COMBINING SENCHA TOUCH AND EXTJS IN YOUR PROJECT

A Journey

Bilal Soylu

Sourc{ 2012, London

Agenda

- About Me
- The Idea
- Reviewing MVC
- The Journey
- How to take this further
- QA

About Me

- CTO Verian Technologies (www.verian.com)
- Charlotte, NC, USA
- Sencha Charlotte User Group Manager
- Open Source Supporter and Contributor

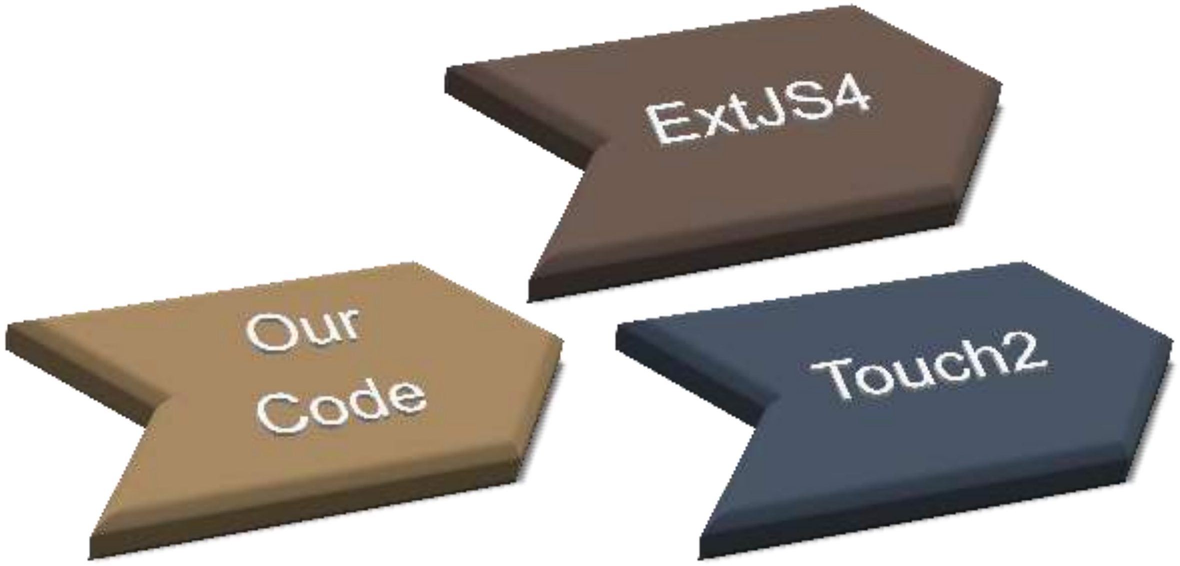
- @BmanClt
- http://BonCode.blogspot.com

- I like Dilbert

We all have abundant time and resources!

If not, it is simple to clone ourselves !

MY BOSS ASKED ME TO CLONE HIM. IS THAT ETHICAL?
YOU'LL EITHER CREATE A SOULLESS ABOMIN-ATION OR, IF THE CLONE IS NORMAL, YOU WILL HAVE SHOWN THAT SOULS ARE IRRELE-VANT.
WHAT IF THE ORIGINAL IS ALREADY A SOULLESS ABOMINATION?
YOU CAN BORROW MY CLONER. IT'S IN THE TRUCK.
www.dilbert.com scottadams@aol.com
© 2001 United Feature Syndicate, Inc.


ExtJS4
Our Code
Touch2

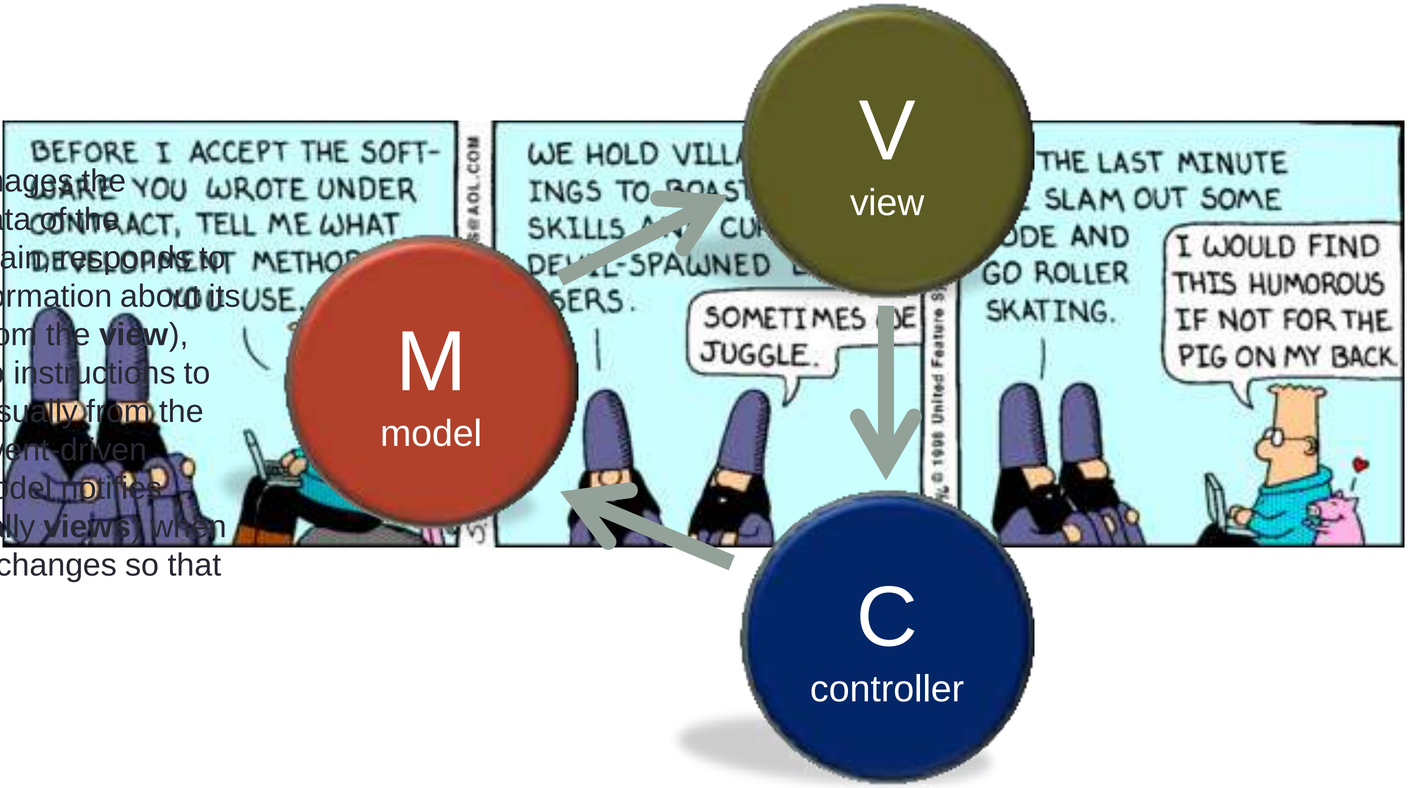
The Startpoint

- Ext code review
- MVC model introduction in Ext4 and Touch2
  - Appear many similarities that can be used
- What If ????
  - Could take the code and move it from Ext4 to Touch2
- Potential Benefits
  - Higher code reuse
  - Reduce maintenance
  - Reduce future rollout time
  - Reduce cost?
- Would there be other side benefits / problems?

