

# Integrating Angular with Node.js RESTful Services

---

## COURSE INTRODUCTION



**Dan Wahlin**

WAHLIN CONSULTING

@DanWahlin [www.codewithdan.com](http://www.codewithdan.com)



# Module Overview



**Pre-requisites 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**



# Pre-Requisites to Maximize Learning

---



# Course Pre-Requisites

**TypeScript**

**Fundamentals**

**Fundamentals**



# Learning Goals

---



# Learning Goals



## Server-Side Learning Goals



**Learn how to use Node.js and Express to create a RESTful service**

- Create convention-based routes
- Expose RESTful endpoints
- Integrate with a database

## 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
- Insert, update and delete data
- Page data



# Server-Side Technologies and Concepts

---



# Server-side Technologies and Concepts



Node.js



MongoDB



HTTP



REST



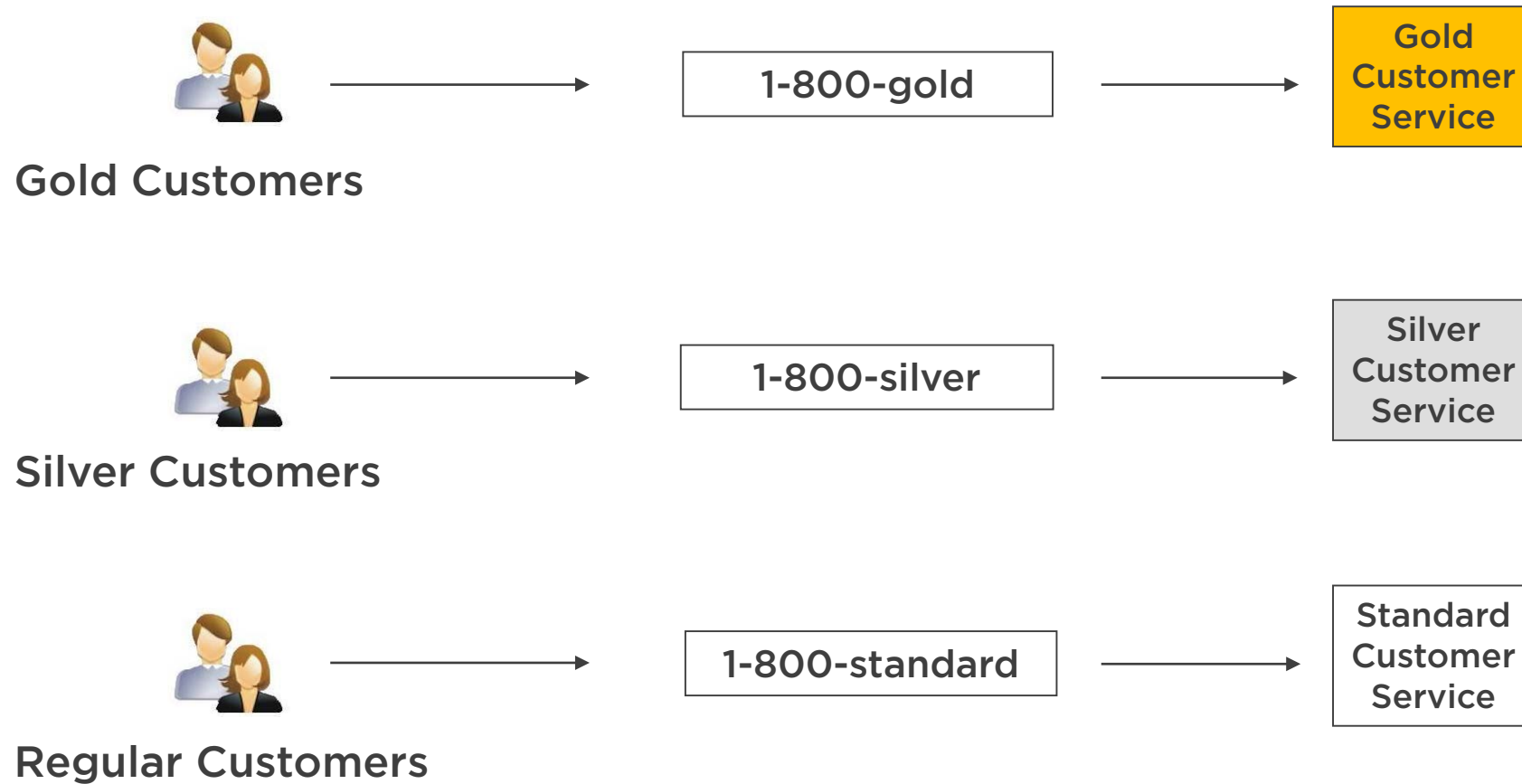
# Introduction to REST



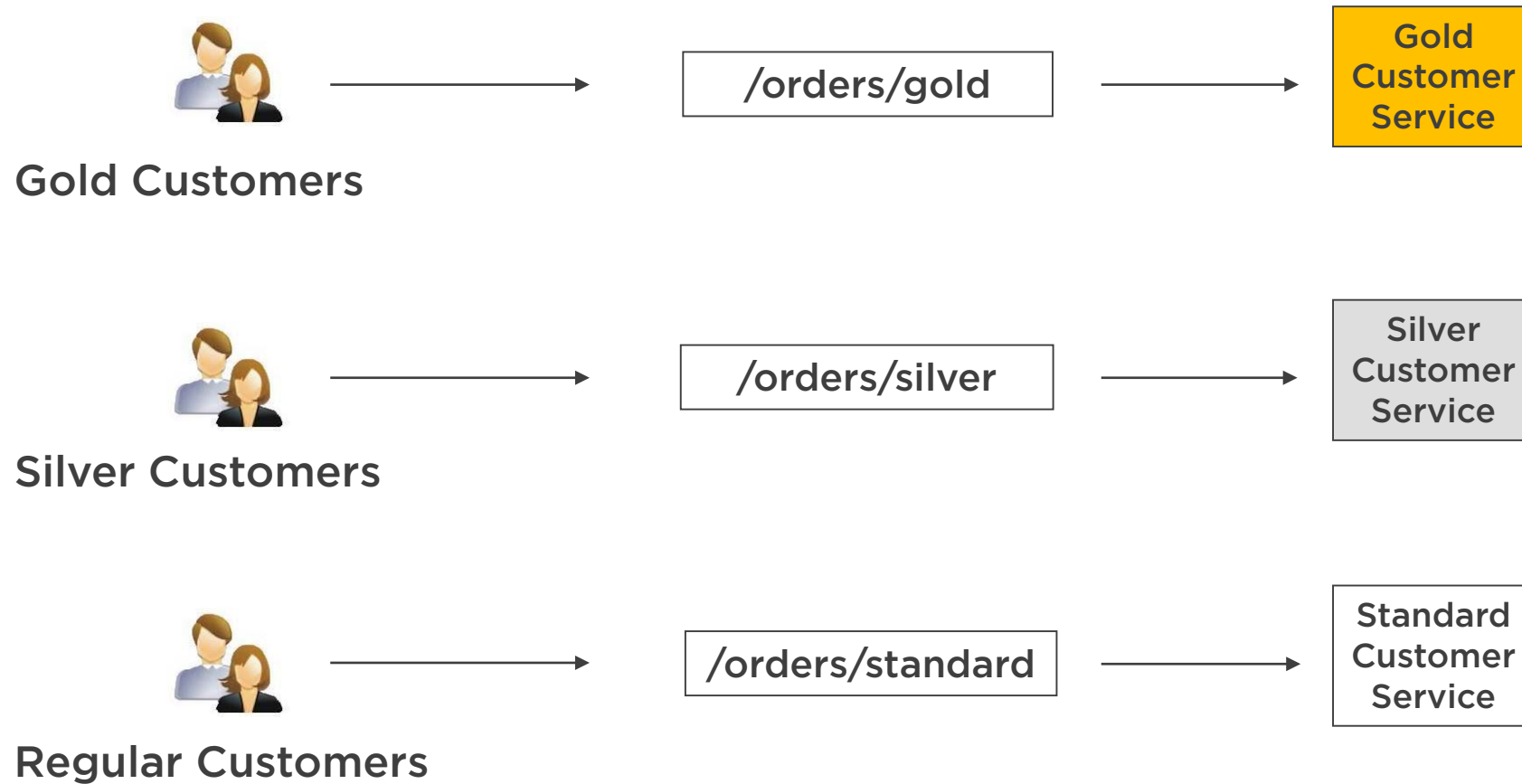
## REST = Representational State Transfer

