

What is Reactive Programming?

Reactive Programming

- Reactive Programming is an asynchronous programming paradigm focused on streams of data.
- “Reactive programs also maintain a continuous interaction with their environment, but at a speed which is determined by the environment, not the program itself. Interactive programs work at their own pace and mostly deal with communication, while reactive programs only work in response to external demands and mostly deal with accurate interrupt handling. Real-time programs are usually reactive.” - Gerard Berry, French Computer Scientist

Common Use Cases

- External Service Calls
- Highly Concurrent Message Consumers
- Spreadsheets
- Abstraction Over Asynchronous Processing
 - Abstract whether or not your program is synchronous or asynchronous

Features of Reactive Programming

- Data Streams
- Asynchronous
- Non-blocking
- Backpressure
- Failures as Messages

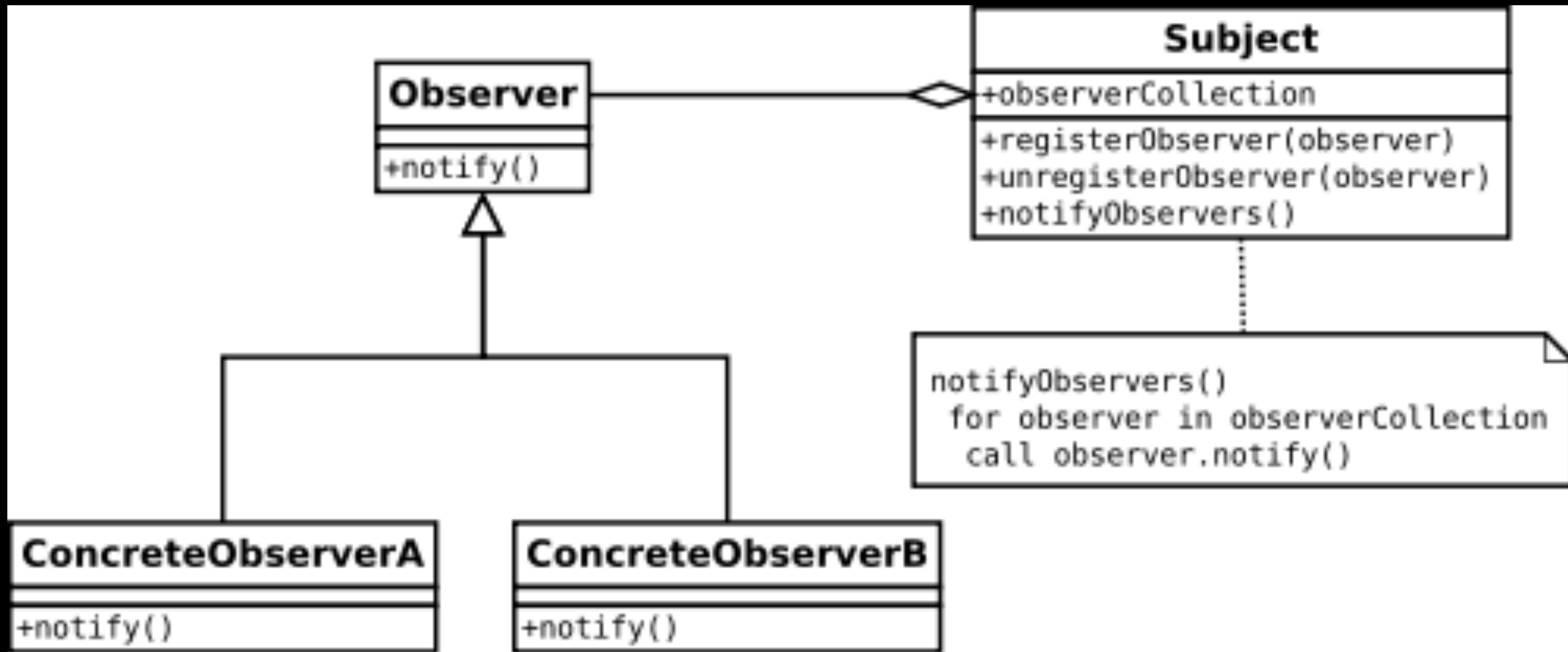
Data Streams

- Data Streams can be just about anything.
- Mouse clicks, or other user interactions.
- JMS Messages, RESTful Service calls, Twitter feed, Stock Trades, list of data from a database.
- A Stream is a sequence of events ordered in time.
- Events you want to listen to.

