Overview of the RSocket Shakespeare Quotes App

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

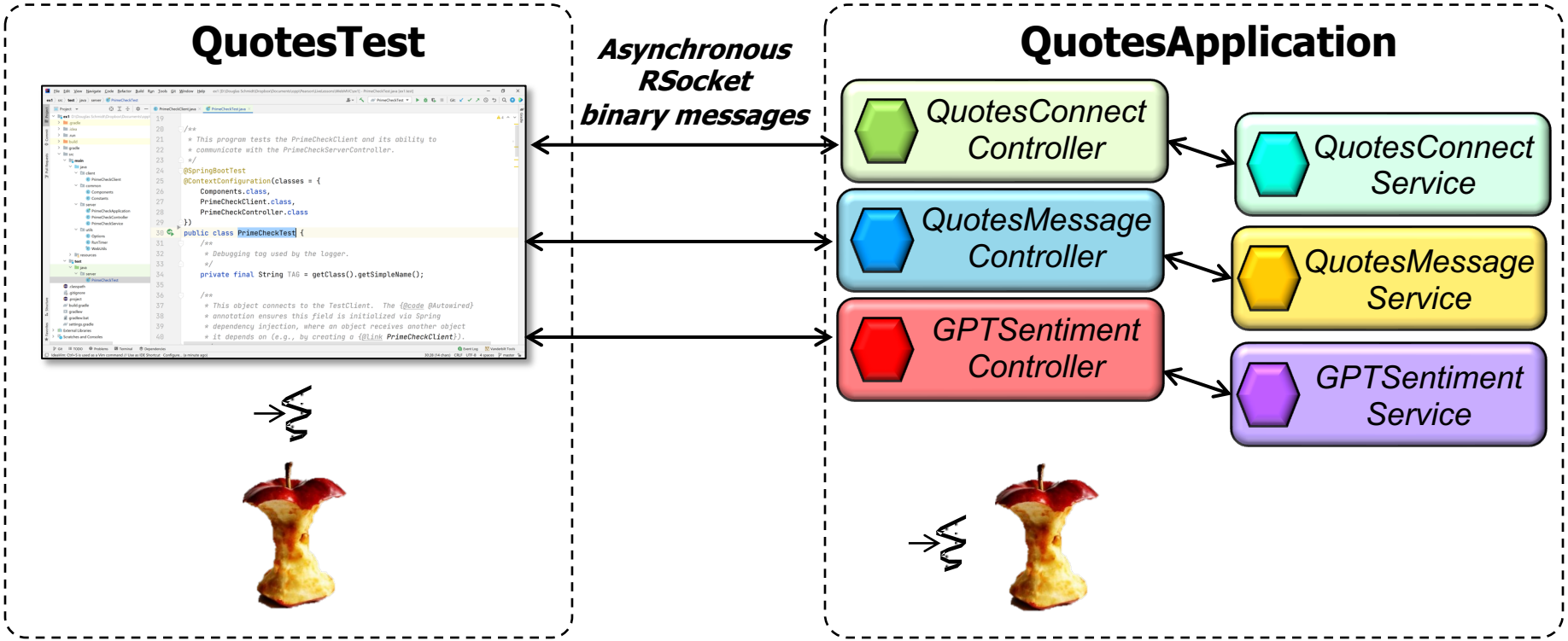
Professor of Computer Science

Institute for Software Integrated Systems

Vanderbilt University
Nashville, Tennessee, USA

Learning Objectives in this Lesson

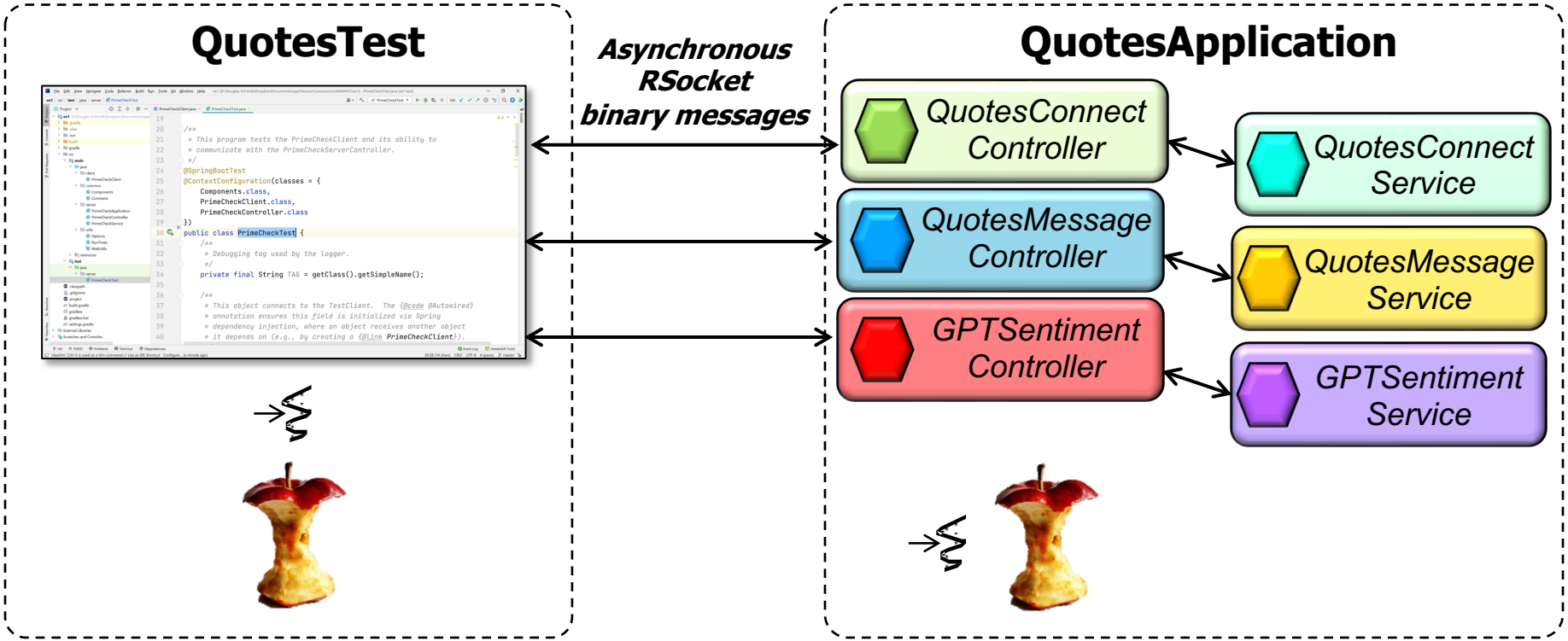
- Understand how Spring RSocket & ChatGPT can be used to analyze the sentiment of famous Shakespeare quotes asynchronously



