



ANGULAR

Introduction to Angular

By: Glenn Greibesland



One framework.
Mobile & desktop.

- Learn one way to build applications with Angular and reuse your code and abilities to build apps for any deployment target. For web, mobile web, native mobile and native desktop.
- The Angular Core team consists of about 30 Google employees and about 32 community members.



Angular 2.0.0 released September 14 2016

Angular 4.0.0 released March 22 2017



Angular v4.0 is out! Smaller, faster, no biggie...

Improvements/Changes in Angular 4

- Production build sizes reduced by 60% in most cases
- Ships with server side rendering support
 - Universal (originally community driven) adopted by angular.
- TypeScript 2.1 and 2.2 compatible
 - Better type checking.

Tentative Schedule After March 2017

Date	Stable Release	Compatibility
September/October 2017	5.0.0	^4.0.0
March 2018	6.0.0	^5.0.0
September/October 2018	7.0.0	^6.0.0

"Angular is the name for the Angular of today and tomorrow.
AngularJS is the name for all v1.x versions of Angular."

- *angular.io*

Angular Architecture

- Angular = Components
- Closer to the Web Components' standard
- The entire web app is a component consisting of other components.
- Create components that don't depend on each other, which are as loosely coupled as possible.

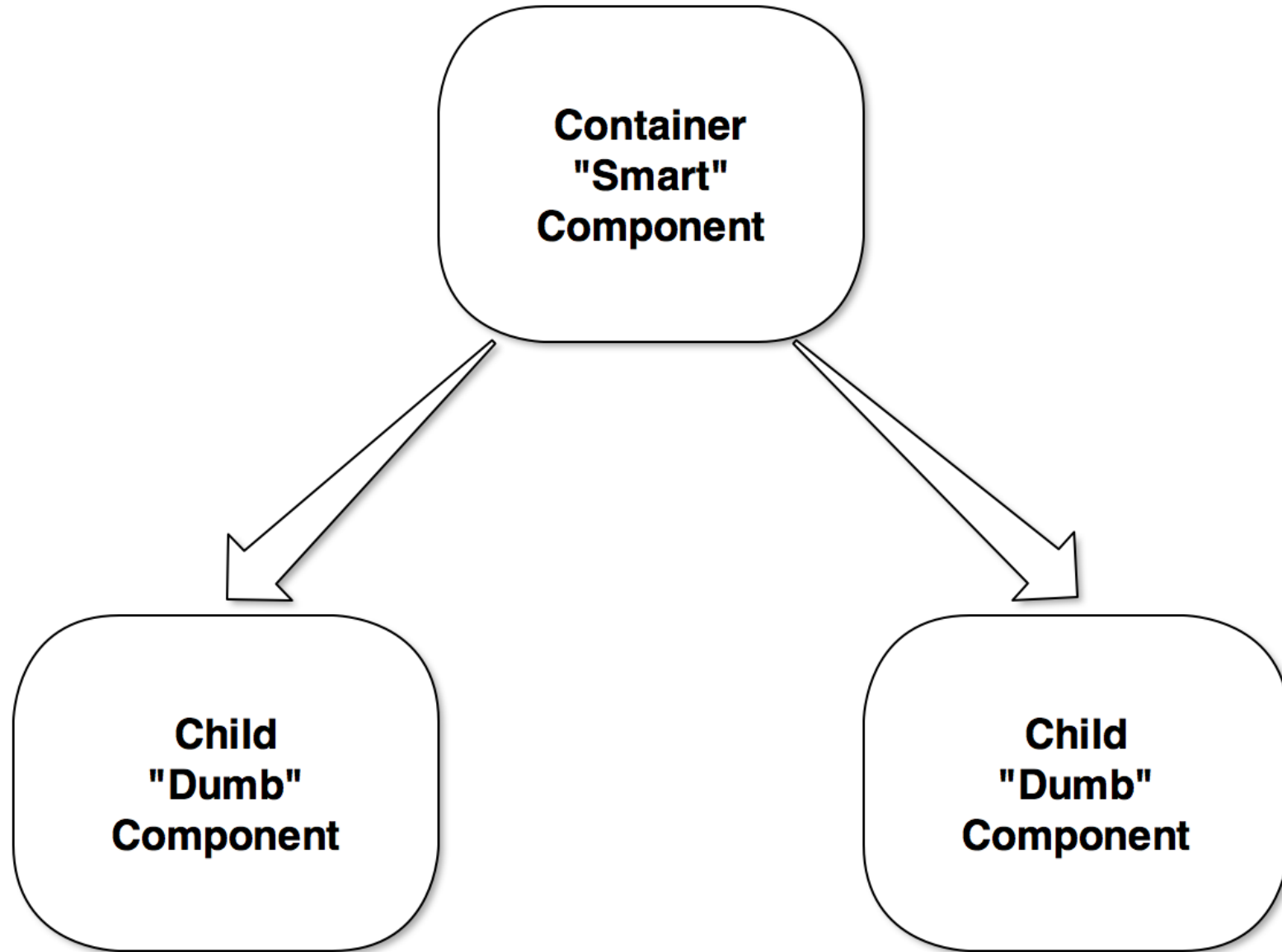
Two ways of creating components

Smart components

- Knows about application state
- Can communicate with services to fetch or modify data.

Presentation components (“Dumb” components)

- They should only have inputs and outputs.
- Can be placed anywhere in the system (or outside)
- Should not know about application state existence.



Writing Angular Apps

- Angular can be written in TypeScript, JavaScript... or Dart
- TypeScript and JavaScript: **<http://angular.io>**
 - TS & JS has the same codebase
- Dart version: <https://webdev.dartlang.org/angular/>
 - Dart version has a different codebase and its own dedicated team

TypeScript is the recommended language to use.

Creating a new Angular project

- Install Node
 - NPM is also a requirement, but it comes bundled with Node
 - Download here: <https://nodejs.org/>

- Install TypeScript

```
npm install -g typescript
```

