Carbon: Towards a Server Building Framework for SOA Platform

Srinath Perera, Ph.D.,
Senior Software Architect, WSO2 Inc.
Visiting Faculty, University of Moratuwa
Member, Apache Software Foundation
Research Scientist, Lanka Software Foundation

WSO2 The open source SOA company

Outline

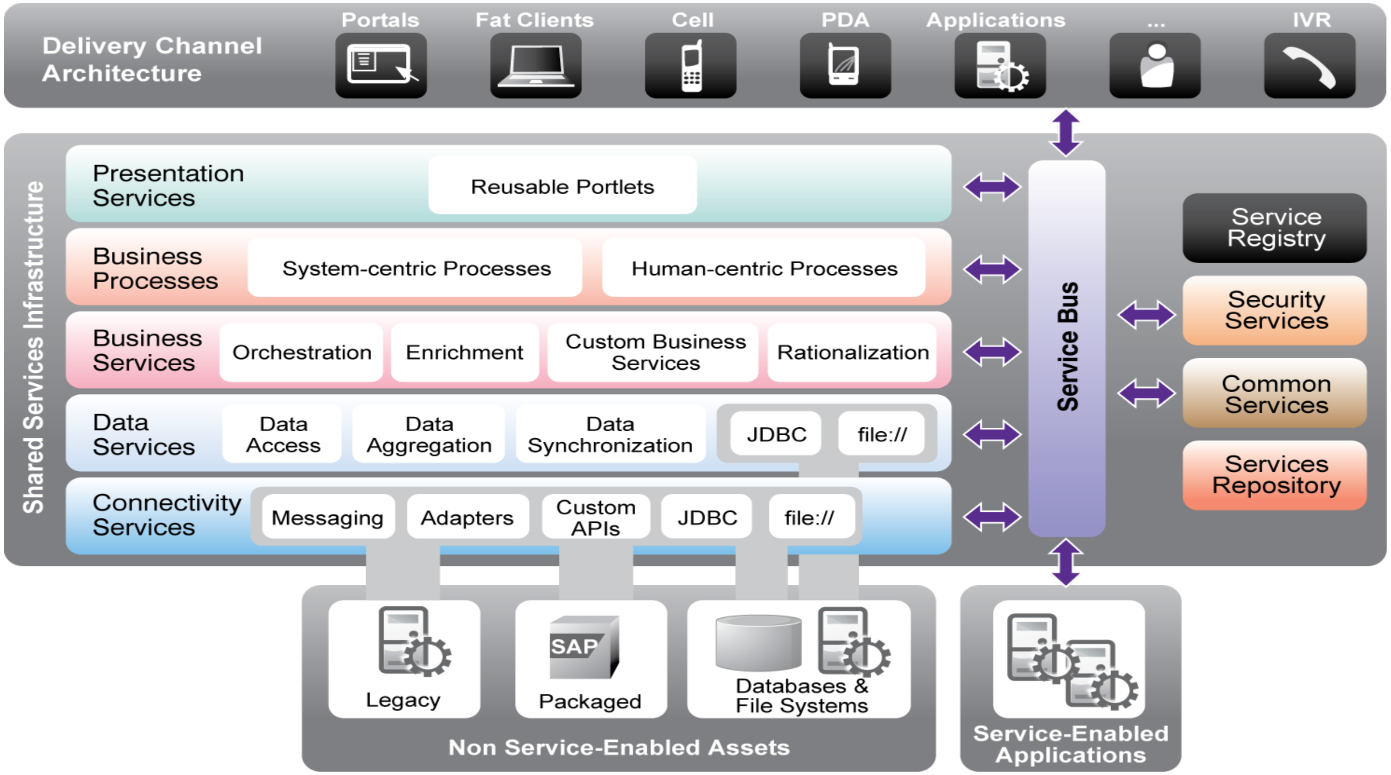
- Challenges of building a Next Generation SOA platform
- A Recipe for a Solution
- Carbon Platform Architecture
- How does it make a difference?
- Conclusion

Building Next Generation SOA Platform

AXIS2 Apache ODE shindig synapse Qpid

- At Year 2005, several contributors to Apache Web Services Project at WSO2 started an effort to build a next generation SOA platform by integrating, and extending Apache WS projects to a one platform.
- After 5 years, 3 major releases, and few thousand man months, we have more than 10 products that covers most of the SOA platform.
- Apart from SOA challenges, we faced major Software Engineering and Middleware challenges, and this is our story.

SOA Platform



- We are taking about a one platform that handles the breadth of SOA that includes a Application Server, ESB, workflow Engine, Gadget Server …
- They have to work as a one unit, not as 10 different projects, with single security model, governance model, look and feel etc..
- It is a single system that takes hours to download, hours to build!

