



Angular

Rich Internet Applications with JavaScript

Sebastian Lausch,
Student Informatik B.Sc.

23.07.2017, München

Aufbau & Inhalt

I. Rich Internet Applications

II. Nutzen von Angular

III. Einführung in Angular

1. Architektur Übersicht
2. Erläuterung der einzelnen Komponenten

IV. Stärken von Angular

V. Schwächen von Angular

VI. Fazit

Rich Internet Application

- **Internetanwendungen / Webapplikationen**
- **Intensive Benutzerinteraktion**
- **NICHT auf Browser beschränkt**
- **Erstellung mit Webframeworks (z.B. Angular) unter Verwendung von Webtechnologien**

HTML

CSS

JavaScript

Ajax

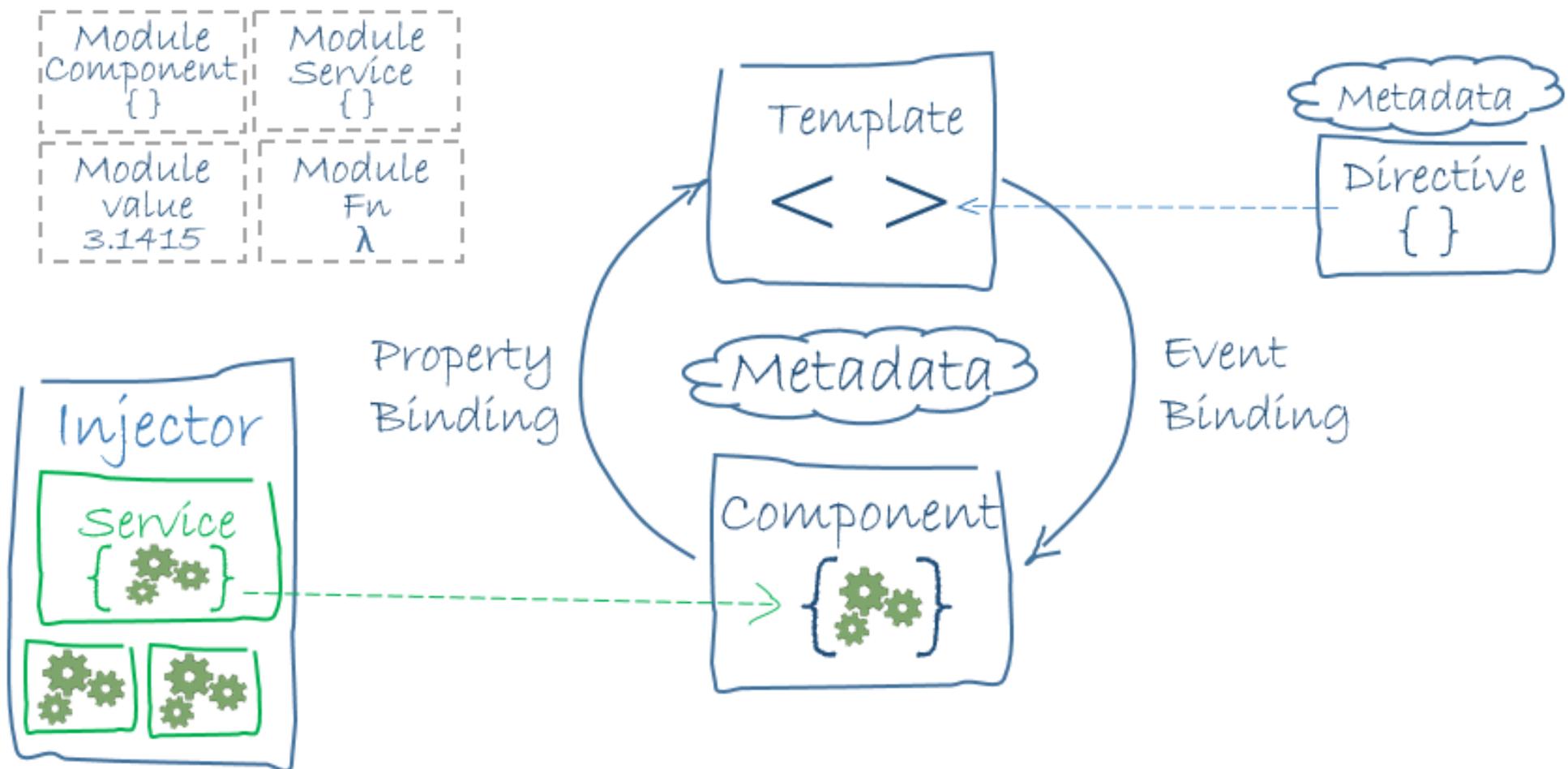
RIA vs klassische Webanwendung

- + Höhere Benutzerfreundlichkeit durch moderne Interaktionstechniken (z.B. Drag and Drop)**
- + Schnelle Reaktion auf Benutzereingaben**
- + Entlastung der Server**
- + Mit Frameworks einfach & schnell zu erstellen**
- eventuell lange Downloadzeiten**
- Ressourcenbelastung von Client**
- Sicherheitslücken**

Nutzen von Angular

- **Webframework mit TypeScript**
- **Einfachere Webapp-Entwicklung**
 - Bietet Routinen für häufige Probleme (Login, Datenaustausch)
 - Abstrahiert externe Datenbankanbindung und HTML, CSS Code
 - dynamische Anpassung an unterschiedliche Endgeräte
