

Your Next Week

<div>Saturday May 16</div> <div>9:00 AM</div> <div><div>— DUE Class 18 Code Challenge</div><div>— DUE Class 18 Lab</div><div>— DUE Class 19 Reading</div><div>— Class 19</div><div>— Interview Prep 04</div></div> <div>MIDNIGHT</div> <div><div>— DUE Class 19 Learning Journal</div></div>	<div>Sunday May 17</div> <div>MIDNIGHT</div> <div><div>— DUE Class 18-19 Feedback</div></div>	<div>Monday May 18</div>	<div>Tuesday May 19</div> <div>6:30 PM</div> <div><div>— DUE Class 19 Mock Interviews</div><div>— DUE Class 19 Lab</div><div>— NO READING Class 20</div><div>— Class 20A</div></div>
<div>Wednesday May 20</div> <div>9:00 AM</div> <div><div>— Class 20B</div></div> <div>MIDNIGHT</div> <div><div>— NO LEARNING JOURNAL Class 20</div></div>	<div>Thursday May 21</div> <div>6:30 PM</div> <div><div>— Co-working</div></div>	<div>Friday May 22</div>	<div>Saturday May 23</div> <div>9:00 AM</div> <div><div>— Midterms Work</div><div>— DUE Class 20 Lab</div></div>

What We've Covered

<i>Module 01</i> Javascript Fundamentals and Data Models <i>C01 — Node Ecosystem, TDD, CI/CD</i> <i>C02 — Classes, Inheritance, Functional Programming</i> <i>C03 — Data Modeling & NoSQL Databases</i> <i>C04 — Advanced Mongo/Mongoose</i> <i>C05 — DSA: Linked Lists</i>	<i>Module 02</i> API Servers <i>C06 — HTTP and REST</i> <i>C07 — Express</i> <i>C08 — Express Routing & Connected API</i> <i>C09 — API Server</i> <i>C11 — DSA: Stacks and Queues</i>	<i>Module 03</i> Auth/Auth <i>C10 — Authentication</i> <i>C12 — OAuth</i> <i>C13 — Bearer Authorization</i> <i>C14 — Access Control (ACL)</i> <i>C15 — DSA: Trees</i>	<i>Module 04</i> Realtime <i>C16 — Event Driven Applications</i> <i>C17 — TCP Server</i> <i>C18 — Socket.io</i> C19 — Message Queues C20 — Midterms Prep Midterms
<i>Module 05</i> React Basics C21 — Component Based UI C22 — React Testing and Deployment C23 — Props and State C24 — Routing and Component Composition C25 — DSA: Sorting and HashTables	<i>Module 06</i> Advanced React C26 — Hooks API C27 — Custom Hooks C28 — Context API C29 — Application State with Redux C30 — DSA: Graphs	<i>Module 07</i> Redux State Management C31 — Combined Reducers C32 — Asynchronous Actions C33 — Additional Topics C34 — React Native C35 — DSA: Review	<i>Module 08</i> UI Frameworks C36 — Gatsby and Next C37 — JavaScript Frameworks C38 — Finals Prep Finals