See github.com/douglascraigschmidt/LiveLessons/tree/master/RSocket/ex3

Learning Objectives in this Lesson

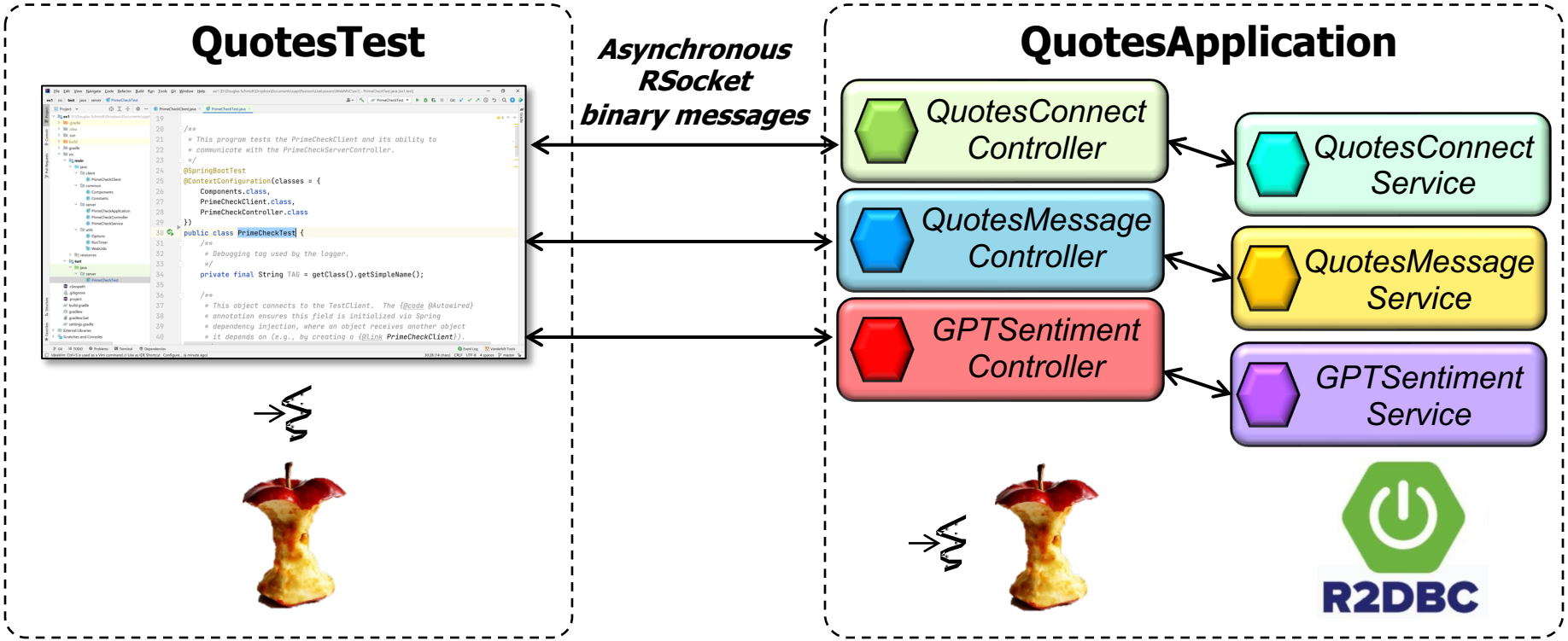
- Understand how Spring RSocket & ChatGPT can be used to analyze the sentiment of famous Shakespeare quotes asynchronously



This example also shows how to use RSocket security & authentication features

Learning Objectives in this Lesson

- Understand how Spring RSocket & ChatGPT can be used to analyze the sentiment of famous Shakespeare quotes asynchronously

