TYRANNOSAURUS

REX

REACTIVE EXTENSIONS

C#

JS

Java

Python

Ruby

C++

Haskell

Clojure

Groovy

Scala

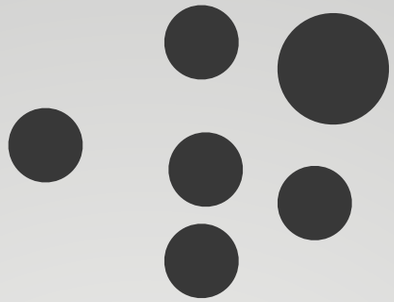
Kotlin

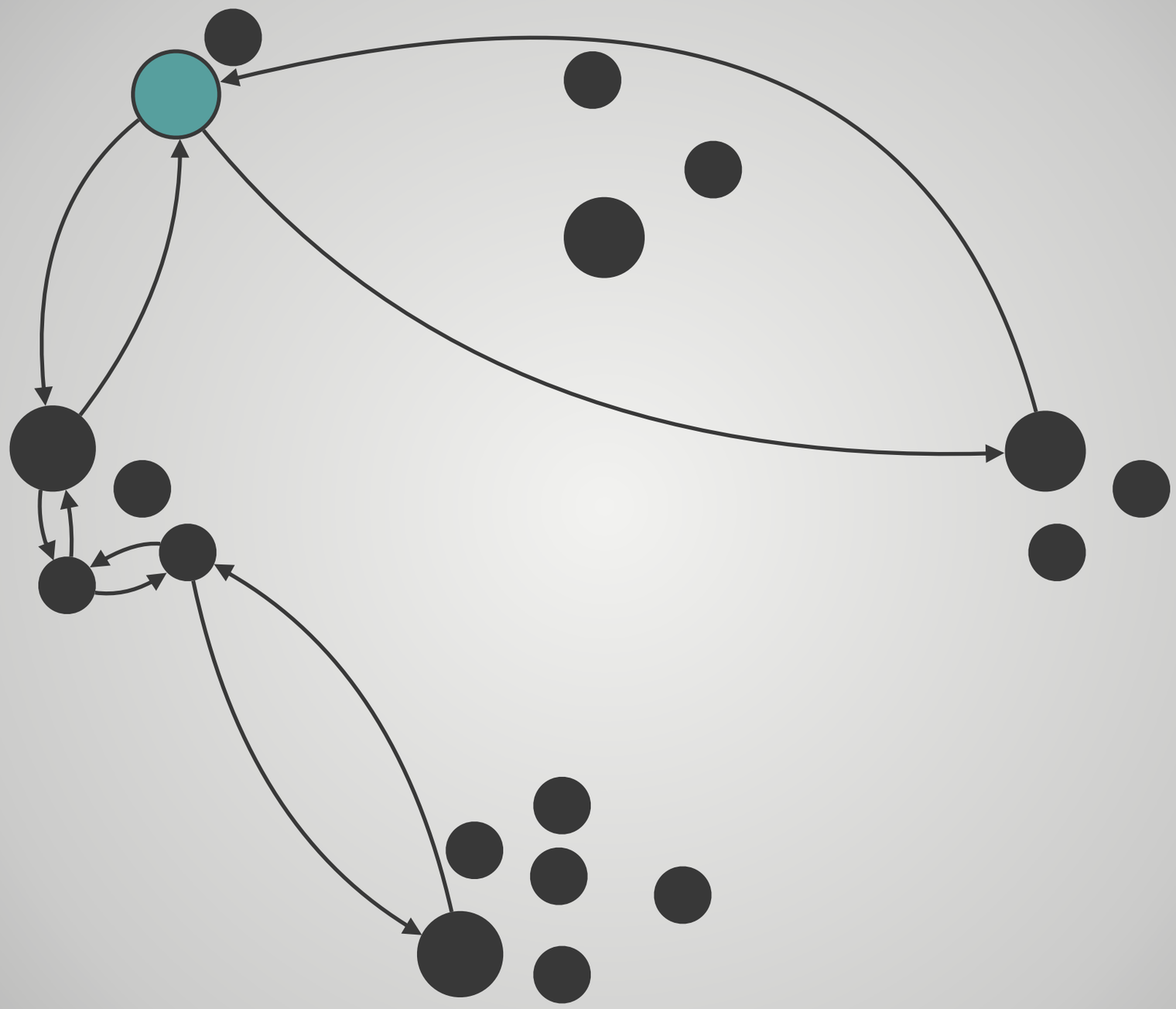
```
var list = [1,2,3,4,5,6,7];
for(var i = 0; i < list.length; i++) {
    console.log(list[i])
}
```

