

A vertical column of white circles connected by thin lines, resembling a circuit board or a tree structure.

TO HELL WITH CALLBACK HELLS

THE POWER OF OBSERVABLES





WHOAMI

Roy Straub

Senior Software Engineer @ CodeSquad



<https://github.com/rstraub>



www.linkedin.com/in/r-straub

AGENDA

- The problem
- Introducing the observable
- Using observables

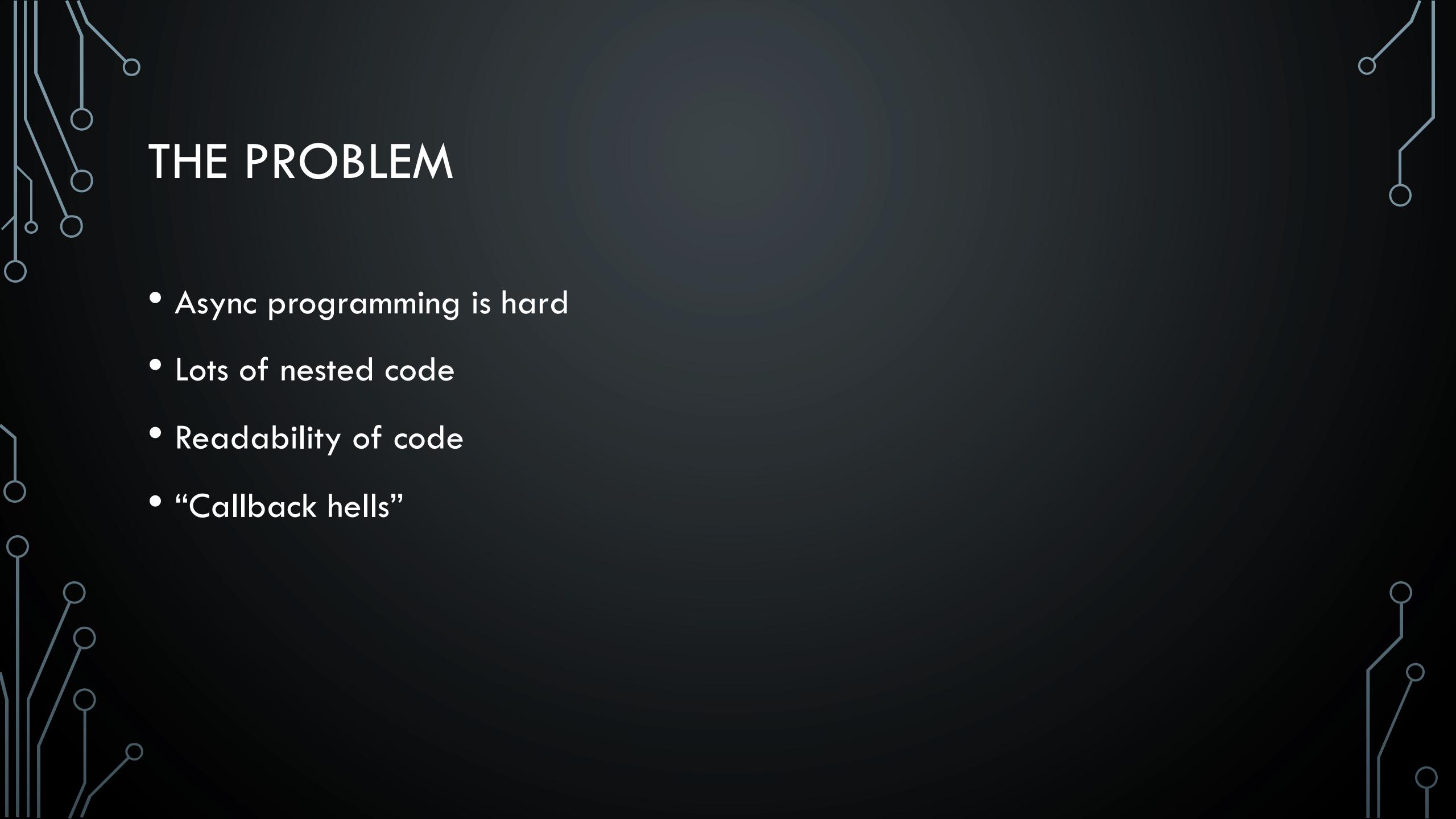


ReactiveX

<http://reactivex.io>



THE PROBLEM



THE PROBLEM

- Async programming is hard
- Lots of nested code
- Readability of code
- “Callback hells”

```
1 // Setup code
2
3 const getEarningsForMoviesWithActor = () => {
4   let sum = 0;
5
6   movieRepository.findMovies(movies => {
7     movies.forEach(movie => {
8       if (movie.year >= 1980) {
9         movieRepository.getMovieDetail(movie.id, movieDetail => {
10           if (movieDetail.actors.includes('Harrison Ford')) {
11             sum += movieDetail.earnings;
12           }
13         });
14       }
15     });
16   });
17 };
18
```

