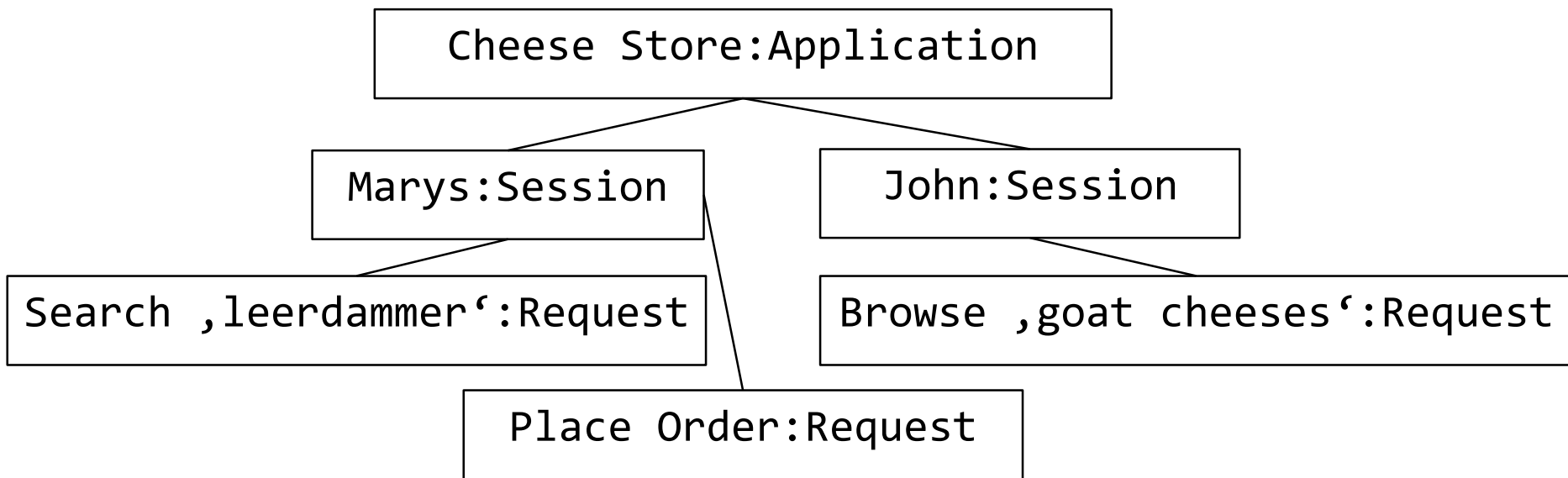


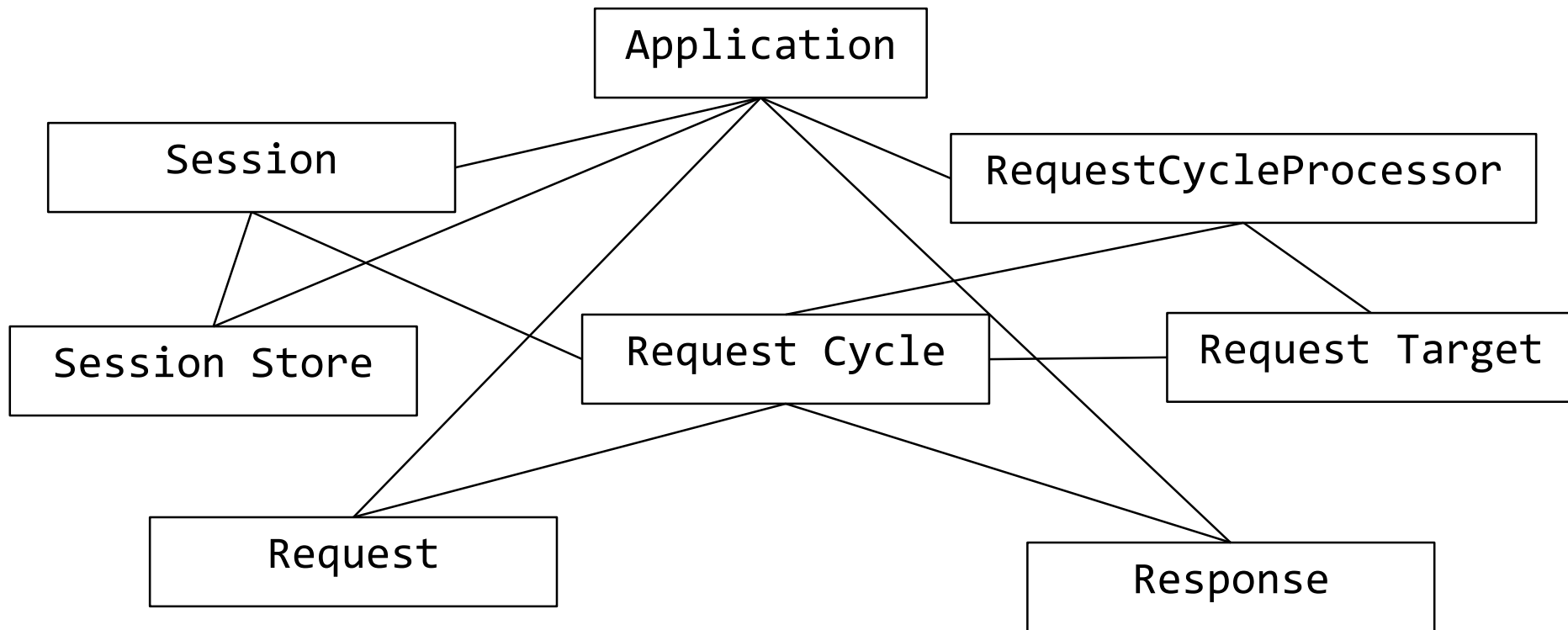
Nach M.Dashorst, E.Hillenius, *Wicket in Action*, Manning 2004

