



Java is a trademark of Sun Microsystems, Inc.

# JavaOne™

## Building Facebook & OpenSocial Applications with Java™ Technology

Richard Pack

Chief Technology Officer



# Goal

- > Learn Why Java Technologies are Perfect for a Social Application
- > Learn How Use These Technologies to Build a “Universal” Social Application

# About Me

- > 2nd Time Speaking at JavaOne
- > 10 Years of Software Development Experience
  - ☑ 8 in Enterprise Software
- > Built Social Applications for Multiple Platforms

# Prerequisites

- > Parsing XML
- > HTML and Javascript
- > DWR and JSON
- > Spring Beans

# Agenda

- > Why Social Applications are Important
- > Application Architectures
- > Write Once, Deploy Your Application Everywhere

# Agenda





- > **Why Social Applications are Important**

Application Architectures

Write Once, Deploy Your Application Everywhere

# Why is this Important?

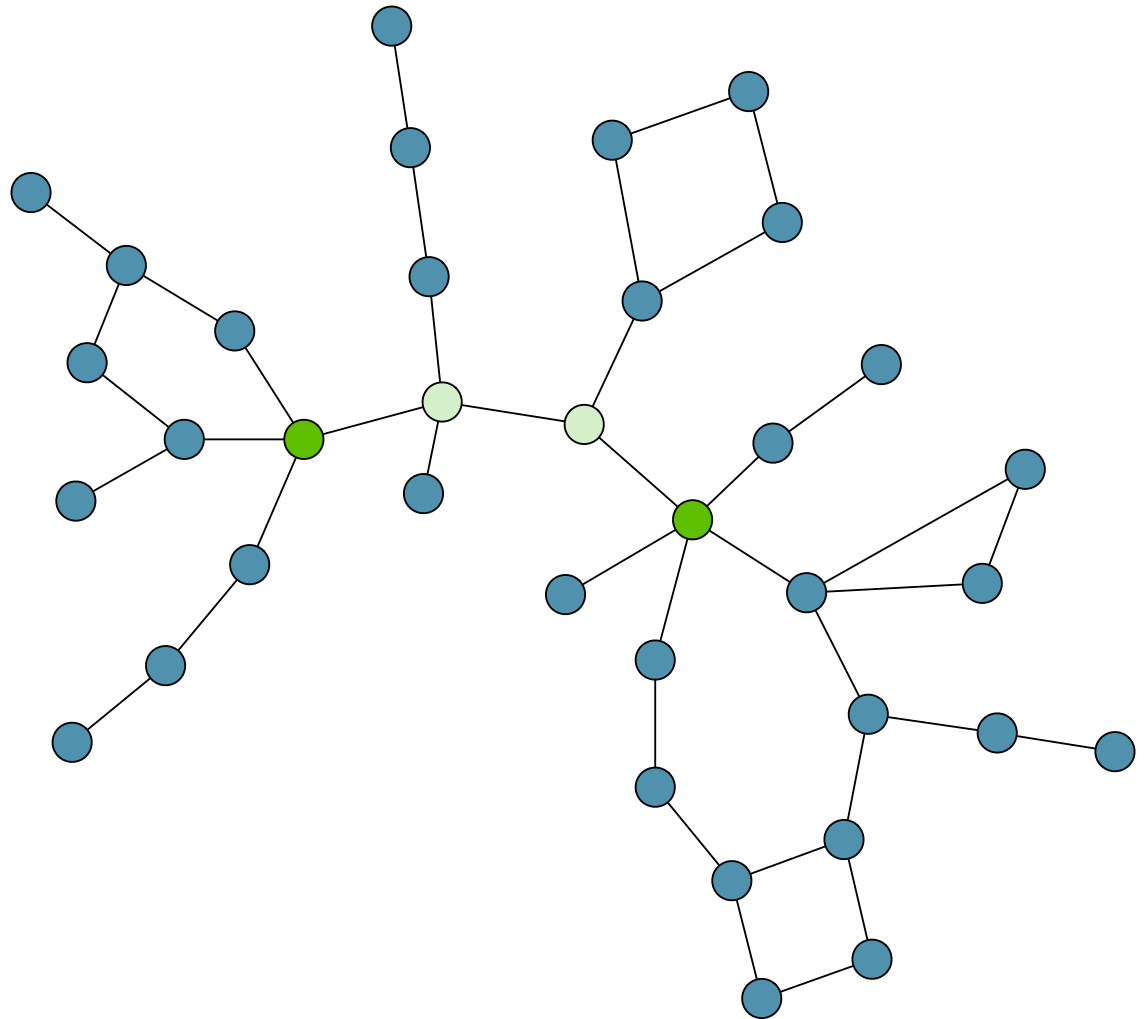
# Why is this Important?

