

Konzeption und prototypische Implementierung einer mobilen Anwendung zur Zeiterfassung

Bachelorarbeit Fabian Trampusch

Bestandteile der Bachelorarbeit

- Grundlagen
 - Frameworks
 - App-Arten
- Anforderungsanalyse
 - SOPHIST
 - Use-Case-Diagramm
 - Umfrage
- Technische Konzeption
 - Oberflächenkonzept
 - Grobarchitektur
- Backend
 - JAX-RS
- Oberfläche
 - jQueryMobile
 - AngularJS
- Tooling
 - Yeoman, Grunt
- Test
 - Karma, Chai, Mocha
 - Istanbul

Bestandteile der Bachelorarbeit

- Grundlagen
 - Frameworks
 - App-Arten
- Anforderungsanalyse
 - SOPHIST
 - Use-Case-Diagramm
 - Umfrage
- Technische Konzeption
 - Oberflächenkonzept
 - Grobarchitektur
- Backend
 - JAX-RS
- Oberfläche
 - jQueryMobile
 - AngularJS
- Tooling
 - Yeoman, Grunt
- Test
 - Karma, Chai, Mocha
 - Istanbul

HTML5 Offline Storage

- Ablegen von Daten und Ressourcen in speziellem Cache
- Cache wird bei nichtvorhandener Verbindung wie statischer Webserver verwendet

[\[demo\]](#)

Web-Entwicklung

Serverseitig / Java

- 1997:
 - Servlets
- 1999:
 - JavaServer Pages
- 2005:
 - Facelets

Web-Entwicklung

Serverseitig / Java

- 1997:
 - Servlets
- 1999:
 - JavaServer Pages
- 2005:
 - Facelets

Clientseitig / JavaScript

- bis 2005:
 - plain old JavaScript
- 2005/2006:
 - Ajax, jQuery
- \geq 2009:
 - Angular, Backbone, Ember

AngularJS

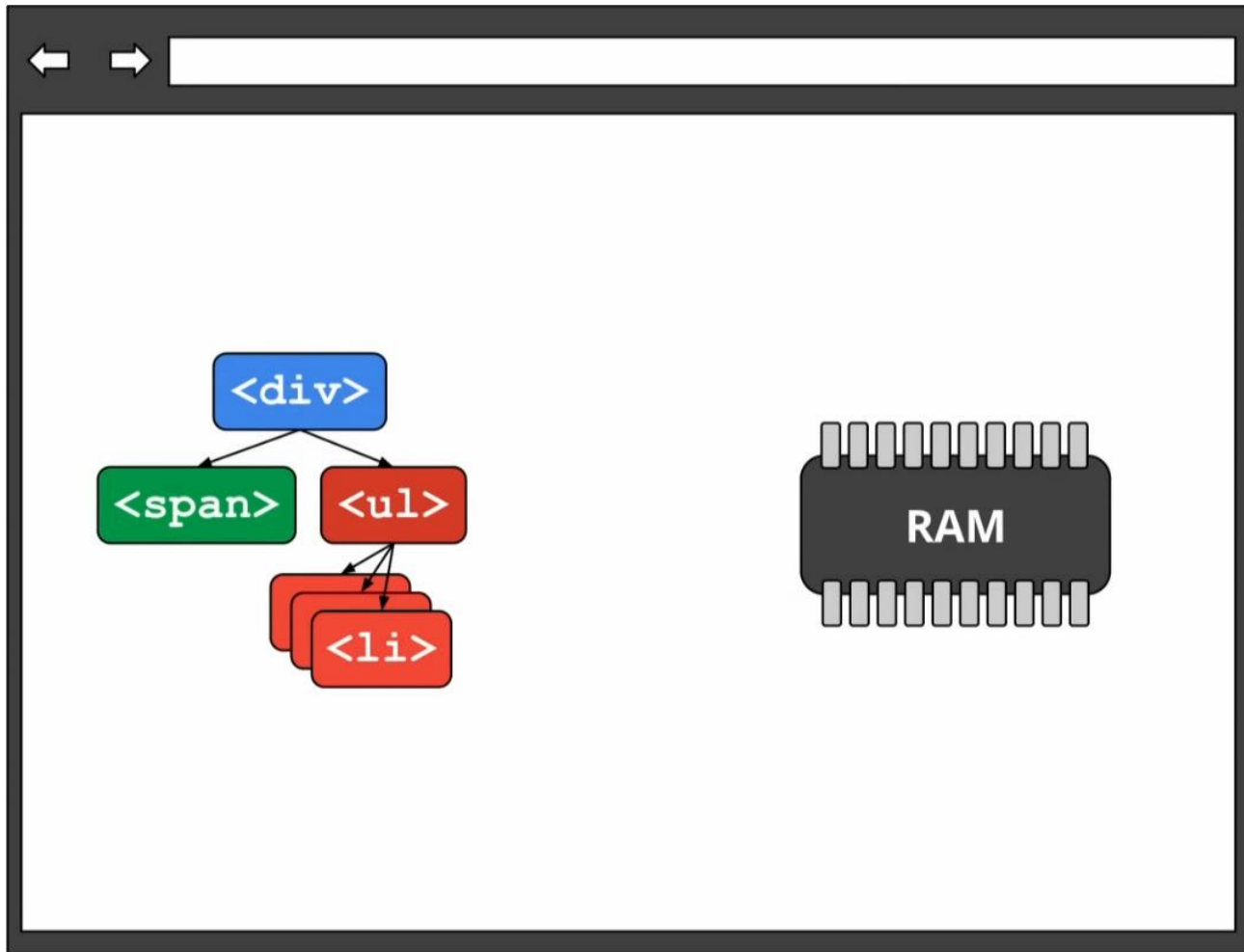
„Other frameworks deal with HTML's shortcomings by either abstracting away HTML, CSS, and/or JavaScript or by providing an imperative way for manipulating the DOM. Neither of these address the root problem that HTML was not designed for dynamic views.“ -<http://angularjs.org>

