AngularJS & materializecss

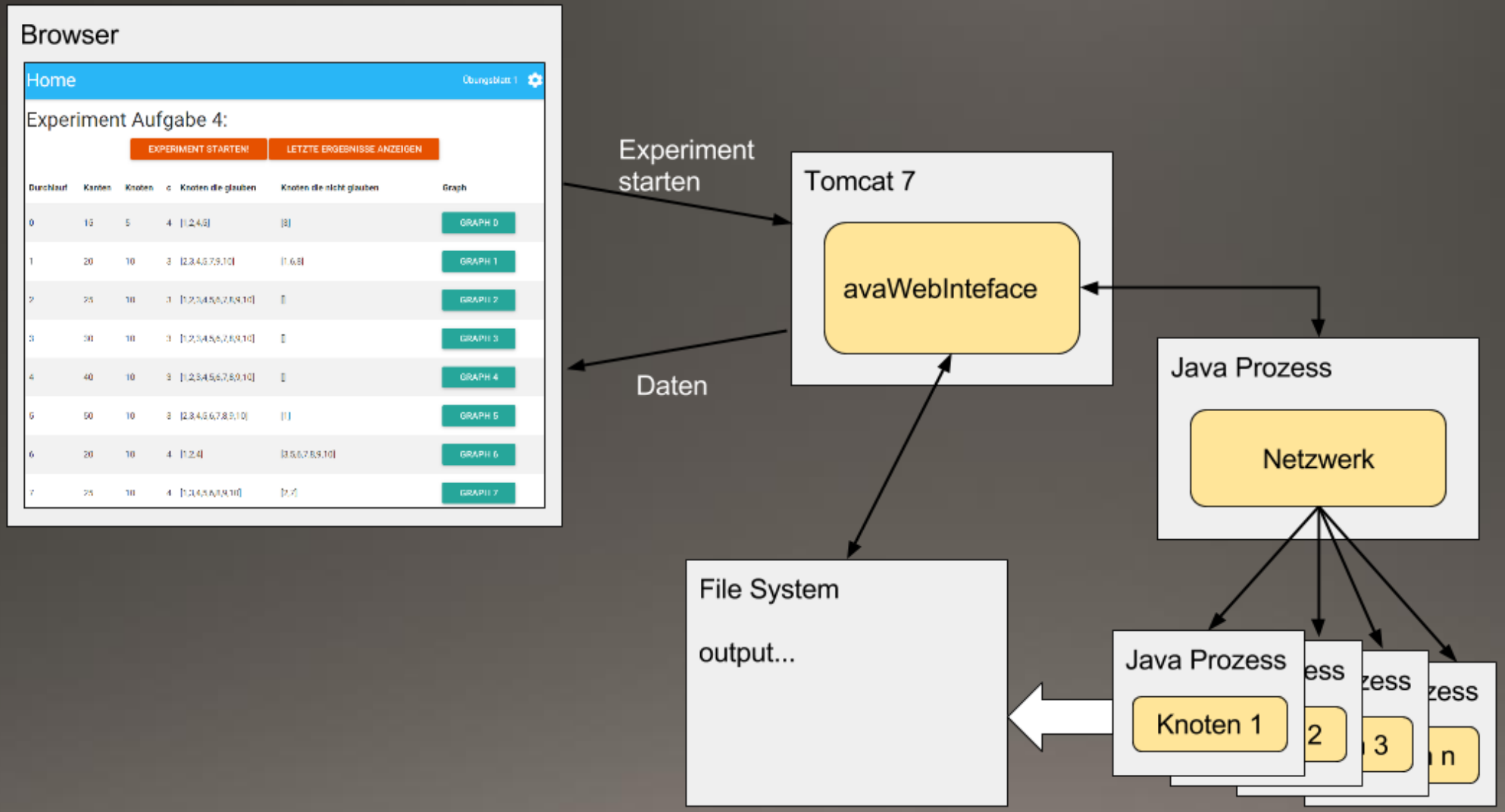
Inhalt

1. Lösung Übung 1
2. AngularJS
3. materializecss
4. vis.js
5. Demo

Lösung Übung 1

- Knoten(JAR) und Netzwerk(JAR) in Java (8)
- Netzwerk führt Experiment und Knoten aus
- Nachrichtenformat: JSON
- Web UI auf Apache Tomcat 7
- Daten bereitgestellt über Servlets
- UI als Webinterface mit AngularJS und materializecss
- Darstellung des Netzwerkes: vis.js