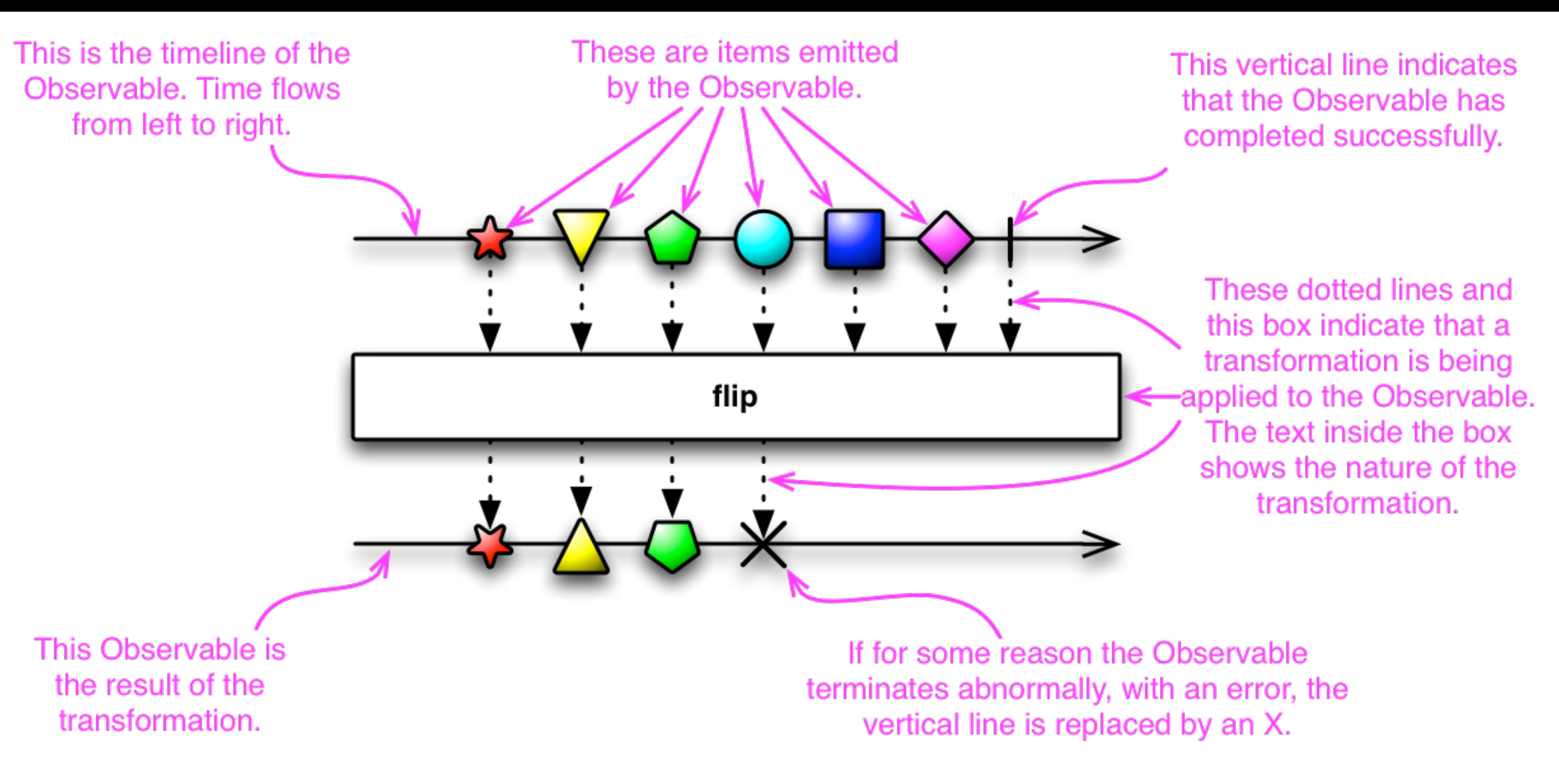
Asynchronous

- Events are captured asynchronously.
- A function is defined to execute when an event is emitted.
- Another function is defined if an error is emitted.
- Another function is defined when complete is emitted.

