Overview of the Java Reactive Streams API (Part 2)

Douglas C. Schmidt
d.schmidt@vanderbilt.edu
### www.dre.vanderbilt.edu/~schmidt

Professor of Computer Science

Institute for Software Integrated Systems

Vanderbilt University Nashville, Tennessee, USA

# Learning Objectives in this Part of the Lesson

- Understand the key benefits & principles underlying the reactive programming paradigm

- Know the Java reactive streams API

  - Be aware of key patterns

  - Recognize key abstractions & their interactions
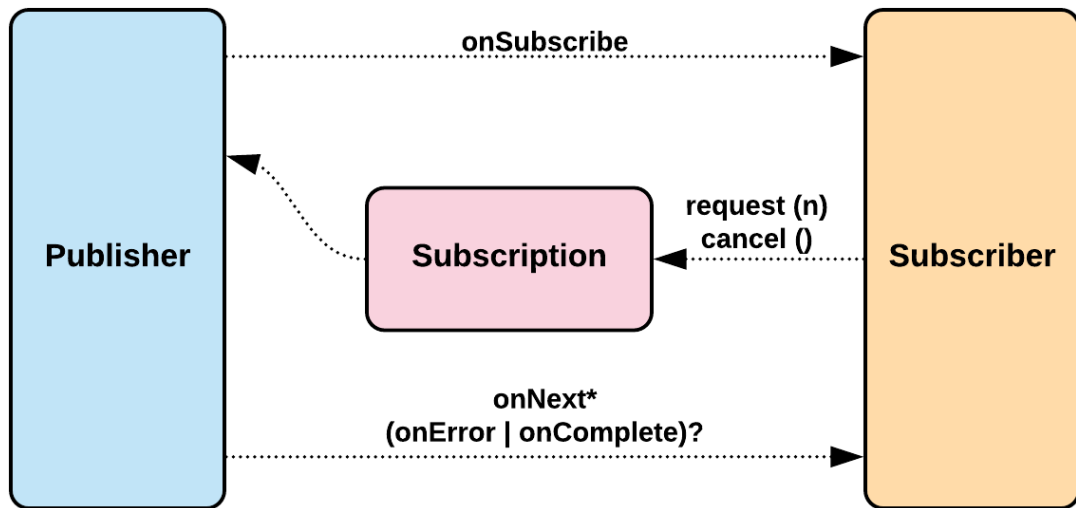
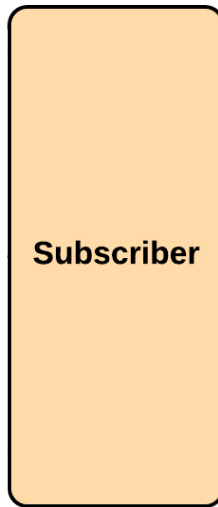# Key Abstractions & Interactions in the Java Flow API

Key Abstractions & Interactions in the Java Flow API

- A "flow" involves interactions between three key abstractions

Key Abstractions & Interactions in the Java Flow API
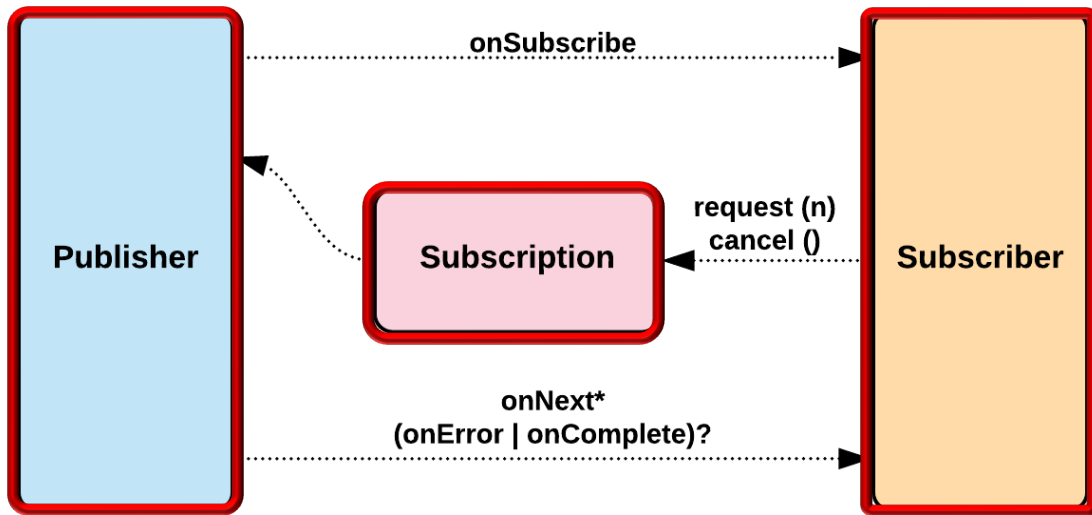
- A "flow" involves interactions between three key abstractions