# DIE WICKET-ARCHITEKTUR – WIE WIRD EIN REQUEST VERARBEITET?



- Application: Oberster Container – Singleton
- Session: „State of user“
- Request: Kapselt den HTTP-Request des Users
- Response: „Schreiboperationen“ – Output des Requests

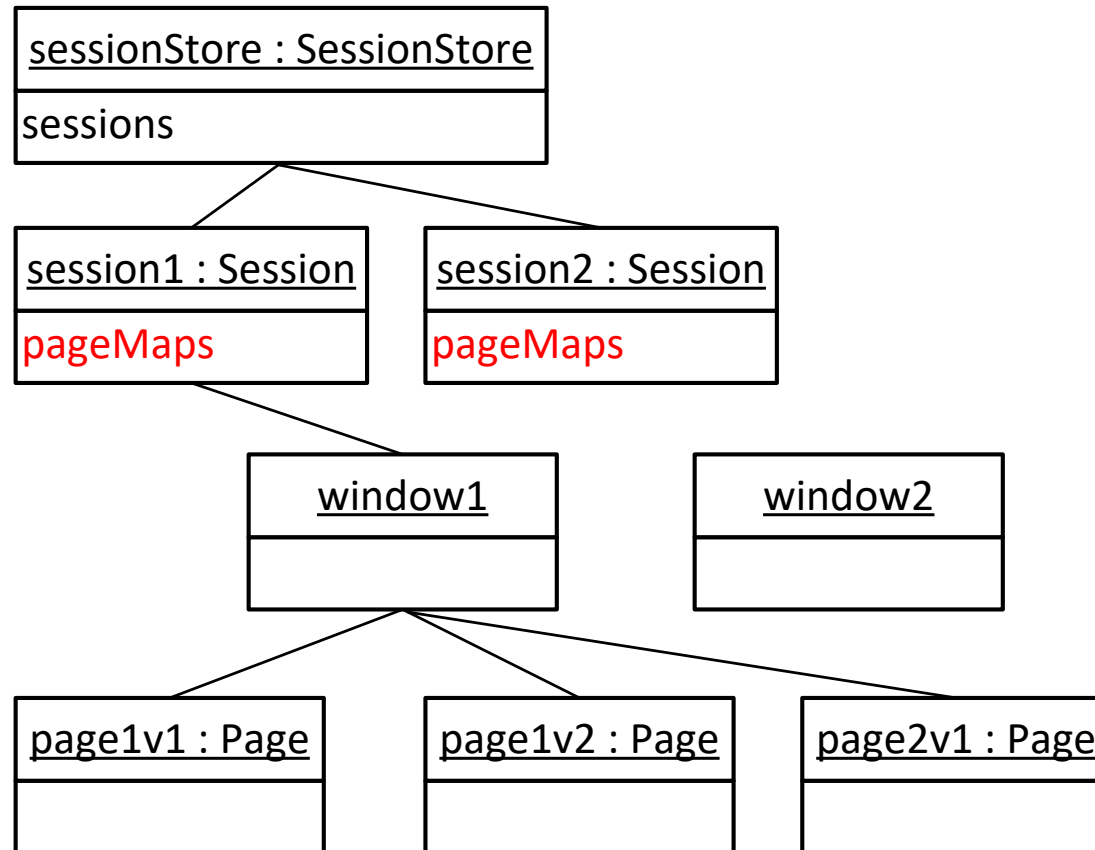
Nach M.Dashorst, E.Hillenius, *Wicket in Action*, Manning 2004