Composition of SOA Platform

- SOA Platform should support many SOA and related concepts: Service, Workflow, Mediation, Mashup, Rules, CEP, Registry, Governance, Monitoring ...
- Also there are many cross cutting concepts like security, UI, logging, clustering, caching ..
- We typically handle platform as collection of servers, where each service supports some of the concepts.

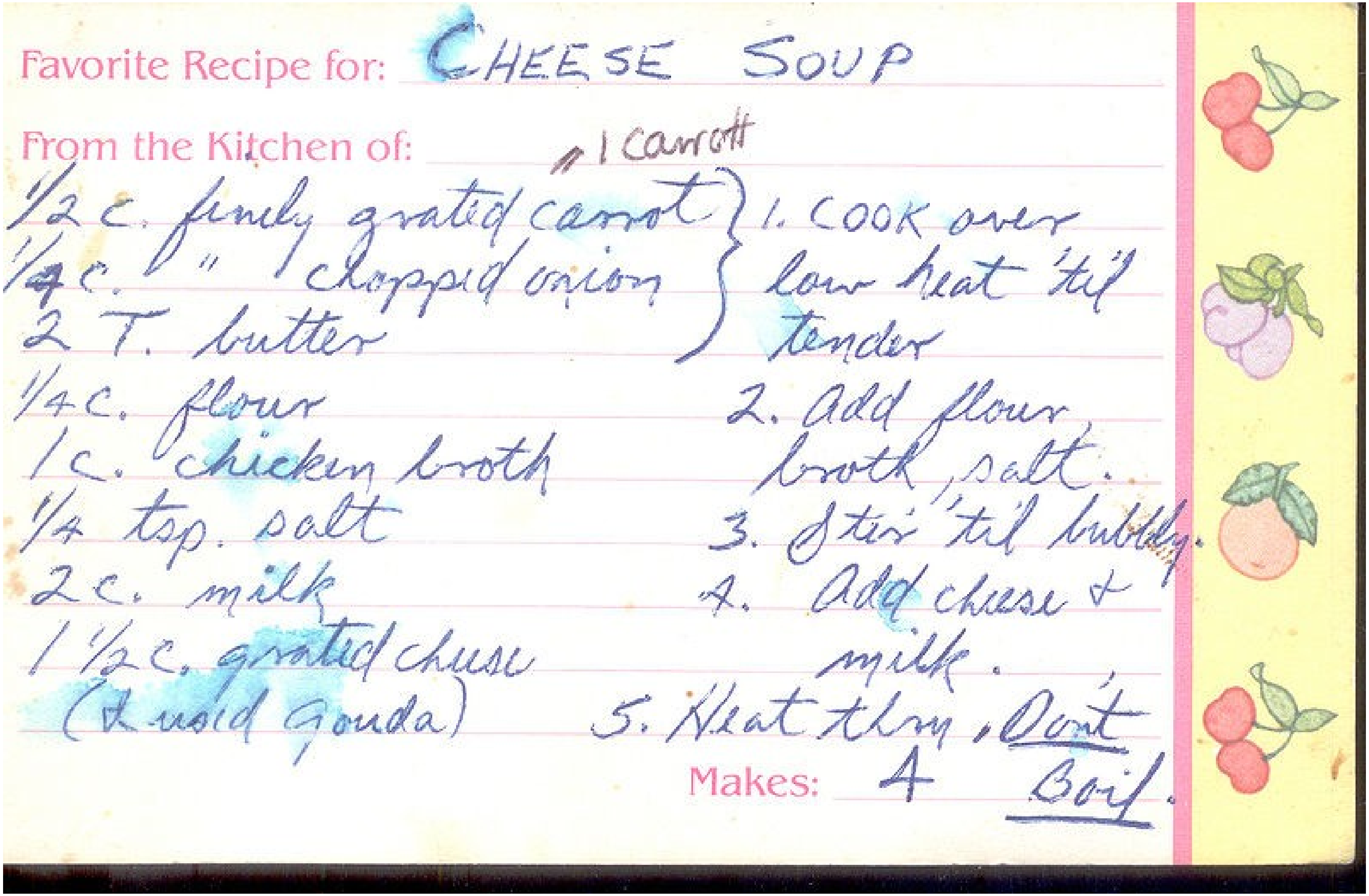
Challenges

- Avoid duplicates and maximize sharing
  - Sharing cross cutting aspects like security, storage, UIs etc.
  - Maximize sharing across different features.
- Reduce the coupling between different parts, and enable composition of features
- Make the platform extensible
- When writing a new extension, reduce the work need to be done to enable cross cutting aspects and to integrate with rest of the platform.
- Do all above with first class support to SOA

Challenges: Pizza Parlor Example

- Customization vs. few sizes fit all solutions
- With software it forces solution architecture to be conformed to available products.