- > The Audience - 550M+ Potential Users
- > The Interactions - Exercising the connections
- > The Data
  -  Extracting Interests and Intent
  -  Behavioral Data
  -  Relationships and Affinity Groups
  -  Location Data




# Why is this Important?

- > Relationships
- > Members
- > Affiliations
- > Messages
- > Profile Info



# The Social Application Containers

>  Facebook Application Platform

>  OpenSocial Container

 35 Containers, 22 of which are live

51.com	goo Home	LinkedIn	OpenNetworx	Viadeo
BIGADDA	hi5	lokalisten	orkut	Webjam
CityIN	Hyves	Mail.ru	phpFox	Webon
FanBox	IDtail	minglebox	Plaxo Pulse	XiaoNei
Freebar	iGoogle	MySpace	QuePasa	Yahoo!
Friendster	imeem	Netlog	Socialtext	YiQi
Gmail	iWiW	Ning	Tianya	Yonja

# Agenda

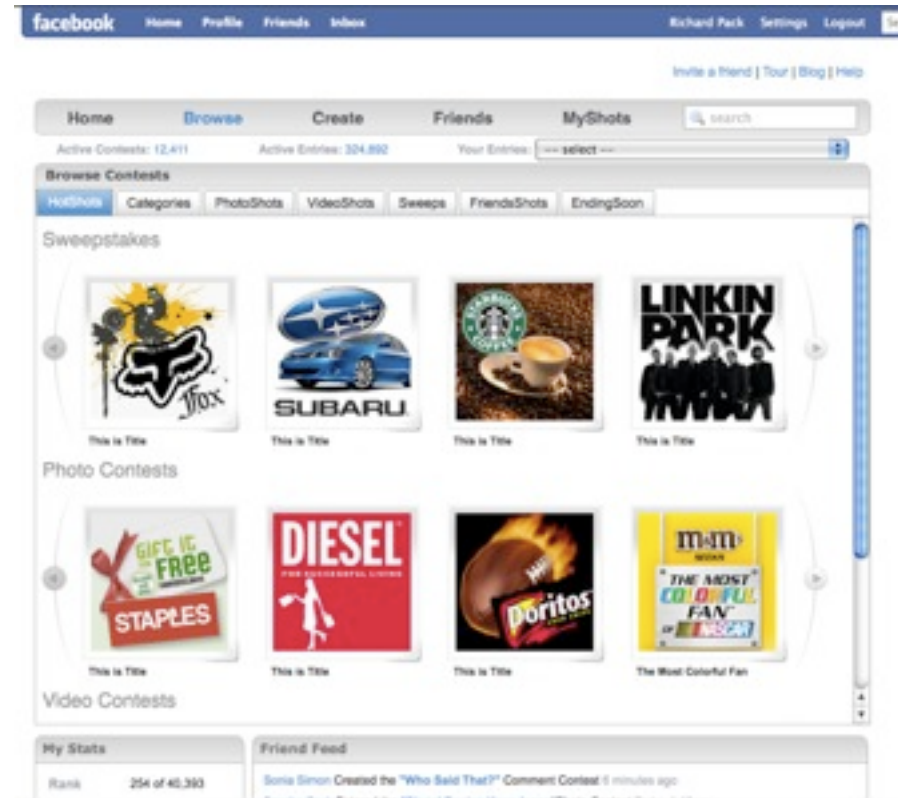
Why Social Applications are Important

> **Application Architectures**

Write Once, Deploy Your Application Everywhere

# What is a Facebook Application?



- > UI that is URL Addressable
- > Surrounded by Facebook Chrome
- > Has Access to Profile Data
- > Multiple Views







# Facebook: What Data is Available

Data		Actionable
Comments	Notes	Events
Events	Notifications	Feed
Feed	Photos	Links
Friends	Profiles	Notes
Groups	Status	Notifications
Links	Users	Photos / Video
Messages	Video	Status





# Facebook: Type of Applications

- > Two Types:
  -  IFrame - HTML with optional XFBML
  -  Facebook Markup Language (FBML) - HTML with FBML
  
- > Applications have Javascript and CSS on the Client and an XML Rest API on the Server (DWR).

# Definitions

- > Social Application API - a Rest style or Javascript interface to the container's data.
  -  Facebook
    -  Admin, Auth, Friends, Events, Feeds, Notifications, Photos, Profile, Status & Users
  -  OpenSocial
    -  Activities, Gadgets, Invitations, Messages, Profile & Person
- > Service API - internal to your system, provides a Rest service for access to data from the UI.