Lösung Übung 1

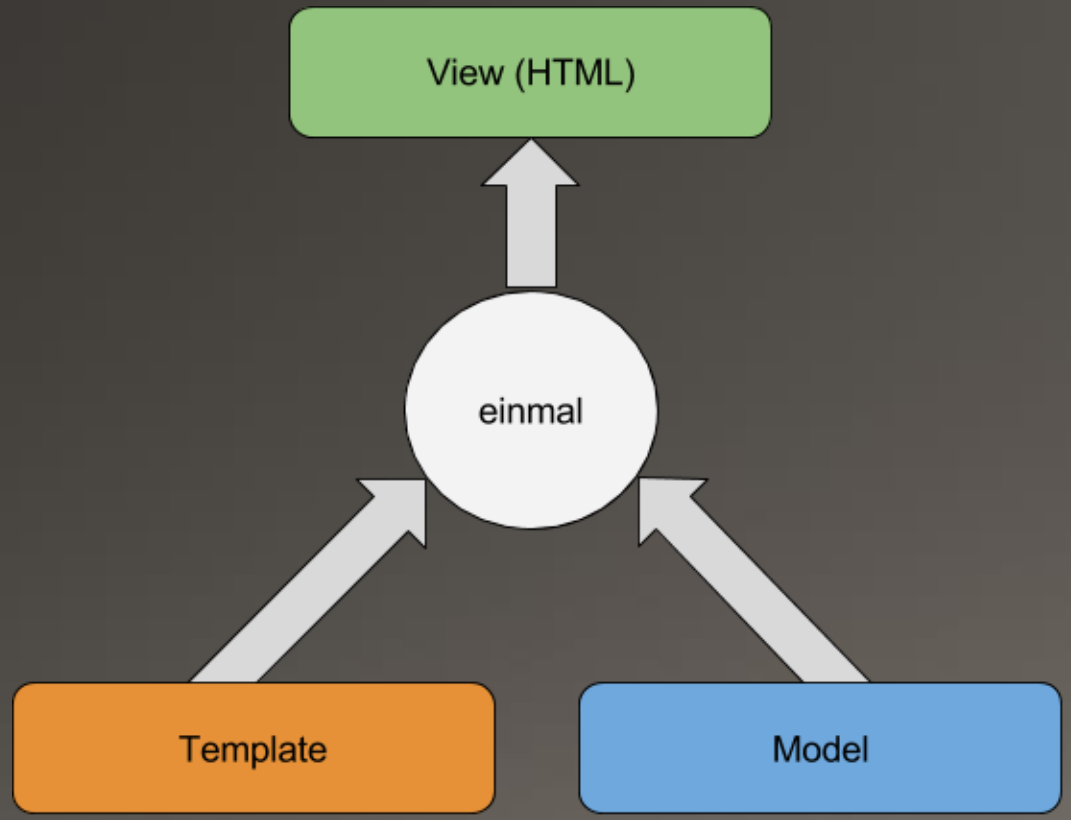


Was ist AngularJS?