The Framework

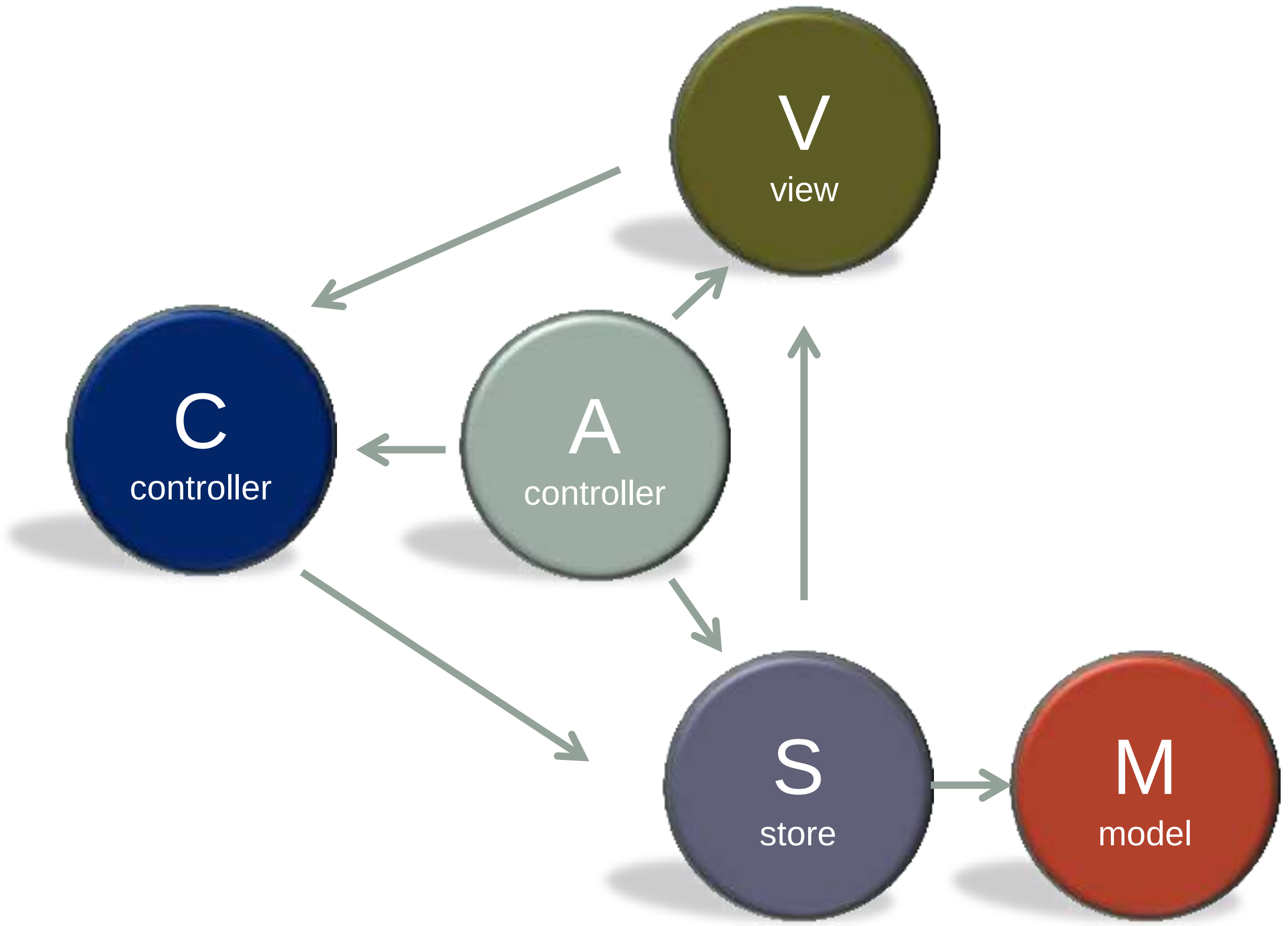
The ***View*** renders the model into a form suitable for interaction, typically a user interface element. Multiple views can exist for a single model for different purposes. A view port typically has a one to one correspondence with a display surface and knows how to render to it.

The ***Model*** manages the behavior and data of the application domain, responds to requests for information about its state (usually from the **view**), and responds to instructions to change state (usually from the controller). In event-driven systems, the model notifies observers (usually **views**) when the information changes so that they can react.

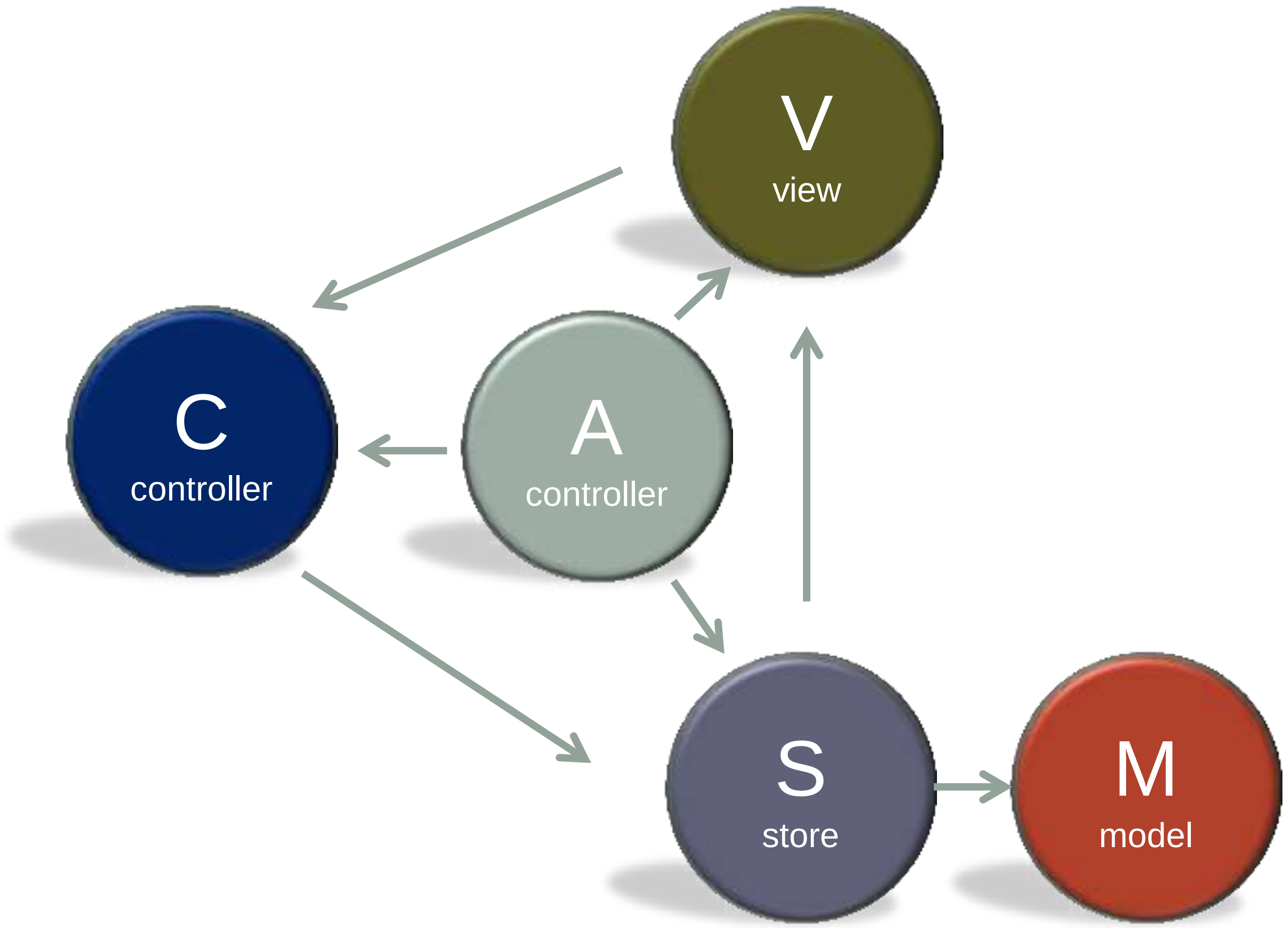

BEFORE I ACCEPT THE SOFTWARE YOU WROTE UNDER CONTRACT, TELL ME WHAT DEVELOPMENT METHOD YOU USE.
WE HOLD VILL INGS TO BOAST SKILLS CU DEVIL-SPAWNED SERS.
SOMETIMES WE JUGGLE.
THE LAST MINUTE SLAM OUT SOME CODE AND GO ROLLER SKATING.
I WOULD FIND THIS HUMOROUS IF NOT FOR THE PIG ON MY BACK.
V
view
M
model
C
controller

The ***Controller*** receives user input and initiates a response by making calls on model objects. A controller accepts input from the user and instructs the model and a view port to perform actions based on that input.

