Comparing Flex Frameworks

Yakov Fain
yfain@faratasystems.com

Slides at:
www.myflex.org/presentations/ComparingFlexFrameworks.pdf

Farata Systems: Quick Intro

FARATA SYSTEMS

- Flex Experts for Your Next Project
- Tools to Turn Your Application into to RAD Project
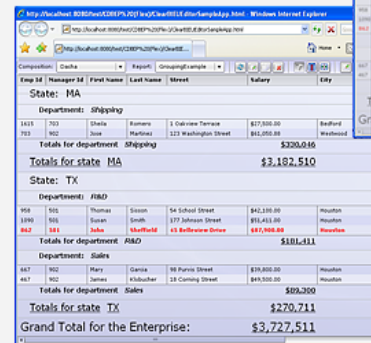- Certified Trainers to Raise Your Flex Talent

ClearDataBuilder

FxAnt

Fx2Doc

ClearBI

LogFx

www.theriabook.com

RICH INTERNET APPLICATIONS

ADOBE
FLEX & JAVA

Written by
Yakov Fain
Dr. Victor Rasputnis
Anatole Tartakovsky

Guest chapter by Ben Stucki

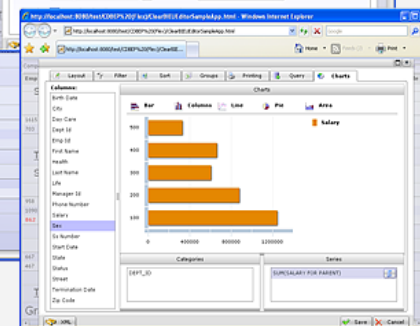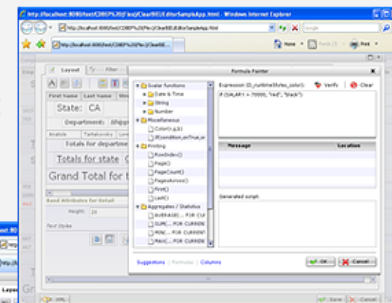Foreword by Bruce Eckel & Matt Chotin

CLEAR BI

The Only RIA Solution
for Enterprise Reporting

- Controlled by the Business Users
- Targeted for the Business Users
- Displayed by Flash Player
- Zero Installation on the User's PC
- Allows adding Excel-like formulas
- Users Create Groupings
- Users Modify Presentation
- Prints reports in PDF format
- Exports to Microsoft Excel
- Persist and share reports

www.myflex.org

Upcoming book with O'Reilly: "Enterprise Application Development with Flex"