# Facebook: Technology

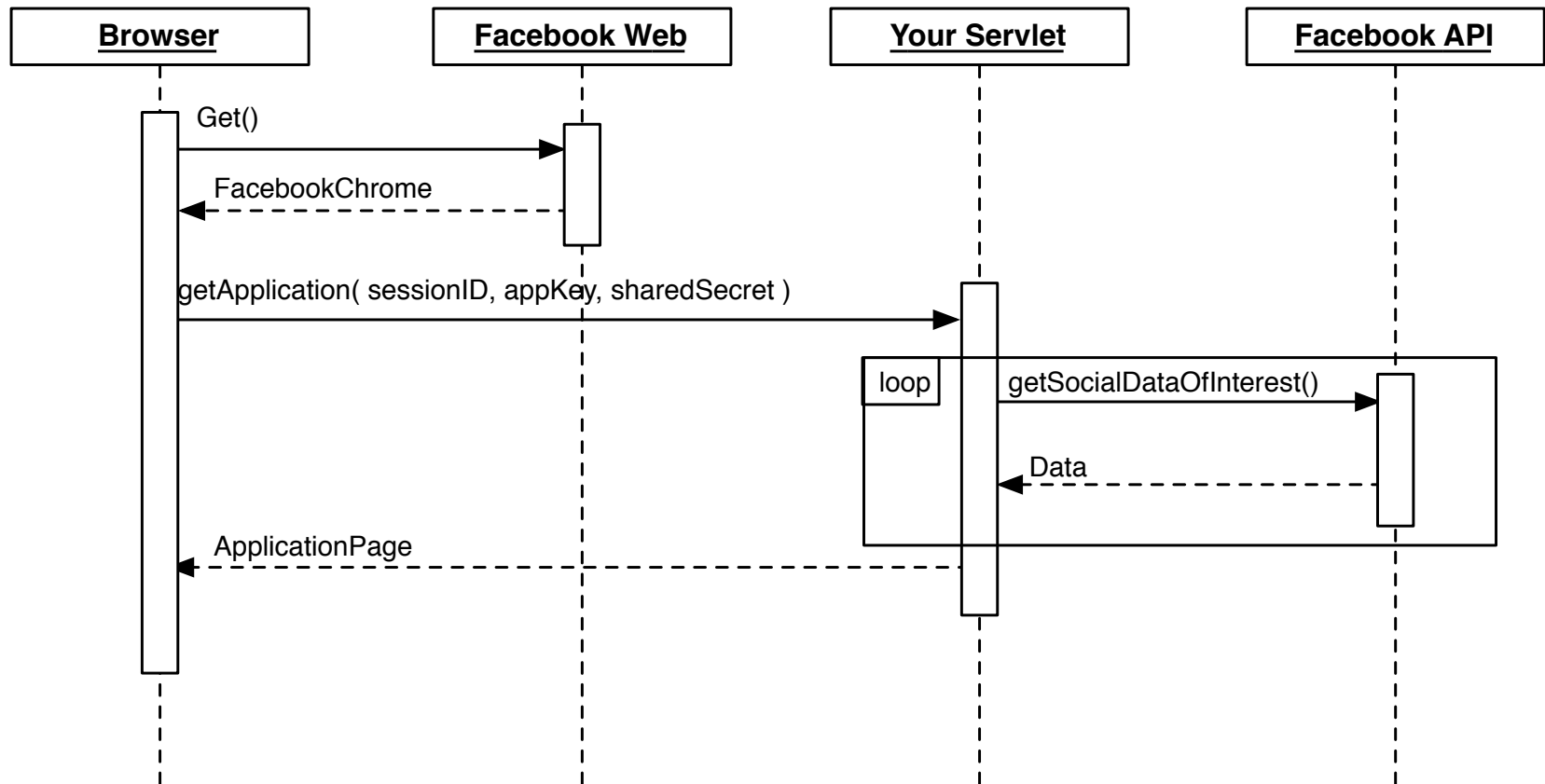
- > Integration with the Facebook API
  -  Facebook-Java-API
  -  DOM4J/XML Beans
  
