SERVICE ORIENTED ARCHITECTURES

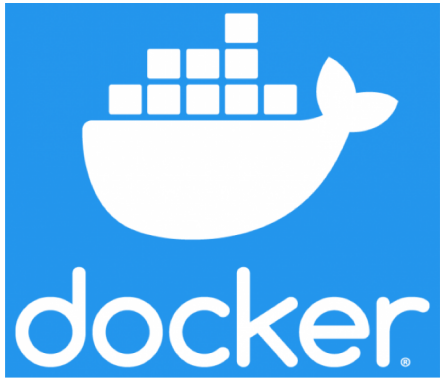
ACIT3855 – WINTER 2024

AGENDA – LESSON 9

- Quick Review
- Quiz 7
- Midterm Review and Assignment
- Topics:
  - Dashboard UI
  - CORS
- Lab 9 – CORS and Dashboard UI

REVIEW

- Docker and Microservices
- Single Original Policy



QUIZ 7

- Quiz is on the Learning Hub
- Open book, but do your own work
- You have <15 minutes to complete it

COURSE SCHEDULE

Week	Topics	Notes
1	• Services Based Architecture Overview • RESTful APIs Review	Lab 1
2	• Microservices Overview • Edge Service	Lab 2, Quiz 1
3	• Database Per Service • Storage Service (SQLite)	Lab 3, Quiz 2
4	• Logging, Debugging and Configuration • Storage Service (MySQL)	Lab 4, Quiz 3
5	• RESTful API Specification (OpenAPI) • Processing Service	Lab 5, Quiz 4
6	• Synchronous vs Asynchronous Communication • Message Broker Setup, Messaging and Event Sourcing	Lab 6, Quiz 5
7	• Deployment - Containerization of Services *Note: At home lab for Monday Set*	Lab 7, Quiz 6 (Sets A and B) , Assignment 1 Due
8	• Midterm Week	Midterm Review Quiz
9	• Dashboard UI and CORS	Lab 8, Quiz 7
10	• Spring Break	No Class
11	• Issues and Technical Debt	Lab 9, Quiz 8
12	• Deployment – Centralized Configuration and Logging	Lab 10, Quiz 9
13	• Deployment – Load Balancing and Scaling *Note: At home lab for Monday Set*	Lab 11, Quiz 10 (Sets A and B)
14	• Final Exam Preview	Quiz 10 (Set C), Assignment 2 Due
15	• Final Exam	

DASHBOARD USER INTERFACE

Dashboard UI

- Contain information such as stats, analytics, schedules, messages and much more
- Typically dynamic – changes as the system data changes

Single Page Application (SPA)

- Dynamically rewrites the page in the browser based on user input and/or requests to a backend
- Typically developed using a JavaScript framework like Angular or React
- Creates a more traditional desktop experience to the user as the page is not completely reloading after each action.

