

# The BerraQuotes App Case Study: Overview

**Douglas C. Schmidt**

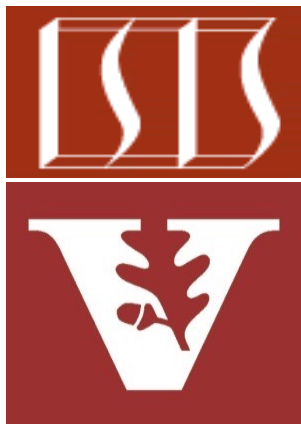
**[d.schmidt@vanderbilt.edu](mailto:d.schmidt@vanderbilt.edu)**

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

**Professor of Computer Science**

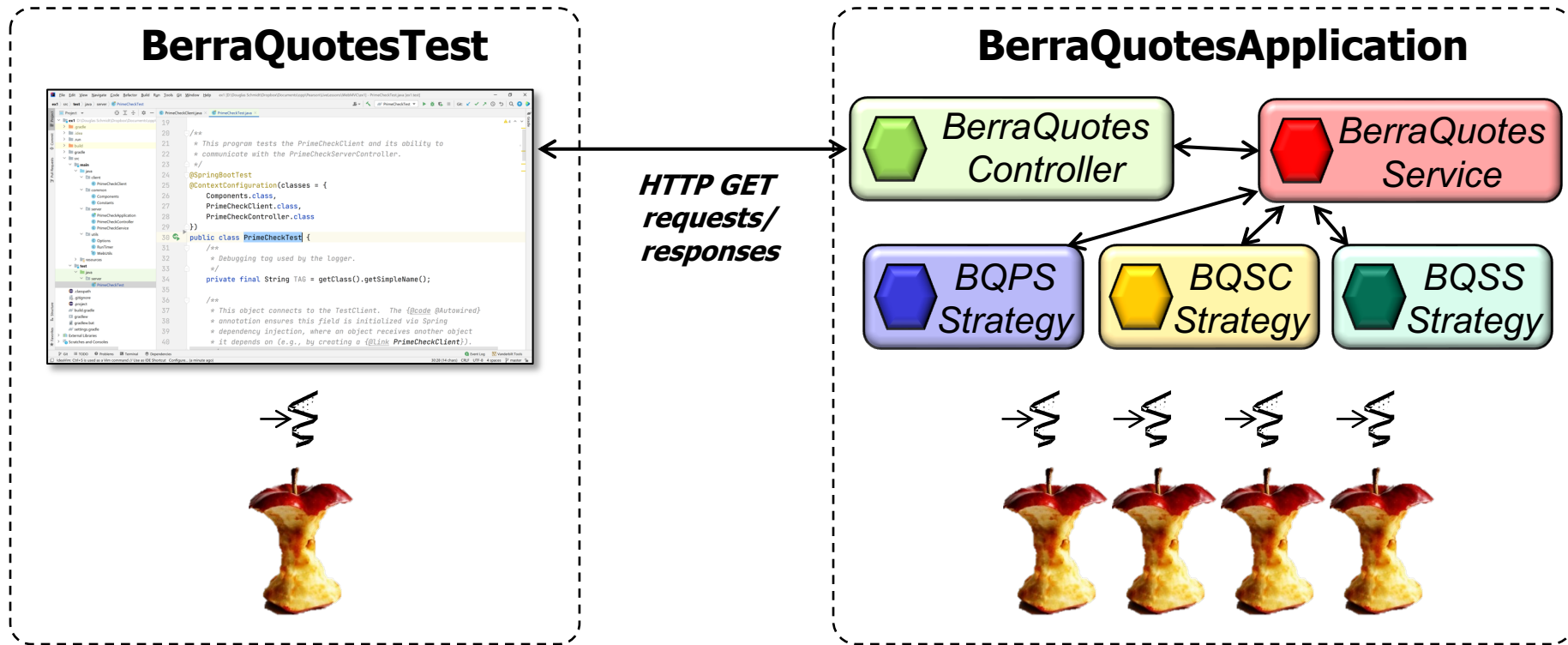
**Institute for Software  
Integrated Systems**

**Vanderbilt University  
Nashville, Tennessee, USA**



# Learning Objectives in this Part of the Lesson

- Understand how functional programming & various Java frameworks are applied in a case study using Spring WebMVC to obtain Yogi Berra quotes



See [github.com/douglasraigschmidt/LiveLessons/tree/master/WebMVC/ex2](https://github.com/douglasraigschmidt/LiveLessons/tree/master/WebMVC/ex2)

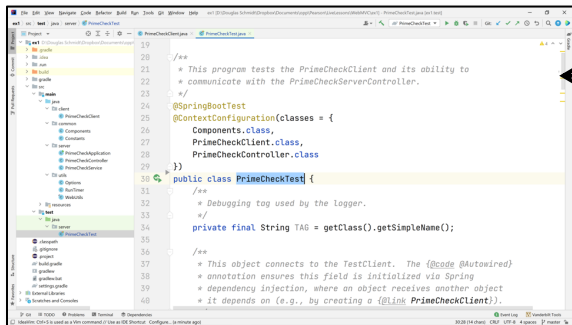
---

# Overview of the Berra Quotes App Case Study

# Overview of the BerraQuotes App Case Study

- This case study shows how Spring WebMVC can be used to send & receive HTTP GET requests to/from several microservice implementation strategies