- 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



```
{  
  "name": "Jim"  
}
```



# Key Technology Players



# Key Technology Players



```
{  
  "name": "Jim"  
}
```



# 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



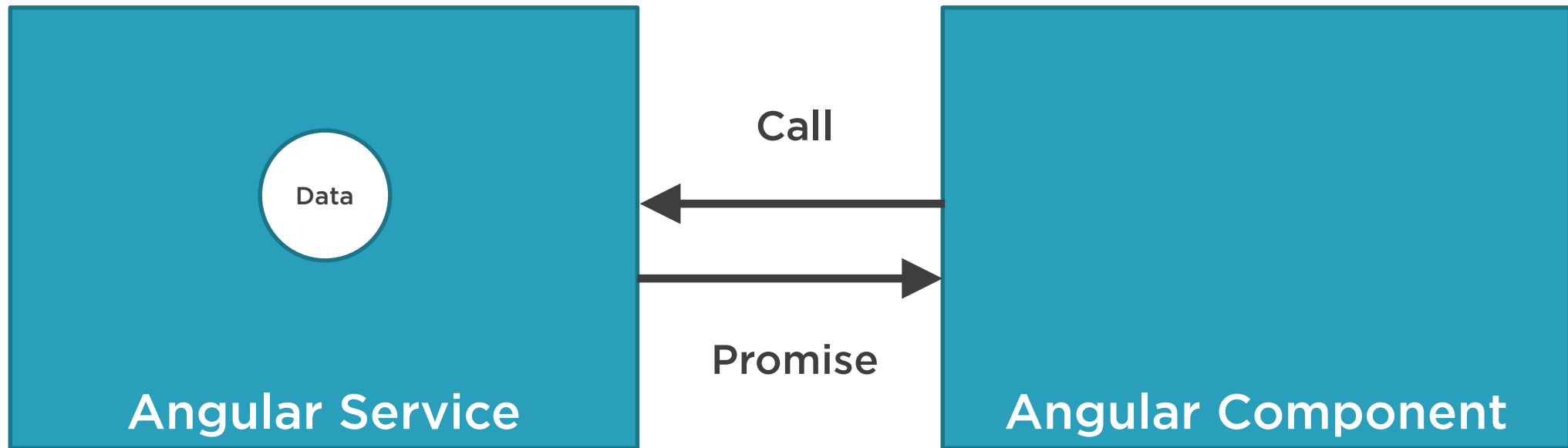
## Promise

- 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

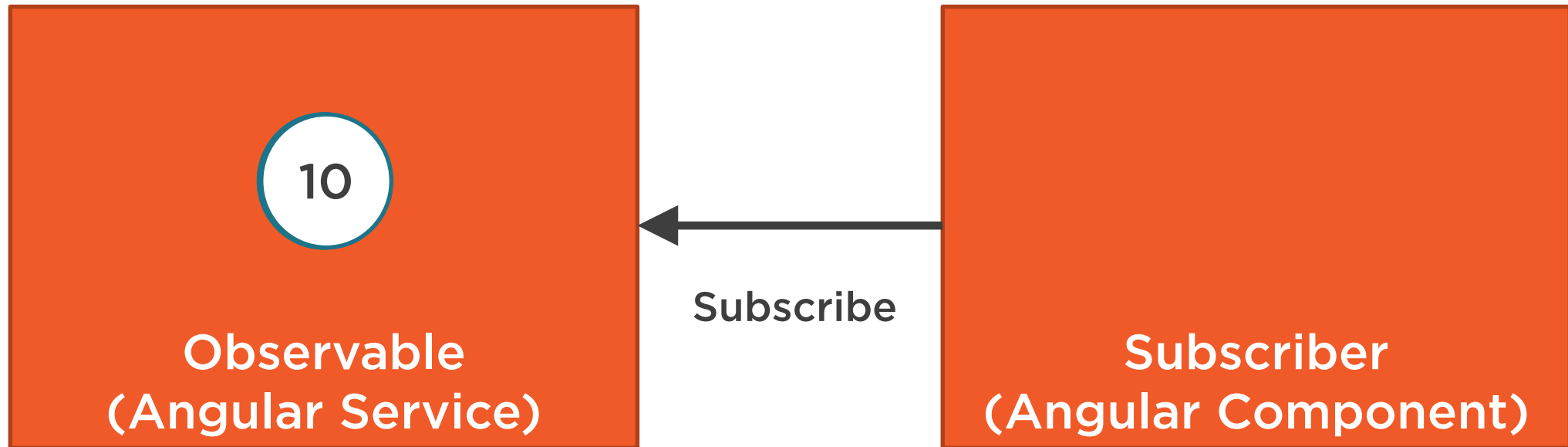
- An object that can be “subscribed” to by other objects
- Can return multiple values over time – an async data stream
- Event based

# Promises Overview

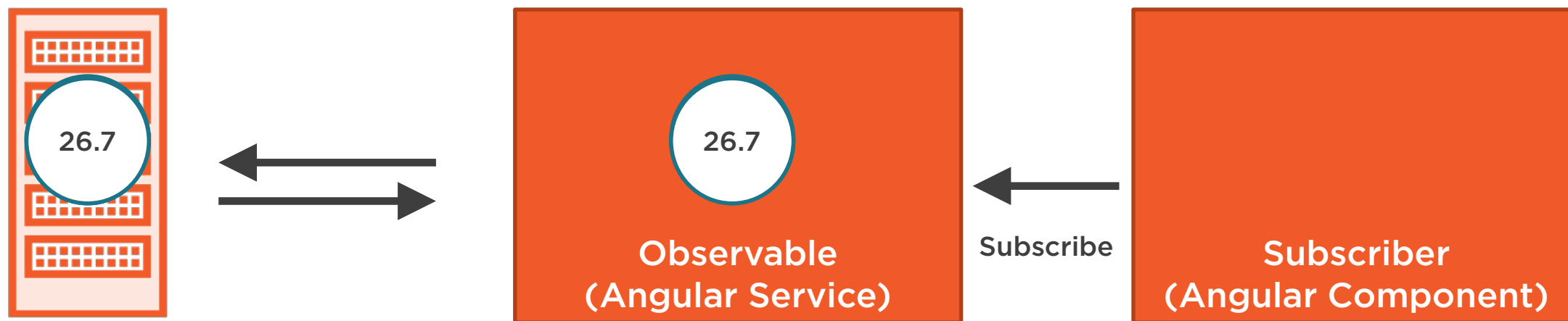




# Observables Overview



# Observables and Async Streams



# Promises and Observables Review

## Promises

Returns a single value

Cannot cancel

Natively supported in browsers

## Observables

Can return multiple values over time

Can cancel

Supports standard array functions  
(map, filter, reduce, etc.)

Relies on a library such as RxJS



# Running the Application

---



# Software Requirements



<https://code.visualstudio.com>



<https://nodejs.org>



[https://docs.mongodb.com  
/manual/installation](https://docs.mongodb.com/manual/installation)



# Running the Application with Docker

---



# Software Installation



<https://code.visualstudio.com>



<https://nodejs.org>



<https://docker.com>



# Summary



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

## **Key technologies and concepts**

- Node/MongoDB/Http/REST
- Angular/RxJS/Observables/Http