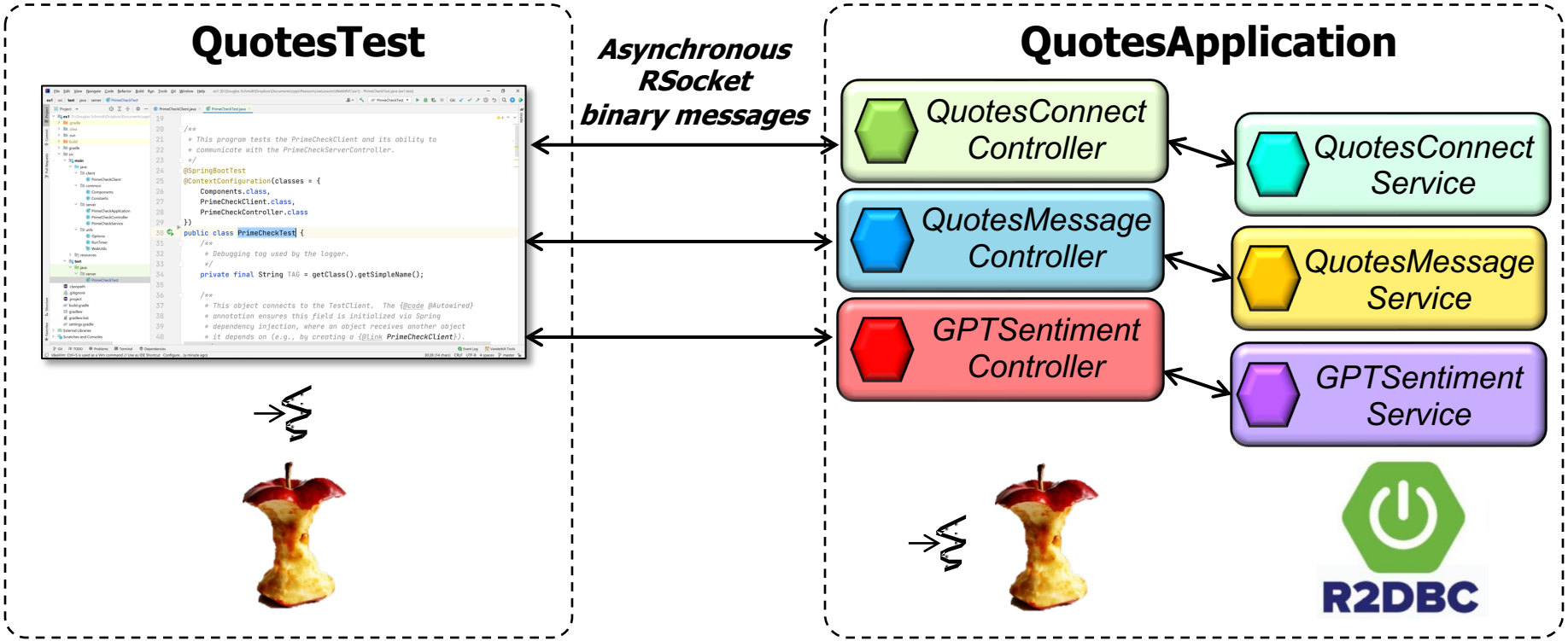


As well as the R2DBC reactive database used to retrieve Shakespeare Quotes

Overview of the RSocket Shakespeare Quotes App

Overview of the RSocket Shakespeare Quotes App

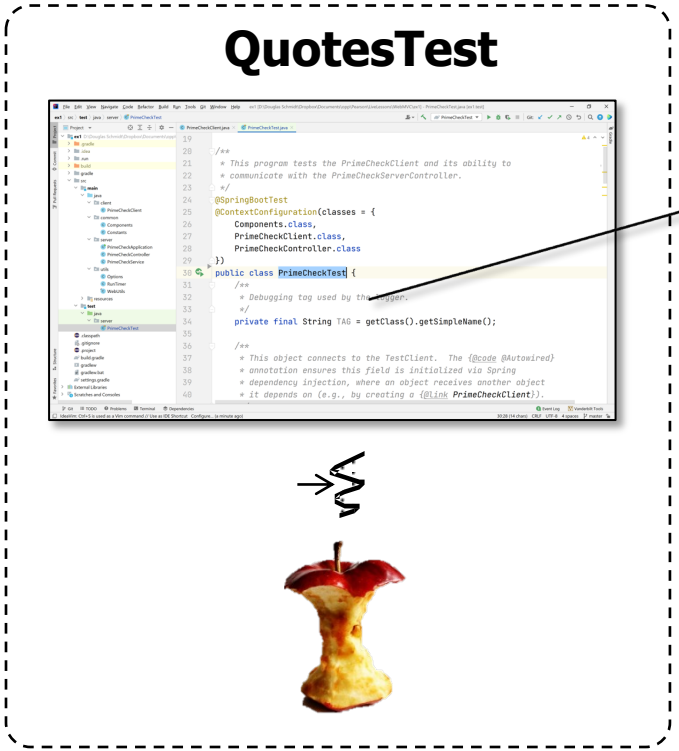
- This case study shows how an RSocket requester can exchange binary messages asynchronously with various controllers & services



See github.com/douglascraigschmidt/LiveLessons/tree/master/RSocket/ex3

Overview of the RSocket Shakespeare Quotes App

- This case study shows how an RSocket requester can exchange binary messages asynchronously with various controllers & services

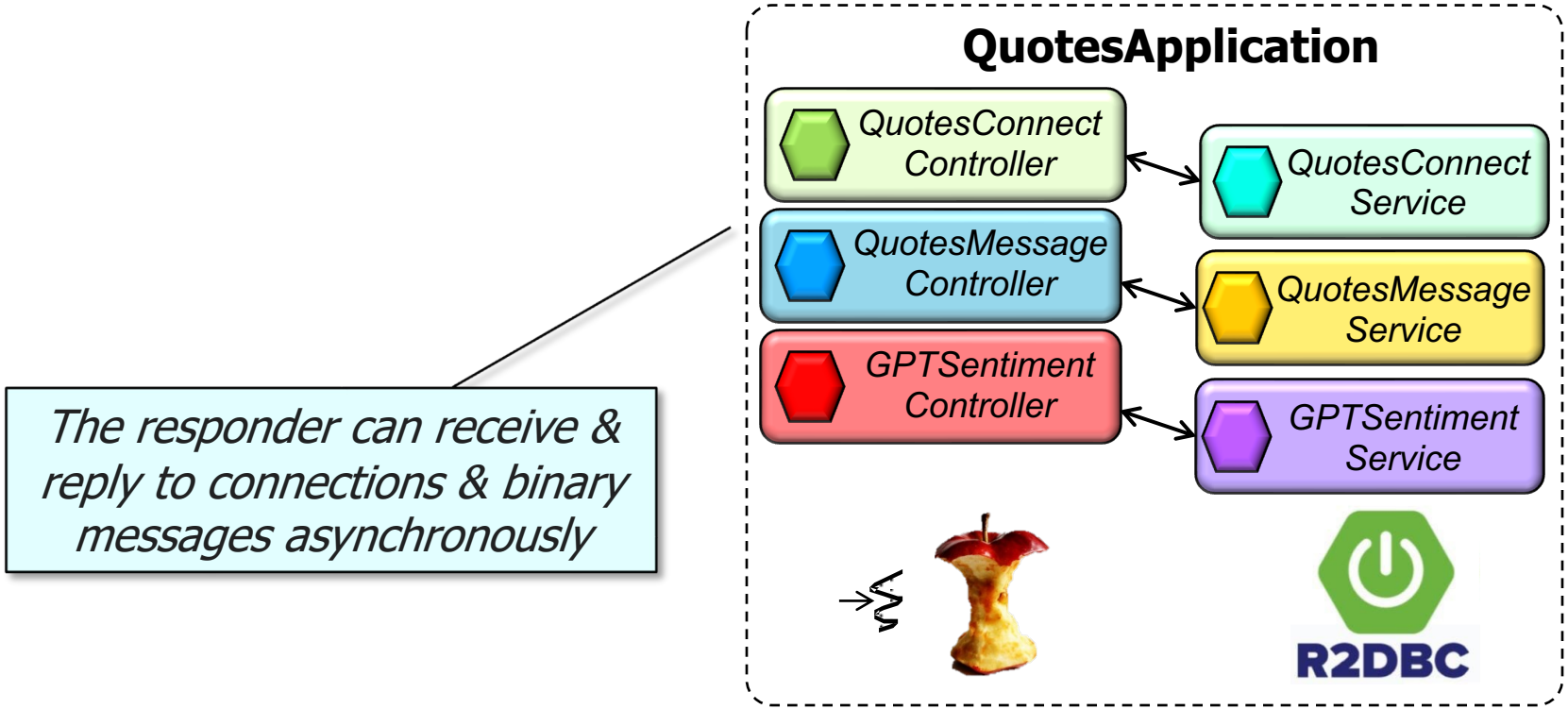


The requester can asynchronously connect to the responder, subscribe to receive shakespeare quotes, & then use ChatGPT to analyze the sentiment of these quotes

The requester also authenticates itself to the responder

Overview of the RSocket Shakespeare Quotes App

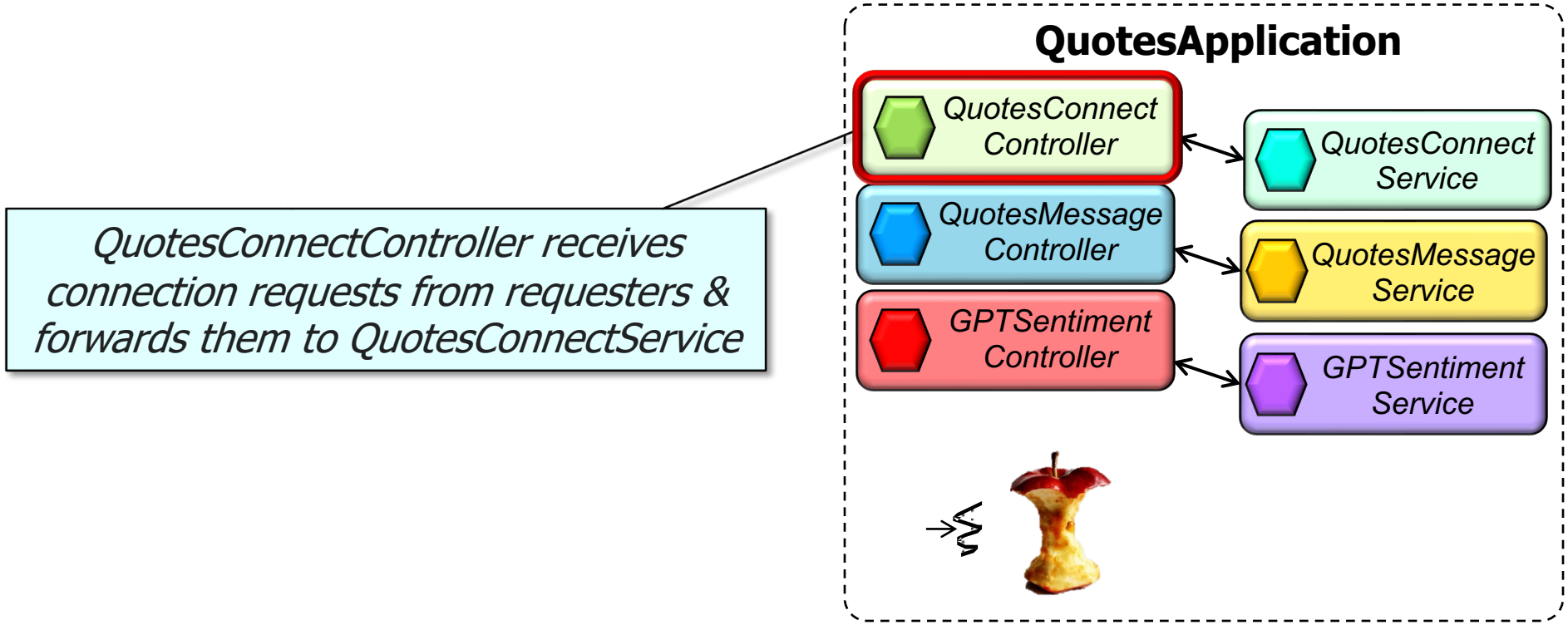
- This case study shows how an RSocket requester can exchange binary messages asynchronously with various controllers & services



