Internals

Zones

> *It's open-heart surgery on the web browser Miško Hevery*

A zone

- Is an execution context for asynchronous operations
- Keeps track of tasks by hooking into `setTimeout()`, `setInterval()`, etc.
- Let's you know how long a tasks takes
- Gives detailed stacktraces when something goes wrong
- Is a native Dart feature
- Has been proposed as a JavaScript feature
  - And until that time, we do it manually with Zone.js

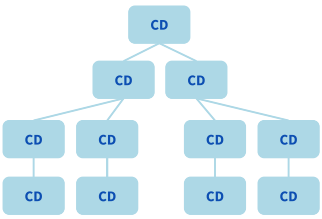
Angular and zones

- Zones are used by Angular to keep track of asynchronous tasks
- Angular exposes a custom zone: `NgZone`

Change detection

Angular uses Zones to get notified of **when something asynchronous has happened** (events, AJAX, timers)

Angular calls goes through the component tree **once from top to bottom** and tells it to update itself



- Data in a view can only come from the associated component
- A unidirectional data flow is more predictable than cycles with dirty checking

image source: https://www.sitepoint.com/angular-2-components-inputs-outputs/

Change detection internals

- Every component has its own change detector (CD)
- This is generated at runtime for every component, so that the JavaScript engine can perform optimizations

Change detection strategy

For performance reasons, it is possible to exclude a component from the change detection cycle.

```
@Component({
    templateUrl: 'some.component.html',
    changeDetection: ChangeDetectionStrategy.OnPush
})
export class SomeComponent implements DoCheck {
    @Input() data;
    private previousLength = 0;

    constructor(private cdr: ChangeDetectorRef) { }

    ngDoCheck() {
        if (this.data.length !== this.previousLength) {
            this.previousLength = this.data.length;
            this.cdr.markForCheck();
        }
    }
}
```

AngularJS and Angular differences

Feature	AngularJS	Angular
Change detection	Dirty checking (digest cycle)	Zones
Dependency injection	Parameter names	Parameter types
Forms	`name` exposed metadata	`ngForm`, `ngControl`, `#trvs`
Events	`$broadcast()` and `$emit()` on `$scope`	`@Input()` and `@Output()`
Data formatting	Filters	Pipes

Big changes in Angular

- **AngularJS**, a completely separate framework.
- **Angular 2** (September 14, 2016)
  - The first final release of the new Angular framework.
  - TypeScript with classes for defining components.
  - Observables over promises.
  - New change detection lifecycle.
- **Angular 4** (March 23, 2017)
  - Enhanced `*ngIf` (`else`) and `*ngFor` (`as`).
  - Server-side rendering with Angular Universal.
  - Animations moved to a separate module.

Big changes in Angular

- **Angular 5** (November 1st, 2017)
  - `ReflectiveInjector` is deprecated by `StaticInjector`.
  - `Http` (`@angular/http`) is deprecated by `HttpClient` (`@angular/common/http`)
    - With support for interceptors.
  - The `Intl` API has been dropped.
  - Introduction of the `@angular/service-worker`.
- **Angular 6** (May 3rd, 2018)
  - RxJS 6, breaking changes in using `Observables`.
  - Workspace for multiple projects
  - Lazy loaded routes
  - Moving towards web components with Angular Elements

Big changes in Angular

- **Angular 7** (Oktober 18th, 2018)
  - CLI prompts different options
  - Angular Material updates
- **Angular 8** (May 5th, 2019)
  - The Ivy compiler is introduced for dealing with Angular expressions
  - New syntax for lazy-loading routes
- **Angular 9** (February 6th, 2020)
  - Ivy is now the default rendering engine