

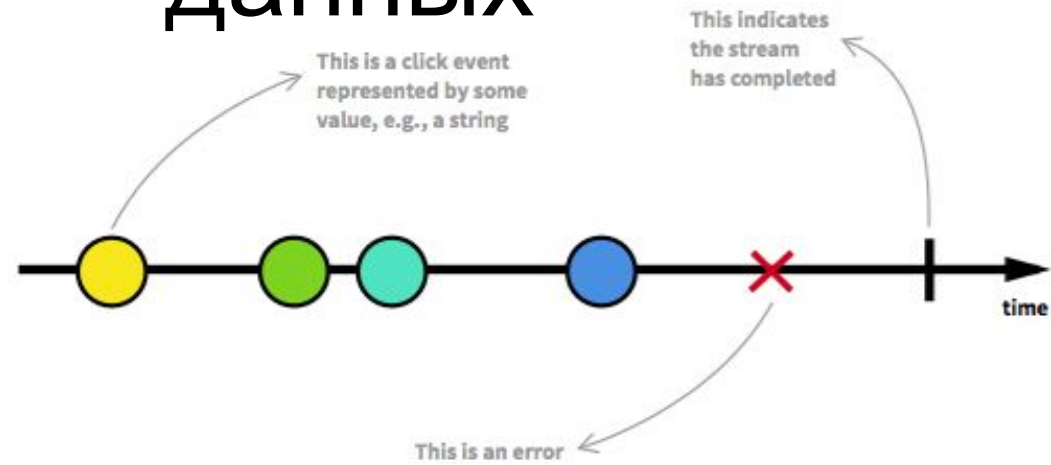
# Реактивное программирование

# Немного теории

Реактивное программирование — парадигма программирования, ориентированная на потоки данных и распространение изменений. Это означает, что должна существовать возможность легко выразить статические и динамические потоки данных, а также то, что нижележащая модель исполнения должна автоматически распространять изменения благодаря потоку данных.

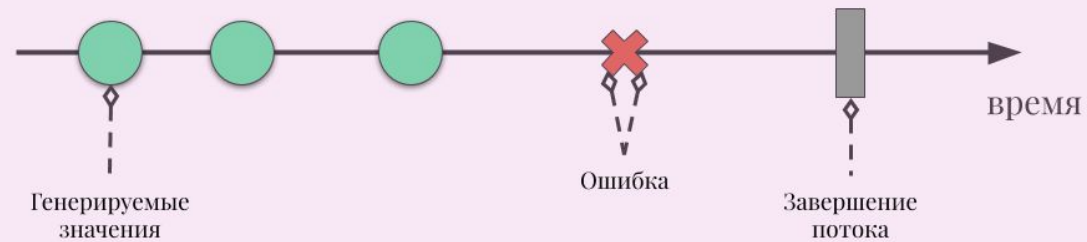
Реактивное программирование — программирование с асинхронными потоками данных.