- Install a code editor, if you don't have one
 - Visual Studio Code is recommended

Getting started using Angular-CLI

- CLI for Angular applications based on the ember-cli project.
- Angular CLI v1.0 released March 24.
- Use it to generate new Angular projects, or components for existing projects.
- As of v1.0, defaults to creating Angular4 projects.
- Install: `npm install -g @angular/cli`

Create your first app with Angular-CLI

- Command for creating a new project called "Webdev-Kristiansand-Demo"

```
ng new Webdev-Kristiansand-Demo
```

- Automatically creates the project and recommended project structure
- Installs the package dependencies with npm
- Creates a new GIT repository and creates the first commit
- Includes setup for running unit tests and e2e tests
- Setup for running local web development server
- Linting / Code analysis

```
$ ng new webdevkrsand-angular-intro
installing ng
  create .editorconfig
  create README.md
  create src\app\app.component.css
  create src\app\app.component.html
  create src\app\app.component.spec.ts
  create src\app\app.component.ts
  create src\app\app.module.ts
  create src\assets\.gitkeep
  create src\environments\environment.prod.ts
  create src\environments\environment.ts
  create src\favicon.ico
  create src\index.html
  create src\main.ts
  create src\polyfills.ts
  create src\styles.css
  create src\test.ts
  create src\tscconfig.app.json
  create src\tscconfig.spec.json
  create src\typings.d.ts
  create .angular-cli.json
  create e2e\app.e2e-spec.ts
  create e2e\app.po.ts
  create e2e\tscconfig.e2e.json
  create .gitignore
  create karma.conf.js
  create package.json
  create protractor.conf.js
  create tsconfig.json
  create tslint.json
Successfully initialized git.
Installing packages for tooling via npm.
Installed packages for tooling via npm.
Project 'webdevkrsand-angular-intro' successfully created.
```

```
▸ e2e
▸ node_modules
▾ src
  ▸ app
  ▸ assets
  ▸ environments
  favicon.ico
  index.html
  main.ts
  polyfills.ts
  styles.css
  test.ts
  tsconfig.app.json
  tsconfig.spec.json
  typings.d.ts
  .angular-cli.json
  .editorconfig
  .gitignore
  karma.conf.js
  package.json
  protractor.conf.js
  README.md
  tsconfig.json
  tslint.json
```