ExtJS 4 MVC

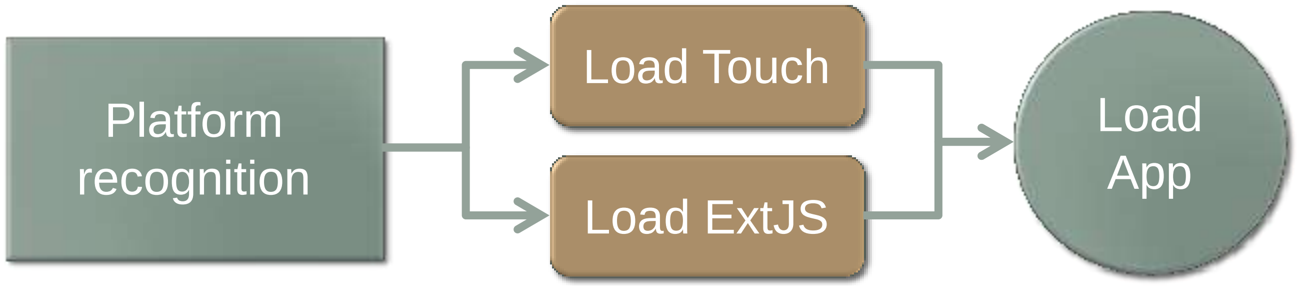


Sencha Touch 2 MVC



Approach 1: Let’s just write code

- Recognize platform
- Write code in MVC style using Sencha APIs
- Assume that we have good coverage of class and packages in both Sencha libraries



Platform Recognition

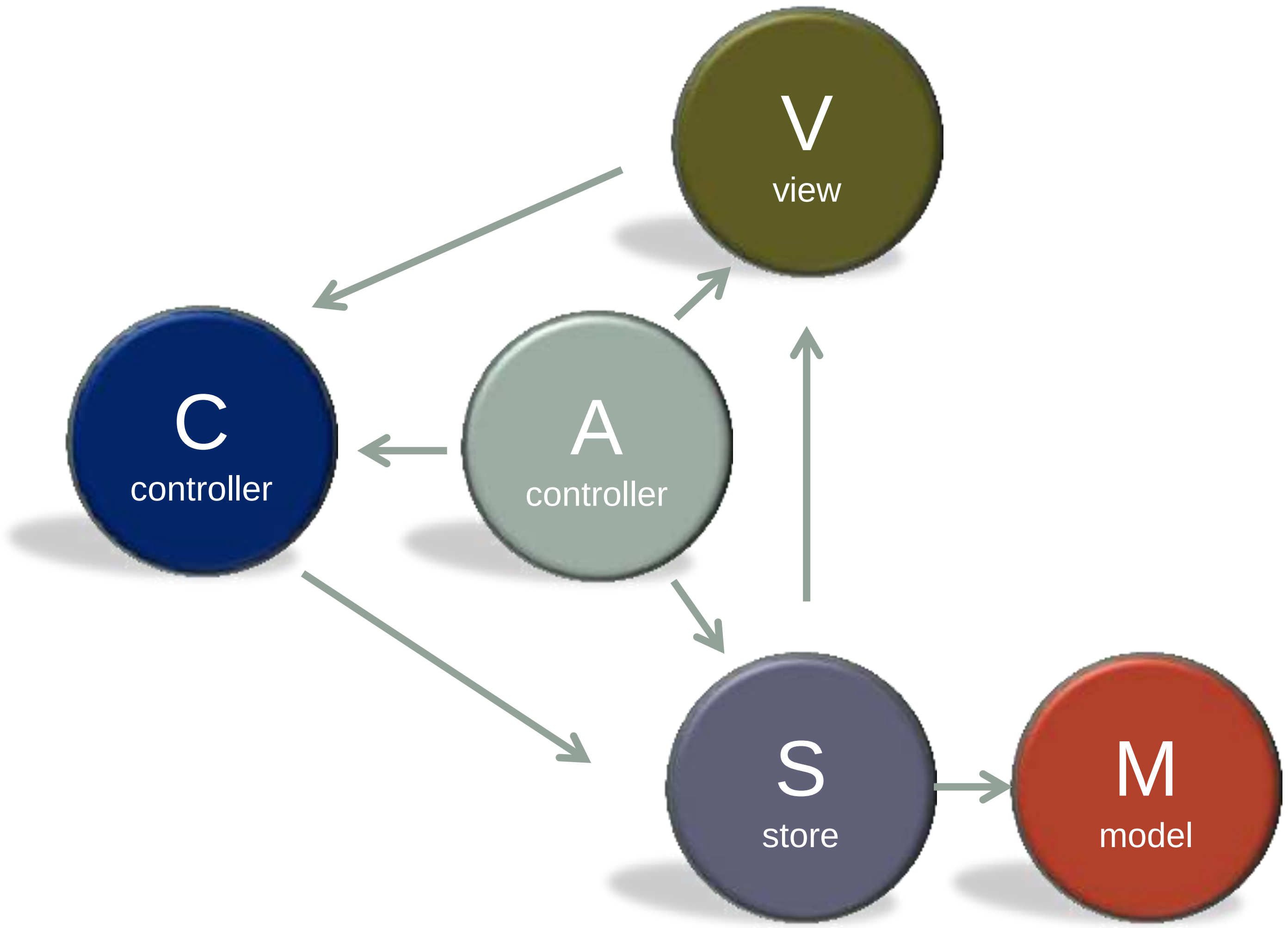
- Detect browser and deduce platform to load the correct library (ExtJS vs Touch)
- Can be performed by application server
- Can be a simple js initiator:

```
if (navigator.platform.indexOf("iP", 0) == 0) {
      txt+= "<hr><br>You are calling from iPhone or iPad";
      document.location.href="touchBooks";
} else {
      txt+= "<hr><br>Standard Browser";
      document.location.href="extBooks";
}
```

