

Motivation

- Effiziente Entwicklung von SPAs
- Bewährte Architekturmuster soweit wie möglich anwenden
- Einheitliche Codestrukturen fördern

Lernziele

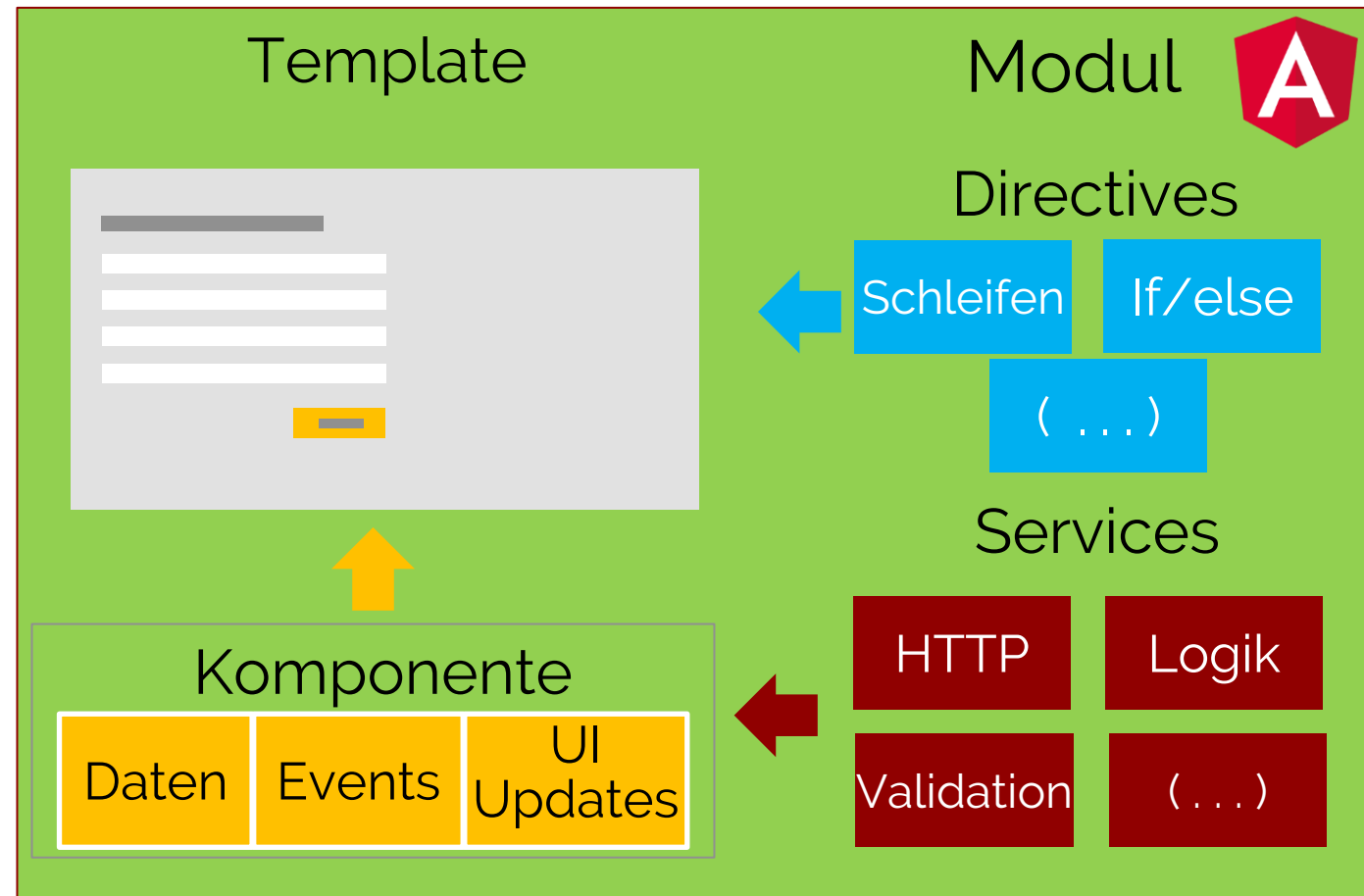
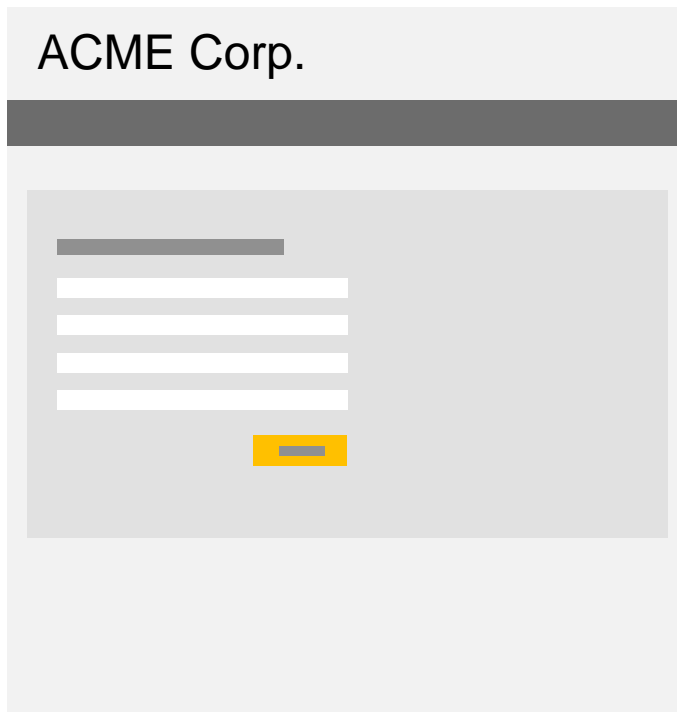
- Sie beschreiben die Konzepte des Frameworks Angular.
- Sie benennen die wesentlichen Bestandteile der Angular-Architektur und erklären grob deren Bedeutung.
- Sie beschreiben die Unterschiede zwischen Angular und anderen Enterprise Frameworks.
- Sie erklären den Unterschied zwischen Bibliotheken und Frameworks.

