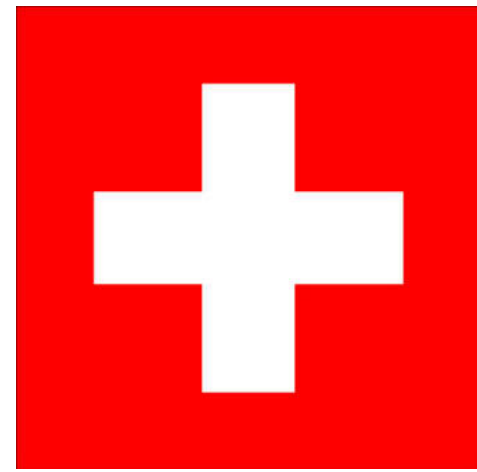


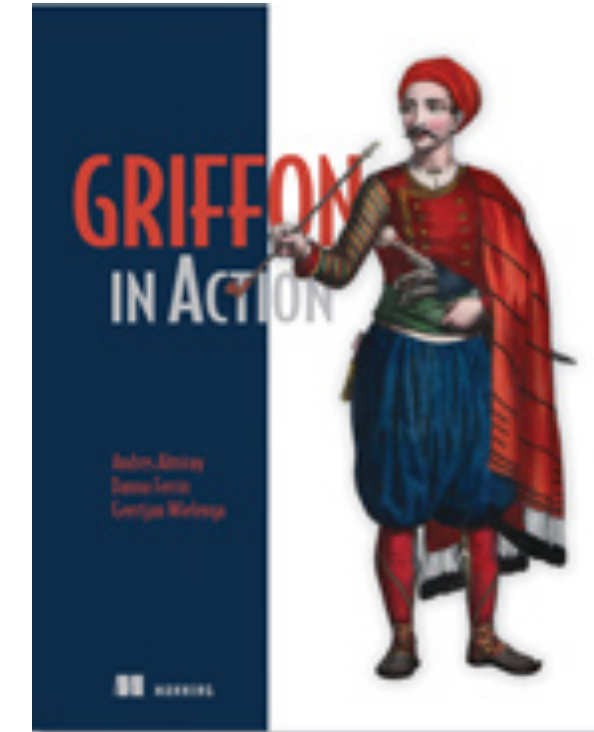
JSR377

What's up and what's next

**Andres Almiray - @aalmiray
Canoo Engineering AG**



canoo



#JSR377 - @aalmiray

PREVIOUS ATTEMPTS

JSR 193 – Client Side Container

JSR 296 – Swing Application Framework

JSR 295 – Beans Binding

JSR 296 had the following goals:

- application life cycle

- localized resources (and injection)