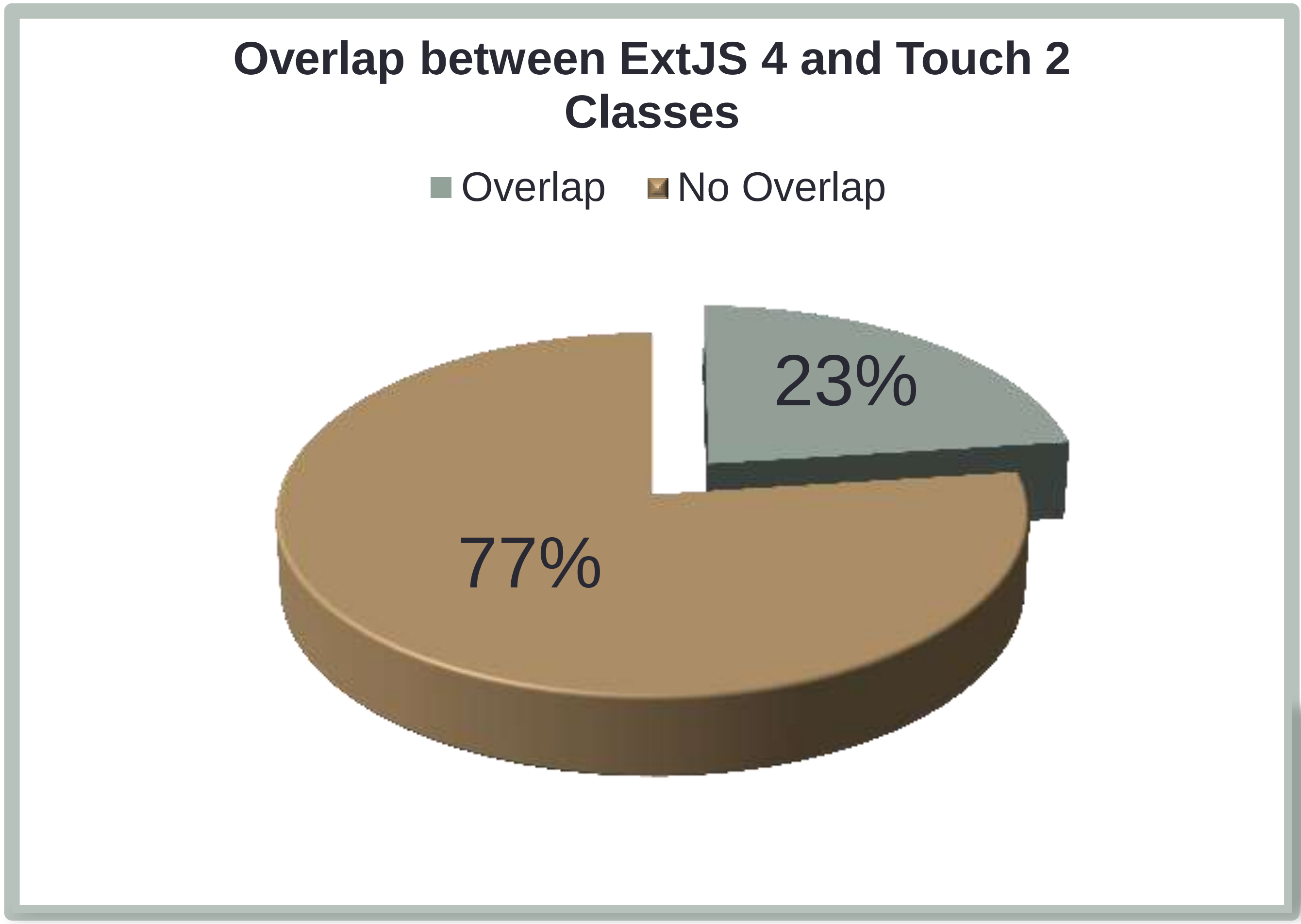
IS THIS ALL WE NEED?

1.

Reusable ?



Using out of the box APIs? Really?



Which packages can be used in both (with care)

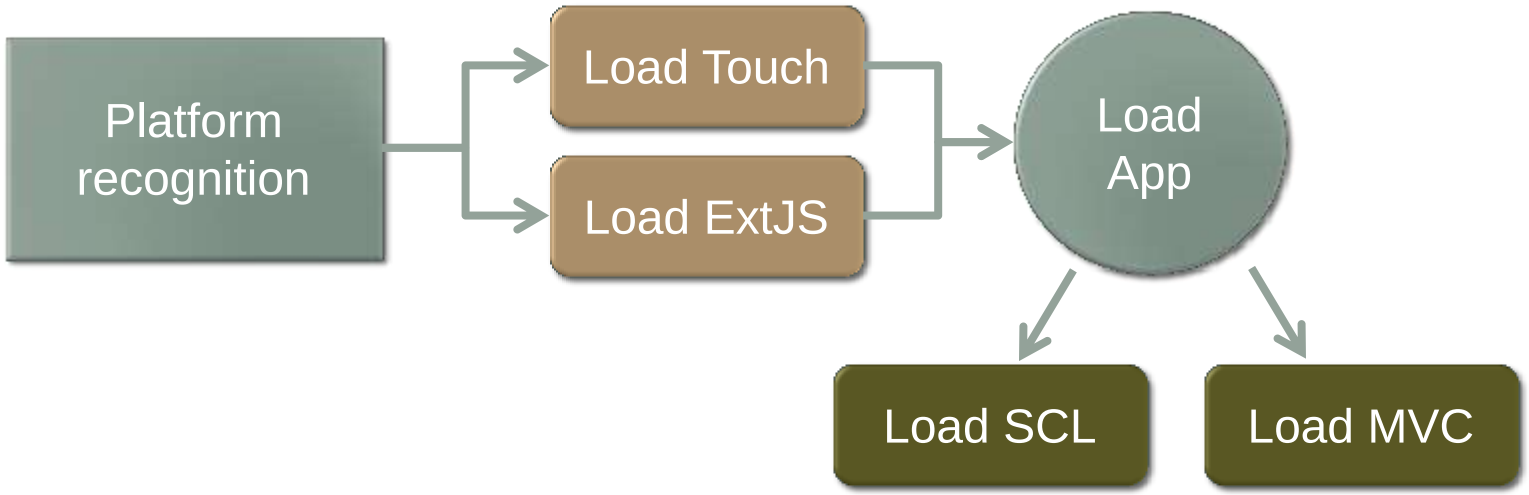
Overlap in Packages
AbstractComponent
AbstractManager
Ajax
app
ComponentManager
ComponentQuery
data
direct
fx
layout
ModelManager
Template
util
XTemplate
Action
Component
form
Img
LoadMask
picker
slider
tab

Missing one or more classes in Packages	
AbstractPlugin	FocusManager
app	form
chart	grid
ComponentLoader	Layer
container	menu
data	picker
draw	ProgressBar
ElementLoader	resizer
fx	Shadow
layout	ShadowPool
panel	slider
PluginManager	tip
selection	toolbar
state	tree
util	window
view	ZIndexManager
button	
dd	
Editor	
flash	

