



facebook.



From Prototype to Success:

Applications for Facebook and OpenSocial
Containers

Steve Chan, AppTsunami
Dave Nielsen, Platform D

Level Setting

- This session is NOT:
 - How to market your application
- This session IS:
 - How to build an application quickly that will grow with you as you expand to other social networks

About Us

Met at an event like this

Create Bracket Madness

Ported it to Hi5

Social Network Apps:

- BracketMadness
- TheFabulousLife (beta)
- Share-A-Ride (in progress)
- RealBucks (in progress)

Emergence of Social Networks

- 5 of the top 10 sites tracked by Alexa are social networks
- Reaching 275 million people (world-wide)
- Over 20,000 applications on Facebook
- > 20,000 OpenSocial developers

OpenSocial Containers

Ning

plaxo

friendster

XING

hi5

orkut^{beta} viadeo

six apart

salesforce.com
Success On Demand™

LinkedIn ORACLE

mixi
mixi, Inc.

天際網
www.tianji.com

ENGAGE

bebo

imeem

Hyves.net
always in touch with your friends

myspace.com
a place for friends

myAOLtm
BETA

MY YAHOO![®] BETA

fb Promise/Reality

Promise

- you can build apps quickly
- lots of users to try it

Lesson Learned

- Building your first Facebook app is not trivial ...

fb Lessons Learned

- FBJS is not JavaScript
- Facebook AJAX is Mock AJAX
- FQL is not SQL
- Predefined Layout Style
 - Color, Layout, Buttons
 - No documentation

os Promise/Reality

Promise

- can build apps quickly
- apps run on multiple social networks

Lesson Learned

- Building your first OpenSocial app is not trivial
- Each OS implementation is different

os Lessons Learned

- JavaScript app model is difficult
- Different Versions (.6 .7 .8)
- Different Features (Invite friends)
- Different Policies (Invite 15 friends)
- Different Metadata (business friends)

Cross Platform Apps

- Porting apps from Facebook to OpenSocial is no picnic
- Individual OpenSocial Containers are still an issue
- Differences between two models

Necessity: Mother of Innovation

- Abstracted reusable code into a *framework*
 - Application developer calls methods in the framework which emits Facebook code
 - Application developers can focus on the application logic

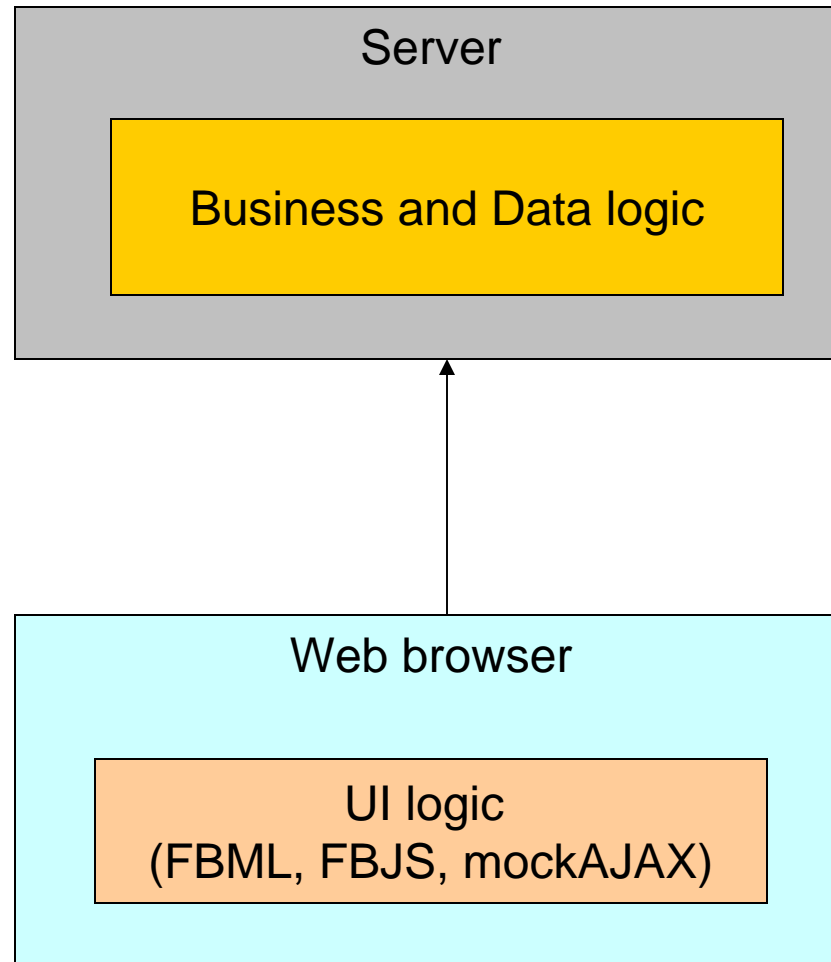
x-platform Lessons Learned

- Start with Facebook
- Next, rewrite the backend of the framework to emit OpenSocial code
- Now the framework makes the application code single-source (almost)