Callback hell



INTRODUCING THE OBSERVABLE

```
1 // Setup code
2
3 const getEarningsForMovies = () => {
4   let sum = 0;
5
6   movieRepository
7     .findMovies()
8     .filter(movie => movie.year >= 1980)
9     .flatMap(movie => movieRepository.getMovieDetail(movie.id))
10    .filter(movieDetail => movieDetail.actors.includes('Harrison Ford'))
11    .map(movieDetail => movieDetail.earnings)
12    .subscribe(earnings => {
13      sum += earnings;
14    });
15  };
16
```

Refactored

WHAT IS AN OBSERVABLE?

Synchronous

Asynchronous

Single

Variable

Promise

Multiple

Array

Observable

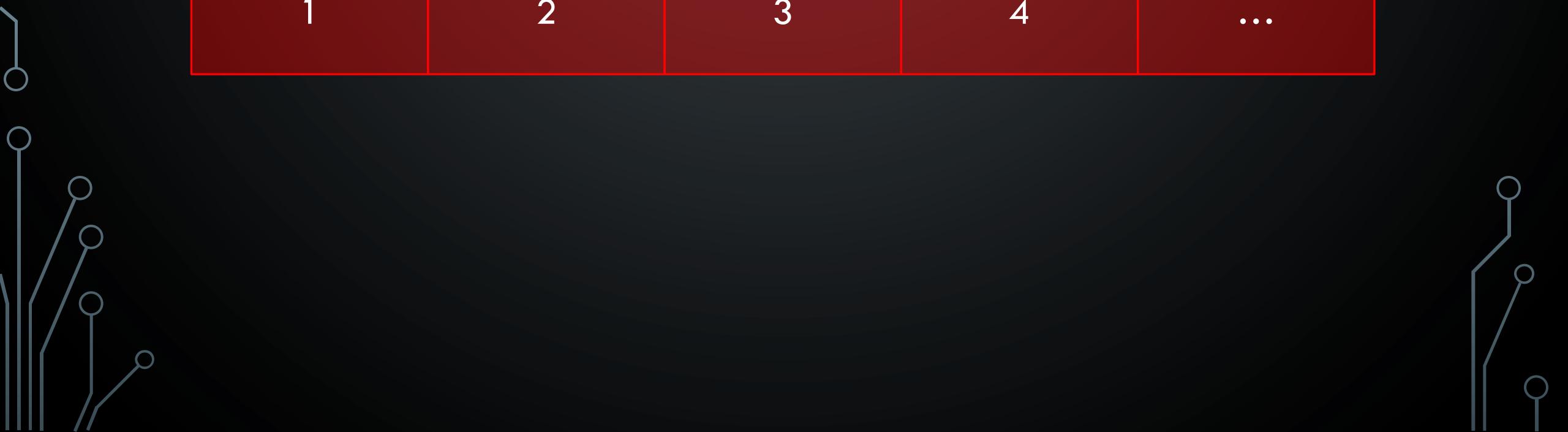


WHAT IS AN OBSERVABLE?