Getting started using a starter project

- Download or clone a starter-project from GitHub
- Run npm install from the project directory
- Example starter projects:
 - [github: angular/quickstart \(4.0.0\)](#)
 - [github: qdouble/angular-webpack2-starter](#)
 - [github: angularclass/angular2-webpack-starter \(2.0.0\)](#)

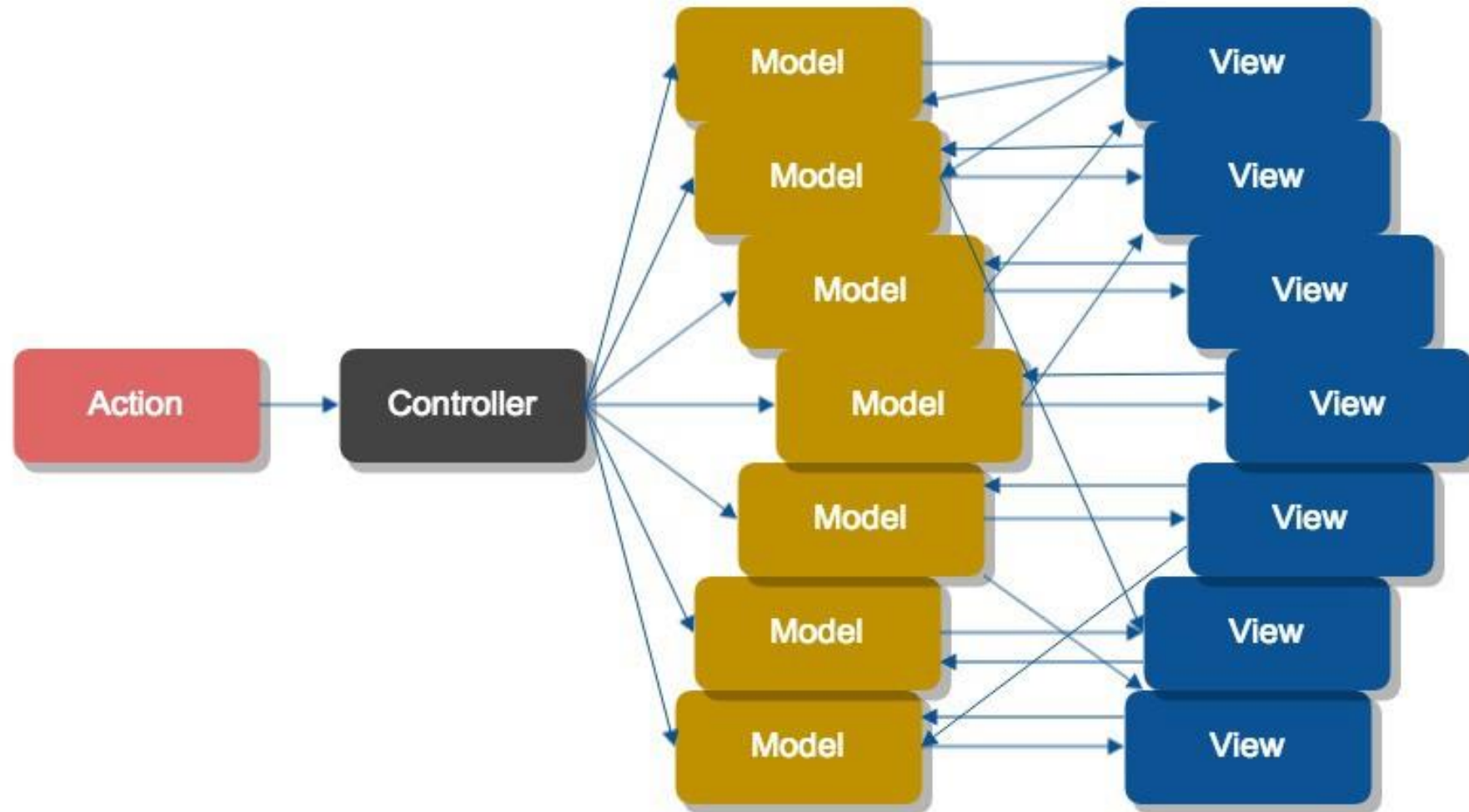
My recommended starter project

- [qdouble/angular-webpack2-starter](#)
- **(minimal branch)**
 - Includes a few components
 - Uses the Angular Router and Lazy loading of modules
 - Nice for coding examples