Recipe for a Solution



Favorite Recipe for: CHEESE SOUP

From the Kitchen of:

1 carrott

1/2 c. finely grated carrot
1/4 c. " chopped onion
2 T. butter
1/4 c. flour
1 c. chicken broth
1/4 tsp. salt
2 c. milk
1 1/2 c. grated cheese
(I used Gouda)

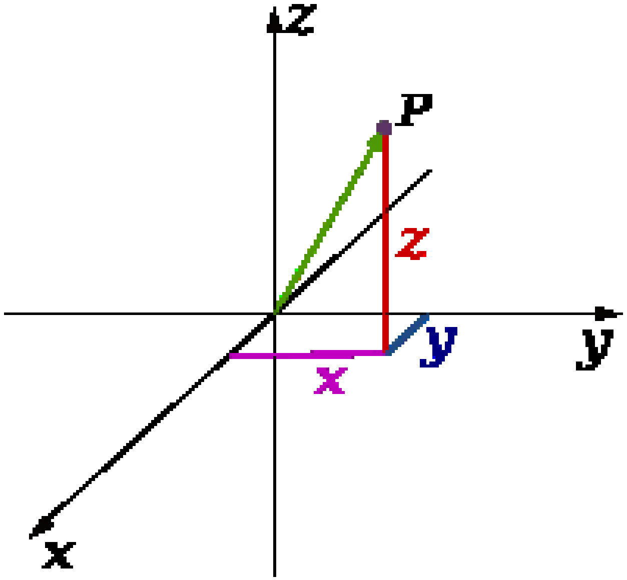
1. COOK over low heat 'til tender
2. Add flour, broth, salt.
3. Stir 'til bubbly.
4. Add cheese & milk.
5. Heat thru, Don't Boil.

Makes: 4

CBSE and Complex Systems

- CBSE (Component based Software Engineering) is used to handle complex systems in a loosely coupled manner.
- Define a component
  - Unit of deployment, versioning, and replacement
  - provide interfaces, support introspection, reflection
  - Have a deploable archive format
  - Include Metadata about itself
  - Define dependancies and often support IOC (Inversion of Control)
  - Can be composed at deployment or runtime

Kernel of the Platform



- As mentioned with challenges, servers, as well as different features shares many things, like security and execution
- Building an kernel which holds the main functionality is often use to handle such situations.
- Idea can be best explained through kernel of vector spaces, which provide a orthonormal basis for the vector space.

A Server Building Framework?

- Since SOA platform includes many servers that have different functionality, we argue that we should try to build a server building framework for SOA platform
- One potential approach is to define different functionality as components and compose them to build servers.
- IOC (Inversion of Control) based component framework

Carbon Platform

Goal

- Mostly what we discussed as challenges
- We look to build a Server Building framework
  - On top of a Component model that support IOC
  - Support first class support to SOA
  - And support building servers through composition of components

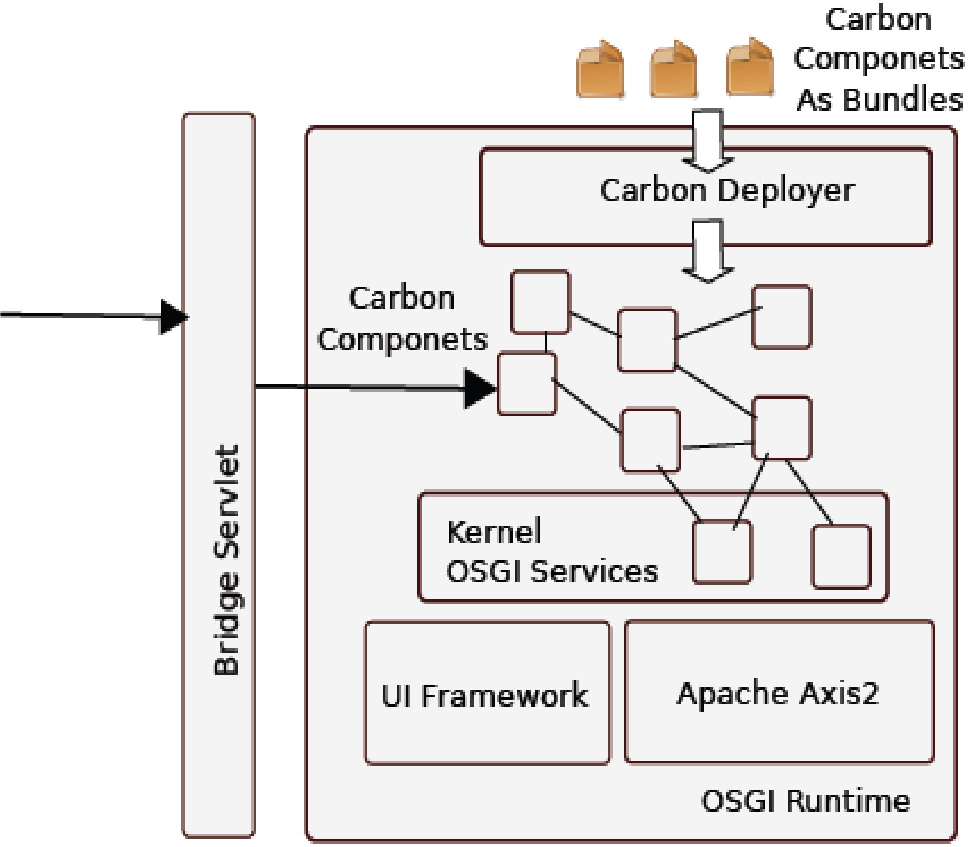
High Level Architecture

- High level architecture includes three parts
  - Carbon component framework which extends OSGi to support SOA artifacts
  - Carbon runtime that integrates Axis2 with OSGi and support security and UI support.
  - Kernel Services
- These are supported within the Carbon Core, and other components are built on top of the Carbon Core.