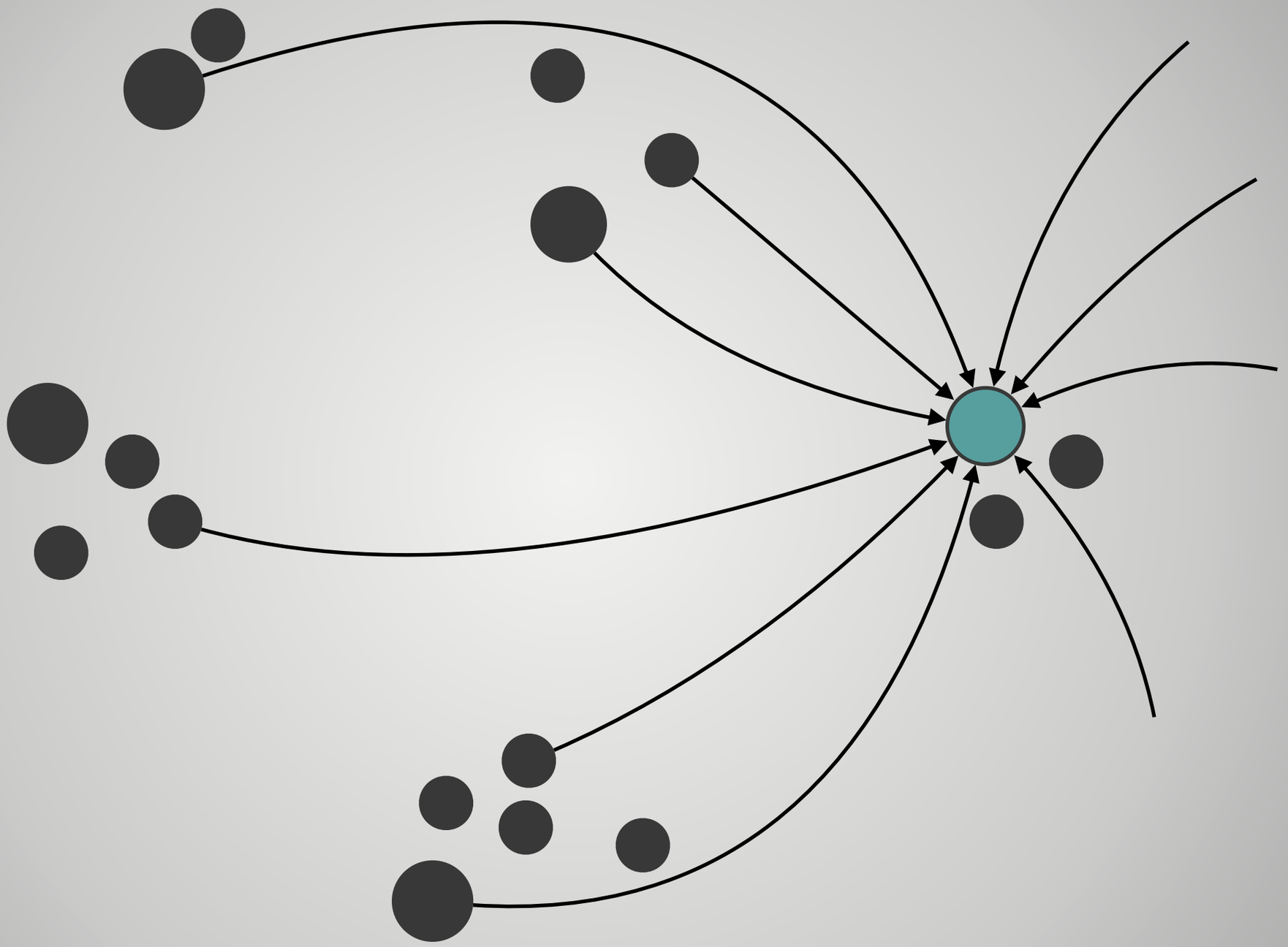
```
var list = [1,2,3,4,5,6,7];

list.forEach(function(item) {
  console.log(item)
})
```

```
var list = [Promise(), Promise()…];

list.forEach(function(item) {
  console.log(item.get())
})
```

TIME RUINS EVERYTHING







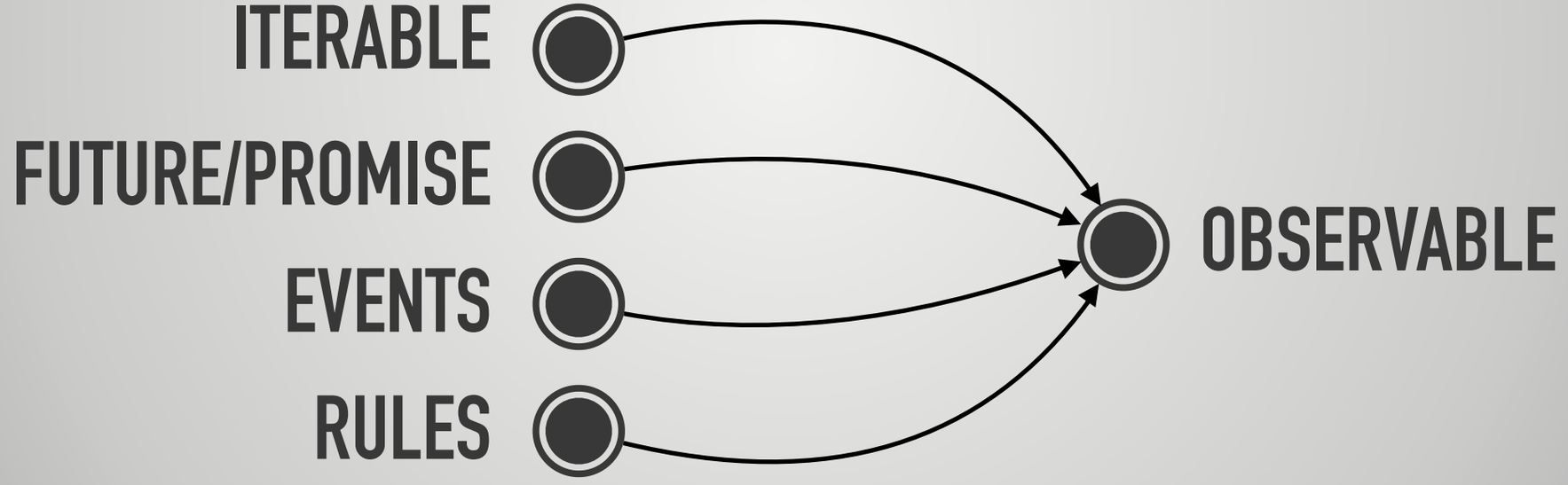
OBSERVABLES

LISTS WITH TIME

REIFIED EVENT STREAM

PUSH BASED

CREATING OBSERVABLES



CREATING OBSERVABLES

```
Observable.from(new String[] {
"The Joker",
"The Riddler",
"Penguin",
"Catwoman"})
```