Publisher

Subscriber

1. Publisher(s) are sources that produce 0+ events that can be pushed to subscriber(s)

- A "flow" involves interactions between three key abstractions

Publisher

Subscriber

2. Subscriber(s) are sinks that register for & consume events pushed by publisher(s)
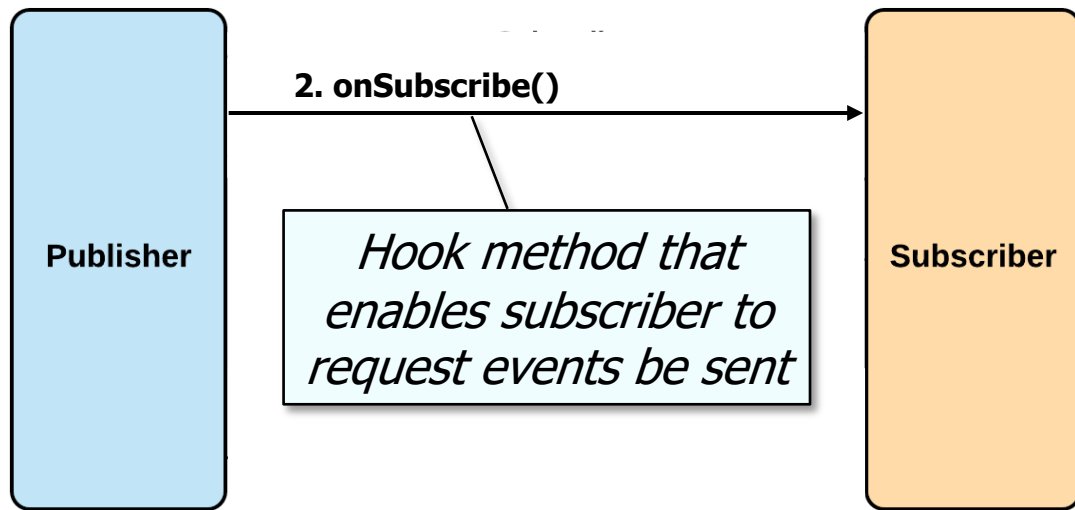
Key Abstractions & Interactions in the Java Flow API

- A "flow" involves interactions between three key abstractions

Publisher

1. subscribe()