The responder requires the requester to authentic before finalizing the connection

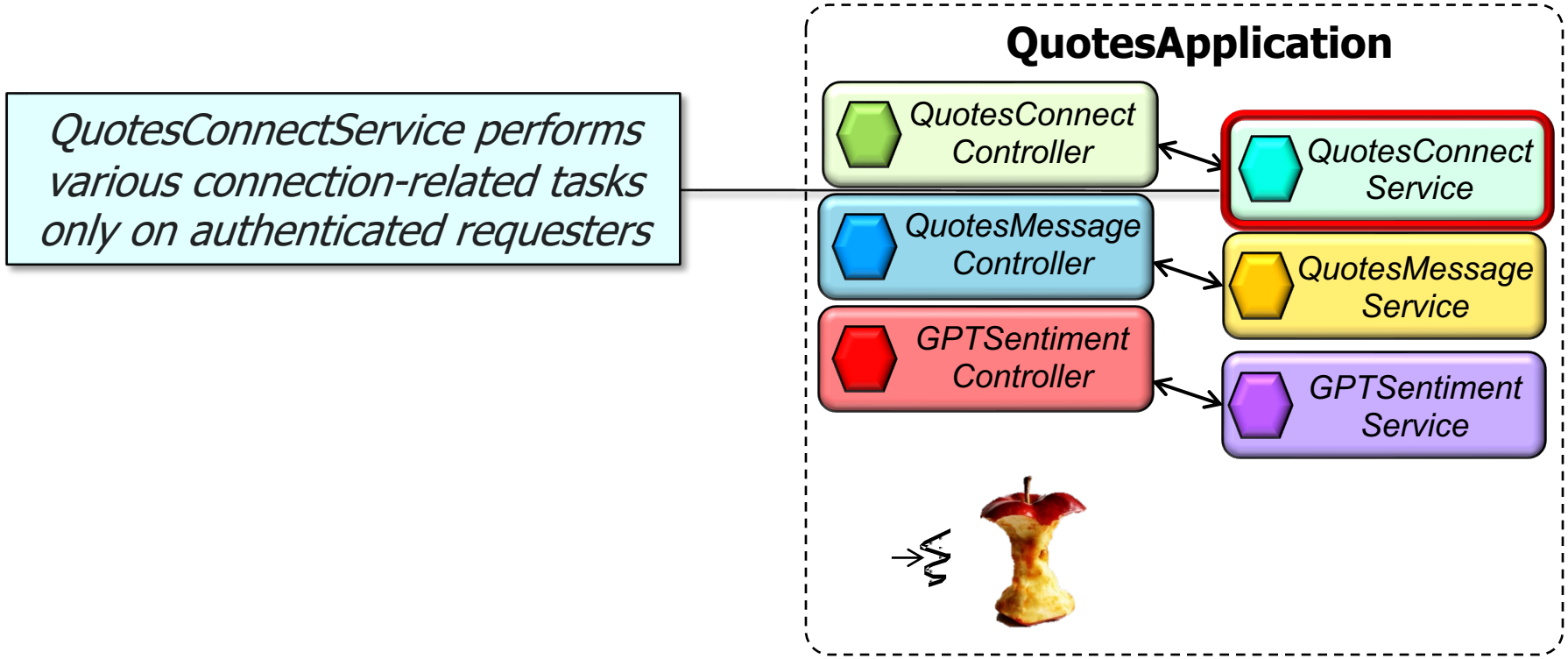
Overview of the RSocket Shakespeare Quotes App

- This case study shows how an RSocket requester can exchange binary messages asynchronously with various controllers & services



Overview of the RSocket Shakespeare Quotes App

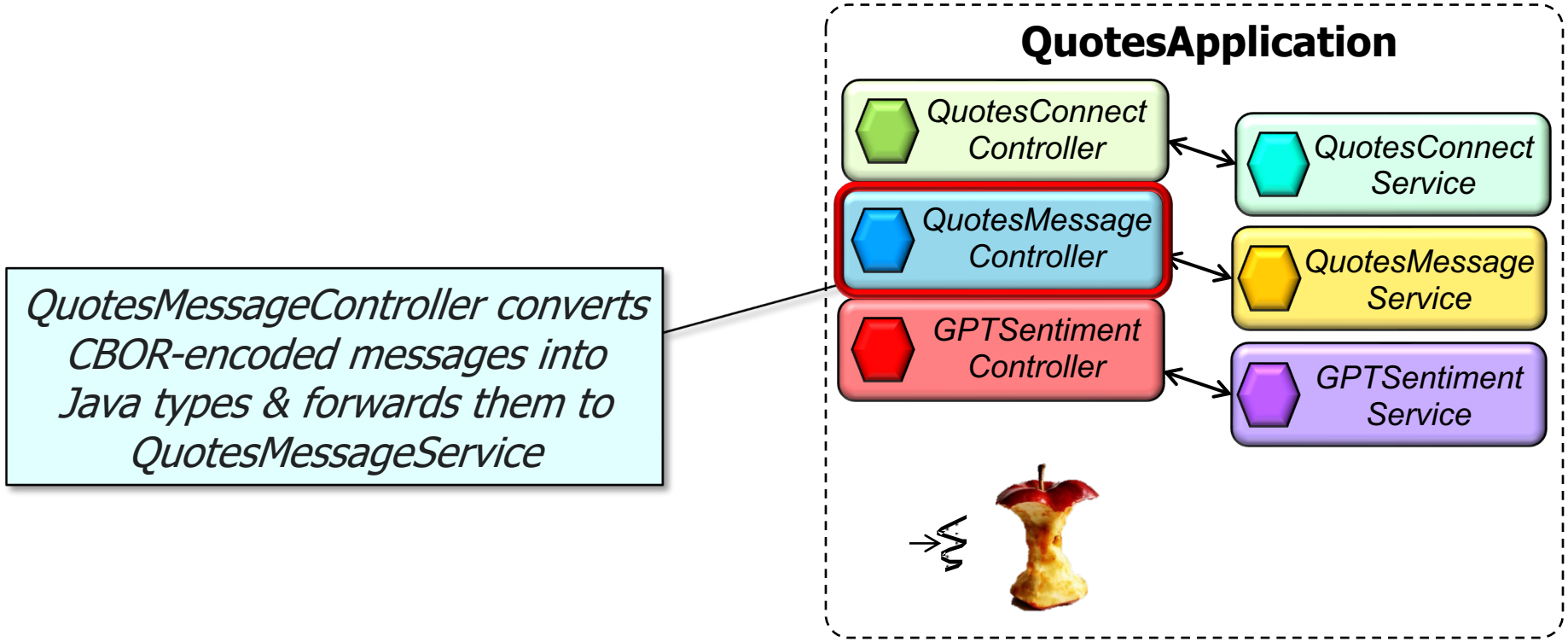
- This case study shows how an RSocket requester can exchange binary messages asynchronously with various controllers & services