CREATING OBSERVABLES

```
baddies.subscribe((baddie) -> {
    out.println(baddie + " is bad.")
})
```

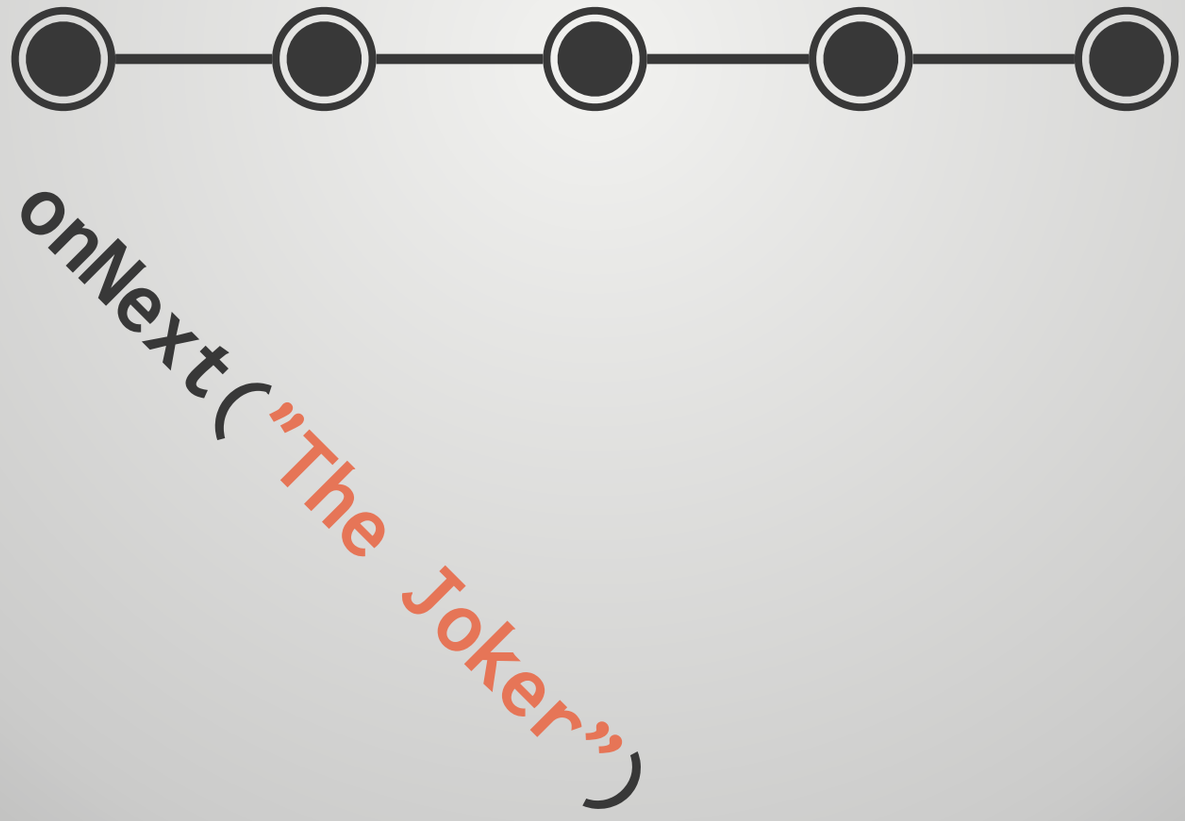
CREATING OBSERVABLES

```
The Joker is bad.
The Riddler is bad.
Penguin is bad.
Catwoman is bad.
```