Request publisher to start streaming data
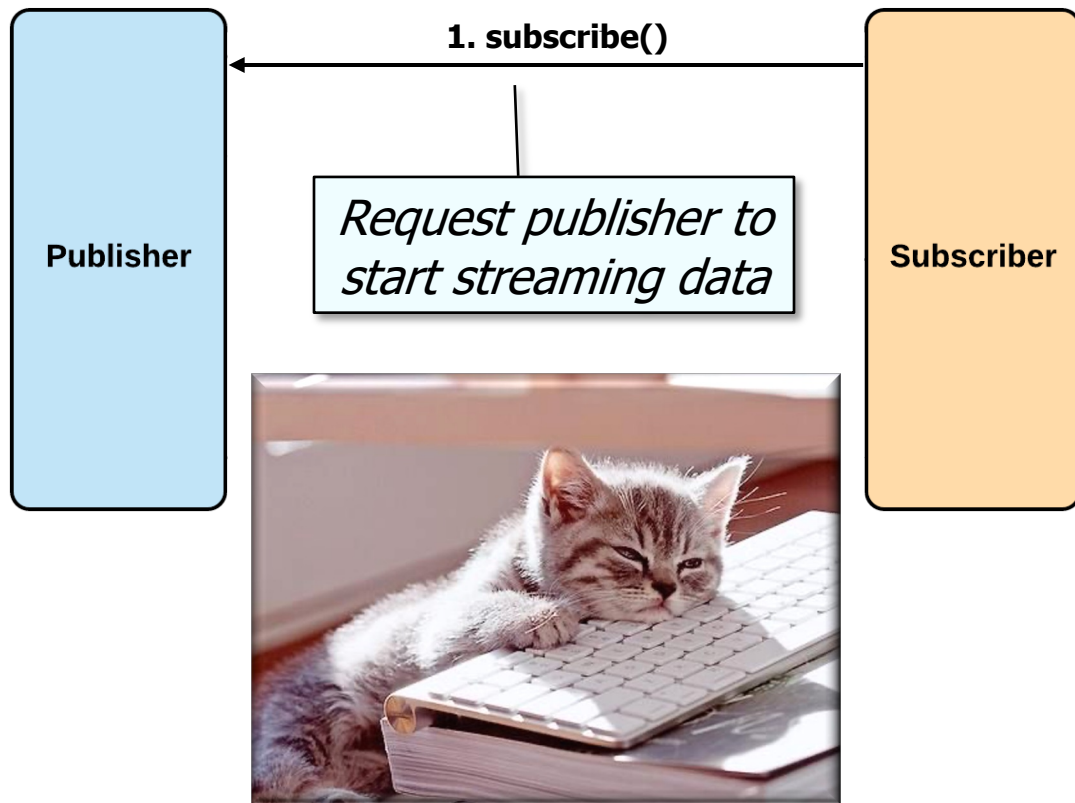
Subscriber

A reactive stream is "lazy" & just starts processing when subscribe() is called

Key Abstractions & Interactions in the Java Flow API

- A "flow" involves interactions between three key abstractions

Publisher

2. onSubscribe()