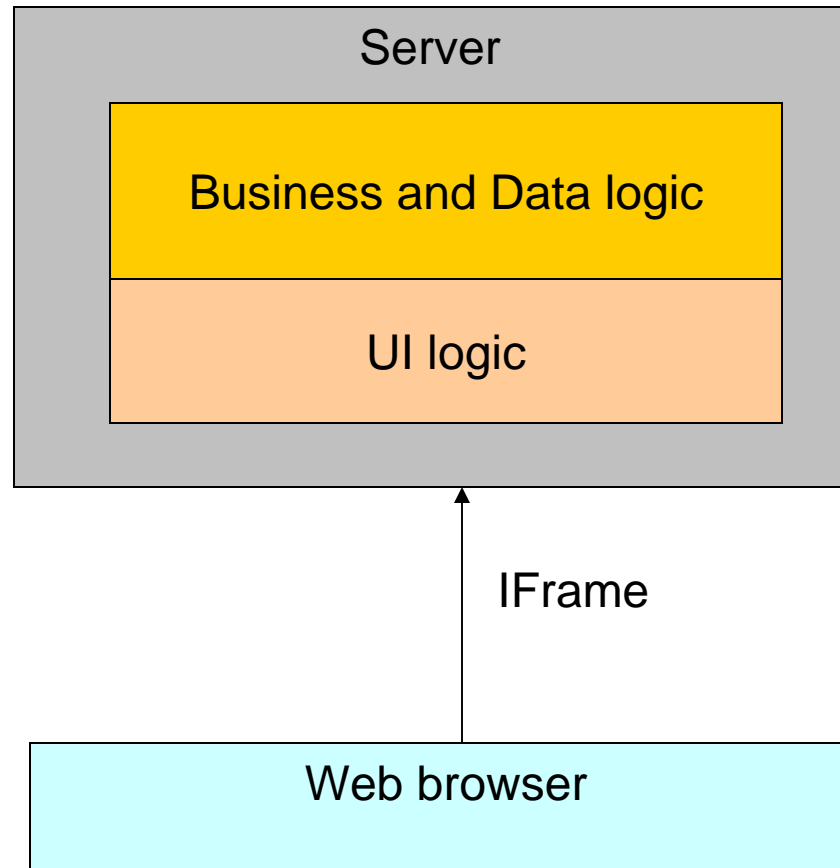
How

- Abstraction and Inheritance
 - Abstract the common container features
 - Create Reference Classes
 - Inherit and extend for each container

