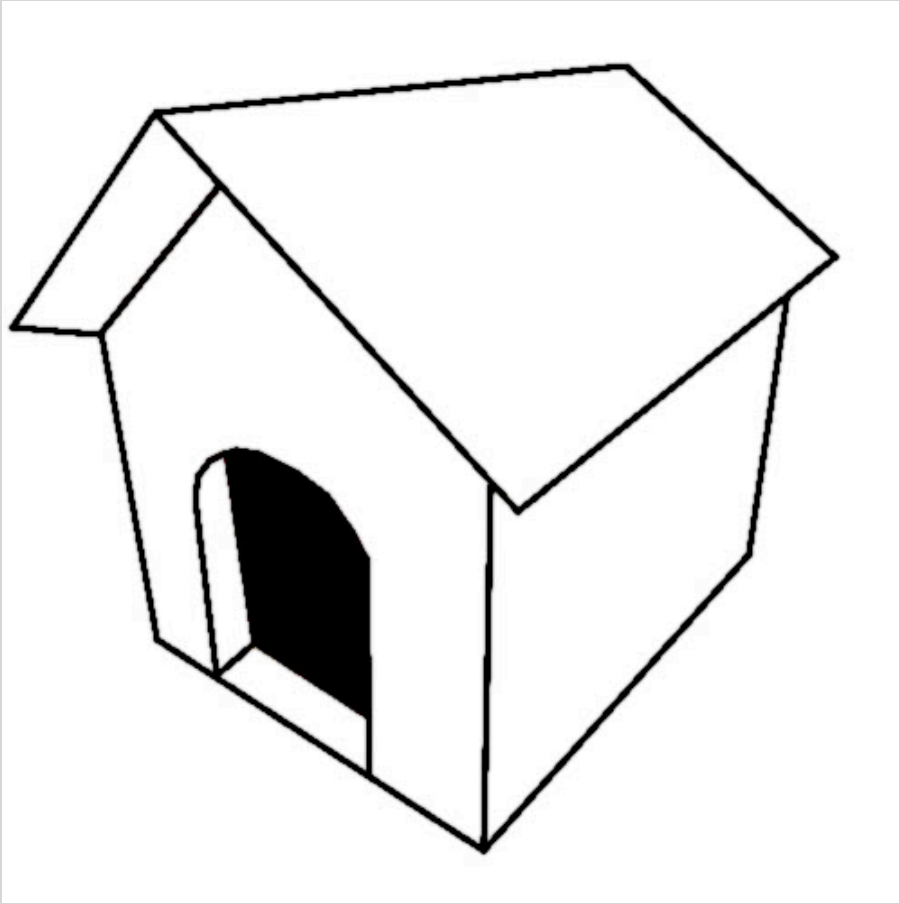


Introduction to Flex Application's Architecture

Architectural Frameworks
Patterns
RIA Methodologies

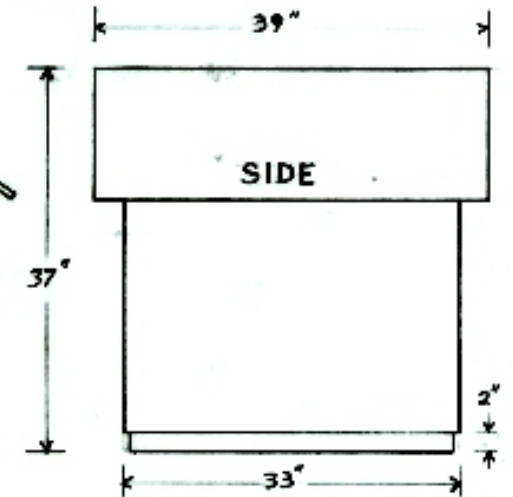
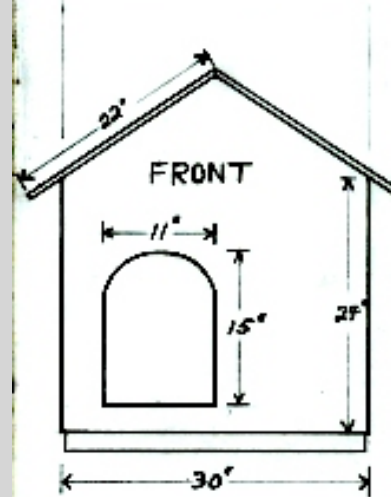
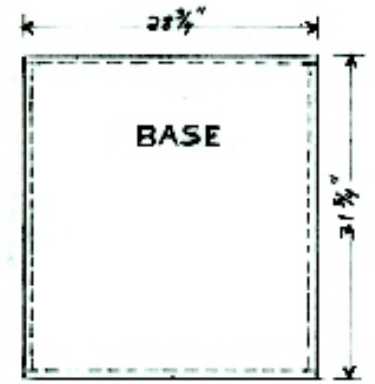
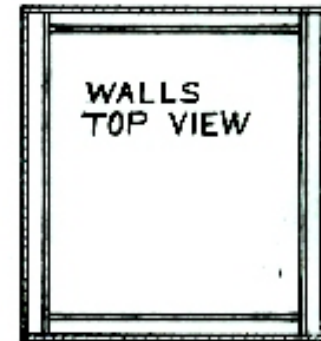