RSocket & Spring perform authentication transparently to the responder itself

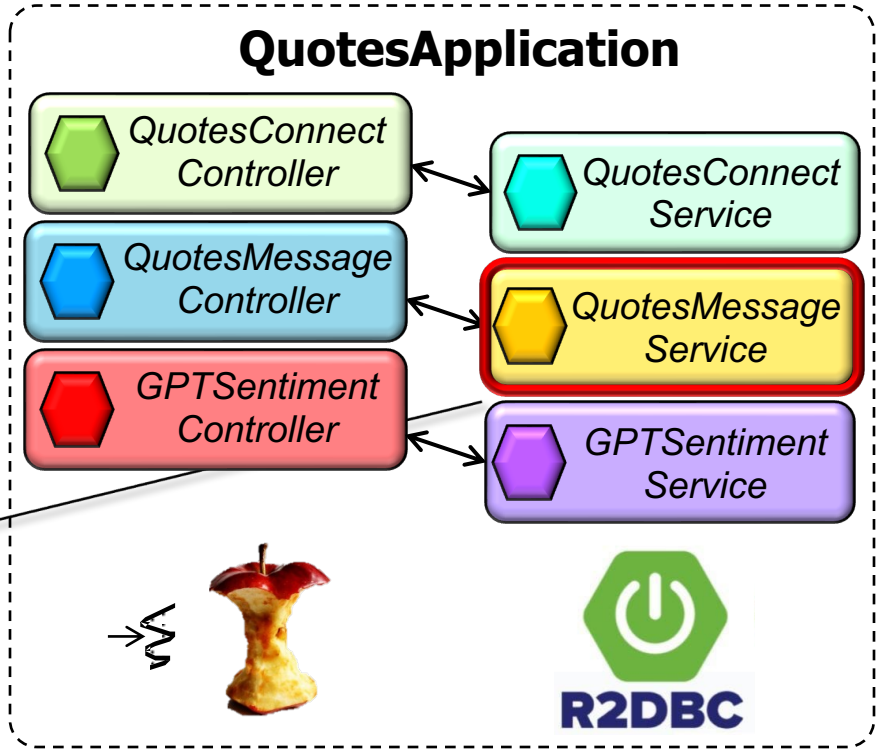
Overview of the RSocket Shakespeare Quotes App

- This case study shows how an RSocket requester can exchange binary messages asynchronously with various controllers & services



Overview of the RSocket Shakespeare Quotes App

- This case study shows how an RSocket requester can exchange binary messages asynchronously with various controllers & services

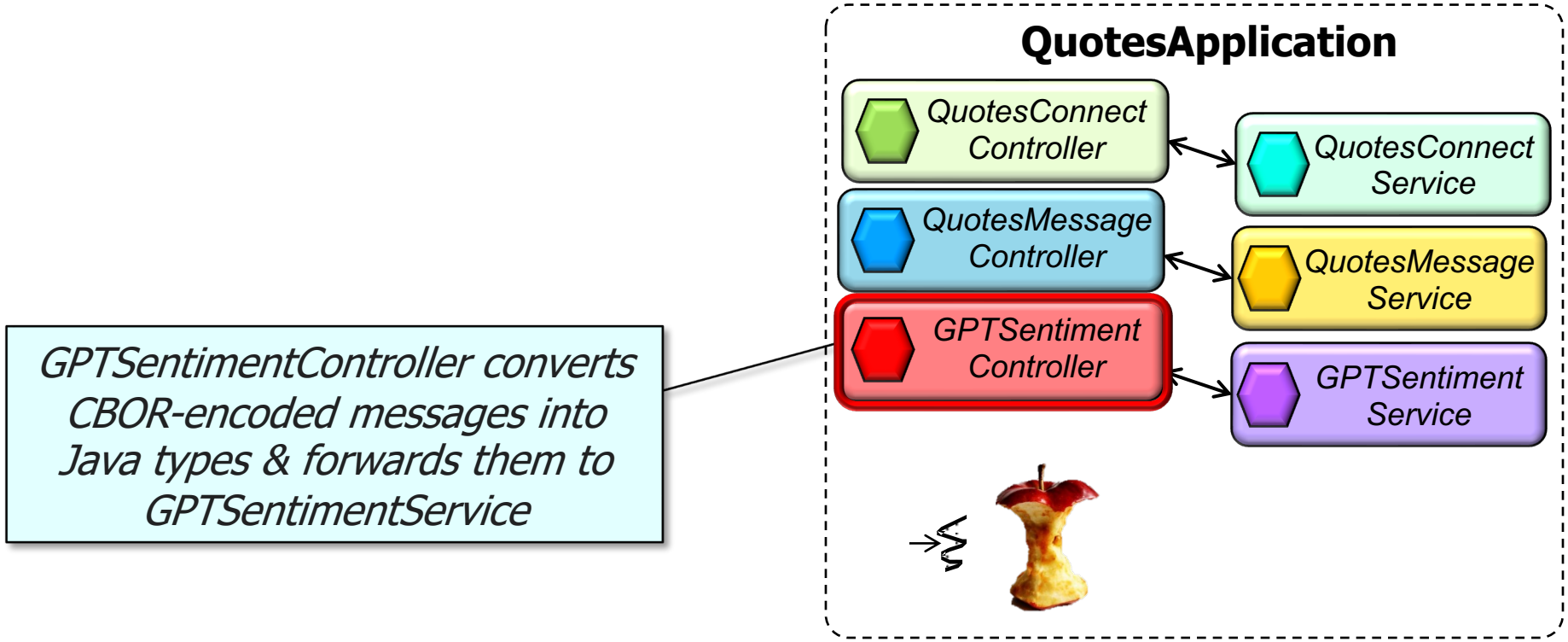


QuotesMessageService implements methods that process requester message requests & return associated subscriptions & Shakespeare quotes

