

Ten interesting features of Google's Angular Project

Foundations of Web Apps Need To Evolve

Need a web programming language that is ...

- Suitable for large-scale apps
- Object-oriented and strongly typed

Need a framework that is ...

- Comprehensive, well-designed, flexible, modern
- From well-funded, large teams of experienced devs

TypeScript and Angular deliver

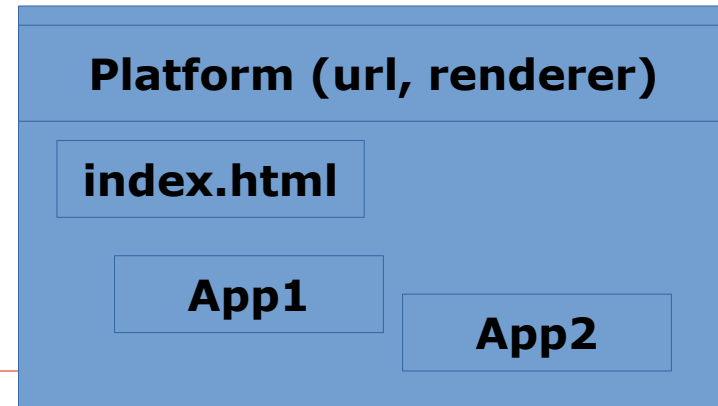
SPA (Single Page Application)

One top-level HTML, whose element tree is altered as app runs

- Via routing, URL that appears in browser's address bar varies too

Multiple apps may run on same page

Platform instance manages singletons, location & rendering



Why We Choose Angular

Angular is a complete framework

- Excellent choice for large applications
- Large team behind it
- In addition to views, also support dependency injection, routing, rich data binding, zones, web workers, ...

An Excellent Web Framework

Suitable for large, multi-year projects

- Comprehensive enterprise production systems

Comes "with batteries included"

- Has everything (well, most things) you need within the one framework, rather than in many bits

Support for multiple renderers

- Separation of rendering from core

Alternatives

React (Facebook)

- React is a library (for views)

jQuery

Coding to DOM directly

Homegrown framework

- Lots and lots of other choices

Angular And Backends

Angular (usually) runs in the web browser

Need to communicate with web server

- Usually via REST APIs (JSON)
- Angular has an HTTP (client) package to handle this

Any web server / framework will do – e.g.

- Nginx / Node.js 9.x / Express
- IIS 10 (Windows Server 2016) / ASP.NET Core

Family Of Projects

Angular is actually a family of projects

- Angular Main Project
- Angular CLI
- Angular Universal, Angular Material
- Angular Fire

External

- NativeScript, Ionic

Angular Main Project = Family Of Packages

Main project is a family of packages

- Core
- Common
- Common/HTTP
- Platforms
- Router
- Forms
- Compiler and Compiler-cli

Angular Material

Material Design spec - a Modern Style Guide

- Design for beautiful and functionally rich controls

Many control families can work with Angular

- Angular Material 2 is an implementation of the Material Design spec
- Components for .. button, cards, checkbox, radio, input, sidenav, toolbar, list, grid-list, icon, progress-circle, progress-bar, tabs, slide-toggle ...

1. Language Selection For Front-end

Really important decision

- Codebase and dev skills are valuable assets
- Very time-consuming/expensive to change a year later

Many language choices for web client

- JavaScript itself, TypeScript, Dart (transpiled to JS), Coffeescript, ... most languages have an avenue

We select TypeScript – why?

2. Building Web Components

(Domain) model

- Business logic + data

View

- .html templates - template compiler converts these to sets of calls to renderer