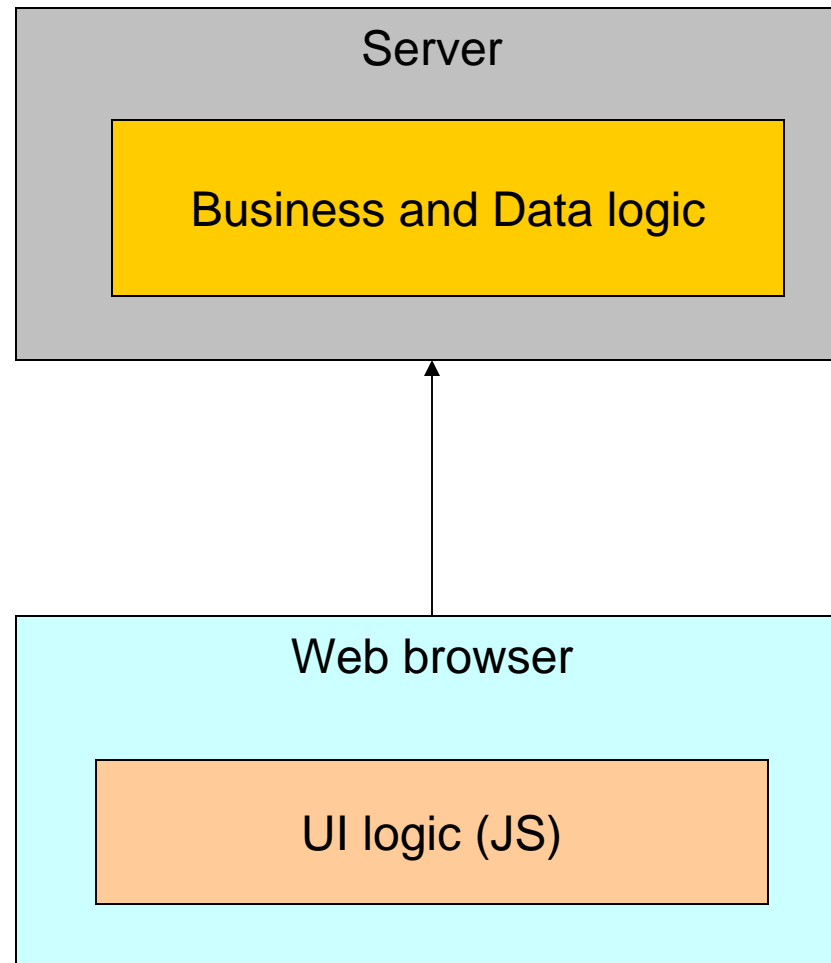
Facebook Model 1



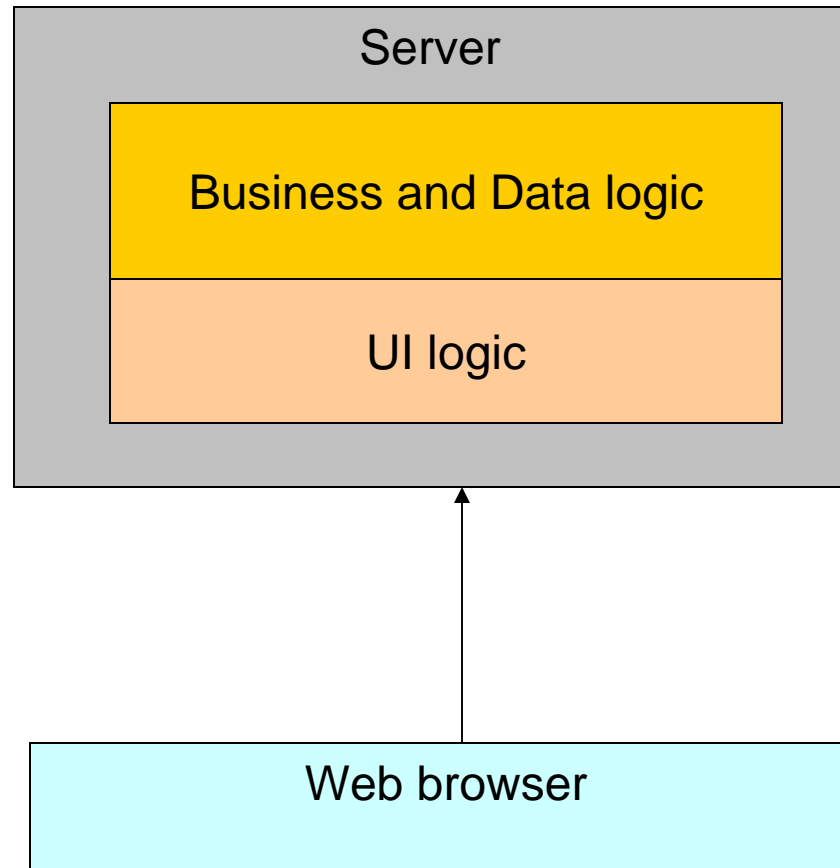
Facebook Model 2



OpenSocial Model 1



OpenSocial Model 2



Subsystems

MVC

Application Logic

User interface classes

Social network access
API

Database classes

Facebook

Open Social

Openface Architecture

- Single entry point called by the platform
 - http://your_host/your_app/openface/php/index.php
 - Facebook: Type in to 'Edit Settings'
 - Open Social: defined in http://your_host/your_app/xml/application.xml
- Index.php calls an application configuration file (TfwApplication.php) to render the top-level 'UI frame' object

Views: UI classes

- TfwFrame
 - TfwFrameSingleCanvas.php
 - TfwFrameMultiCanvas.php
 - TfwFrameMultiCanvas2.php
- TfwCanvas
 - __construct()
 - getLabel()
 - getTag()
 - getIcon()
 - render()
- TfwPortlet
 - __construct()
 - loadData()
 - render()
- TfwPanel
 - __construct()
 - loadData()
 - render()

More UI Classes

- actions
 - TfwActions.php
- callbacks
 - TfwCallbacks.php
- dialogs
 - TfwDialogs.php

models: Platform Classes

- TfwDataSource
 - URL-generation
- TfwDialogs
 - Invite dialogs
- helpers
 - JavaScript generation helpers
 - HTML generation helpers

models: Database classes

- TfwDataTable
 - Provides commonly used functions on MySQL to make writing portlets easy
- TfwDataConnect
 - Database connection methods