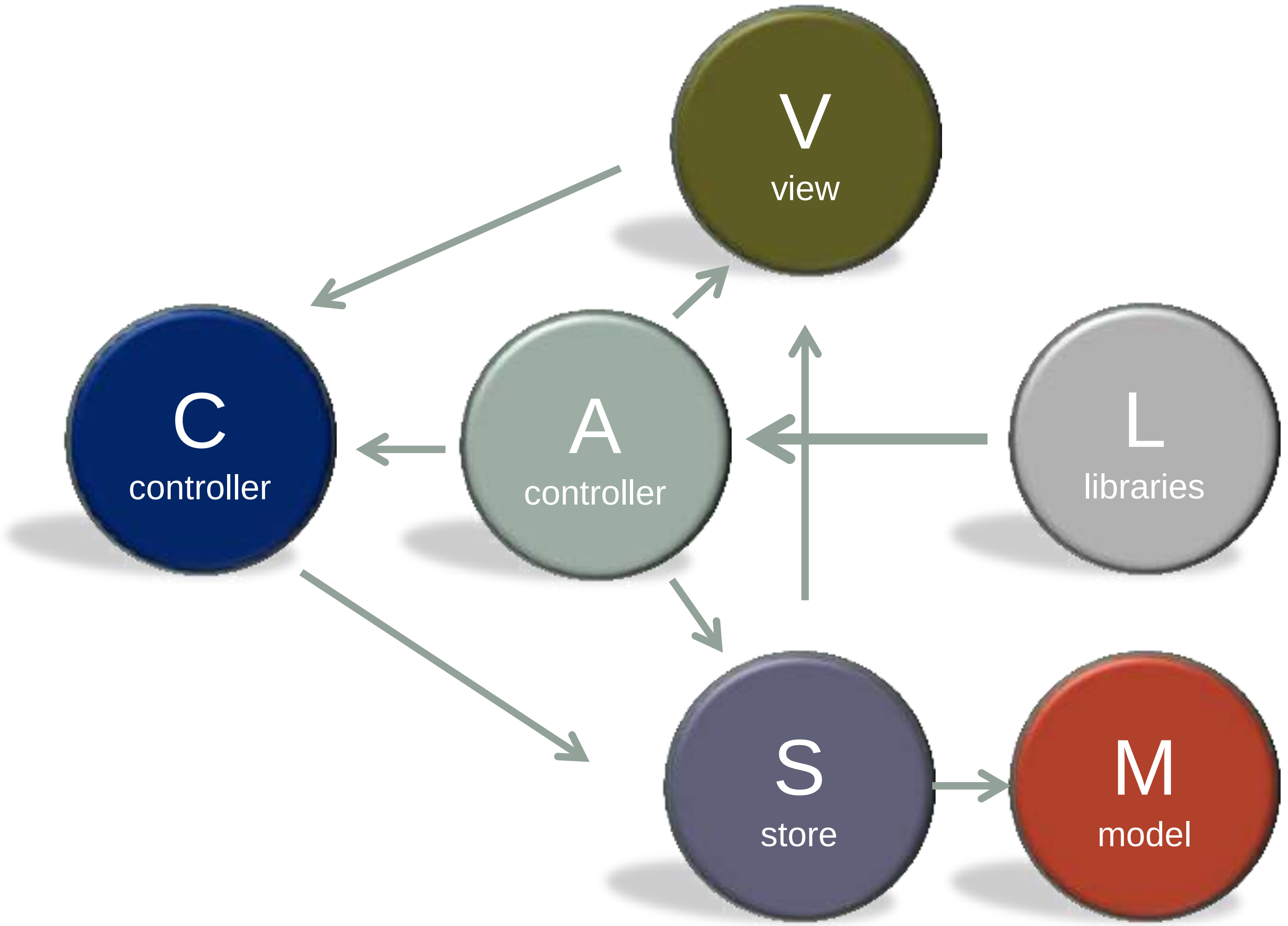
* Download full analysis spreadsheet from blog (http://www.boncode.net/downloads/ClassAnalysis.xlsx)

Approach 2: Shared Common Library (SCL)

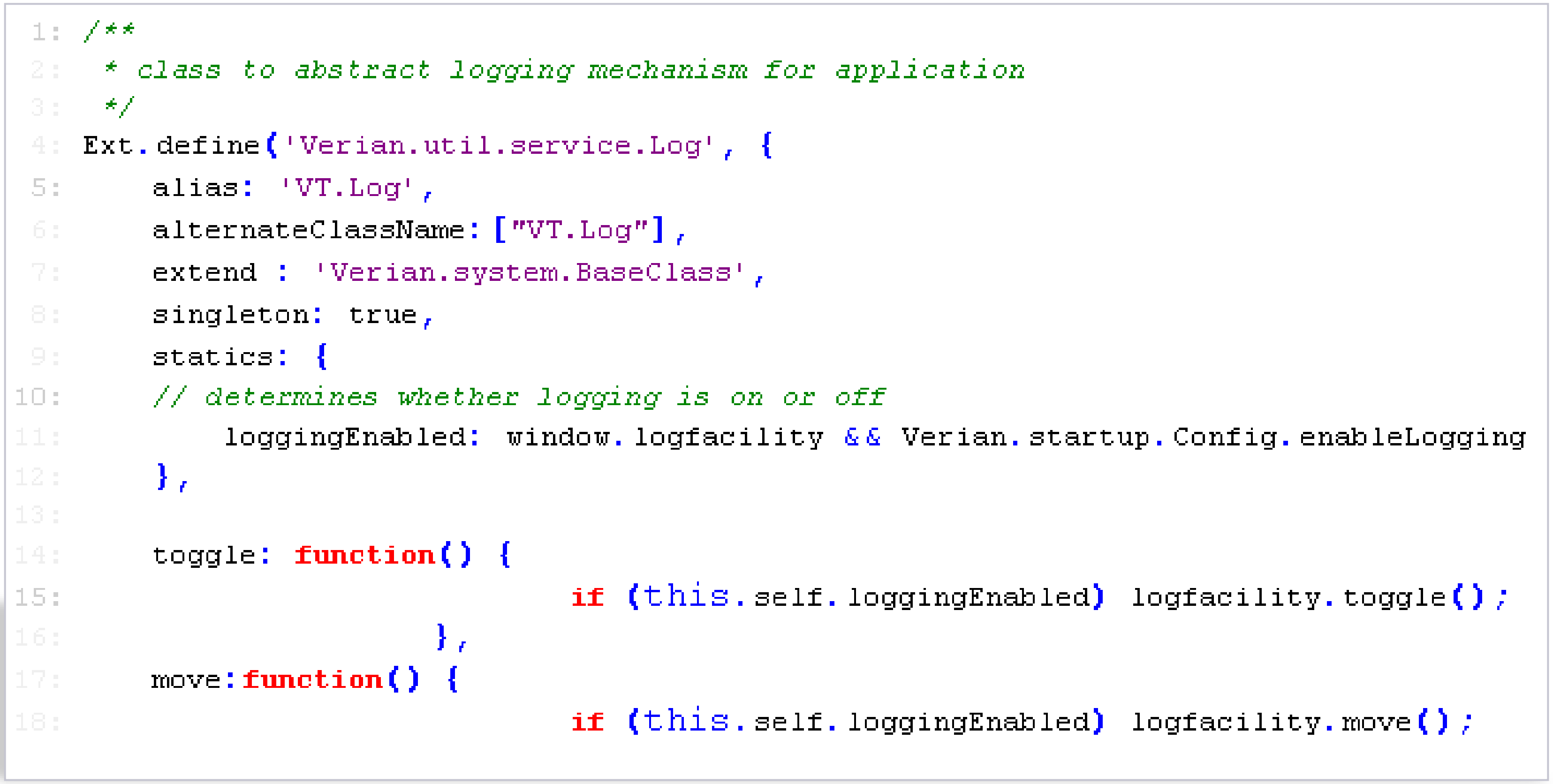
- Only uses overlapping core libraries and APIs
- Create shared libraries of classes using: Ext.define()
- Use them in your code via: Ext.create()
- Load SCL via “requires” config option



Approach 2: Model with Shared Common Libraries (SCL)



Shared Common Libraries Example



```
/**
 * class to abstract logging mechanism for application
 */
Ext.define('Verian.util.service.Log', {
    alias: 'VT.Log',
    alternateClassName: ["VT.Log"],
    extend : 'Verian.system.BaseClass',
    singleton: true,
    statics: {
    // determines whether logging is on or off
        loggingEnabled: window.logfacility && Verian.startup.Config.enableLogging
    },

    toggle: function() {
                            if (this.self.loggingEnabled) logfacility.toggle();
                    },
    move:function() {
                            if (this.self.loggingEnabled) logfacility.move();
```