Why architecture is important?

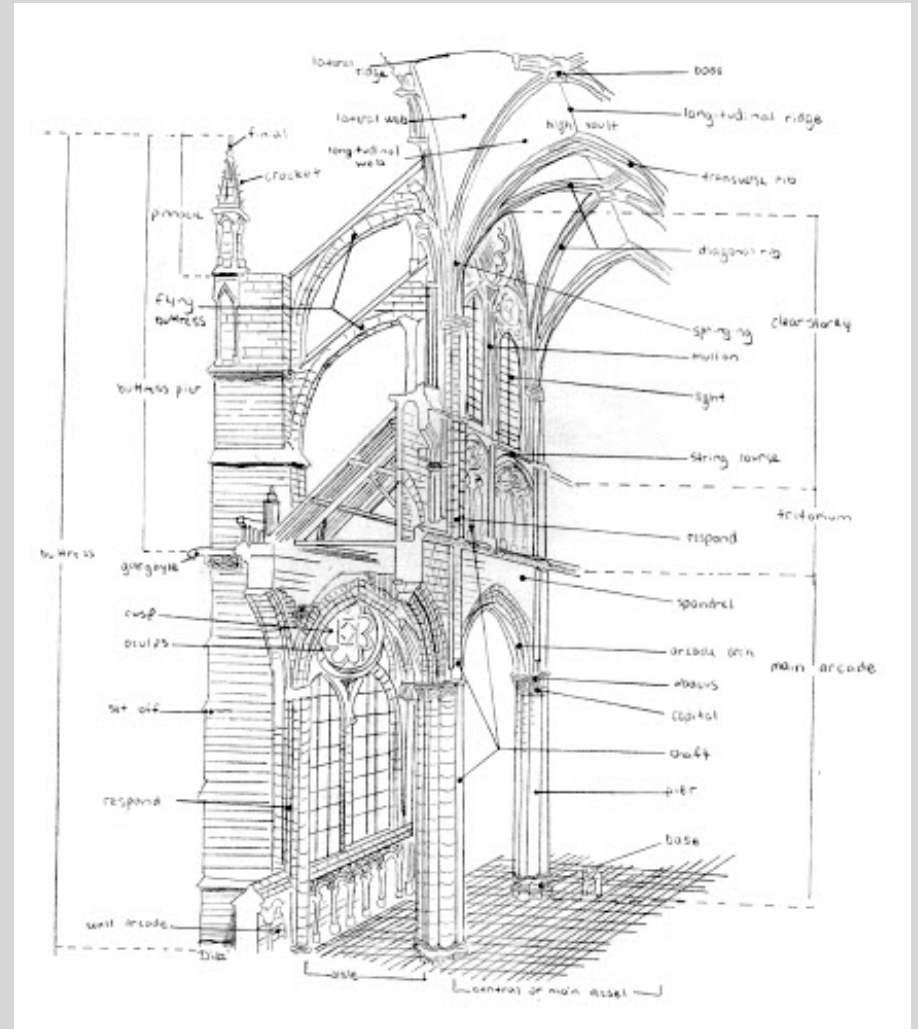


INSULATED DOG HOUSE

30" W. • 33" L. • 37" H



Let's build a dog house!



