP and the flexible Commander framework form the basis for future developments
- Well-defined extension points allow new suites and modules to be rapidly developed and built upon existing resources
- New Commander capabilities are easily deployed through external update sites
- In-house proprietary applications and open source third party tools can seamlessly be integrated to provide new capabilities





Questions

Thank you for your attention.

TECH-X CORPORATION