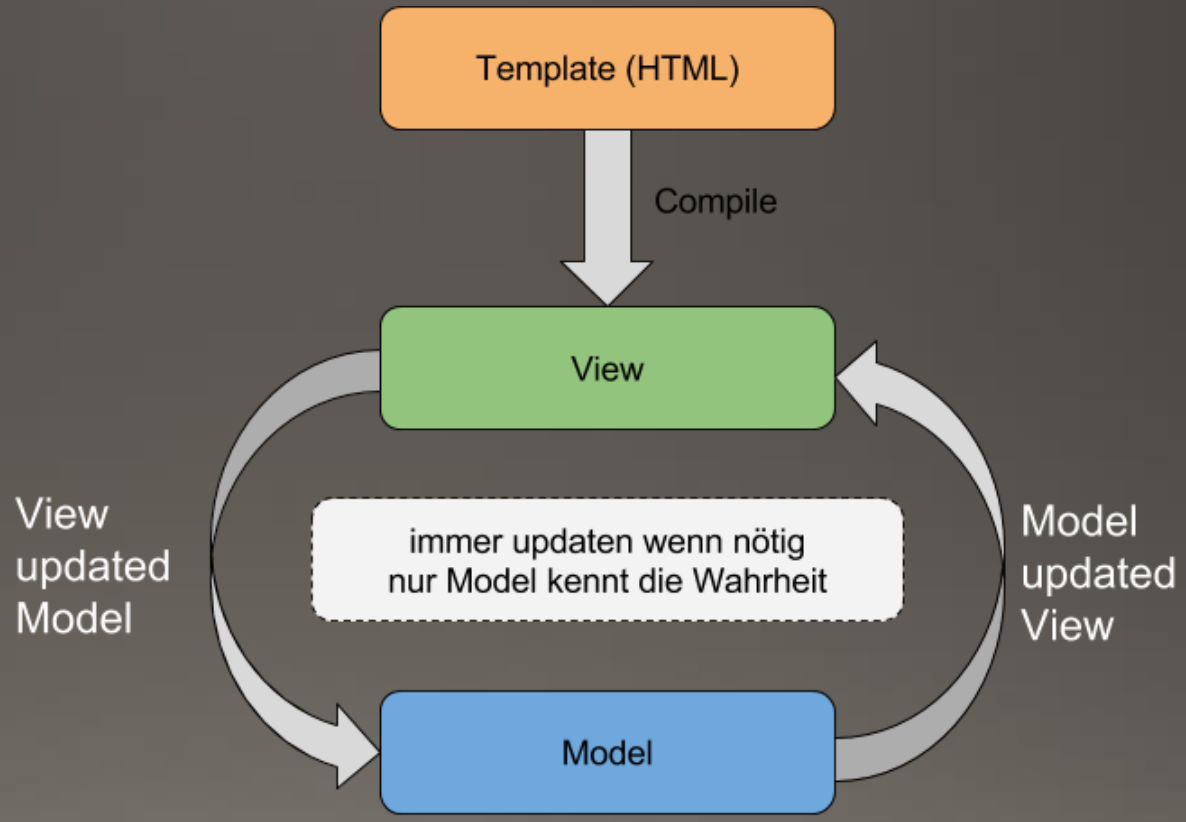
- clientseitiges Javascript Framework
- entwickelt von Google (seit 2009, aktuell v1.4.7)
- open-Source (MIT Lizenz)
- alle gängigen Browser unterstützt (IE 8 nur bis v1.2)
- Model-View-Controller Konzept (MVC)
- **Bidirektionales Databinding (Two-Way)**
- erweitert HTML um Attribute und dynamische Inhalte