- > Creating your own Rest Service interface to Spring Beans
  -  DWR
  
- > Building the User Interface
  -  HTML and Javascript or GWT/GXT



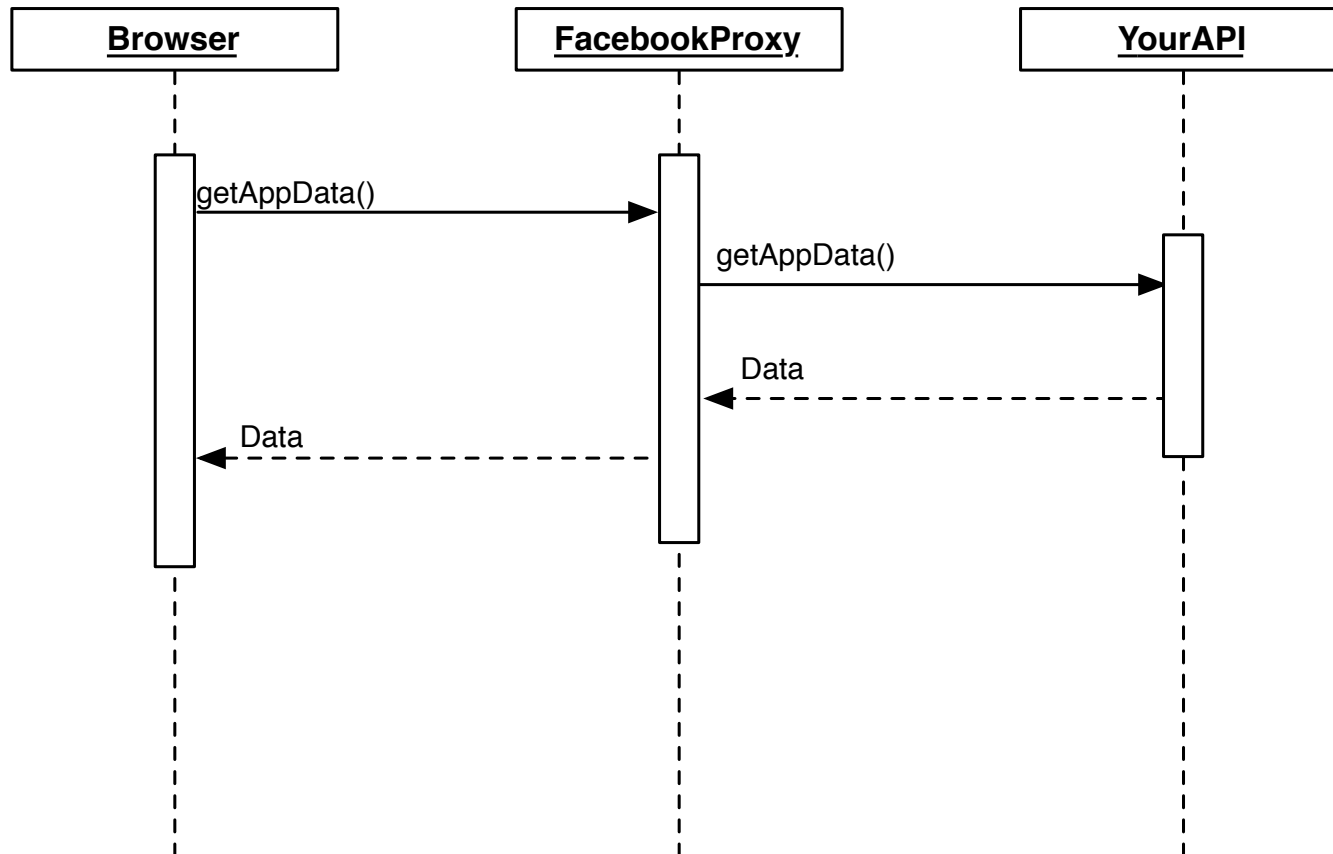
# Facebook: iFrame Stack

XFBML / HTML	Javascript	FBJS
DWR		FB API Servlet
Spring		
iBatis/SQLMaps/Hibernate		