- RequestCycle: Wickelt den Request ab - Delegiert
- RequestCycleProcessor: Schritte / Events im Request
- RequestTarget: Aufrufbare Seite, AjaxTarget, etc.

Nach M.Dashorst, E.Hillenius, *Wicket in Action*, Manning 2004

# Request-Handling: Sessions



Nach M.Dashorst, E.Hillenius, *Wicket in Action*, Manning 2004

- Idee: Single-Threaded soweit möglich
- Components: synchronized auf Pages
- Achtung:
  - Instanz einer Komponente nur auf einer Page
  - **Nicht Thread-Safe:**
    - Application
    - Session
    - SessionStore



Nach M.Dashorst, E.Hillenius, *Wicket in Action*, Manning 2004

## Request Handling

- Kontrolliert durch Framework
- Abstraktion der Servlet API
  - ServletWebRequest
  - ServletWebResponse
- RequestCycle
  - Statischer Zugriff: RequestCycle.get()
  - Verarbeitung von Request & Response
  - Bestimmung konkreter URLs für Pages
  - Hook-Methoden und Listener
  - Ein Thread pro Cycle

# Request-Handling: Ablauf

GET /?wicket:interface=:2:actionLink::ILinkListener::

Decode Request

**parameters: RequestParameters**  
componentPath = "2:actionLink"  
versionNumber = 0  
interfaceName = ILinkListener

Determine Target

**target: ListenerInterfaceRequestTarget**  
component = Home\$1  
listener = linkListener  
page = Home

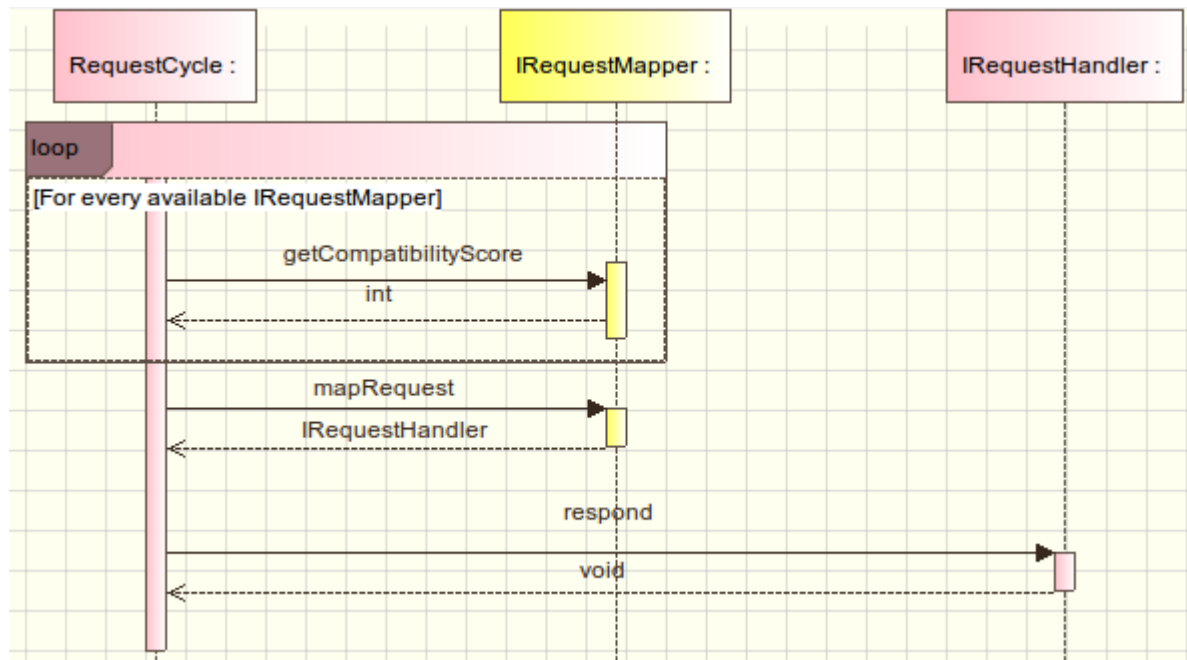
Process Events

onClick(..)