Let's build a cathedral!

Cathedral compared to a Dog House

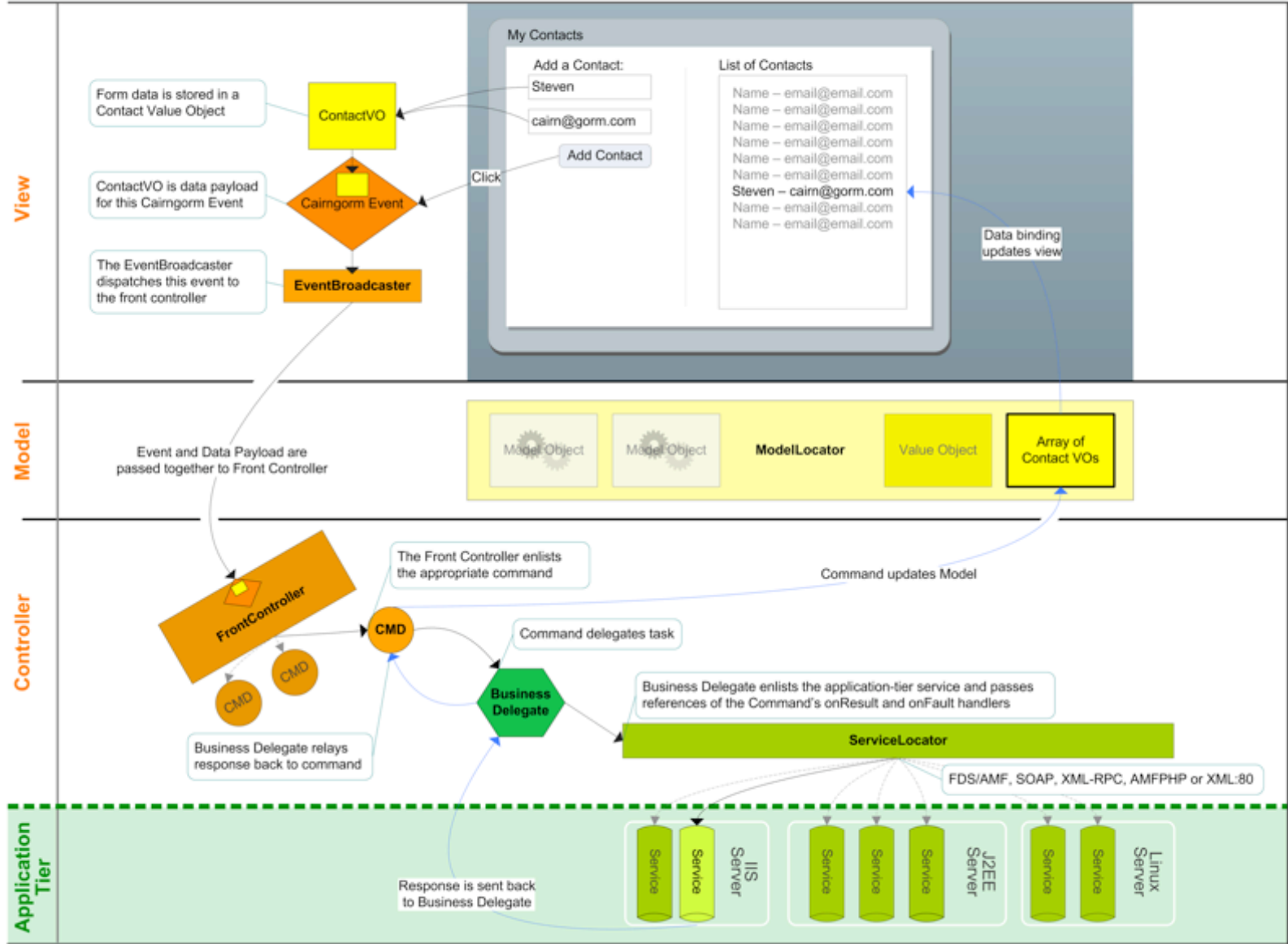
- 100 times bigger in size
- 1,000,000 times mass increase, but its strength only
- 10,000 times stronger, which is actually
- 100 times weaker