# Facebook: iFrame Render Sequence

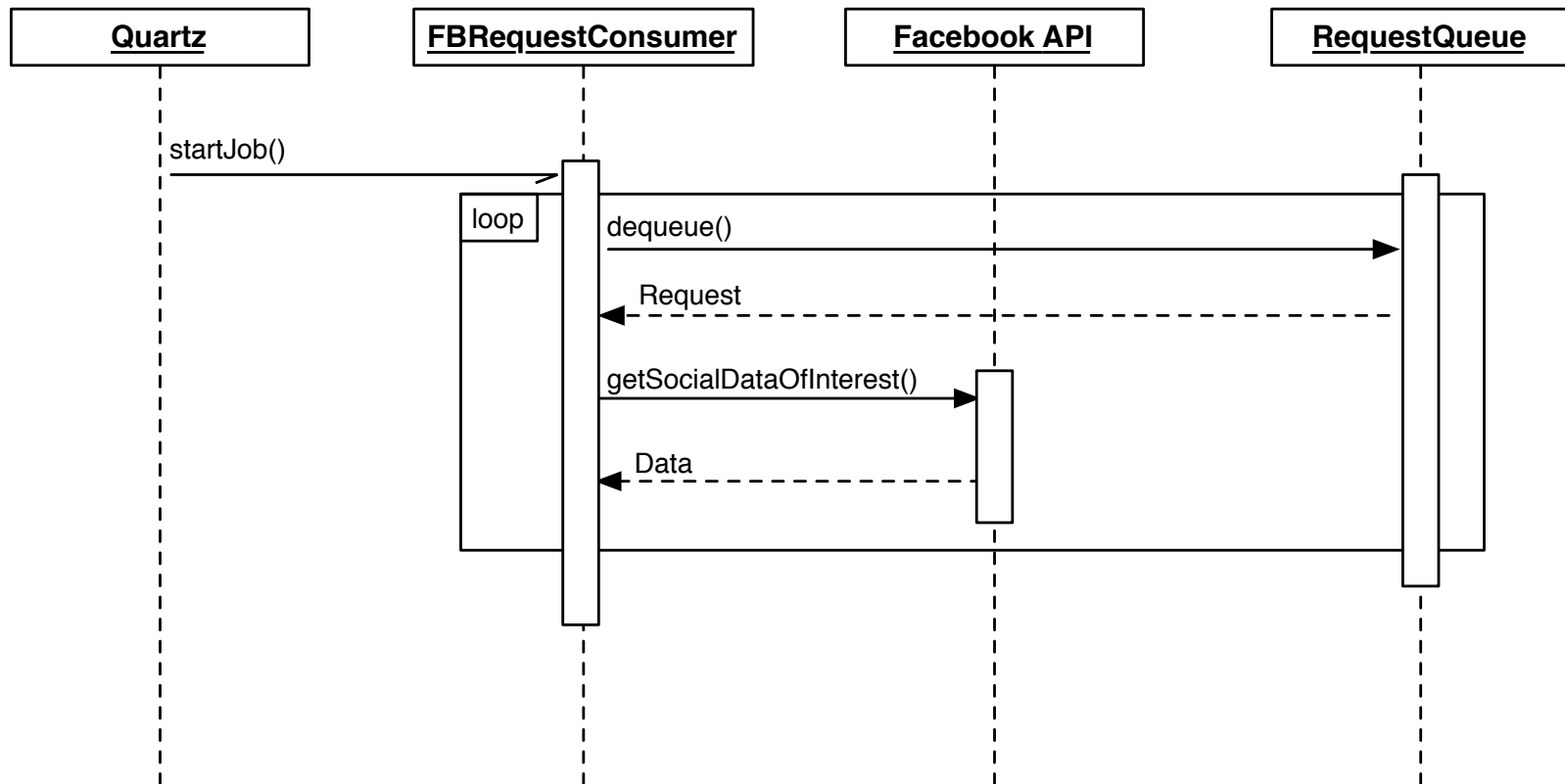


# Facebook: Your API Calls in FBML

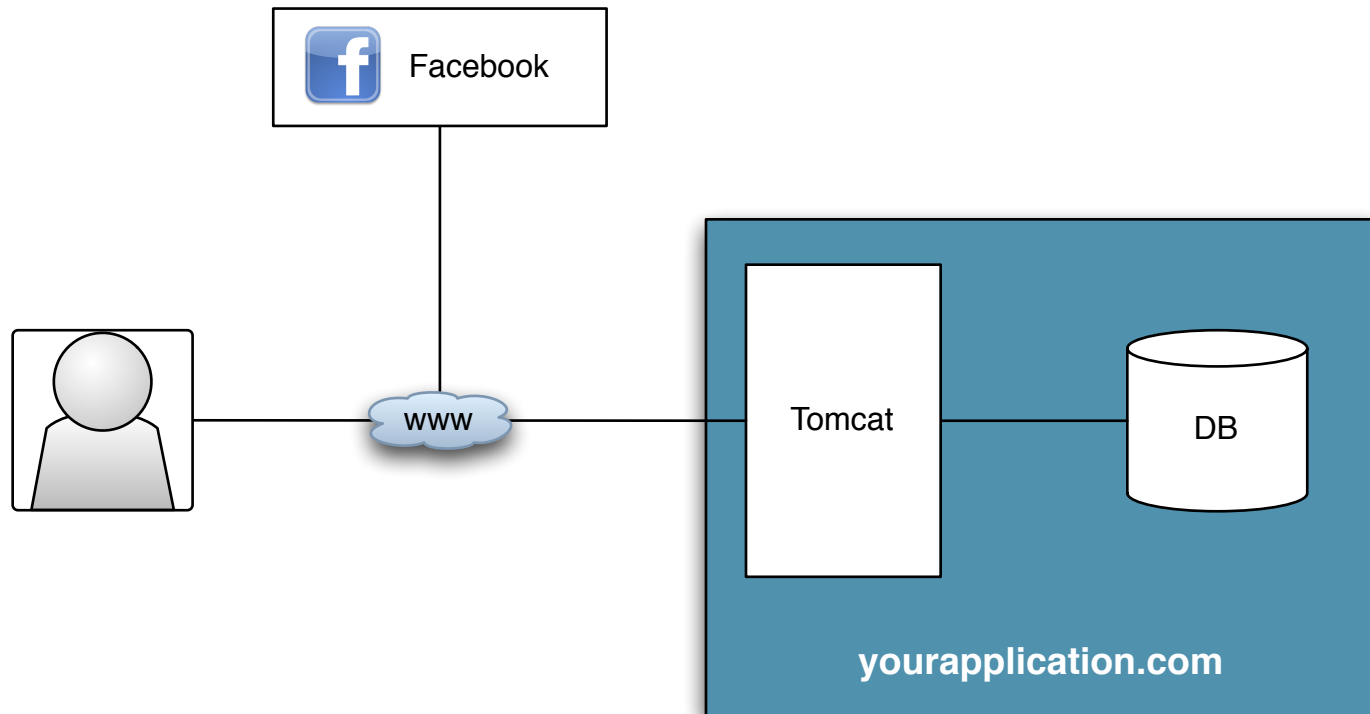


Note: All XHRs are proxied through Facebook for FBML Applications


# Facebook: Offline FB API Requests



# Facebook: Application Deployment



# Facebook: Construction Steps





- > Build your HTML UI
  -  Identify the Data the you need
- > Build your Social API interface
  -  Gather the Data you need
- > Build your Service API
  -  Provide the UI with the data gathered
- > Deploy the Application
- > Configure the App on the Facebook Dev Page

