JSF is Alive

Jakarta Faces 4.0

Rudy De Busscher

Agenda

- Some History
  - And why JSF is still/again relevant
- What is new/changed in Faces 4.0
- ‘Quick’ tour of many basic features
  - Crash course in 30 min
  - RAD tool

Rudy De Busscher

- **Jakarta EE Expert**
  - Owner of Atbash
- **Involved in**
  - Committer in Eclipse EE4J groups (Jakarta EE)
  - Committer of MicroProfile
  - Java EE Security API Expert group member

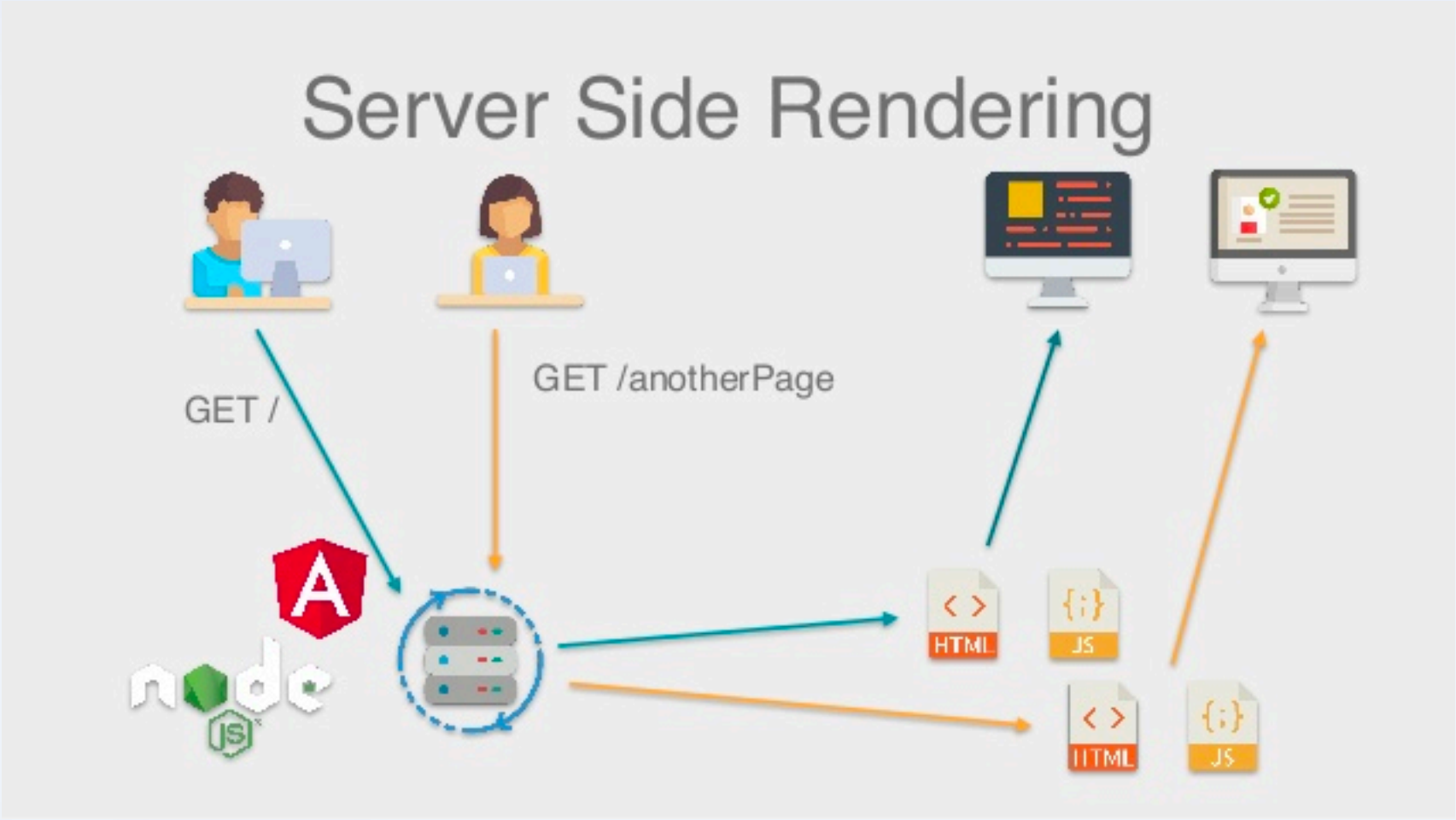

@rdebusscher


https://www.atbash.be

History

- JavaServer Faces 1.0
  - March 2004
- JavaServer Faces 2.0
  - Integration of facelets
  - XHTML based (JSP still supported)
- Constant improvements
  - Ex 2.3 -> web socket support
- Jakarta Faces
  - 3.0 -> namespace
  - 4.0 -> Several new features - Many removals (JSP, Managed beans, …)