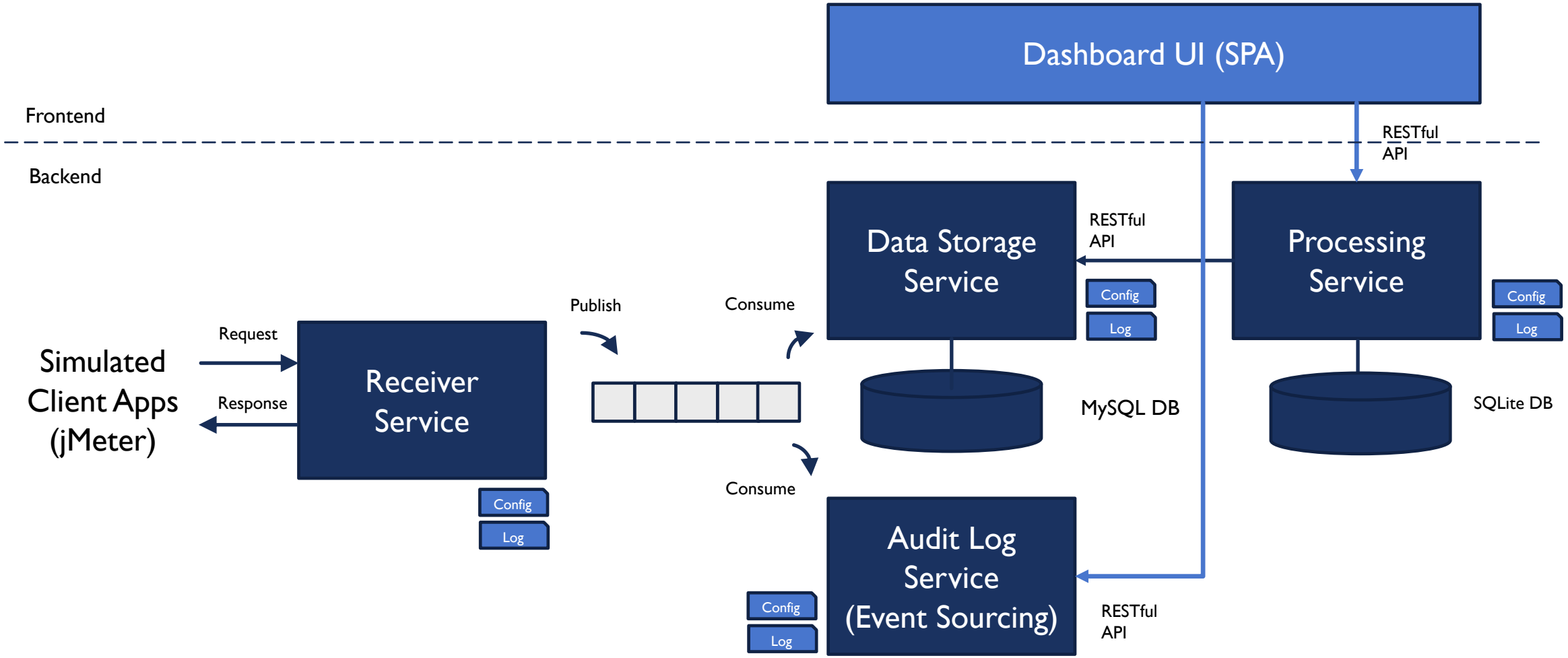
SPA CHARACTERISTICS

- Web Server serves HTML/CSS/JavaScript pages to the user's browser
- JavaScript dynamically updates the HTML/CSS as the user performs actions and upon requests/responses to backend services
- Frontend – The SPA
- Backend – The RESTful services providing the GET/POST/PUT/DELETE methods to the front end
- Developers sometimes specialize in the Frontend or the Backend, or work across both as in Full-Stack Developers