Lab 18 Review

Code Challenge 18

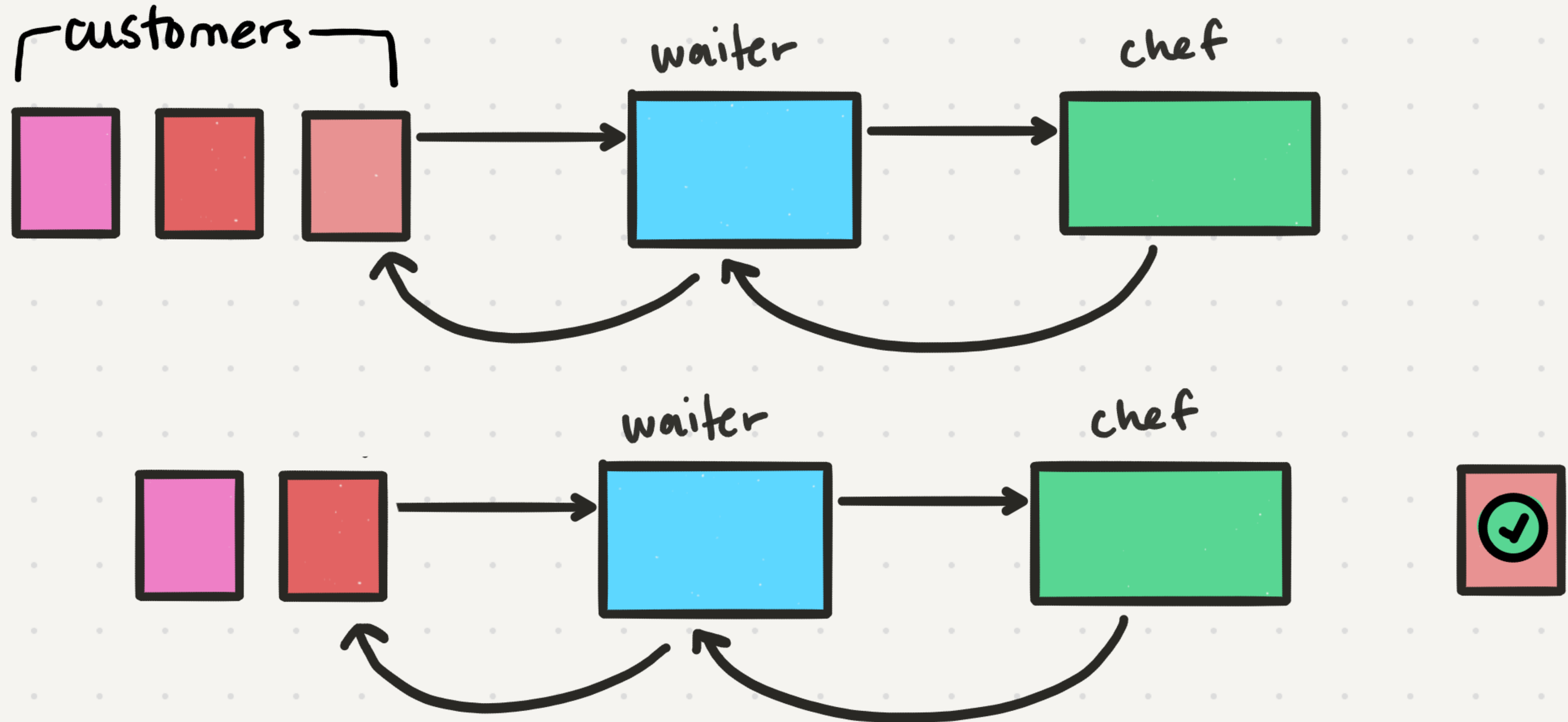
Review

Class 19

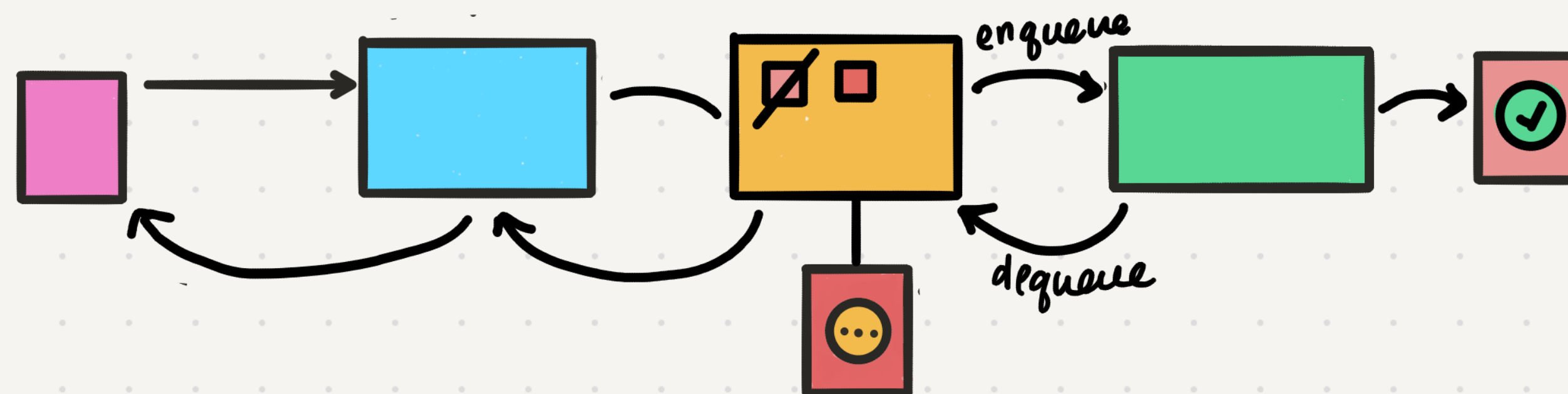
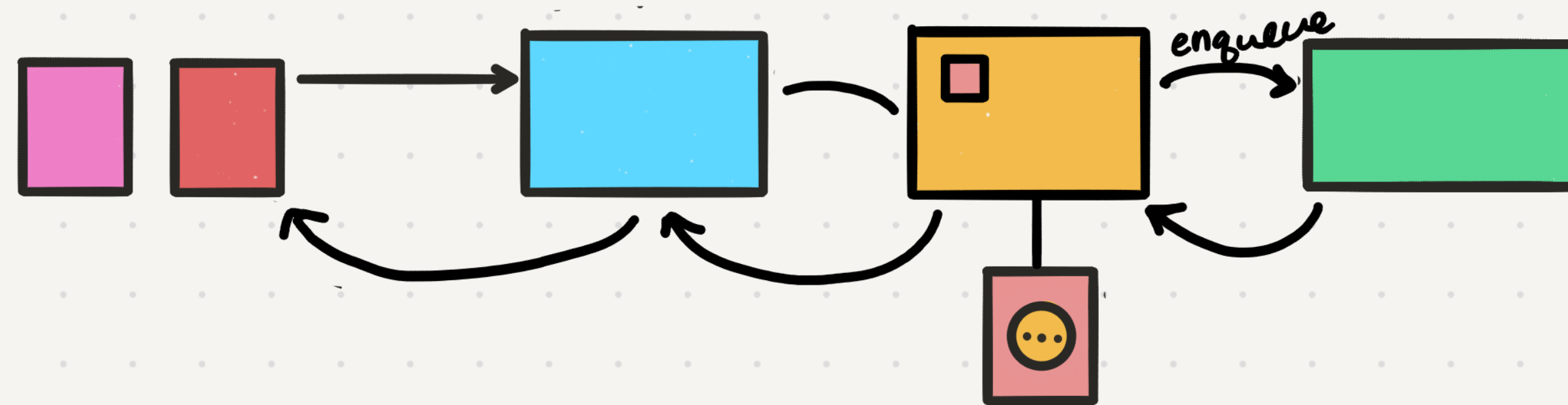
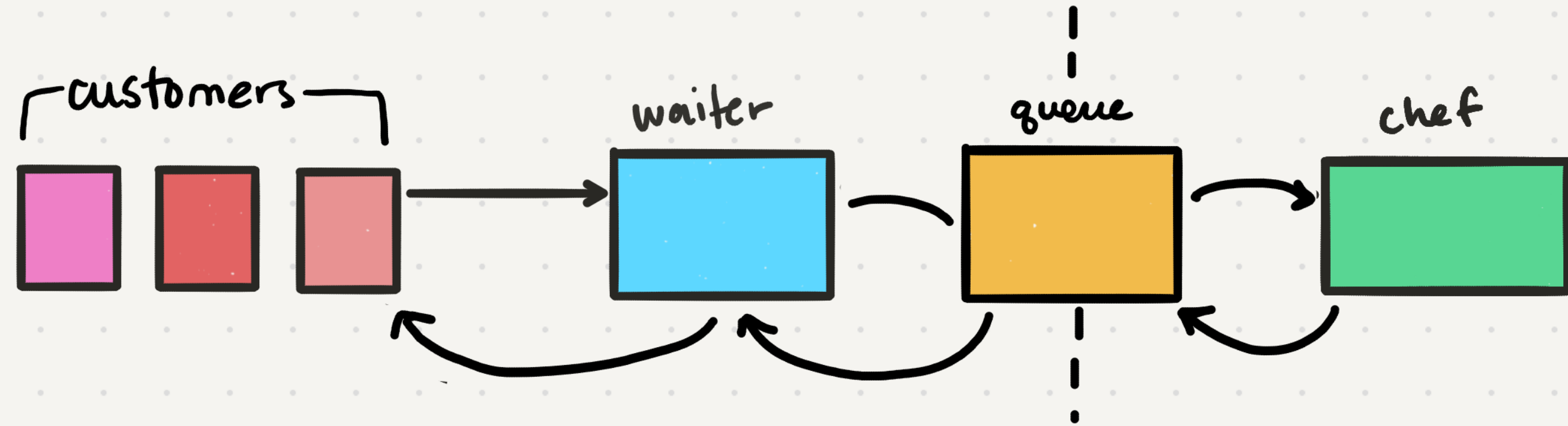
Message Queues

seattle-javascript-401n16

Working in a Restaurant



Working in a Restaurant



Message Queues

- Allows the client to get a quick response
- Waiter can say “your order has been taken and is on the way”
- Instead of the client standing in line waiting, blocking other clients