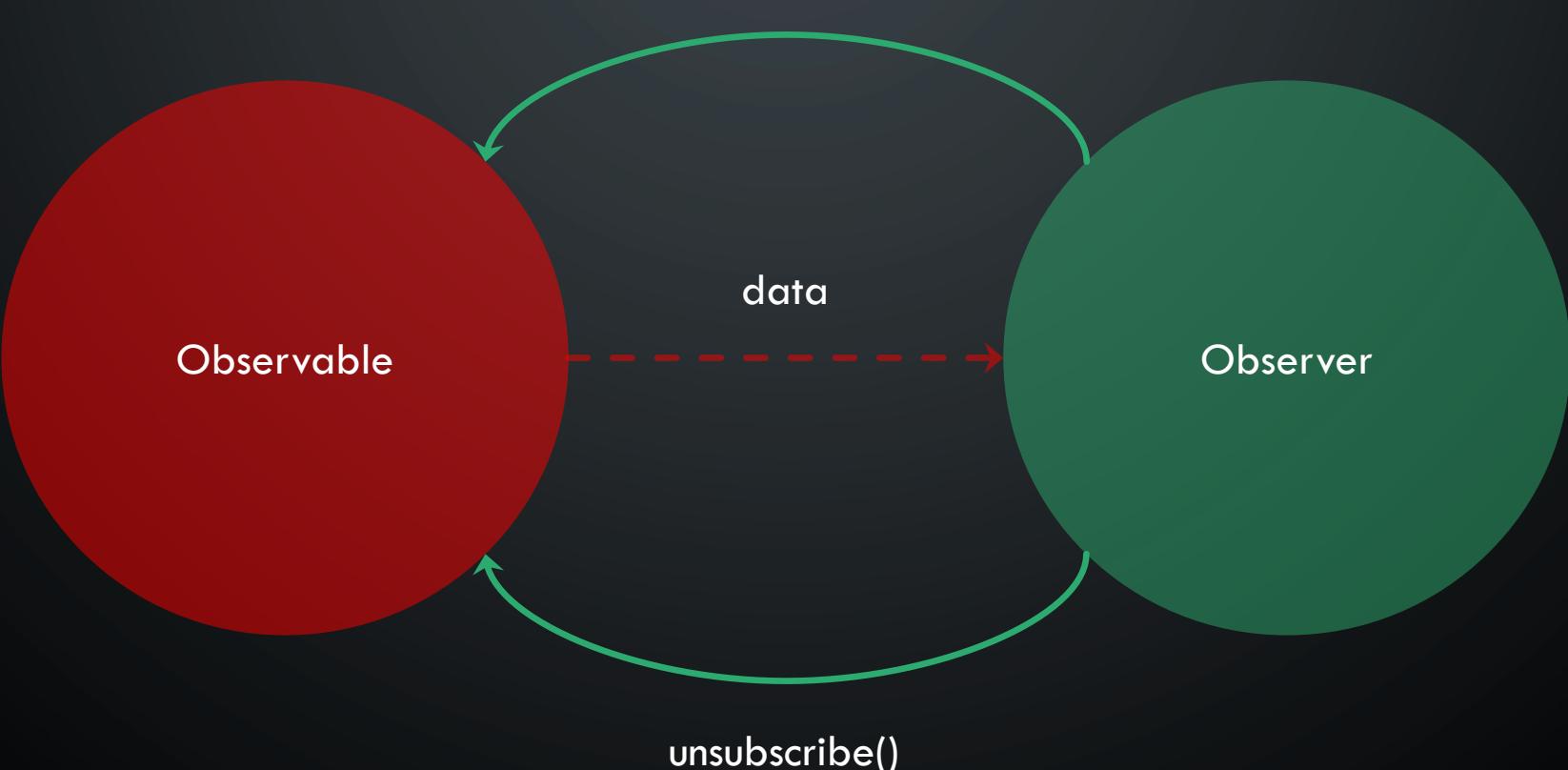
Think: Collection of data delivered over time



1 2 3 4 ...