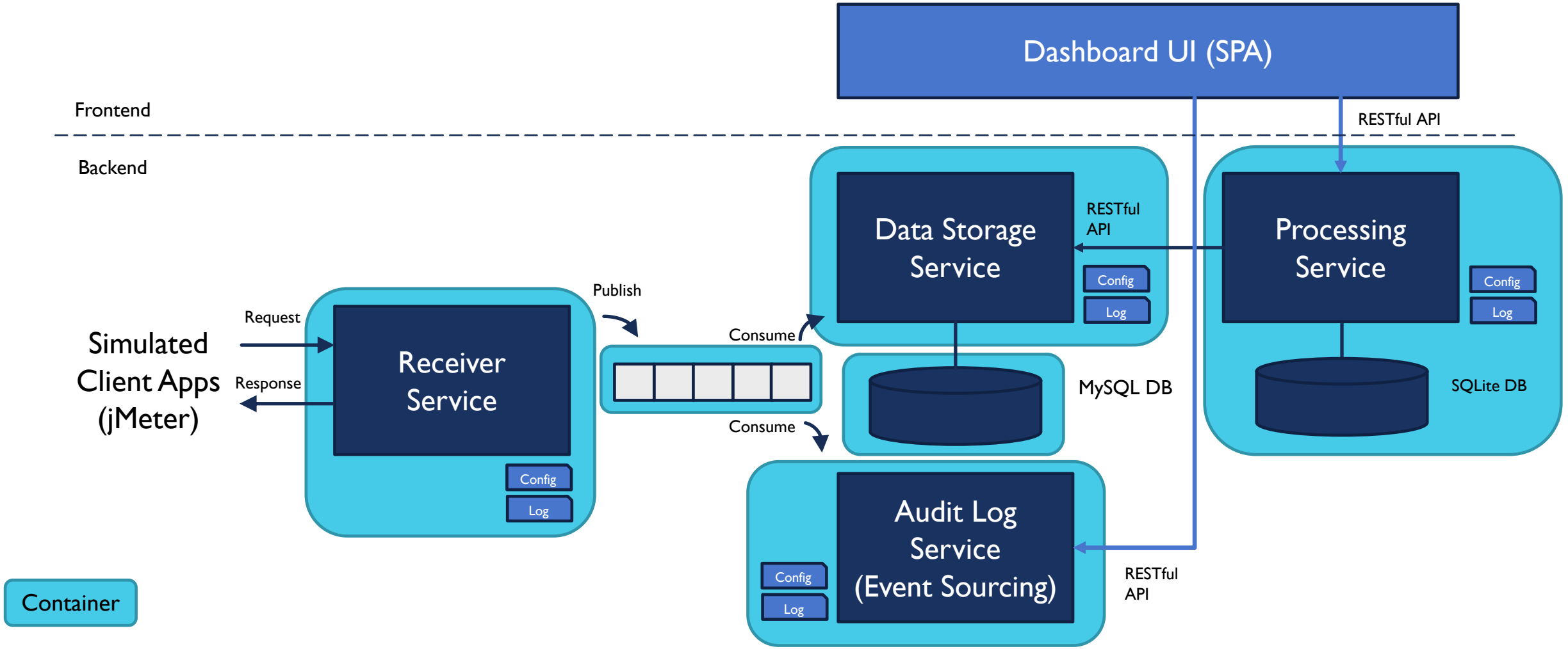
SOP AND CORS

- Often the server for your Frontend SPA and the server(s) for your Backend services are hosted on the same domain (i.e., *www.example.com*)
- But what if your Backend services are on a different domain (i.e., *www.example.com* for the Frontend and *api.example.com* for the Backend)?
  - The Browser has a restriction called SOP (Single Origin Policy). It will prevent a Frontend on one domain from calling Backend services on a different domain.
  - This can be particularly troublesome for Microservices where there are many services possible deployed on different domains
  - What are the solutions?
    - Proxy (Client Side) – Frontend has a proxy so that all backend services calls go to the same domain but are redirected to the correct domain
    - Reverse Proxy (Server Side) – Backend services are deployed behind a reverse proxy so they are all accessed from the same domain
    - Enable CORS (Cross-Origin Resource Sharing). on the backend services (i.e., tell the Frontend client it's okay to call this service)