 - vieles mehr ...
- **Verbesserung von Wartbarkeit und Erweiterbarkeit der Webapplikation**

Einführung - Architektur Überblick



Einführung - Components

- **@Component-Dekorator**
- **Zentraler Bestandteil der Angularapplikation**
- **Definiert "View"**
- **Maßgeblich für Benutzerinteraktion verantwortlich**

Einführung - Templates

- **Für den Nutzer sichtbarer Teil**
- **Syntax ähnlich zu HTML**
- **Verwendung von Angular-Template-Syntax**
- **Ermöglicht Wiederverwendung von anderen Templates (DRY-Prinzip)**

Einführung – Directives

- ◆ **@Directive-Dekorator**
- ◆ **Manipulieren Aussehen der Templates im DOM**
- ◆ **Structural Directives**

Verändern Aussehen durch hinzufügen, verändern oder ausblenden von Elementen im DOM
- ◆ **Attribute Directives**

Verändern Aussehen oder Verhalten von existierenden Elementen (Bsp. Two-Way-Databinding)

Einführung – Data Binding I

- **Mechanismus, der erheblich Arbeit abnimmt**
- **Kommunikation zwischen Component und Template**
- **Property Binding / Interpolation**
 - Abfrage und Austausch von Daten/Werten zwischen Component und Template
- **Event Binding**
 - Aufruf von Methoden in Component bei Event in der Template (z.B. Klicken eines Buttons)

