The Apache MyFaces Universe in 2012

Gerhard Petracek
Mark Struberg

APACHE
MyFaces

Agenda

- What is MyFaces?

- Why MyFaces?

  - Core

  - Extensions

Apache MyFaces is … - 1

- … a great <u>Community</u>

- … the umbrella project for
  JSF related projects

- … **Core** (JSF Implementation)

- … Add-ons for JSF based projects

  – Orchestra

  – **ExtVal** (Extensions Validator)

  – **CODI** (Extensions CDI)

  – **Ext-Scripting** (Extensions Scripting)

  – Portlet Bridge

Apache MyFaces is … - 2

- … Component Libraries
  - Tomahawk
  - Trinidad (base for ADF-Faces)
  - Tobago
- … Utils
  - Test (with Mocks)
  - Commons
  - Maven Plugins and Archetype
  - (Sandbox)

JSF and Apache MyFaces

- JSF **2** is easy

- With MyFaces

  - … Core JSF is **fast**

  - … Add-ons JSF is **awesome**

```
<html xmlns="..."><head>...</head>
 <body>
  … #{menu.entries} …
</body></html>
```
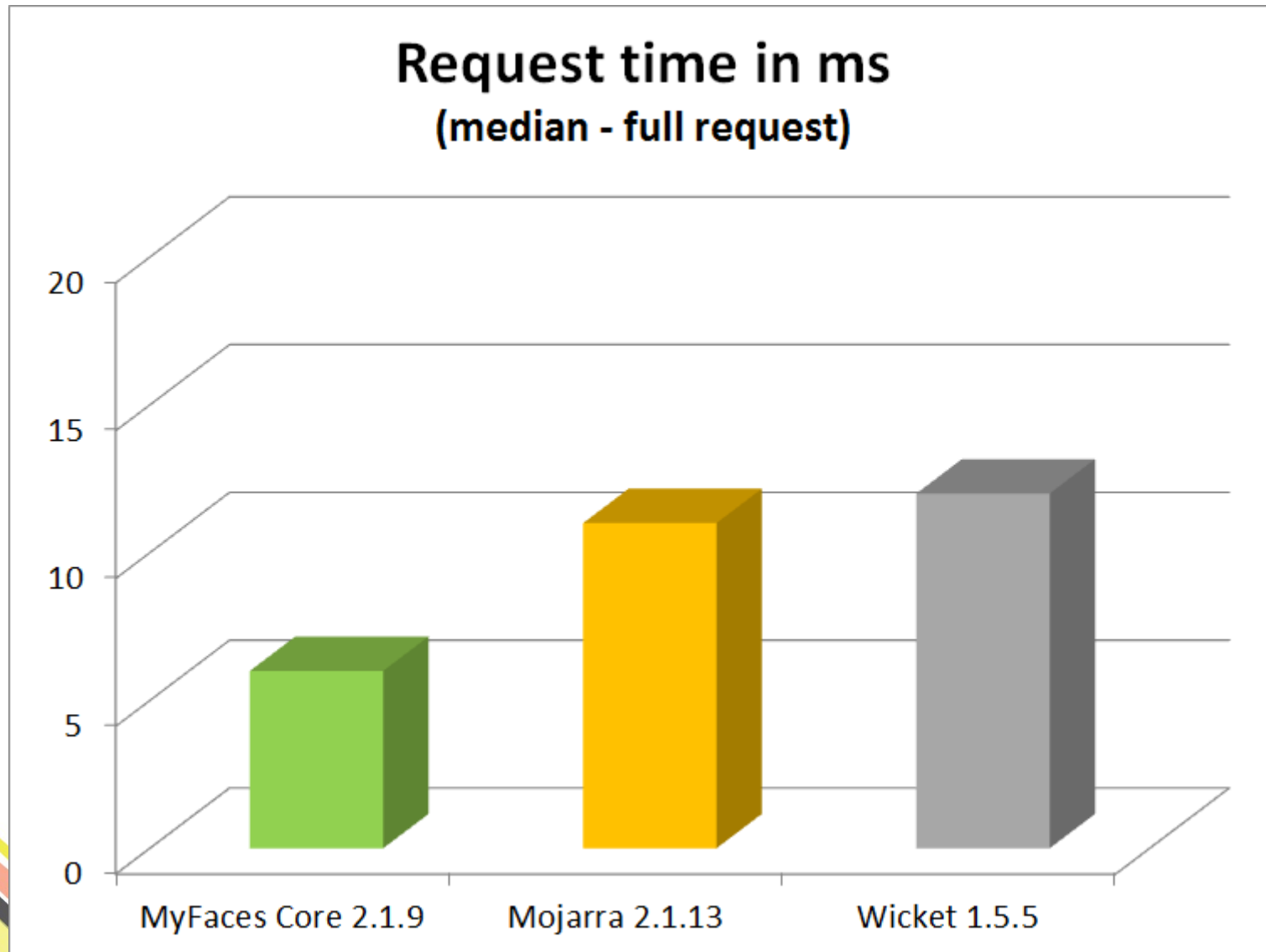
```
@Named
@WindowScoped
public class Menu
 implements Serializable  {
  //...
}
```
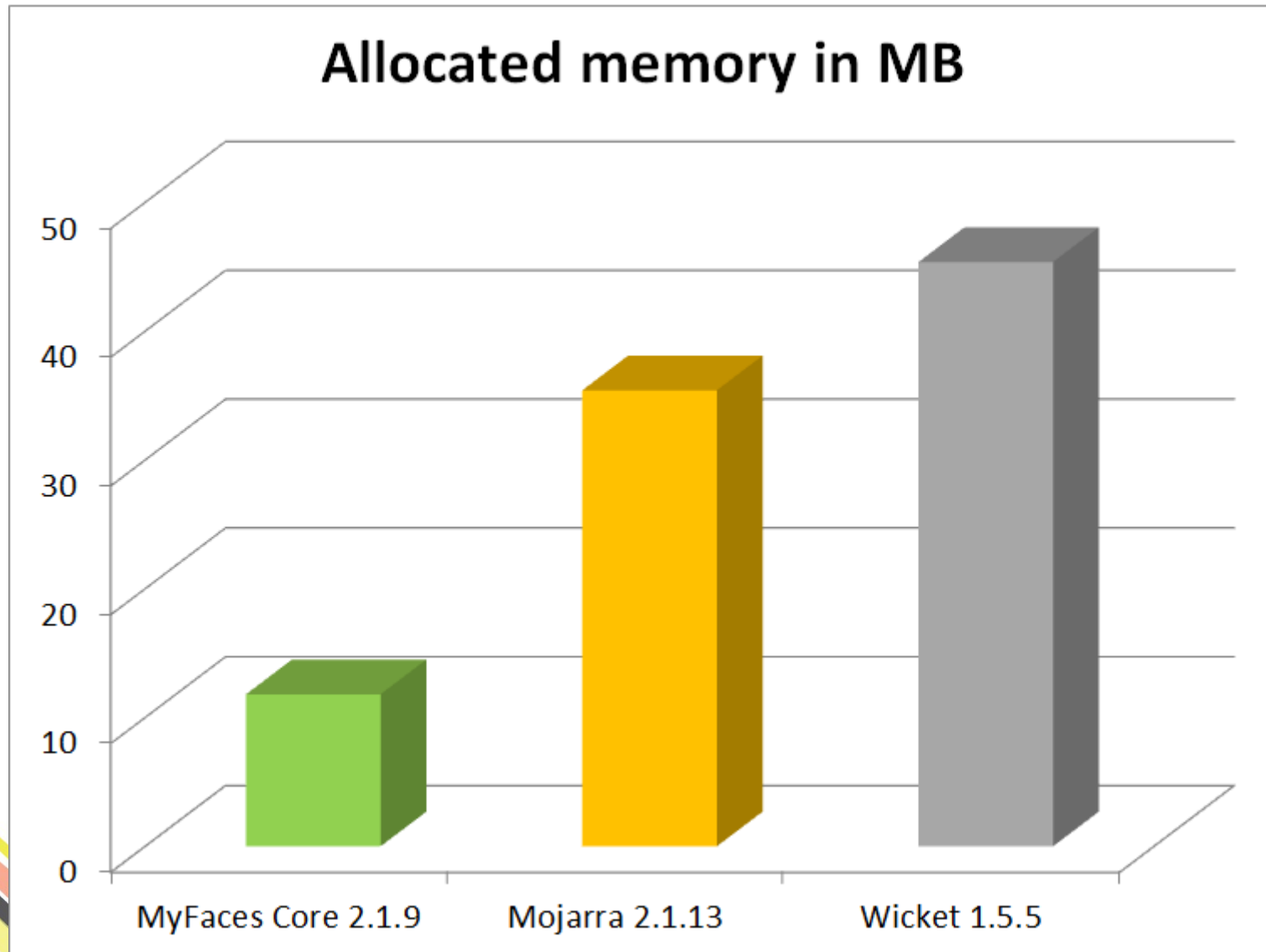
Apache MyFaces Core is ...

