

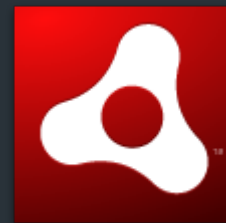


mate  
flex framework



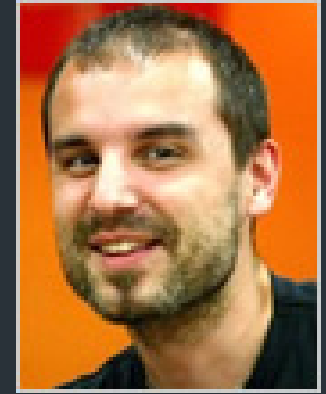
# Mate Flex Framework

Tomáš Lehuta / Flexets  
[tomas@flexets.com](mailto:tomas@flexets.com)



# Tomáš Lehuta (lharp)

Senior RIA Developer / Flexets



Info:

- vek 33 rokov
- absolvent FEI STU v Bratislave (Soft. Ing.)

Znalosti:

- Web design - HTML/JS/CSS (4 roky)
- Server-side - PHP/SQL (5 rokov)
- Client-side (RIA) - Flash/Flex/AIR/AS3 (7 rokov)

Skúsenosti:

- Goalscape (<http://www.goalscape.com>)



# Čo je MVC?

## MVC = Model/View/Controller

- oddelenie dátovej, prezentačnej a interaktívnej logiky

### Model = dátová logika

- pracuje len s dátami a definuje všetky operácie
- vo Flexe: AS3 EventDispatcher

### View = prezentačná logika

- zobrazuje dáta a posiela eventy z GUI
- vo Flexe: MXML/AS3 UIComponent

### Controller = interaktívna logika

- odchyťava a riadi interaktivitu užívateľa v GUI
- vo Flexe: AS3 EventListener

# Mate Flex Framework

## Overview

- tag-based event-driven framework = **Flex only!**
- modularita a flexibilita = **MXML Tagy**
- stromová štruktúra DisplayListu = **Event Bus**
- využitie mechanizmov:
  - **Event Bubbling** - komunikácia medzi komponentami aplikácie
  - **Data Binding** - automatické data updates
  - **Dependency Injection** - zníženie závislostí v kóde a lepšia testovateľnosť (Unit Testing)
- cachovanie inštancií objektov = **Singletons free!**
- podpora lokálnych controllerov = **Flex Modules**
- prehľadná architektúra aplikácie = **maintainability**

# Architektúra Mate Aplikácie

**Model** = ľubovoľné data objekty

- natívny Flex data binding medzi Model a View
  - public data properties: [Bindable]
  - automatické notifikácie pri zmene dát
- nezávislosť od Controller a View

**View** = Flex UI komponenty (MXML/AS3)

- dispatching custom eventov a event bubbling
- využitie techniky Code-behind:
  - **MXML** - rozmiestnenie UI komponentov
  - **AS3** - implementácia interaktivity v UI

**Controller** = EventMap Mate objekty

- MXML štruktúra a Mate MXML Tagy
- natívny AS3 event handling (built-in & custom eventy)
- globálne a lokálne event handlers

# Príklad - MateApplication

## 1. MainView

- užívateľské rozhranie aplikácie

## 2. MainController

- riadiaci modul aplikácie
- extends *EventManager* (alebo *LocalEventManager*)
- *EventHandlers* tag: *MethodInvoker*
- *Injectors* tag: *ObjectBuilder* a *PropertyInjector*

