

# Implementing the RSocket Quotes Backpressure Case Study App Server (Part 2)

**Douglas C. Schmidt**

**[d.schmidt@vanderbilt.edu](mailto:d.schmidt@vanderbilt.edu)**

**[www.dre.vanderbilt.edu/~schmidt](http://www.dre.vanderbilt.edu/~schmidt)**

**Professor of Computer Science**

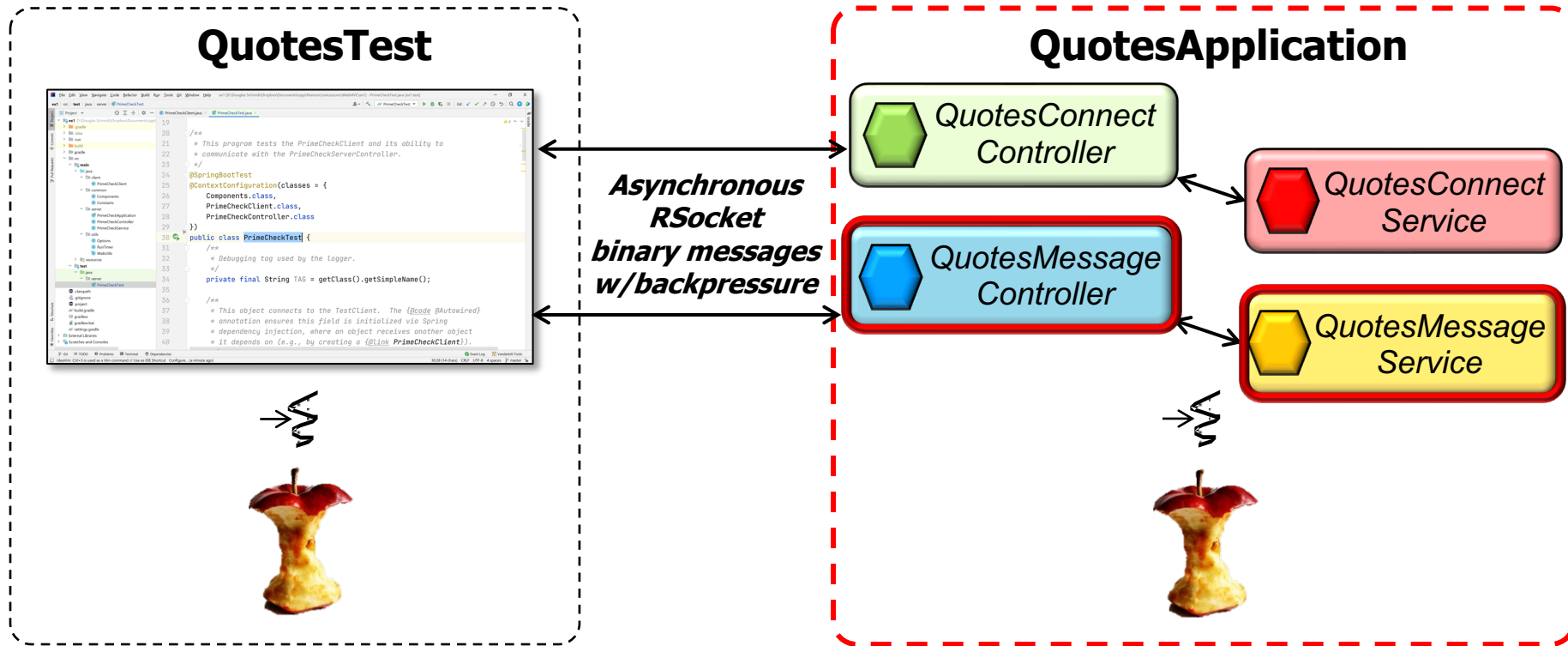
**Institute for Software  
Integrated Systems**

**Vanderbilt University  
Nashville, Tennessee, USA**



# Learning Objectives in this Lesson

- Understand the implementation of the messaging portion of the RSocket Quotes backpressure app server



---

# Implementing the RSocket Quotes Backpressure App Server

# Implementing the RSocket Quotes Backpressure App Server

```
ex2 | src | main | java | quotes | server | QuotesMessageController
QuotesMessageController.java
51 @Controller
52 public class QuotesMessageController {
53     /**
54      * The {@link QuotesMessageService} that's associated with this {@link
55      * QuotesMessageController} via Spring's dependency injection facilities,
56      * where an object receives other objects that it depends on (in
57      * this case, the {@link QuotesMessageService}).
58      */
59     @Autowired
60     private QuotesMessageService mService;
61
62     /**
63      * This method must be called before attempting to receive a Flux
64      * stream of Zippy quotes. It implements a two-way async RSocket
65      * request/response call that sends a response back to the client.
66      *
67      * @param subscriptionRequest A {@link Mono} that emits a {@link
```

See [github.com/douglascraigschmidt/LiveLessons/tree/master/RSocket/ex2](https://github.com/douglascraigschmidt/LiveLessons/tree/master/RSocket/ex2)

---

# End of Implementing the RSocket Quotes Case Study Backpressure App Server (Part 2)