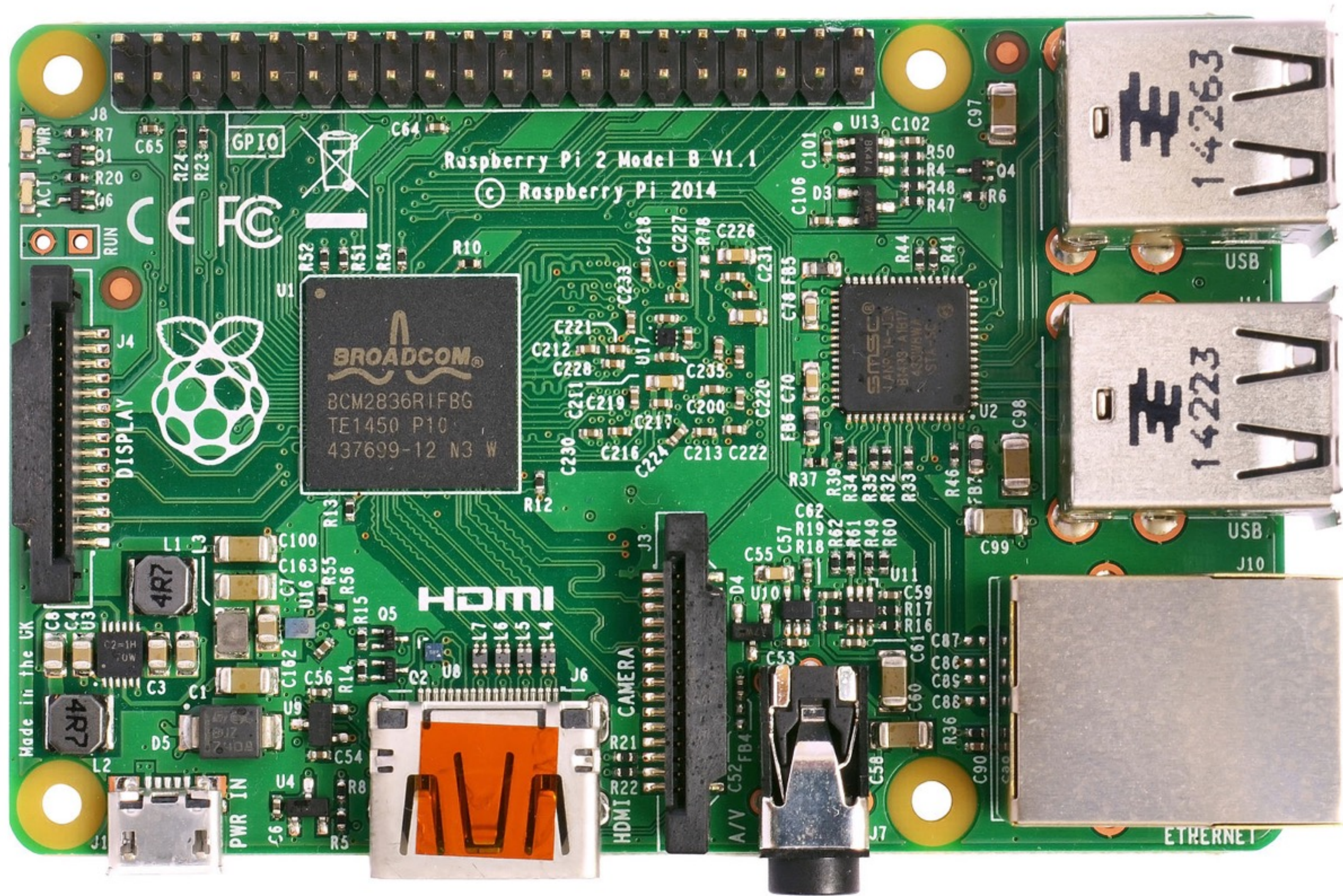
- persisted session state

- loosely coupled actions

- targeted only Swing for obvious reasons

WHICH UI TOOLKIT?





[https://en.wikipedia.org/wiki/File:Raspberry Pi 2 Model B v1.1 top new %28bg cut out%29.jpg](https://en.wikipedia.org/wiki/File:Raspberry_Pi_2_Model_B_v1.1_top_new_%28bg_cut_out%29.jpg)

#JSR377 - @aalmiray

WHAT'S OUT THERE

Eclipse 4 Platform, NetBeans

Griffon, Basilisk, Gluon Particle,

DataFX, JacpFX, MvvmFX, JVx,

JRebirth, and more ...

COMMON FEATURES

Many of the listed frameworks offer the following capabilities implemented in different ways:

application life cycle

dependency injection

localized resources (and injection)

event system

persisted session state

centralized error management

loosely coupled actions

extension points via plugins

TARGET ENVIRONMENT

All of the listed frameworks support the Desktop as target environment.

Only a few can be used in an Embedded environment (where Java SE is supported).

Embedded Java UI applications can be built as applications that target the Desktop; even share their codebase.

GOALS

Target Desktop and Embedded environments

Support several toolkits

Be an standalone JSR, i.e, no need to include it in the JDK

Leverage existing JSRs:

- JSR 330 – Dependency Injection

- JSR 365 – Event bus (from CDI 2.0 ?)

CORE FEATURES

application life cycle

dependency injection

localized resources (and injection)

event system

configuration

centralized error management

loosely coupled actions

extension points via plugins

MVC artifacts

OUT OF SCOPE V1

Runtime:

persisted session state

artifact introspection API

UI toolkit updates

Buildtime:

test support

deployment

API

Application Phases

```
package javax.application;

public enum ApplicationPhase {

    INITIALIZE,

    STARTUP,

    READY,

    MAIN,

    SHUTDOWN

}
```

Application Lifecycle

```
package javax.application;

public interface Application {

    void initialize();

    void startup();

    void ready();

    ExitState shutdown();

    boolean canShutdown();

    void addShutdownHandler(@Nonnull ShutdownHandler handler);

    void removeShutdownHandler(@Nonnull ShutdownHandler handler);

}
```

Application Lifecycle

```
package javax.application;
public interface ShutdownHandler {
    boolean canShutdown(Application application);
    void onShutdown(Application application);
}
```

```
package javax.application;
public interface ExitState {
    int exitCode();
    boolean canShutdown();
}
```

Application Properties

```
package javax.application;

public interface Application {

    Configuration getConfiguration();

    ApplicationPhase getPhase();

    Locale getLocale();

    String[] getStartupArguments();

    . . .

}
```


UI Threading

Toolkit	isUIThread	runSync	RunAsync
Swing	yes	yes	yes
JavaFX	yes	no	yes
SWT	yes	yes	yes
Pivot	yes	yes	yes
Lanterna	yes	yes	no

UI Threading

```
package javax.application;

public interface ThreadingHandler {

    boolean isUIThread();

    void runInsideUIAsync(Runnable runnable);

    void runInsideUISync(Runnable runnable);

    <R> R runInsideUISync(Callable<R> callable);

    void runOutsideUI(Runnable runnable);

}
```

Internationalization

```
package javax.application;

public interface MessageSource {
    String getMessage(String key)
        throws NoSuchMessageException;

    String getMessage(String key,
                     String defaultValue);
}
```

Combined arguments:
(Object[] / List),
Locale

Resource Injection (1)

```
package javax.application;

@Retention(RetentionPolicy.RUNTIME)
@Target({ElementType.FIELD, ElementType.METHOD})

public @interface InjectedResource {

    String key() default "";

    String[] args() default {};

    String defaultValue() default "";

    String format() default "";

}
```

Resource Injection (2)

```
import javax.application.InjectedResource;
```

```
import javafx.scene.paint.LinearGradient;
```

```
public class SomeBean {
```

```
    @InjectedResource
```

```
    private LinearGradient gradient;
```

```
}
```

Resources

DesktopEmbedded Application API

<https://jcp.org/en/jsr/detail?id=377>

<https://github.com/jsr377/>

<http://jsr377.github.io/site/>

#JSR377 - @aalmiray



Thank you!

**Andres Almiray - @aalmiray
Canoo Engineering AG**