Architectural Frameworks

Cairngorm 2.0 Microarchitecture – Basic Server RPC

From 'Developing Flex RIAs with Cairngorm Microarchitecture' by Steven Webster
 Copyright: Adobe.com - http://www.adobe.com/devnet/flex/articles/cairngorm_pt1.html

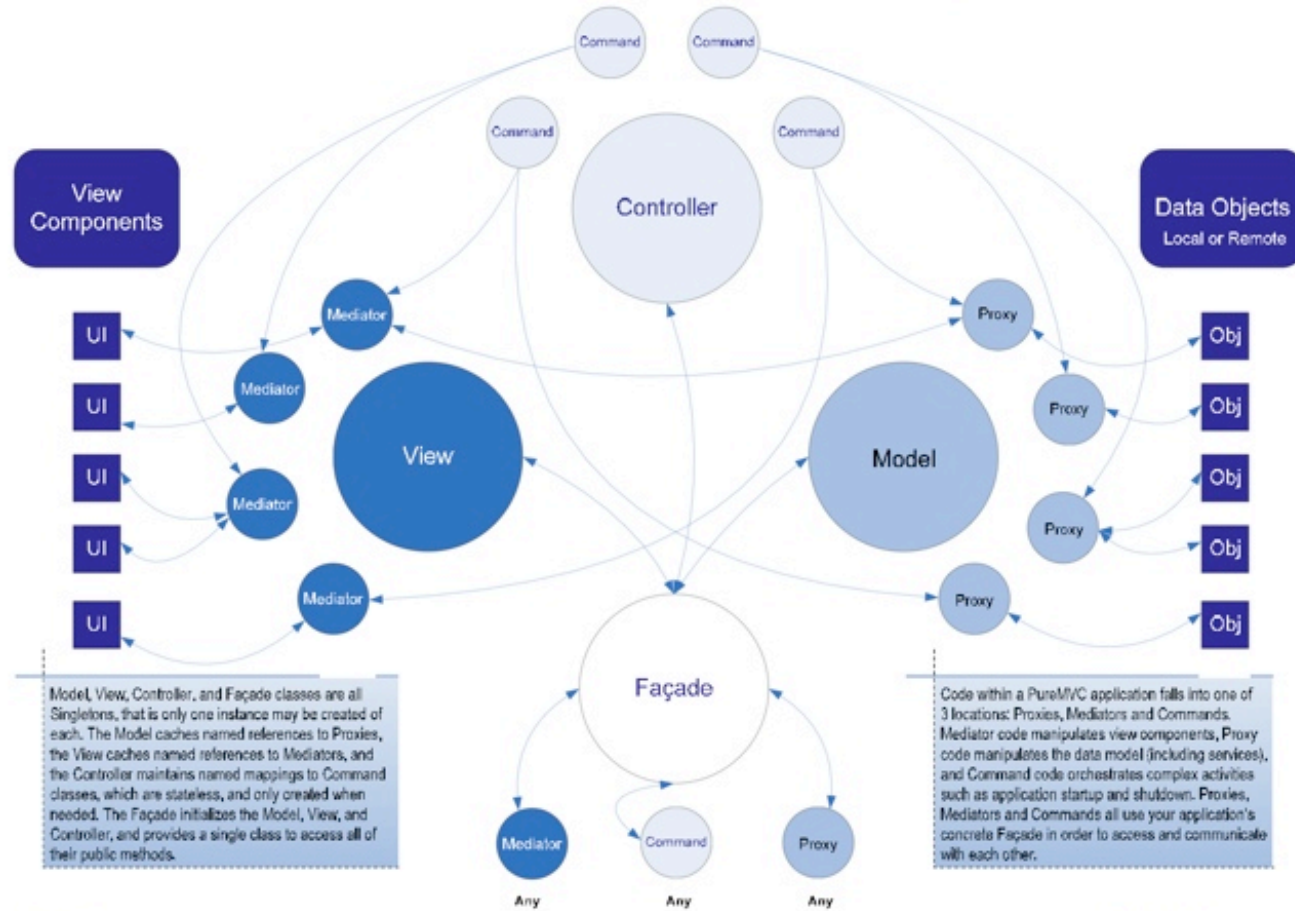


Updated: 11/23/2006 - Diagram by Evan Gifford (evan@usa.com) with special thanks to Steven Webster, Alex Uhlmann, Russell Munro, Jesse Warden, Dan Nielsen, Tom Chiverton and everyone on the flexcoders group - flexcoders.org