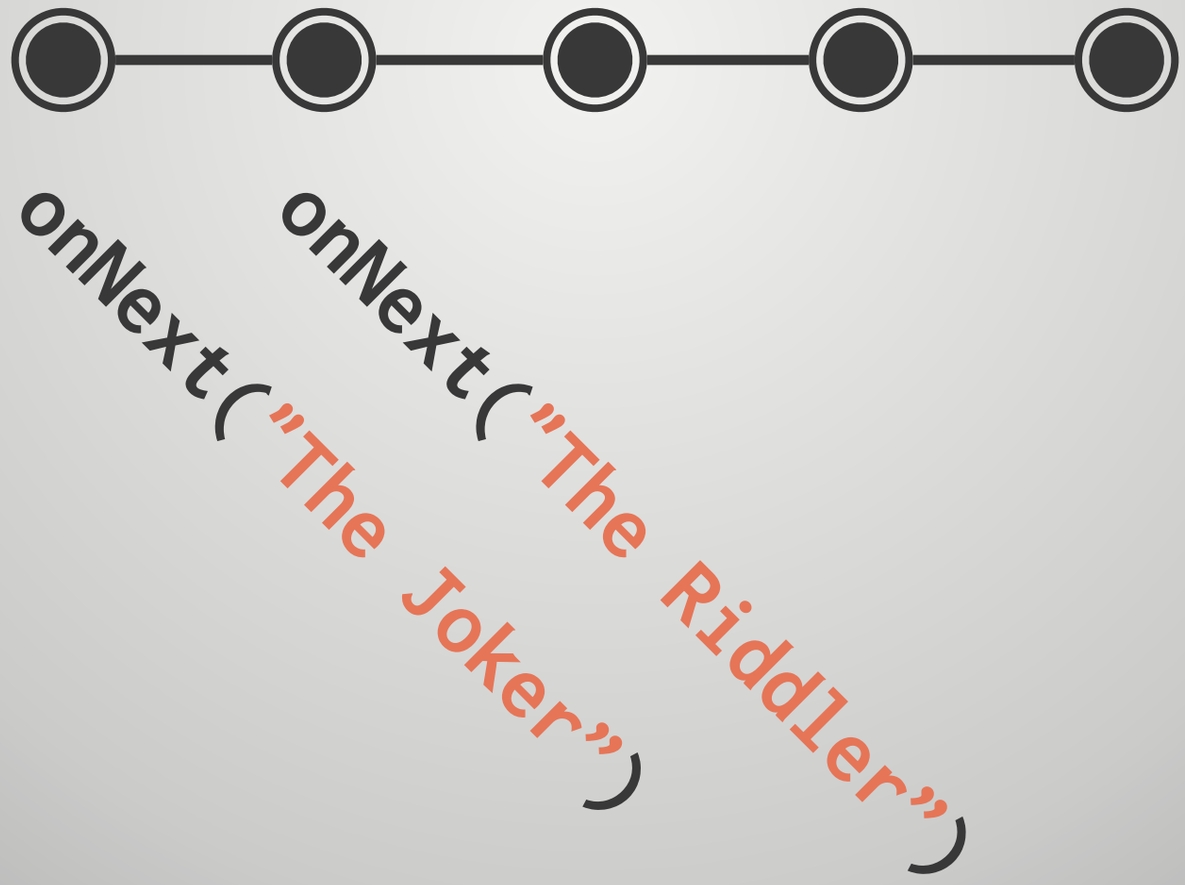
CREATING OBSERVABLES

```
class _ extends Subscriber<String> {
  void onCompleted() {}
  void onError(Throwable t) {}
  void onNext(String s){}
}
```

