Integrating Angular with ASP.NET Core RESTful Services

Dan Wahlin
WAHLIN CONSULTING

@DanWahlin www.codewithdan.com

Course Introduction

Dan Wahlin
WAHLIN CONSULTING

@DanWahlin www.codewithdan.com

Module Overview

- Prerequisites to maximize learning
- Learning goals
- Server-side Technologies and concepts
- Client-side technologies and concepts
- Running the sample application
- Running the sample application with Docker

Prerequisites to Maximize Learning

Course Prerequisites

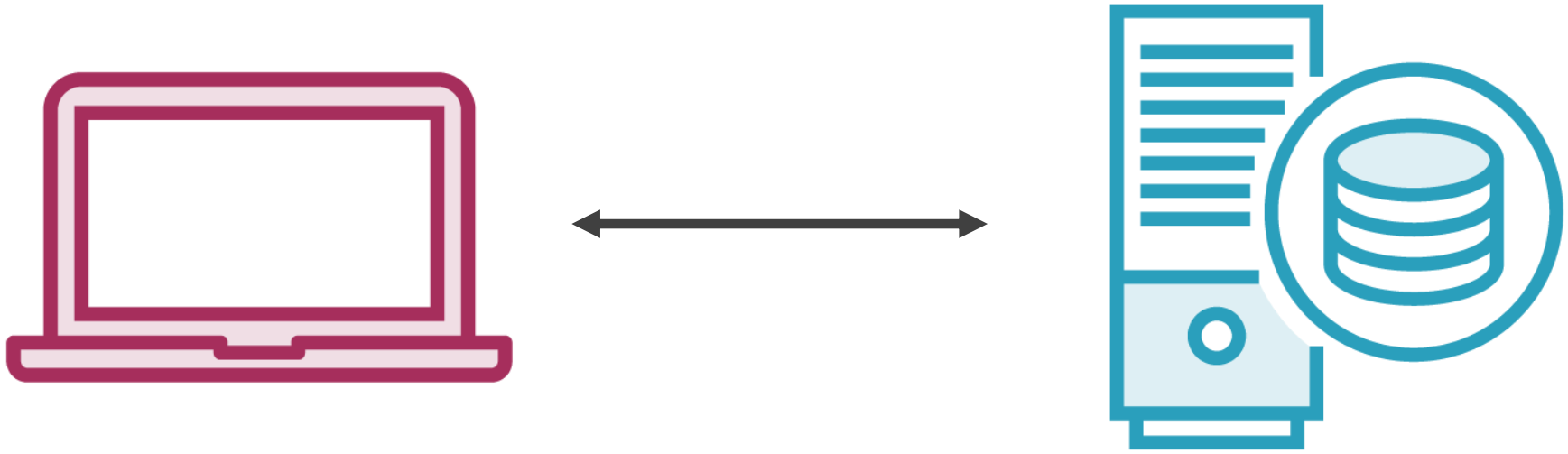
JavaScript/
TypeScript

Angular
Fundamentals

C# and
ASP.NET Core
Fundamentals

Learning Goals

Learning Goals



Client-Side

Server-Side

Server-Side Learning Goals

Learn how to use ASP.NET Core to create a RESTful service

- Create a web API using controllers, actions and attributes
- Create model classes
- Create a repository layer with Entity Framework Core
- Integrate with a database
- Provide web API documentation with Swagger

Client-Side Learning Goals

Learn how to use the Angular Http client to integrate with a RESTful service

- Understand the role of RxJS and observables
- Retrieve and display data from a RESTful service using Http
- Create, read, update and delete data (CRUD)
- Page data