Respond

Quelle: M.Dashorst, E.Hillenius, *Wicket in Action*, Manning 2004

## Request Handling



Quelle: Wicket User Guide – The Reference Documentation



## Sessions

- Statischer Zugriff: Session.get()
- Aktueller Zustand des Komponentenbaums
- Eigene Implementierungen üblich
  - Beispiel: Durchführung des Logins
  - Registrierung in Application-Klasse
- Einklinken von Listenern
- Stateless Page: temporäre Session

```
public class BasicAuthenticationSession extends AuthenticatedWebSession {

    private String username;

    public BasicAuthenticationSession(Request request) {super(request);}

    @Override
    public boolean authenticate(String username, String password) {
        boolean success = false;

        // Dies ist nur ein Beispiel!
        if (username.equals("Wicket") && password.equals("GFU")){
            this.username = username;
            success = true;
        }

        return success;
    }

    @Override
    public Roles getRoles() {return null;}

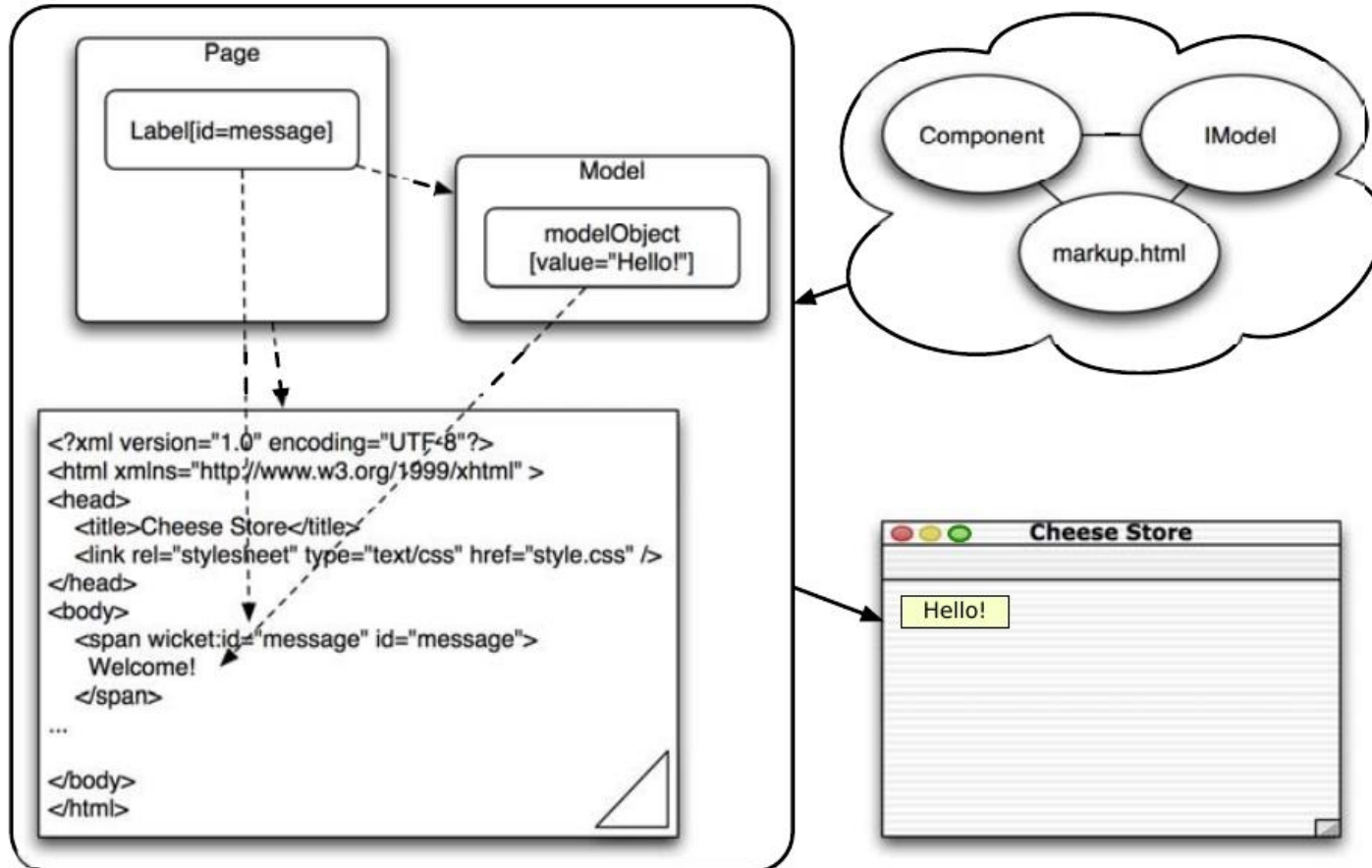
    public String getUsername() {return username;}
}
```