## BerraQuotesTest

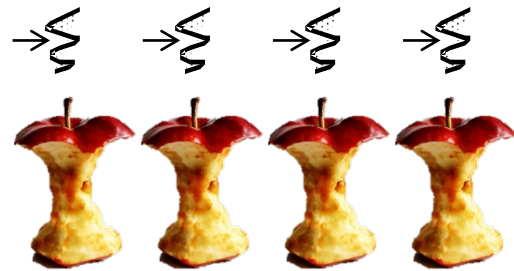
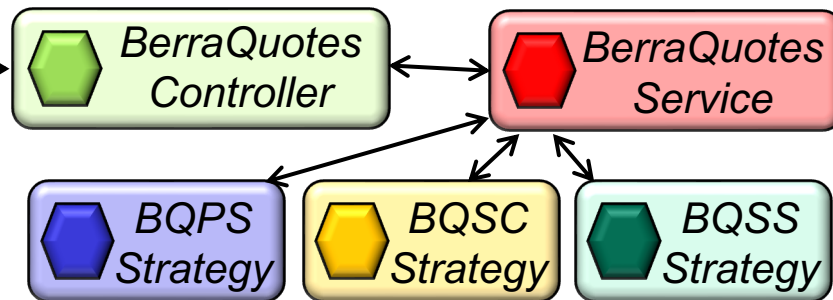


```
20 /**  
21  * This program tests the PrimeCheckClient and its ability to  
22  * communicate with the PrimeCheckServerController.  
23  */  
24 @SpringBootTest  
25 @ContextConfiguration(classes = {  
26     Components.class,  
27     PrimeCheckClient.class,  
28     PrimeCheckController.class  
29 })  
30 public class PrimeCheckTest {  
31     /**  
32      * Debugging tag used by the logger.  
33      */  
34     private final String TAG = getClass().getSimpleName();  
35   
36     /**  
37      * This object connects to the TestClient. The @Autowired  
38      * annotation ensures this field is initialized via Spring  
39      * dependency injection, where an object receives another object  
40      * it depends on (e.g., by creating a @Link PrimeCheckClient).
```



*HTTP GET  
requests/  
responses*

## BerraQuotesApplication



# Overview of the BerraQuotes App Case Study

- This case study shows how Spring WebMVC can be used to send & receive HTTP GET requests to/from several microservice implementation strategies

## BerraQuotesTest

```
19 //**
20 // *
21 // * This program tests the PrimeCheckClient and its ability to
22 // * communicate with the PrimeCheckServerController.
23 // */
24 @SpringBootTest
25 @ContextConfiguration(classes = {
26     Components.class,
27     PrimeCheckClient.class,
28     PrimeCheckController.class
29 })
30 public class PrimeCheckTest {
31     /**
32     * Debugging tag used by the logger.
33     */
34     private final String TAG = getClass().getSimpleName();
35
36
37     /**
38     * This object connects to the TestClient. The @Autowired
39     * annotation ensures this field is initialized via Spring
40     * dependency injection, where an object receives another object
41     * it depends on (e.g., by creating a @Link PrimeCheckClient).
42     */
43     @Autowired
44     PrimeCheckClient testClient;
```

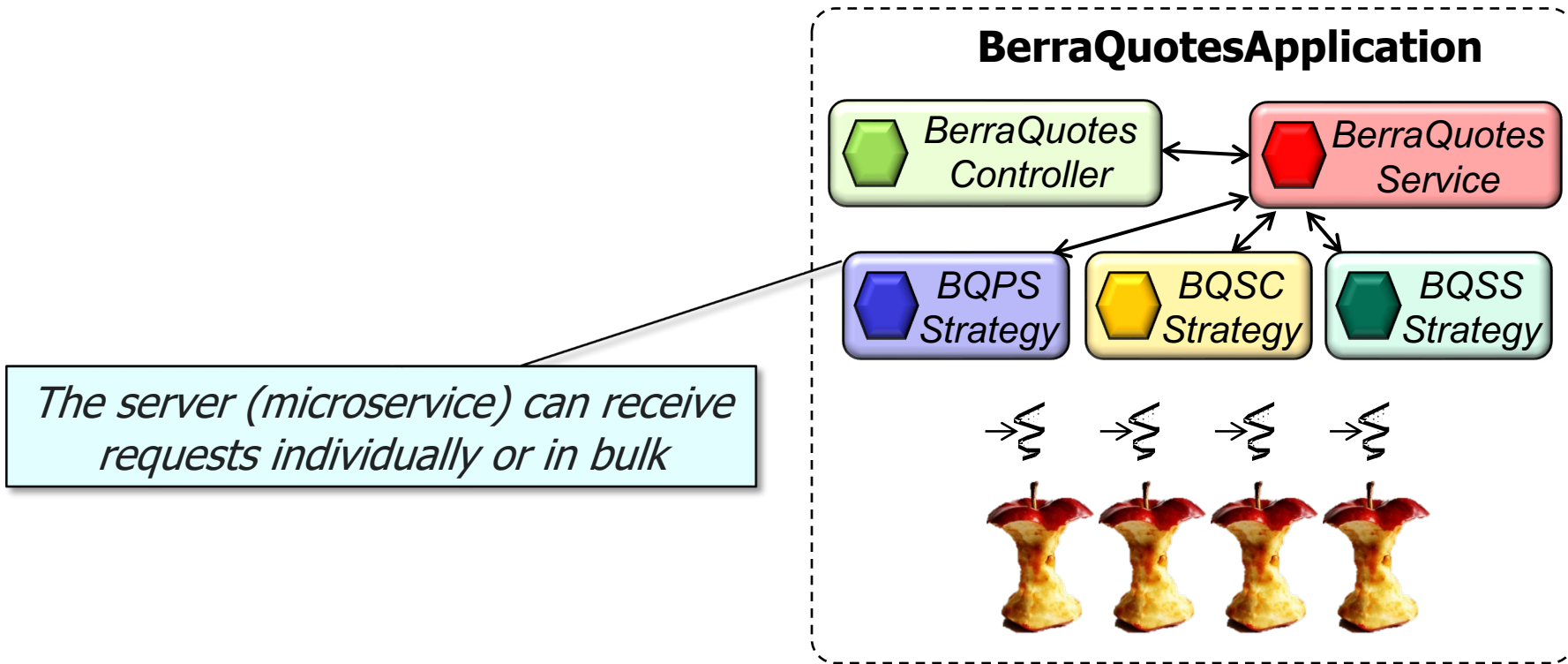
*The client can send requests individually or in bulk*