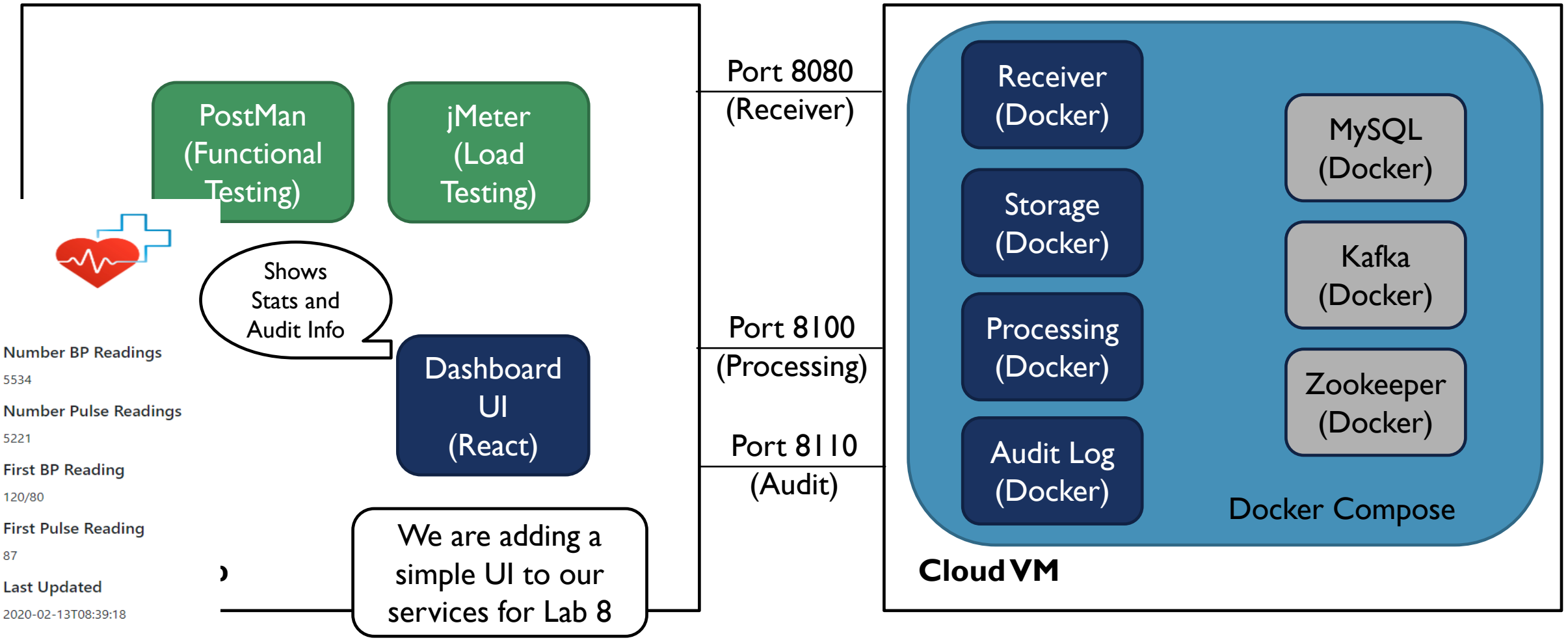
OUR SAMPLE APPLICATION – LAB 8



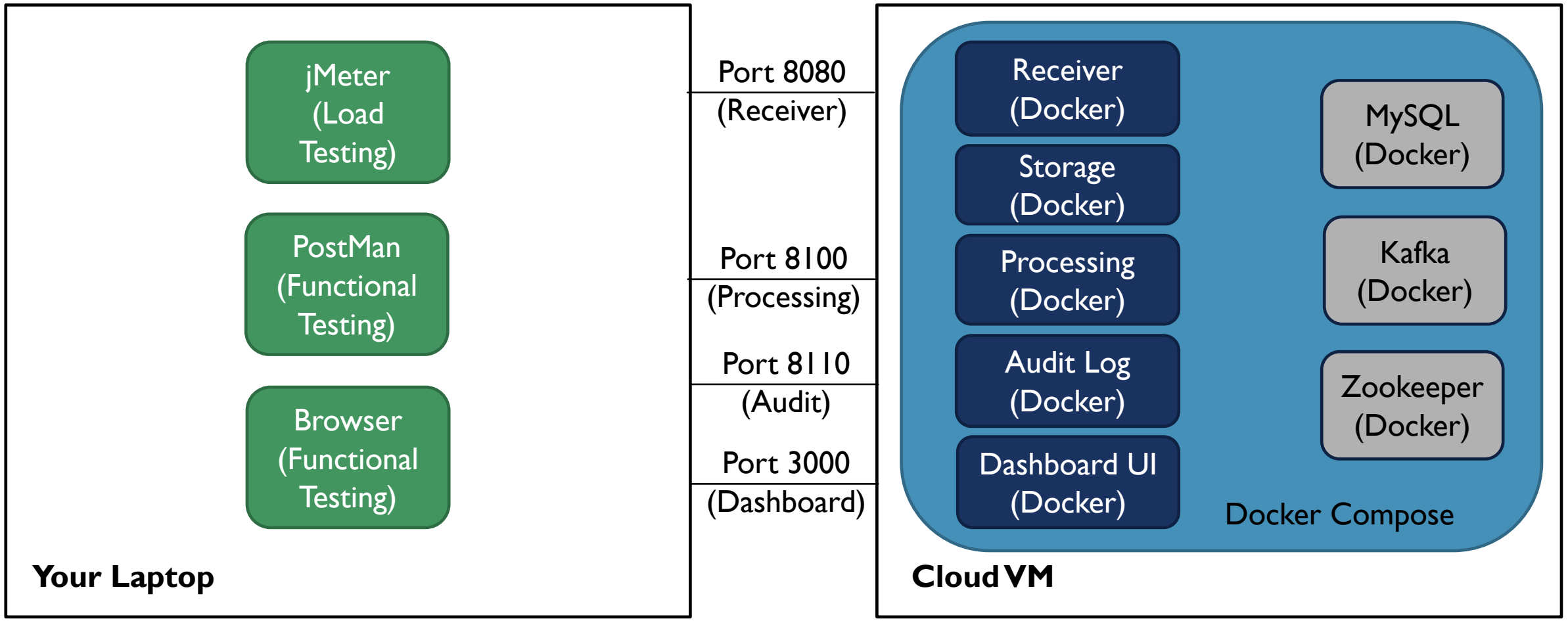
OUR SAMPLE APPLICATION – LAB 7



LAB 8 TEST ENVIRONMENT



LAB 8 TEST ENVIRONMENT



DASHBOARD UI

- Lets go through the updates you will need to make in the sample code…
- Note: There are two options for this lab – React App or Pure HTML/JS

TODAY'S LAB

Today you will:

- Demo your Lab 7 by the end of class.
  - If you're having problems accessing your services with curl or Postman, update the app.py line:
    - app.run(port=<port>, host="0.0.0.0")
    - You may have to restart the Storage Service to get it to connect to Kafka
- Work on Lab 8 – It is due for Demo next class.
  - Collaborate on the Dashboard UI in Discord. Use one of the sample apps as a starting point or create something from scratch (if you want).