Java EE - Jakarta EE History


Server Side Rendering
GET /
GET /anotherPage
HTML
JS
HTML
JS

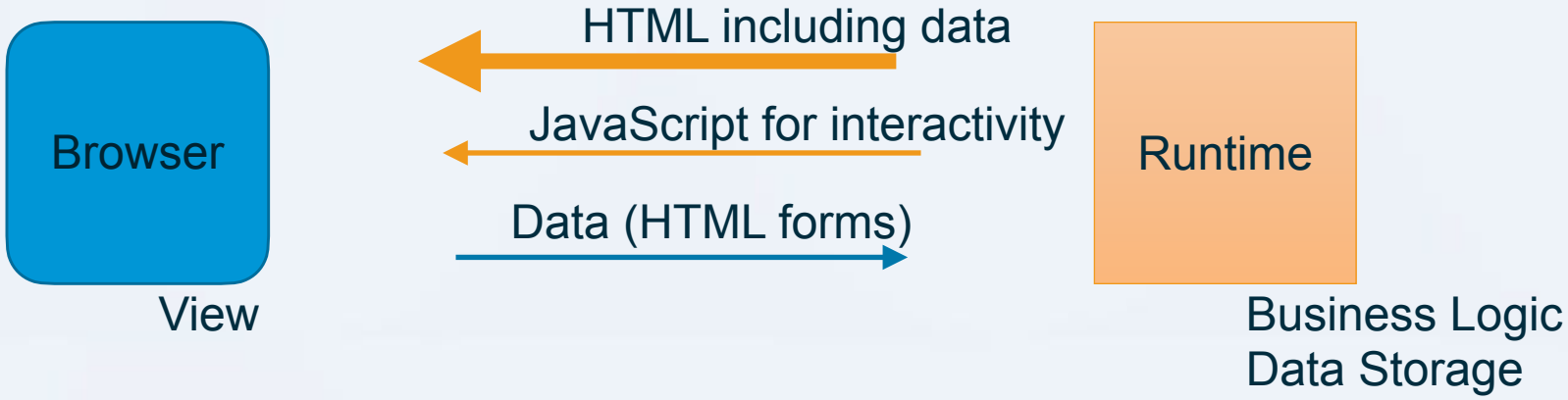
Server-side Rendering (10 years ago)

- JSF is Dead
  - Outdated
  - Slow / memory requirements
  - Difficult to learn

Server-side Rendering (Today)

- All JavaScript based framework support it (Angular, React)
  - Easier
  - Faster, better user experience
  - SEO

Server-side rendering



Client-side rendering

315 requests, 6,65 MB (transferred: 6,61 MB), 37,53 s

Developer Benefits (JSF)

- Abstraction of HTML and CSS
- Separation of concerns
  - Business logic
  - User interaction Rendering
- Integrated out of the box in platform
- Stable

- Server side

  - SEO, RAD, Bussiness logic
  - Resources

- Client side

  - Scalability
  - Slow startup, network usage, additional frameworks

Actual Problem

- HTML is not made for applications
  - Static
- Browser is not made for applications
  - Back button

- Why web apps?
  - Easy of installation

Faces 4.0

- Removed
  - JSP Support
  - Managed Bean
  - Several smaller aspects (less used)
- Multi-file upload
- ClientWindowScoped
- Programmatic views
- Extension-less mapping
- …

ClientWindow(Scoped)

- For each window/tab
- When multiple windows can be launched
  - To be able to switch between windows/tabs

RequestScoped

ConversationScoped

TransactionScoped

FlowScoped

ClientWindowScoped

ViewAccessScoped

SessionScope

ApplicationScope

Demo

Hello World

- Getting started
- PrimeFaces
- OmniFaces
- (Deltaspike)
- AJAX support
- <f:selectItemGroups> (new - through view - small)
- Multiple File upload (new - already possible with PF)