Steckbrief SPA Framework – Angular

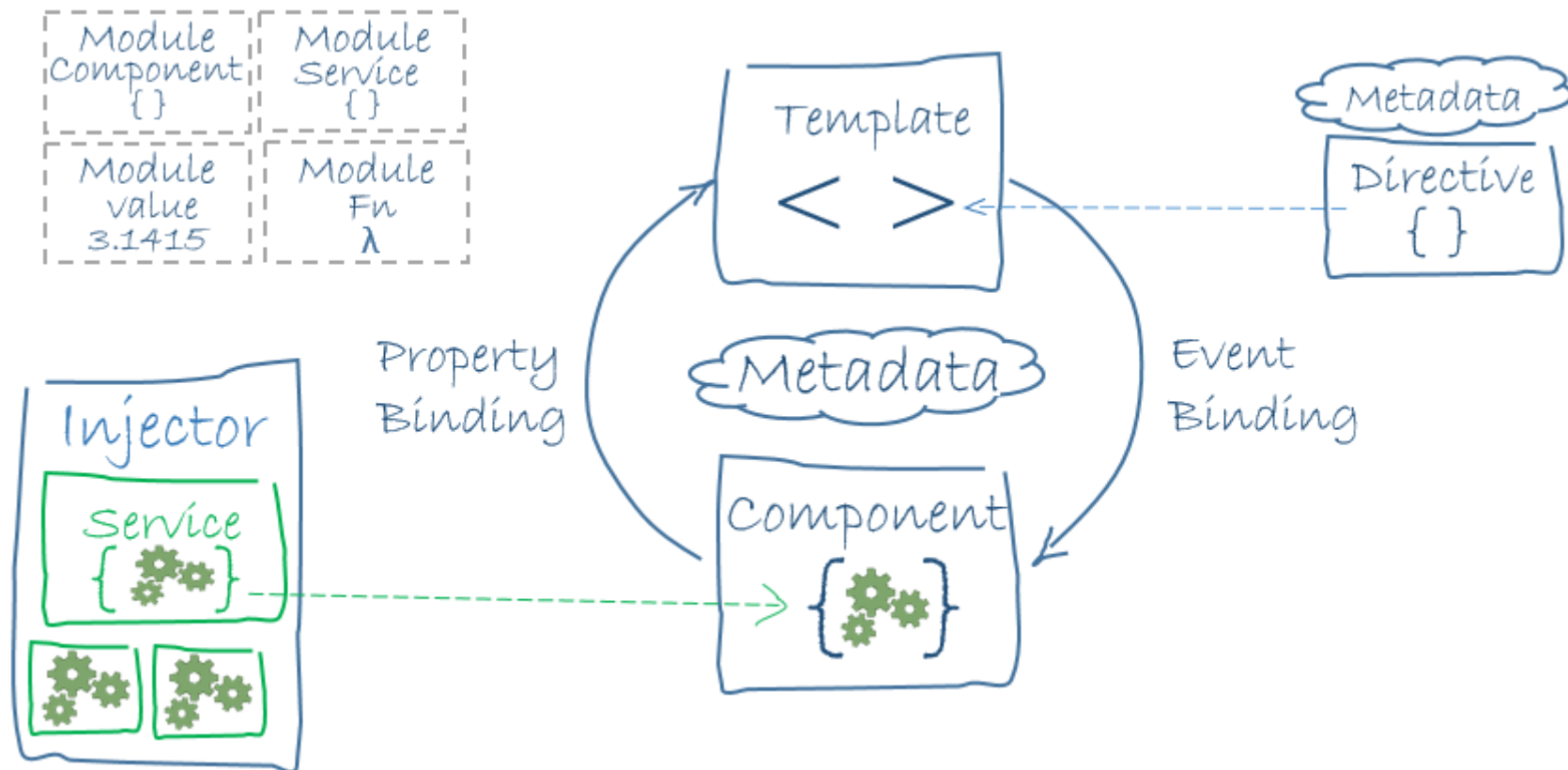
- Client Framework zur Implementierung von Webanwendungen im Browser
 - Basis: HTML und JavaScript
 - **Bevorzugter Weg:** TypeScript → Transpilierung nach JavaScript
- Hersteller und Maintainer: Google
- Lizenz: MIT
- Neuentwicklung des verbreiteten Frameworks AngularJS (Für AngularJS ab Version 2 Namenswechsel auf Angular)
 - AngularJS = Version 1.x
 - Angular >= Version 2.0.0

