



Rxjs Subjects



@coder_aishya



What is a Subject?

- A Subject is a **special type of Observable** from Rxjs Library that allows values to be **multicasted(one to many)** to many Observers.
- The subjects are also observers because they can subscribe to another observable and get value from it, which it will multicast to all of its subscribers.
- Basically, **a subject can act as both observable & an observer.**

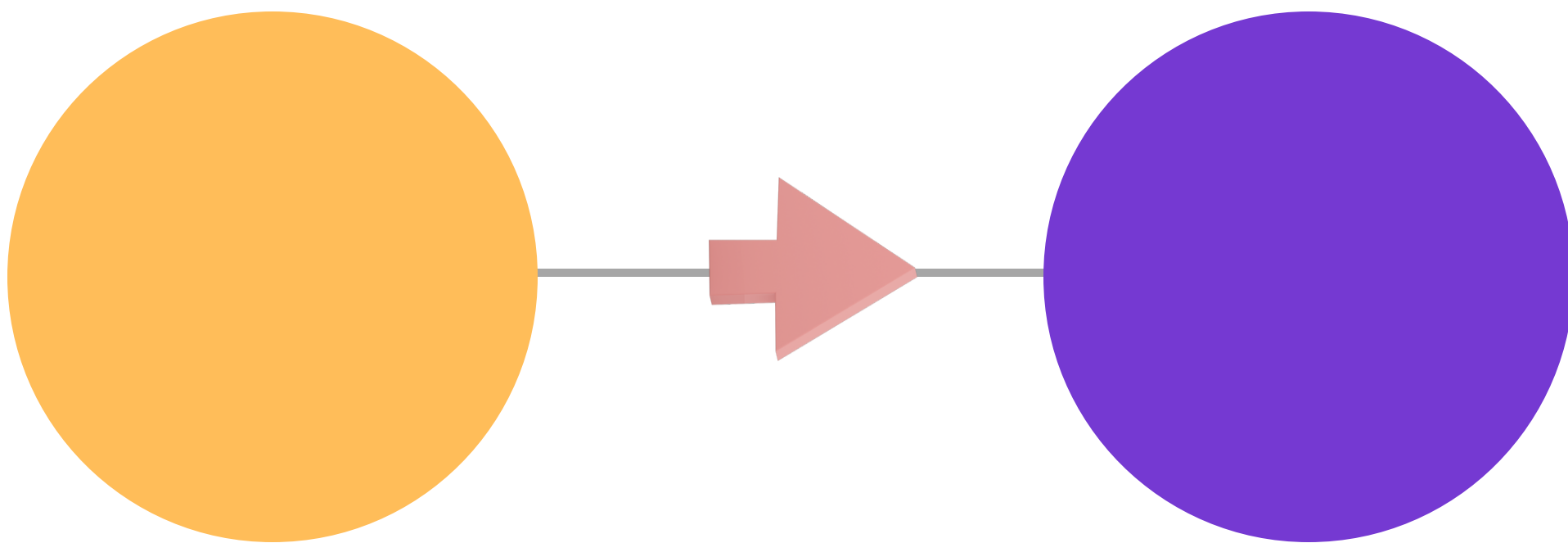
Let's understand the **difference between unicast and multicast first**