Two-Way-Databinding

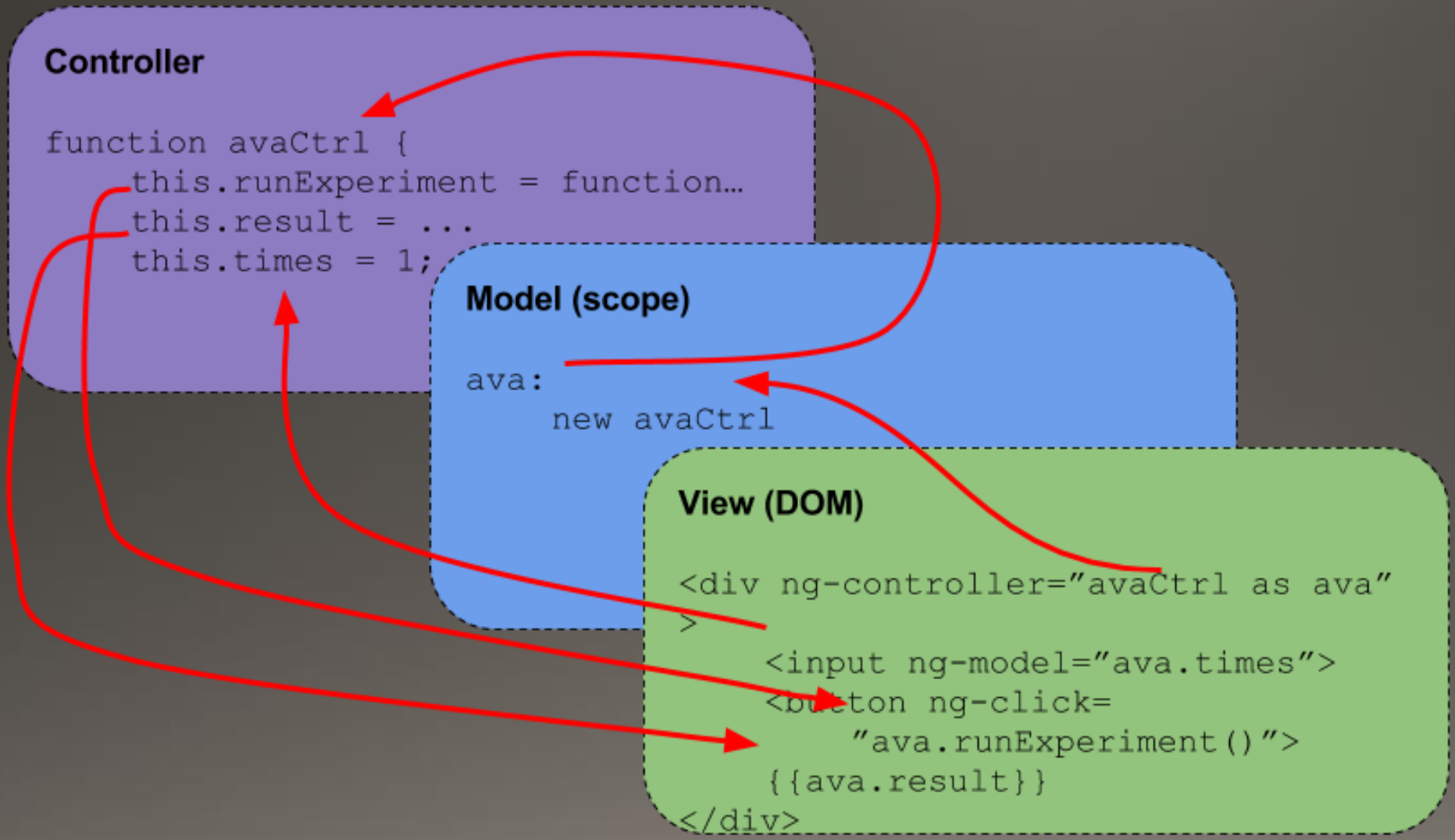
one-way-databinding



two-way-databinding



MVC mit AngularJS



Direktiven

- beginnen mit `ng-`
- Attribute eines HTML-Elements
- eigene Definition möglich
- Beispiele:
  - `ng-show="<Bedingung>„`
    zeigt das Element, wenn Bedingung true
  - `ng-model="variable„`
    bindet Variable an Input-Feld
  - `ng-repeat="car in catalog„`
    wiederholt für alle Elemente in `catalog`

Expressions

- binden Daten in die Webseite ein
- Ausdruck in `{{}}`
- Beispiele:
  - `{{'Hello World'}}`
  - `{{17 + 23}}`
  - `{{$scope.result}}`

Controller

- `<body ng-controller="avaMainCtrl"/>`
  => Untergliederung der Anwendung in Controller (eigener Scope)
- `avaWebApp.controller('avaMeinCtrl'...`

- enthalten für die View erforderliche Business Logik
- weitere Logik in Services auslagern

Bootstrapping AngularJS

- `<script src="..."/>` => lädt Angular nach
- `<html ng-app="avaWebApp">` => definiert die Anwendung
- beliebiges HTML Element wählbar -> Was gehört zur Anwendung?
- pro Anwendung eigener Scope (Speicherbereich)
- alle Direktiven und Expressions in Anwendung:
  - Zu Beginn „kompiliert“
  - Danach überwacht

