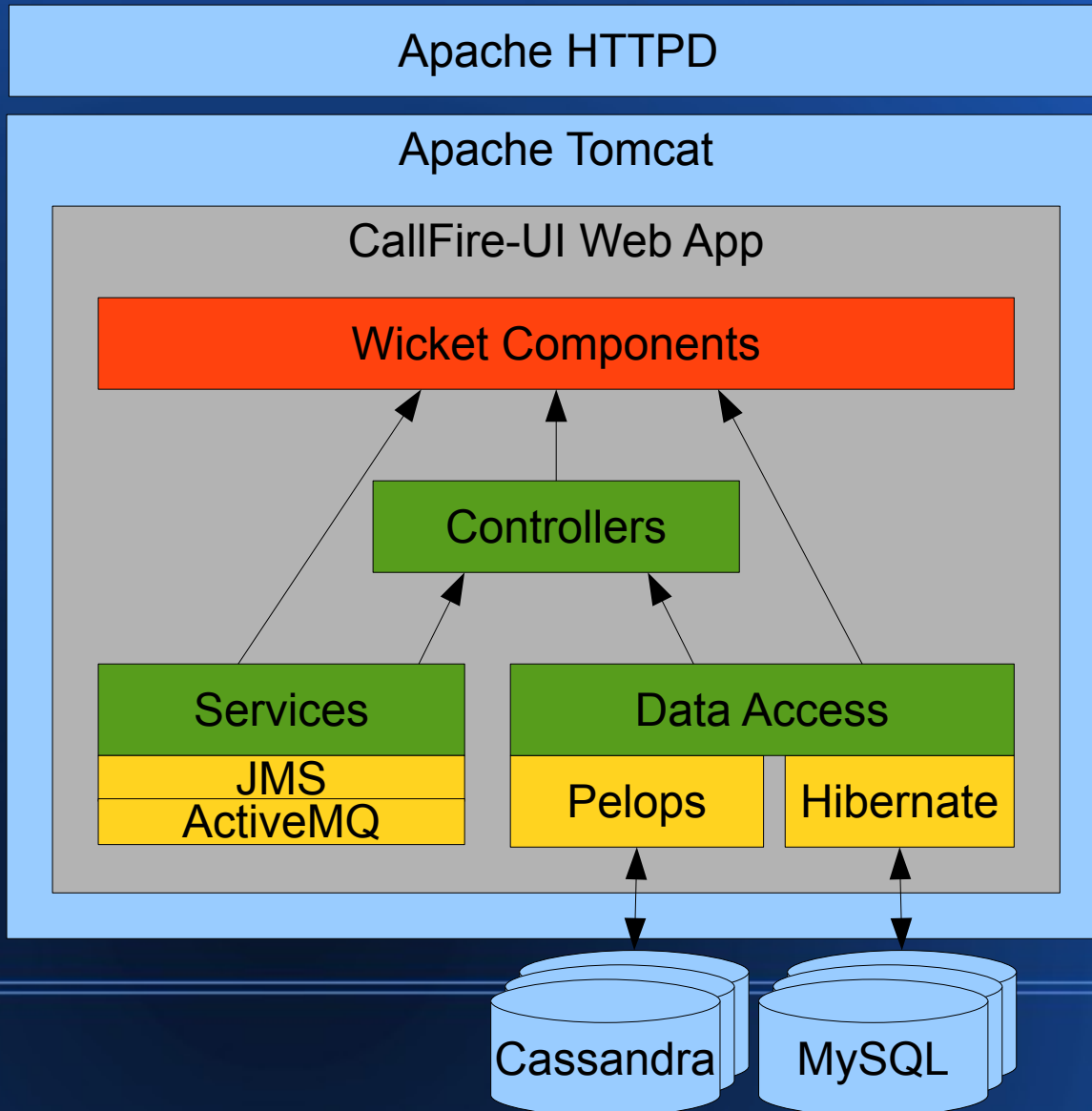


Introduction to Apache Wicket

Dan Retzlaff, VP of Engineering, CallFire – Santa Monica Java Users Group, 9/27/11

- Wicket at CallFire
- Wicket Features
 - Component-Oriented
 - Separation of Concerns, I18N, Events, Sessions
 - Back Button Support
 - Stateful Components, IoC, Models
 - Markup Interaction, AJAX Requests
- Calculator Demo

Wicket at CallFire



- `@Inject` – Guice DI
- `@Transactional` – Guice AOP
- `@Entity` – Hibernate (OSIV)
- `@ServiceQueue` – CF-JMS

Telephony

- Asterisk, FreeSwitch, OpenSIPS

Development

- Eclipse, JRebel
- Git, Maven, Jenkins
- JUnit, EasyMock, Selenium

Component-Oriented

- Model-View-Controller – Struts, Spring MVC
 - Configuration maps URLs to controller methods
 - Controller exposes model data to view templates
- Component-Oriented – Wicket, Tapestry
 - Response is a tree of Java objects/components
 - Components maintain their own state, children
 - Components control the source of their markup

