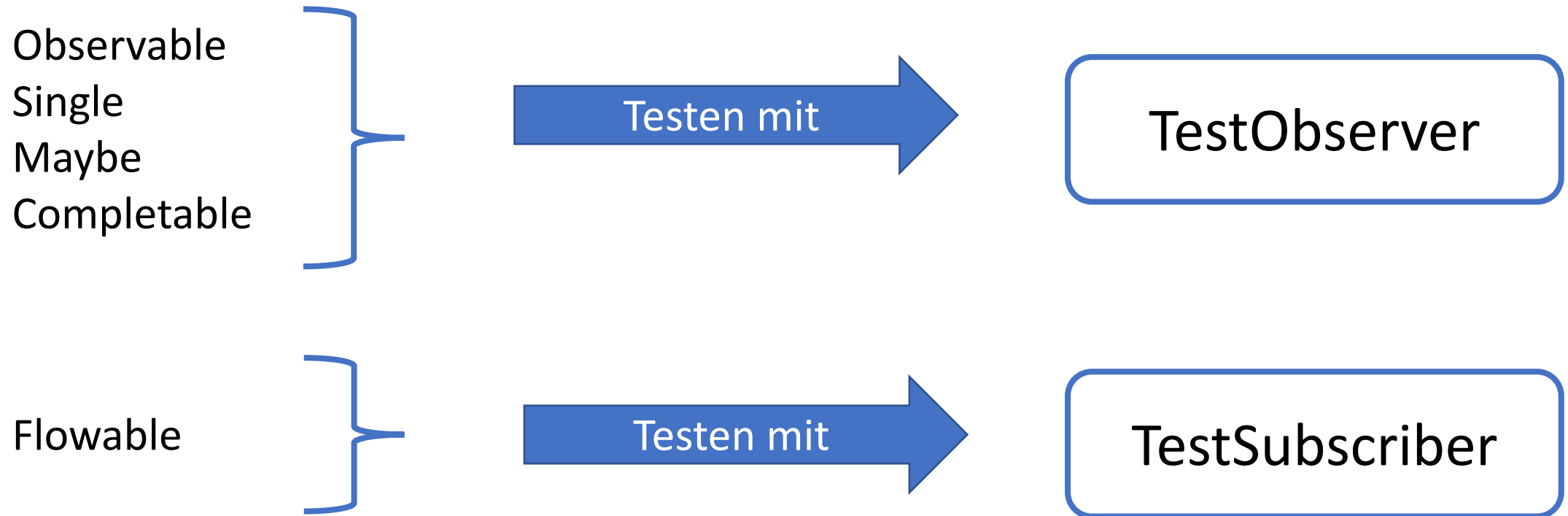


# RXKotlin 2 Testing

Overview für einfache Tests

# TestSubscriber und TestObserver

Falls die Subscription nicht auf dem selben Thread ist wie der Aufruf!



# Nützliche Funktionen auf TestObserver und TestSubscriber

- `assertComplete()`
- `assertNoErrors()`
- `assertValueCount(x: Int)`
- `assertResult(T...items)`
- `awaitDone(time: Long, unit: TimeUnit)`
  - Wartet blockierend auf den „terminal Event“ und bricht ab nach dem Timeout
- ...

# TestScheduler

## Hilft die Zeit zu kontrollieren

- *advanceTimeBy(long delayTime, TimeUnit unit)*
  - Die Clock vom Scheduler vor stellen (relativ vom jetzigen Zeitpunkt)
- *advanceTimeTo(long delayTime, TimeUnit unit)*
  - Die Clock vom Scheduler auf einen bestimmten Zeitpunkt stellen (absolut)

## Scheduler wird dem Observable mitgegeben

# Observables testen, die bereits einen Scheduler benutzen?

- Lösung -> RxJavaPlugins
- Eine TestRule implementieren, welche alle Schedulers (IO, Computation und NewThread) mit dem eigenen TestScheduler überschreibt.

```
var testScheduler = TestScheduler()  
RxJavaPlugins.setIoSchedulerHandler( {scheduler -> testScheduler} )  
RxJavaPlugins.setComputationSchedulerHandler({scheduler -> testScheduler})  
RxJavaPlugins.setNewThreadSchedulerHandler({scheduler ->testScheduler})
```

# Quellen

- <http://reactivex.io/RxJava/2.x/javadoc/io/reactivex/subscribers/TestSubscriber.html>
- <https://www.infoq.com/articles/Testing-RxJava2>
- <https://github.com/ReactiveX/RxJava/wiki/What%27s-different-in-2.0>