

Concurrency and Observable Transformations



Russell Elledge

ENTERPRISE ARCHITECT

@MC2FTW mastercraftcoding.com



Controlling Concurrency



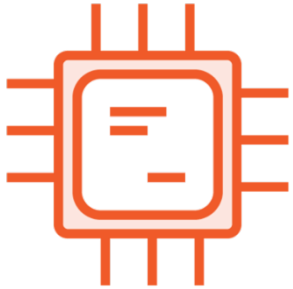
Schedulers

`subscribeOn`

`observeOn`



Schedulers



Computation



I/O



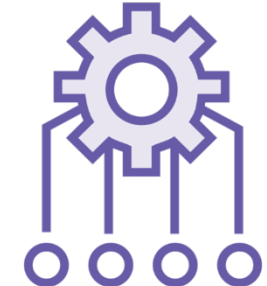
New Thread



Single



Trampoline



From(Executor)

Controlling Concurrency

Observable.subscribeOn

- Determines the Scheduler that is used to drive the emission of events.
- Highest in call chain wins

Observable.observeOn

- Determines what Scheduler processes the emissions of an Observable.
 - Position dependent



Demo



`subscribeOn`

`observeOn`



Transformation Operations



zip

map

flatMap

collect

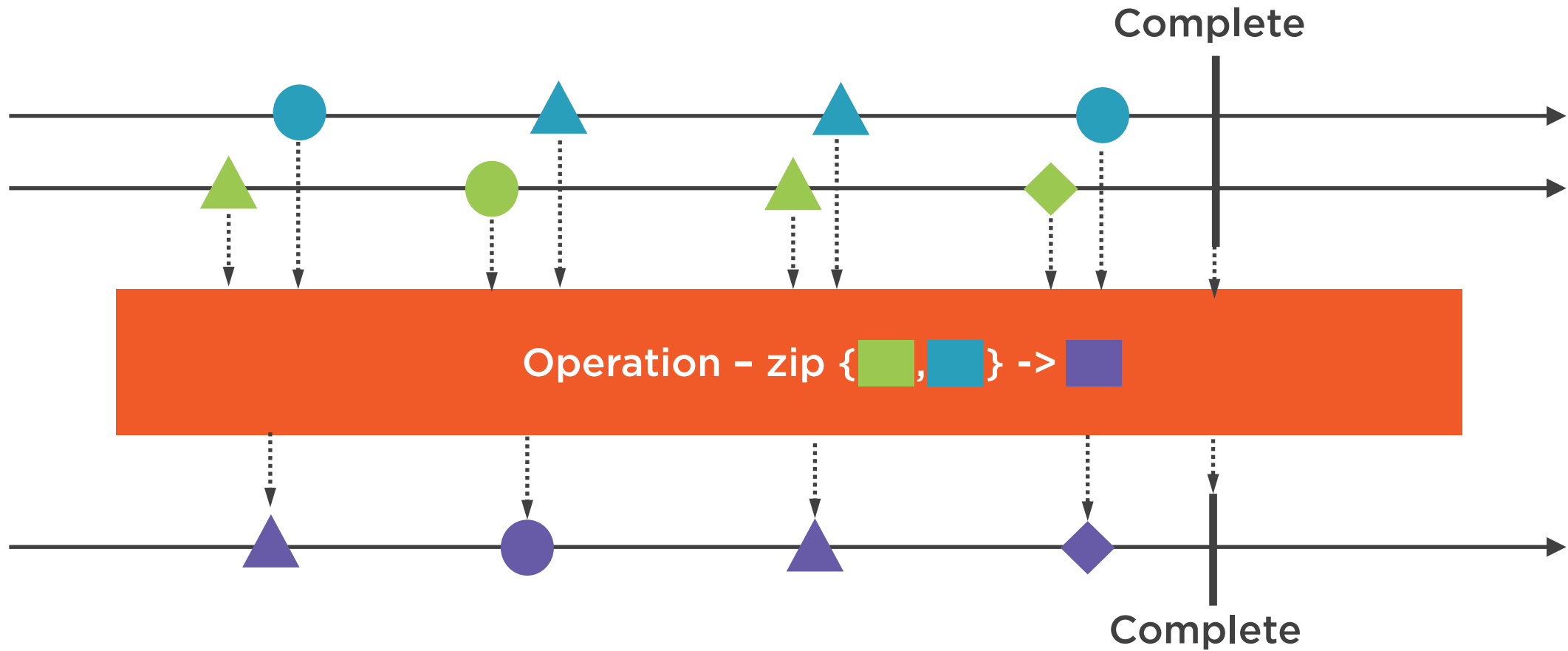
scan

groupBy

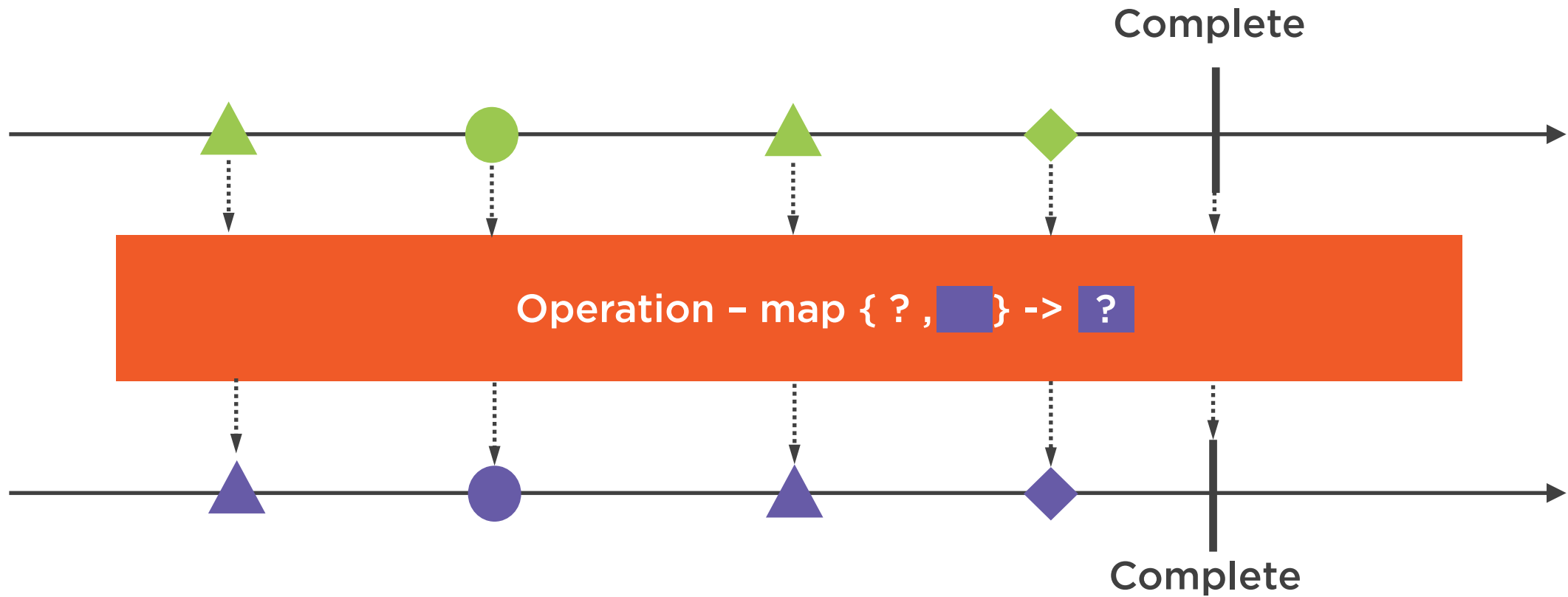
buffer