Separation of Concerns, I18N, Events, Sessions

```
<wicket:panel>
  <span wicket:id="name">User Name</span> |
  <a href="#" wicket:id="logout">
    <wicket:message key="logout">Logout</wicket:message>
  </a>
</wicket:panel>
```

LogoutPanel.html

```
public class LogoutPanel extends Panel {
    public LogoutPanel(String id) {
        super(id);
        add(new Label("name", MySession.get().getUser().getName()));
        add(new Link("logout") {
            public onClick() {
                MySession.get().invalidate();
            }
        });
    }
}
```

LogoutPanel.java

`logout=Logout` LogoutPanel.properties

`logout=Salir` LogoutPanel_es.properties

Back Button Support

- Page state is *versioned*, not just session key/value.
- Page object graph serialized after each request.
- URLs include page ID, enabling the back button.
- Developers need to mind what gets serialized.

Stateful Components, IoC Integration, Models

```
public class InboxPanel extends Panel {
    private int pageNumber;
    @Inject private InboxDao inboxDao;

    public InboxPanel(String id) {
        super(id);

        add(new Link("nextPage") {
            public onClick() {
                ++pageNumber;
            }
        });

        IModel<List<Message>> messages = new AbstractReadOnlyModel() {
            public List<Message> getObject() {
                return inboxDao.getPageOfMessages(
                    MySession.get().getUser(), pageNumber);
            }
        };

        add(new ListView<Message>("message", messages) {
            protected void populateItem(ListItem<Message> item) {
                item.add(new Label("subject", item.getModelObject().getSubject()));
            }
        });
    }
}
```

Stateful Components, IoC Integration, Models

```
public class InboxPanel extends Panel {
    private int pageNumber;
    @Inject private InboxDao inboxDao;

    public InboxPanel(String id) {
        super(id);

        add(new Link("nextPage") {
            public onClick() {
                ++pageNumber;
            }
        });

        IModel<List<Message>> messages = new AbstractReadOnlyModel() {
            public List<Message> getObject() {
                return inboxDao.getPageOfMessages(
                    MySession.get().getUser(), pageNumber);
            }
        };

        add(new ListView<Message>("message", messages) {
            protected void populateItem(ListItem<Message> item) {
                item.add(new Label("subject", item.getModelObject().getSubject()));
            }
        });
    }
}
```


Stateful Components, IoC Integration, Models

```
public class InboxPanel extends Panel {
    private int pageNumber;
    @Inject private InboxDao inboxDao;

    public InboxPanel(String id) {
        super(id);

        add(new Link("nextPage") {
            public onClick() {
                ++pageNumber;
            }
        });

        IModel<List<Message>> messages = new AbstractReadOnlyModel() {
            public List<Message> getObject() {
                return inboxDao.getPageOfMessages(
                    MySession.get().getUser(), pageNumber);
            }
        };

        add(new ListView<Message>("message", messages) {
            protected void populateItem(ListItem<Message> item) {
                item.add(new Label("subject", item.getModelObject().getSubject()));
            }
        });
    }
}
```


Stateful Components, IoC Integration, Models

```
public class InboxPanel extends Panel {
    private int pageNumber;
    @Inject private InboxDao inboxDao;

    public InboxPanel(String id) {
        super(id);

        add(new Link("nextPage") {
            public onClick() {
                ++pageNumber;
            }
        });

        IModel<List<Message>> messages = new AbstractReadOnlyModel() {
            public List<Message> getObject() {
                return inboxDao.getPageOfMessages(
                    MySession.get().getUser(), pageNumber);
            }
        };

        add(new ListView<Message>("message", messages) {
            protected void populateItem(ListItem<Message> item) {
                item.add(new Label("subject", item.getModelObject().getSubject()));
            }
        });
    }
}
```

Markup Interaction

- Easy to interact with markup from Java

```
new Label("label", "Hello, world") {  
    protected void onComponentTag(ComponentTag tag) {  
        tag.append("class", "fancy", " ");  
        super.onComponentTag(tag);  
    }  
};  
  
class FancyClassBehavior extends Behavior {  
    public void onComponentTag(Component c, ComponentTag tag) {  
        tag.append("class", "fancy", " ");  
    }  
}  
myComponent.add(new FancyClassBehavior());
```

AJAX Requests

- Many AJAX-enabled components and behaviors
 - Links, form submits, pageable tables, etc.

```
<span wicket:id="time">Mon Sep 27 19:00:00 2011</span>
```

```
<a href="#" wicket:id="link">Refresh Date</a>
```

```
final Label time = new Label("time", Model.of(new Date()));  
add(time.setOutputMarkupId(true));
```

```
add(new AjaxLink("link") {  
    @Override  
    public void onClick(AjaxRequestTarget target) {  
        time.setDefaultModelObject(new Date());  
        target.add(time);  
    }  
});
```

Calculator Demo

- Maven Quickstart

<http://wicket.apache.org/start/quickstart.html>

- Incrementing Value Link
- $A+B=C$
- Validation and Feedback
- AJAX Submit