- ... an implementation of the JSF specification

- ... used by several application servers

- ... well tested

- ... well supported

  - Community

  - Commercial support

    Even in 2012 we had a maintenance-release of 1.1.x (8 years after the spec.-release)

- ... currently the fastest JSF implementation

- ... faster than many other web-frameworks

Request-Performance of MyFaces Core

Request time in ms
(median - full request)

MyFaces Core 2.1.9   Mojarra 2.1.13   Wicket 1.5.5

Memory Footprint of MyFaces Core

Allocated memory in MB

50	
40	
30	
20	
10	
0	

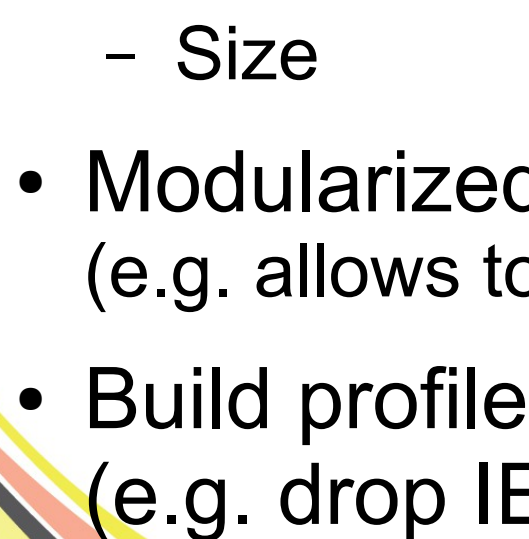| MyFaces Core 2.1.9 | Mojarra 2.1.13 | Wicket 1.5.5 |

MyFaces and Public Applications

- One of many examples: TISS

- TISS uses Tomcat with
  OpenWebBeans, OpenJPA
  MyFaces Core and MyFaces CODI

- TISS serves
  5 million full JSF requests
  (excluding resources and AJAX requests)
  for 40.000 users
  easily per day

- Most requests take less than 15ms

jsf.js of MyFaces Core

- Partial-Page-Submit
  (only submit parts which get processed)

- Advanced (dynamic) queue control
  - Delayed requests (even before JSF 2.2)
  - Timeout
  - Size

- Modularized OO JavaScript infrastructure
  (e.g. allows to replace parts in a consistent way)

- Build profiles to optimize the footprint
  (e.g. drop IE6 support)

Apache MyFaces ExtVal is ...

- … a pluggable validation-platform for constraint based JSF validation

- … compatible with JSF 1.x and 2.x

- … tested with several servers

- … **not bound** to MyFaces-Core

MyFaces ExtVal – TOP 10 - 1

- Easy cross-property validation

- Transparent JPA based validation

- Component initialization

- Constraint mapping and constraint-sources

- Advanced Bean-Validation integration
  - @Valid for properties with custom types
  - Bypass validation via action methods
    (BV groups via add-on)

MyFaces ExtVal – TOP 10 - 2

- Constraint severity
  (warn once via add-on)

- Constraint parameters
  (BV groups is a subset of it)

- Model validation (transactional via add-on)