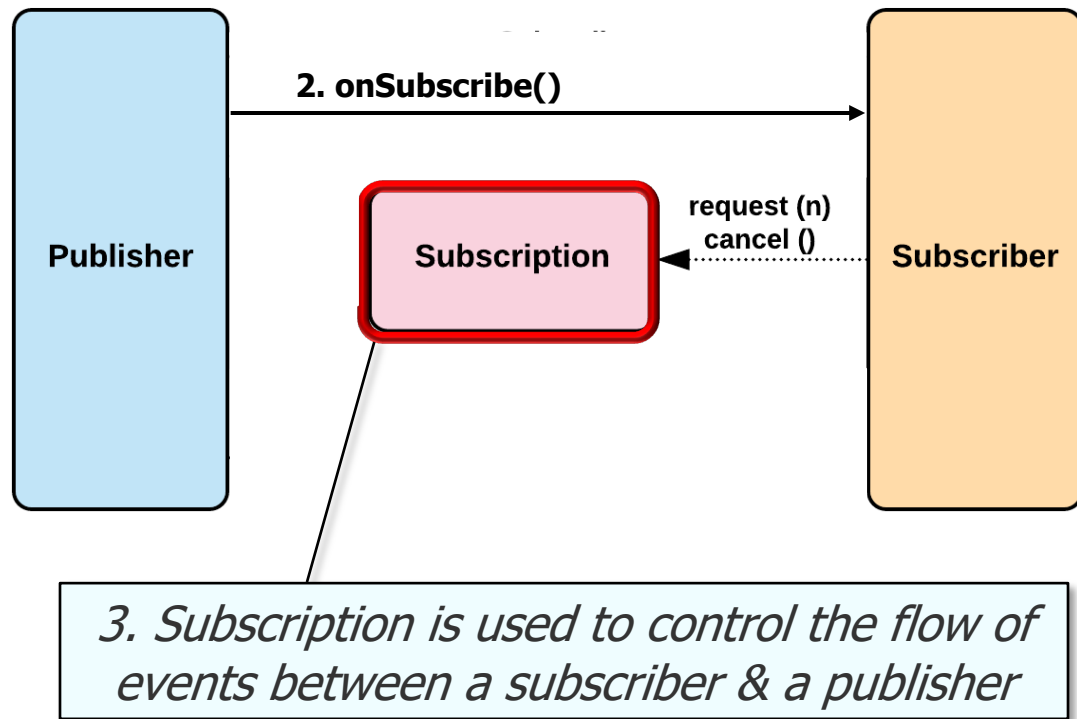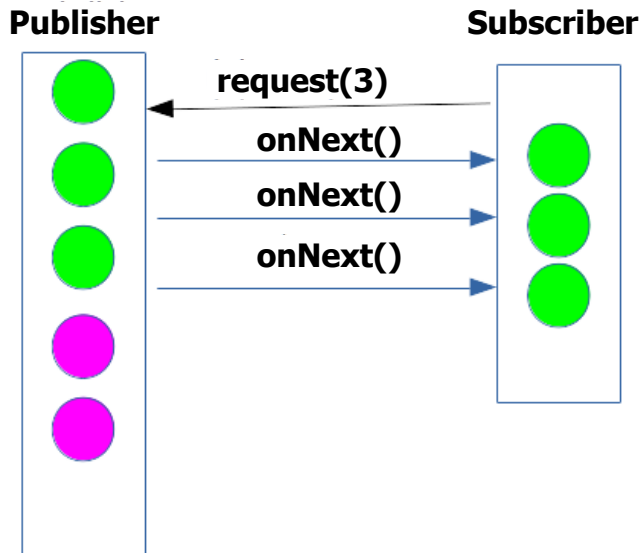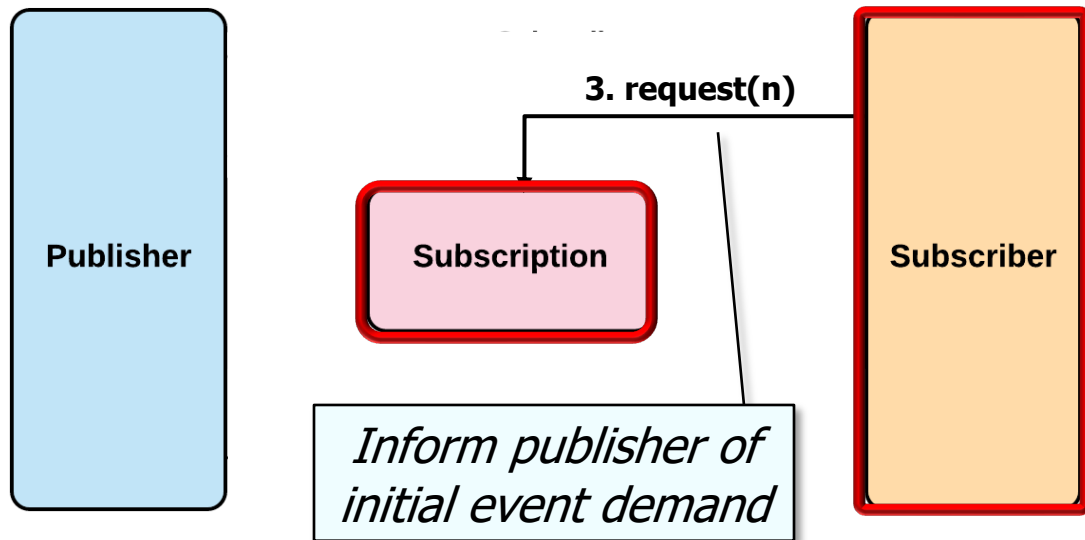*Hook method that enables subscriber to request events be sent*

Subscriber

Key Abstractions & Interactions in the Java Flow API

- A "flow" involves interactions between three key abstractions

Publisher   **Subscriber**

request(3)

onNext()

onNext()

onNext()

Publisher   **2. onSubscribe()**   **Subscriber**

Subscription   request (n) cancel ()

3. Subscription is used to control the flow of events between a subscriber & a publisher

See en.wikipedia.org/wiki/Flow_control_(data)

- A "flow" involves interactions between three key abstractions

Publisher

3. request(n)