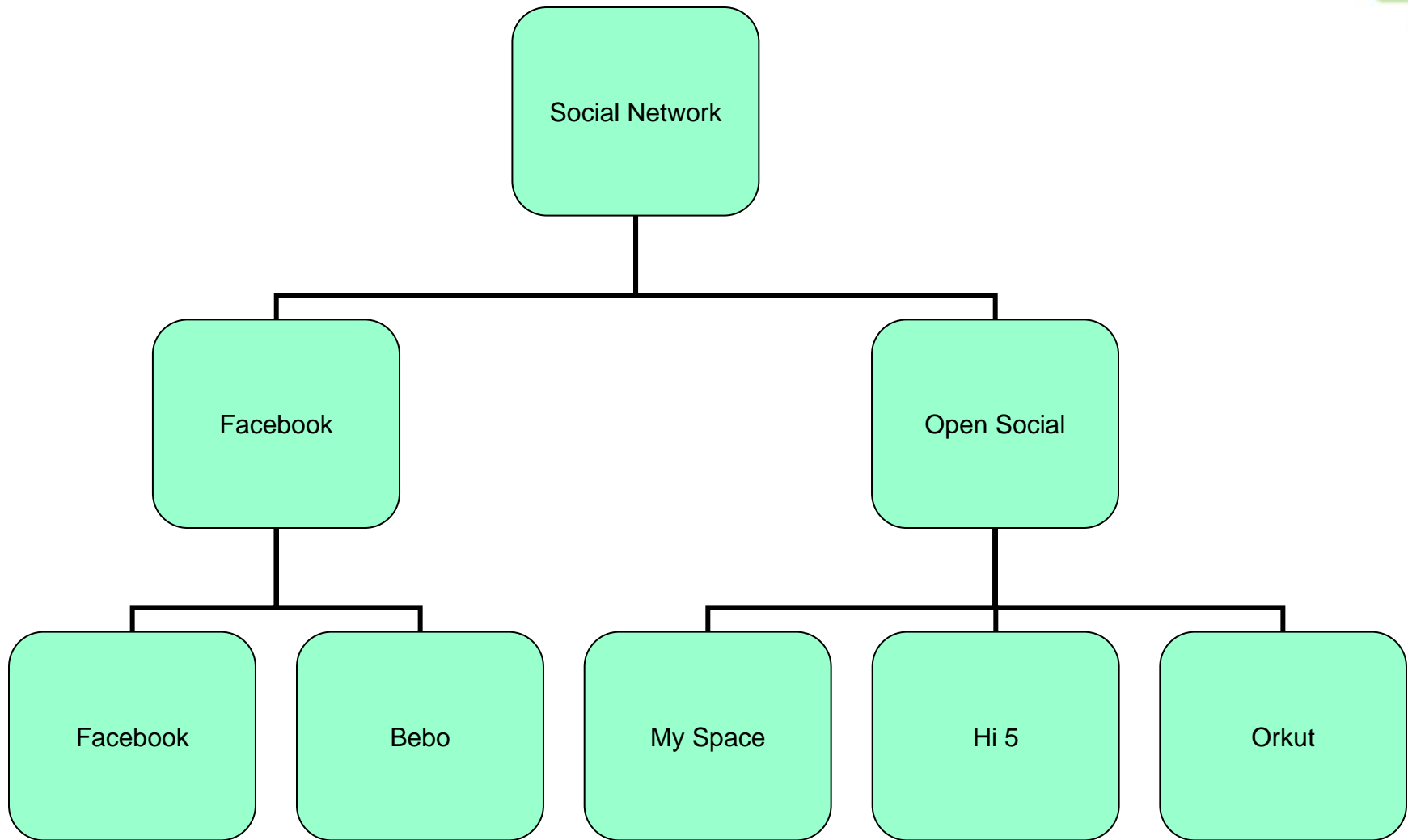
Value Add

- CRM
 - A/B Testing
 - Monetization
- Mashup
 - Google API
- Game
 - Virtual economy

What

- MVC
- Single Point of Entry / Controller
- 4 Levels of User Interface objects
 - Frame
 - Canvas
 - Portlets & Panels
 - Controls & Content

Future: More Platforms



Future: Languages & Platforms

- Ruby
- Java
- .Net
- iPhone?
- Android?

code.google.com/p/openface/

Steve - schan@apptsunami.com

Dave – dave@platformd.com