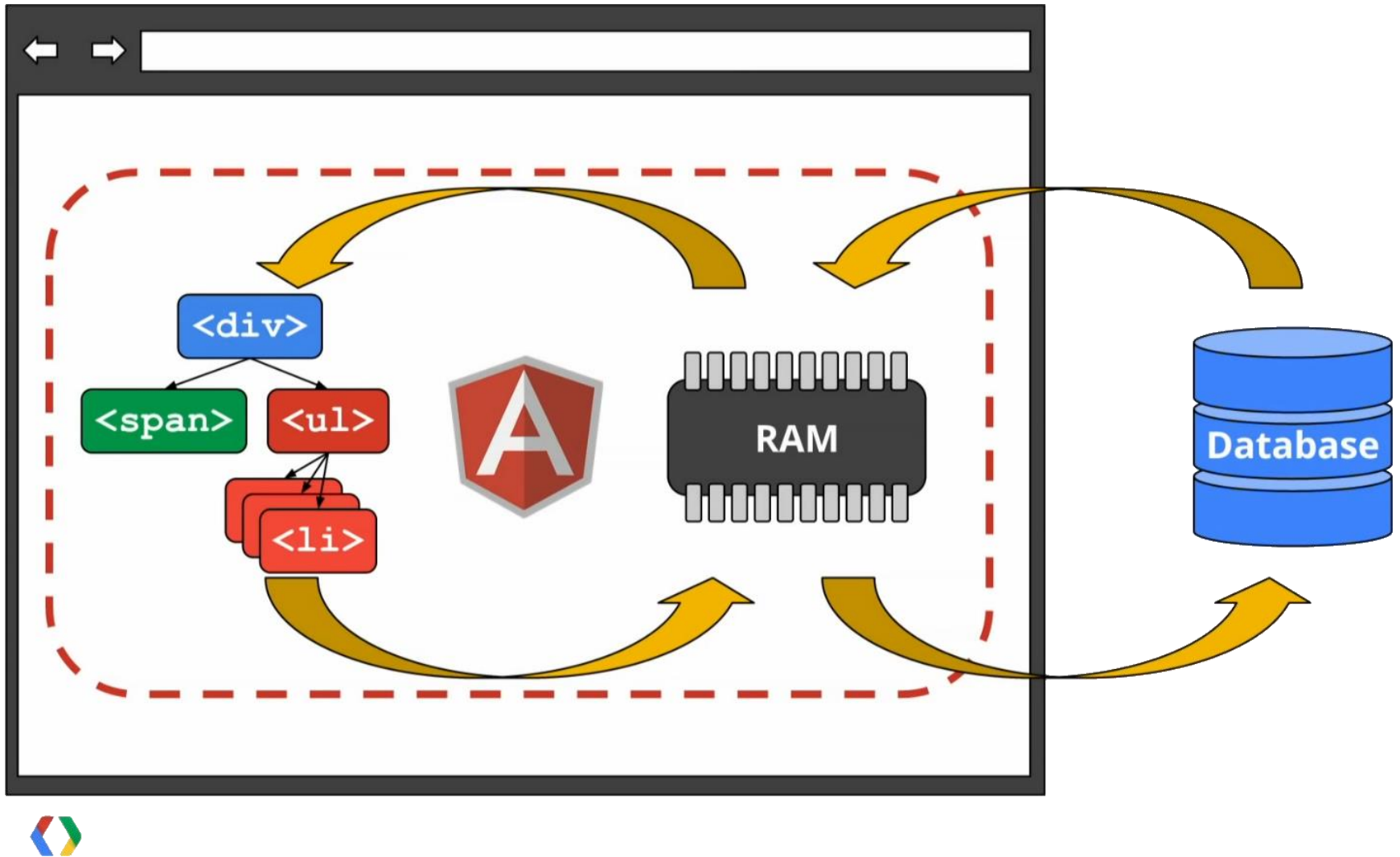
Entwicklungsziele von AngularJS

- D.R.Y.
- Boilerplate vermeiden
- Struktur und testbare Resultate schaffen

„Three D's“:

- Direktiven
- Data-Binding
- Dependency Injection





Beispiel 1

- Data-Binding und Direktiven
- Keine Geschäfts- oder Oberflächenlogik

[\[demo\]](#)

Beispiel 1 - Quellcode

```
<html>
  <head>...</head>
  <body ng-app>
    <input ng-model="name">
    <p ng-show="name">
      Hallo {{name}}!
    </p>
  </body>
</html>
```

Beispiel 2

- Controller (Oberflächenlogik) und Scope (Datenbereich, Model)

[demo]

Beispiel 2 - Quellcode

```
<input ng-model="user.name">
<button ng-click="add()" ng-disabled="!user.name">Eintragen</button>
<ul>
  <li ng-show="users.length == 0">Keine Einträge vorhanden</li>
  <li ng-repeat="user in users">{{user.name}}</li>
</ul>
```

```
function ItemController($scope) {
  $scope.users = [];
  $scope.add = function() {
    var newUser = {name: $scope.user.name};
    $scope.users.push(newUser);
    $scope.user.name = "";
  };
}
```

Beispiel 3

- Geschäftslogik in Form eines Services

[demo]

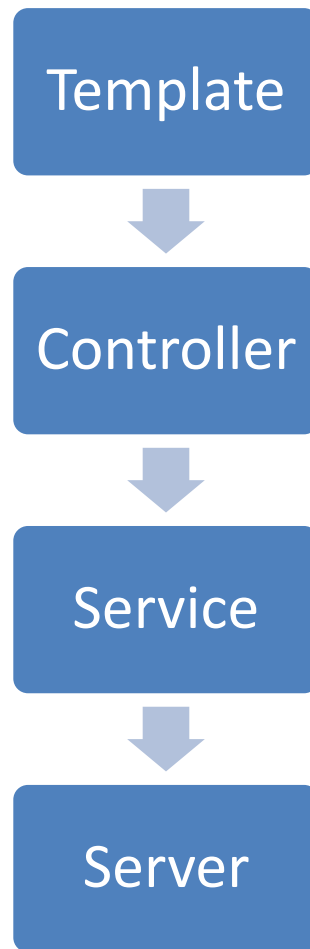
