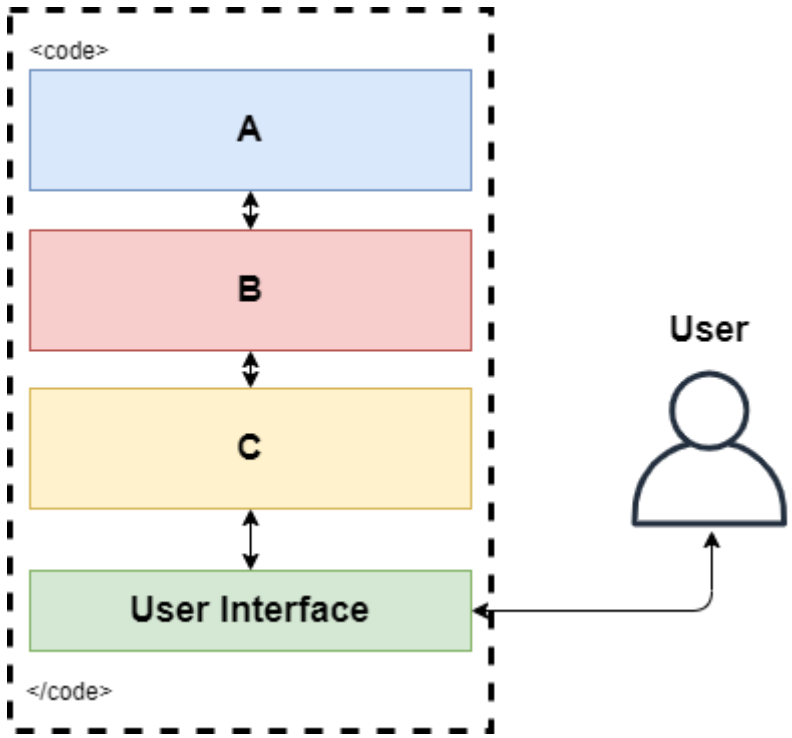
COSC 331 – Microservices and Software

Fall 2020

Introduction

- Syllabus is posted on Moodle
- Topics we'll be covering:
  - How to implement a microservices software architecture
  - Deployment of microservices
  - Designing APIs with RESTful, stateful, and streaming interfaces
  - Containerization of microservices with Docker and Kubernetes