PrimeFaces

- JSF renderer
  - Can supply additional client resources like JavaScript and CSS files
- Strict separation between functionals and cosmetics
- Switch on the fly the Look&Feel through themes
- Design company based theme

OmniFaces

- Utilities
  - Nothing visually

- Some are interesting
  - Exception Handlers
  - Filters (like Gzip)
  - Advanced validation
  - …

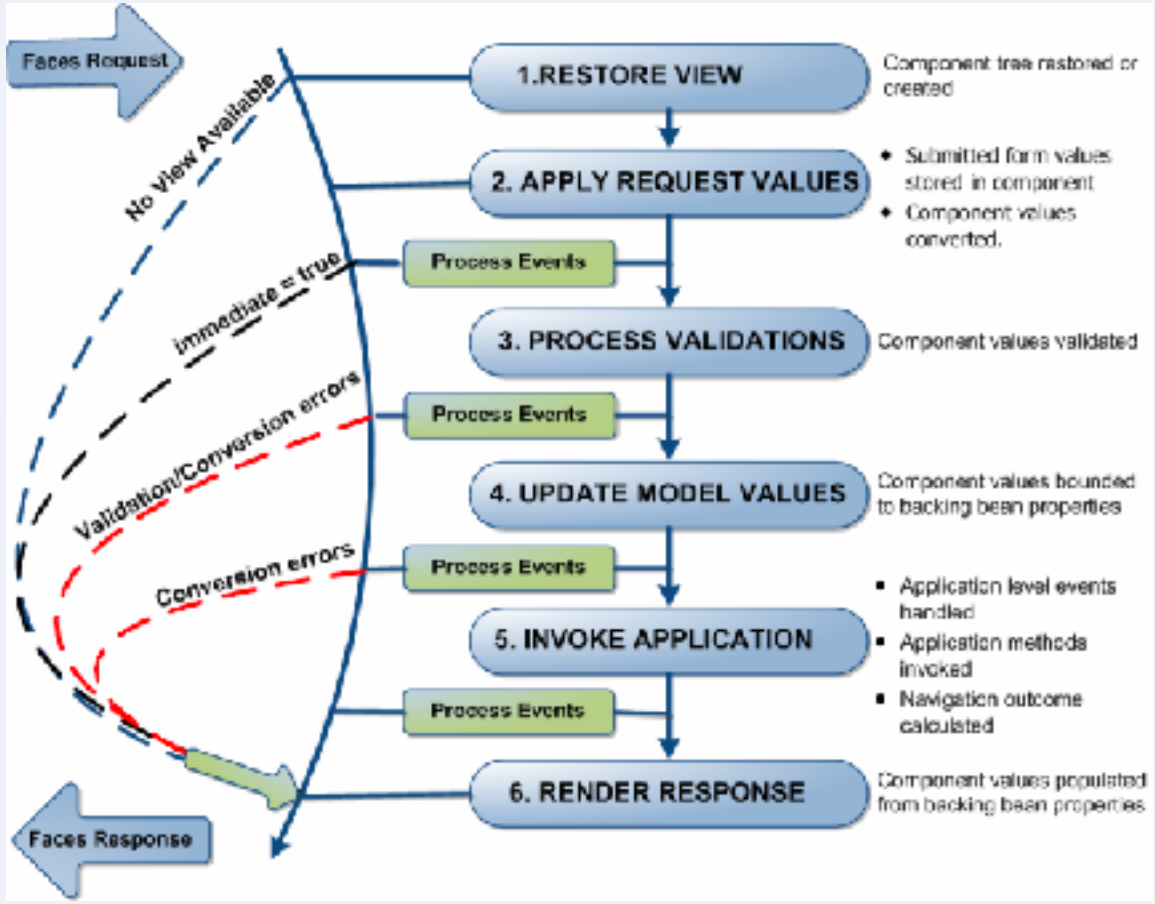
- Nothing that you can’t define yourself if needed.