Subscription

Subscriber

Inform publisher of initial event demand

No events are sent by a publisher until demand is signaled via this method

Key Abstractions & Interactions in the Java Flow API

- A "flow" involves interactions between three key abstractions
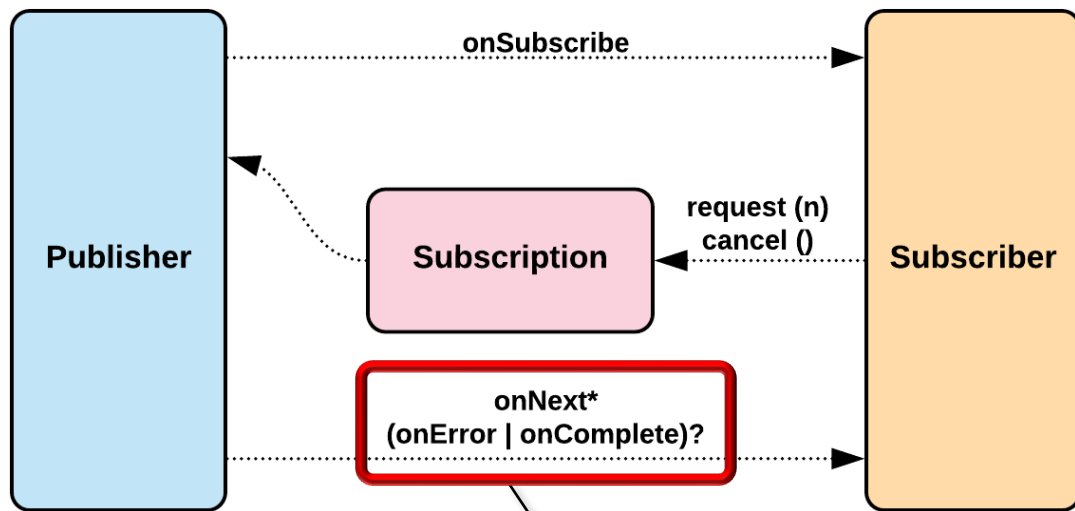
Publisher(s) push events to registered subscriber(s) by invoking hook methods

Key Abstractions & Interactions in the Java Flow API

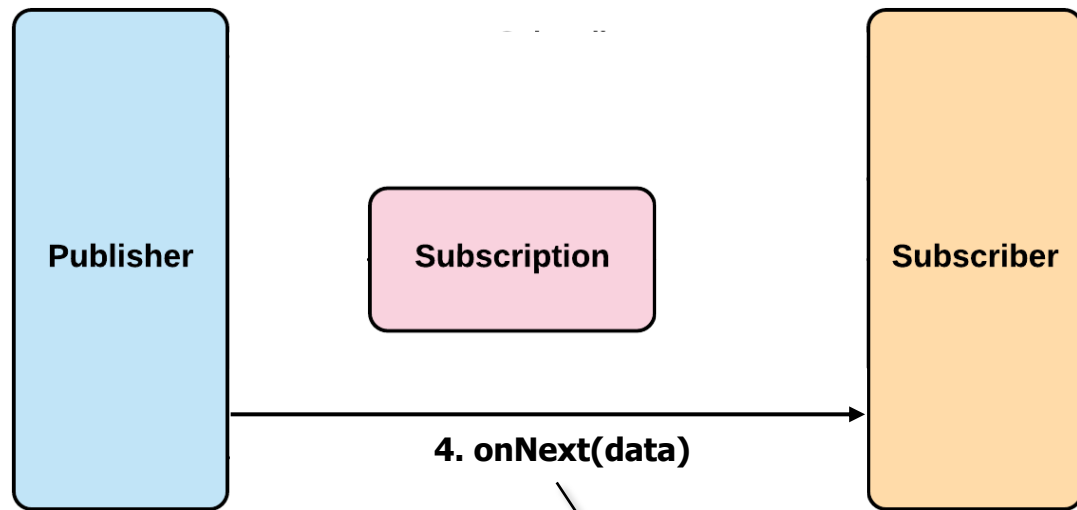- A "flow" involves interactions between three key abstractions