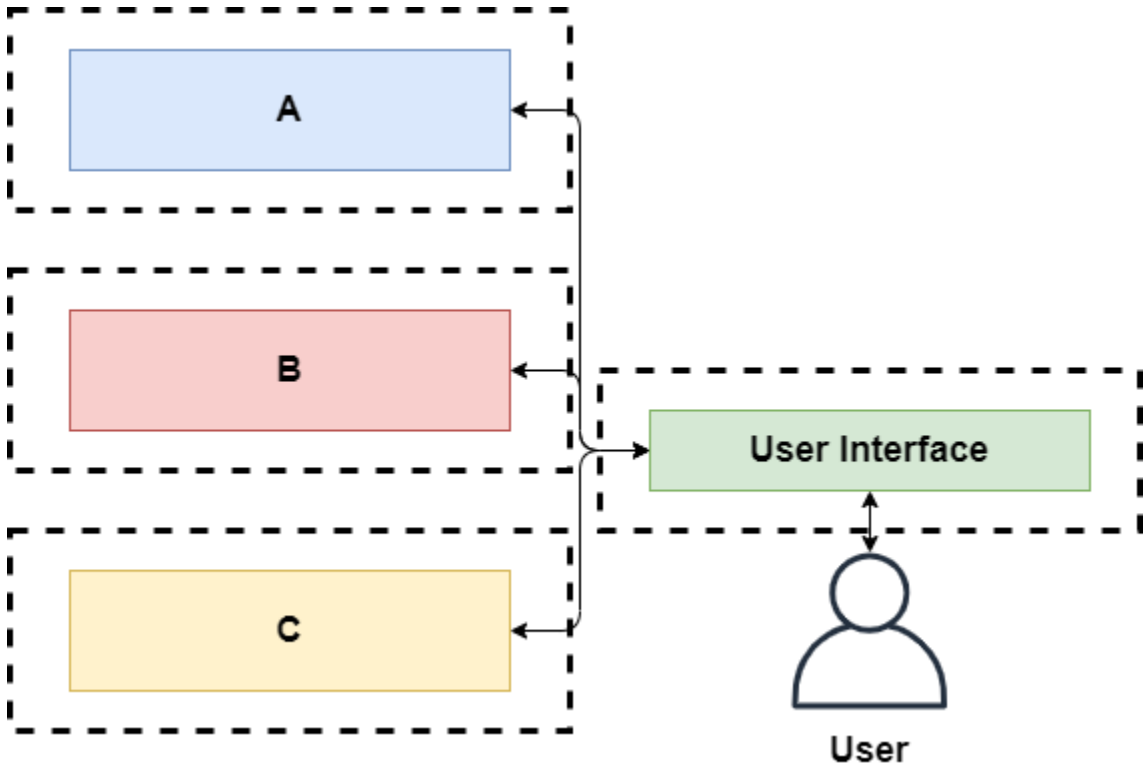
Old Way: Monolithic Application Design



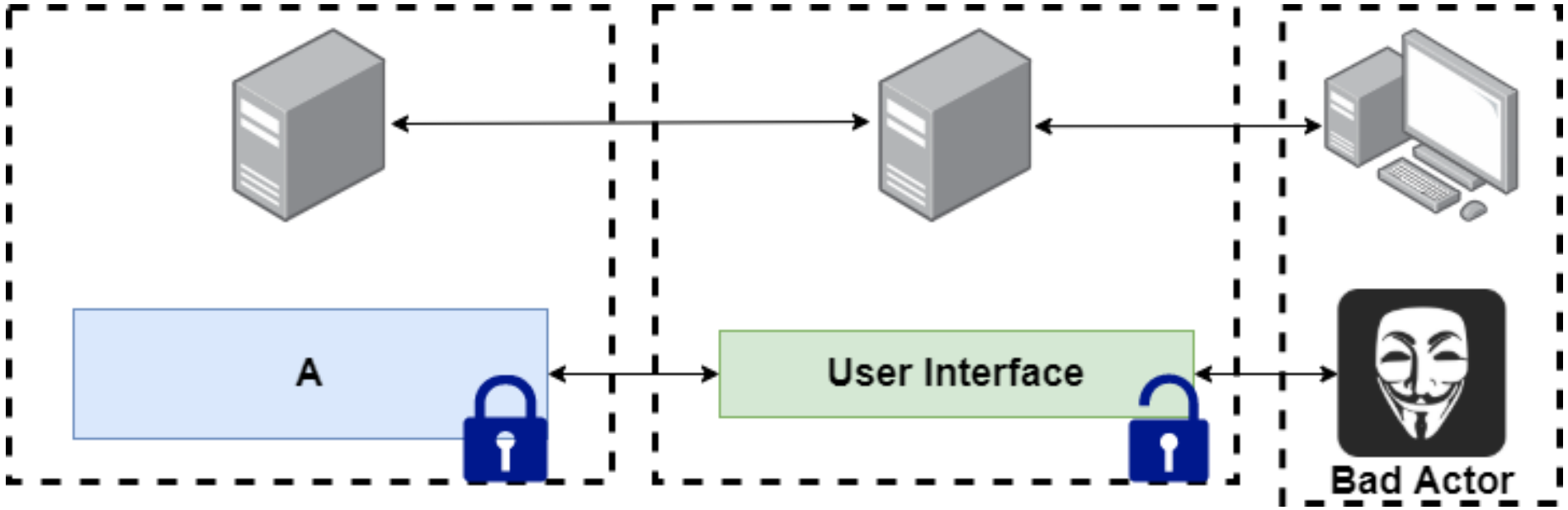
- A traditional monolithic application design uses a single (monolithic) process
- User interface, logic, and databases are all ***tightly coupled***, meaning the components are strongly linked together

So What is a Microservice Architecture?