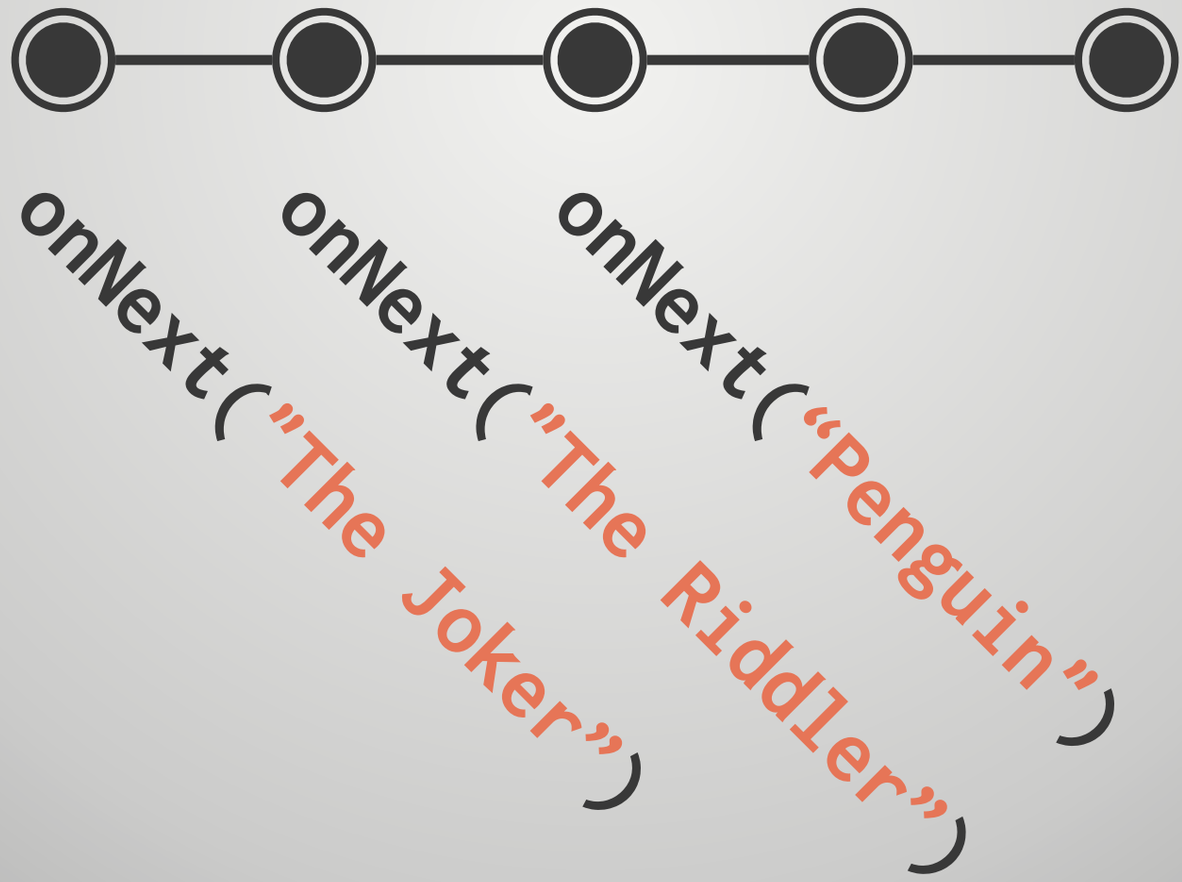
CREATING OBSERVABLES



CREATING OBSERVABLES



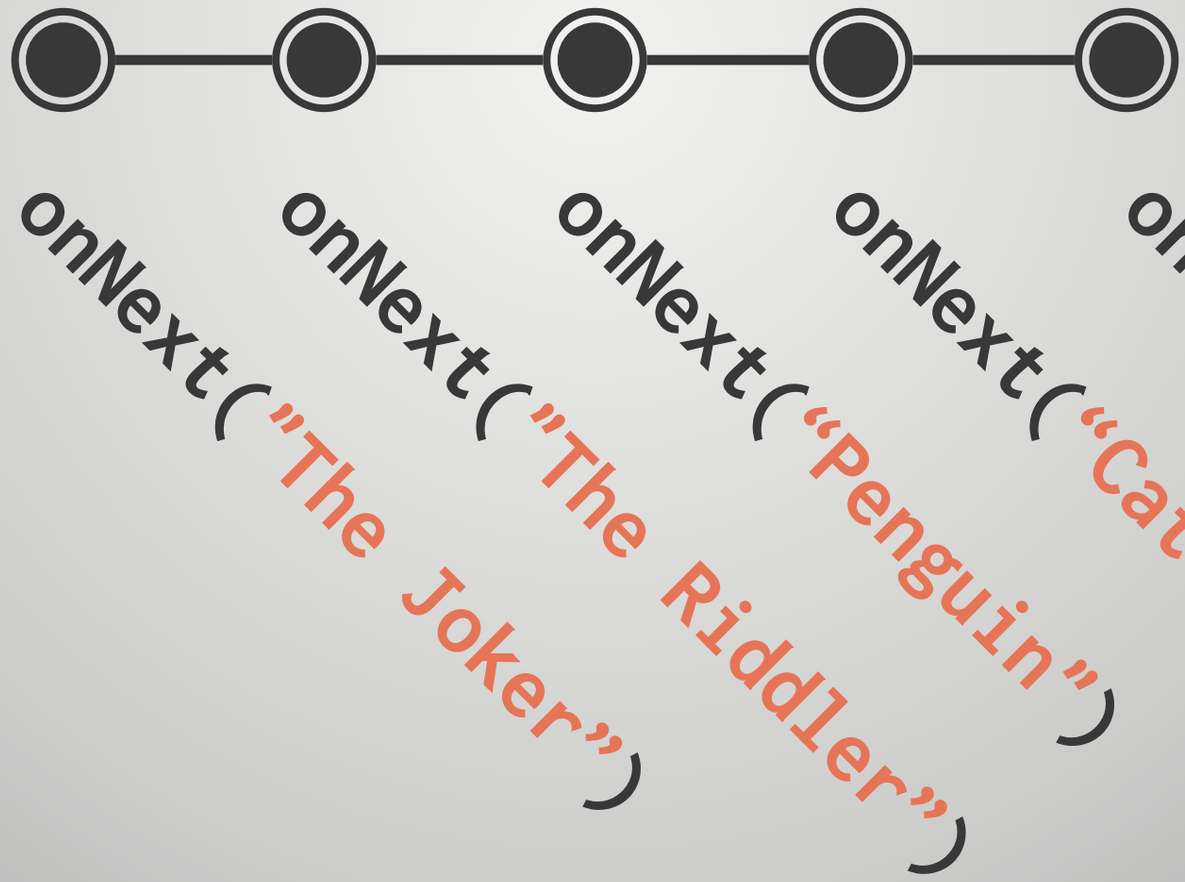
CREATING OBSERVABLES



CREATING OBSERVABLES

```
onNext("Catwoman")
onNext("Penguin")
onNext("The Riddler")
onNext("The Joker")
```

CREATING OBSERVABLES



ERROR HANDLING

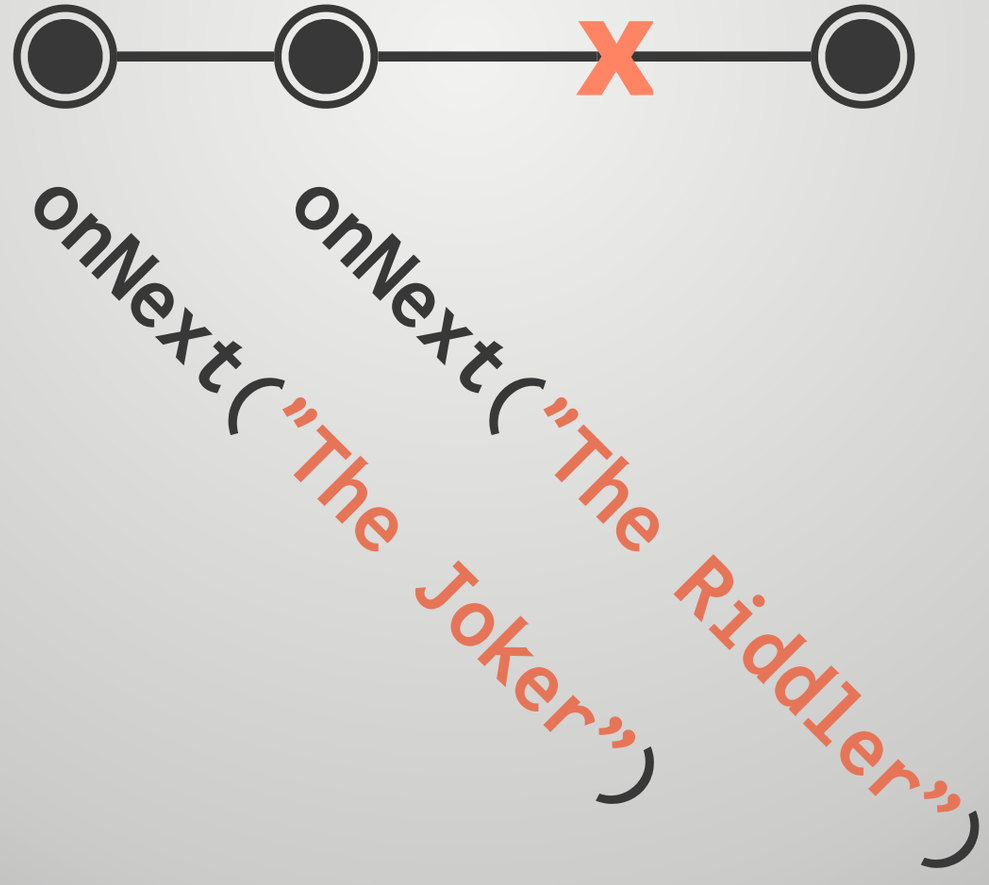
onNext("The Joker")