# OpenSocial: What is an OS Gadget?

- > URL Addressable Module and Resources
- > Surrounded By Chrome
- > Has access to the Local and Remote APIs
- > Multiple Views



# OpenSocial: What is an OS Gadget?

- > Bundle of XML that contains
  -  HTML
  -  Inline CSS and JS
  -  External CSS and JS references
  -  Configuration



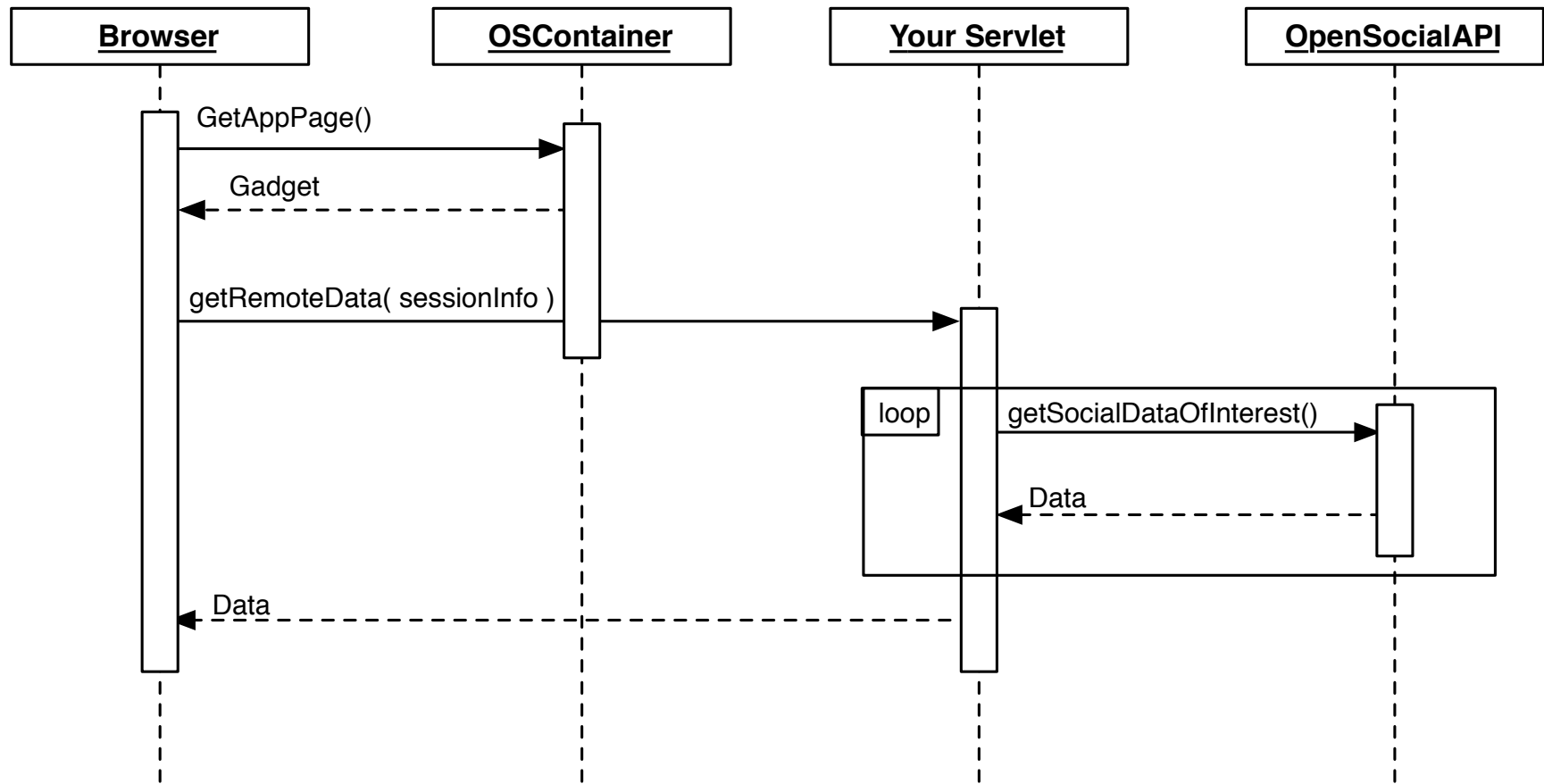
# OpenSocial: More Data

Data	Actionable
Activities	Activities
Friends	Data
Message	Messages
Organizations	Presence
People	
Person Details	
Presence	