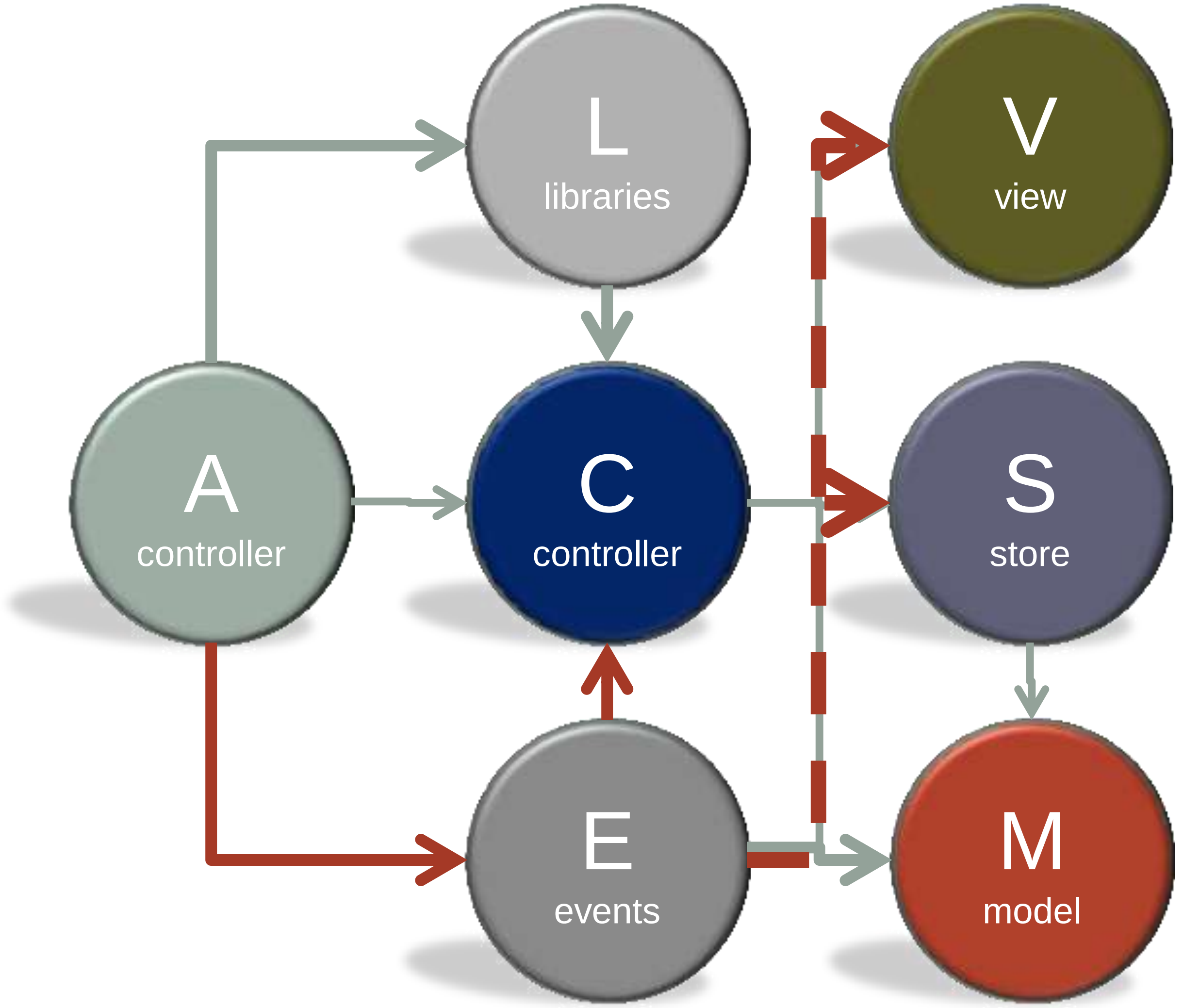
IS THIS ALL WE NEED?

2.

Still outstanding with Approach 2

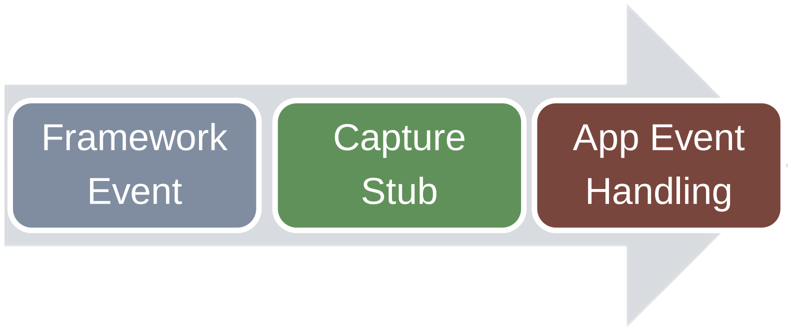
- Event System does not align
  - Tab, DoubleTab, Pinch vs. Click, MouseOut, MouseOver
  - MVC tight coupling to native events
- View mechanisms do not align
  - Using different controls for UI in touch vs extjs
- Controllers ?

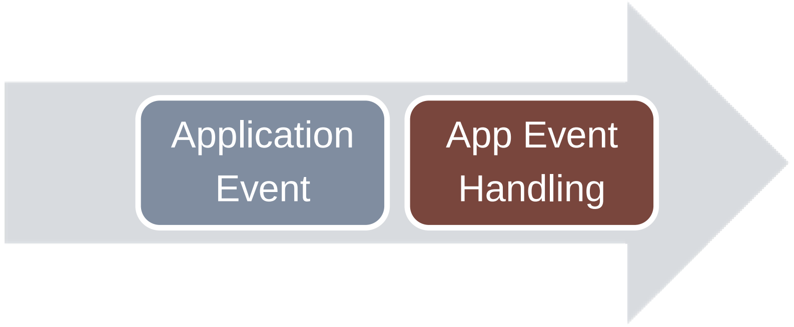
Approach 3: SCL and Event System



Event System Overlay (Demo)

- We chose a Publish / Subscribe system because of loose coupling of components and flexibility of subscriptions

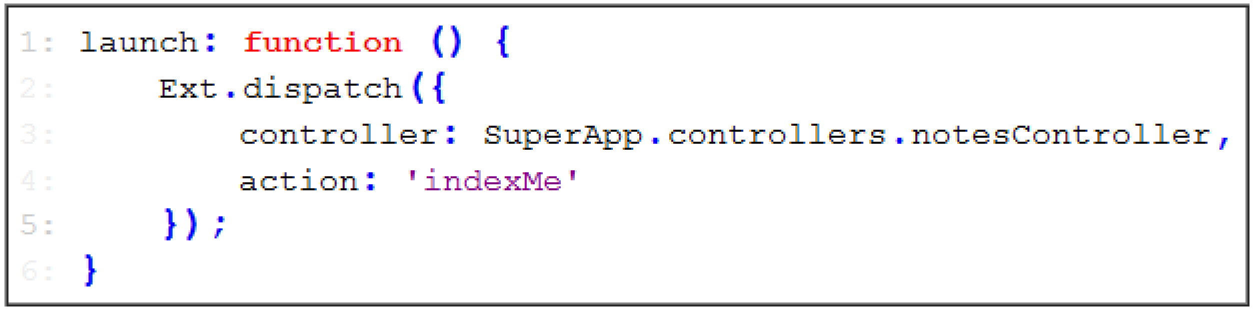




- Our Event System also acts as Application Message Bus

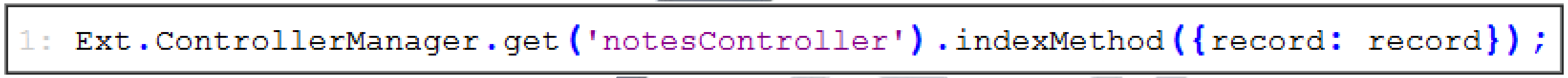
WE'RE PLANNING TO INTRODUCE OUR NEW MILITARY PRODUCT WITH A LIGHT SHOW IN NEW YORK HARBOR.
Dilbert.com DilbertCartoonist@gmail.com
WALLY, CAN YOU HANDLE THE WEAPON DEMO AND THE LIGHT SHOW?
SURE. WHAT COULD GO WRONG?
7-26-10 ©2010 Scott Adams, Inc./Dist. by UFS, Inc.
ONE WEEK LATER
THEY'RE CALLING IT "THE STUMP OF LIBERTY."
NO ONE IS SAYING IT WAS A BORING SHOW.

