



the  
**POWER**  
of  
**JAVA™**



JavaOne  
Part of the Oracle and Sun Microsystems

# RAD for the Java™ Platform Web Tier: Frameworks Panel Discussion

**Roger Kitain,  
Kevin Osborn**  
Sun Microsystems

**Geert Bevin**  
Uwyn [RIFE]

**Graeme Rocher**

**Chris Nelson**  
Fusion Alliance [Trails]

**Maxim [Grails]**  
TS-1664

# Goal

Learn the differences, similarities, pros and cons of three rapid web application development technologies.

# Agenda

Grails

RIFE

Trails

Open Discussion

Summary

# Agenda

**Grails**

RIFE

Trails

Open Discussion

Summary

# Grails: Overview

- New MVC framework inspired by:
  - Convention over Configuration
  - Don't Repeat Yourself (DRY)
  - Ruby on Rails
- Built on solid foundations:
  - Spring IoC, MVC and WebFlow
  - Hibernate
  - SiteMesh
- The first truly Rails-like framework available for Java™ technology that utilizes a dynamically-typed language: Groovy

# Grails: Key Advantages

- **Agility**—Save and reload changes immediately
- **Dynamic**—Dynamically registered persistence and finder methods, no need to extend or configure
- **Code Generation**—Scaffolding of views and controllers based on application domain
- **Power and Simplicity**—Groovy leveraged on all tiers, advanced features still available
- **Integration**—Java Platform software, J2EE and the Java VM, Legacy DB support with Hibernate and IoC with Spring

# Grails: Key Usages

- **General Usage**—As a general purpose web application framework
- **Prototyping**—Quick application prototyping
- **View/Controller Layer**—integrate with Java technology services and domain models
- **CRUD**—Rapid creation of CRUD oriented applications
- **Scripting**—To allow scripting of a Hibernate domain model
- To make the complex jobs simple and the simple jobs ludicrous!