See [WebMVC/ex2/src/test/java/berraquotes/client](https://github.com/leandrotavares/berraquotes/blob/master/src/test/java/berraquotes/client/WebMVC/ex2/src/test/java/berraquotes/client/PrimeCheckTest.java)

# Overview of the BerraQuotes App Case Study

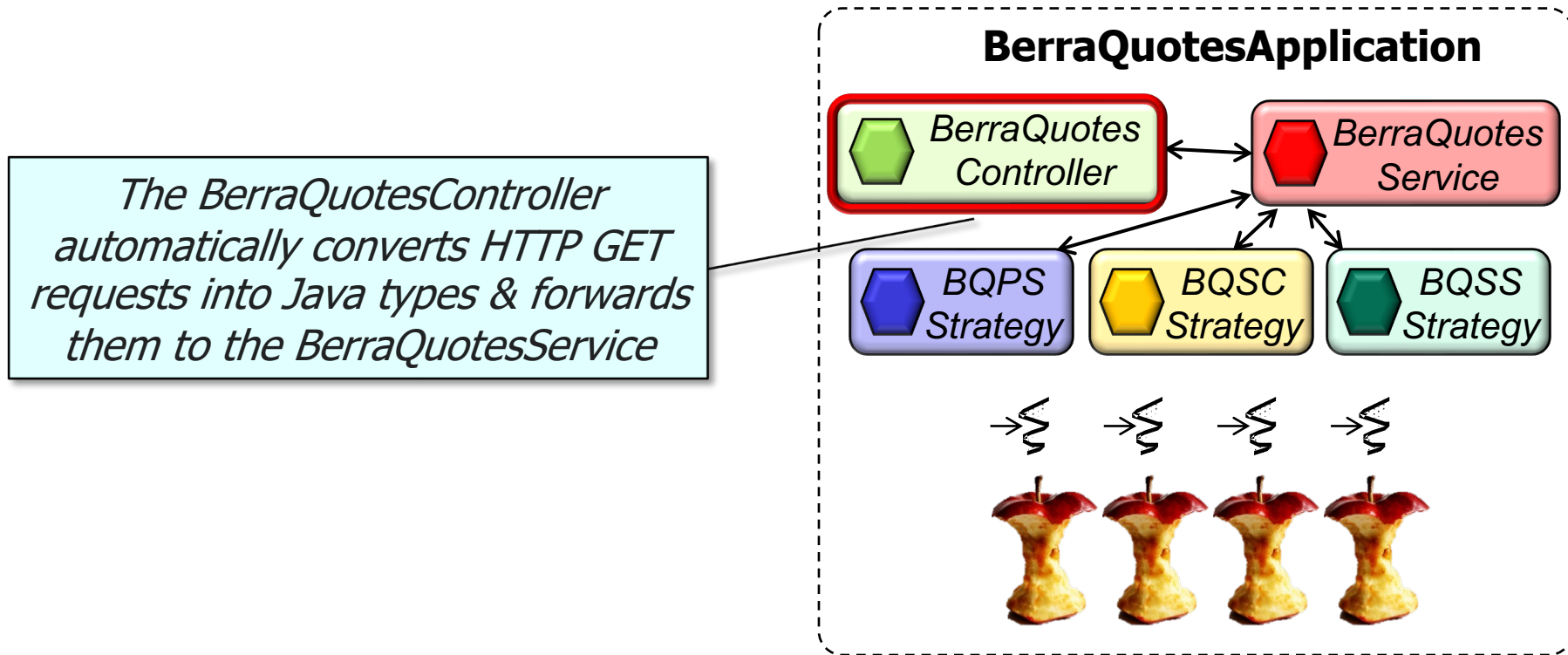
- This case study shows how Spring WebMVC can be used to send & receive HTTP GET requests to/from several microservice implementation strategies



See [WebMVC/ex2/src/main/java/server](http://WebMVC/ex2/src/main/java/server)