Beispiel einer AngularJS Anwendung

```
1  <html lang="de" ng-app="avaWebApp">
2  <head>
3    <title>AVA</title>
4    <!-- CSS  ...-->
5    <script src="js/angular.min.js"></script>
6    <script src="js/controllers.js"></script>
7  </head>
8  <body ng-controller="avaMainCtrl">
9          <table class="striped"ng-hide="running">
10           <thead><!-- Header Definition--></thead>
11           <tbody>
12           <tr ng-repeat="result in results">
13               <td>{{result.run}}</td>
14               <td>{{result.m}}</td>
15               <td>{{result.n}}</td>
16           </tr>
17           </tbody>
18         </table>
19
```

Applikation

Javascript

Controller

Direktiven

Expressions

Beispiel einer AngularJS Anwendung

```
1  var avaWebApp = angular.module('avaWebApp', ['ngStorage']);
2
3  avaWebApp.controller('avaMainCtrl',
4          function ($scope, $http, $localStorage) {
5      $scope.$storage = $localStorage;
6      $scope.running = false;
7      $scope.getData = function (){
8          $http({
9              method: 'GET',
10             url: '/runData'
11         }).then(function successCallback(response) {
12             console.log('Got a response:' + response.data);
13             $scope.results = response.data;
14         }, function errorCallback(response) {
15             // called asynchronously if an error occurs
16             // or server returns response with an error status.
17         });
18     };
19     ...
20
```

Service

Applikation

Controller

Filtern in AngularJS

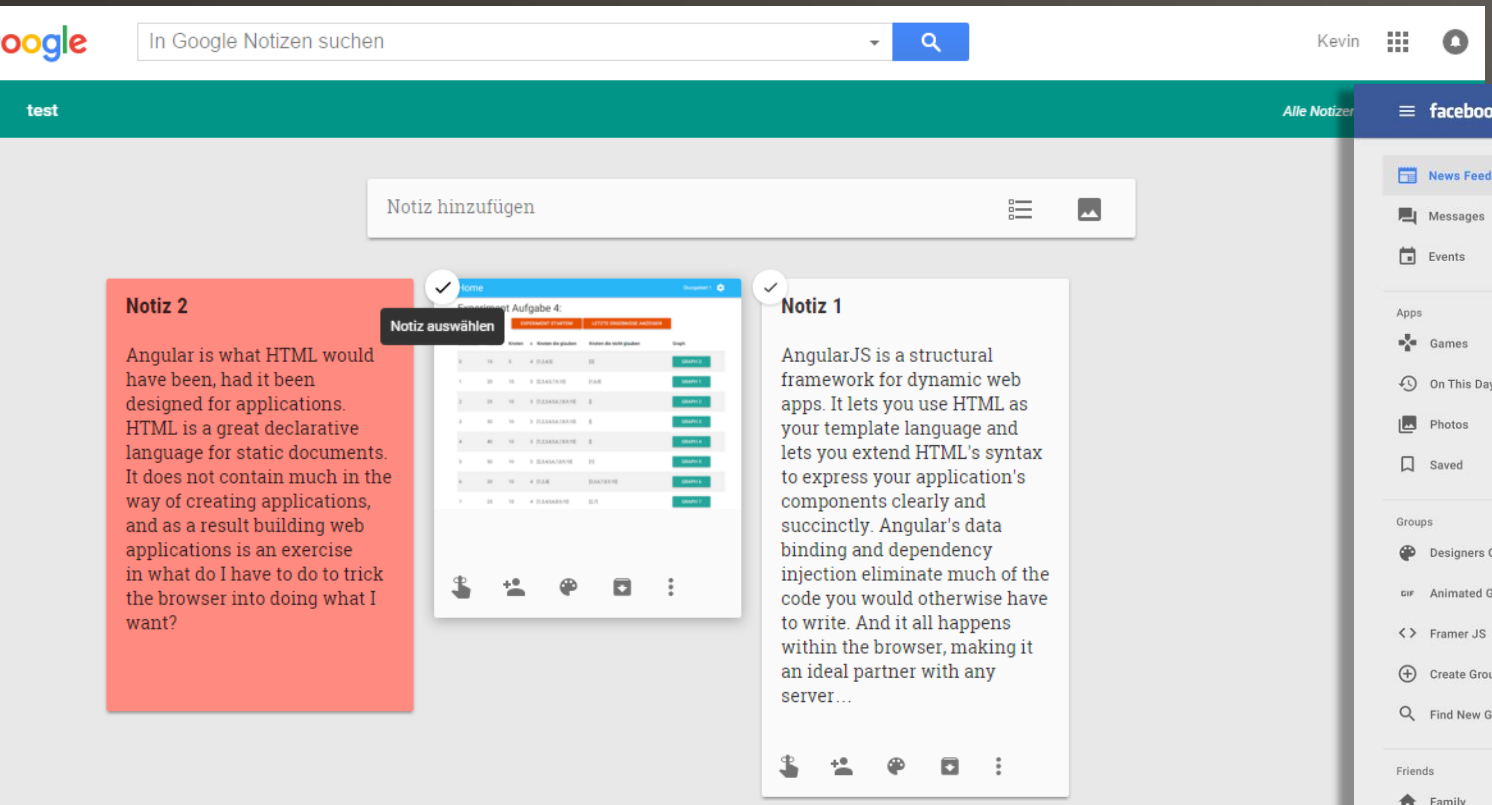
- durch „|“ Symbol verwendbar
- eigene Definition möglich