Carbon Components

- Carbon is built on OSGi (Eclipse Equinox)
  - Using Components, IOC support, and P2 support from OSGi
- Extends the OSGi components to support SOA artifacts as first class.
- Carbon integrates Axis2 with OSGi and supports SOA specific metadata within Carbon components
  - E.g. Component can include service definitions which are identified and deployed by Carbon.

Runtime Architecture



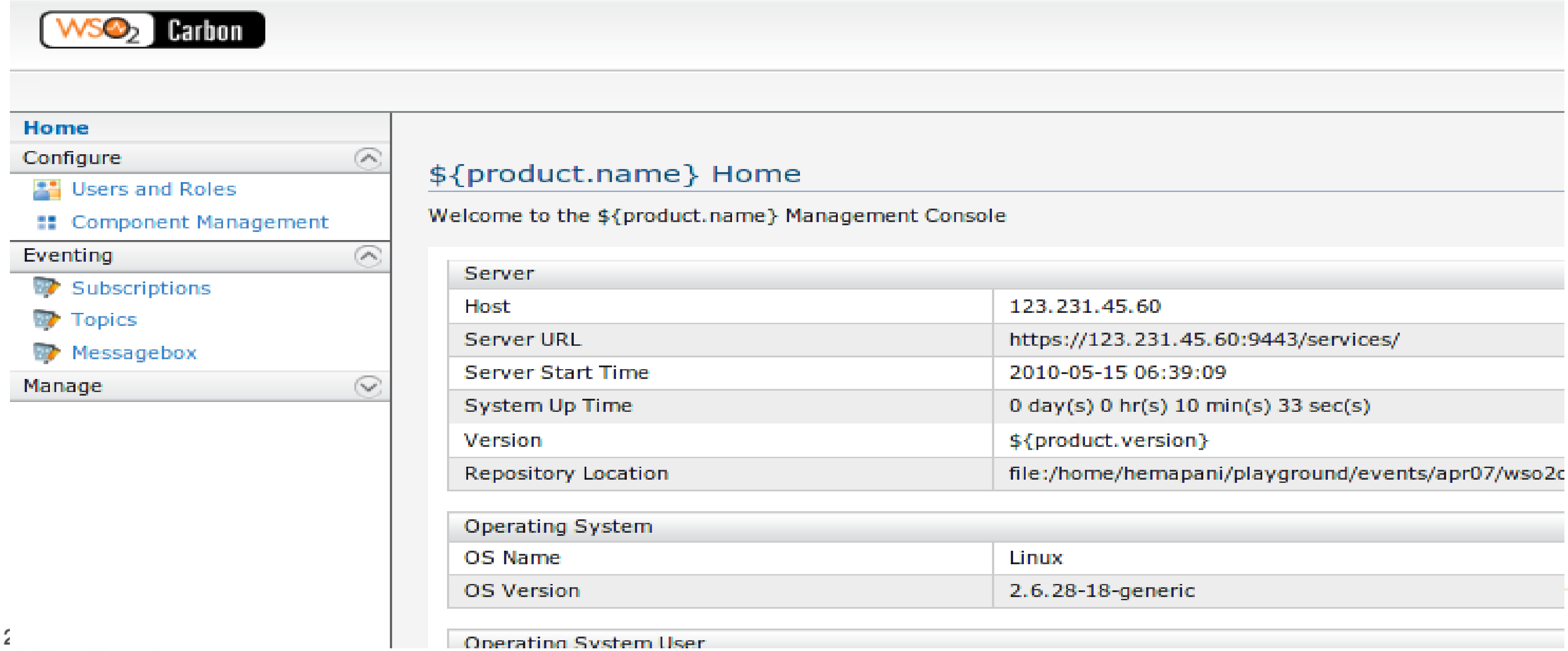
- Integrates Axis2 to OSGi
- Detect, process, and support SOA specific metadata from Carbon bundles.
- Intercept all requests through “Bridge Servelt” and support
  - Security model
  - UI support

Kernel of the Platform

- We support following as kernel services through Carbon core (Available as OSGi services through IOC)
  - Execution (supporting services and workflows)
  - Data Storage
  - Security (user management, authentication, authorization)
  - User Interfaces
  - Other Services (monitoring, caching, clustering etc.)
- They are used by most components and simplify development of new components.