Today, we'll talk about

- **Cairngorm** by Adobe Consulting

- **Mate** by ASFusion

- **PureMVC** by Cliff Hall

- **Clear Toolkit** by Farata Systems
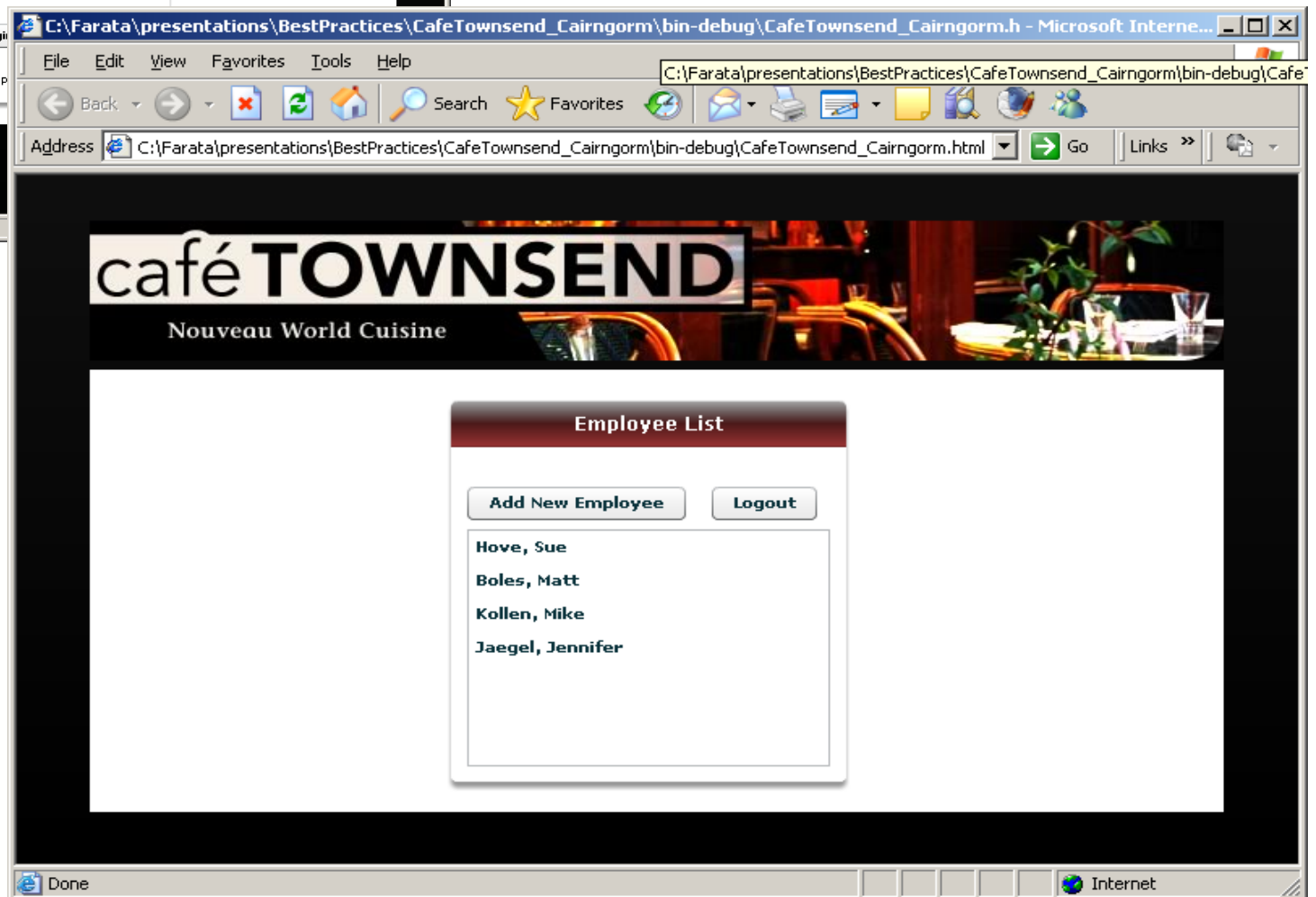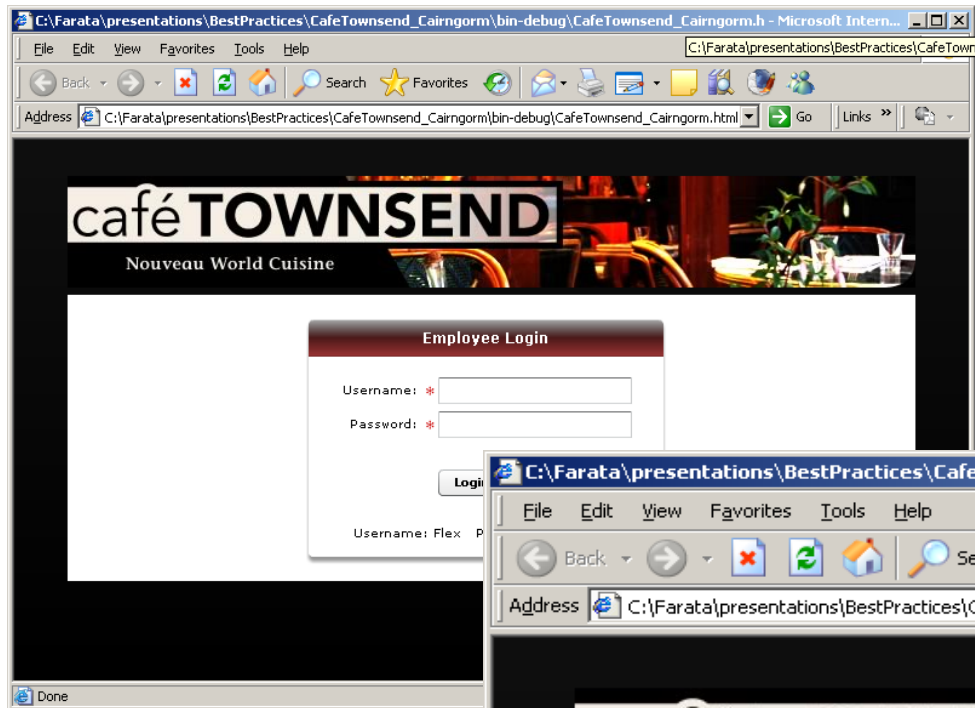
Getting Employee List

```xml
<?xml version="1.0" encoding="utf-8"?>
<employees>
     <employee>
          <emp_id>1</emp_id>
          <firstname>Sue</firstname>
          <lastname>Hove</lastname>
          <email>shove@cafetownsend.com</email>
          <startdate>01/07/2006</startdate>
     </employee>
          ...
</employees>
```

```xml
<?xml version="1.0" encoding="utf-8"?>
<mx:Application xmlns:mx="http://www.adobe.com/2006/mxml" layout="absolute" creationComplete="empService.send()">
    <mx:Panel title="Employee List" horizontalCenter="0">
        <mx:HBox paddingTop="25">
            <mx:Button label="Add New Employee" />
            <mx:Spacer width="100%" />
            <mx:Button label="Logout" />
            <mx:Spacer width="100%" height="20" />
        </mx:HBox>
        <mx:List id="employees_li" dataProvider="{ employees}" labelFunction="fullName" width="100%"/>
    </mx:Panel>

    <mx:HTTPService id="empService" url="assets/Employees.xml" result="employeeDataHandler(event)" />

    <mx:Script>
        <![CDATA[
        import mx.rpc.events.ResultEvent;
        import mx.collections.ArrayCollection;

        //data provider for the list is an ArrayCollection a.k.a. model
        [Bindable]
        private var employees: ArrayCollection=new ArrayCollection;

        private function employeeDataHandler(event:ResultEvent):void{
            employees=event.result.employees.employee;
            }
        // format the names to display last and first names in the List
        public function fullName( empItem : Object ) : String {
            return empItem.lastname + ", " + empItem.firstname;
        }
        ]]>
    </mx:Script>

</mx:Application>
```