Quelle: Wicket User Guide – The Reference Documentation

## Komponenten

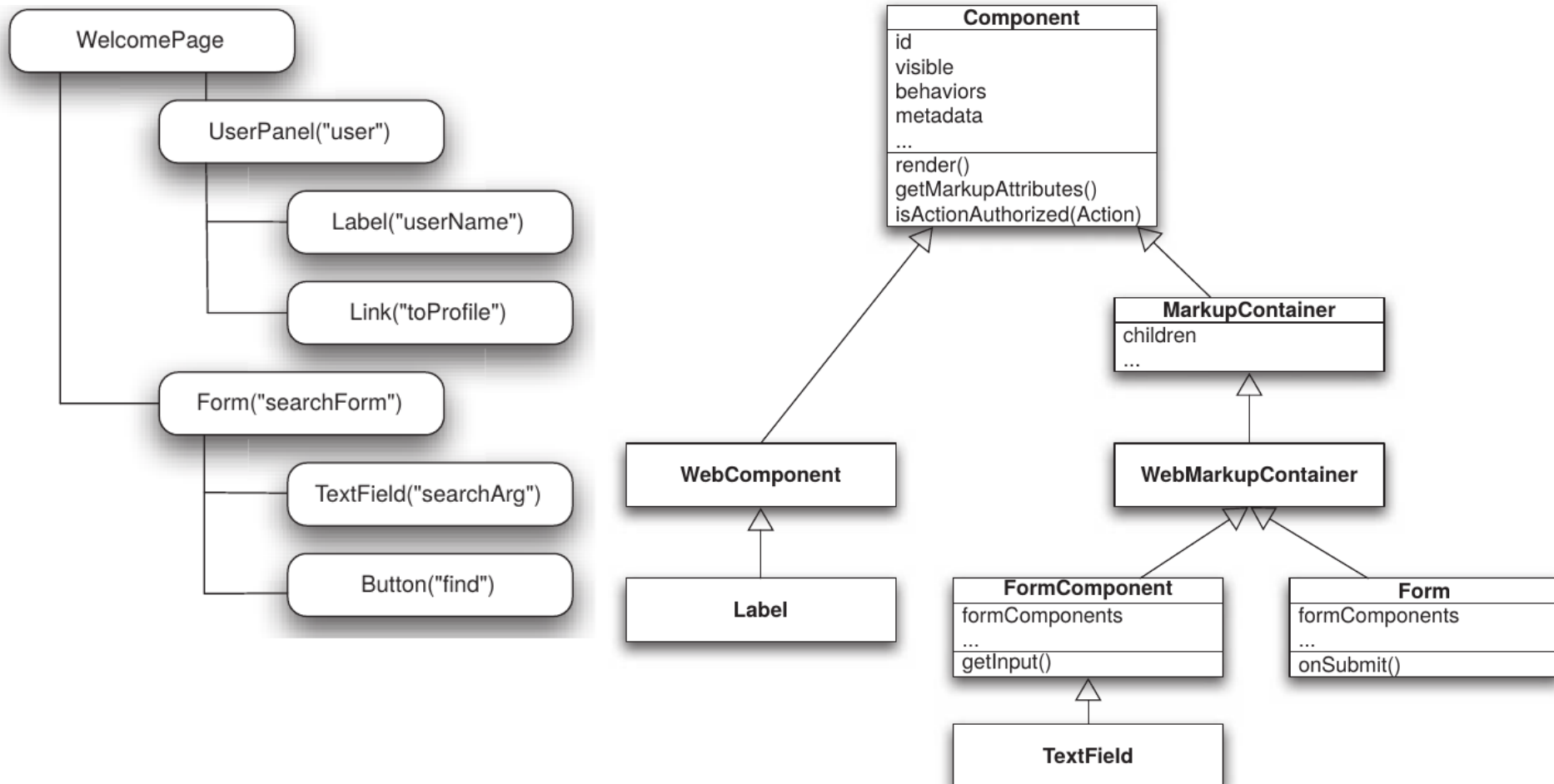
- Wiederverwendbare UI-Bausteine
- Self-contained
- HTML + Java (ggf. JavaScript...) – Strukturen sind ähnlich
- Stateful vs. stateless
- Vorgefertigte Komponenten
  - Labels
  - Forms
  - Links
  - Buttons