Templates

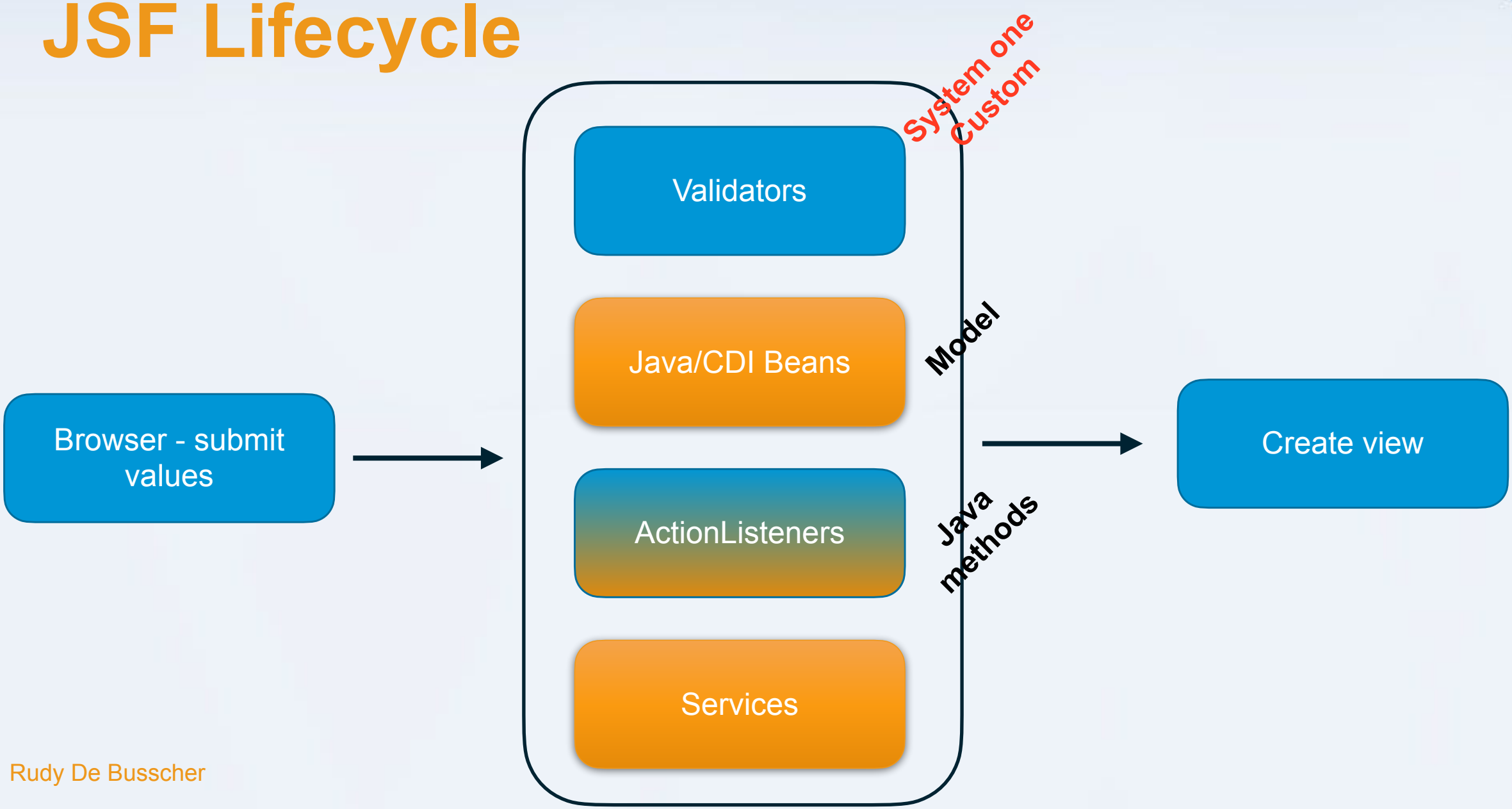
- <ui:composition>
- Organise page snippets in modular way
- Page is like code using blocks

- Composite components (macros / includes)
- <composite:interface>
- <composite:implementation>
- Preferred : Use Custom Namespace