## 3. MainModel

- dátový model aplikácie
- data binding properties

# Komunikácia v Mate Aplikácii

## 1. MainView

- interaktivita užívateľa v GUI
- dispatching custom eventov a posielanie dát

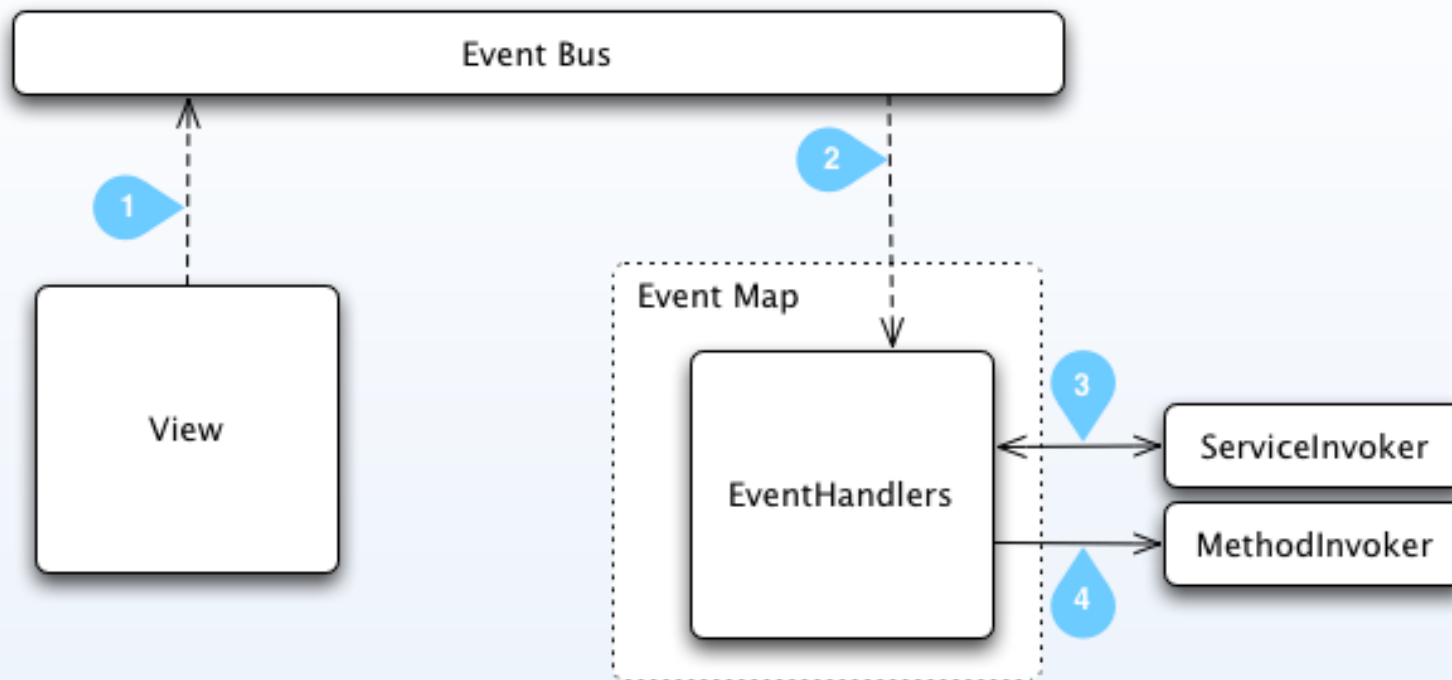
## 2. MainController

- zachytenie konkrétnych eventov
- volanie metód objektov príp. dátových služieb
- dispatching ďalších eventov