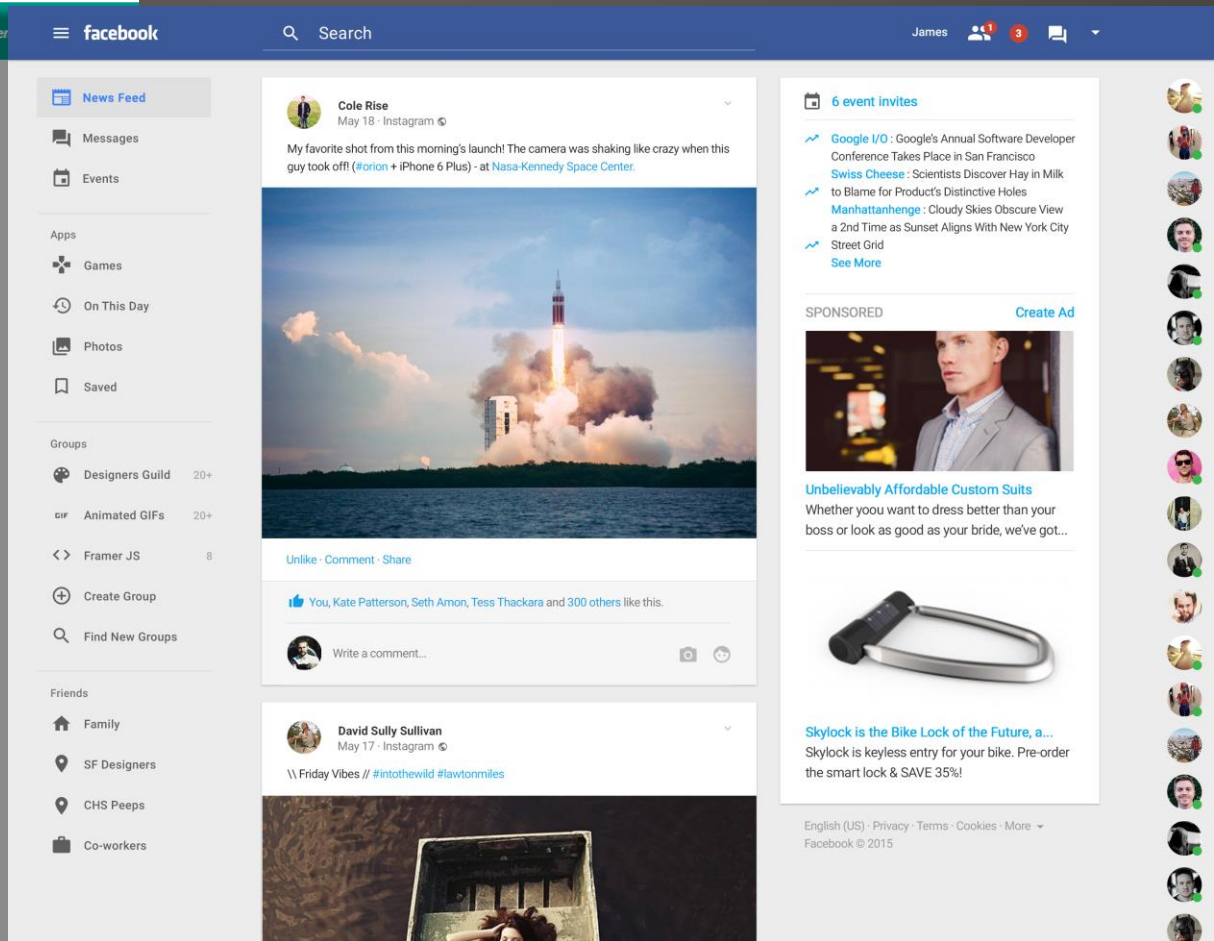
- Beispiel:

```
ng-repeat="result in results | filter: searchText"
```