NOTIFICATION

Publisher
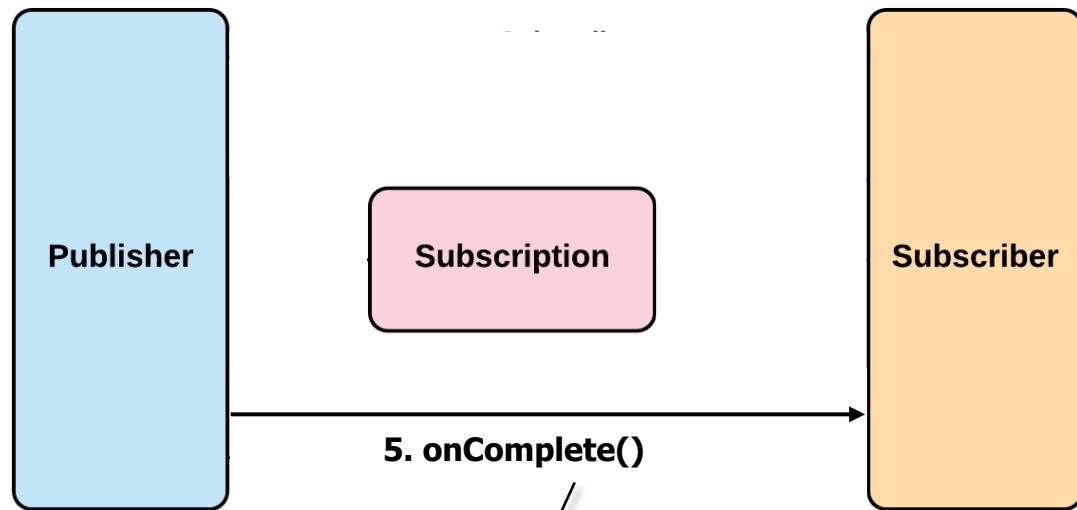
Subscription

Subscriber

4. onNext(data)

Data notification hook method called by the publisher in response to requests

There can be 0 or more onNext() notifications, which form a "stream"

Key Abstractions & Interactions in the Java Flow API

- A "flow" involves interactions between three key abstractions

Publisher

Subscription

Subscriber

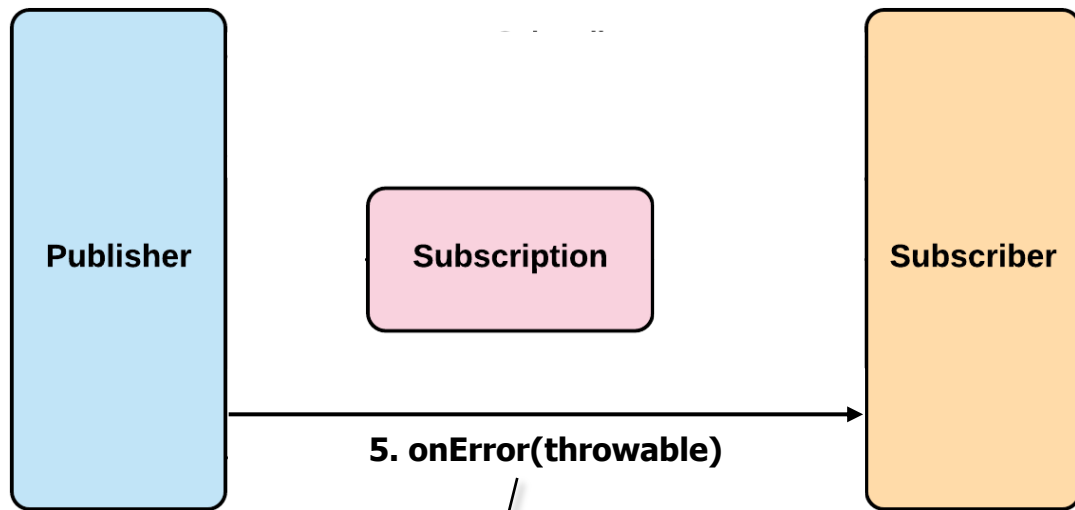**5. onComplete()**

SUCCESS!

Hook method called by publisher when all events have been sent successfully

Key Abstractions & Interactions in the Java Flow API

- A "flow" involves interactions between three key abstractions

Publisher

Subscription

Subscriber

5. onError(throwable)

FAILURE

Hook method called by a publisher when an error occurs to convey the exception

End of Overview of the Java Reactive Streams API (Part 2)