List of Employees with Flex (no other frameworks)

Code Review

Café Townsend Cairngorm

Cairngorm

Sample dataflow

Front Controller

1. Dispatch Event

2. Execute command

10. Update bound view component

View

9. Update model

Command

Model Locator

8. Return result/fault

3. Call business function

5. Return RemoteService

Services

Delegate

4. Find remote service

6. Call remote service

7. Return result/fault

© Farata Systems, 2008

Remote Service
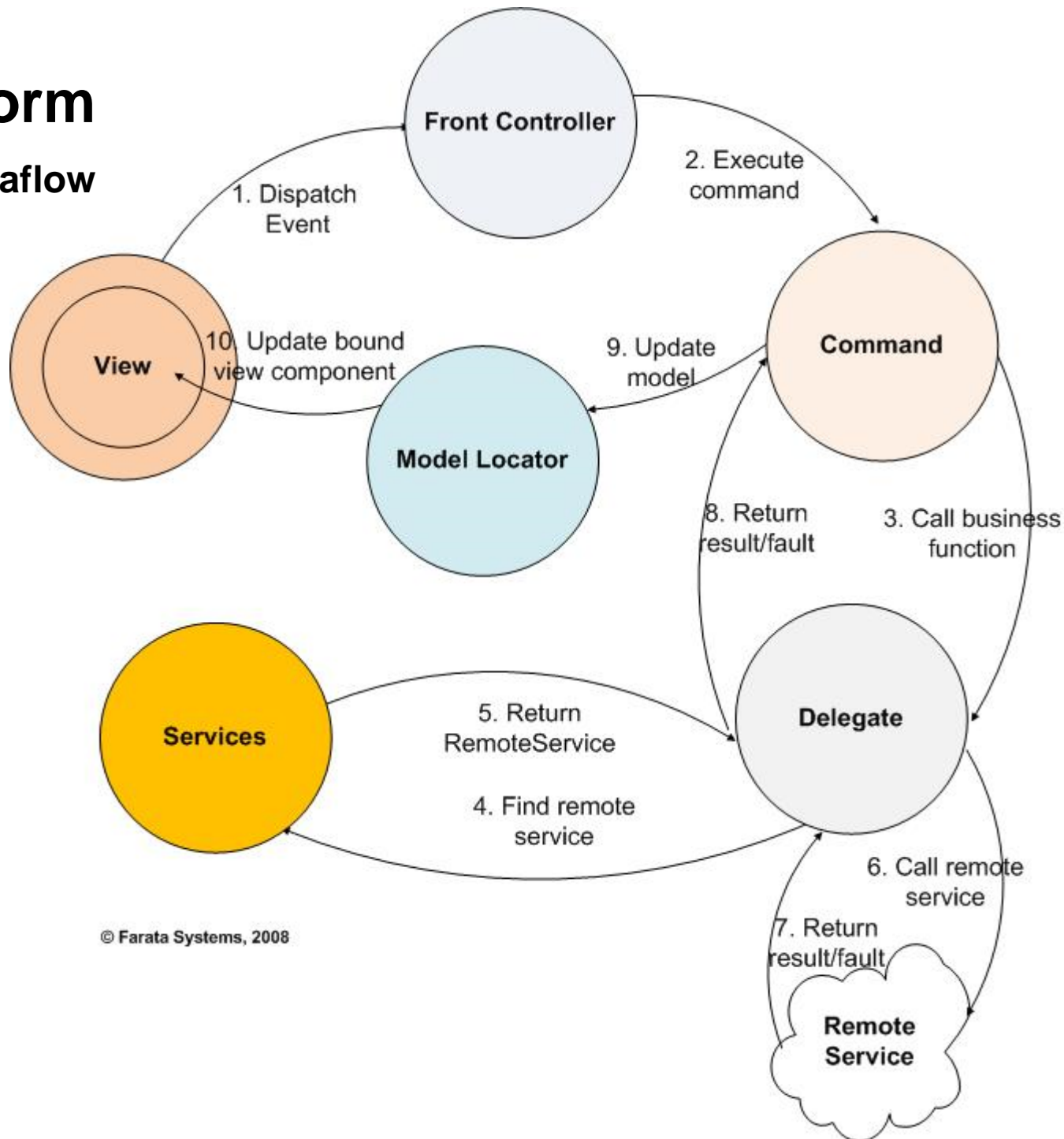
Cairngorm: Pros and Cons.

Pros: - It's a popular framework- many Flex developers know it
- Allows to separate responsibilities of developers
- Lowers the requirements to the skill-set of developers

Cons: - Requires developers write lots of additional classes, which adds to project timeline
- Built on global singletons, which complicates modularization
- Forces you to use only binding for updating views
- Allows only one-to-one mapping between events and commands
- Having several singletones leads to tight object coupling