Material Design

Google's Designsprache (design.google.com/)





materializecss

- CSS Framework
- orientiert sich an Google‘s Material Design
- von Studenten der Carnegie Mellon University (Pittsburgh, PA)
- visuelle Führung
- responsive Design
- grid Layout

materializecss einbinden

```
1  <html>
2      <head>
3        <!--Import Google Icon Font-->
4        <link href="http://fonts.googleapis.com/icon?family=Material+Icons" re
5        <!--Import materialize.css-->
6        <link type="text/css" href="css/materialize.min.css"/>
7      </head>
8      <body>
9        <!--Import jQuery before materialize.js-->
10       <script type="text/javascript" src="https://code.jquery.com/jquery-2.
11       <script type="text/javascript" src="js/materialize.min.js"></script>
12     </body>
13   </html>
14
```

Komponenten (Auswahl)

Button, gerendert:

Definition im HTML:

```
1  <a class="waves-effect waves-light orange darken-4 btn">
2      Experiment starten!</a>
3  <a class="waves-effect waves-light orange darken-4 btn">
4      Letzte Ergebnisse anzeigen</a>
```

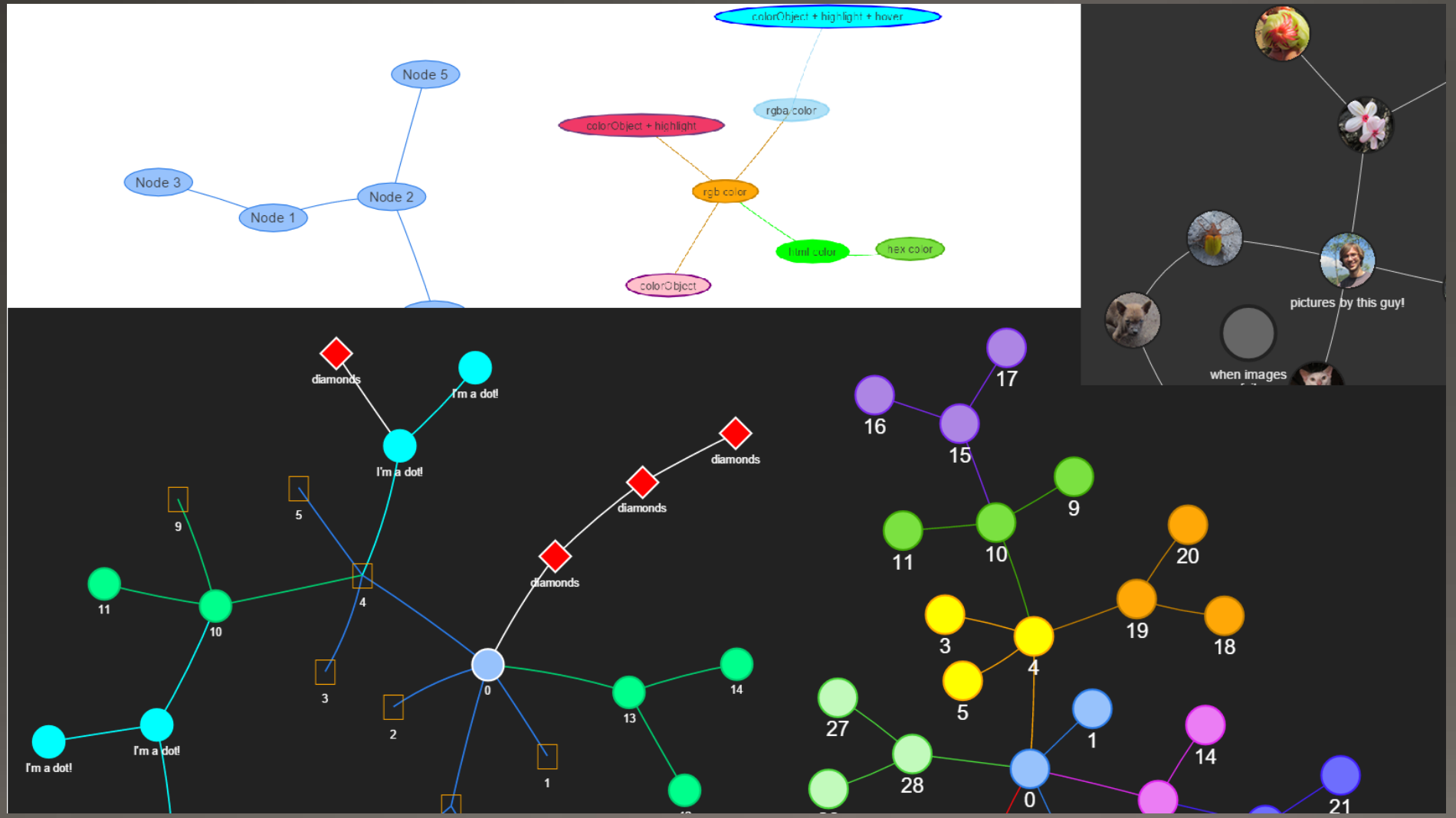
Komponenten (Auswahl)

Preloader, gerendert:

Definition im HTML:

```
1  <div class="progress">
2      <div class="indeterminate"></div>
3  </div>
```

vis.js


Node 5
Node 3
Node 2
Node 1