Einführung – Data Binding II

- ◆ **Two-Way-Databinding**

Kombination aus Property und Event Binding

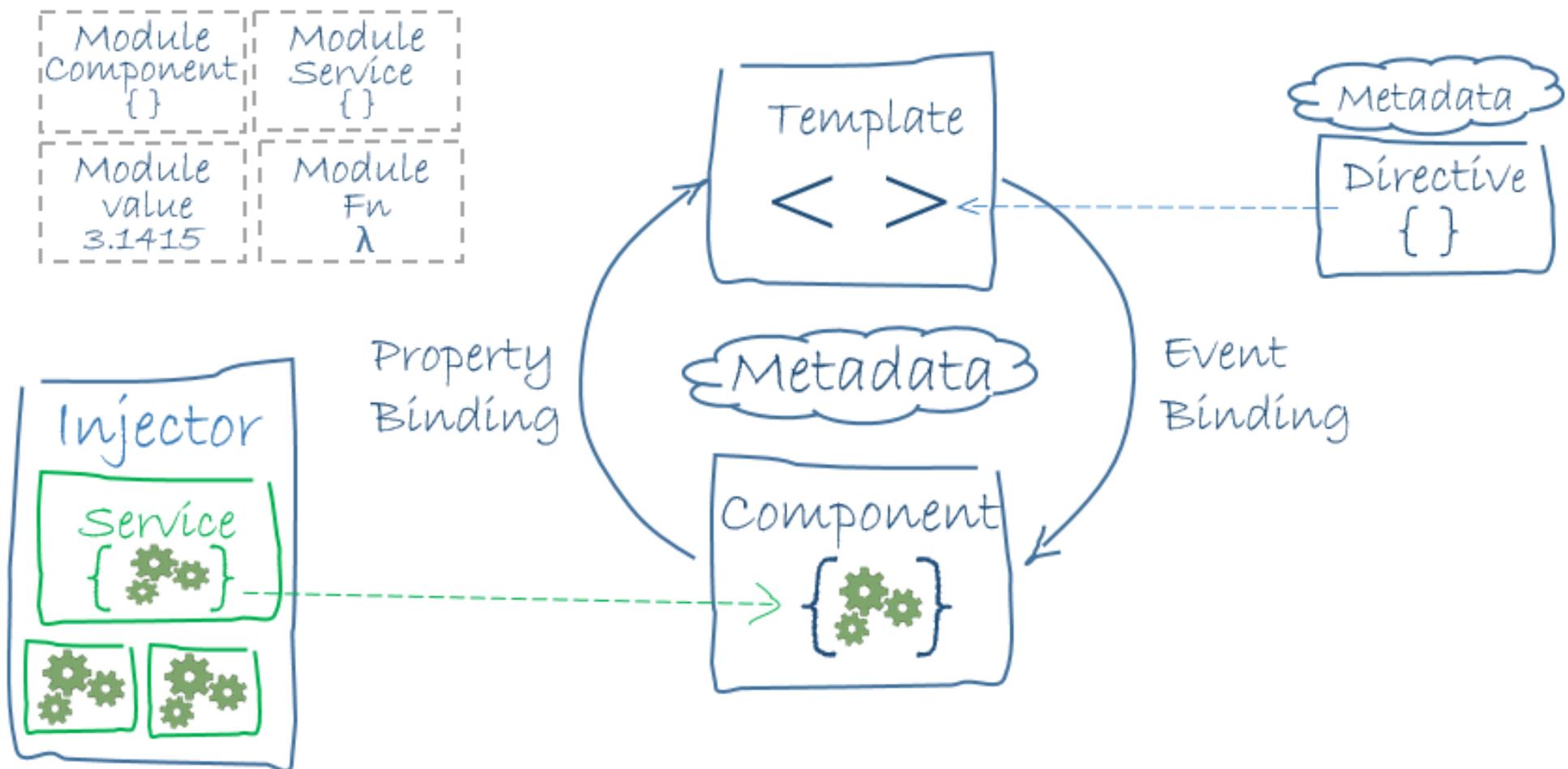
Initialer Austausch von Werten

Bei Änderung der Werte (z.B. Usereingabe) →

Übertragung der geänderten Werte an Component

- ◆ **Erhebliche Erleichterung gegenüber jQuery o.Ä.**

Einführung - Architektur Überblick



Einführung - Services

- **Unklare Definition von Services**
- **Meist für eine spezielle Aufgabe**
- **Abbildung von Logik (nach MVC → Controller)**
- **Werden von Components benutzt**
- **Designprinzip der Trennung von Logik & View nicht erzwungen**

