The LockManager App Case Study: Implementing the Server Components

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

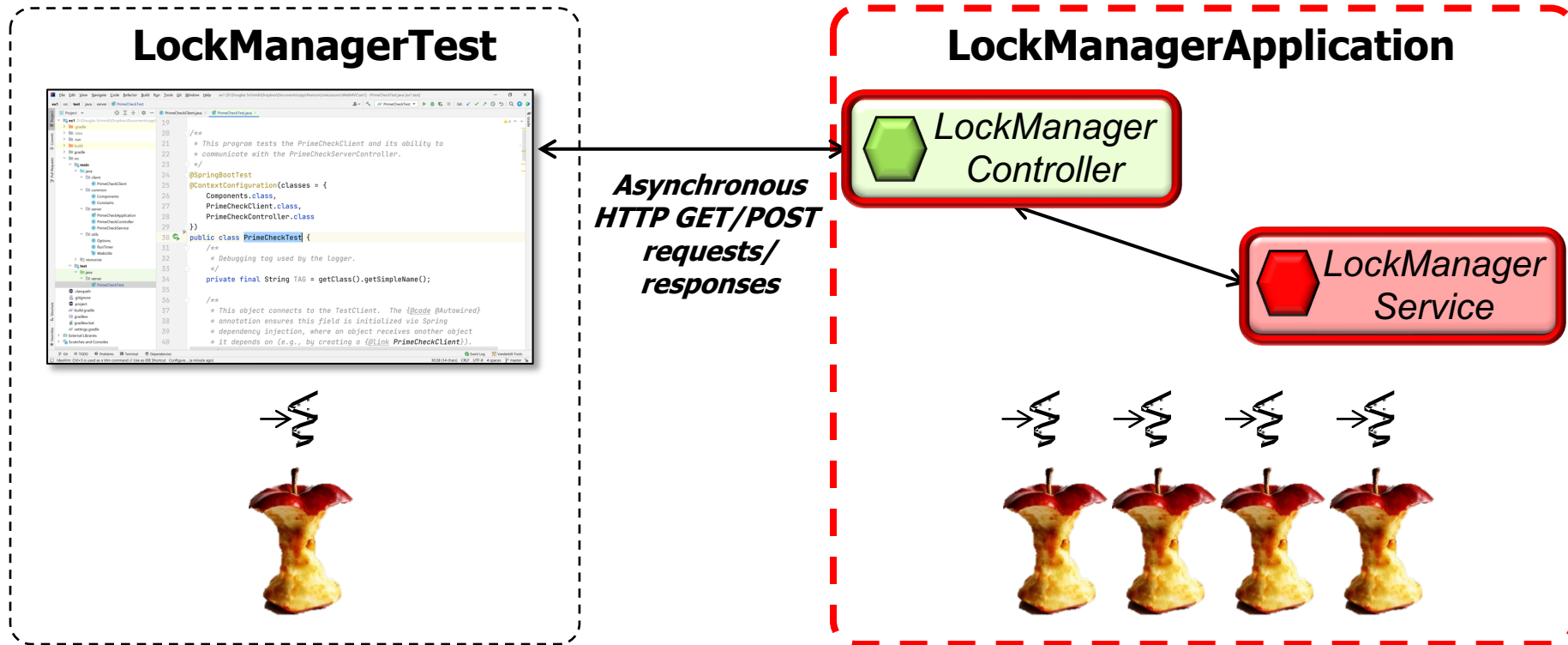Professor of Computer Science

Institute for Software
Integrated Systems

Vanderbilt University
Nashville, Tennessee, USA

Learning Objectives in this Part of the Lesson

- Understand the WebFlux implemention of the LockManagerController & Lock ManagerService classes in the reactive LockManagerApplication microservice

LockManagerTest

LockManagerApplication

LockManager Controller

LockManager Service

Asynchronous HTTP GET/POST requests/ responses

See github.com/douglascraigschmidt/LiveLessons/tree/master/WebFlux/ex1

Implementing the LockManager App Server

Implementing the LockManager App Server

LockManagerService.java

```java
21   @Service
22   public class LockManagerService {
23       /**
24        * An {@link ArrayBlockingQueue} that limits concurrent access to
25        * the fixed number of available locks managed by the {@link
26        * LockManagerService}.
27        */
28       private ArrayBlockingQueue<Lock> mAvailableLocks;
29
30       /**
31        * Initialize the {@link Lock} manager.
32        *
33        * @param permitCount The number of {@link Lock} objects to manage
34        * @return A {@link Mono} that emits {@link Boolean#TRUE} if
35        * the {@code permitCount} changed the state of the
36        * lock manager and {@link Boolean#FALSE} otherwise.
37        */
```

See WebFlux/ex1/src/main/java/edu/vandy/lockmanager/server

End of the LockManager App Case Study: Implementing the Server Components