Beispiel 3 - Quellcode

```
function ItemController($scope, greeter) {
    $scope.greet = function() {
        var name = $scope.user.name;
        greeter.greet(name);
    };
}

function GreeterService() {
    this.greet = function(text) {
        alert("Hi " + text);
    };
};

var app = angular.module("app", []);
app.controller("ItemController", ItemController);
app.service("greeter", GreeterService);
```


Zusammenfassung der Architektur



Toolunterstützung & Tests

- Projektsetup mit Yeoman
- Build mit Grunt
 - Konfiguration in „Gruntfile.js“
 - Automatisierung von Aufgaben
- Testing mit Karma, Mocha und Chai
 - Ähnlich zu JUnit
 - Mocking macht Testen von AngularJS einfach
 - Test-Coverage mit Istanbul einfach messbar

Ausblick und Fazit

- Integration mit jQueryMobile manuell nötig
- Interessante Technologie für mobile Geräte/Web-Apps
- Roadmap von AngularJS 2.0 zielt voll auf Mobilgeräte
- Ausgereifte Toolunterstützung

Danke!
Fragen?

Quellen und weiterführendes Material

- <https://github.com/trampi/bachelor-presentation>
- <http://angularjs.org/>
- <http://www.youtube.com/watch?v=HCR7i5F5L8c>
- f.trampusch@isys-software.de