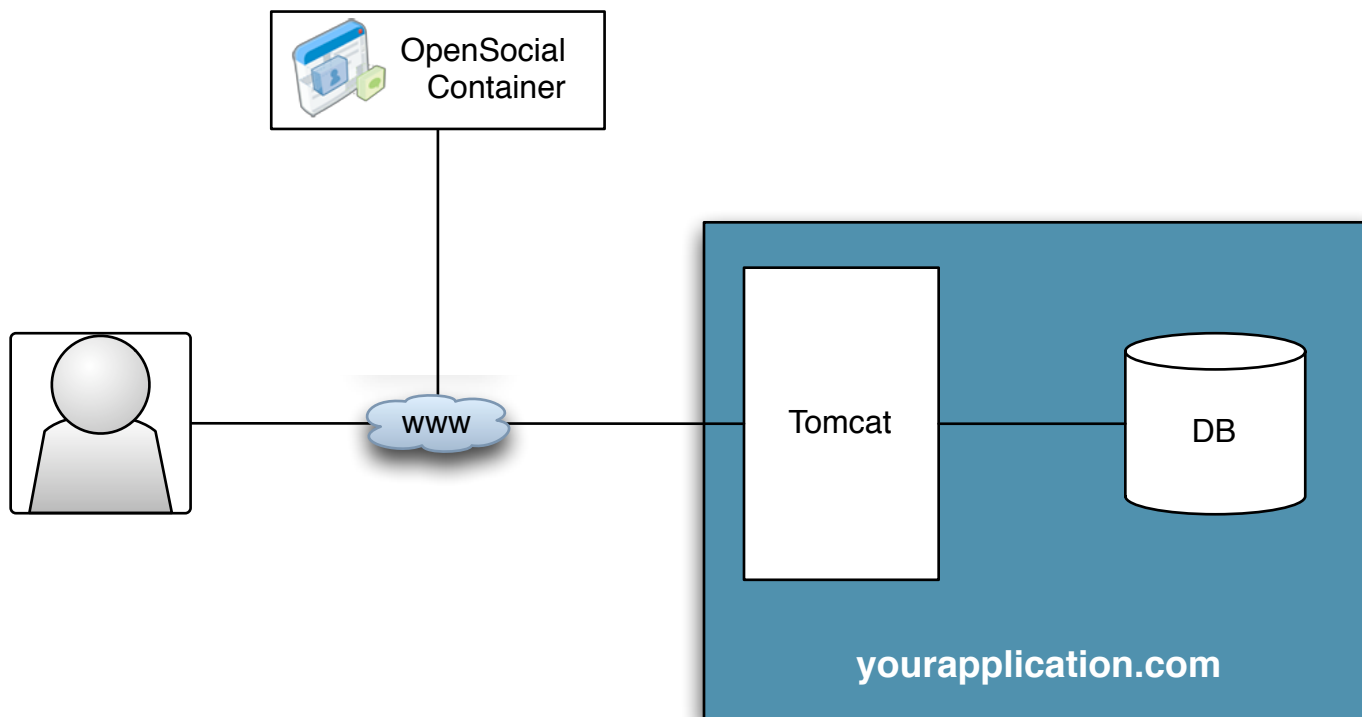
# OpenSocial: Stack

HTML	Javascript	XML
DWR	OS API Servlet	
Spring		
iBatis/SQLMaps/Hibernate		




# OpenSocial: Render Sequence



# OpenSocial: Application Deployment



# OpenSocial: Construction Steps

- > Build your HTML UI and define your Module
  -  Identify the Data the you need
- > Build your Social API interface
  -  Gather the Data you need
- > Build your Rest Service
  -  Provide the data gathered to your UI
- > Deploy the Application
- > Register the Application with the Containers


# Agenda

Why Social Applications are Important

Application Architectures

- > **Write Once, Deploy Your Application Everywhere**

# Convergence

- > Goal: To create a single “Universal” Application
- > So why choose iFrame
  -  Reuse for other social applications besides Facebook

# Convergence: Technologies

## > APIs:

- ☒ Facebook APIs

  - ☒ XML Beans

- ☒ OpenSocial

  - ☒ XML Beans

## > Server:

- ☒ Spring - services

- ☒ DWR - Javascript interfaces to service methods

- ☒ Services exposed with JSON

## > Client:

- ☒ GWT

- ☒ JS Widget Libraries

- ☒ FBJS/OSJS



# Convergence: The Stack



HTML	Javascript	FBJS	XML
DWR		RestService	
Spring	Universal API Servlet		
iBatis/SQLMaps/Hibernate			

# Convergence: Universal API

## > Build a common interface

- ☐ Leave the specifics up to the implementing classes
- ☐ Ex. `getFriends(UUID)` returns `List<Person>` of the hosting container

# Convergence: Rest API

- > Provides your application data to the UI
  -  Use DWR or JSON via Servlet/Filter
  -  Expose the Spring Services