Code Review

Café Townsend Mate

Mate
Sample dataflow

MyEventMap.mxml

3. Find Corresponding EventHandler

2. Event bubbles up

7. Optionally chain business functions

<EventHandler>

View

1. Dispatch system or custom event

MyEvent

10. Inject data into view

8. Update model

6. Call business function

4. Call service

5. Return result/fault

<PropertyInjector>

9. Injector gets data

MyModel

MyComponent

Remote Service

© Farata Systems, 2008
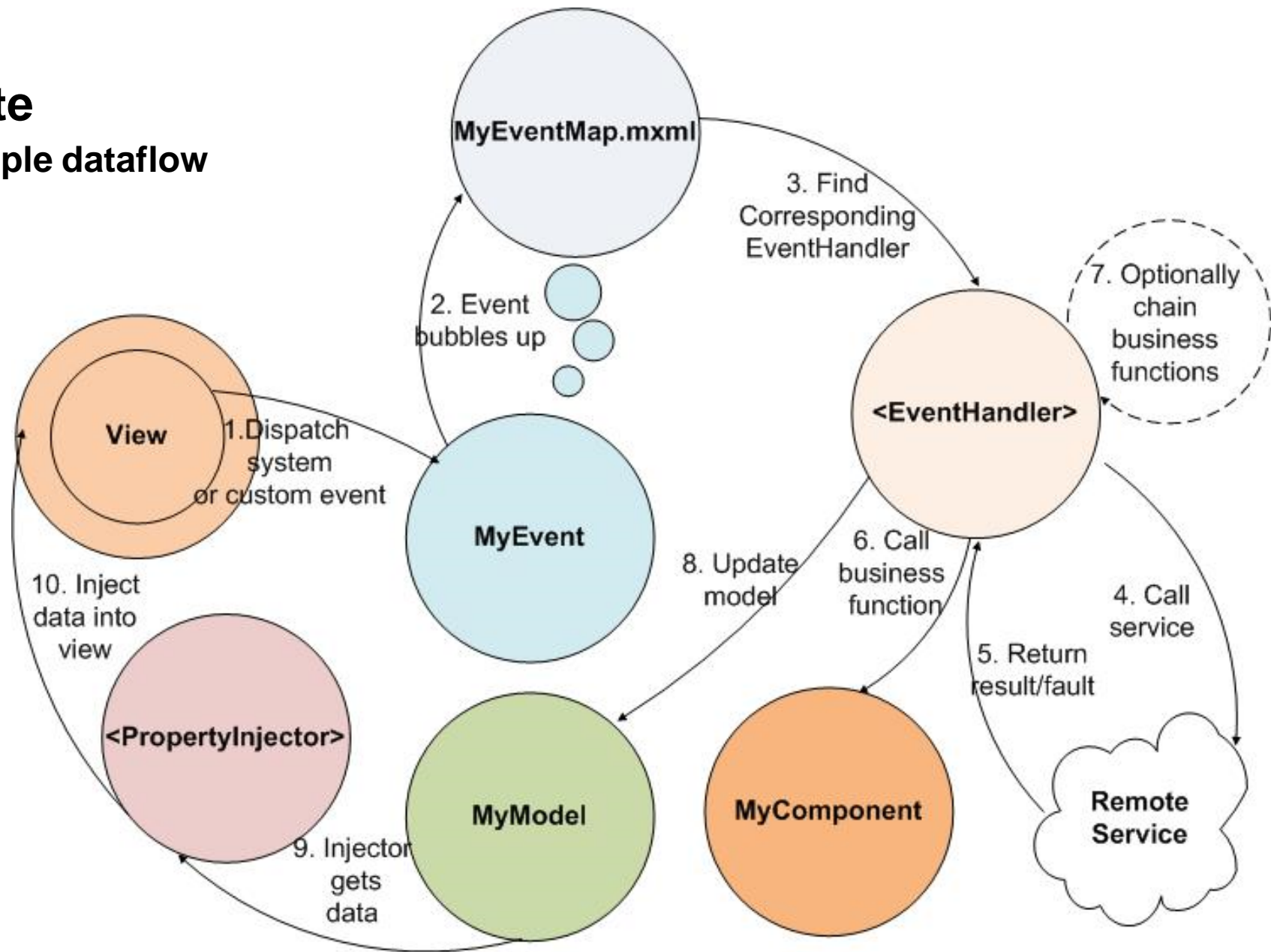
Mate: Pros and Cons.