- Validation interceptors

- DI Support
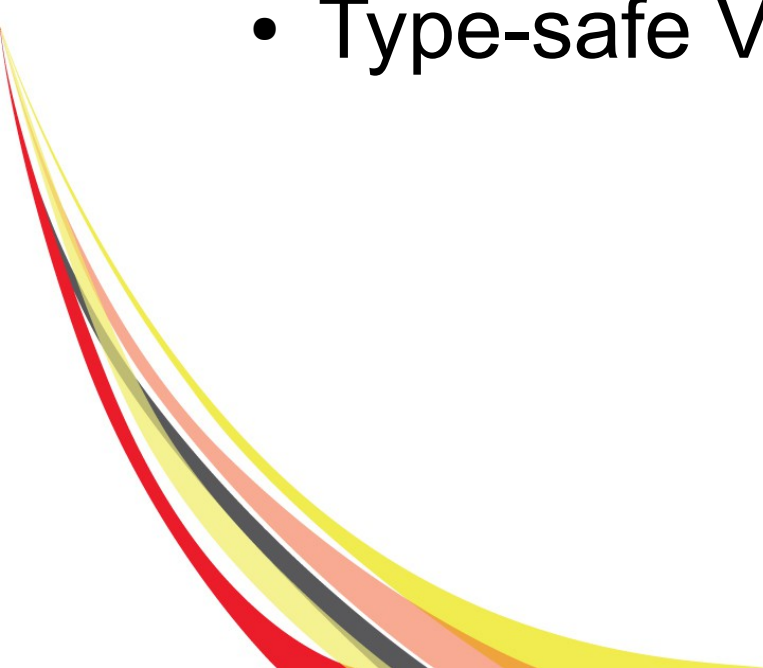 (different DI containers supported)

Apache MyFaces CODI is ...

- ... a portable CDI extension (**not** a CDI implementation)

- ... compatible with Java-EE**5**+

- ... tested on most major servers

- ... **not bound** to MyFaces-Core or OpenWebBeans

- … easy to use, extensible, fast and stable

- … optimized for large applications

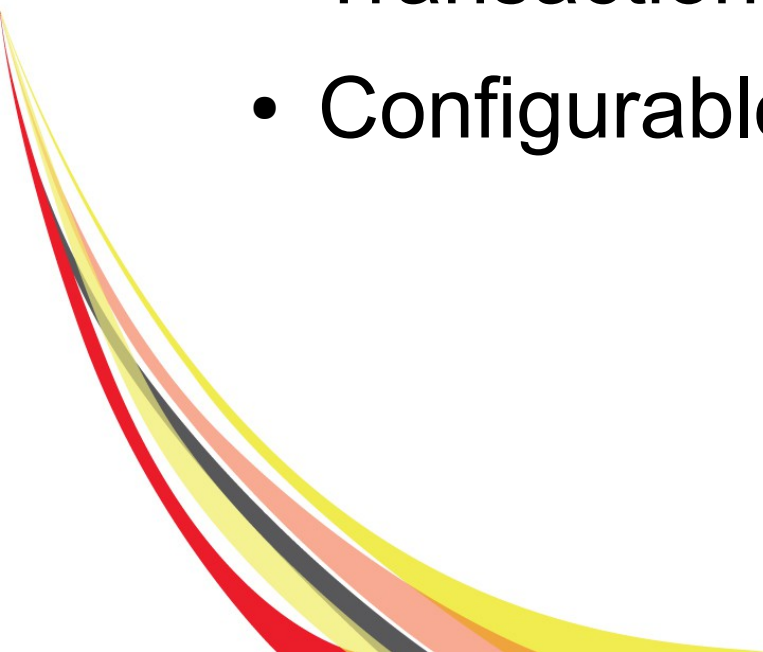- … one of few popular CDI extensions

History of MyFaces CODI

- Apache MyFaces Orchestra introduced interesting concepts in combination with Spring