# Потоки данных

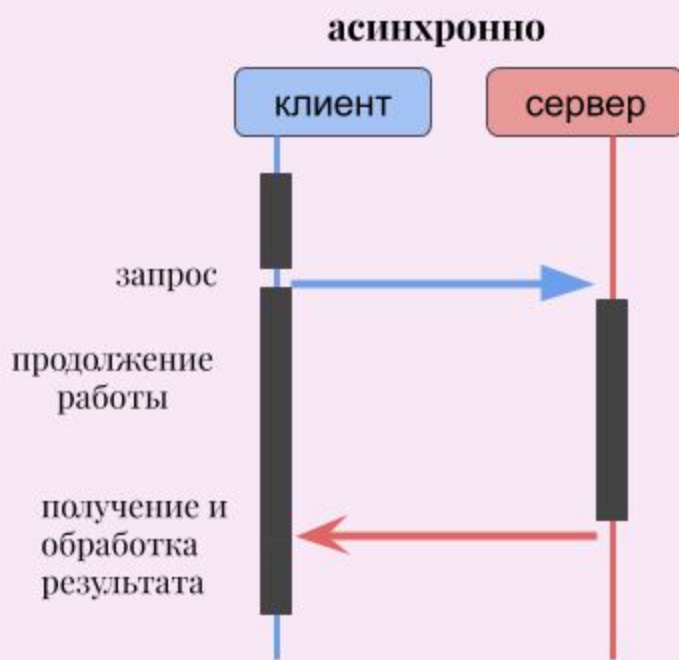
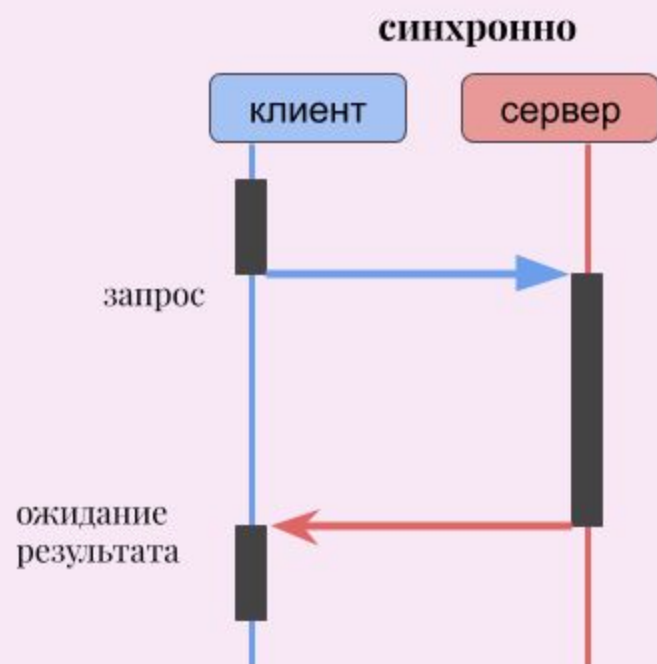