# Overview of the BerraQuotes App Case Study

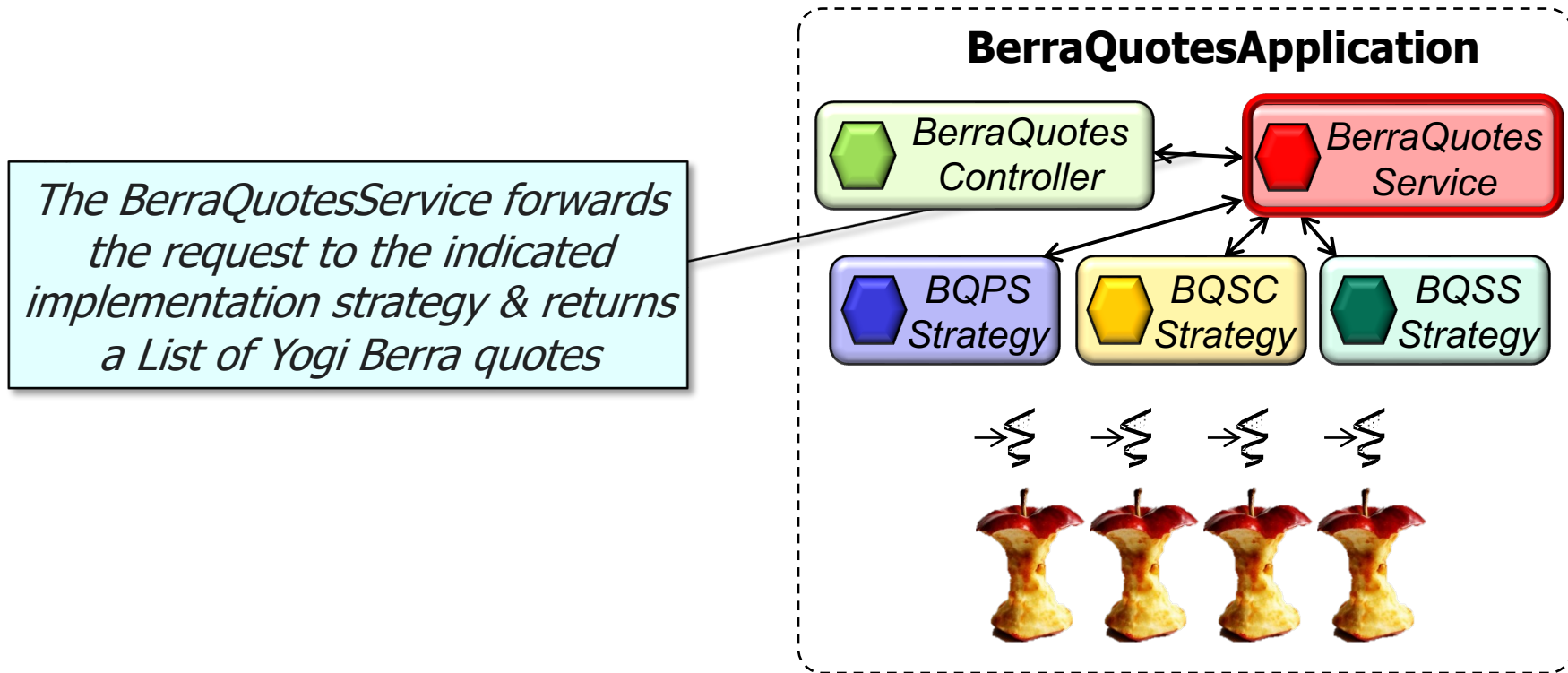
- This case study shows how Spring WebMVC can be used to send & receive HTTP GET requests to/from several microservice implementation strategies



See [WebMVC/ex2/src/main/java/berraquotes/server/BerraQuotesController.java](#)

# Overview of the BerraQuotes App Case Study

- This case study shows how Spring WebMVC can be used to send & receive HTTP GET requests to/from several microservice implementation strategies

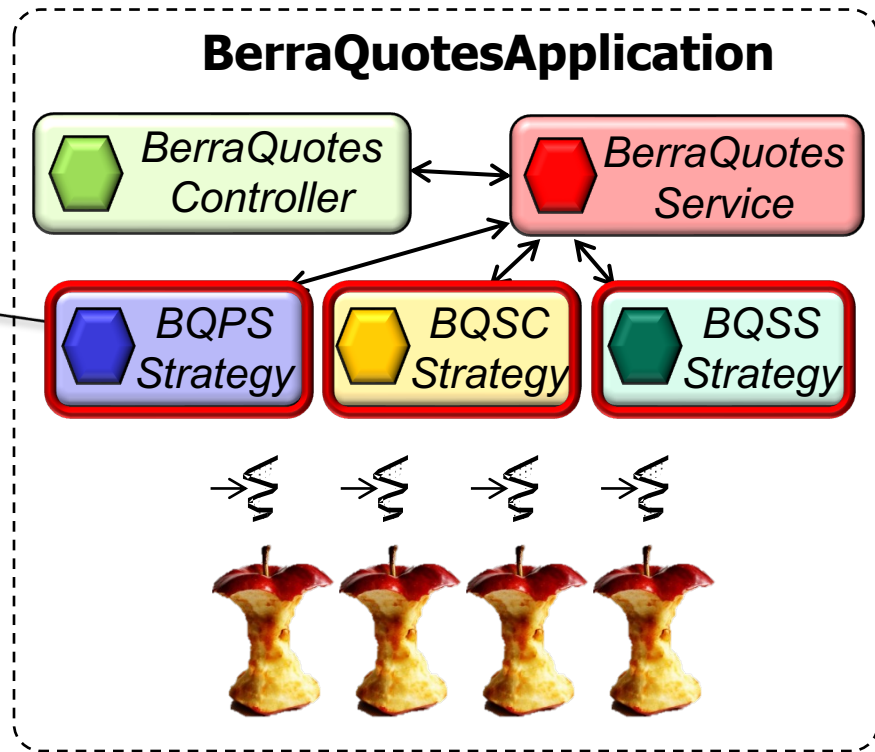


See [WebMVC/ex2/src/main/java/berraquotes/server/BerraQuotesService.java](http://WebMVC/ex2/src/main/java/berraquotes/server/BerraQuotesService.java)