Server-Side Technologies and Concepts

Server-Side Technologies and Concepts

ASP.NET Core

Database

HTTP

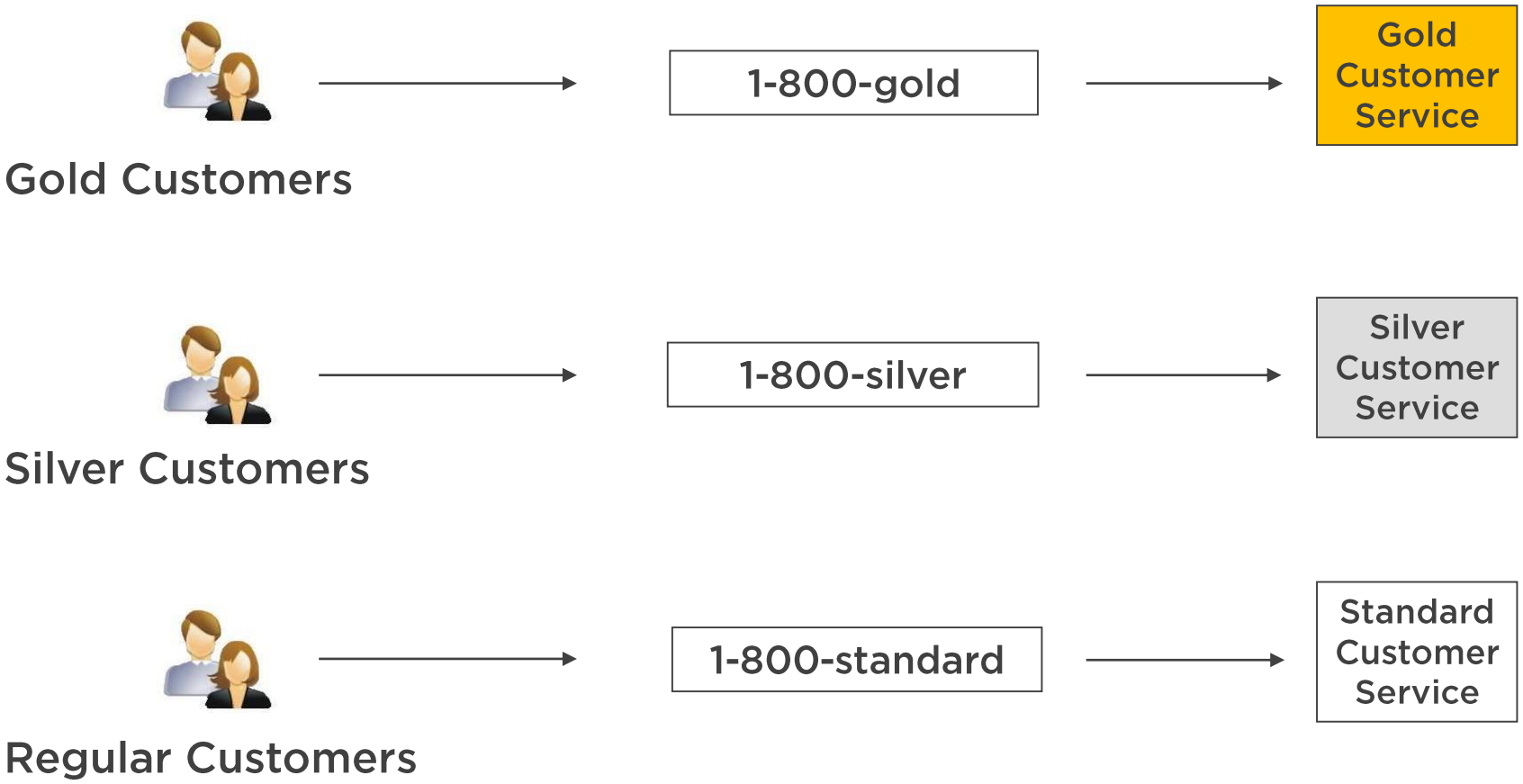
REST

Introduction to REST

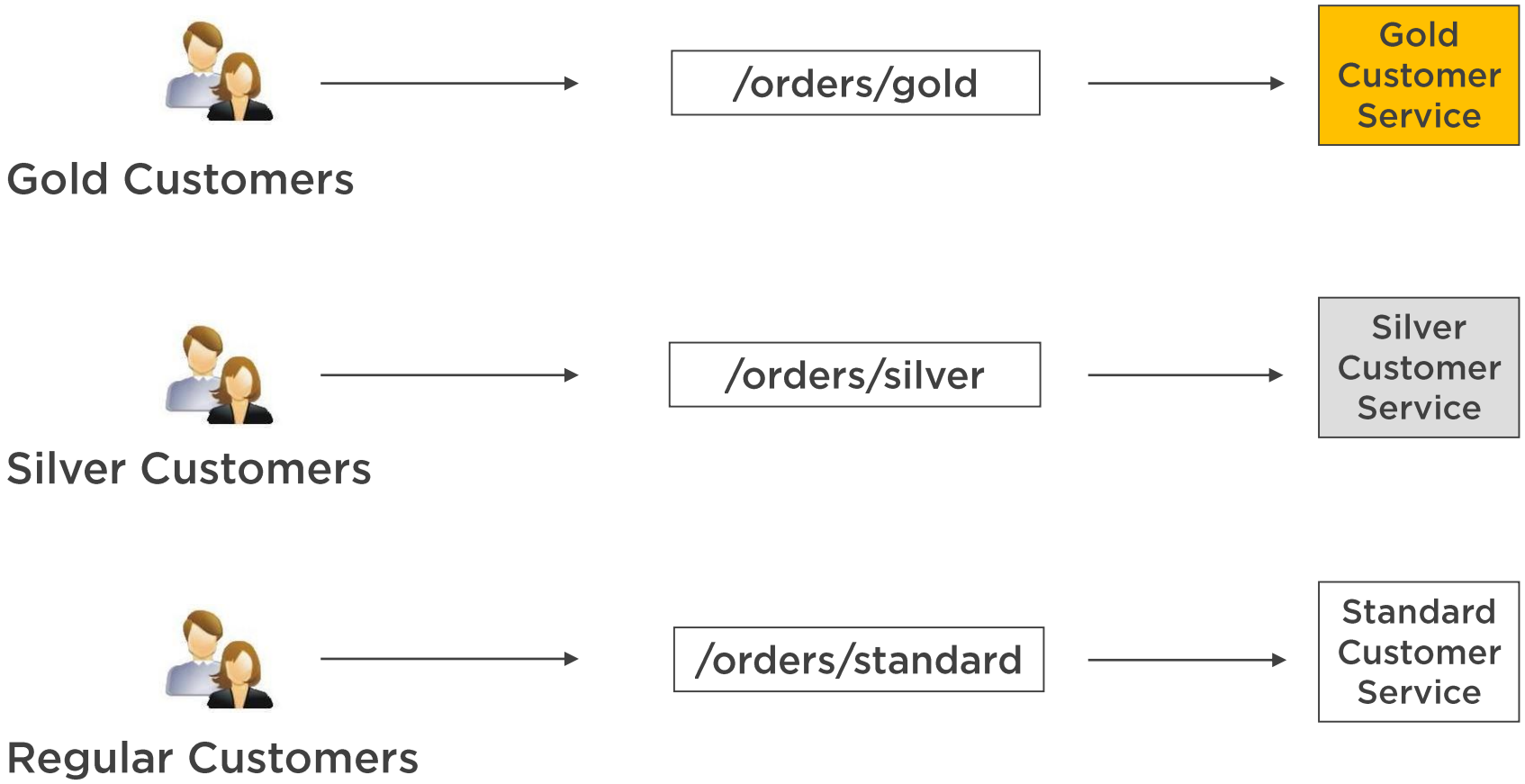
REST = Representational State Transfer

- Architectural style for distributed systems
- Exposes resources (state) to clients
- Resources identified with a URI
- Uses HTTP, URIs, MIME types