GoF Observer Pattern



ReactiveX Observable



Non-Blocking

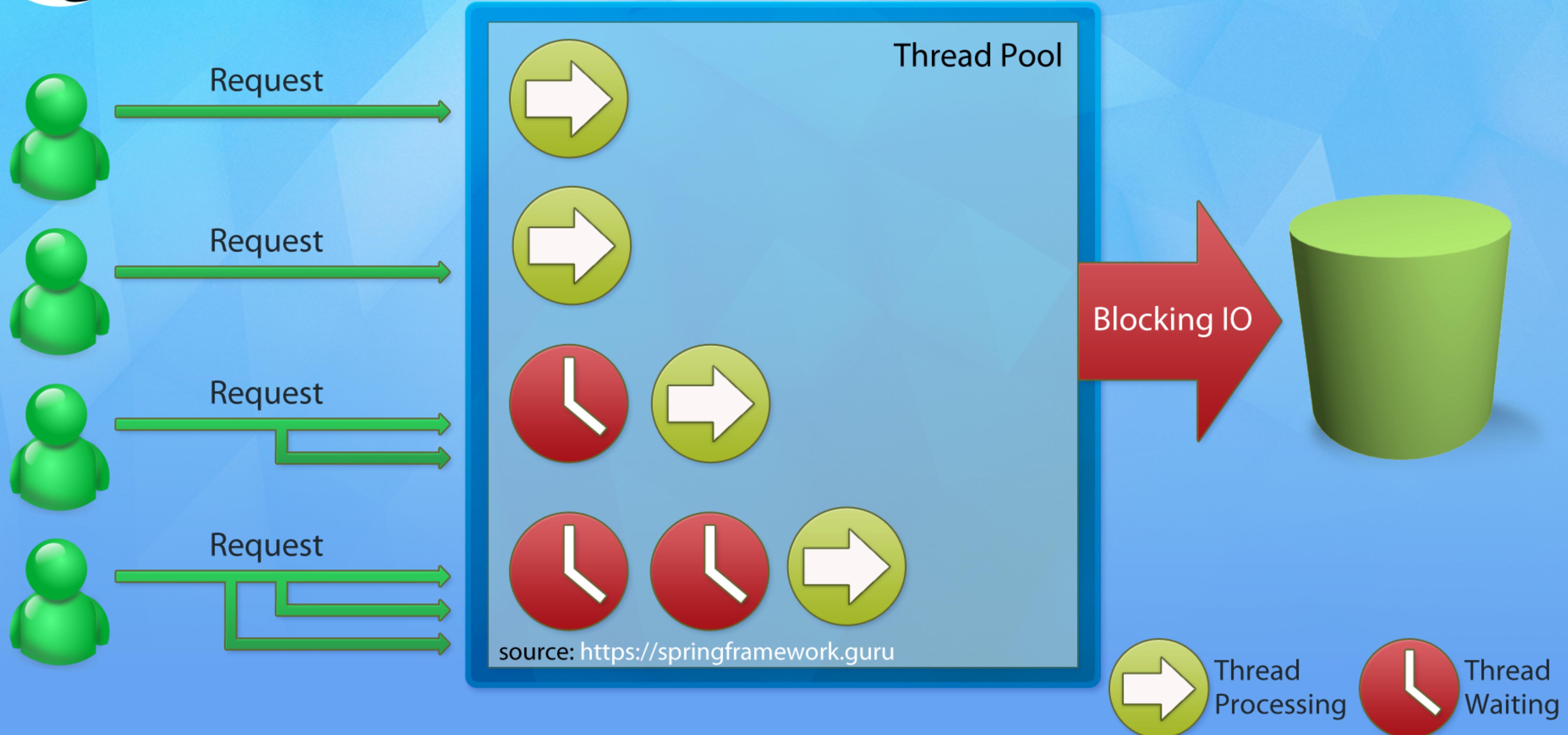
- The concept of using non-blocking is important.
- In Blocking, the code will stop and wait for more data (ie reading from disk, network, etc)
- Non-blocking in contrast, will process available data, ask to be notified when more is available, then continue.