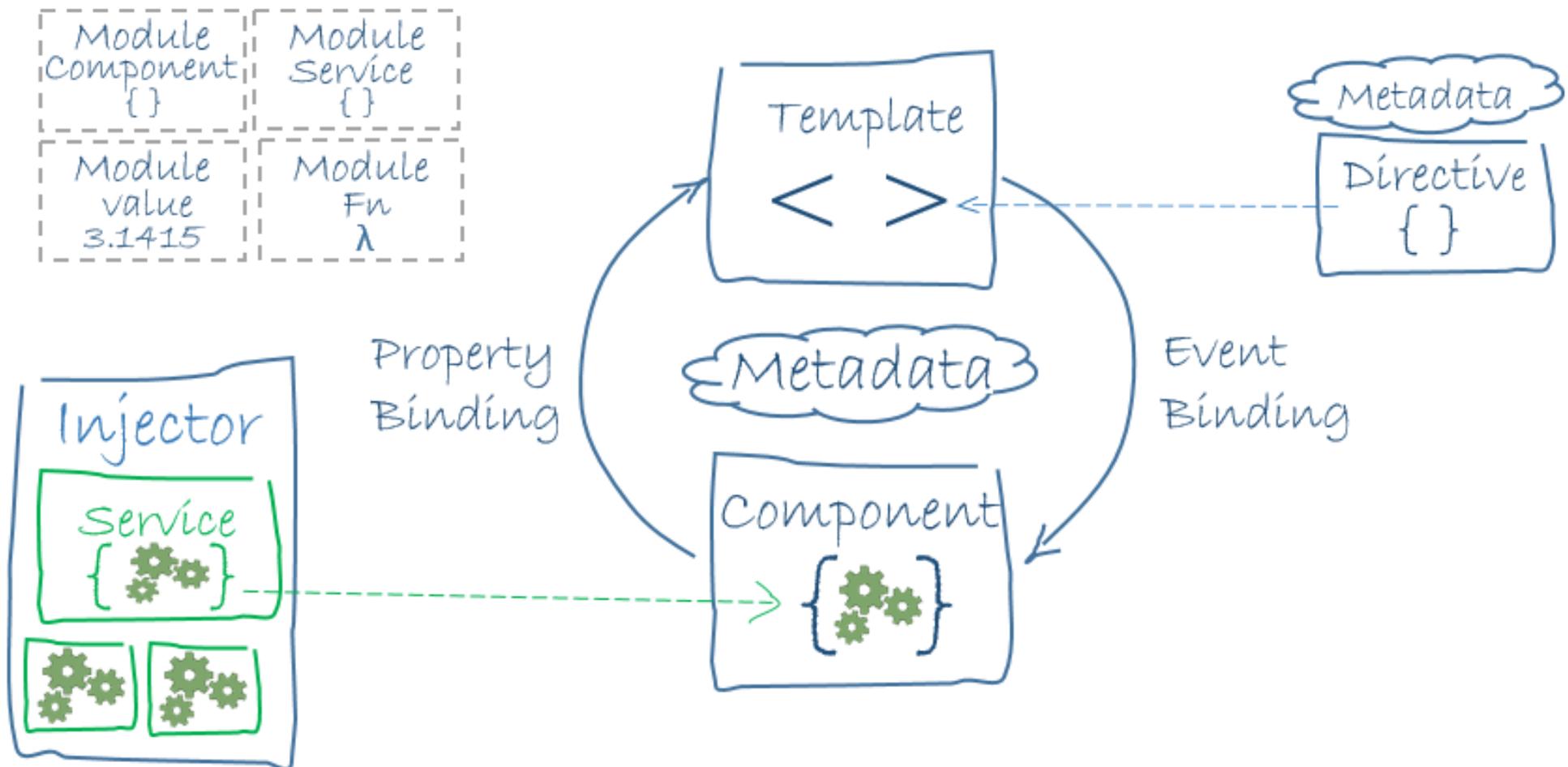
Einführung - Dependency Injection

- **Grundlegender Mechanismus**
- **Auflösung von Abhängigkeiten**
- **“Injector” verwaltet Container von allen “@Injectables” (z.B. Services)**
- **Verbindet Services und Components**

Einführung – Angular Modules

- **Grundgerüst der Angularapplikation**
- **@NgModule-Dekorator**
- **Trennen Fachdomänen einer Anwendung auf**
- **Rootmodule**
 - Legt Aufbau der Anwendung fest
 - Kennt alle Components, Services und benötigten Module
 - Wird als Startpunkt der Applikation verwendet

Einführung - WrapUp



Stärken von Angular

- **Plattformunabhängigkeit**
- **Breite Unterstützung durch Browser**
- **Data-Binding**
- **Aufbau nach MVC**
- **Weiterentwicklung**

Schwächen von Angular

- **Unterstützung älterer Browser**
- **Hohe Einarbeitungszeit**
- **Inkompatibilität zu AngularJS**

Fazit

- **Besonders geeignet für hohe Benutzerinteraktion**
- **Hohe Verbreitung und große Community**
- **Stabiles und schnelles Framework**
- **Schneller Einstieg durch Dokumentation mit gezeigter Beispielapplikation**

Quellenangabe

Angular Homepage

<https://angular.io>

Angular Dokumentation (mit Tutorial)

<https://angular.io/docs>

RIA Erklärung

https://de.wikipedia.org/wiki/Rich_Internet_Application



**Herzlichen Dank
für ihre Aufmerksamkeit**

Angular - Rich Internet Applications with JavaScript

Sebastian Lausch,
Student Informatik B.Sc.

23.07.2017, München