# Convergence: Object and Data Model

## > Handling the Containers Data

- ☒ Build an Java object model (POJOs)
  - ☒ Ensure that the structure makes sense for your application
- ☒ Persistence Model
  - ☒ Ensure that the model can handle the load of the application. Ex. Separate model data and metric data

# Convergence: Testing

- > OpenSocial

-  Shindig

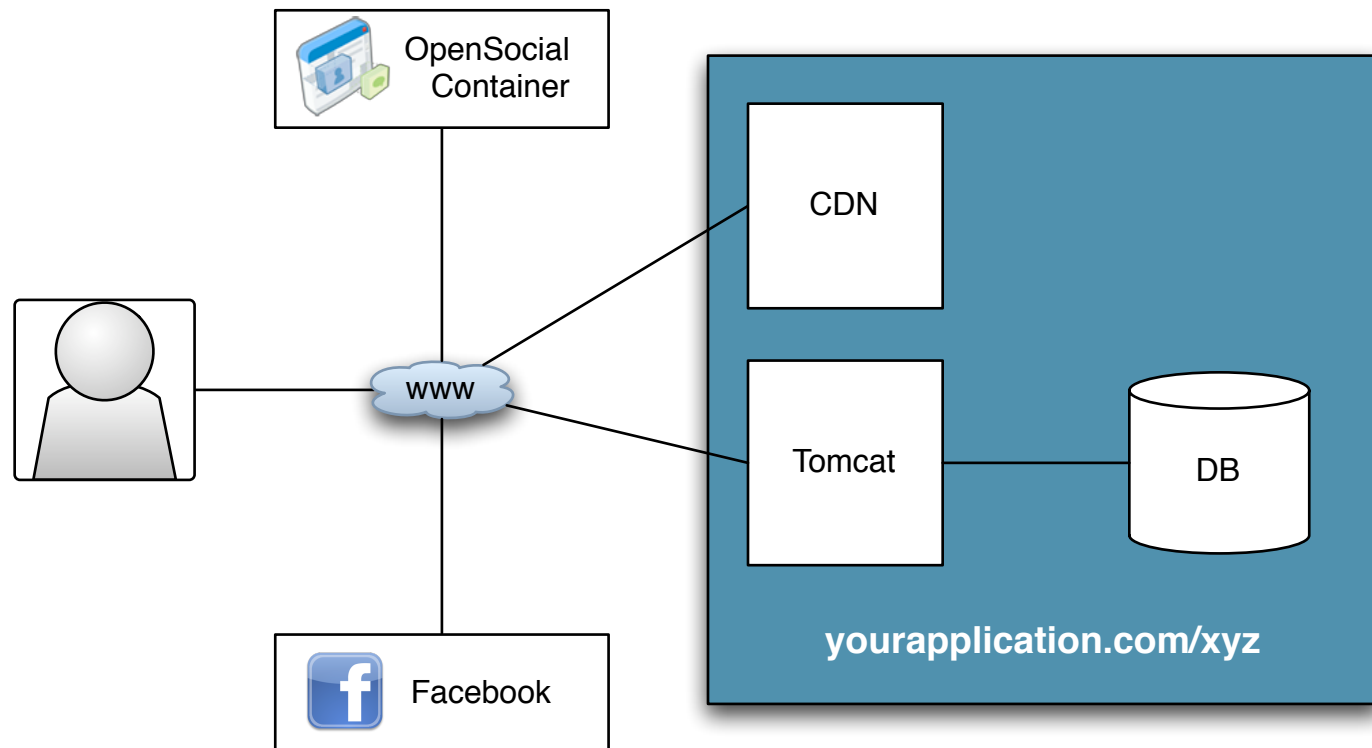
- > Facebook

-  Facebook Tools


- > Building a Test Container

# Convergence: Deployment

- > Consider Using the Cloud and a Content Delivery Network (CDN)



# Convergence: Example

- > Download:
  -  [developer.girnshot.com/javaone/example.zip](http://developer.girnshot.com/javaone/example.zip)
- > Create an EC2 Tomcat/Linux AMI
- > Remote deploy the WAR
- > Register the example App with Facebook and a OpenSocial container
- > URL: `http://<elasticip>/example/`

# Questions?

## More Information

- > OpenSocial
  -  [opensocial.org](http://opensocial.org)
- > Facebook
  -  [developers.facebook.com](http://developers.facebook.com)
- > Apache Shindig
  -  [incubator.apache.org/shindig/](http://incubator.apache.org/shindig/)
- > opensocial-java-client
  -  [opensocial-java-client.googlecode.com/](http://opensocial-java-client.googlecode.com/)
- > DOM4J
  -  [dom4j.org/](http://dom4j.org/)
- > Facebook-Java-Apifacebook-java-api
  -  [code.google.com/p/facebook-java-api/](http://code.google.com/p/facebook-java-api/)
- > DWR
  -  [directwebremoting.org/](http://directwebremoting.org/)





# JavaOne™

# Thank You

Richard Pack  
richard@grinshot.com

