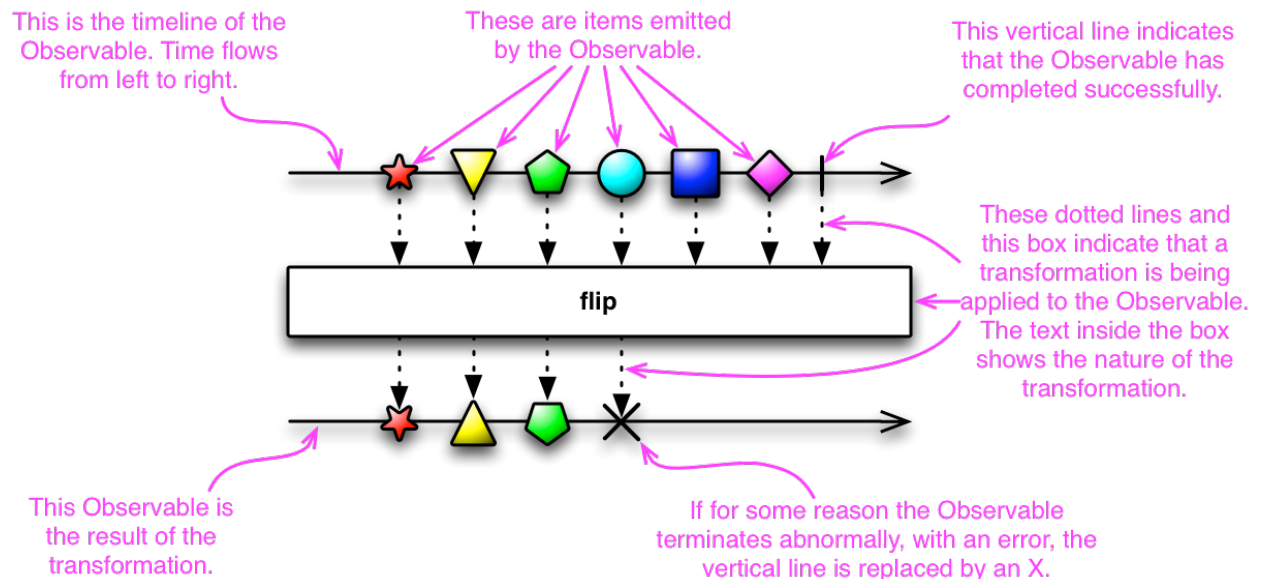


# Observable (Reactive Pattern)

1. In ReactiveX an observer subscribes to an observable, then the observer reacts to whatever item or sequence of items the observable emits. This pattern doesn't need to block while waiting for the observable to emit object.



2. You define a mechanism for retrieving and transforming the data, in form of 'Observable', and then subscribe an observer to it
3. An observer subscribes to an observable. An observable emits items or sends notification to observer by calling the observer's methods
4. An observer sometime is called 'subscriber', 'watcher' or 'reactor'
5. This pattern also known as 'reactor pattern'
6. Async model flow:
  - Define a method that does something useful with the returned value from the asynchronous call, this method is part of the observer ( callback **s** function of observer?)

- Define the asynchronous call itself as an observable
  - Attach the observer to that observable (this also initiates the actions of the observable)
  - Go on with your business. Whenever the call returns, the observer's method will operate on its return value or values - the items emitted by the observable
- 7.** Observer's methods. the observer must implement some subset of the following methods:
- onNext: called when the observable emits an item. This method takes the item emitted by the observable as its parameter
  - onError: the observable calls this method to indicate that it can't generate the expected value. It'll not make further calls to onNext or onCompleted. onError takes one parameter to indicate what caused the error
  - onCompleted: the observable calls this method after it has called onNext for the final time, if it has not encountered any errors
  - onNext called 'emissions'
  - onCompleted, onError called 'notifications'
  - unsubscribe method
- 8.** Hot observable begins emitting items as soon as it is created
- 9.** Cold observable waits until an observer subscribes to it before it begin emitting items
- 10.** Connectable observable doesn't begin emitting items until its Connect method is called
- 11.** Observable and observer are the start of reactivex. By themselves, they'd be nothing. The real power comes with reactive extension

- the operators allow you to transform, combine, manipulate, and work with the sequences of items emitted by observable

To learn more and get OneNote, visit:

<http://reactivex.io/documentation/observable.html>