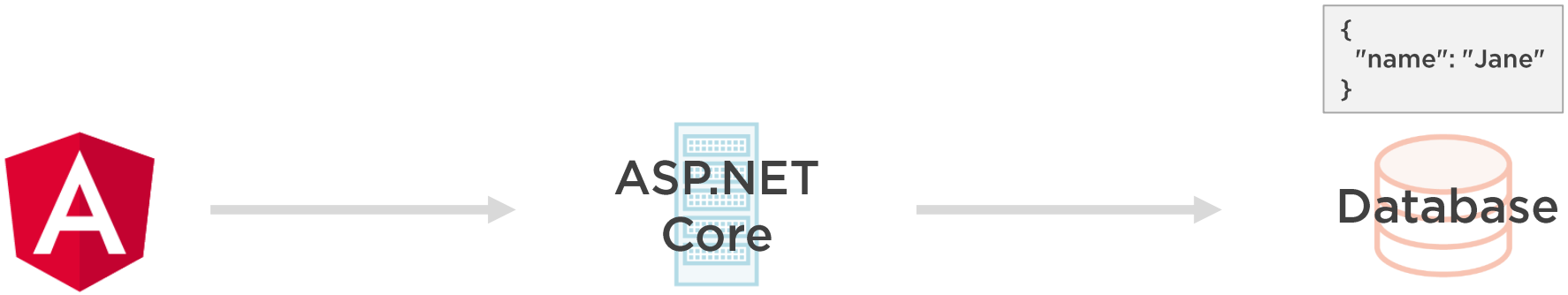
"RESTful" System Overview



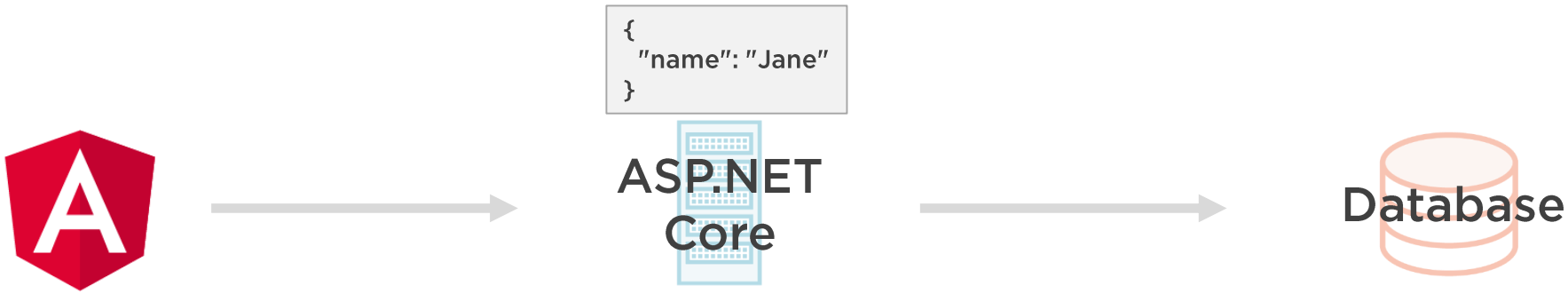
RESTful Services and URIs



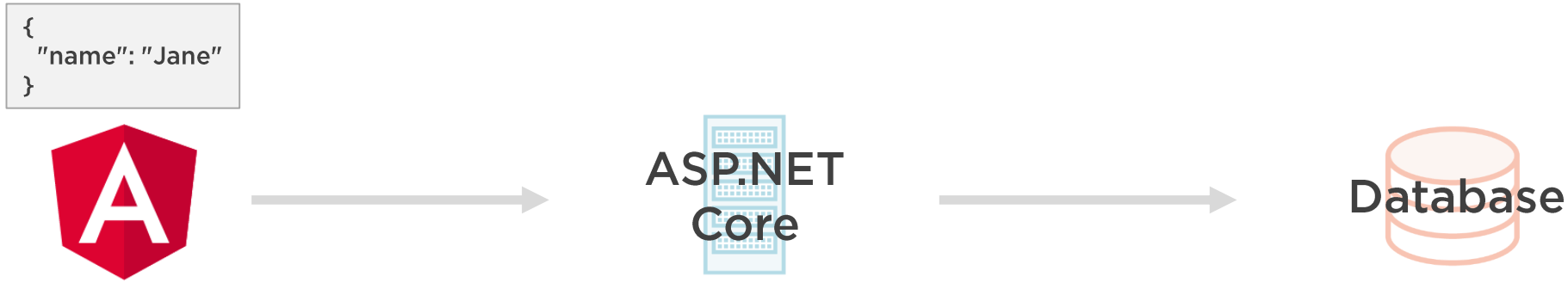
Key Technology Players



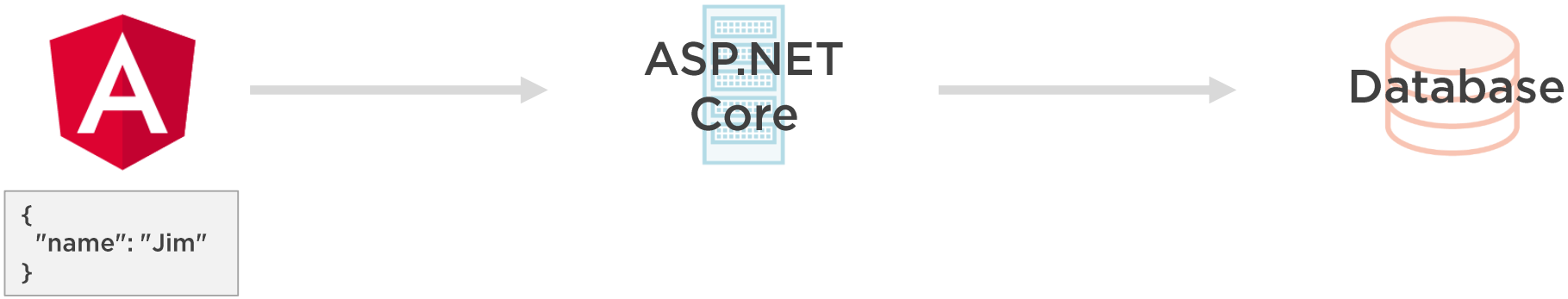
Key Technology Players



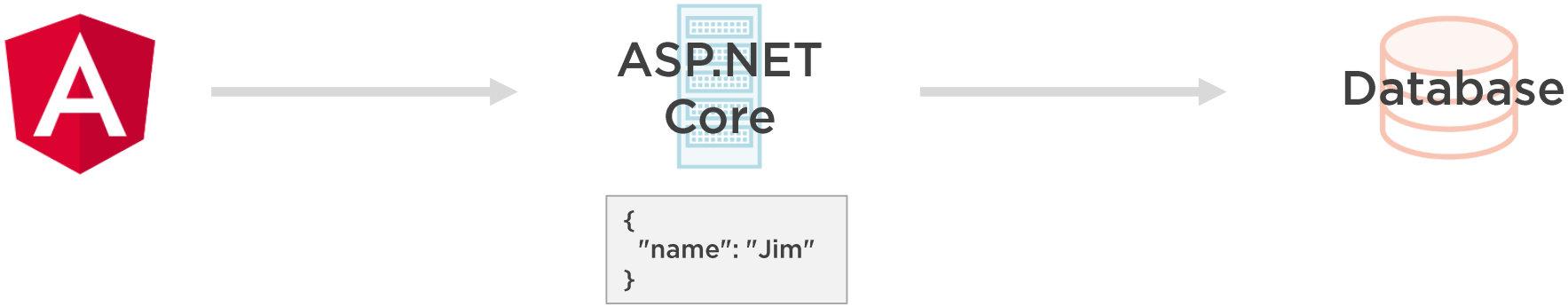
Key Technology Players



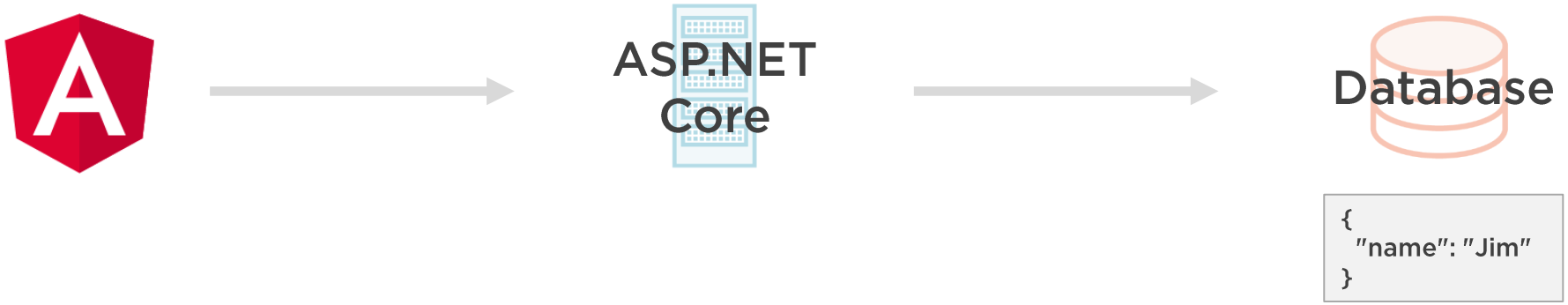
Key Technology Players



Key Technology Players



Key Technology Players



Client-Side Technologies and Concepts

Client-Side Technologies and Concepts

Angular

RxJS

XHR/HTTP

Observables

RxJS

http://reactivex.io/rxjs

Reactive Extensions for JavaScript

- Library for composing asynchronous and event-based programs
- Relies on observable sequences
- Used with Angular