Controller

- These are web components

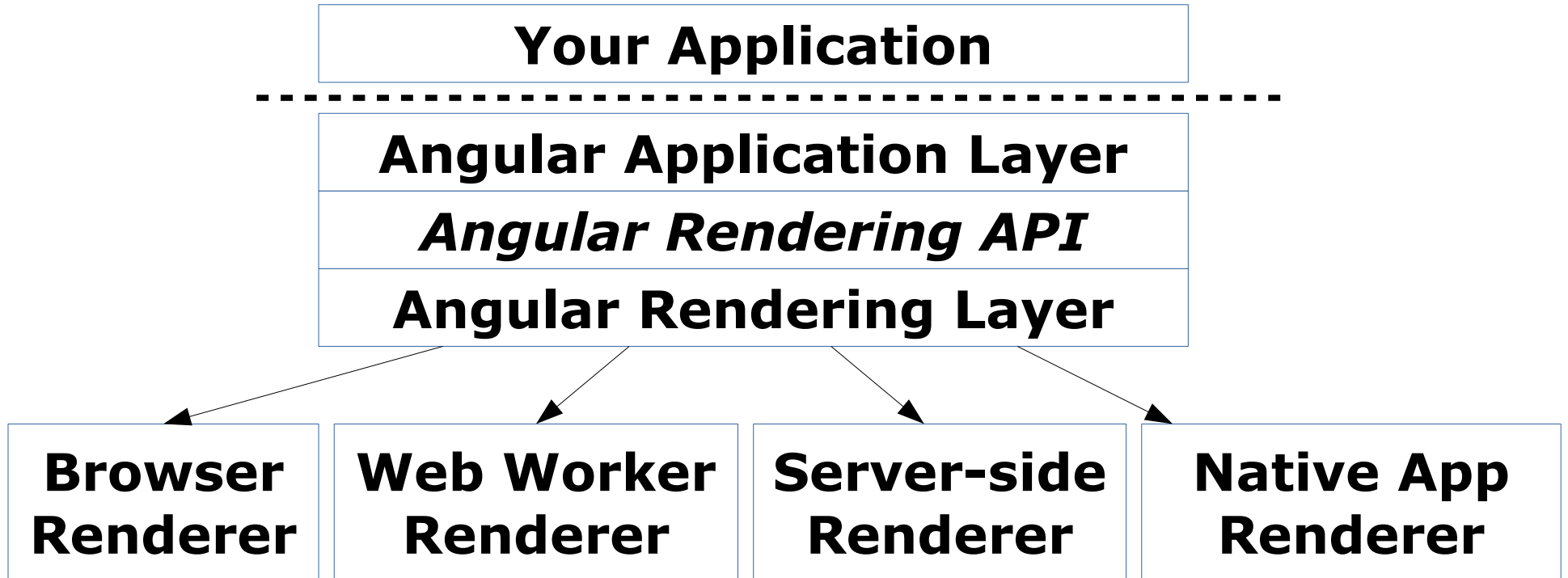
3. Dependency Injection

A hierarchy of injectors

If what is sought is not in current injector,
look at its parent

Platform is root of injector tree

4. Angular Rendering Architecture



Ionic and NativeScript

From Ionic and Telerik (Progress)

Different approaches to building native apps

- Installed from Apple App Store Or Google Play

Ionic uses Cordova and WebView

NativeScript does not use WebView

- More device-like app (more work)

5. Template Compiler

JIT vs. AOT compilation

- Anywhere you see "Compiler" in Angular docs, it refers to html template compilation
- Anywhere you see "dynamic" it refers to JIT