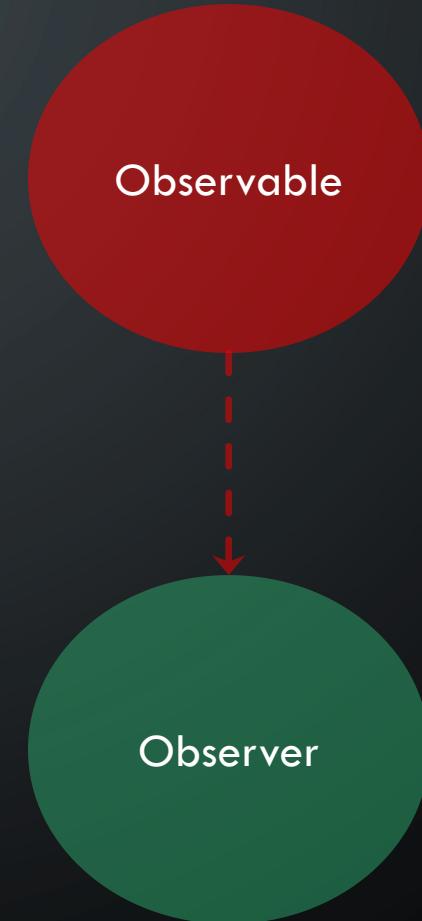


OBSERVABLE & OBSERVER



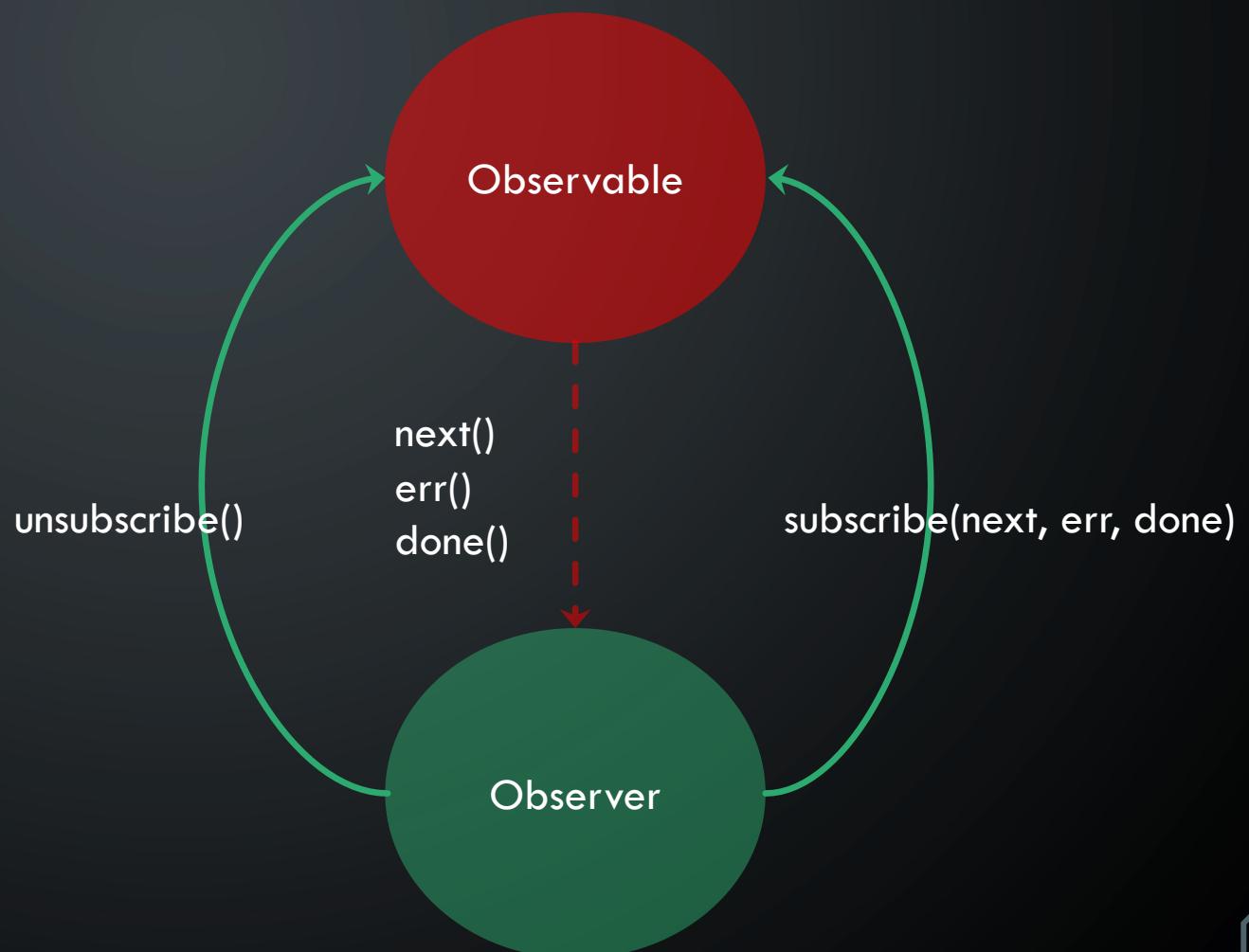
OBSERVABLE

- Source of **async** data
- Notifies observer
- Vars conventionally ends with \$
- Hot and Cold



OBSERVER

- Handles data from observable
- Subscribes to observable
- Active until
 - unsubscribe
 - completed
 - error