## 3. MainModel

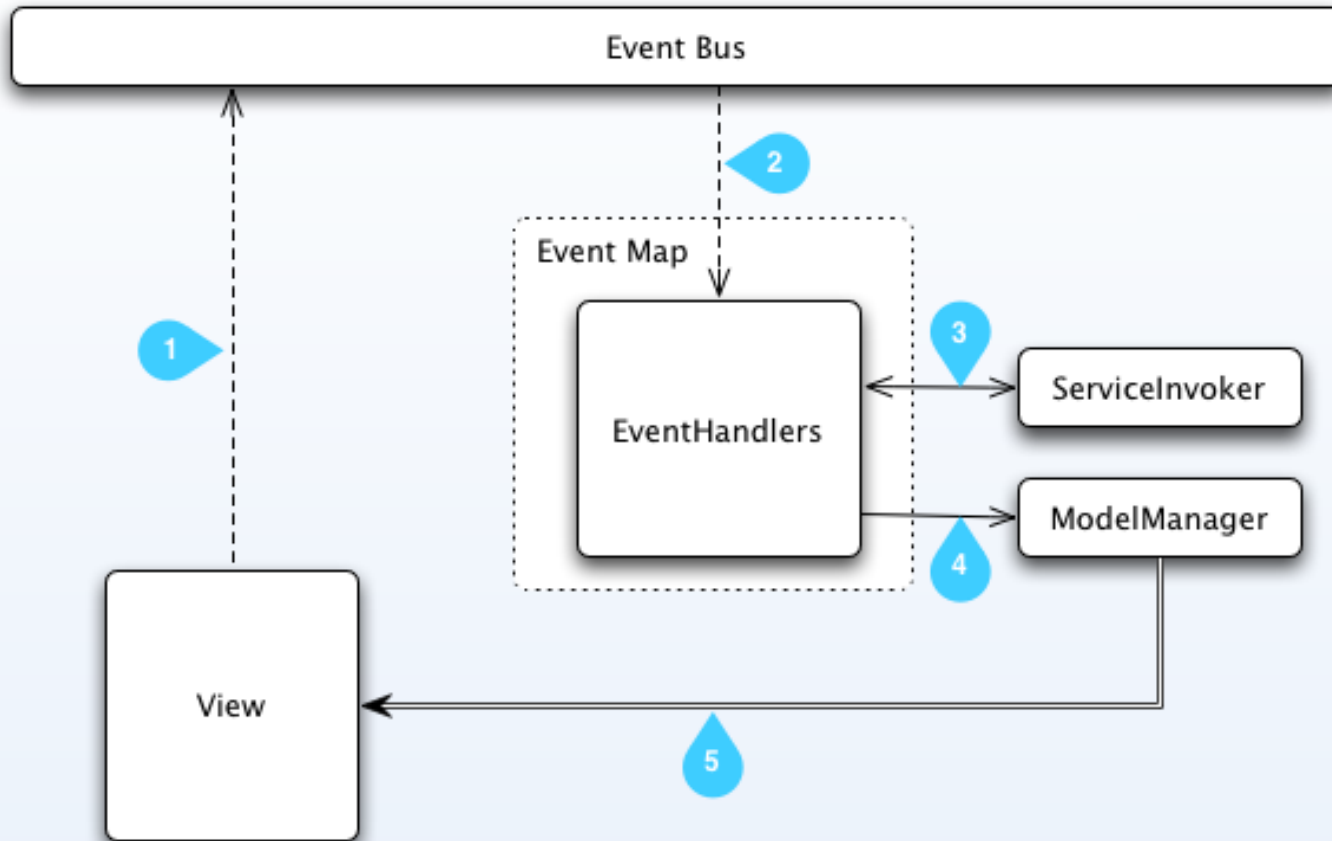
- načítanie požadovaných dát
- notifikácia zmien cez data binding

## One-way communication: from views to business logic



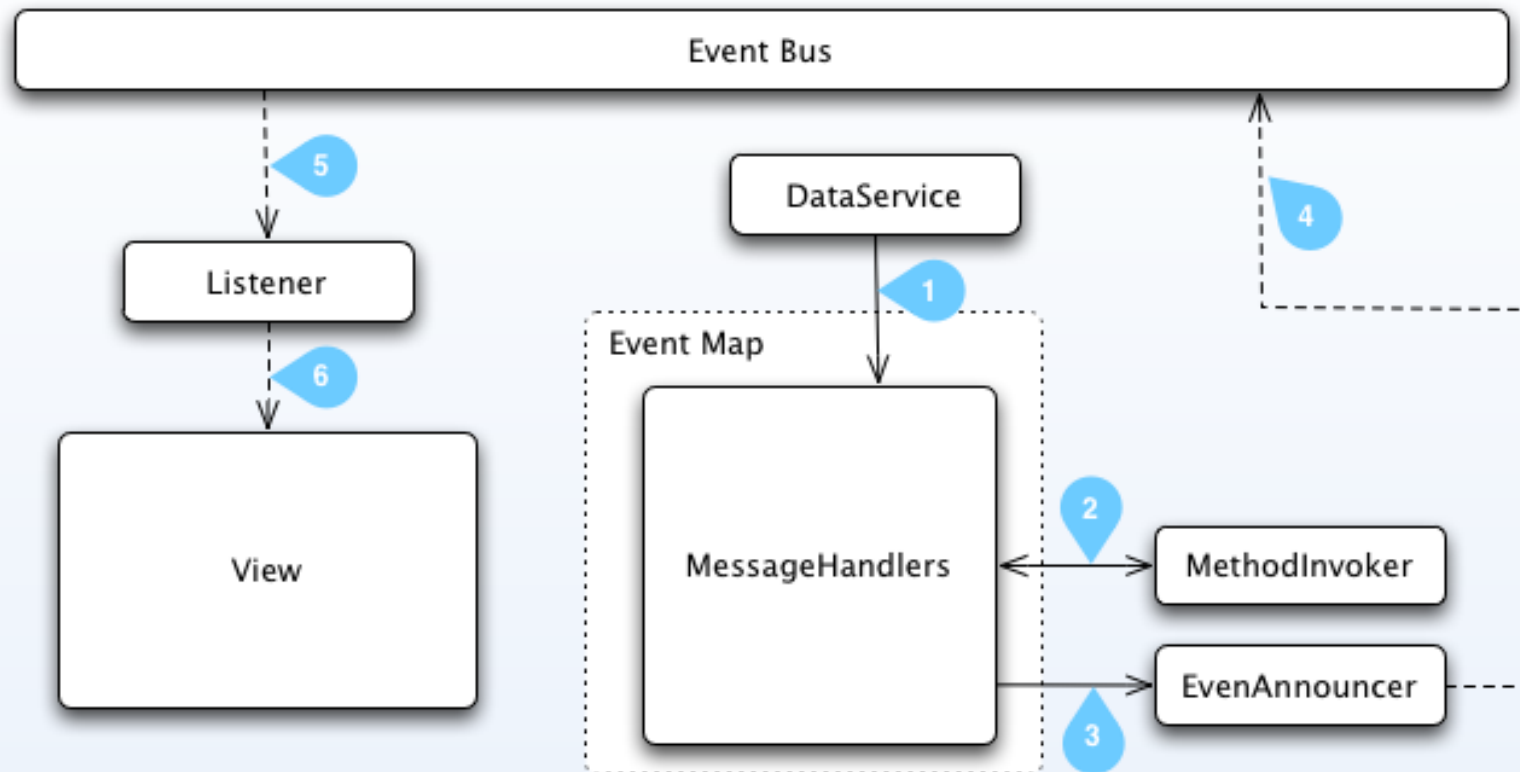
- 1 View dispatches an event that bubbles up (`bubbles = true`) or view dispatches an event using the **Dispatcher** tag
- 2 The event arrives to an **EventHandlers** block in the EventMap that is registered to listen to this event type
- 3 Inside the **EventHandlers**, a list of actions is executed in order. In this example, a server call is made.
- 4 The second action is calling a method using a "MethodInvoker". Those actions can be replaced for any other action