# Agenda

Grails

**RIFE**

Trails

Open Discussion

Summary

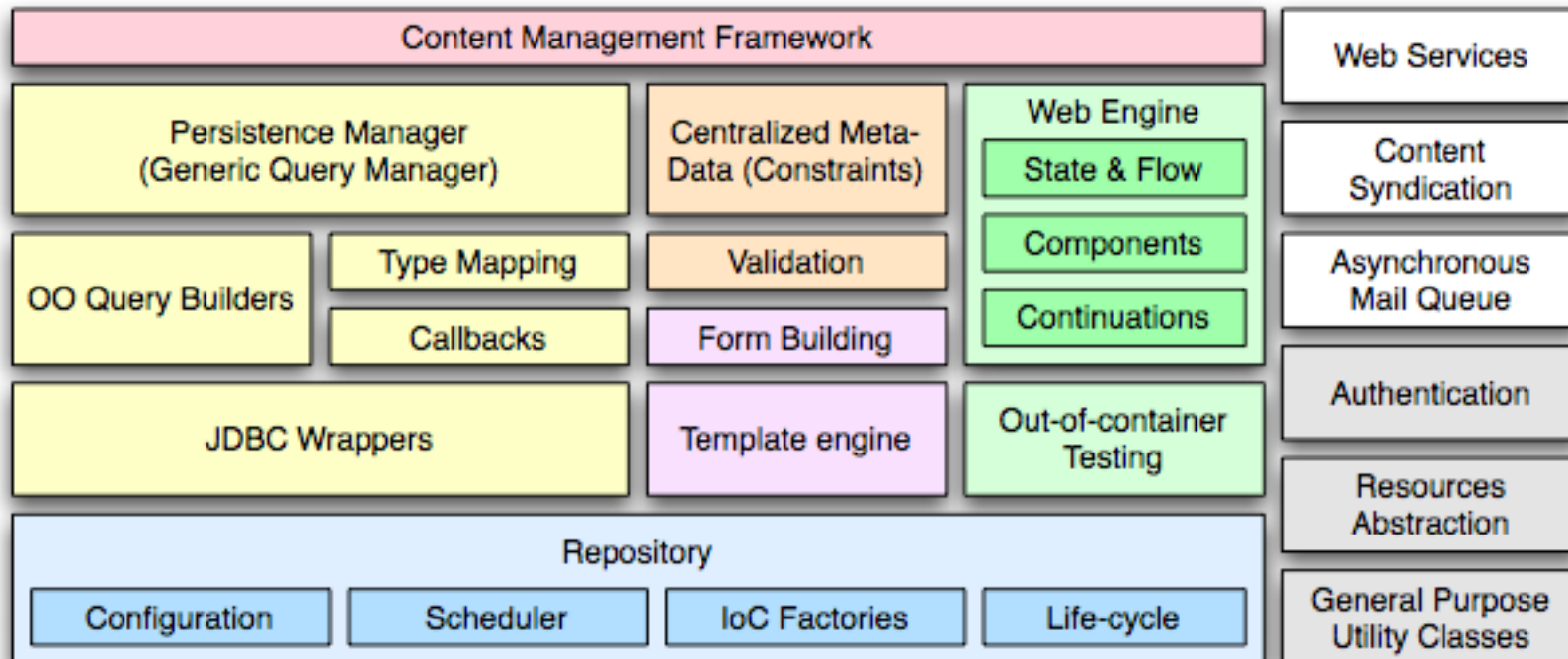


# RIFE: Overview

- Full-stack component framework to quickly and consistently develop and maintain Java-based web apps
  - Standardized structure for your application
  - Integrated layers give quick results with minimal code
  - Logic-less HTML templates, usable with standard tools
  - Components that can be reused in many contexts
  - Creating maintainable applications is our first goal
  - A lot of attention goes to code-level developer comfort
  - Creative solutions for difficult problems
  - Embraces established standards and wraps around them instead of mapping external concepts to them

# RIFE

## Architecture



# RIFE: Key Advantages

- High-level structure and flow
  - Declaration of all state transitions with data and logic
  - Clear overview of your application's building blocks
  - XML, Java language, Groovy and Annotations
  - Declaration can be automated: RIFE/Crud
- Metaprogramming
  - Domain specific API for building with larger blocks
  - High-level approach to easily achieve otherwise complex or time-consuming tasks
  - Still access to the underlying framework for customizations or finer-grained implementations

# RIFE: Key Usages

- Public websites that are developed by a team of people that want to:
  - Deliver quickly
  - Need to maintain the application over time,
  - Are interested in building up a collection of reusable web components
- Instant CRUD interfaces that should be:
  - Driven by your domain model,
  - Fully customizable, both in looks as in functionality,
  - Possible to integrate into existing web flows

# Agenda

Grails

RIFE

**Trails**

Open Discussion

Summary

# Trails Overview

- Accelerates development by removing steps
- Develop your domain model, get your app for free
- Extremely customizable
  - Per type
  - Per property
  - Or replace pieces of the framework itself
- Uses best of breed solutions
  - Spring, Hibernate, Tapestry

# Trails: Key Advantages

- Lots of functionality for free
  - Relationships: many-to-one, one-to-many, many-to-many
  - Validation, Security, i18N
- No generated source code
  - Descriptors are built at startup
  - Intelligent components render UI based on descriptors
- No wheel reinvention
  - Leverages other frameworks
  - Minimized risk: other frameworks do the “heavy lifting”

# Trails: Key Usages

- Applications that need a web UI to a persistent domain model
- Portions of a larger application
  - Admin screens
- “Good enough” software
  - Quickly show something to customer
- As a starting point
  - Customize to produce the final application



# Agenda

Grails

RIFE

Trails

**Open Discussion**

Summary

# Open Discussion

- What type of scenario is not suited for your framework? What framework would you recommend instead?
- What is the future of your framework? Is AJAX supported?
- What are some of the myths about your framework?

# Open Discussion

- What type of scenario is not suited for your framework? What framework would you recommend instead?
- **What is the future of your framework?  
Is AJAX supported?**
- What are some of the myths about your framework?

# Open Discussion

- What type of scenario is not suited for your framework? What framework would you recommend instead?
- What is the future of your framework? Is AJAX supported?
- **What are some of the myths about your framework?**

# Agenda

Grails

RIFE

Trails

Open Discussion

**Summary**

# Summary

- Grails, RIFE and Trails
  - RAD frameworks for web application development
  - Inspired by Ruby On Rails
  - CRUD frameworks
  - Open source

# For More Information

- BOF-2521: Rapid Web Application Development With Grails
- BOF-2450: Cutting-Edge Productivity With RIFE
- BOF-2946: Trails In Depth



the  
**POWER**  
of  
**JAVA™**



JavaOne  
Part of the Network and Business Solutions

# RAD for the Java™ Platform Web Tier: Frameworks Panel Discussion

**Roger Kitain,  
Kevin Osborn**  
Sun Microsystems

**Geert Bevin**  
Uwyn [RIFE]

**Graeme Rocher**

**Chris Nelson**  
Fusion Alliance [Trails]

**Maxim [Grails]**  
TS-1664