Pros: - Mate is non-intrusive – Mate-specific code can be
encapsulated in a handful of objects.
- It's MXML based and allows you to keep using Flex
event model
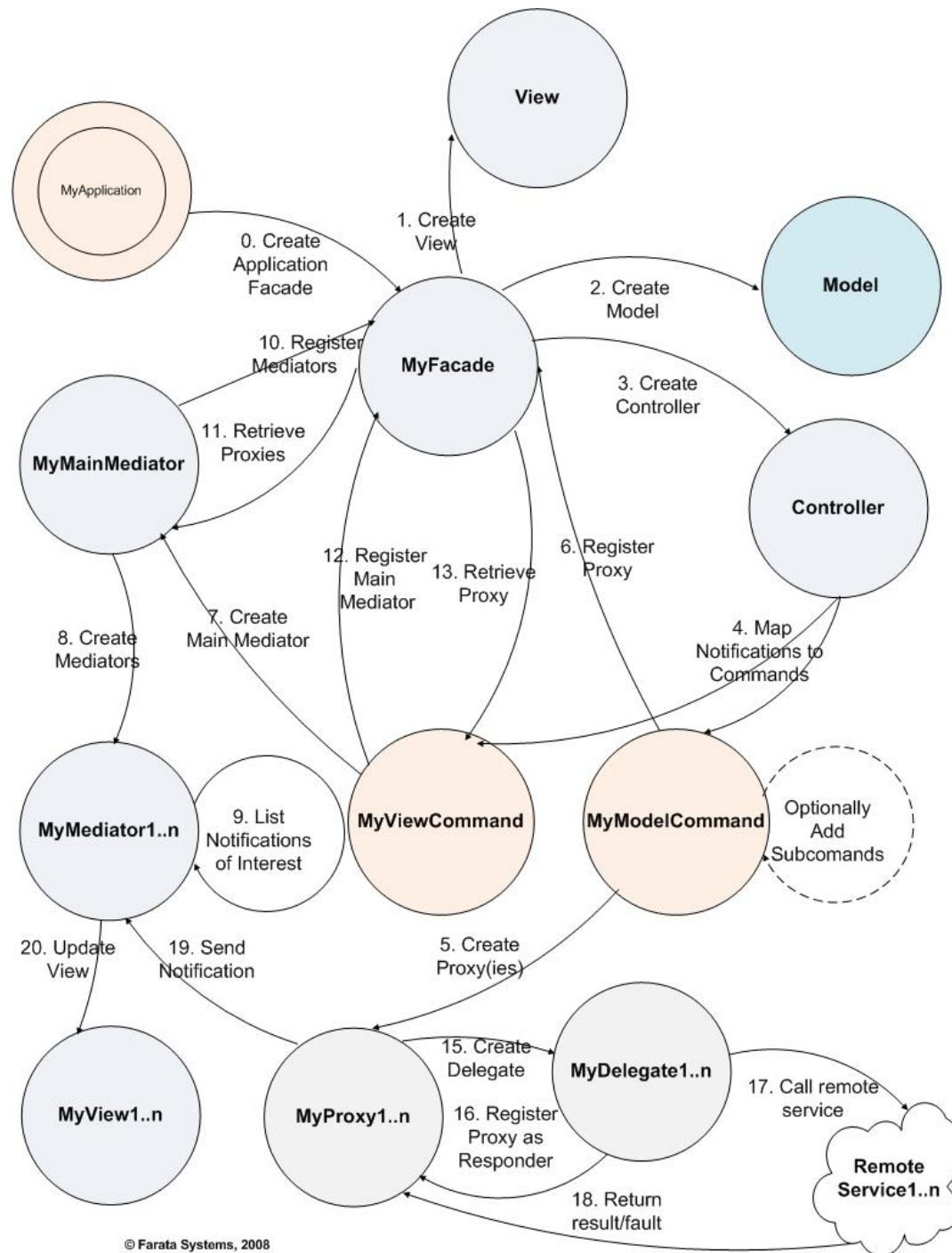- promotes lose coupling between components by
implementing Dependency Injection
- Well documented

Cons: - It hasn't been released yet (currently in Alpha)
- Doesn't support working with Data Management
Services offered by LCDS