The QuotesMessageService retrieves quotes from an R2DBC database

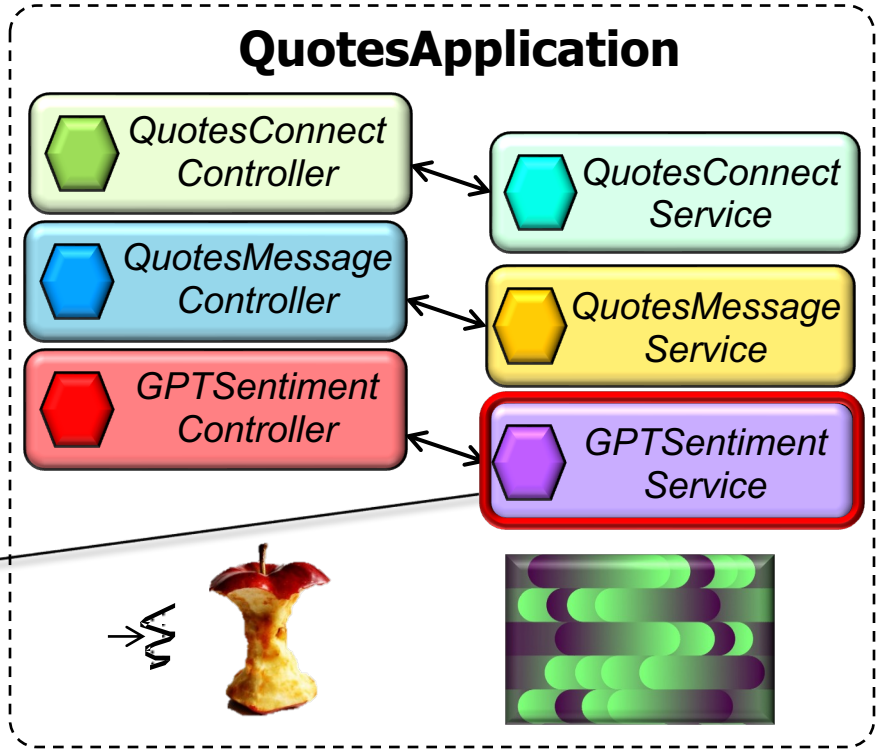
Overview of the RSocket Shakespeare Quotes App

- This case study shows how an RSocket requester can exchange binary messages asynchronously with various controllers & services



Overview of the RSocket Shakespeare Quotes App

- This case study shows how an RSocket requester can exchange binary messages asynchronously with various controllers & services



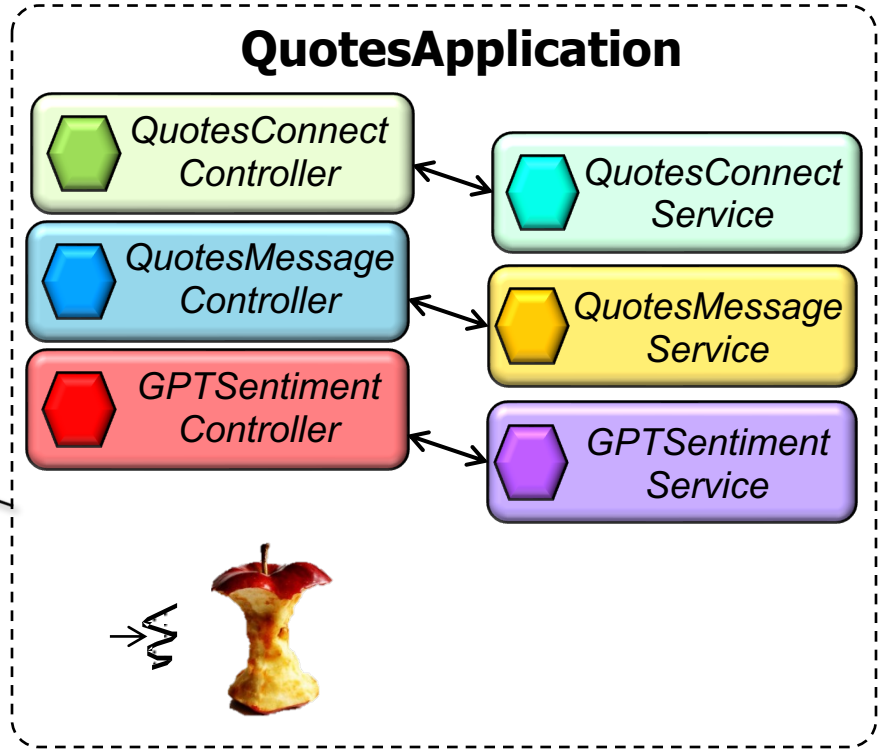
GPTSentimentService implements the methods that use ChatGPT to analyze the sentiment of Shakespeare quotes

See openai.com/blog/introducing-chatgpt-and-whisper-apis

Overview of the RSocket Shakespeare Quotes App

- This case study shows how an RSocket requester can exchange binary messages asynchronously with various controllers & services

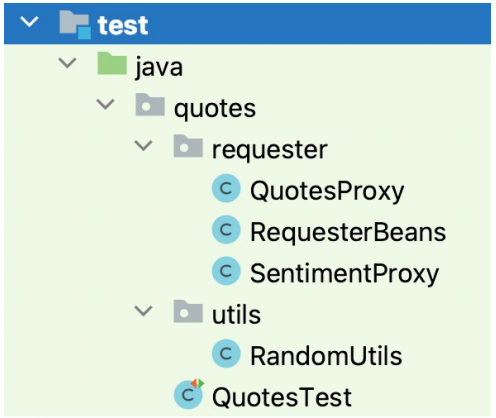
These controllers & services all currently reside in a single microservice to simplify testing, but it's easy to separate them

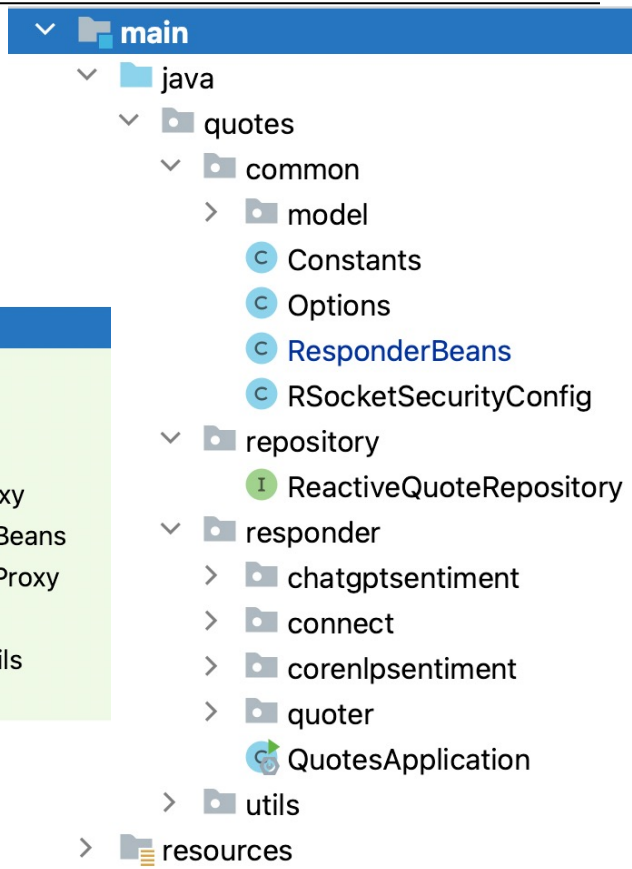


Structure & Functionality of the Shakespeare Quotes App

Structure & Functionality of the Shakespeare Quotes App

- The RSocket Shakespeare Quotes App project source code is organized into several packages