- 12-2009 CDI 1.0

- 2-2010 MyFaces ExtCDI (aka CODI) started

- 11-2010 first stable release

- 12-2011 merged with Seam3 and others to Apache **DeltaSpike**

MyFaces CODI – TOP 10 - 1

- Scopes

- Conditional bean activation

- Type-safe CODI-Config

- Type-safe and extensible Project-Stages
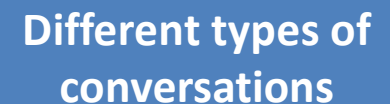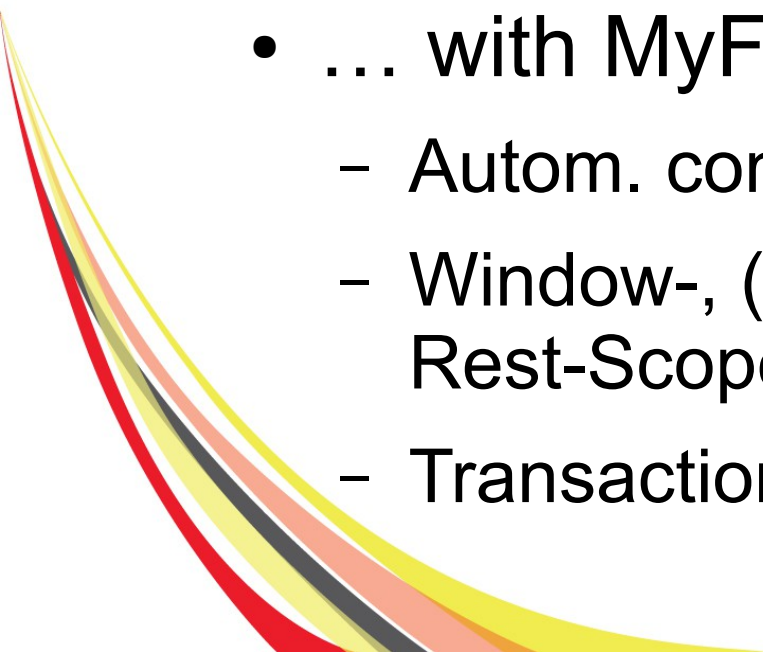
- Type-safe View-Config and Navigation

MyFaces CODI – TOP 10 - 2

- View-Controller
- Extensibility
  (Security, Interceptors, Factories, …)
- I18n
- Transactional beans without EJBs
- Configurable JPA DataSource

Scopes

- … with std. CDI and JSF
  (JSF Scopes not integrated   )
  - Request-, Session-, Application-Scope
  - View-, Flash-, Custom-, None-Scope (JSF)
  - Conversation-, Dependent, Singleton-Scope (CDI)

- … with MyFaces CODI
  - Autom. converts JSF beans to CDI beans
  - Window-, (Grouped-)Conversation-, ViewAccess-, Rest-Scope (for CDI + JSF)
  - Transaction-Scope (for CDI)

Different types of conversations

CODI Scopes by Example

- Without MyFaces CODI – e.g.:

  @SessionScoped

  public class Menu implements Serializable {

      //...

  }

- -> multiple Browser-Tabs use the same instances

- With MyFaces CODI – e.g.:

  @WindowScoped

  public class Menu implements Serializable {

      //...

  }

Destroy CODI Conversations

- ## Close a conversation
  @Inject

  private Conversation conversation;

  this.**conversation.close**();

- ## Use the Window-Context
  @Inject

  private WindowContext windowContext;

  this.**windowContext.close**();   //and #close*

JSF and Browser Windows

- JSF 2.2 ClientWindow is inspired by CODI Window-Handler and others

- CODI Window-Handler implementations

  - ServerSideWindowHandler

  - DefaultWindowHandler (URL)

  - ClientSideWindowHandler

    - Lazy Window-ID dropping

    - Intermediate windowhandler.html page

DEMO
Client-Side Window-Handler

Apache MyFaces ExtScript is ...

- ... dynamic recompilation and reloading of JSF artifacts without redeployment

- ... compatible with Groovy and Scala

- ... going to support JRuby

- ... compatible with CDI and Spring

- ... somewhere in between Mojarra-Scripting and JRebel

- ... tested with several servers

- ... currently **bound** to MyFaces Core