Output of AOT is code (not html) that calls the Renderer API

6. RxJS and Angular EventEmitter

Observable

Observer

Subject

Operators

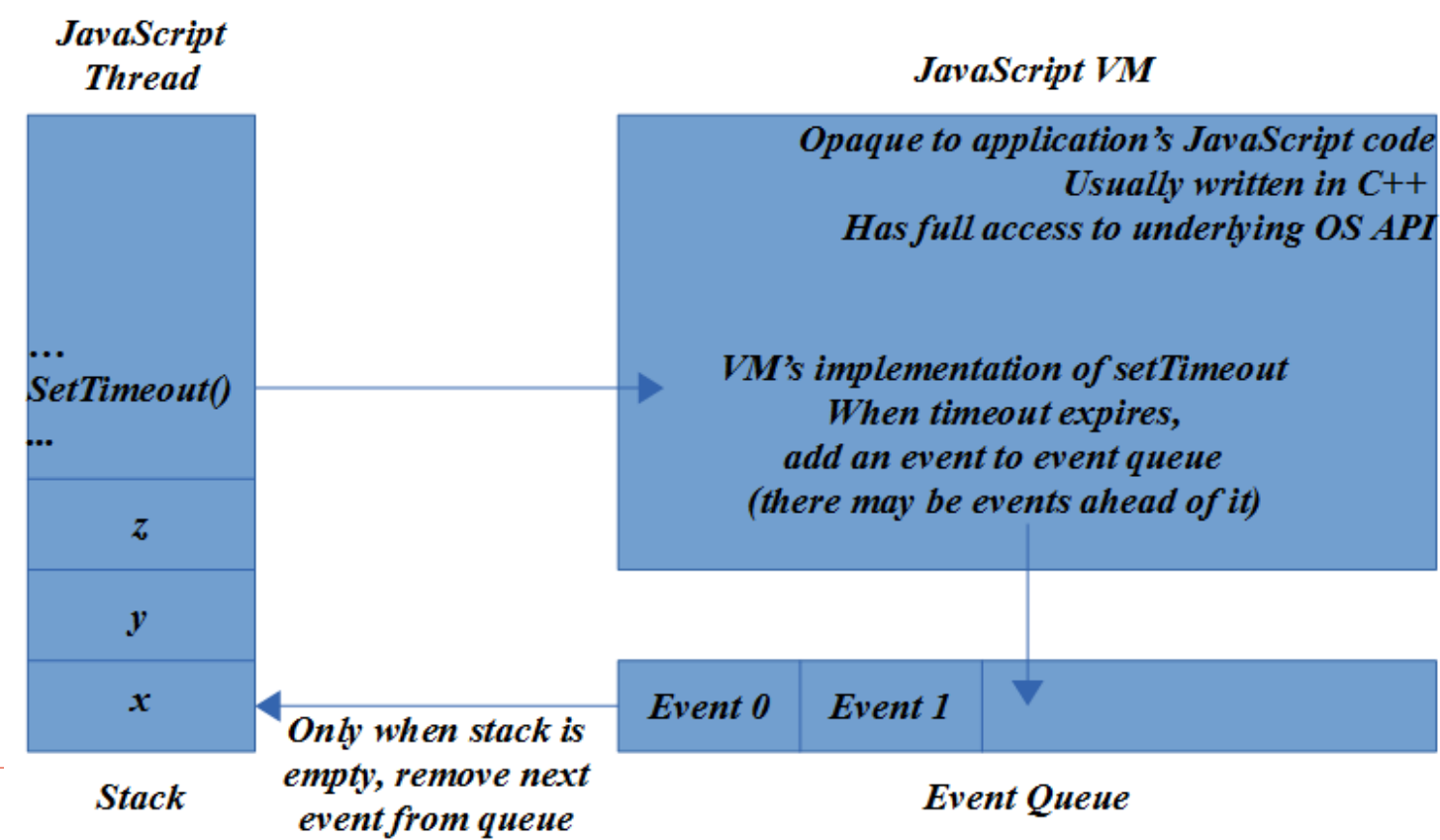
Use of EventEmitter

The JS VM Event loop

Promises

Async/await (TS 2.x)