Event Server as a Example

- Components
  - A Service
  - Admin UI
  - Admin Services
- All Implemented as components and composed with Carbon.



${product.name} Home

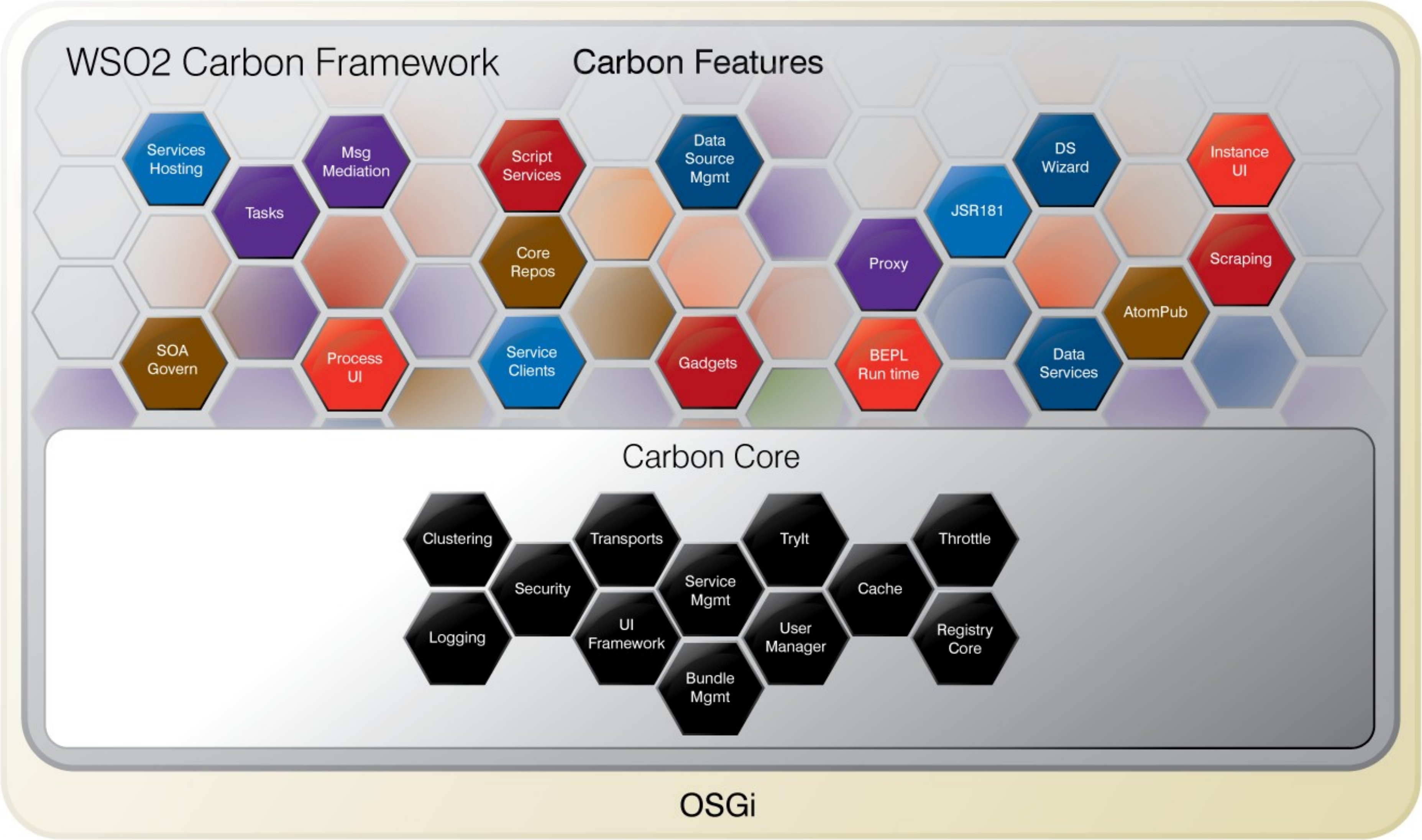
Welcome to the ${product.name} Management Console

Server	
Host	123.231.45.60
Server URL	https://123.231.45.60:9443/services/
Server Start Time	2010-05-15 06:39:09
System Up Time	0 day(s) 0 hr(s) 10 min(s) 33 sec(s)
Version	${product.version}
Repository Location	file:/home/hemapani/playground/events/apr07/wso2c