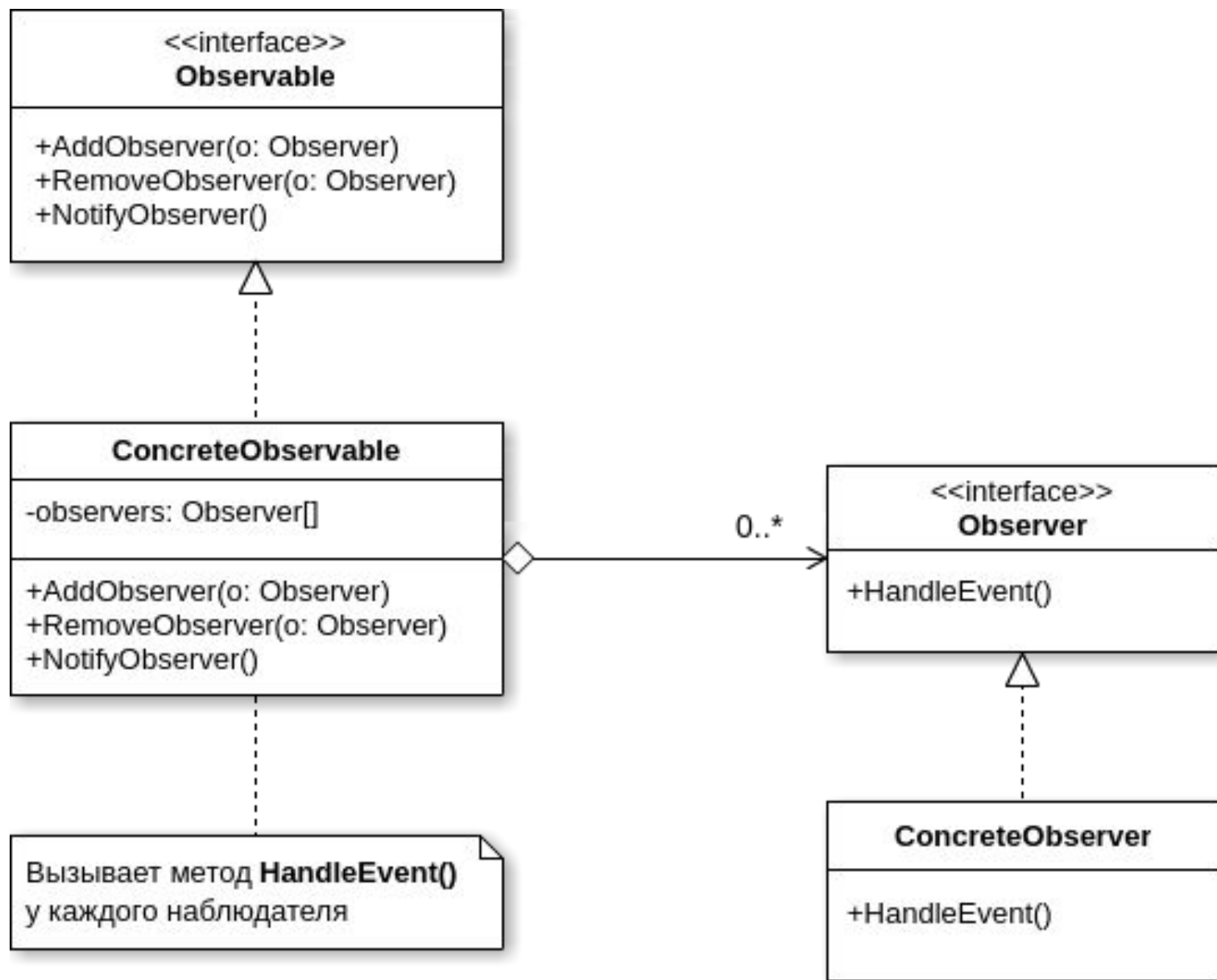
## Потоки

Поток — это массив данных или событий, отсортированных по времени, который может сообщать о том, что данные изменились.