Promises and Observables

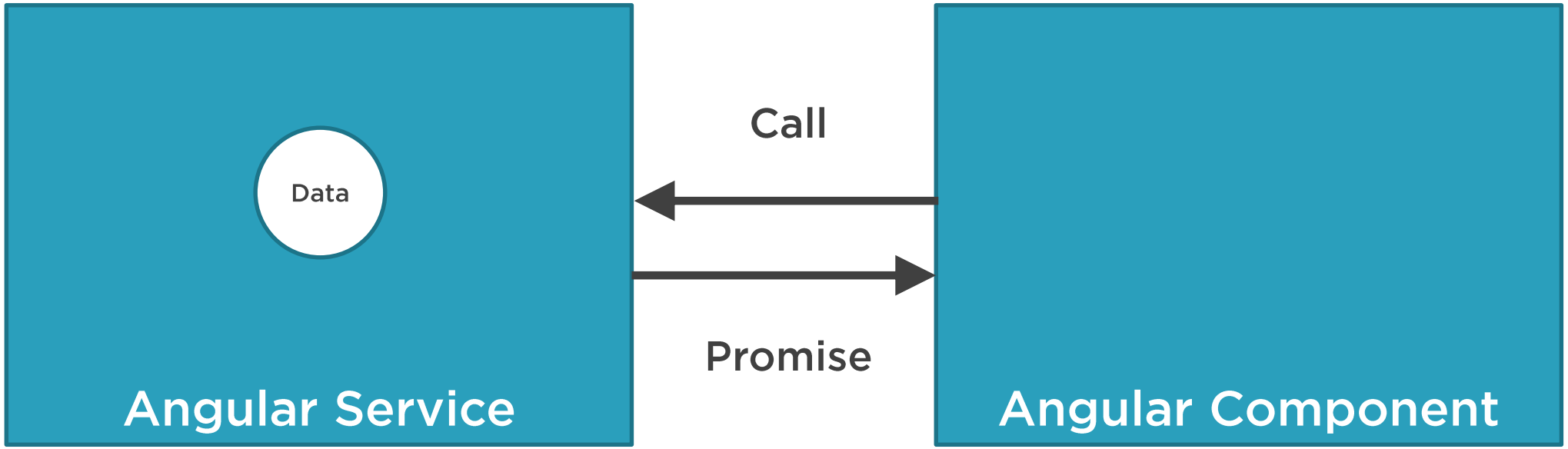
Promise

- An operation that hasn't completed yet, but is expected in the future
- Used with async/deferred operations
- Can be hooked to a callback