„The Angular Way“ für Entwicklung von SPAs

HTML



Angular Big Picture – Einführung Architektur



Angular Eigenschaften

- Gibt eine Struktur vor
- Baut auf bewährten Entwurfs- und Architekturmustern auf:
 - Strikte Trennung zwischen den Schichten Darstellung, Business und Daten
 - Komponentisierung
 - Modularisierung
 - Dependency Injection
- Fördert Wiederverwendbarkeit via Services (DRY)

Unterschiede zu Libraries (jQuery)

- Lästiger Boilerplate Code wird abstrahiert
- Entwickler können sich auf Implementierung von Business-Logik konzentrieren → Nicht auf Details die HTML zu manipulieren
- Einheitlicher(er) Codestil
- Vermittelt das Gefühl, dynamisches HTML zu schreiben anstatt JavaScript, welches HTML modifiziert

Templates

```
<!doctype html>
<html lang="en">
<head>
  <meta charset="utf-8">
  <title>ToDoApp</title>
  <base href="/">

  <meta name="viewport" content="width=device-width, initial-scale=1">
  <link rel="icon" type="image/x-icon" href="favicon.ico">
</head>
<body>
  <registration-form></registration-form>
</body>
</html>
```


Komponenten

```
@Component({  
  selector: 'hello-world',  
  templateUrl: 'hello-world.component.html',  
  styleUrls: ['hello-world.component.css']  
})  
  
export class HelloWorldComponent {  
  let greetings = 'Hello Angular Component';  
  
  public getComponentName(): string {  
    return 'HelloWorldComponent';  
  }  
}
```

Services

```
Import { Injectable } from `@angular/core`  
  
@Injectable()  
export class ExampleService {  
  
    constructor()  
  
    getExampleData() {  
        ...  
    }  
}
```

Verfügbare Dekoratoren

- @Component:** Komponente. Beinhaltet Präsentationslogik, sowie Schnittstelle zwischen Service und Template.
- @Injectable:** Service. Beinhaltet Geschäftslogik und Datenzugriffsschnittstelle.
- @Directive:** Direktive. Drei Arten: Komponenten, Strukturelle sowie Attribut Direktiven.
- @NgModule:** Das Angular Framework basiert auf Modulen. Apps werden modular entwickelt.
- @Pipe:** Pipe. Dienen der Formatierung von Template-Ausgaben.

Zusammenfassung

Um was ging es in diesem Modul?

- Philosophie von Angular
- Vorgehen bei der SPA-Entwicklung mit Angular (The Angular Way)
- Überblick über Bestandteile der Architektur von Angular
- Entwurfsmuster von Angular

Wozu brauche ich das? Was will ich damit machen?

- Projekte auf fundierten Erkenntnissen / etablierter Basis aufbauen
- Komplexe Webanwendungen effizient entwickeln
- Bewährte Lösungsansätze für typische Probleme der Webentwicklung nutzen

Kontrollfragen

- Was ist der „Angular Way“?
- Was sind Gründe dafür, Angular statt jQuery einzusetzen?
- Welche verbreiteten Entwurfsmuster unterstützt Angular?