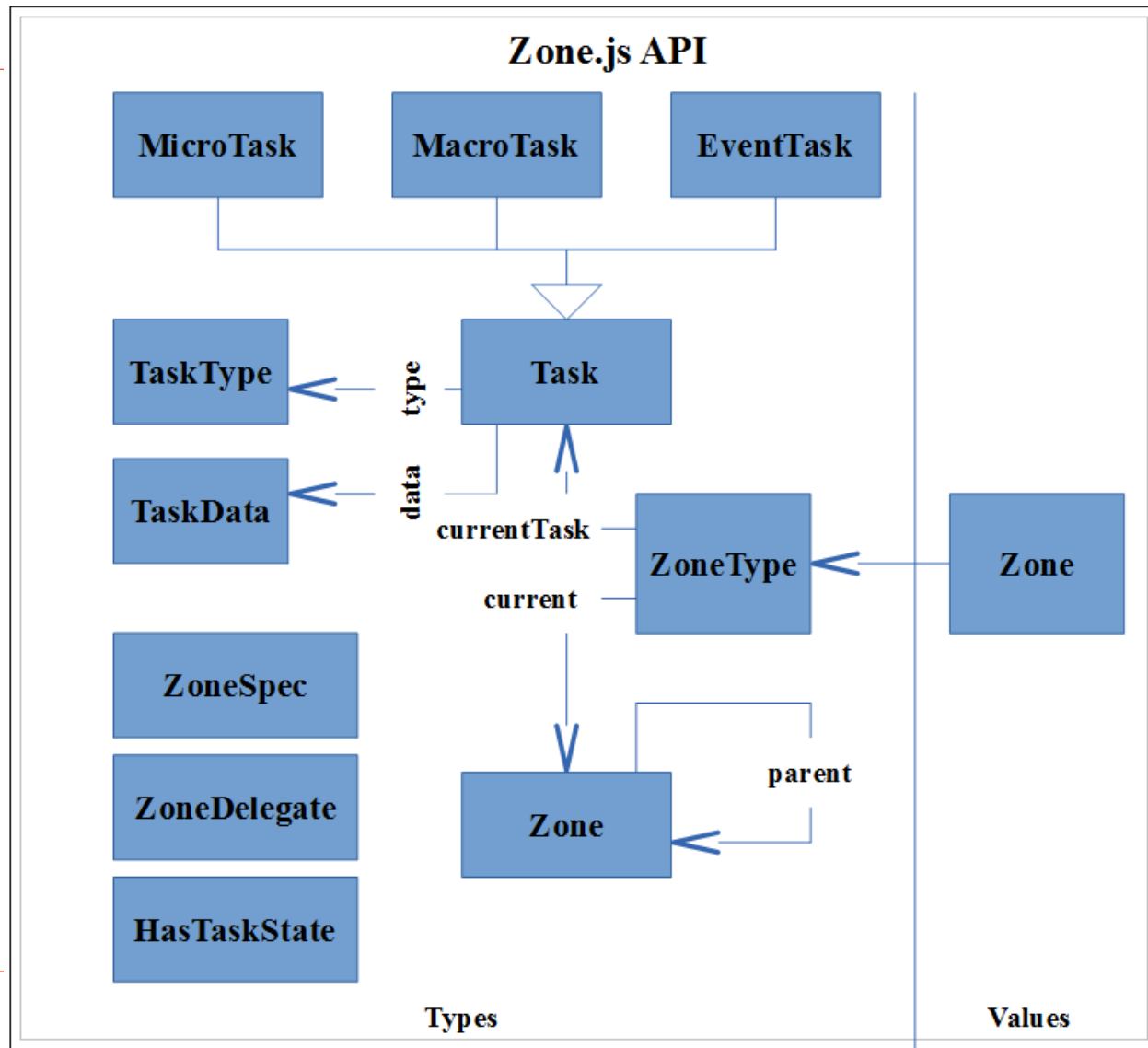
7. Async Thinking



8. Zone.js & NgZone

Subdividing A JS Thread

- Importance for Angular change detection



9. Angular Inside Web Workers

Message Bus

Message Broker

How UI events work with web workers

10. Angular App Architecture

Interaction Model (templates, data binding, styling, ng-xi18n)

Domain Model
(Domain Services/Entities)

Messaging Model
(HTTP client/REST)

Data Model
(RxJS)

Angular System Programming

Angular DevOps
(Angular CLI, WebPack,
ts-node, gulpfile.ts)

Angular CLI

Setting up and configuring an Angular project involves a number of tasks

Angular CLI is a command-line interface that automates common these tasks

- Creates projects
- Generates a variety of elements
- Deploys

11 (BONUS): Eamon's Ideas for Future

Protocols: HTTP/2, WebRTC, WebSockets

Rich graphics (Canvas, SVG, WebGL2)

Editor & Reporting Tools

Additional Renderers

- Shared workspace, Server-to-client (X Window like)
To document (.odt, .docx, PDF)

Platform (system, native, etc.)