# Komponenten - MVC



Quelle: M.Dashorst, E.Hillenius, *Wicket in Action*, Manning 2004

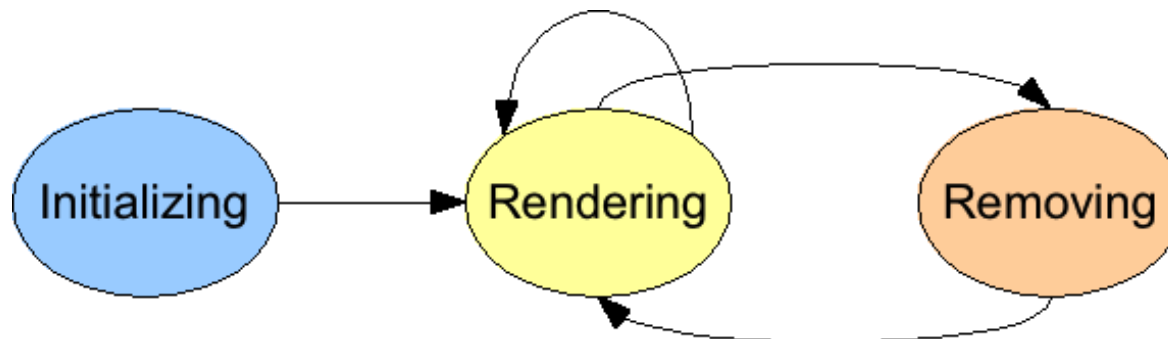
# Komponentenbaum & -struktur



Quelle: M.Dashorst, E.Hillenius, *Wicket in Action*, Manning 2004

## Komponenten

- Lebenszyklus
  - Initializing Instanziierung durch Wicket
  - Rendering Generierung von Markup
  - Removing Entfernung aus Komponentenbaum
- Hook-Methoden
  - onInitialize()
  - onConfigure(), onRender(), ...
  - onRemove()



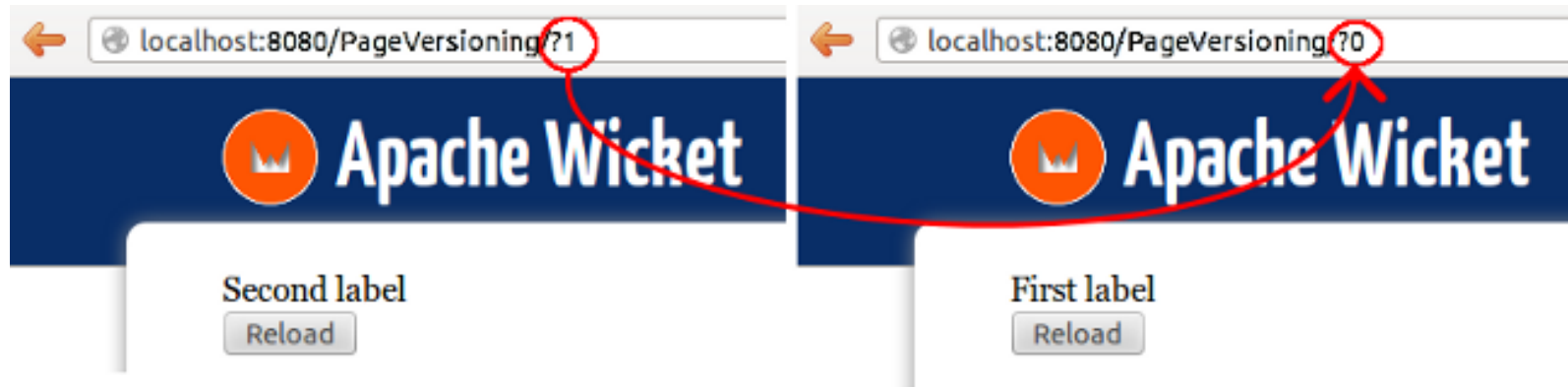
Quelle: Wicket User Guide – The Reference Documentation

## Pages

- Navigation über Link-Komponenten
  - Definition eines onClick-Handlers in Java
- Deeplinks: Mountpoints in der Application-Klasse definieren:
  - `mountPage("/pageMount/${foo}", MountedPageWithPlaceholder.class);`
- Wie viele Pages braucht eine Anwendung?
  - Wie viele abstrakte Pages?
  - Events

## Stateful Pages

- Verwendung über mehrere Requests hinweg
- Zustand in der Session (Java Serialization)
- Versionierung (→ Back-Button)



Quelle: Wicket User Guide – The Reference Documentation



## Stateless Pages

- Einsparung von Ressourcen
- Beispiel: Login Page
- Voraussetzungen
  - Instanziierung durch das Framework
  - Alle enthaltenen Komponenten ebenfalls stateless

- Modifizieren sie die Echo-Anwendung
- Erstellen Sie eine eigene Submit-Button Komponente
- Geben Sie im Server Antwort Bereich aus, in welcher Sequenz folgende Methoden durchlaufen werden:
  - `onInitialize()`
  - `onConfigure()`
  - `onRender()`
  - `onRemove()`