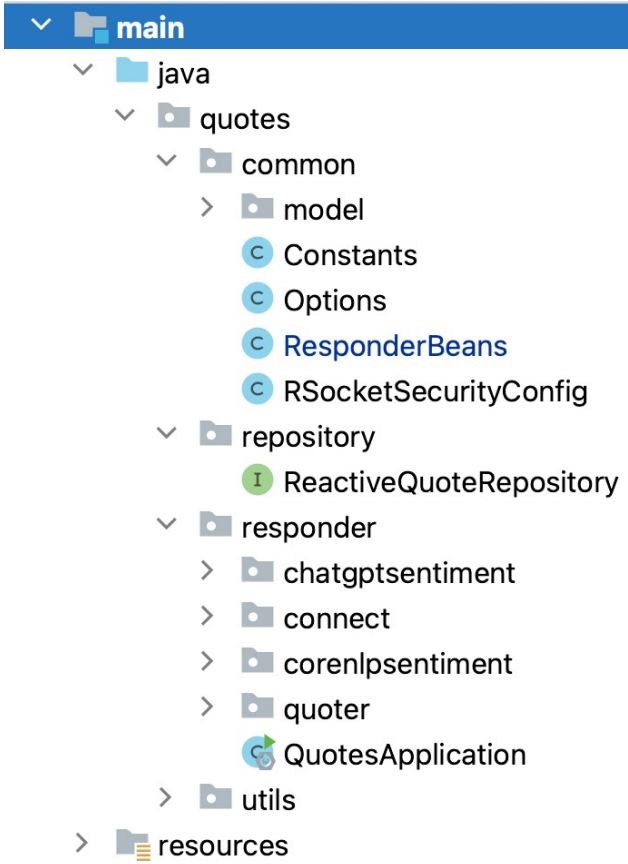




See github.com/douglascraigschmidt/LiveLessons/tree/master/RSocket/ex3

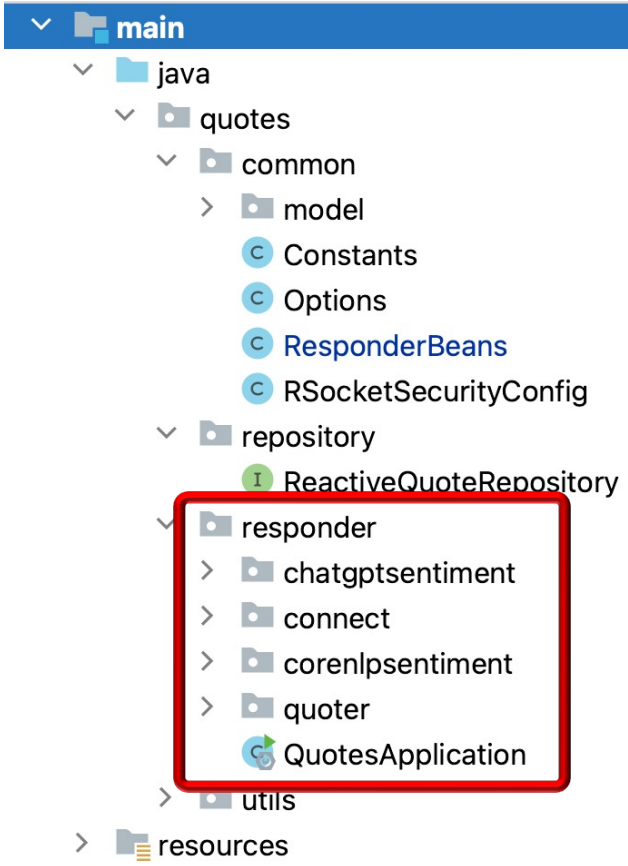
Structure & Functionality of the Shakespeare Quotes App

- The RSocket Shakespeare Quotes App project source code is organized into several packages
  - The main folder contains the responder-side classes



Structure & Functionality of the Shakespeare Quotes App

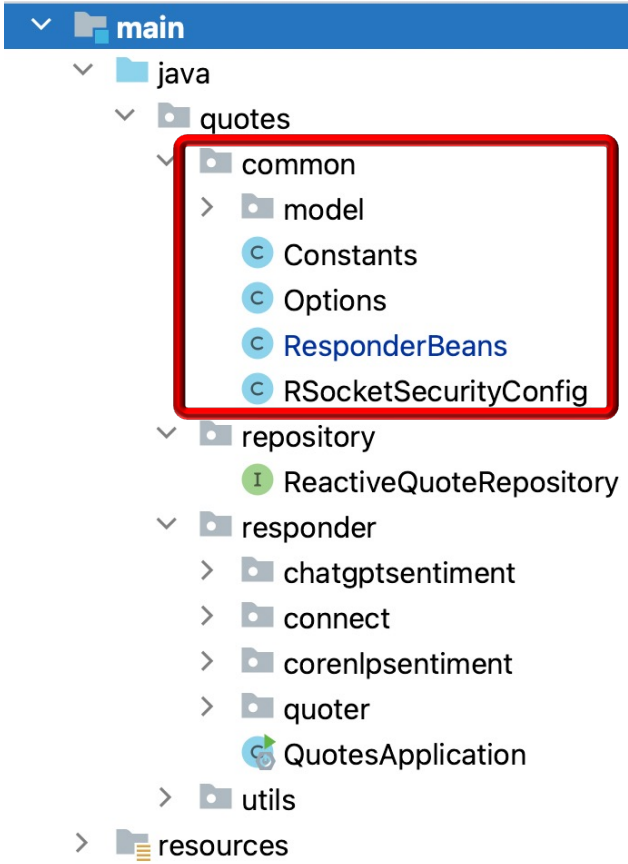
- The RSocket Shakespeare Quotes App project source code is organized into several packages
  - The main folder contains the responder-side classes
    - responder
      - The Application, Controller, & Service classes that enable message reception & responses, as well as ChatGPT sentiment analysis



These controllers & services could be separated into different microservices

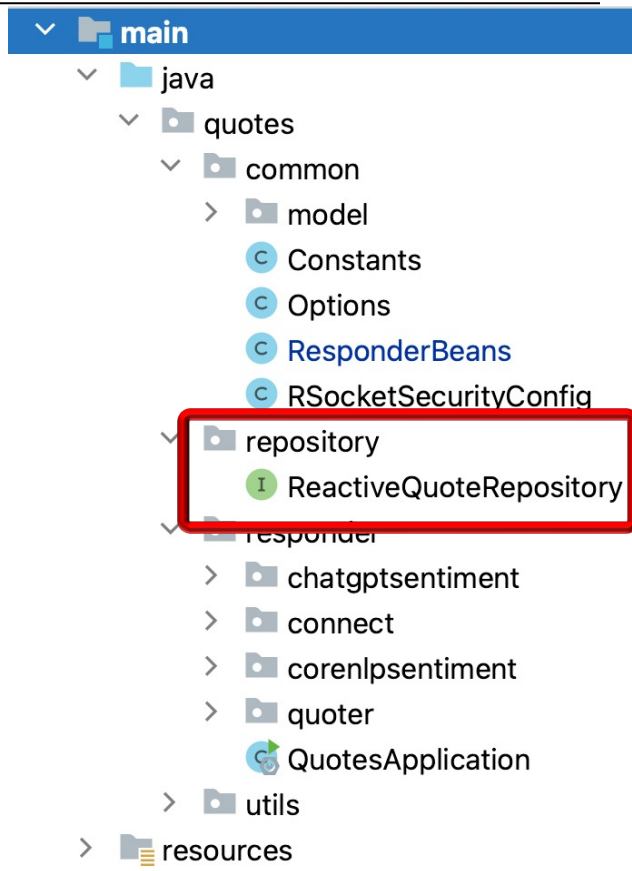
Structure & Functionality of the Shakespeare Quotes App