Moving towards loosely coupled events



```
launch: function () {
    Ext.dispatch({
        controller: SuperApp.controllers.notesController,
        action: 'indexMe'
    });
}
```

or



```
Ext.ControllerManager.get('notesController').indexMethod({record: record});
```

eventRegister(“ui.click.**”,…)

eventRegister(“ui.click.doubleClick”,…)

IS THIS ALL WE NEED?

3.

Still outstanding with Approach 3

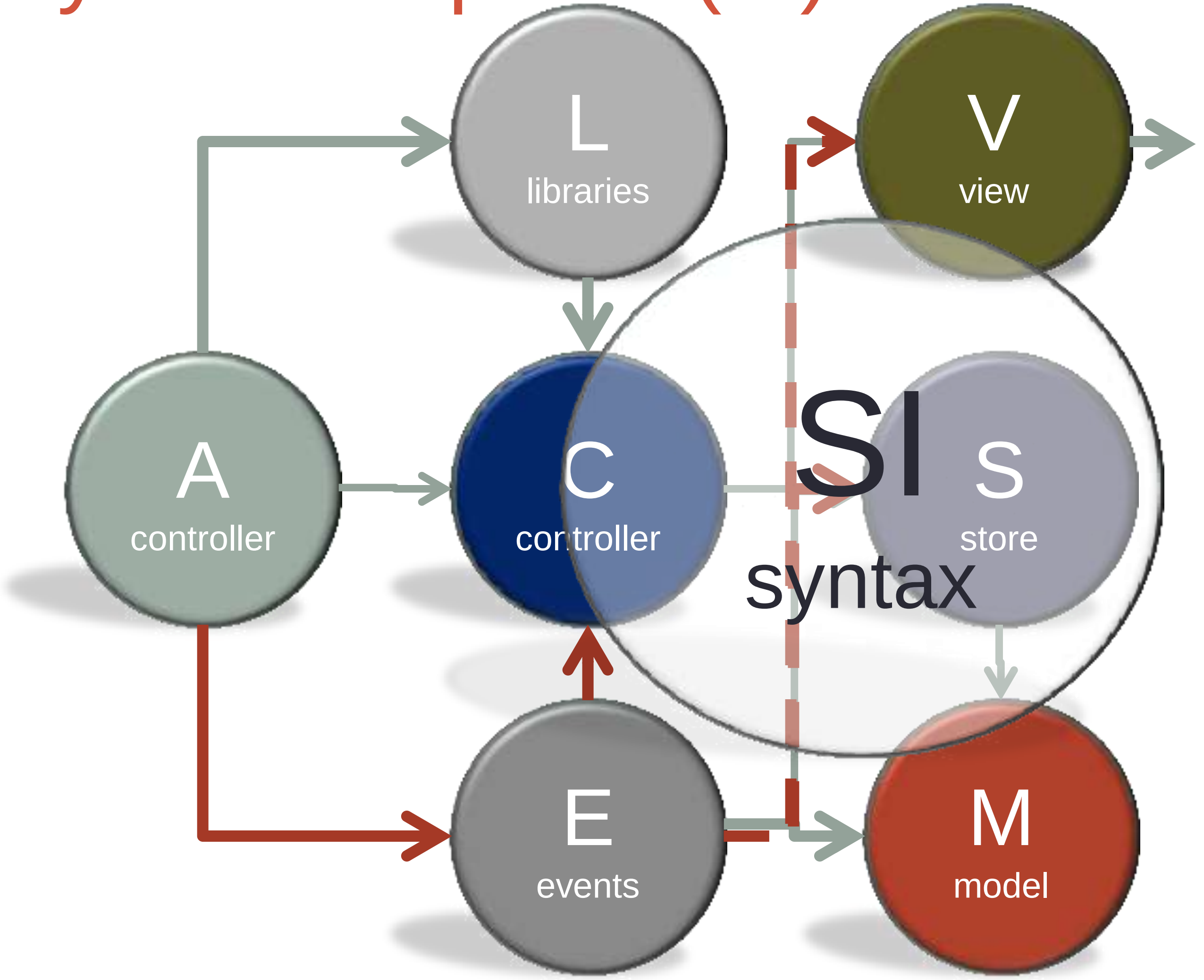
- Views
  - Visual elements will not be the same (difference in platforms)
- Controls
  - Are tightly coupled to views

- Need something that will express visual content and application business logic across frameworks.

Meta Logic / Language

- Addresses the need for higher level of abstraction needed
  - separate design language / commands
  - separate control language / commands
  - separate flow language / commands
- Ease of communication the goals and outcomes

Approach 4: SCL, Event System, & syntax interpreter (SI) for views and logic



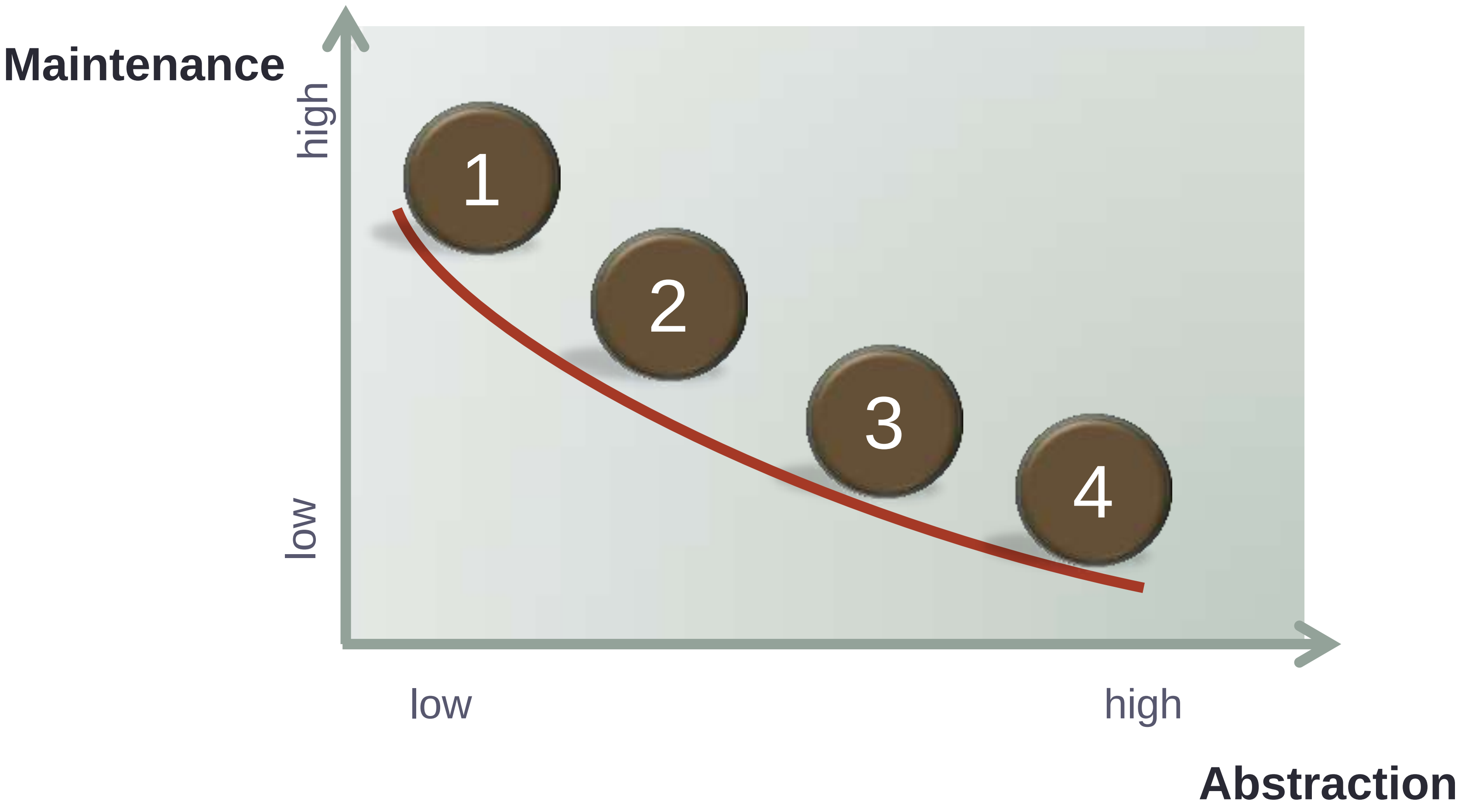
Design / Control Language

- Meta Language
- Used as mechanism for higher level abstraction
- Choices are available
- The Language is Interpreted by Syntax(Language) Interpreter. Most likely the place where most platform specific implementation should be housed.
- We used XML based and created own dialect
  - Feel free to experiment