colorObject + highlight + hover
rgba color
colorObject + highlight
rgb color
hex color
colorObject
pictures by this guy!
when images
diamonds
I'm a dot!
I'm a dot!
diamonds
diamonds
diamonds
I'm a dot!
I'm a dot!
I'm a dot!
9
5
11
10
4
3
2
0
1
13
14
17
16
15
9
11
10
20
3
19
18
4
5
27
28
1
14
0
21

vis.js verwenden

```
1  var parsedData = vis.network.convertDot(dotData);
2  var data = {
3          nodes: parsedData.nodes,
4          edges: parsedData.edges
5      }
6  var options = parsedData.options;
7  options.nodes = {}
8  var container = document.getElementById('graphNetwork');
9  network = new vis.Network(container, data, options);
```

Demo...

Home

Übungsblatt 1

Experiment Aufgabe 4:

EXPERIMENT STARTEN! LETZTE ERGEBNISSE ANZEIGEN

Durchlauf	Kanten	Knoten	c	Knoten die glauben	Knoten die nicht glauben	Graph
0	15	5	4	[1,2,4,5]	[3]	GRAPH 0
1	20	10	3	[2,3,4,5,7,9,10]	[1,6,8]	GRAPH 1
2	25	10	3	[1,2,3,4,5,6,7,8,9,10]	[]	GRAPH 2
3	30	10	3	[1,2,3,4,5,6,7,8,9,10]	[]	GRAPH 3
4	40	10	3	[1,2,3,4,5,6,7,8,9,10]	[]	GRAPH 4
5	50	10	3	[2,3,4,5,6,7,8,9,10]	[1]	GRAPH 5

Quellen

AngularJS

- https://docs.angularjs.org/guide
- https://github.com/gsklee/ngStorage

vis.js

- http://visjs.org/docs/network/

materialize CSS

- http://materializecss.com/
- http://www.materialup.com/
- https://design.google.com/