ERROR HANDLING

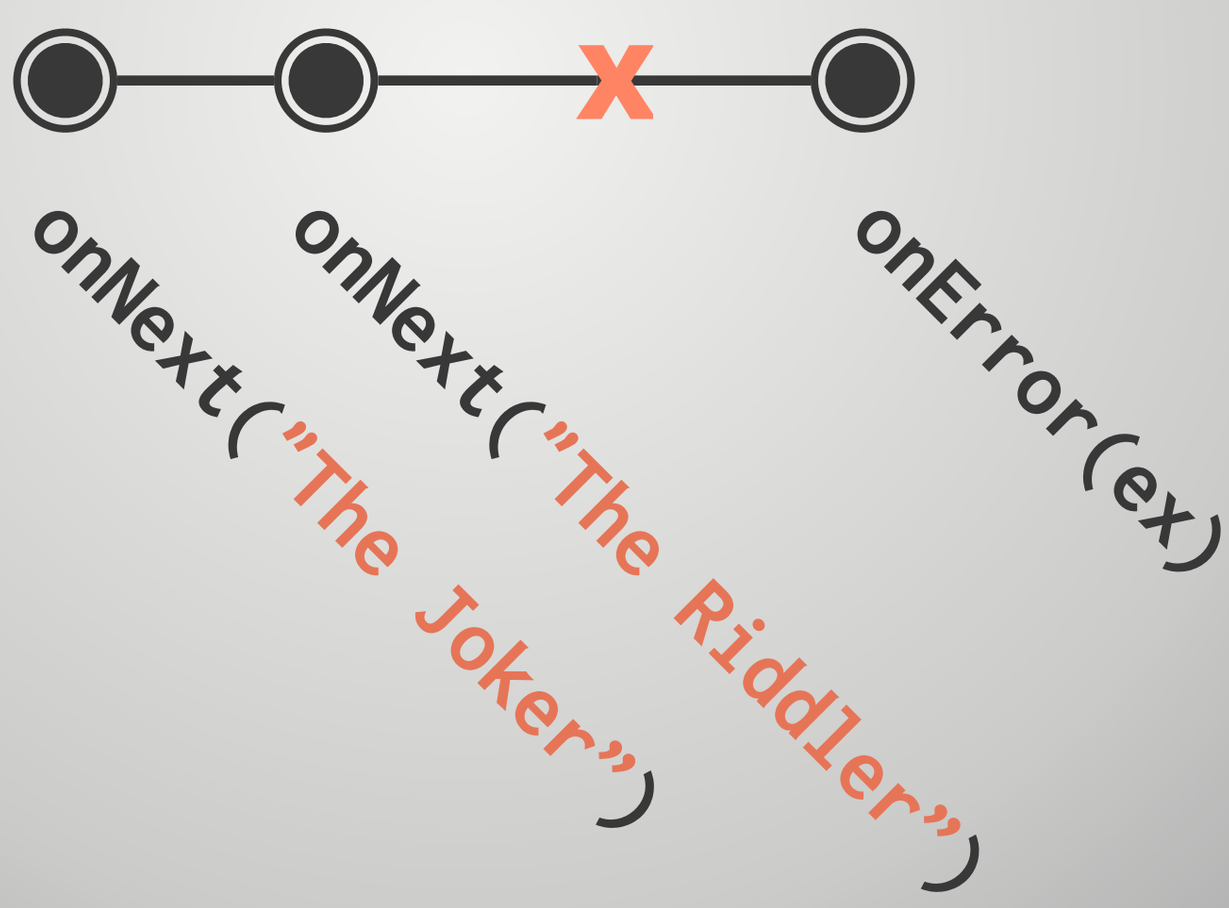
onNext("The Riddler")

onNext("The Joker")