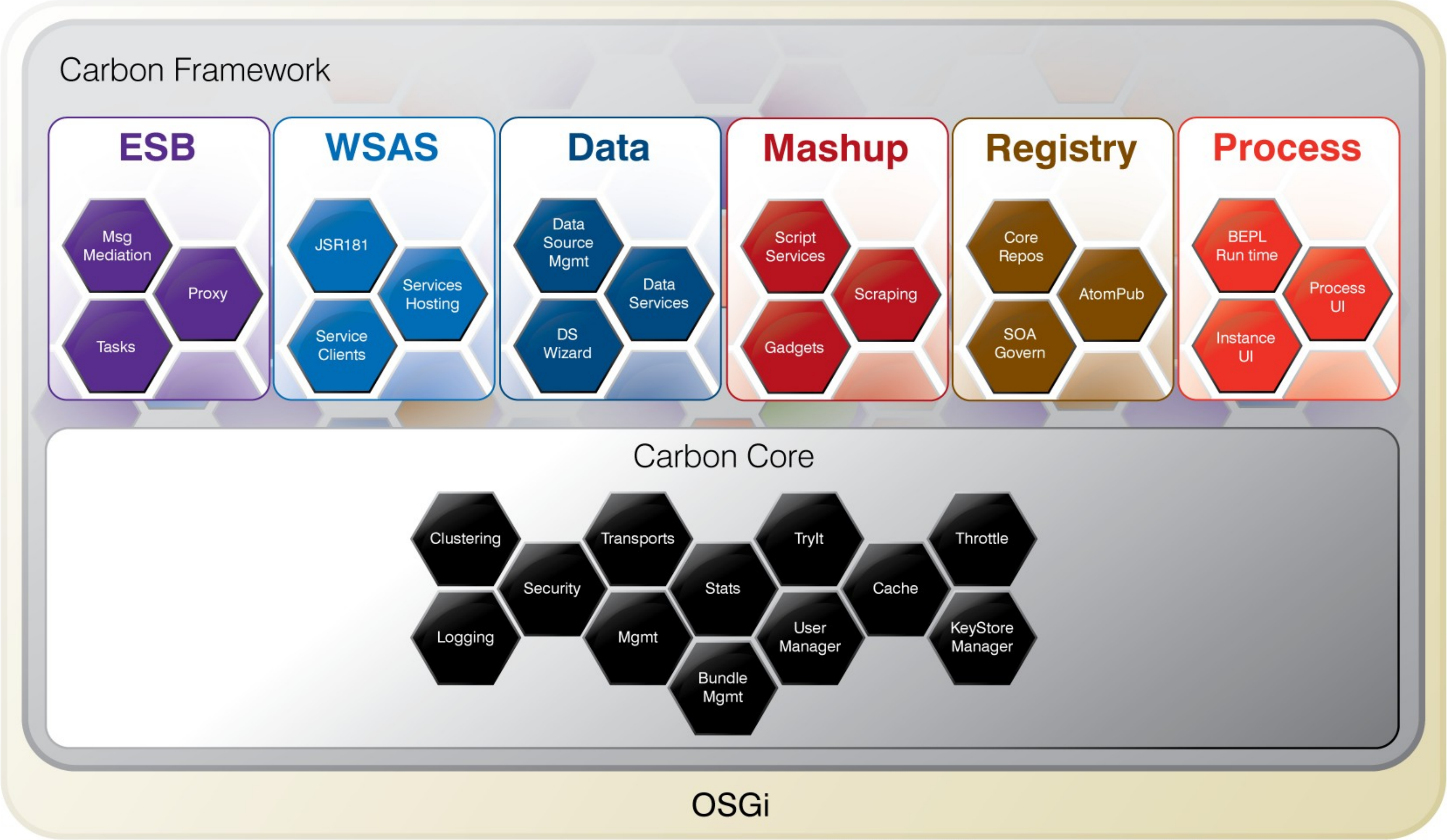
Operating System	
OS Name	Linux
OS Version	2.6.28-18-generic