Cairngorm

puremvc framework

CONCEPTUAL DIAGRAM AND INTRODUCTION



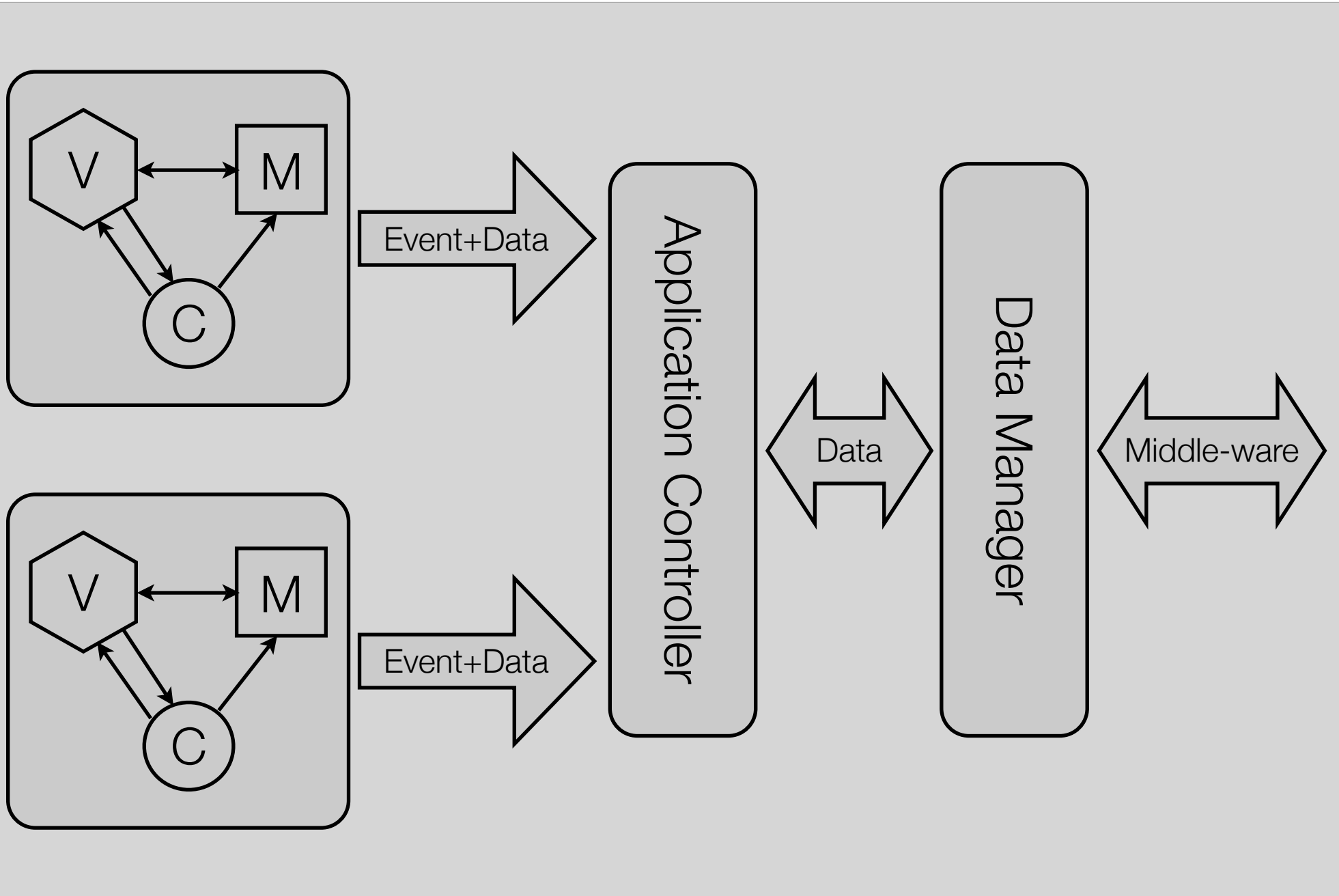
AUTHOR: Cliff Hall <cliff@puremvc.org>

LAST MODIFIED: 3/05/2008

PureMVC is a free, open source framework created and maintained by Futurescale, Inc. Copyright © 2006-08. Some rights reserved.
 Reuse is governed by the Creative Commons 3.0 Attribution Unported License. PureMVC, as well as this documentation and any training materials or demonstration source code downloaded from Futurescale's websites is provided 'as is' without warranty of any kind, either express or implied, including, but not limited to, the implied warranties of fitness for a purpose, or the warranty of non-infringement.

Why MVC sometimes considered harmful?

- big flat monolithic Controller
- complicated Model
- logic too far from View



The Gugga Approach

Data Binding Patterns

```
package com.adobe.ac.util
{
    public class Observe
    {
        public var handler:Function;

        public function set source(source:*):void
        {
            handler.call();
        }
    }
}
```

```
<util:Observe source="{model.myProperty}"
    handler="{this.myFunction}"/>
```

Observer Tag

Function is executed on every property change

Before

```
<mx:List dataProvider="{getCustomerNames(
    model.customers, model.product) }"/>
<mx:TextBox text="{getTotalCost(
    model.customers, model.product) }"/>
```

After

```
<adapters:ProductStatisticModel
    id="statistic"
    customers="{model.customers}"
    product="{model.product}"
    discount="{model.discount}"/>

<mx:List dataProvider="{statistic.names}"/>
<mx:TextBox text="{statistic.cost}" />
```

Adapter

Optimizing and structuring of complex and ineffective data binding expressions

Inversion of Control Patterns

IoC Goals

- reduce class dependencies
- extracting the creational logic and the components dependencies management

```
package com.samples
{