Non-Blocking

- The concept of using non-blocking is important.
- In Blocking, the code will stop and wait for more data (ie reading from disk, network, etc)
- Non-blocking in contrast, will process available data, ask to be notified when more is available, then continue.

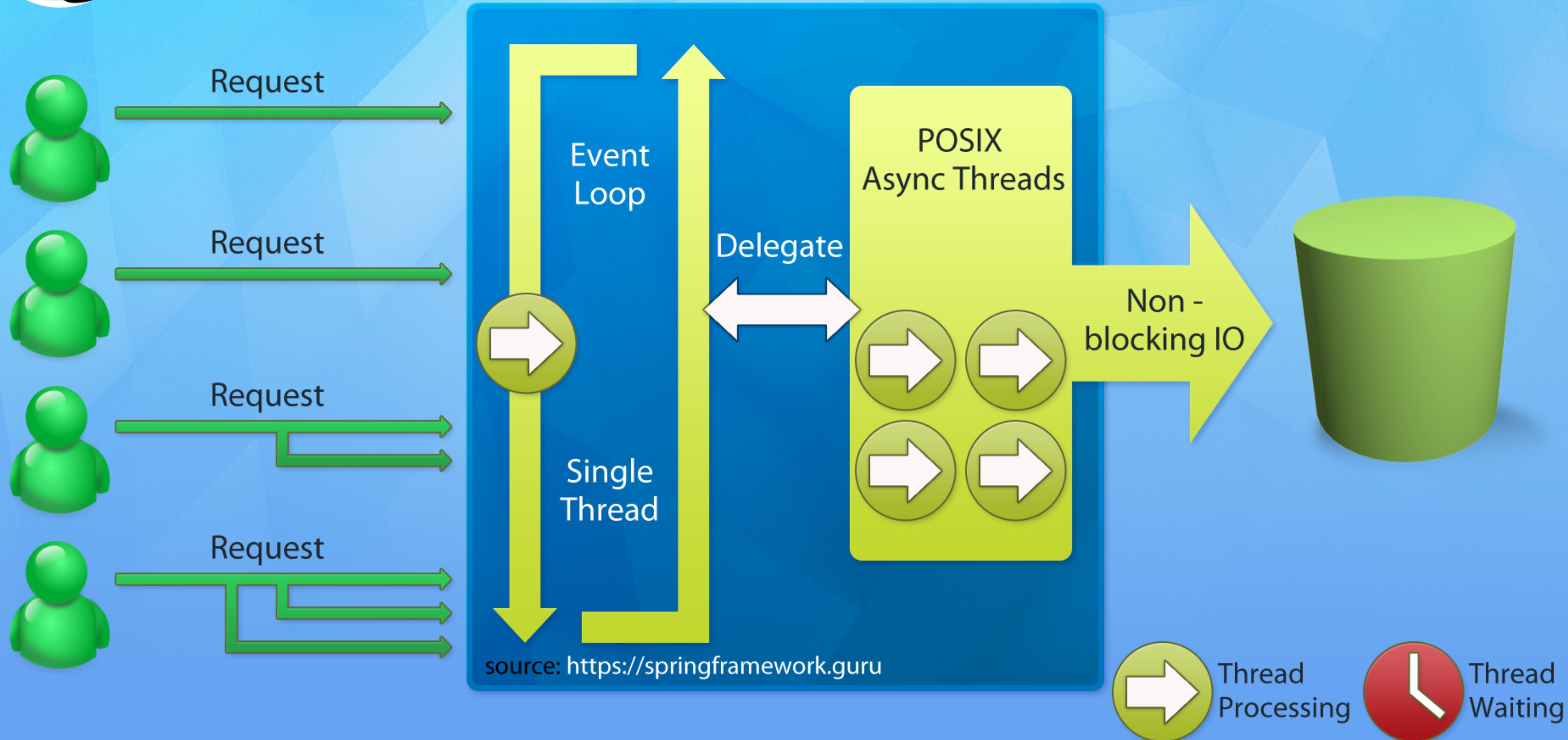


Multi Threaded Server





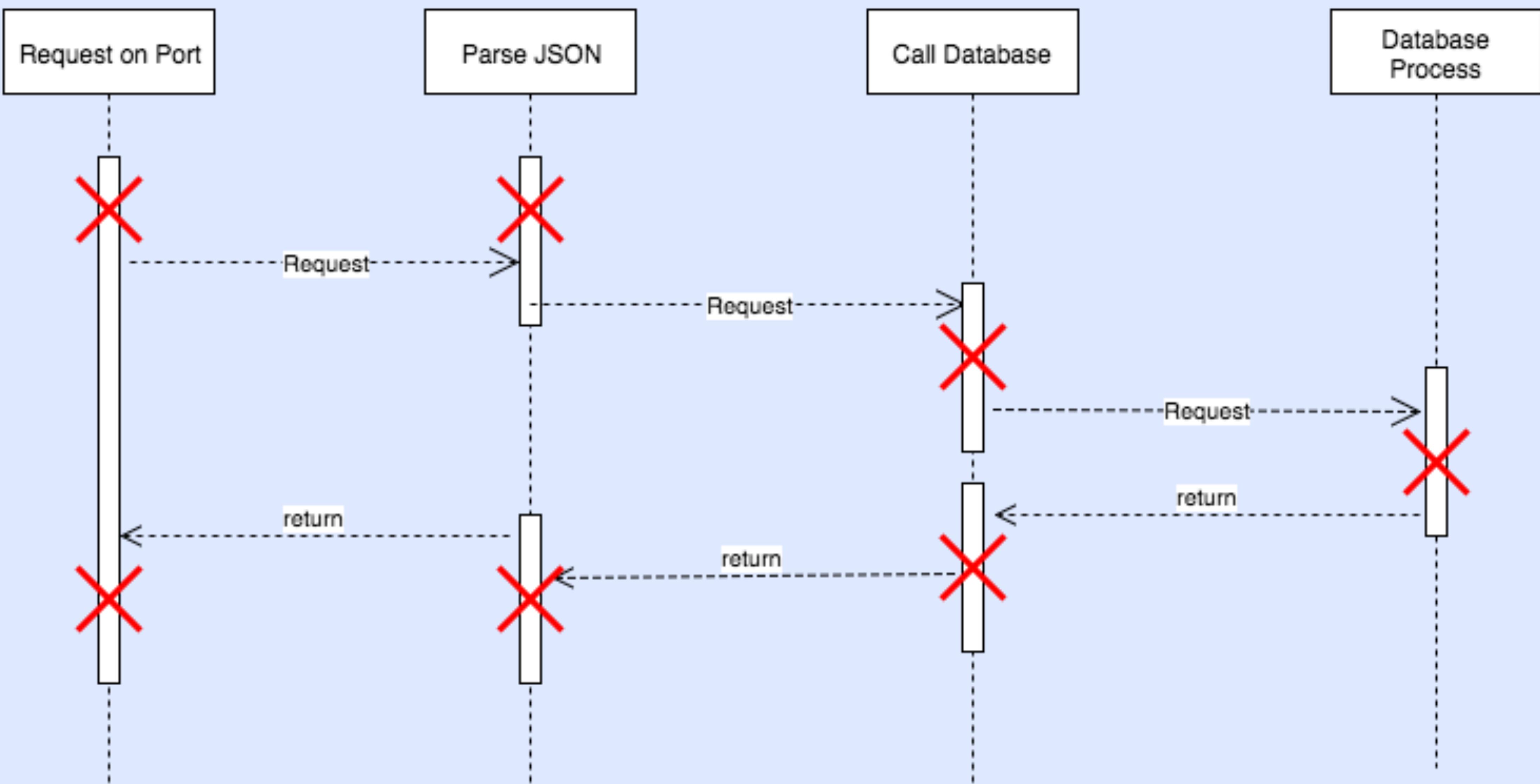
Node.js Server







Web Request



Back Pressure

- The ability of the subscriber to throttle data.



Failures as Messages

- Exceptions are not thrown in a traditional sense.
 - Would break processing of stream.
- Exceptions are processed by a handler function.

Key Take Aways

- Reactive Programming focuses on processing streams of data.
- Traditional CRUD applications are still alive and well.