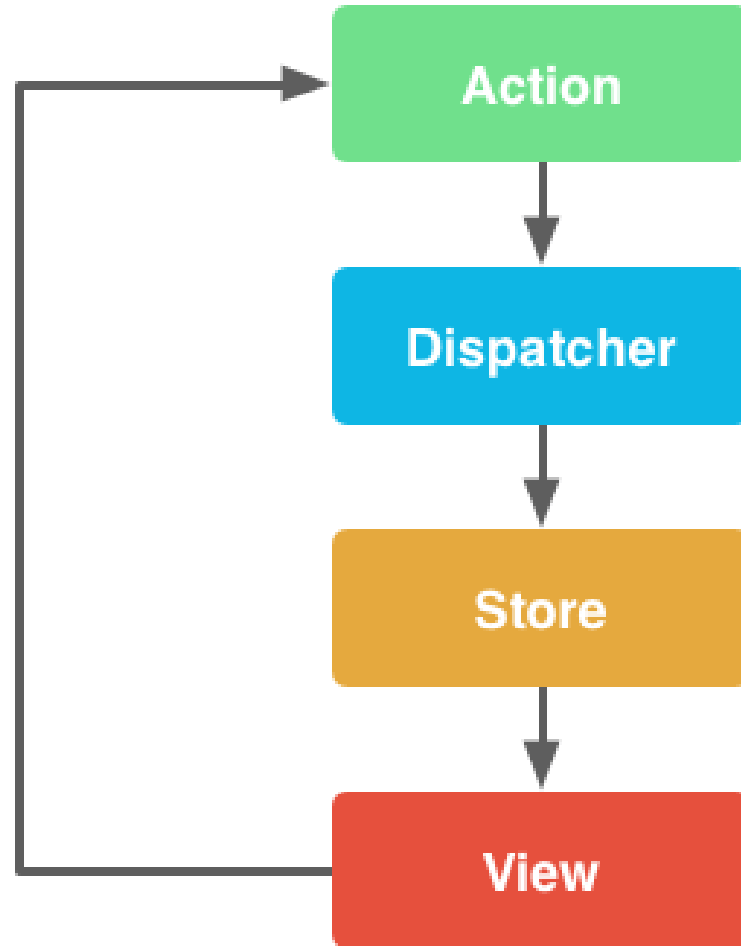
My recommended starter project

- **qdouble/angular-webpack2-starter**
- **(no-universal-support branch)**
 - HMR (Hot Module Replacement)
 - Includes Material Design (@angular/material)
 - State management with @ngrx/store (<https://github.com/ngrx/store>)
- *Alternative without @ngrx: angularclass/angular2-webpack-starter*