# Overview of the BerraQuotes App Case Study

- This case study shows how Spring WebMVC can be used to send & receive HTTP GET requests to/from several microservice implementation strategies



*The implementation strategy indicated in the request then performs the request*

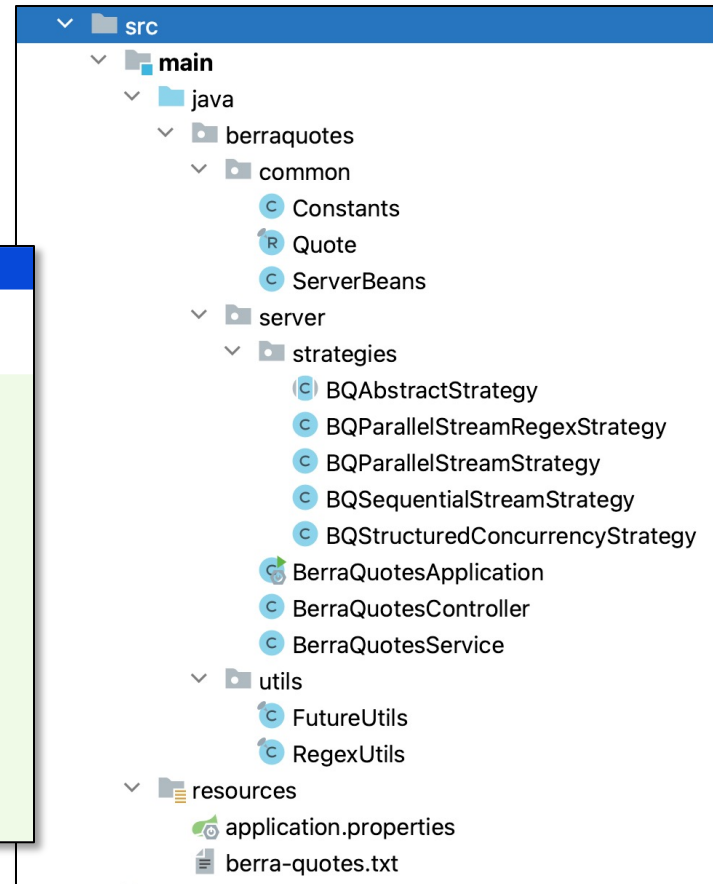
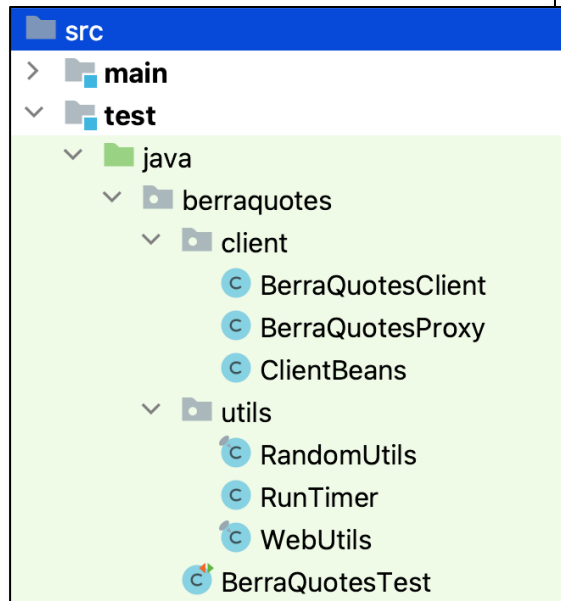
See [WebMVC/ex2/src/main/java/berraquotes/server/strategies](http://WebMVC/ex2/src/main/java/berraquotes/server/strategies)

---

# Structure of the BerraQuotes App Project

# Structure of the BerraQuotes App Project

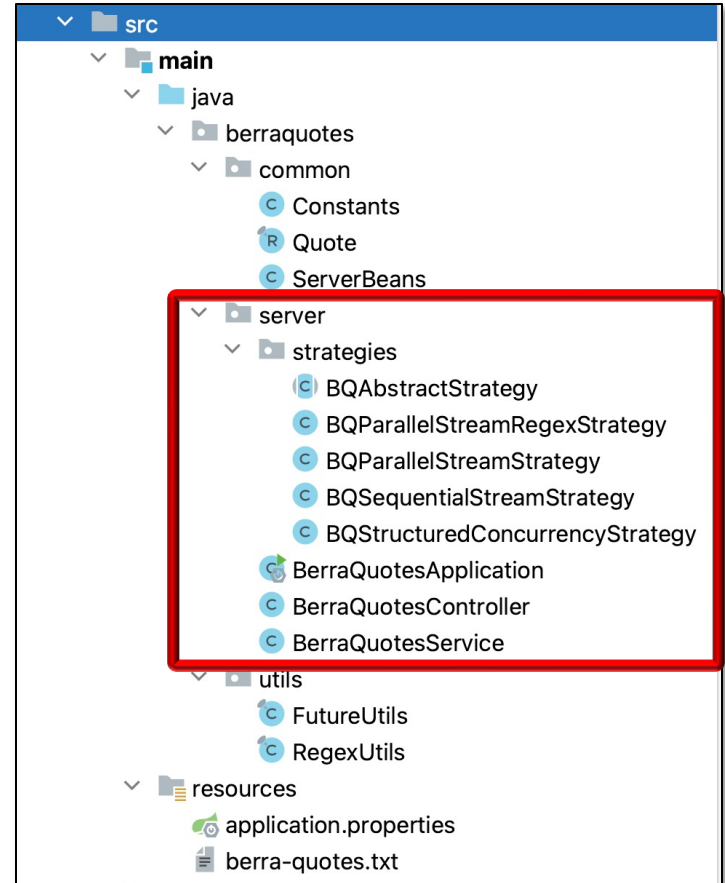
- The BerraQuotes App project source code is organized into several packages