- A microservice architecture replaces a monolithic design with individual microservices
- Each microservice handles only part of the underlying application logic

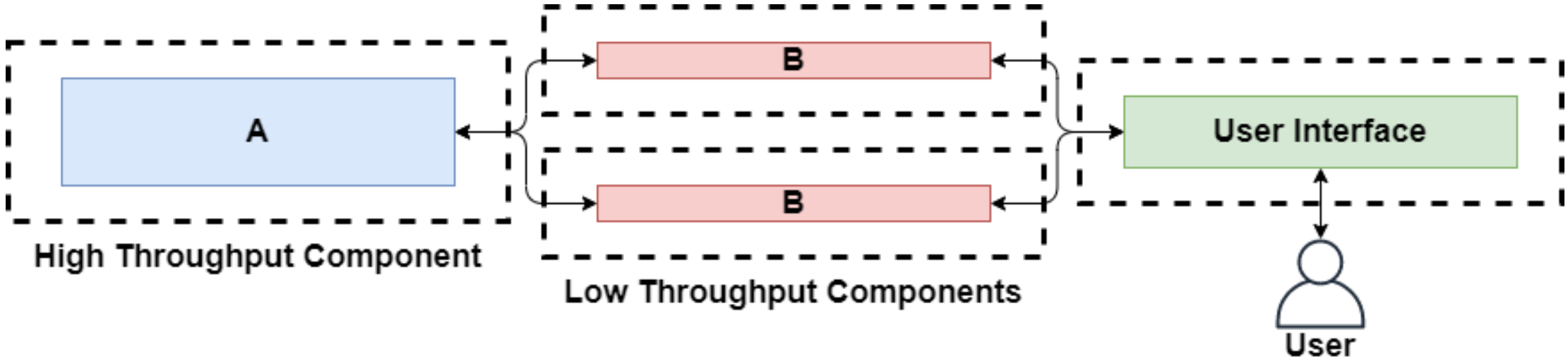


Why Use a Microservice? Isolation.



- Microservices allow for greater ***isolation*** between working components by spreading the application code across multiple servers

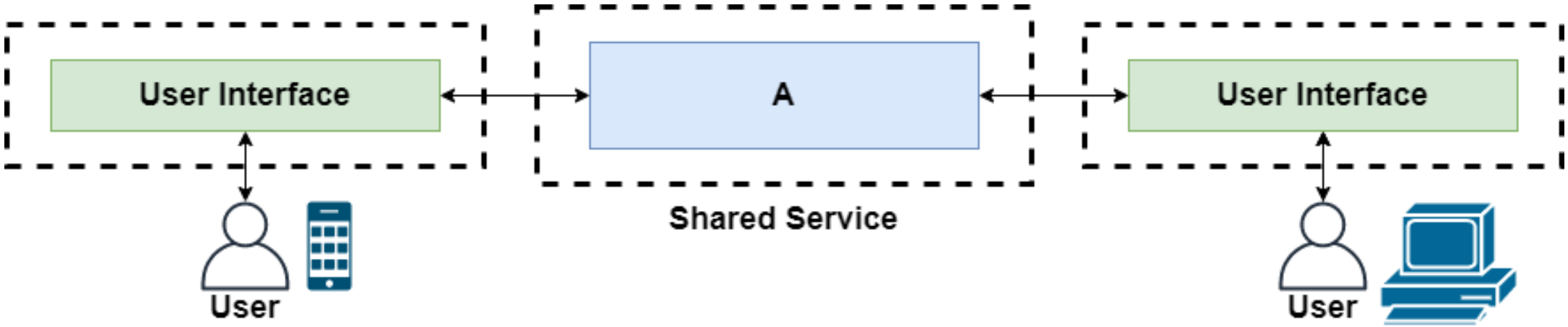
Why Use a Microservice? Scalability.



- Microservices also allow individual components to be ***scaled*** up or down to meet demand
- For example, a low-throughput (slow) service can be duplicated to handle additional user load

Why Use a Microservice? Reusability.