Operating System User

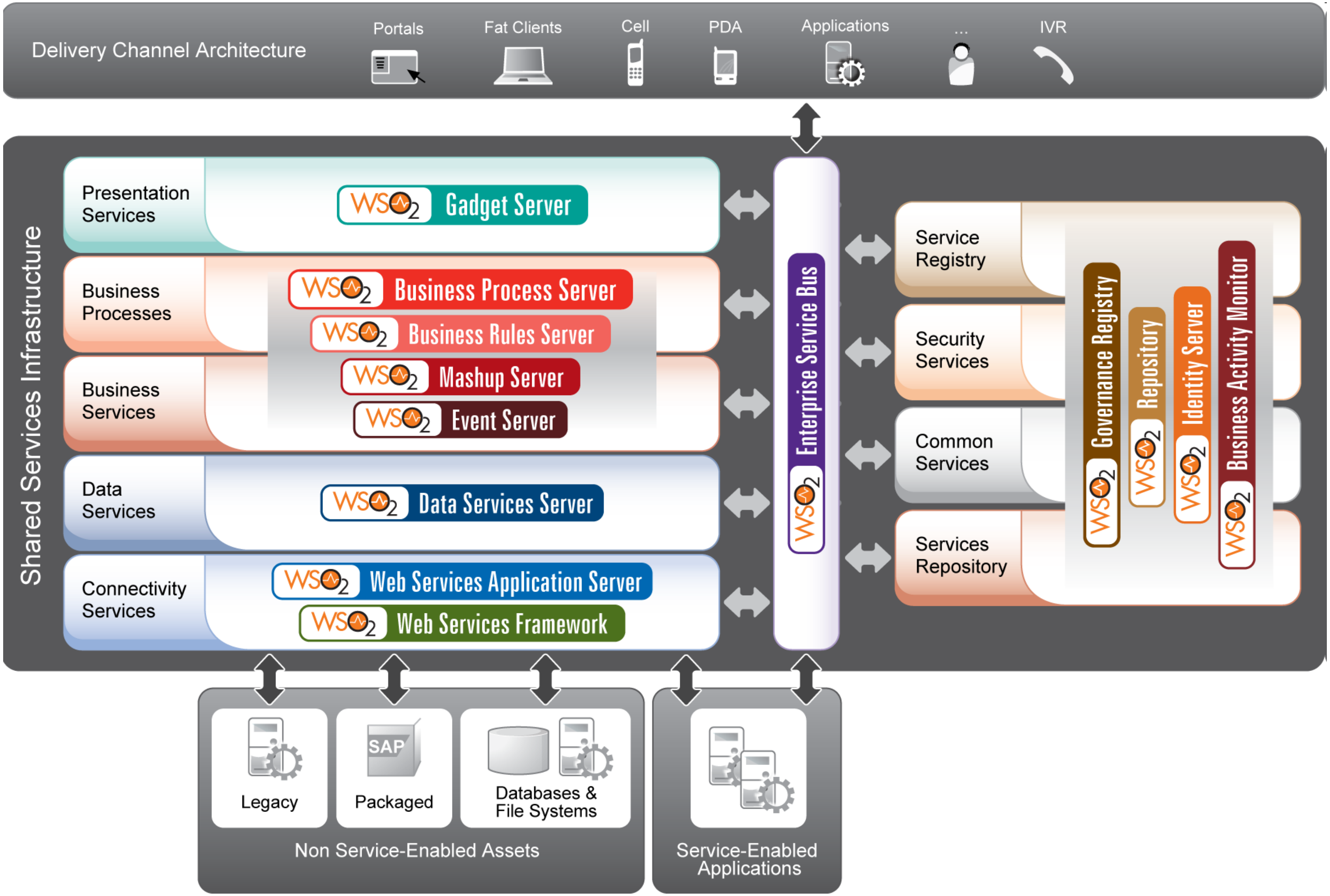
Carbon Components



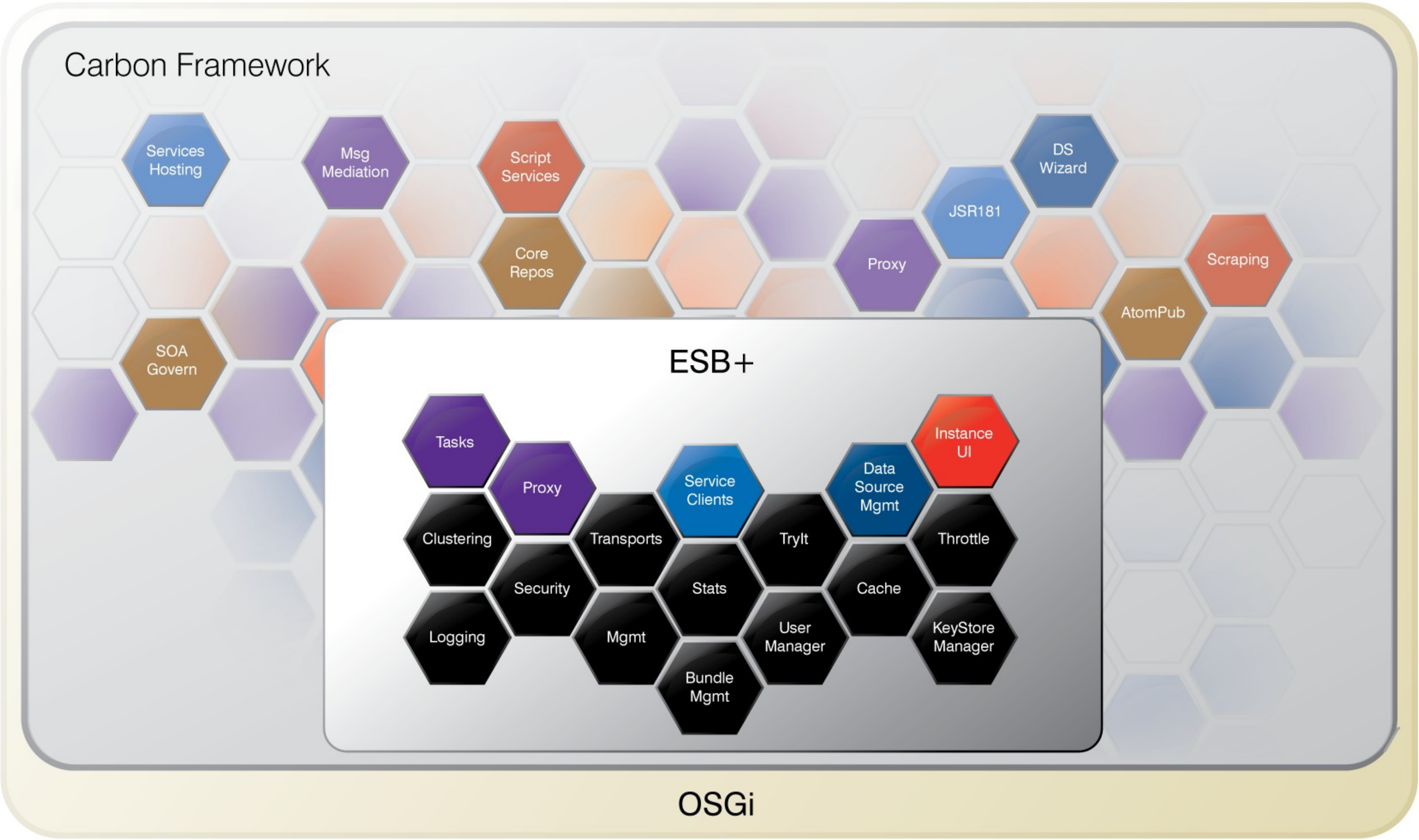
How Products are Built?