See [github.com/douglasraigschmidt/LiveLessons/tree/master/WebMVC/ex2](https://github.com/douglasraigschmidt/LiveLessons/tree/master/WebMVC/ex2)

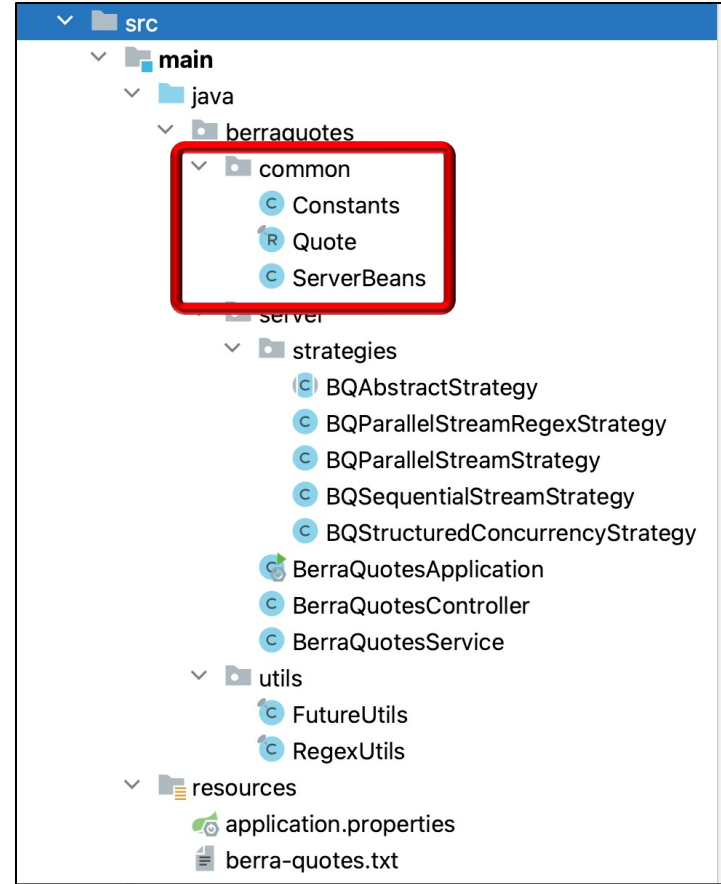
# Structure of the BerraQuotes App Project

- The BerraQuotes App project source code is organized into several packages
  - main
    - berraquotes
      - Contains the “app” entry point, the controller, the service, & the implementation strategies



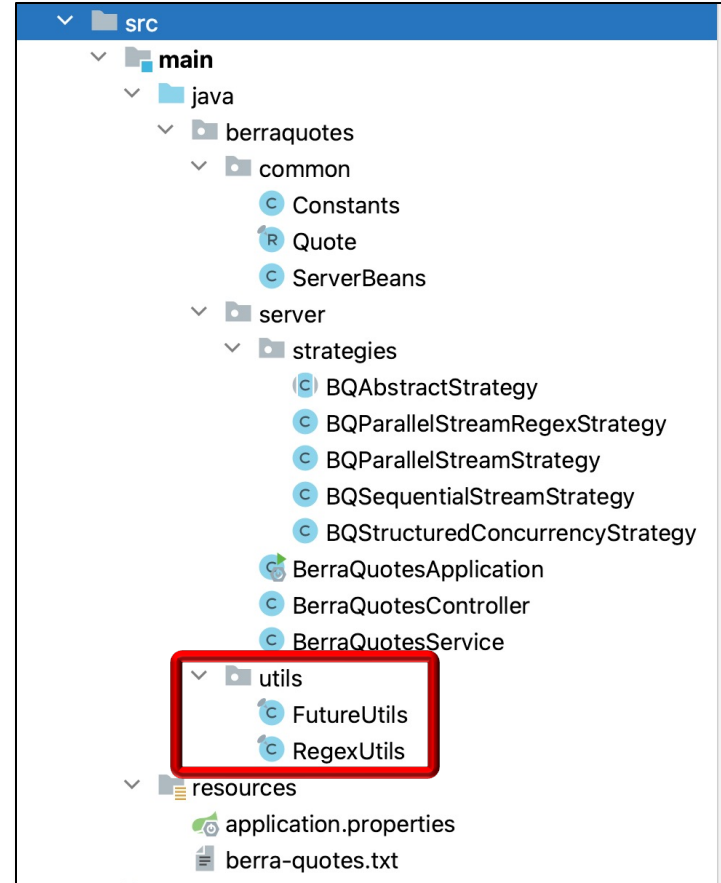
# Structure of the BerraQuotes App Project

