The Scenario

- For some projects it is convenient to partition functionality as separate apps (war files) e.g.
  - REST-ful services app
  - User facing GUI app
  - Admin GUI app
  - Batch processing/Quartz app
The Problem

- Want to share common set of domain classes, services and taglibs amongst these apps

App1  App2  App3  App4

Domain Classes Services & Taglibs
The Solution

- Package common domains, services and taglibs as a plugin
- Install plugin in each app
Example: IP Telecom

- Modular IP telecom systems are generally composed of more than one application
- An “application router” is used to compose apps
Example: IP Telecom

- Modular IP telecom systems are generally composed of more than one application
- An “application router” is used to compose apps
Example: IP Telecom

- Modular IP telecom systems are generally composed of more than one application
- An “application router” is used to compose apps
Example: IP Telecom

- Modular IP telecom systems are generally composed of more than one application
- An “application router” is used to compose apps
Example: IP Telecom

- Modular IP telecom systems are generally composed of more than one application
- An “application router” is used to compose apps
Example: IP Telecom

- Modular IP telecom systems are generally composed of more than one application

- An “application router” is used to compose apps
Converged Apps

• Each app includes a SIP servlet and, optionally, an HTTP servlet
• The Converge app development framework uses
  
  • E4SS (ECharts for SIP Servlets) – to define an app’s SIP servlet as an ECharts state machine
  
  • Grails – to define an app’s HTTP servlet and database schema
The ConvergeCore Plugin

- Converge apps use a plugin that provides access to Converge core domain classes, services and taglibs
Recurse!

- A Converge app consists of two components:
  - A domain plugin including an app’s E4SS state machine, and defining its own domain classes and services
  - An optional application that uses the plugin e.g. an admin interface
Example: Converge Monitor Application

Monitor App

MonitorDomainPlugin  ConvergeCorePlugin

MonitorControlMachine
JavaToMonitorControlMachineService
MonitorControlMachineToJavaService
ConvergeMonitorFeatureInstanceData

FeatureOperationalDataService
AbstractUser, Individual, UserGroup,
Address, UserAddress, PatternAddress,
Role, CallRecord, Feature,
FeatureSubscription, FeatureData,
FeatureAddressData, FeatureInstanceData, etc

Monday, May 11, 2009
Reusing App Plugins

- An app can use another app’s domain plugin e.g. to:
  - Offer the app as a web service
  - Provide an integrated UI for a set of apps
Building

- Challenges when plugins undergo development at same time as apps:
  - must build and package plugins prior to building apps that depend on them
  - a plugin should only be (re-)installed if an app depends on it and the plugin was modified
Grails Build Support

- Grails does not consider plugin source code mods as part of its build so must use a build tool for this e.g. Ivy, Maven, custom Gant script

- Grails plugins can include ‘dependsOn’ constraint to reference plugins but doesn’t help unless using 1.1 and plugins located in a repository – easier to use a build tool
Mega-Wars!

• An “empty” Grails war file weighs in at 21.6 MB

• Nice to have container support for sharing Grails jars and core plugin jars when deploying multiple apps

• SailFin/GlassFish provide ‘--libraries’ deployment option
Summary

- Grails plugins are an effective way to structure a project consisting of applications that share the same domain classes, services and taglibs.

- Building a project requires a build system to manage plugin source code dependencies.

- Deploying a project requires container support for sharing libraries across apps.
Converged Containers

- The following open source containers support both SIP and HTTP servlets:
  - SailFin (extends GlassFish)
    - http://sailfin.dev.java.net/
  - Mobicents (extends Tomcat or JBoss)
    - http://www.mobicents.org/
  - Cipango (extends Jetty)
    - http://www.cipango.org/
For More Info

• Check out echarts.org for:
  • E4SS development kit – open source
  • Converge development kit (to be released this month) – open source

• Talk by me and Tom Smith at JavaOne: “Web 2.0 Phone Home: Rapid Development of Telecom-Enabled Web Applications”