

Flex and Portlets

Cornel Creanga

Platform evangelist



Problem Definition

- Over the years, enterprises have built hundreds of applications with different technologies and UI approaches
- These applications are hard to use...
 - User has to start several disparate applications to perform a task
 - Each application works in its own way, with a different navigation system and overall user experience
 - Applications don't share context (common data) forcing the user to re-enter the same data in different applications
- ... and hard to maintain
- Traditional portal solutions have tackled this problem, but are not meeting today's new expectations in terms of user experience
- Flex can be used to improve the user experience of individual portlets, but the overall portal experience is still greatly impacted by the rigidity of the portal container

Nexus in a Nutshell

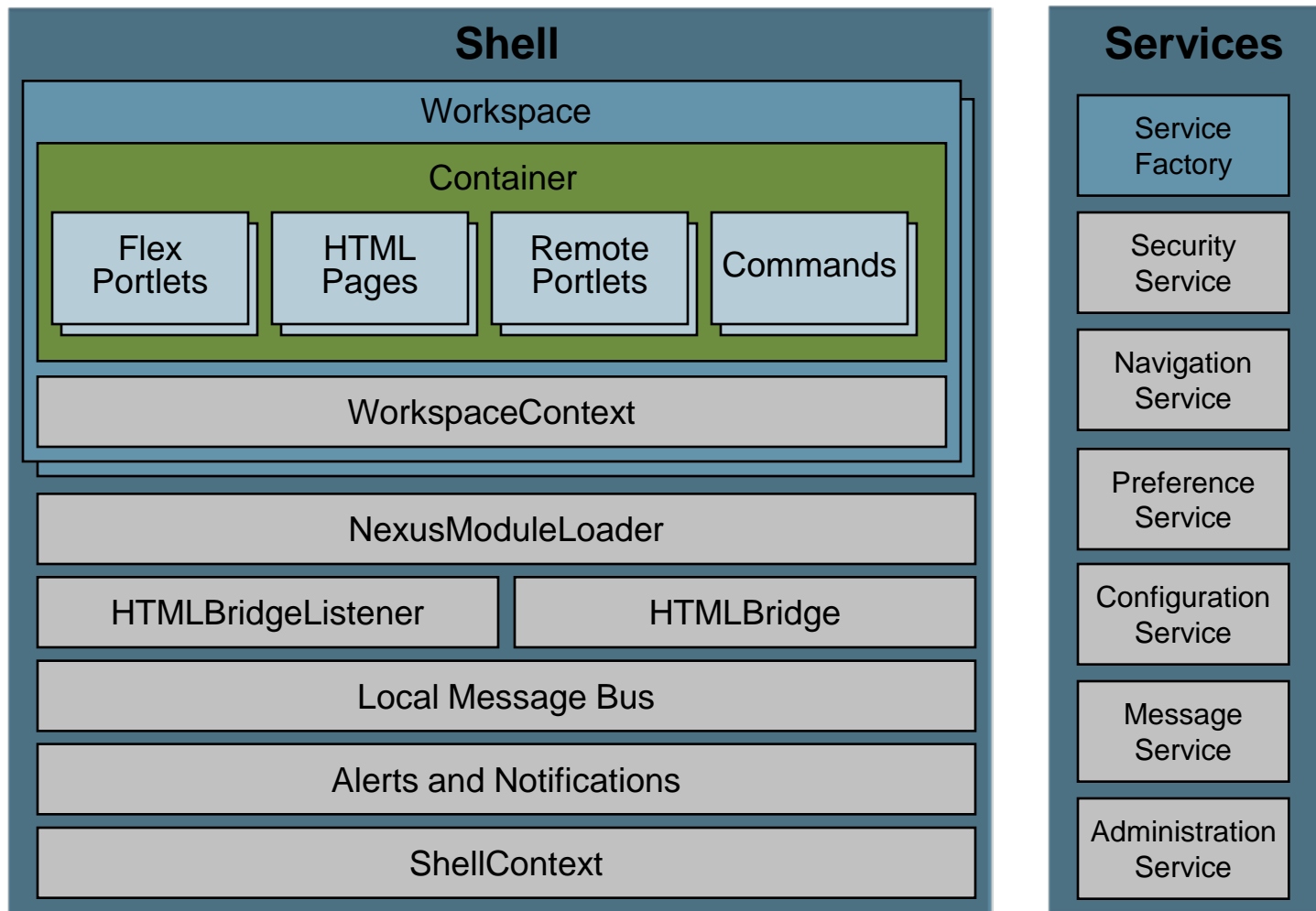
- Framework for application aggregation and composition
- Common user experience
- Flex and HTML content and APIs
- Role-based
- Common and pluggable navigation system
- Pluggable layout management
- Sophisticated context-sharing and inter-portlet communication APIs
- Context isolation
- Web and Desktop deployment options
- Memory management and versioning
- Alerts and notification APIs
- Enterprise integration

Built on Flex

- De facto industry standard for rich internet applications
- Unparalleled “expressiveness”
- Rich data visualization
- Single code base for browser and desktop deployment
- Large ecosystem of developers
- Virtually no learning curve:
Nexus applications are Flex applications



Nexus Client Architecture





Adobe