ERROR HANDLING



ERROR HANDLING



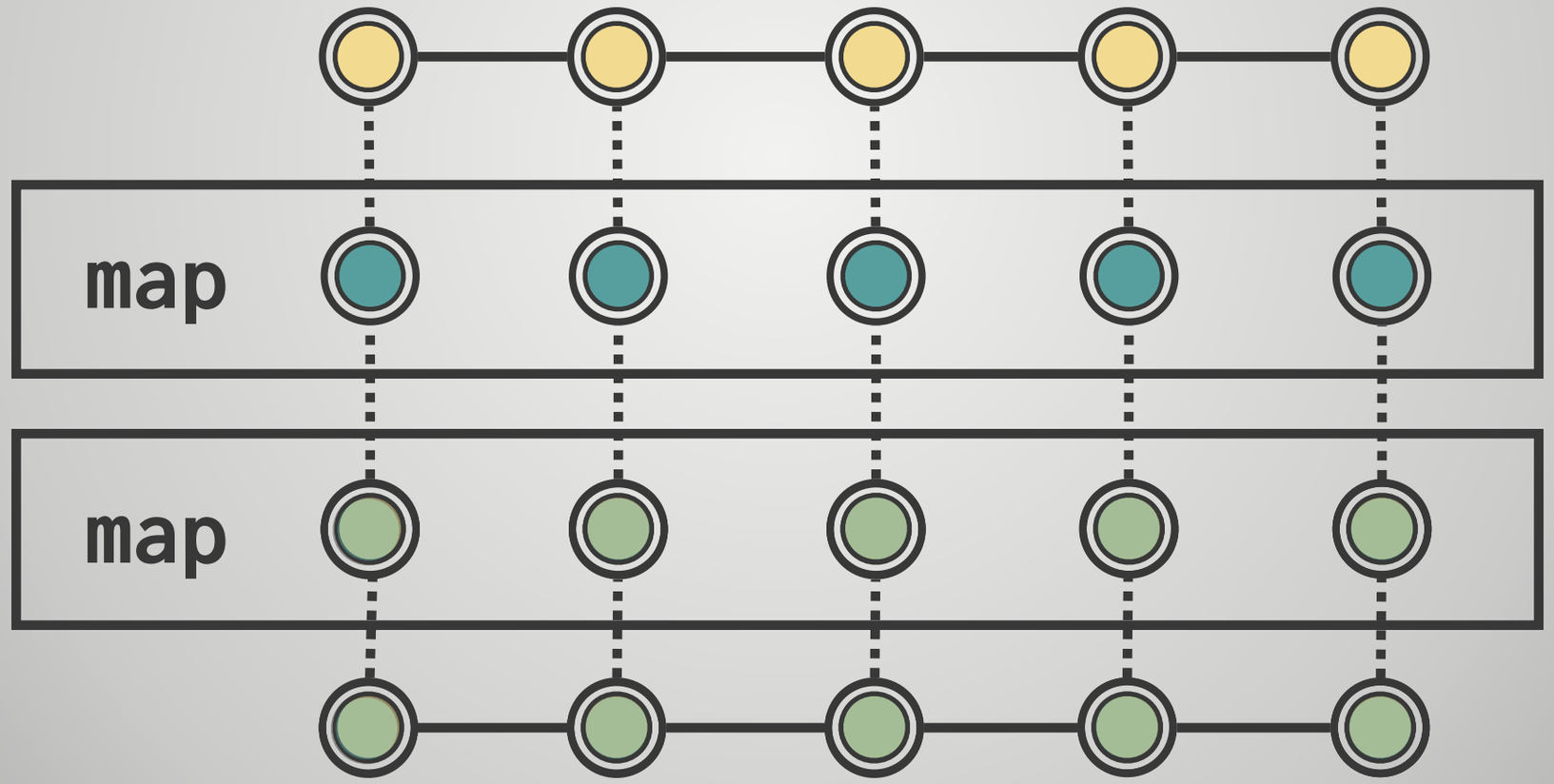
TRANSFORMING OBSERVABLES

TURN AN OBSERVABLE OF ONE TYPE INTO ANOTHER

TRANSFORMING OBSERVABLES

```
Observable ints = Observable
        .interval(1, SECONDS);

ints.map((i) -> i + 1)
        .map((i) -> i + " seconds")
        .subscribe(out::println)
```

TRANSFORMING OBSERVABLES



TRANSFORMING OBSERVABLES

```
1  seconds
2  seconds
3  seconds
4  seconds
...
```