@coder_aishya



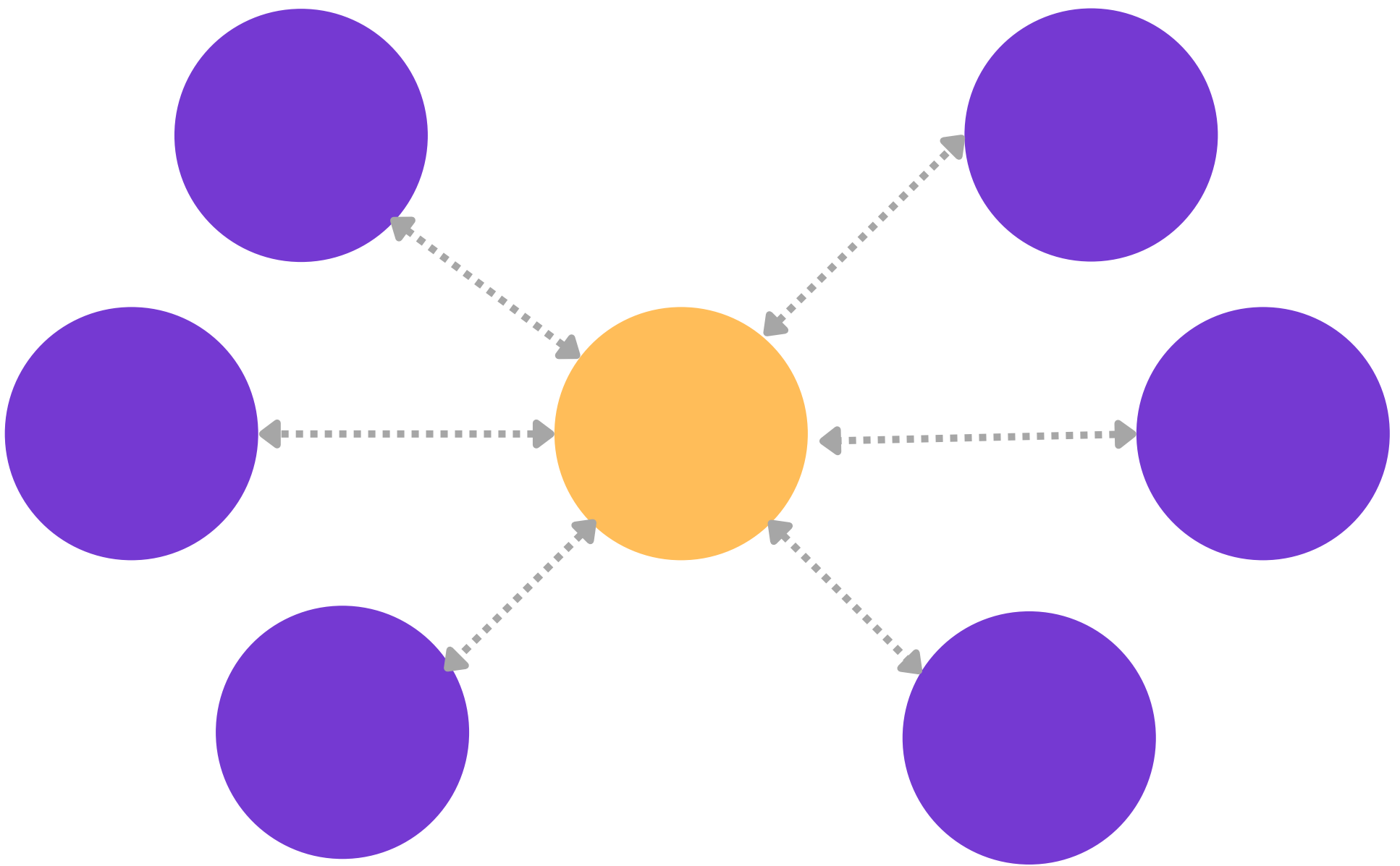
UNICAST



- An **Observable** is **unicast**.
- An Observer and its Subscriber have a **one-to-one relationship**.
- Each subscribed Observer owns an **independent execution** of the Observable.