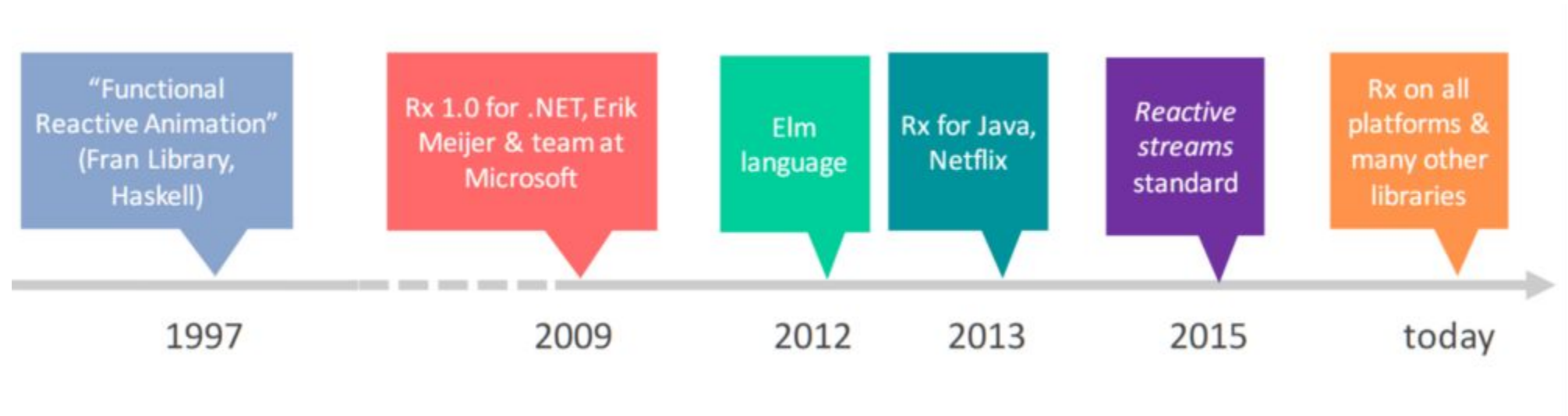


# Асинхронность



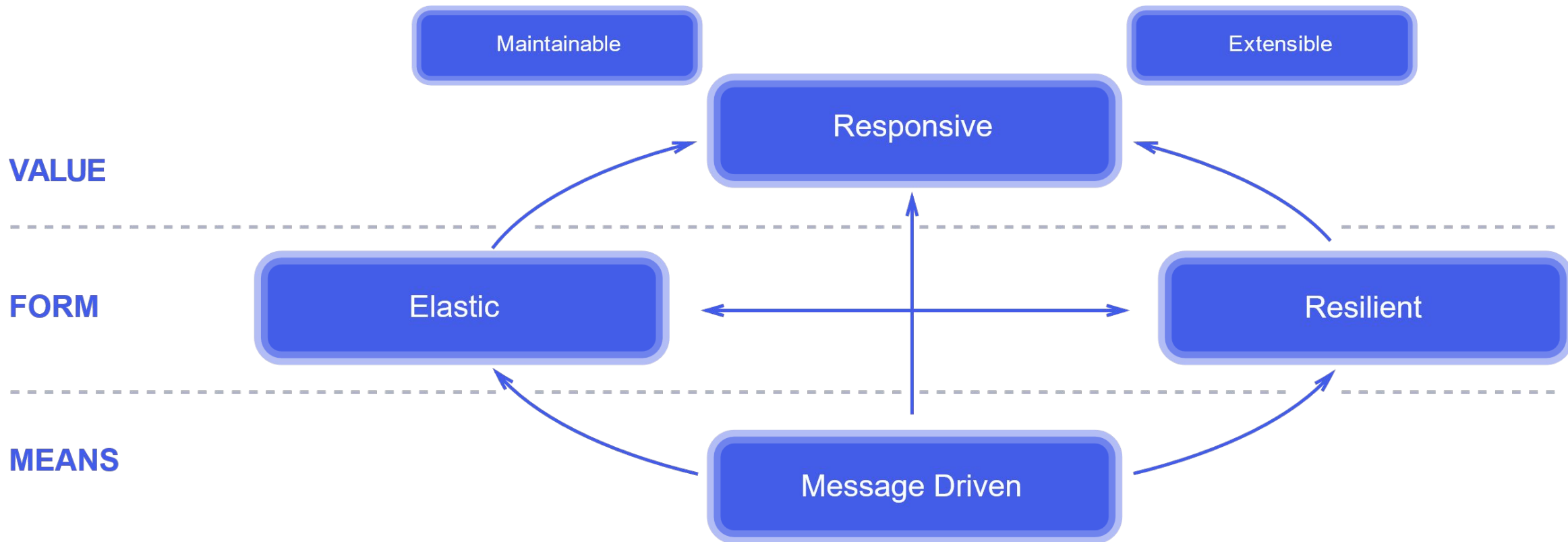


# История реактивности



# Пример реактивной системы

	A	B	C	D	E
1	Item Name	Chk	Availability		Check value
2	Coffee	<input type="checkbox"/>	Out of Stock		FALSE
3	Tea	<input type="checkbox"/>	Out of Stock		FALSE
4	Snacks	<input type="checkbox"/>	Out of Stock		FALSE
5	Spieces	<input type="checkbox"/>	Out of Stock		FALSE
6	Cleaning supplies	<input type="checkbox"/>	Out of Stock		FALSE
7	Cheese	<input type="checkbox"/>	Out of Stock		FALSE
8	Kombucha	<input type="checkbox"/>	Out of Stock		FALSE
9	Coconut Water	<input type="checkbox"/>	Out of Stock		FALSE
10	Truffles	<input type="checkbox"/>	Out of Stock		FALSE
11	Gourmet Honey	<input type="checkbox"/>	Out of Stock		FALSE
12					
13					
14			<b>Total Items</b>	10	
15			<b>Items Available</b>	0	
16			<b>Items Unavailable</b>	10	
17					





# Spring Boot 2.0



## Reactor

OPTIONAL DEPENDENCY

### Reactive Stack

Spring WebFlux is a non-blocking web framework built from the ground up to take advantage of multi-core, next-generation processors and handle massive numbers of concurrent connections.

Netty, Servlet 3.1+ Containers

Reactive Streams Adapters

Spring Security Reactive

**Spring WebFlux**

**Spring Data Reactive Repositories**

Mongo, Cassandra, Redis, Couchbase

### Servlet Stack

Spring MVC is built on the Servlet API and uses a synchronous blocking I/O architecture with a one-request-per-thread model.