MVC / MVP / MVVM



Redux - Flux - @ngrx/store

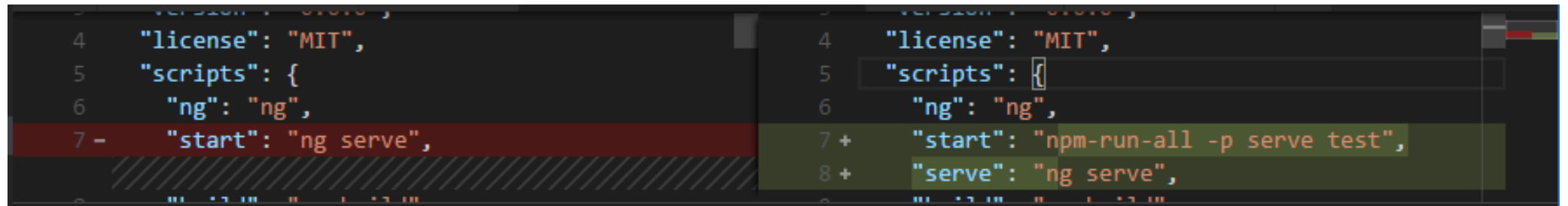


Adding productivity tools

- Running unit tests while you code

```
npm install -save-dev npm-run-all
```

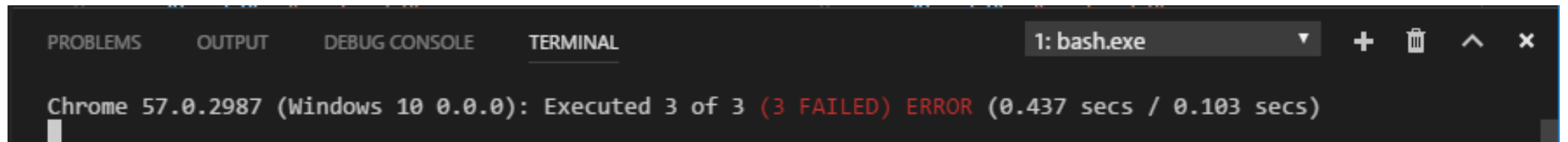
- Edit package.json:



The image shows two side-by-side code editors displaying the 'scripts' section of a package.json file. The left editor shows the original configuration with 'start' set to 'ng serve'. The right editor shows the modified configuration where 'start' is 'npm-run-all -p serve test' and a new 'serve' script is added, set to 'ng serve'.

```
4  "license": "MIT",  
5  "scripts": {  
6    "ng": "ng",  
7 -  "start": "ng serve",  
7 +  "start": "npm-run-all -p serve test",  
8 +  "serve": "ng serve",
```

```
npm start
```



The image shows a terminal window with the title '1: bash.exe'. The output displays the results of running 'npm start', indicating that 3 tests failed.

```
PROBLEMS OUTPUT DEBUG CONSOLE TERMINAL 1: bash.exe  
  
Chrome 57.0.2987 (Windows 10 0.0.0): Executed 3 of 3 (3 FAILED) ERROR (0.437 secs / 0.103 secs)
```

Adding productivity tools

- **Visual Studio Code extensions**
 - Angular Language Service (Angular)
 - Angular v4 TypeScript Snippets (johnnpapa)
 - TSLint (egamma)
 - Debugger for Chrome (Microsoft)
- *Needs some work, but still useful if you don't like ng-cli*
 - *Angular-Files (Alexander Ivanichev)*

Compiling for production

JIT

- Just in Time compilation
 - The Angular compiler JS library is sent to the browser, and the browser compiles "on the fly"

AOT

- Ahead of Time compilation
 - App is compiled to plain JavaScript, ready to be consumed by a browser

Compiling for production

- Using Angular-CLI

```
ng build -aot --prod
```



```
npm build
```

- Minifies/uglifyfies the code
- Tree-shaking (removes modules that are not in use)

Compiled output

Output File	Description
dist/main.[hash].bundle.js	Your application bundled
dist/vendor.[hash].bundle.js	Bundled dependencies (@angular, RxJS...)
dist/polyfill.[hash].bundle.js	Bundled polyfill dependencies
dist/index.html	entry point of your application.
dist/inline.[hash].bundle.js	webpack loader
dist/style.[hash].bundle.css	the style definitions
dist/assets	resources copied from the Angular CLI assets configuration

Compiled output size comparison

- Project with ~15 components, ~5 modules, ~5 services and a couple of directives

Angular Version	Type	Payload Size	Download Size	Document Load	Load
2.4.0	JIT	9.7M	5.3M	1.78s	2.52s
2.4.0	AOT	9.4M	5M	786ms	1.39s
2.4.0	AOT / PROD	1.5M	5M	786ms	1.39s
4.0.0	JIT	8.4M	4.3M	1.49s	2.00s
4.0.0	AOT	8.2M	4.3M	734ms	1.21s
4.0.0	AOT / PROD	1.3M	1.7M	473ms	1.00s

Questions?