WSO2 Carbon Platform



Building Custom Products



How to Handle Conflicts?

- Can components clash when they are composed in different combinations?
- Mainly through conventions
  - All inter-component communication is through OSGI components
  - Store all configurations in its own space
  - OSGI ensure all dependancies are available
- Resulting system has the same characteristics as a loosely coupled SOA system where users pick and choose services and create applications using them.

How does Carbon make a Difference?

How does it make a difference?

- Enable architects to draw server boundaries according to the problem at hand by defining their custom servers.
- Just like eclipse, Carbon enables users to customize servers by adding or removing features at runtime.
- Through the kernel, components get support for cross cutting concerns (e.g. security) with minimal work, thus let users focus on main features of his component.
- First class support for User Interfaces as a part of components. (Written as JSP + Metadata)

How does it make a difference? (Contd.)

- Uniformity across the platform: same look and feel, startup, configurations, programming model etc. Hence if a user knows one server, he knows others.
- Kernel and components enable reuse, avoid repetition, and promote composition.
- Support platform wide aspects like a logging, governance, monitoring etc. And those aspects work consistently across the platform out of the box.
- As most features are drawn from the component repository, Carbon simplifies the development of new servers.
- Provides an OSGI based component framework for SOA developers

Few Further Steps

- Carbon supports multi- tenancy: Discussed in “Afkham Azeez, Srinath Perera, et. al., "Multi-Tenant SOA Middleware for Cloud Computing" 3rd International Conference on Cloud Computing, Florida, 2010”
- Complete Carbon platform is now available as a Service: (Stratos, see http://wso2.com/cloud/stratos/). Discussed in “Afkham Azeez, Srinath Perera, et. al., WSO2 Stratos: An Industrial Stack to Support Cloud Computing”, submitted to IT: Methods and Applications of Informatics and Information Technology Journal.

Open Questions/Challenges

- Investigate in to “Kernel” of the SOA platform, add remove features to get the right mix.
- UI Framework and integration with components (e.g. Single Sign on Support)
- Optimizing Carbon programming model
- CApp – Carbon archive format, which enable users to write a one archive that will include many SOA artifacts like Web Services, Workflows etc. (like EAR for J2EE)
- Carbon Studio – an integrated development environment for Carbon.

Few Places Where Carbon is in Use

Deutsche Bank
Alcatel·Lucent
Prudential
ca Transforming IT Management.
ebay
Chapdelaine & Co. Leadership has its advantages
Trimble
LOCKHEED MARTIN
National IT and Telecom Agency
Ministry of Science Technology and Innovation
software AG
Mercedes-Benz
BRITISH AIRWAYS
KAISER PERMANENTE
hp
concur TECHNOLOGIES
Intermountain Healthcare
FOX MOBILE

Conclusion

- We discussed Carbon Server Building Framework
  - Motivation
  - How it is build?
  - How does it make a difference?
- Currently Carbon is the core of WSO2 platform which includes 10+ servers and hundreds of deployments (70+ customers) .
- Everything is available under Apache open source license. All discussions public! We welcome your comments as well as contributions.

More Info

- Corporate website: http://wso2.com
- Developer portal: http://wso2.org

WSO2 OXYGENTANK

OXYGEN FOR WEB SERVICE DEVELOPERS

- Business development team: bizdev@wso2.com
- srinath@wso2.com