Servlet Containers

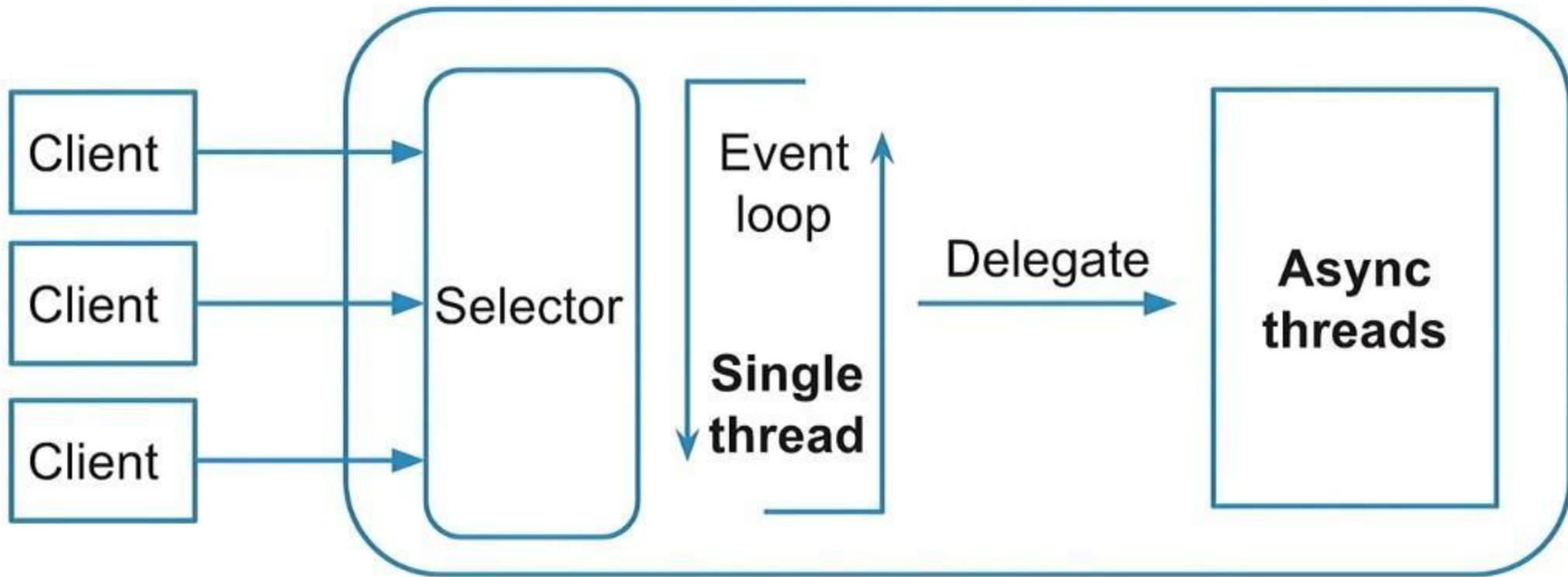
Servlet API

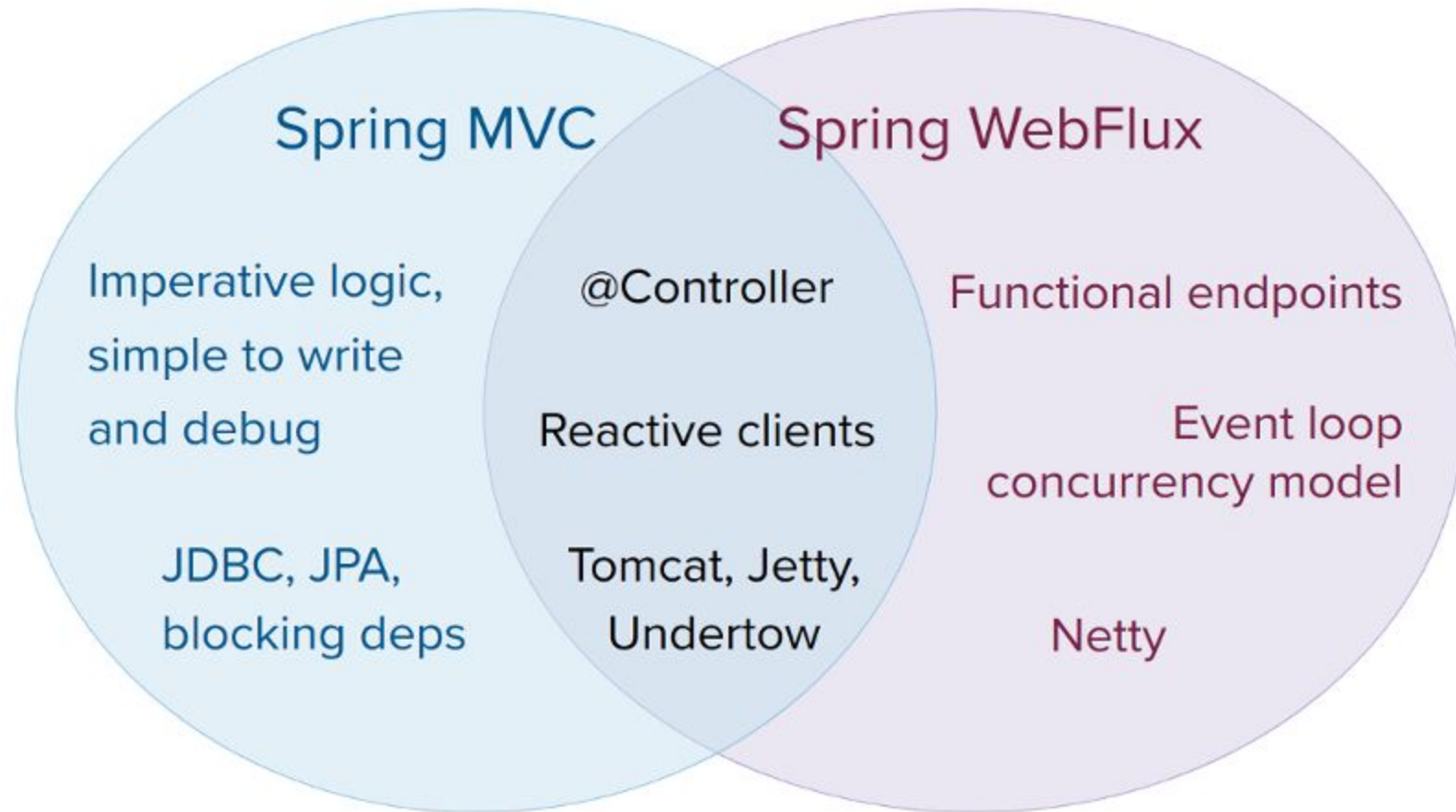
Spring Security

**Spring MVC**

**Spring Data Repositories**

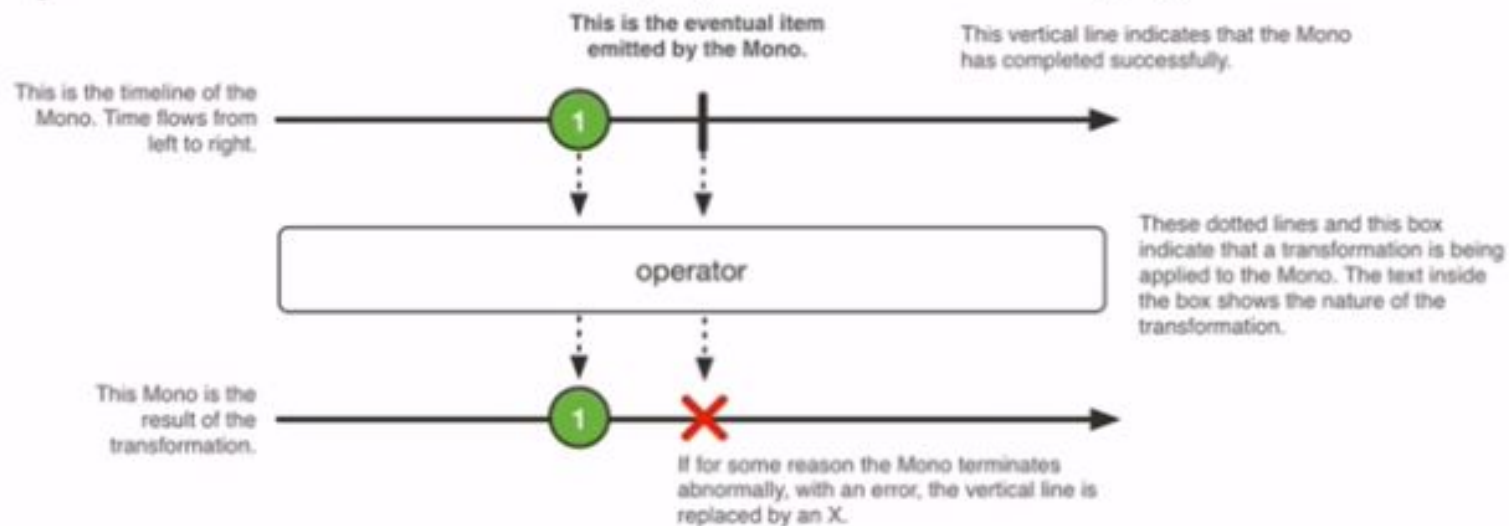
JDBC, JPA, NoSQL





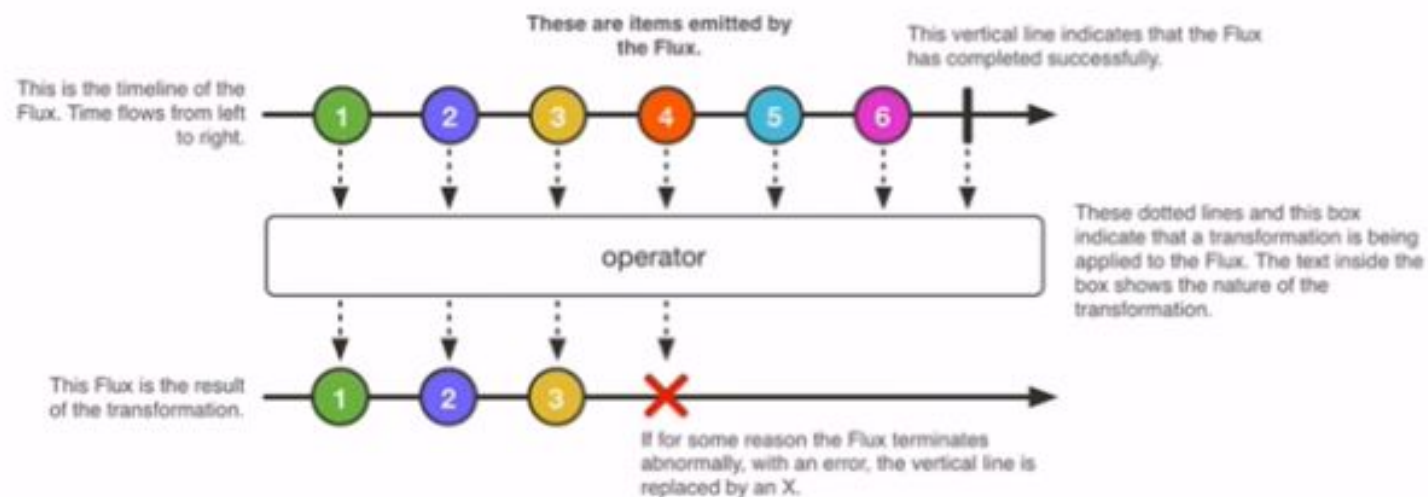
# Mono<T>

- Implements Reactive Streams **Publisher**
- 0 to 1 element
- Operators: `mono.then(...).otherwise(...)`



# Flux<T>

- Implements Reactive Streams **Publisher**
- 0 to n elements
- Operators: `flux.map(...).zip(...).flatMap(...)`



# Литература

1. <https://clck.ru/Twr48> - Wiki
2. <https://www.reactivemanifesto.org/>
3. <https://clck.ru/MUWen> - Наблюдатель (шаблон проектирования)  
Wiki
4. <https://clck.ru/Twr5K> - познавательная статья
5. <http://www.reactive-streams.org/>
6. <https://projectreactor.io/>
7. <https://spring.io/guides/gs/reactive-rest-service/>