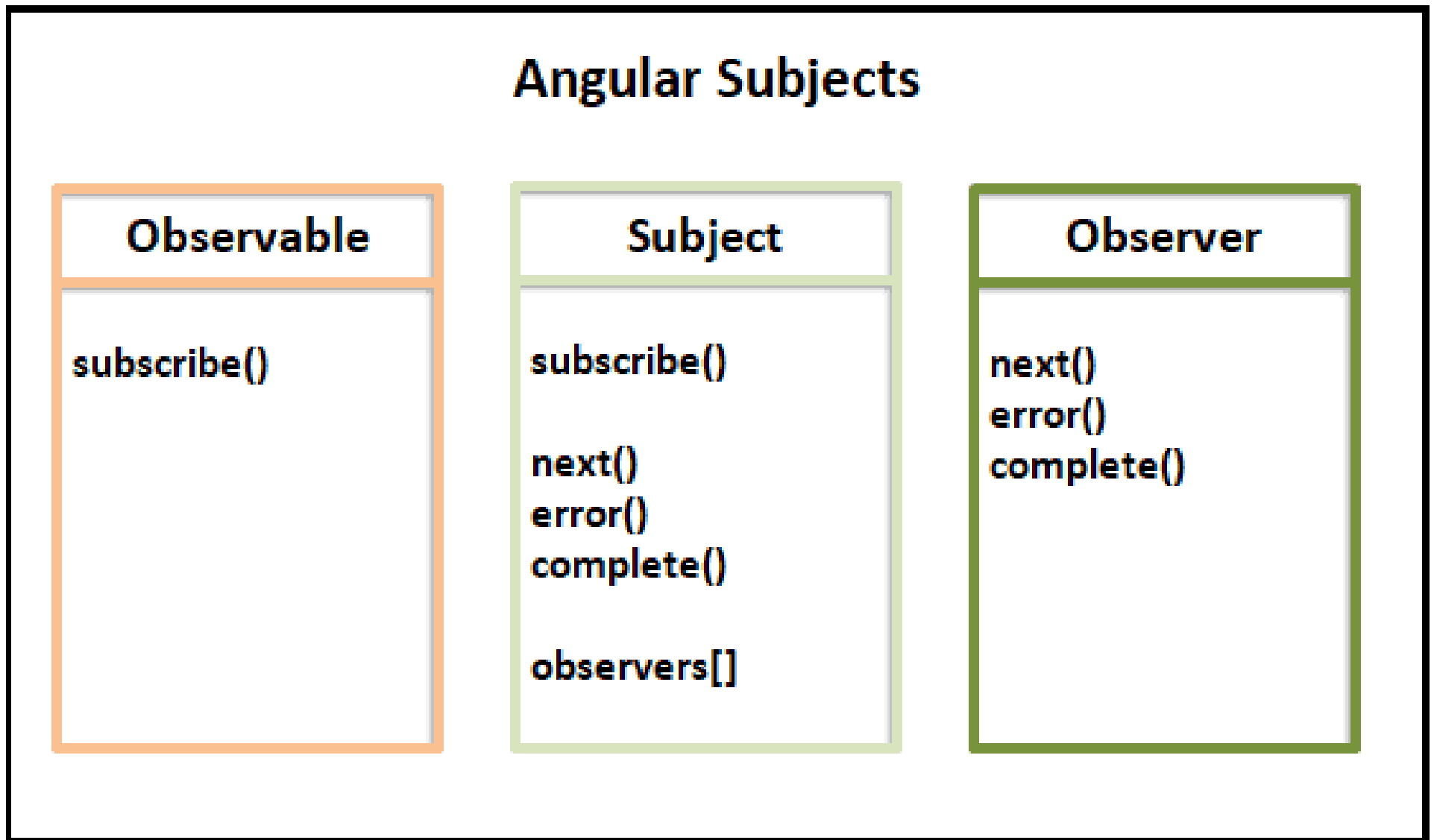
MULTICAST



- In comparison to a regular Observable, a **Subject allows values to be multicasted** to many Observers.
- A Subject and its subscribers have a **one-to-many relationship**.



How does Subjects work ?



- Subjects implement both **subscribe method** and **next, error & complete methods**.
- It also **maintains a collection of observers[]**



Methods of Subject

A Subject has three methods that you can use:

subscribe: with this method, you can activate the subscription of a new subscriber.

next: with this method, you can pass new values. All the current subscribers will receive this.

complete: with this method, you close all the subscriptions to the Subject.



Important points to be noted

- A vital detail is that a **Subject doesn't have an initial value**
- Every value passed with the next method will send the values to all the subscribers.
- But if the value is already sent before a subscriber is subscribed, it won't receive that data

Subject is hot Observable

Observables are classified into two groups.

- **Cold Observable**
- **Hot Observable**



Cold Observable

- The cold observable **does not activate the producer until there is a subscriber.**
- This is usually the **case when the observable itself produces the data.**

VS

Hot Observable

- The hot observable **does not wait for a subscriber to emit the data.** It can start emitting the values right away.
- This happens **when the producer is outside the observable.**



Creating a Subject in Angular



```
import { Component, VERSION } from "@angular/core";
import { Subject } from "rxjs";

@Component({
  selector: "my-app",
  templateUrl: "./app.component.html",
  styleUrls: ["./app.component.css"]
})
export class AppComponent {

  subjectdemo = new Subject();

  ngOnInit() {

    this.subjectdemo.subscribe(val => {
      console.log(val);
    });

    this.subjectdemo.next("1");
    this.subjectdemo.next("2");
    this.subjectdemo.complete();
  }
}
```



Follow me for more



aishwarya-dhuri



coder_aishya