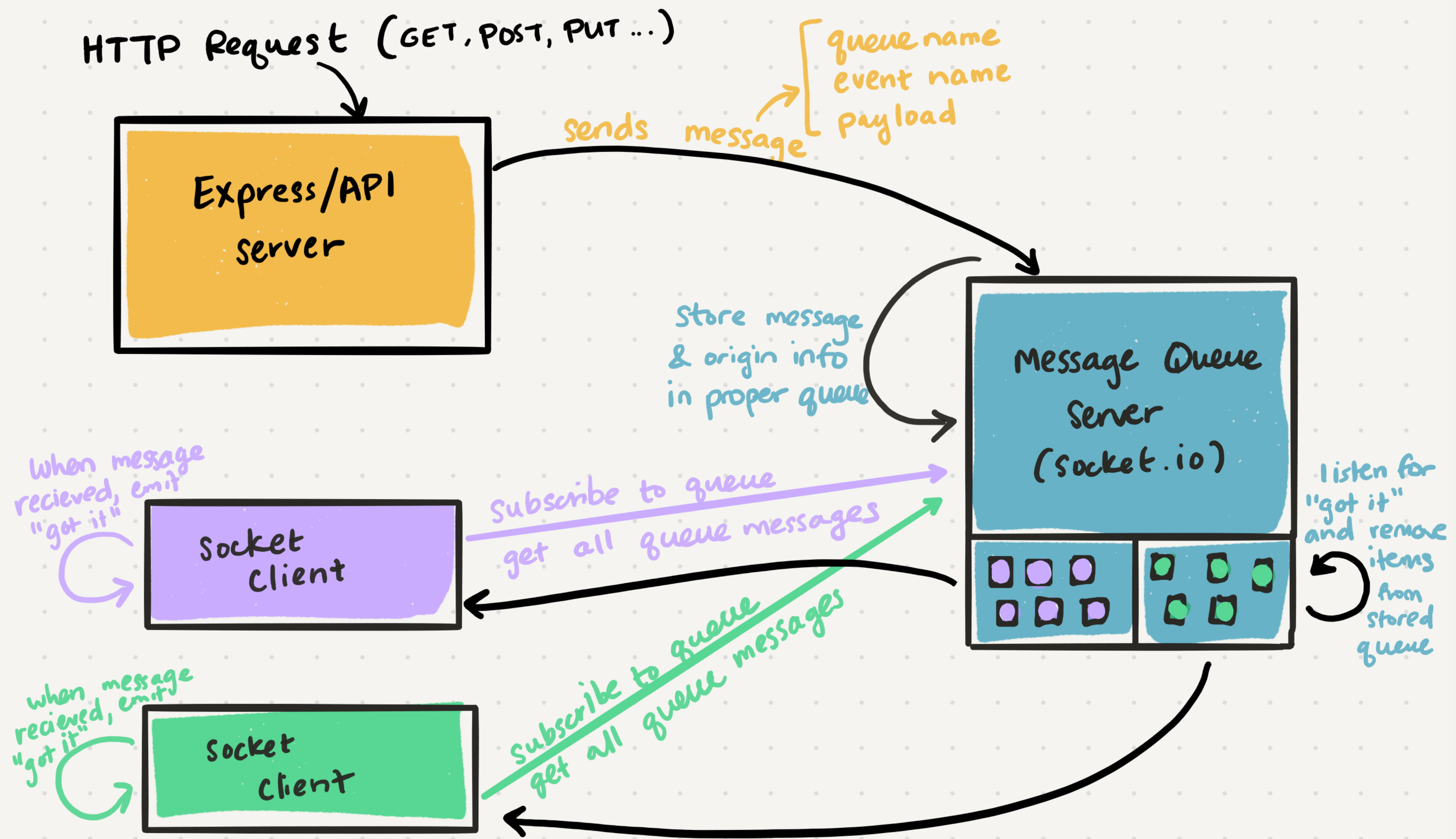
If you wait for the waiter, does that make you the waiter?



Message Queues

- The queue's job is to keep track of orders that haven't been "seen" by the chef
- When the chef sees an order, they will dequeue, make the order and then tell the waiter to serve
- If the chef is on break, the queue persists





Lab 19 Overview