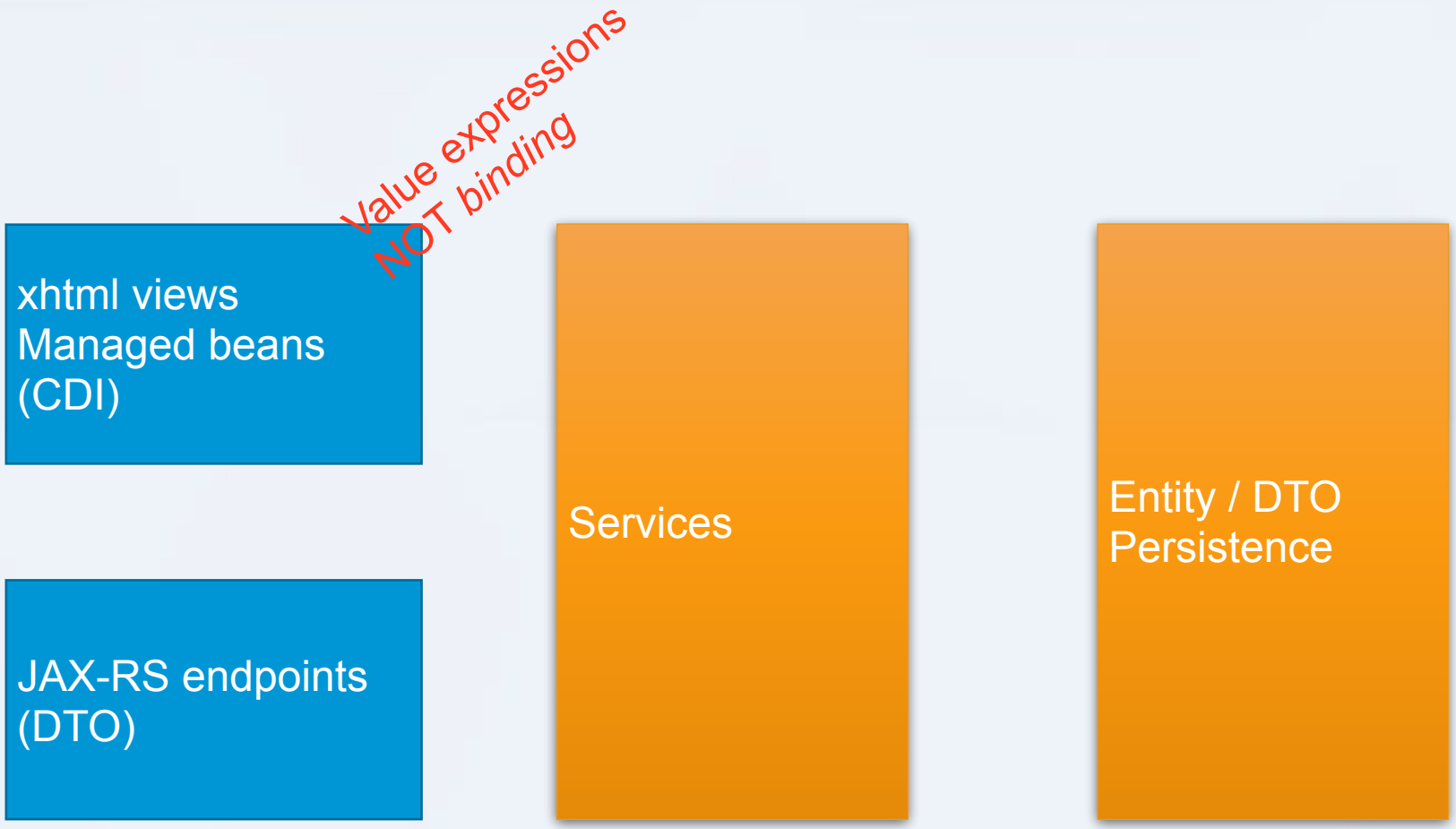
JSF Lifecycle



JSF Lifecycle



Architecture



Architecture

- Generic Exception handling
- Services -> Business logic (view neutral)
- Views (JSF, REST)

Navigation

- Page navigation
  - Use action attribute
  - Conditional navigation?
    - View can be determined by Java code.
  - URL is one step behind
    - Application vs web pages

- Extensionless URLs (new in Faces 4.0)