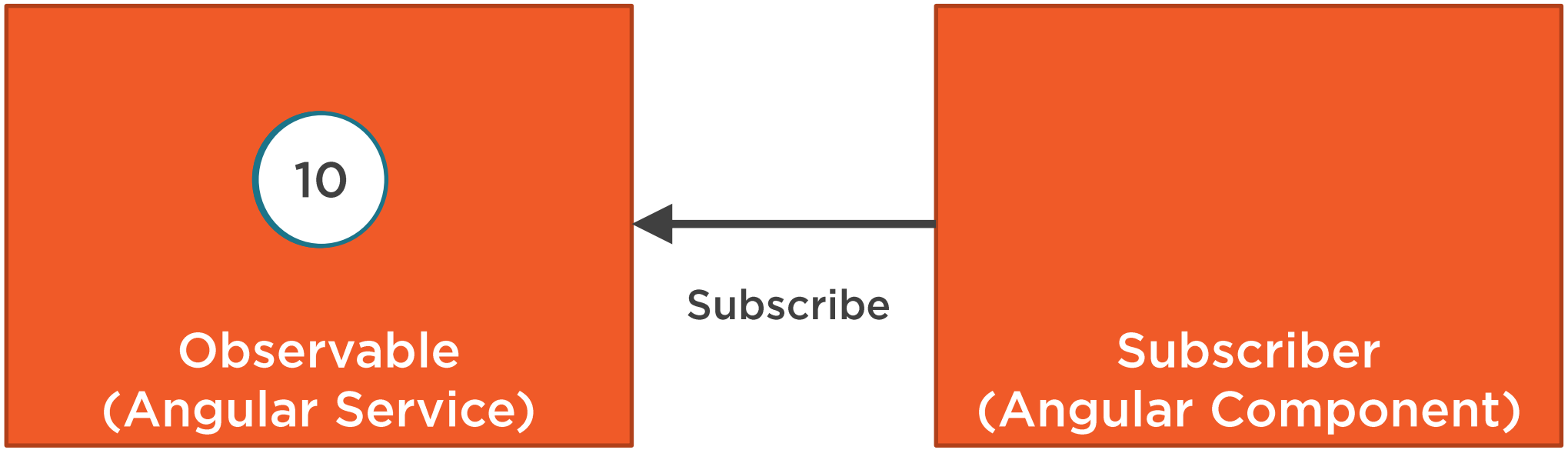
Observable

- An object that can be "subscribed" to by other objects
- Can return multiple values over time – an async data stream
- Event based

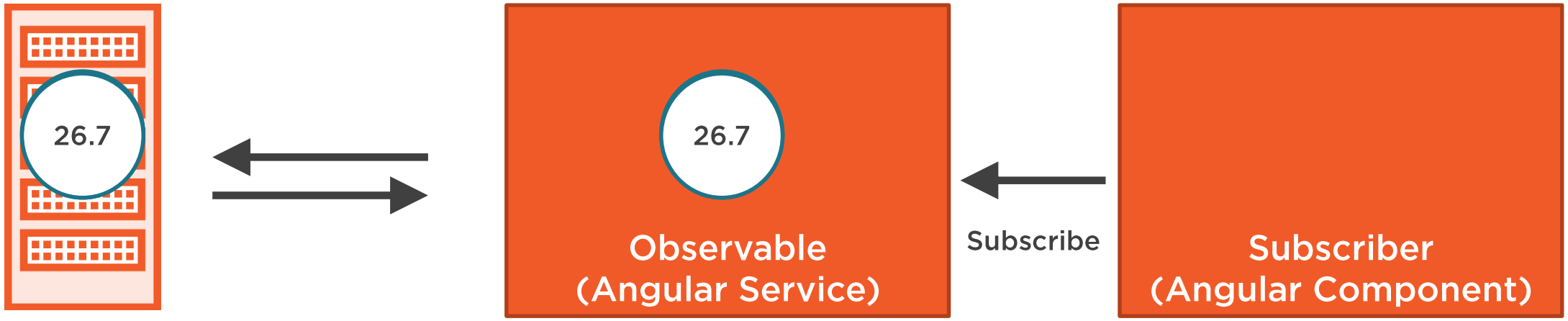
Promises Overview



Observables Overview



Observables and Async Streams



Promises and Observables Review

Promises	Observables
Returns a single value	Can return multiple values over time
Cannot cancel	Can cancel
Natively supported in browsers	Supports standard array functions (map, filter, reduce, etc.)
	Relies on a library such as RxJS

Running the Application on Windows

Software Requirements

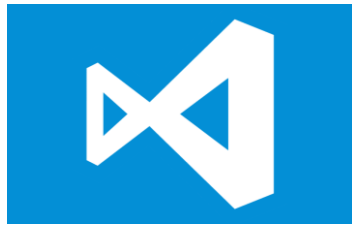
https://visualstudio.com

https://dot.net

Sqlite, PostgreSQL,
SQL Server

Running the Application on Mac

Software Requirements



https://code.visualstudio.com

ASP.NET Core

https://dot.net

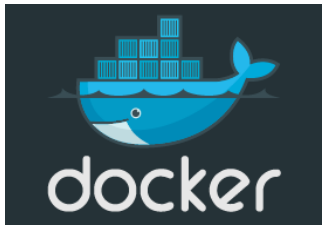
Database

Sqlite, PostgreSQL,
SQL Server

Running the Application with Docker

Software Installation

https://dot.net



https://docker.com

Summary

Key learning goals include understanding how to move data to and from a RESTful service

- Key technologies and concepts
- ASP.NET Core/Http/REST
- Angular/RxJS/Observables/Http