- The RSocket Shakespeare Quotes App project source code is organized into several packages
  - The main folder contains the responder-side classes
    - responder
    - common
      - The project-specific reusable classes, including a security module



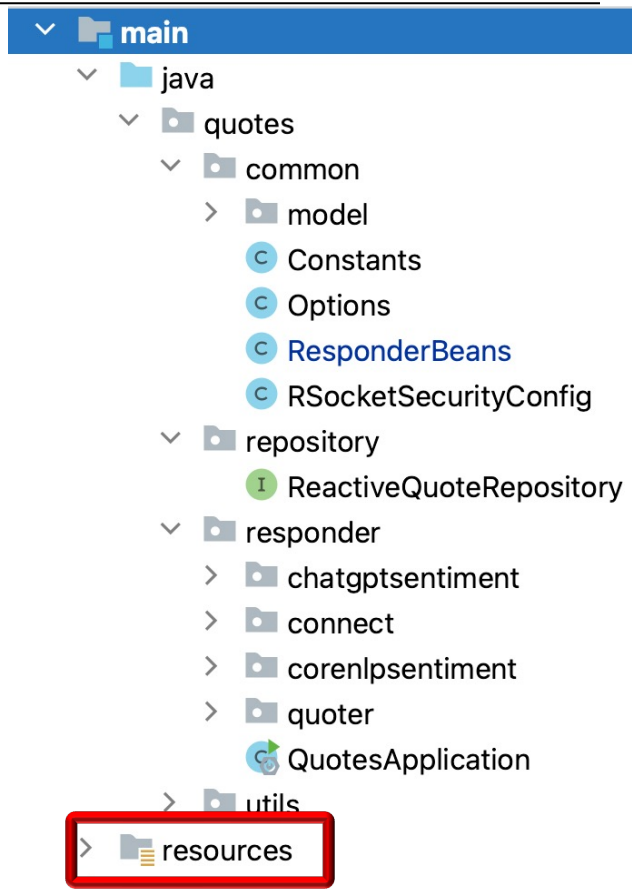
Structure & Functionality of the Shakespeare Quotes App

- The RSocket Shakespeare Quotes App project source code is organized into several packages
  - The main folder contains the responder-side classes
    - responder
    - common
    - repository
      - The R2DBC database containing the Shakespeare quotes



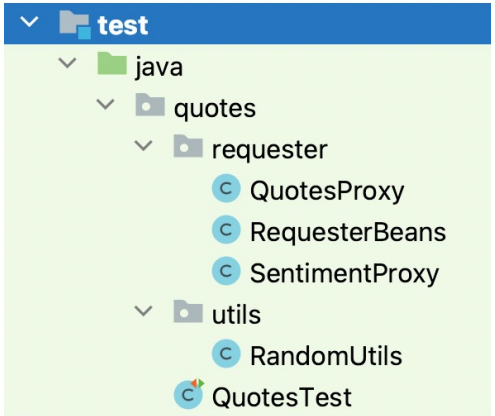
Structure & Functionality of the Shakespeare Quotes App

- The RSocket Shakespeare Quotes App project source code is organized into several packages
  - The main folder contains the responder-side classes
    - responder
    - common
    - repository
    - resources
      - The responder name, port number, R2DBC database configurations, & Shakespeare quotes data in SQL format



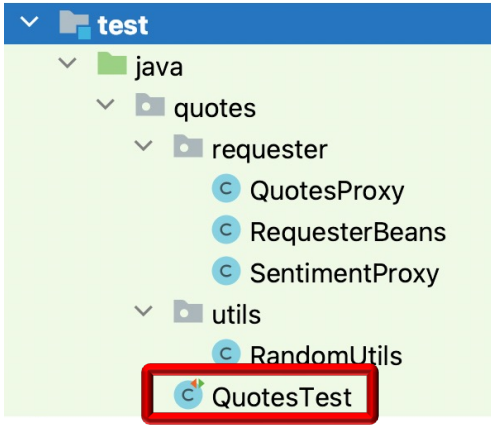
Structure & Functionality of the Shakespeare Quotes App

- The RSocket Shakespeare Quotes App project source code is organized into several packages
  - The main folder contains the responder-side classes
  - The test folder contains the requester-side classes



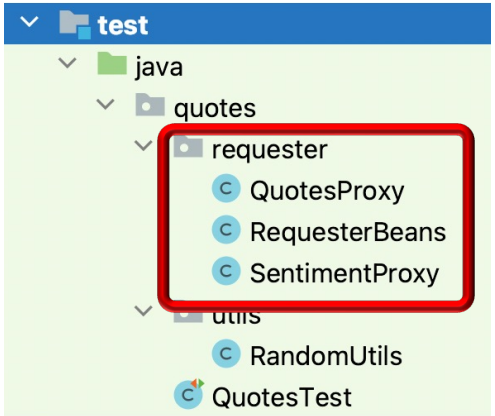
Structure & Functionality of the Shakespeare Quotes App

- The RSocket Shakespeare Quotes App project source code is organized into several packages
  - The main folder contains the responder-side classes
  - The test folder contains the requester-side classes
    - The test driver that connects with the responder & exchanges messages



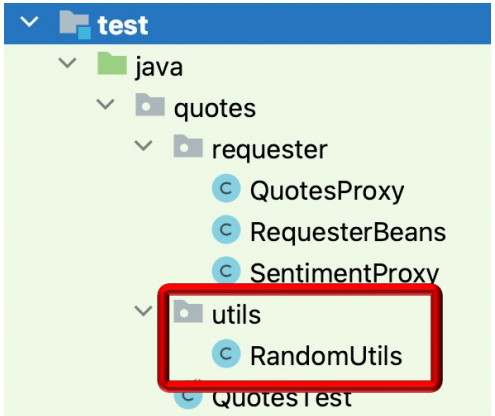
Structure & Functionality of the Shakespeare Quotes App

- The RSocket Shakespeare Quotes App project source code is organized into several packages
  - The main folder contains the responder-side classes
  - The test folder contains the requester-side classes
    - The test driver that connects with the server & exchanges messages
    - requester
      - The proxies & requester bean that establishes a secure connection with the responder & sends/receives binary messages



Structure & Functionality of the Shakespeare Quotes App

- The RSocket Shakespeare Quotes App project source code is organized into several packages
  - The main folder contains the responder-side classes
  - The test folder contains the requester-side classes
    - The test driver that connects with the server & exchanges messages
    - requester
    - utils
      - A project-independent reusable class that generates random numbers



End of Overview of the RSocket Shakespeare Quotes App