Demo: Digging into variants of Approach 4

- Simple App Definition
  - UI Controls
- More complex scenarios

Solution Spectrum



Solution Spectrum

- 1: Using Common API and Platform Detection
- 2: Apps with shared commons libraries
- 3: Apps with shared events and commons libraries
- 4: Interpreted App (with design and control language)

Why / Benefits

- Goals:
  - higher code re-use, reduced maintenance, faster turn-around, easier upgradeability
- Overcome common hurdles
  - Sencha libraries are large (Learning Curve)
  - Experience with creating apps is limited
  - IDEs are less developed
- Why XML for Meta Language
  - Well formed XML is easily understood
  - Broad IDE support
  - Can easily be processed and generated on backend

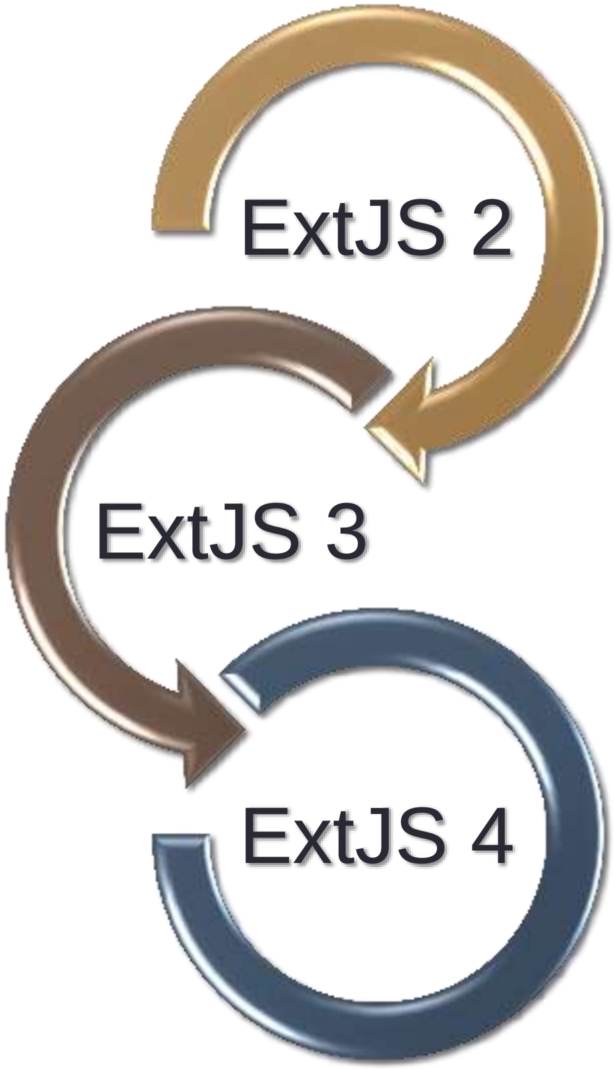
Alternatives for Meta Language

- The design language currently used is derived by needs of the project.
- Expanded application scope can also expand the need for complex language constructs
  - Workflow (View1 -> View2)
  - Data binding
  - Event binding
  - Exceptions
  - Customizations
  - Inheritance

More options: How to take this further

- Interpret JSON
- Use Other Dialects of Design / Control Markup
    - XAML (Microsoft)
    - MXML (Adobe) -- open source
        - Flash builder translates/compiles apps to HTML5/JS/CSS3 from MXML, thus should be possible to do this in Ext
    - SmartClient XML
- Source from DB

Another Side Effect

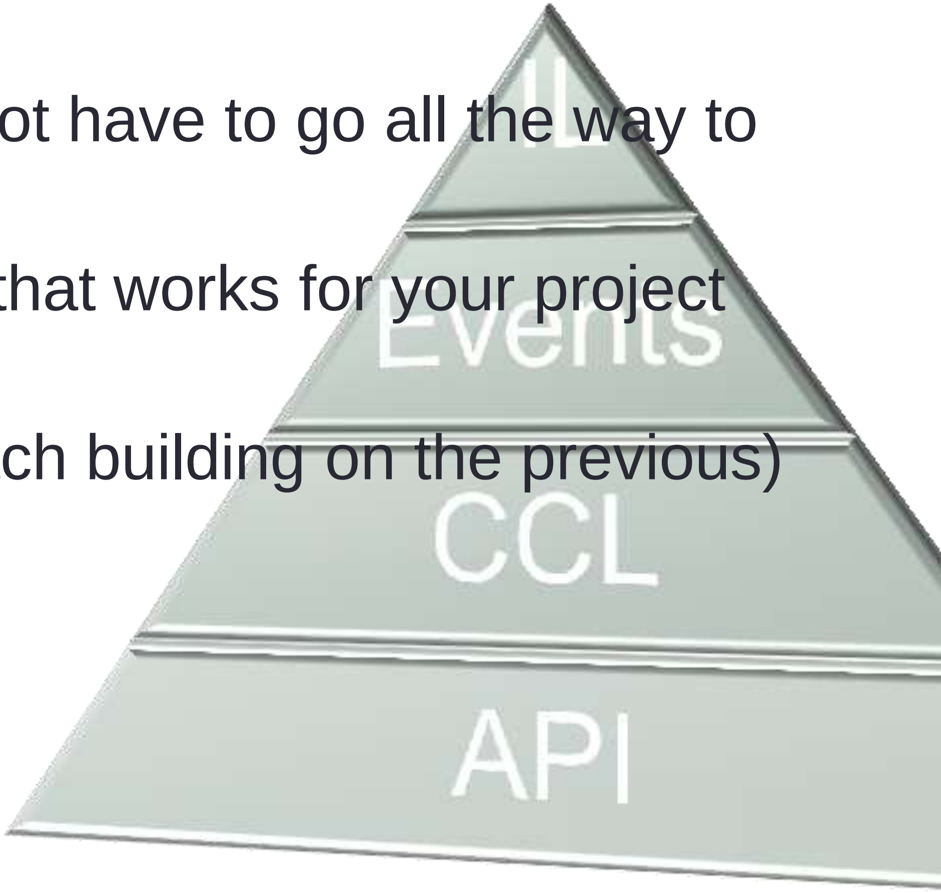


Drawbacks

- Thinking through unified application and code re-use requires extended planning
- Your application may require platform uniqueness / optimizations that are hard to abstract
- There is no long term investment calculation needed
- Still need to use platform specific CSS

Summary

- This may not work for your situation, but if it does, cool !
- With planning and design it is possible to re-use substantial amounts of code.
- A code reuse solution does not have to go all the way to work for you.
- Find a degree of abstraction that works for your project based on time/value analysis
- Levels of solution include (each building on the previous)
  - Common Class Libraries
  - Abstracted Event System
  - Abstracted View Definition
  - Abstracted Logic Definition



THANK YOU

@BmanClt

http://BonCode.blogspot.com