## Two-way communication via model: Injecting the view



- 1** View dispatches an event that bubbles up (bubbles = true) or view dispatches an event using the **Dispatcher** tag
- 2** The event arrives to an **EventHandlers** block in the EventMap that is registered to listen to this event type
- 3** Inside the **EventHandlers**, a list of actions is executed in order. In this example, a server call is made.
- 4** A method on a **Model Manager** is executed, setting some data on the manager.
- 5** Via Bindings, the **View** listens for changes on the **Model Manger** and updates itself.

## One-way communication from business logic to views: Listener tag



- 1 A DataService pushes data to the client. The message is caught by the **MessageHandlers**
- 2 Inside the **MessageHandlers**, a list of actions is executed in order. In this example, a "MethodInvoker" is used.
- 3 After that, an Event is created by an **EventAnnouncer**
- 4 The event is dispatched
- 5 A Listener tag receives this event because it is registered to listen for this event type
- 6 The view handles this event by a method or by the event handler in the Listener tag

# Mate MXML Tagy

- **Helpers** - pomocné operácie
  - ObjectBuilder - vytváranie inštancií
  - EventAnnouncer - event dispatching
  - DataCopier - kopírovanie dát
- **Invokers** - volanie metód objektov alebo služieb
  - MethodInvocationer - metódy inštancií objektov
  - InlineInvoker - statické metódy objektov
  - CommandInvoker - Command pattern (á la Cairngorm)
  - ServiceInvoker - HTTP service metódy (GET/POST, AMF, WSDL/SOAP)
  - DelegateInvoker - metódy business service delegátov
- **Handlers** - handlovanie eventov a messagov
  - EventHandlers - handlovanie eventov v aplikácii
  - MessageHandlers - handlovanie messagov z FMS (server push)
- **Injectors** - injektovanie properties alebo listenerov
  - PropertyInjector - object properties
  - ListenerInjector - event listeners



mate  
flex framework



## Priestor pre diskusiu

Mate Web: <http://mate.asfusion.com>

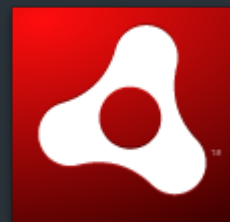
Examples: <http://mate-examples.googlecode.com>



**Flexets**

LET'S FLEX YOUR BUSINESS

LET'S FLEX YOUR BUSINESS



# Ďakujem za pozornosť

Tomáš Lehuta / Flexets

Email: [tomas@flexets.com](mailto:tomas@flexets.com)

Twitter: <http://twitter.com/lharp>

Blog: <http://flexets.com/blog>

Slides: <http://tinyurl.com/yhkso57>