TRANSFORMING OBSERVABLES

`flatMap()`, `groupBy()`,
`buffer()`, `window()`

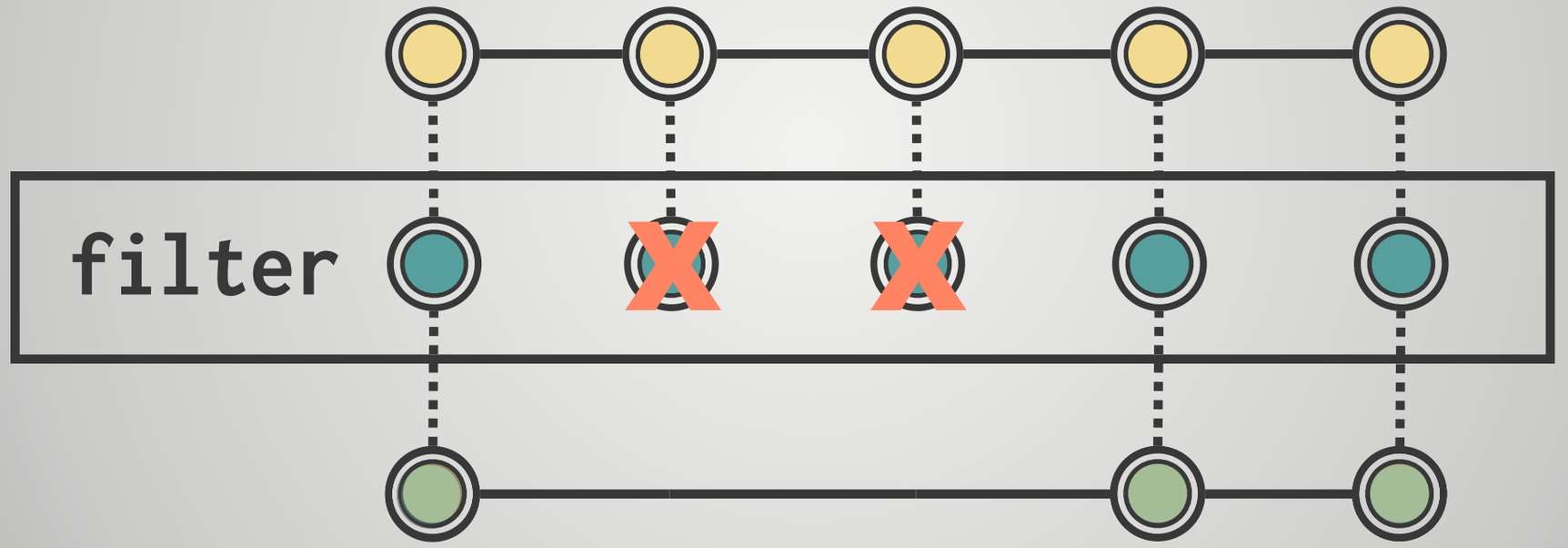
FILTERING OBSERVABLES

SELECT AND REJECT ITEMS
EMITTED BY AN OBSERVABLE

FILTERING OBSERVABLES

```
var keys = Rx.DOM.keypress(el)
var ints = keys.filter(isNumeric)

ints.subscribe(console.log)
```

FILTERING OBSERVABLES



FILTERING OBSERVABLES

`distinct()`, `timeout()`,
`ignoreElements()`, `throttleFirst()`,
`debounce()`, `sample()`, `first()`,
`last()`, `take()`, `skip()`

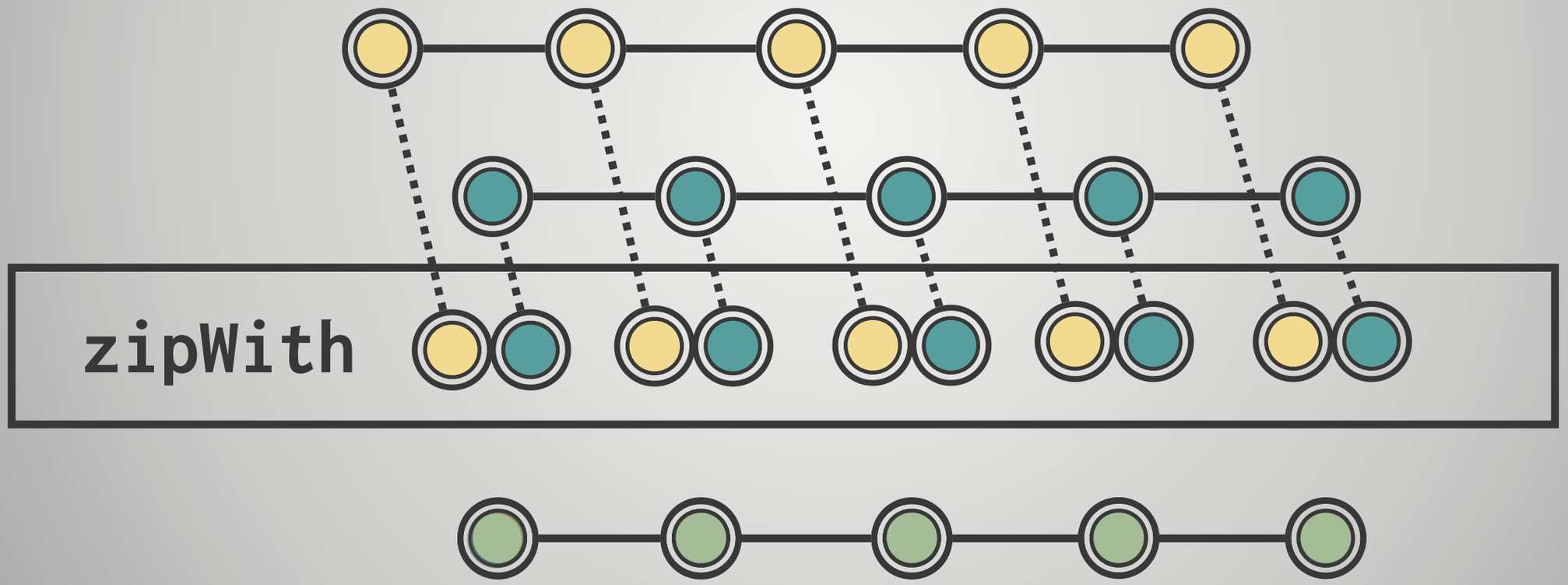
COMBINING OBSERVABLES

COMPOSING MULTIPLE OBSERVABLES INTO A SINGLE OBSERVABLE

COMBINING OBSERVABLES

```
Observable goodies = Observable.from(
        new String[]{
            "Batman", "Robin",
            "Alfred", "Batgirl"});

goodies.zipWith(baddies,
    (g,b) -> g + " punches " + b)
```

COMBINING OBSERVABLES



COMBINING OBSERVABLES

```
Batman punches The Joker
Robin punches The Riddler
Alfred punches Penguin
Batgirl punches Catwoman
```

STUFF AND OTHER THINGS

`retry()`, `delay()`, `timestamp()`,
`amb()`, `defaultIfEmpty()`,
`reduce()`, `count()`

TYRANNOSAURUS

REX