Code Review

Café Townsend PureMVC

PureMVC
Sample dataflow

View

MyApplication

0. Create Application Facade

1. Create View

2. Create Model

Model

10. Register Mediators

11. Retrieve Proxies

MyFacade

3. Create Controller

MyMainMediator

12. Register Main Mediator

13. Retrieve Proxy

6. Register Proxy

Controller

7. Create Main Mediator

8. Create Mediators

4. Map Notifications to Commands

MyMediator1..n

9. List Notifications of Interest

MyViewCommand

MyModelCommand

Optionally Add Subcomands

20. Update View

19. Send Notification

5. Create Proxy(ies)

MyView1..n

MyProxy1..n

15. Create Delegate

MyDelegate1..n

17. Call remote service

16. Register Proxy as Responder

18. Return result/fault

Remote Service1..n

© Farata Systems, 2008

PureMVC: Pros and Cons.

Pros: - It's well documented
- It's available for developers that want to use
  ActionScript only (i.e. Flash programmers)

Cons: - It's not a framework written for Flex hence does use
  features offered by MXML
- Doesn't use standard Flex events
- Has too many layers, which are tightly coupled
- Requires staffing projects with more senior developers
- Developers have to write lots of additional classes,
  which adds to project timeline
- It's built on several singletons, and application code
  becomes cluttered by making multiple calls to them.

Clear Toolkit 3.0

Library of Components

Clear Data Builder	DTO2Fx	Log4Fx	Fx2Ant

ClearBI Reporter

It's **not** an architectural framework, but a set of components that extend Flex framework and tools. It substantially increases productivity of the enterprise Flex developers.

Clear Toolkit: Pros and Cons.

Pros: - Offers a library of enriched Flex components like super grid, data-aware components, et al
- Automatically generates a lot of code, which minimizes the amount of code to be written by developers
- Offers data synchronization functionality in free BlazeDS similar to Data Services from LCDS

- Its components can be used on as needed basis
- It offers Web reporting tool for developers and end-users
- It automates creation of the build scripts in Ant
- Clear Toolkit is easier to spell than Cairngorm

Cons: - It hasn't been released yet (currently in private Beta)
- It's not an architectural framework and requires to have a senior architect on the mid-to-large size projects

```xml
<?xml version="1.0" encoding="utf-8"?>
<mx:Application xmlns:mx="http://www.adobe.com/2006/mxml" layout="absolute" creationComplete="empService.send()">
    <mx:Panel title="Employee List" horizontalCenter="0">
        <mx:HBox paddingTop="25">
            <mx:Button label="Add New Employee" />
            <mx:Spacer width="100%" />
            <mx:Button label="Logout" />
            <mx:Spacer width="100%" height="20" />
        </mx:HBox>
        <mx:List id="employees_li" dataProvider="{ employees}"  labelFunction="fullName" width="100%"/>
    </mx:Panel>

    <mx:HTTPService id="empService" url="assets/Employees.xml" result="employeeDataHandler(event)" />

    <mx:Script>
        <![CDATA[
        import mx.rpc.events.ResultEvent;
        import mx.collections.ArrayCollection;

        //data provider for the list is an ArrayCollection a.k.a. model
        [Bindable]
        private var employees: ArrayCollection=new ArrayCollection;

        private function employeeDataHandler(event:ResultEvent):void{
                employees=event.result.employees.employee;
        }
        // format the names to display last and first names in the List
        public function fullName( empItem : Object ) : String {
                return empItem.lastname + ", " + empItem.firstname;
        }
        ]]>
    </mx:Script>
</mx:Application>
```

List of Employees with Flex (no other frameworks)

Let's have a moment of silence in memory of slide 6 of this preso.
Bringing a list of Employees was so simple with Flex framework!

Please stand up.

Slides URL & Contact Info

Slides:

www.myflex.org/presentations/ComparingFlexFrameworks.pdf

Email: yfain@faratasystems.com

Web sites:

http://www.faratasystems.com

http://flexblog.faratasystems.com

http://www.myflex.org

FARATA