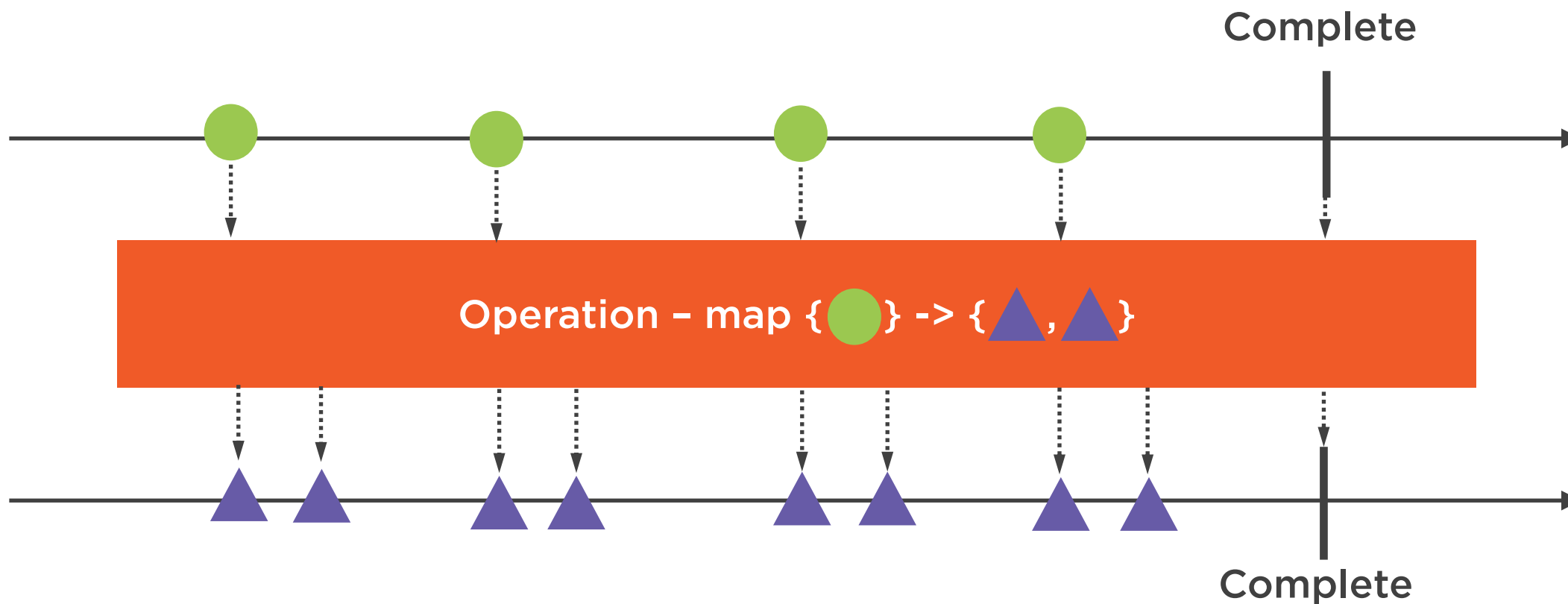
Zip Operator



Map Operator



FlatMap Operator



Demo



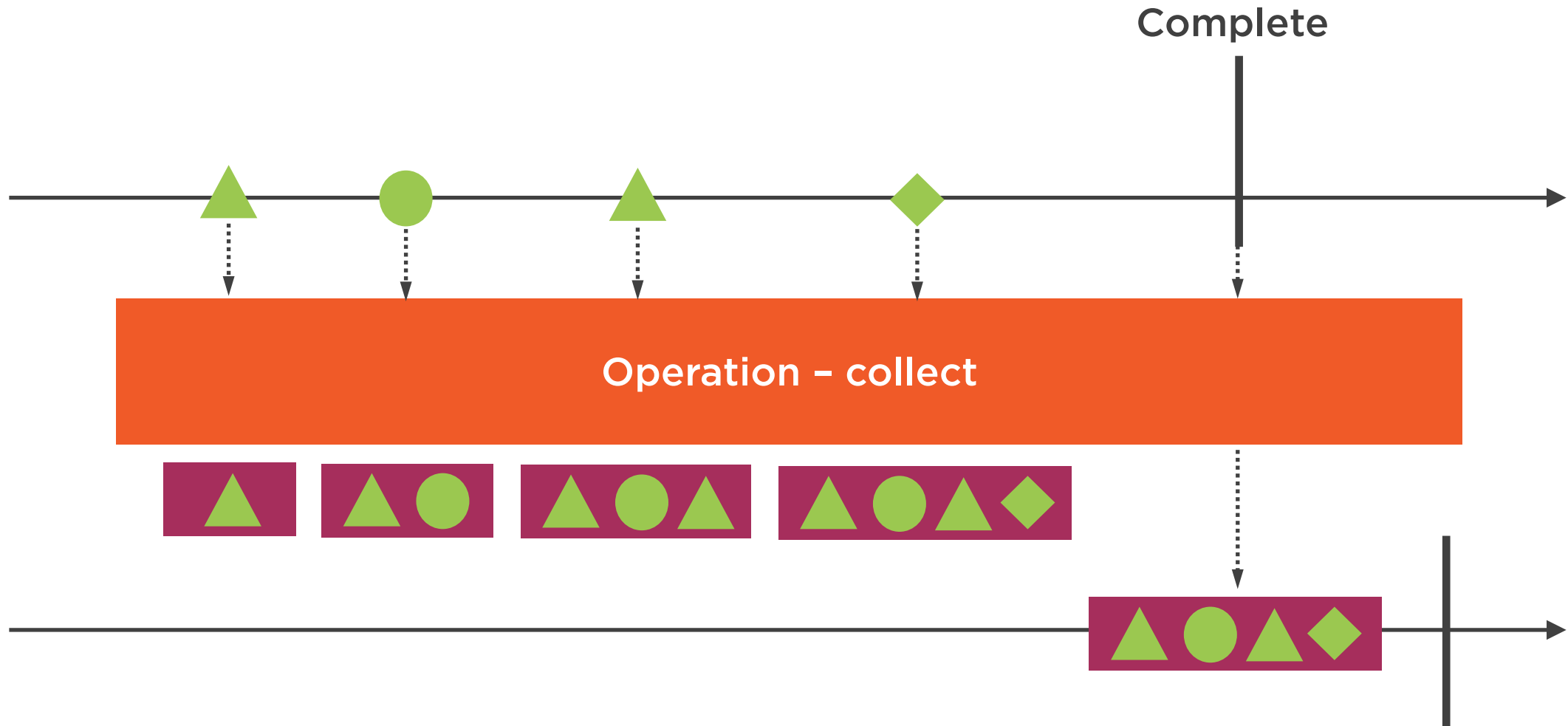
zip

map

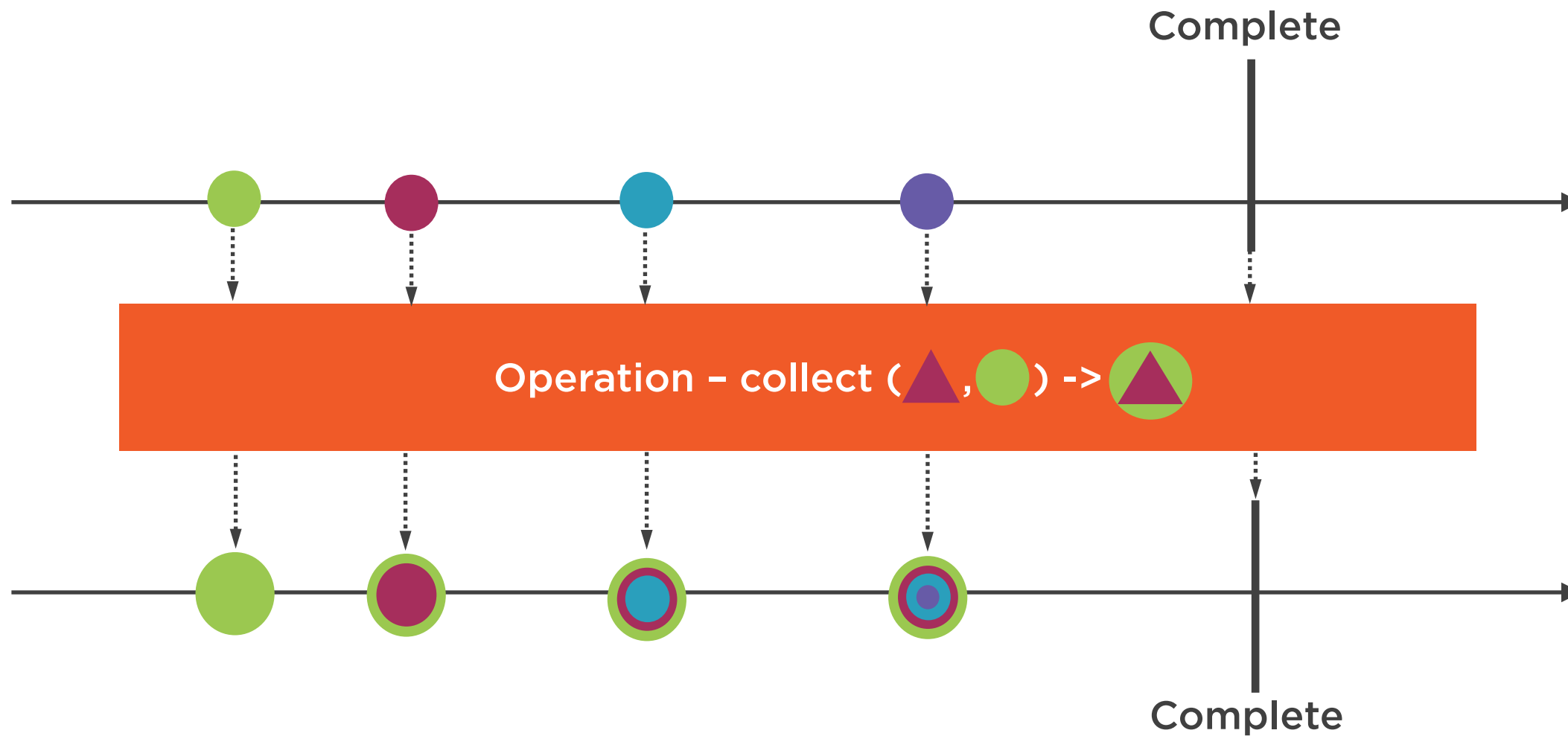
flatMap