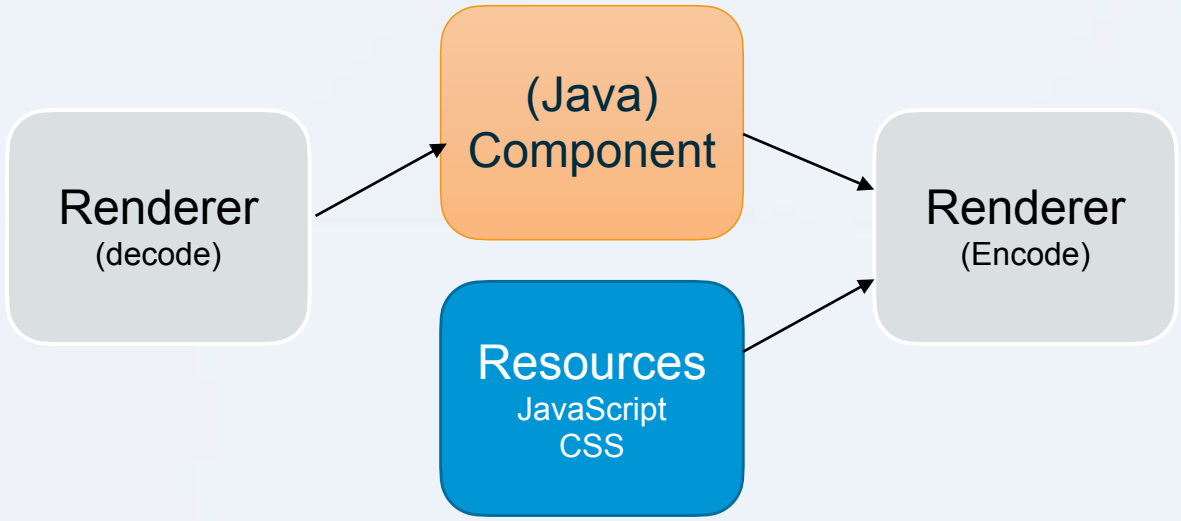
Post Redirect Get pattern

- URL is always correct
- Redirect performed for each page

- RequestScoped becomes difficult
  - Need additional view parameters
- Use with ViewAccessScoped of DeltaSpike.

Custom Components

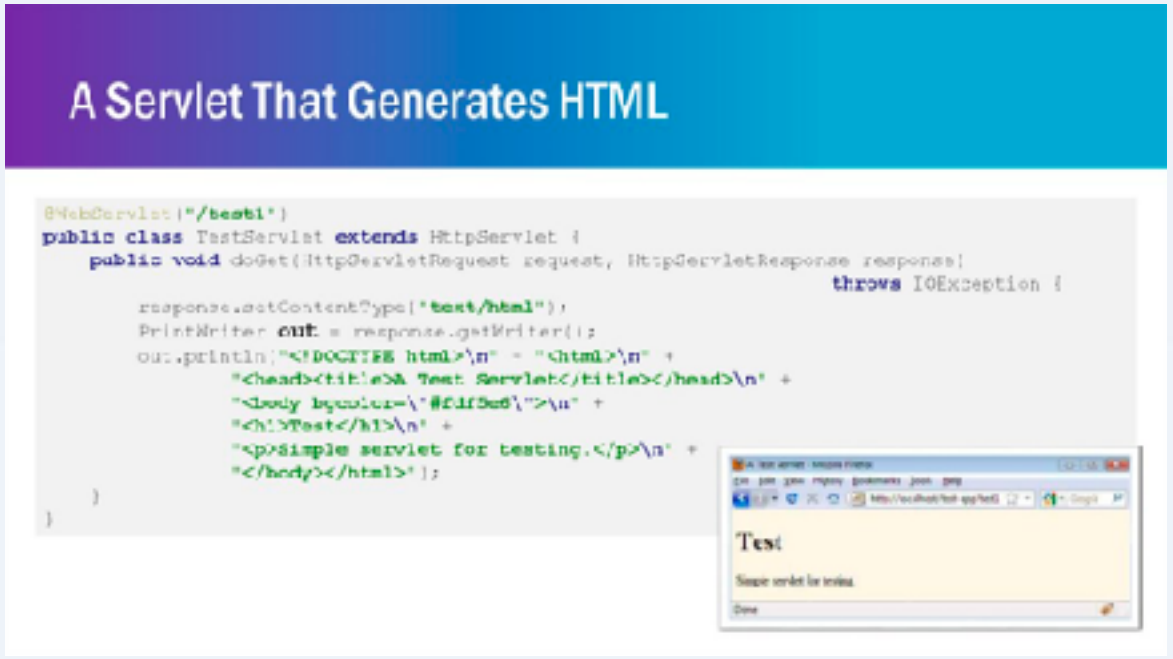
When Composite Components are not enough



Config
tag

Programmatic view

- New in faces 4.0
- Good idea?
- Same as servlet that prints out HTML
- No separation!