    public class Loader
    {
        public function load():void
        {
            ...
            var bar:IBar = new LoopingProgressBar();
            ...
        }
    }
}
```

Setter Injection Pattern

- use interfaces instead of concrete classes
- use the setter method to initialize the referenced component

```
package com.samples
{
    public class Loader
    {
        private var _bar:IBar;
        public function set bar(value:IBar):void
        {
            ...
            bar = value;
            ...
        }
        public function get bar():IBar
        {
            return _bar;
        }

        public function load():void
        {
            //using the injected reference bar
        }
    }
}
```

MVC with IoC

- sample Flex dependencies management
- use data binding within the MXML for setter dependencies injection

```
<model:StockModel id="model"/>  
  
<control:controller id="controller"  
    model="{model}"/>  
  
<view:StockDisplay id="view"  
    model="{model}"  
    buy="controller.buy(event.stocks)"/>
```

RIA Methodologies

Front to Back Development

- User experience and interactions first
- Early prototype development (use mockup data)
- Requirements elicitation boost
- Client communication boost
- Client satisfaction boost
- User experience boost

Feature Driven Development

- Develop an overall application model
- Collect list of features
- Start implementing feature sets within short iterations (usually 2 weeks)

Reference

- Alan Key: The Computer Revolution hasn't happened yet OOPSLA 97 - <http://video.google.com/videoplay?docid=-2950949730059754521&q=alan+kay&ei=J-4RSNS7Lle82wKR7uSeAg>
- Sho Kuwamoto: MVC considered harmful - http://weblogs.macromedia.com/sho/archives/2005/08/mvc_considered.html
- Branimir Angelov: Patterns and Practices for building RIAs with Adobe Flex FlexCamp Sofia 2008 - <http://flex.riabg.org/>
- My blog: <http://npacemo.com/>