- The BerraQuotes App project source code is organized into several packages
  - main
    - berraquotes
      - Contains the “app” entry point, the controller, the service, & the implementation strategies
      - Consolidates various project-specific helper classes & the model



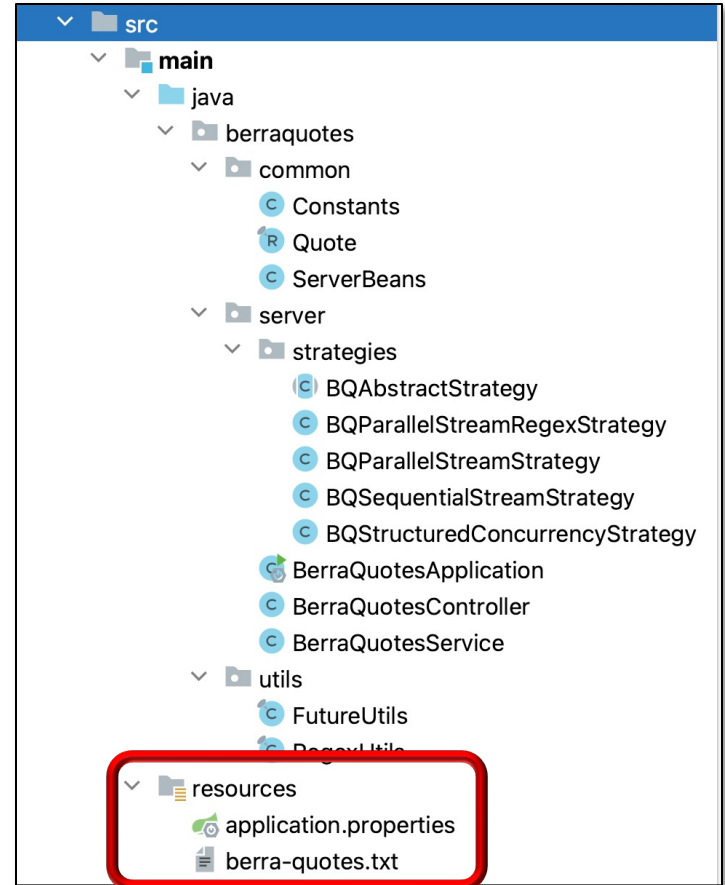
# Structure of the BerraQuotes App Project

- The BerraQuotes App project source code is organized into several packages
  - main
    - berraquotes
      - Contains the “app” entry point, the controller, the service, & the implementation strategies
      - Consolidates various project-specific helper classes & the model
    - General-purpose utilities



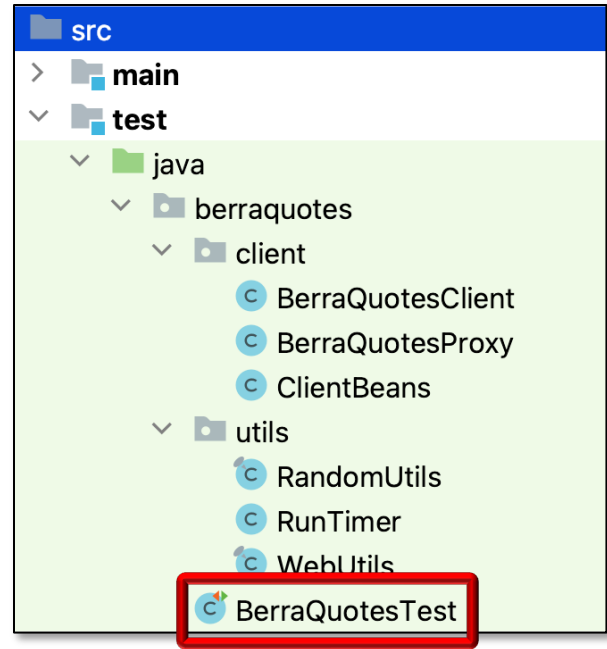
# Structure of the BerraQuotes App Project

- The BerraQuotes App project source code is organized into several packages
  - main
    - berraquotes
    - resources
  - Defines various application properties
    - e.g., name & port number
  - Also contains the list of Berra quotes



# Structure of the BerraQuotes App Project

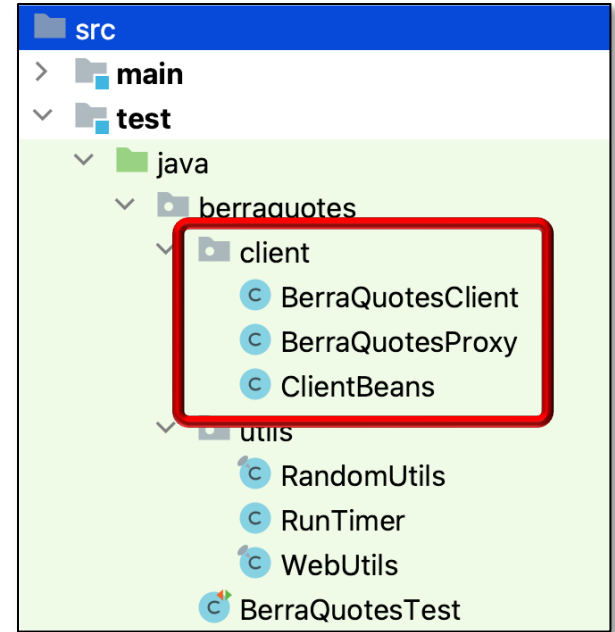
- The BerraQuotes App project source code is organized into several packages
  - test
    - BerraQuotesTest
      - This test driver initiates calls to the BerraQuotes microservice





