

# Cairngorm Microarchitecture

<http://www.adobe.com/go/cairngorm>

**Павел Кожин**

[vertex.mailbox@gmail.com](mailto:vertex.mailbox@gmail.com)

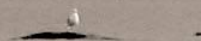
<http://www.riapriority.com/vertex>

Exigen Services

# Произношение

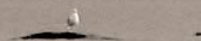
**Cairngorm (*karn gorm*)**

Желто-коричневая разновидность топаза,  
залегающая преимущественно в Шотландии.



# Что такое Cairngorm?

- Набор классов и интерфейсов
- Микроархитектура на базе паттернов
- Проверенное решение для создания Flex-приложений



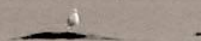
# Зачем?

- Нет необходимости изобретать велосипед
- Стандартный подход
- Легкий способ добавления новой функциональности
- Повторное использование



# Используемые паттерны

- Singleton
- Command
- Delegate
- ValueObject
- Observer



# Решаемые проблемы

- **Обработка действий пользователя**
  - FrontController – перехват
  - Command – обработка
  - Service to Worker микроархитектура
- **Бизнес логика**
  - BusinessDelegate – обработка бизнес логики
  - ServiceLocator – доступ к дата-сервисам
- **Клиентская объектная модель**
  - ValueObject – модель
  - ModelLocator – управление состоянием и обновление представления

# Модель

- Состояние приложения
- Данные с сервера или локальные данные
- Реализует Cairngorm ModelLocator
  - Singleton для хранения клиентской модели

# Представление

- Пользовательский интерфейс (GUI)
- mxml файлы и контролы
- Данные доставляются в модел через binding

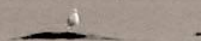


# Контроллер

- Взаимодействие между уровнями приложения
  - Events
    - `com.adobe.cairngorm.control.CairngormEvent`
  - Commands
    - `com.adobe.cairngorm.commands.ICommand`
    - `com.adobe.cairngorm.business.IResponder`
- Реализует Cairngorm ModelLocator
  - Singleton для хранения клиентской модели
- Связка команд и событий
  - Пользовательские события/смена модели

# События

- Парметры передаваемые между уровнями приложения



# Комманда

- “Service to Worker” паттерн
- Реализует Cairngorm Command Interface
- Responder Interface (опционально)
- execute() получает событие в виде аргумента

# Сервис Локатор

- Singleton паттерн
- Выделяет уровень взаимодействия с данными



# Бизнес Делегат

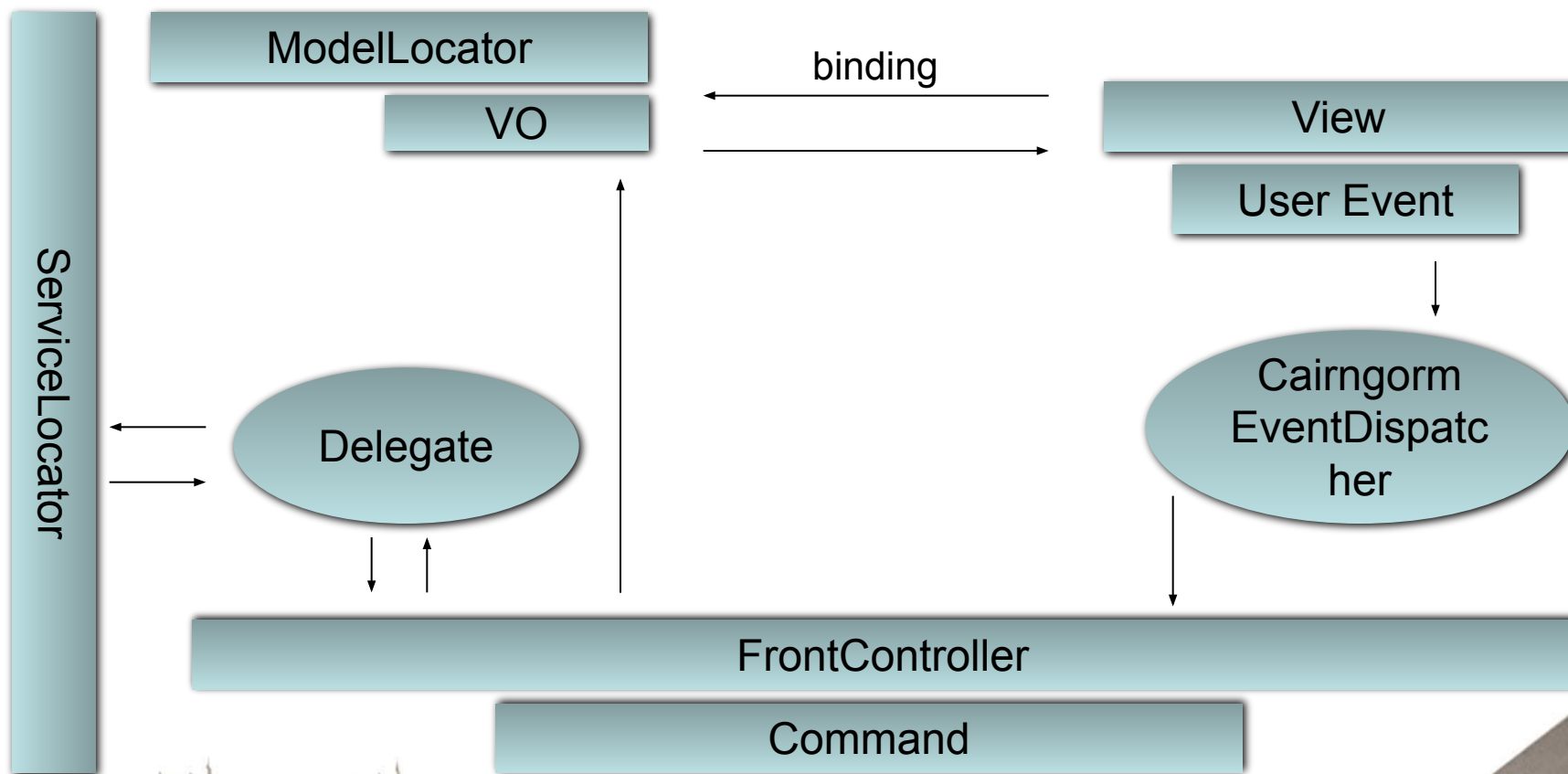
- Обращается и получает результаты от сервера используя ServiceLocator
- Позволяет команде вызывать удаленный сервис, обрабатывать результат или ошибку

# Workflow

- Регистрируем CairngormEvent в FrontController
- Вещаем событие
- FrontController – вызов execute() в нужной команде
- Command вызывает сервис используя BusinessDelegate
- Результат обрабатывается в result() команды
- Изменение модели через ModelLocator
- Изменения представления

# Диаграмма

<http://www.cairngormdocs.org/tools/CairngormDiagramExplorer.html>



# За и против

- Ясный и прямолинейный подход
- Легко оценивать трудозатраты
- Много кода при добавлении простой функциональности
- Много зависимостей







# Вопросы?

**Павел Кожин**

[vertex.mailbox@gmail.com](mailto:vertex.mailbox@gmail.com)

<http://www.riapriority.com/vertex>

Exigen Services