- Using a microservice also allows us to easily ***share and re-use*** an existing service
- In this example, a single service is providing data to two different applications – a web app, and a mobile app

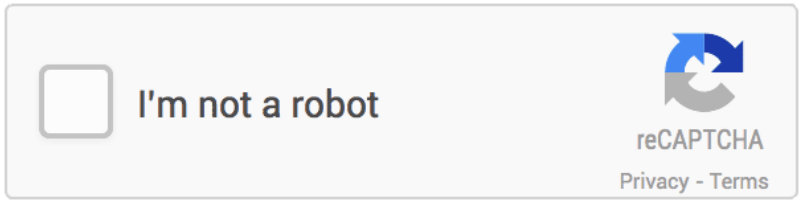


The Definition of a Microservice

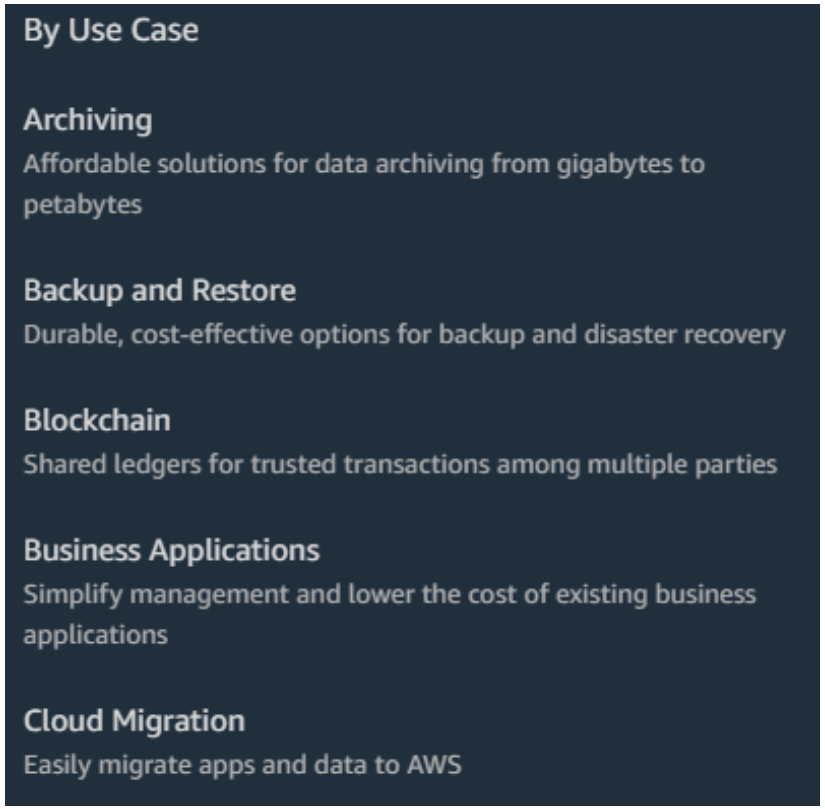
- “Microservice” is not a well-defined term
- However, we can define a basic set of criteria that a microservice meets:
  - It is an application of ***limited scope*** – it only handles input and output directly related to it’s function
  - It supports some form of output, usually via an **Application Programming Interface (API)**
  - It is typically served using common protocols like **HTTP**

Examples of Microservices

- Mailgun is a commercial microservice that provides simple mail handling APIs to send and receive email
- ReCAPTCHA is a human verification microservice provided by Google
- Stripe is a payment processing microservice that allows you to easily integrate payment systems into your application



Microservices in Modern Development



- Due to their security and scalability benefits, microservice architectures have become ubiquitous in modern development
- They've become a major market by themselves – AWS and other providers offer many different managed microservices
- Easier and cheaper to build with modules than do bespoke development

What We're Going to Learn

- In this course, we'll be covering how to build and deploy your own microservices, and how to combine those microservices into a full-fledged web application
- Starting next week, we'll begin writing our services in a variety of languages – Java, Python, and JavaScript – and learning how to interface with them remotely
- We'll be using DigitalOcean virtual servers for this course – login details for your machine will be emailed to you prior to next week's lab

Any Questions?