THE BENEFITS

Reduce code

Increases readability

Extensive api



USING OBSERVABLES

GETTING AN OBSERVABLE

Create your own

Use an operator

Libraries/Frameworks

DEMO – CREATING AN OBSERVABLE



+



OPERATORS

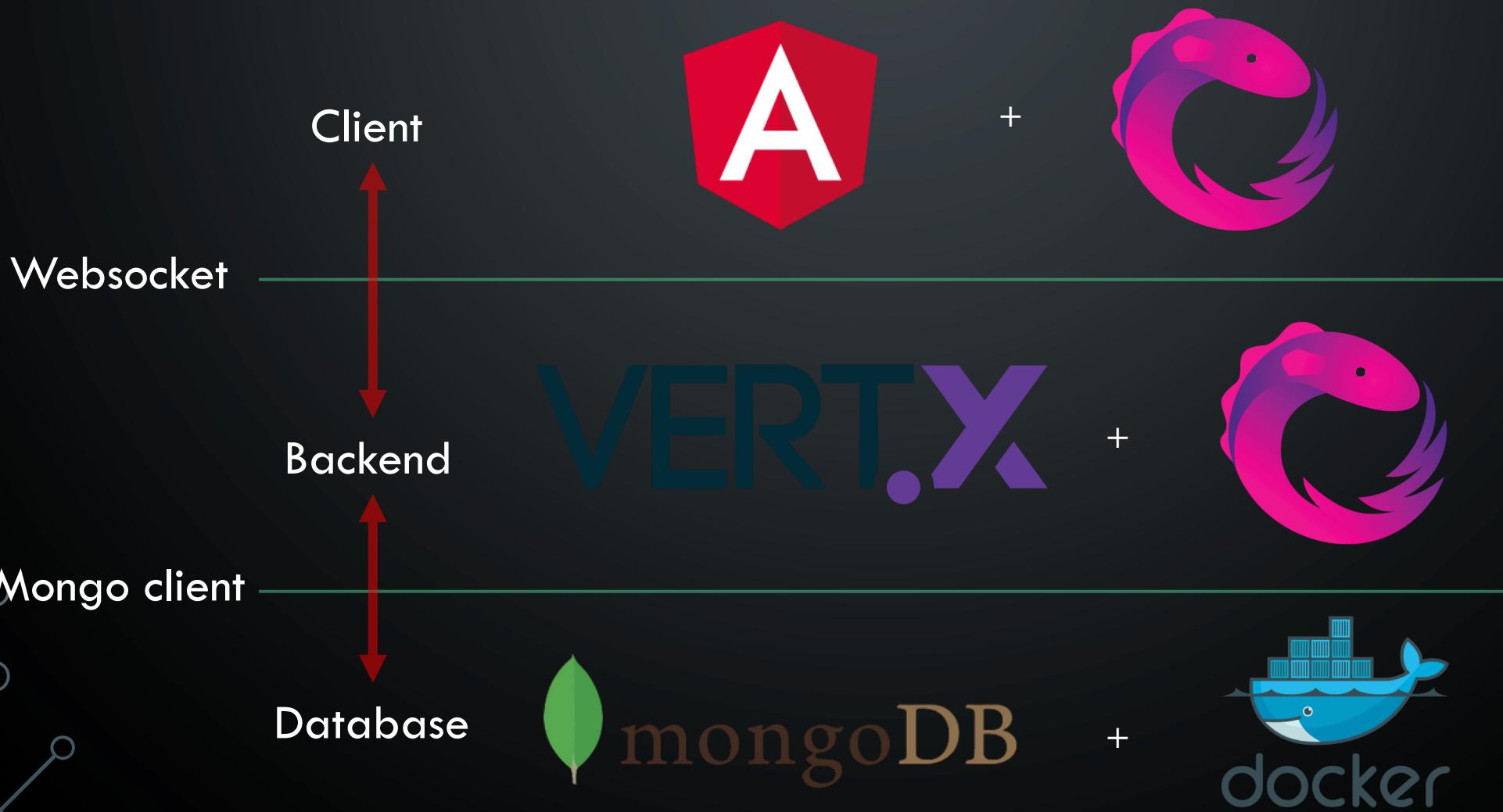
- Create Observables
- Transform data
- Filter data
- Combine observables
- Retries
- ...



`.map(value => value * 2)`
`.subscribe(...)`



DEMO – ASYNC MOVIE DB



SUMMARY

- An observable produces data
- The observer consumes the data
- Operators provide utility
- Observables simplify async programming