Collect Operator



Scan Operator



Demo

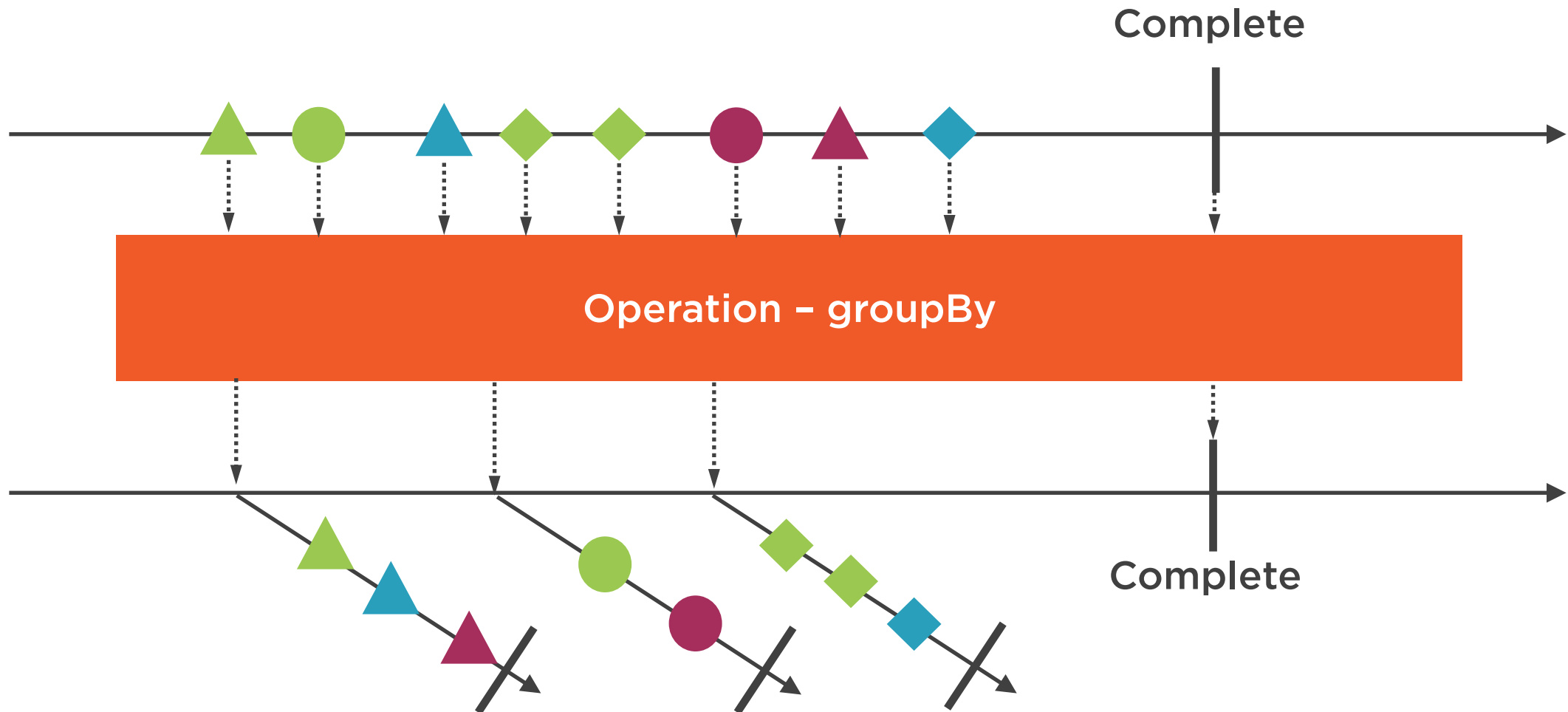


collect

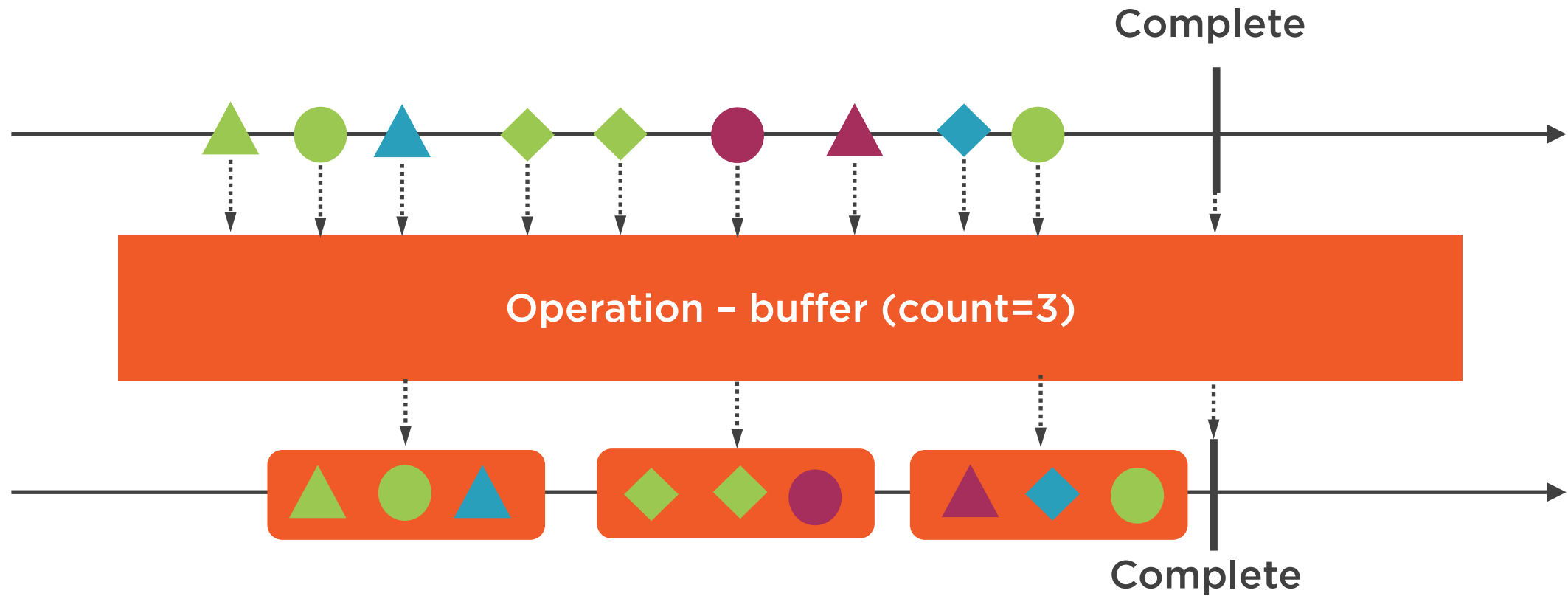
scan



groupBy Operator



buffer Operator



Demo



`groupBy`

`buffer`