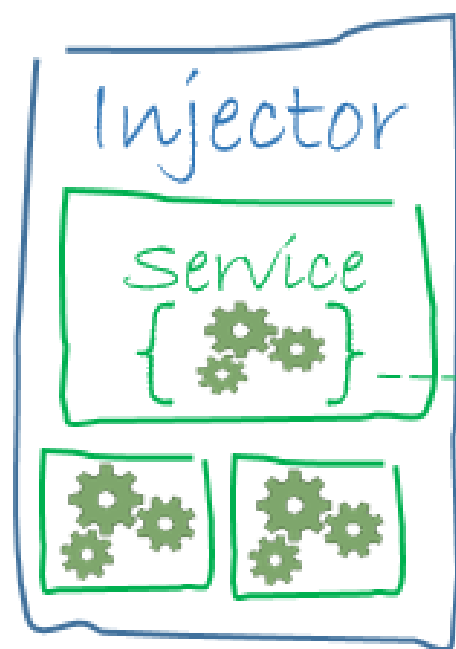


Coding Angular Apps

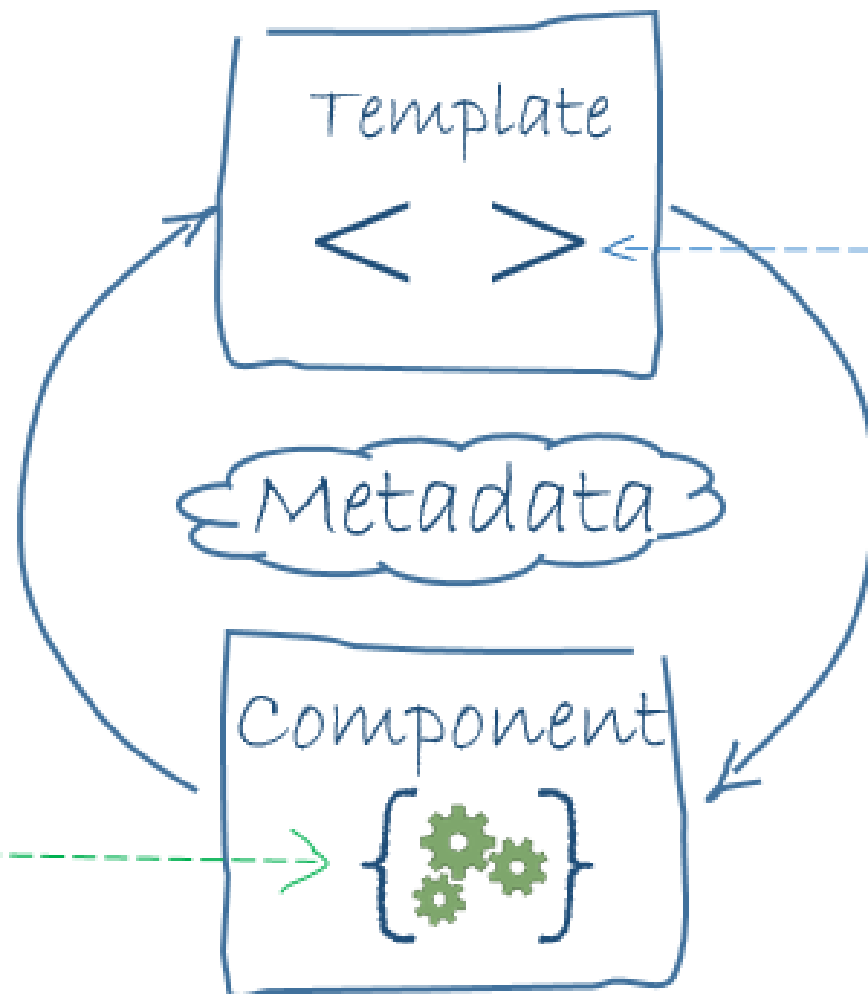
- Quick look at the CLI-generated project
 - Decorators (NgModule, Component etc.)
 - Creating components and modules
 - Exporting through barrels
 - Inputs, outputs, ng-content
 - Binding (One-way + event)
 - Angular IF- and FOR-syntax in HTML-templates
- Two-way binding
 - Angular Pipes
 - Angular Services and Dependency Injection
 - Angular Directives
 - HTTP server communication
 - ChangeDetectionStrategy



Module Component { }	Module Service { }
Module value 3.1415	Module Fn λ



Property Binding



Event Binding