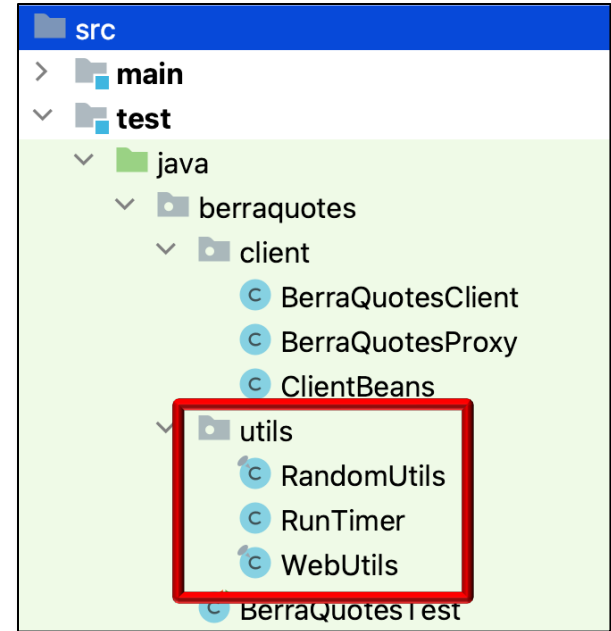
# Structure of the BerraQuotes App Project

- The BerraQuotes App project source code is organized into several packages
  - test
    - BerraQuotesTest
  - client
    - Sends HTTP GET requests to the BerraQuotes microservice



# Structure of the BerraQuotes App Project

- The BerraQuotes App project source code is organized into several packages
  - test
    - BerraQuotesTest
    - client
    - utils
      - Consolidates various reusable helper classes



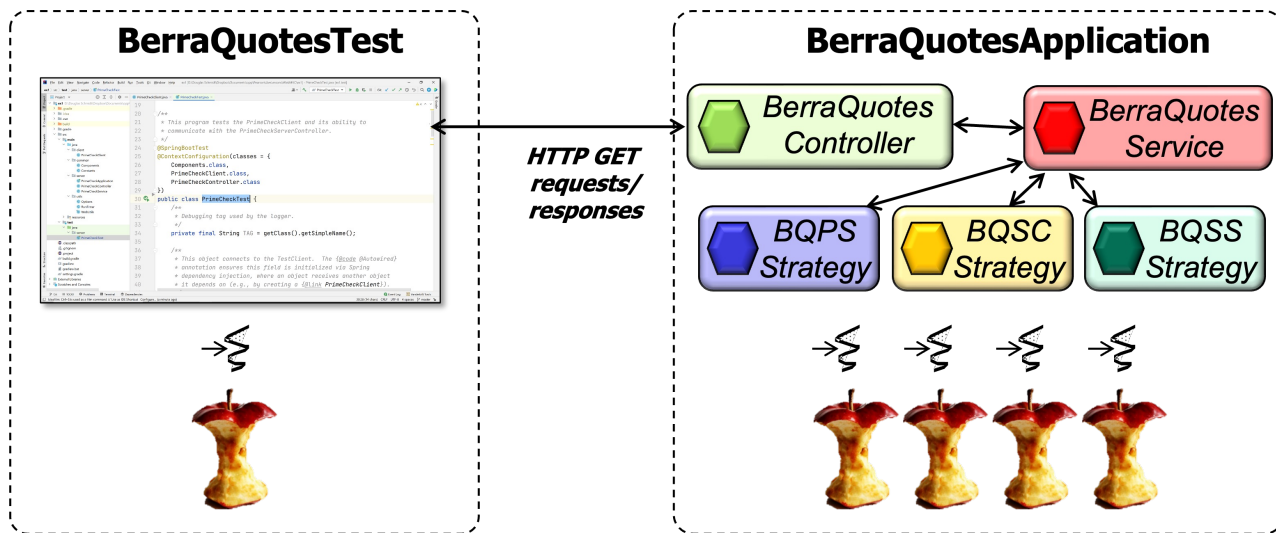
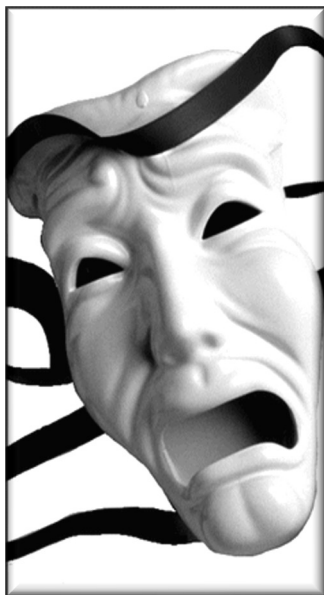
---

# Pros & Cons of the BerraQuotes App



# Pros & Cons of the BerraQuotes App

- Cons
  - All service implementations run in a single process, which can degrade system scalability (i.e., there's a single point of contention) & reliability (i.e., there's a single point of failure)



---

# End of the BerraQuotes App Case Study: Overview