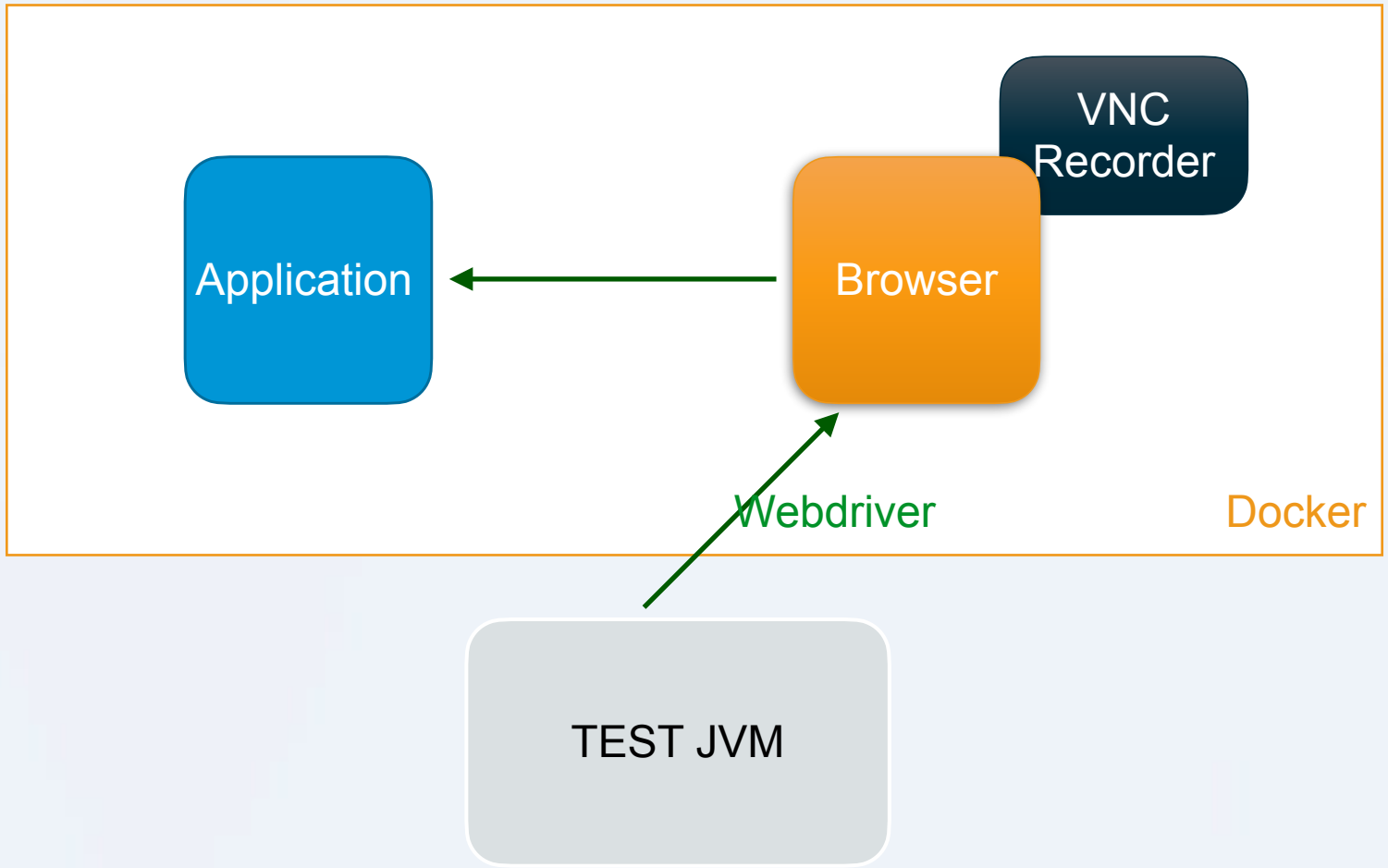


Integration testing

- Selenium
- For ex *testcontainers-selenium*
  - Stable browser version

- Integration within Jakarta EE Integration testing framework
  - Planned, prototype ready.

TestContainers selenium



Security

- Programmatic
  - Through rendered-attribute
  - Java statements
  - **Apache Shiro**
- Declarative
  - Custom tags within components
  - CDI interceptors
  - **Atbash Octopus**

Atbash Octopus

- Declarative
  - JSF Tags
  - CDI interceptors
  - JavaFX FXML tags
- Integration with
  - DB
  - OAuth2
  - JWT
  - SAML
  - OTP
  - …

Q & A

Thank you

- Atbash

- Blog
  - https://www.atbash.be
- Github
  - https://github.com/atbashEE