

# Microservices

Cloud Computing and SaaS



# Announcements

- Full Project Skeleton Released
  - Go code, dependencies, Dockerfiles
  - docker-compose file...more on this later
- Checkpoint 1 due Friday 10/16
  - Written assignment to demonstrate understanding of microservices and how they relate to the project
- AWS Educate Accounts
  - Must sign up in order to complete later parts of this course

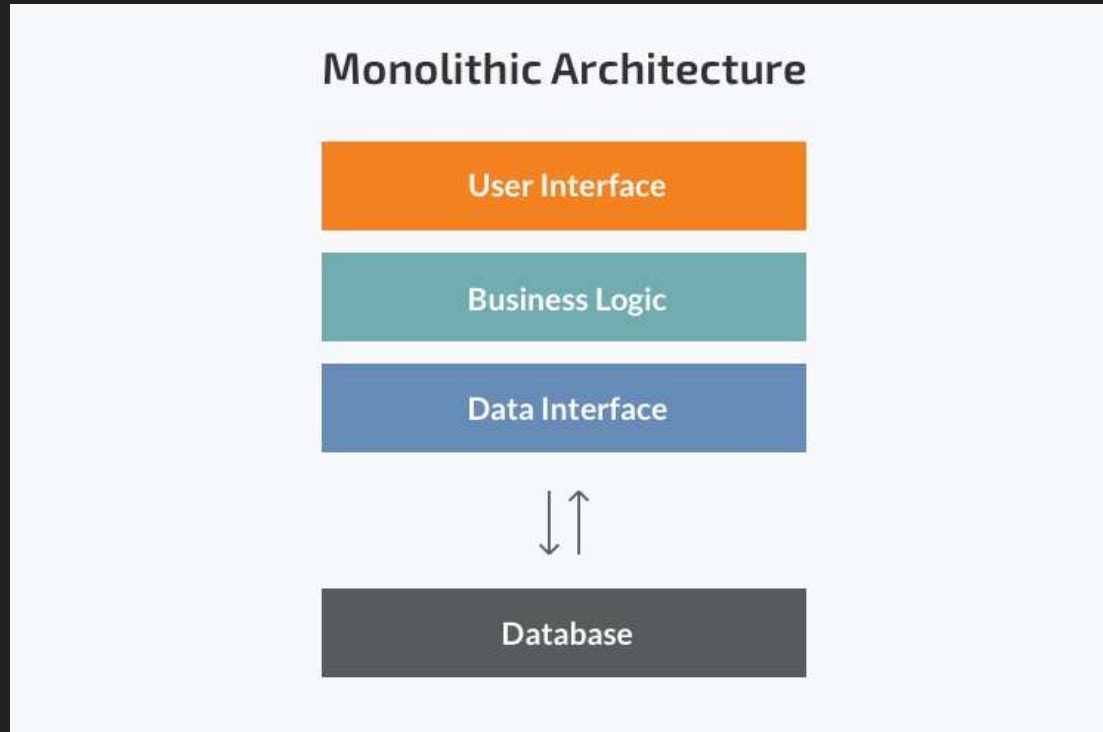
# Last Time

- Docker
- SQL
- Resume Workshop
  - Thanks Shane!
- LinkedIn Workshop

# Today

- Monolith Architecture
  - Benefits, Drawbacks
- Microservice Architecture
  - Benefits
  - Drawbacks
  - Better than Monoliths?

# Monolith Architecture

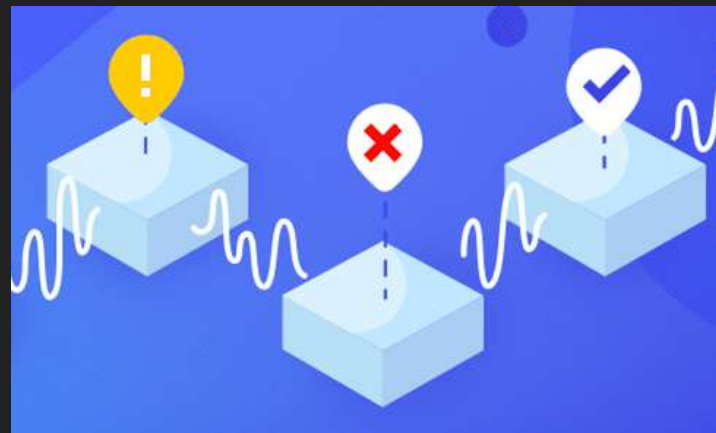


# Monolith Architecture

- Key Idea
  - Entire application is deployed as a whole
  - Run the application => ready to deploy
- Examples
  - Golang Web Server
  - CS61A Hog Project
  - CalCentral
    - <https://github.com/ets-berkeley-edu/calcentral>

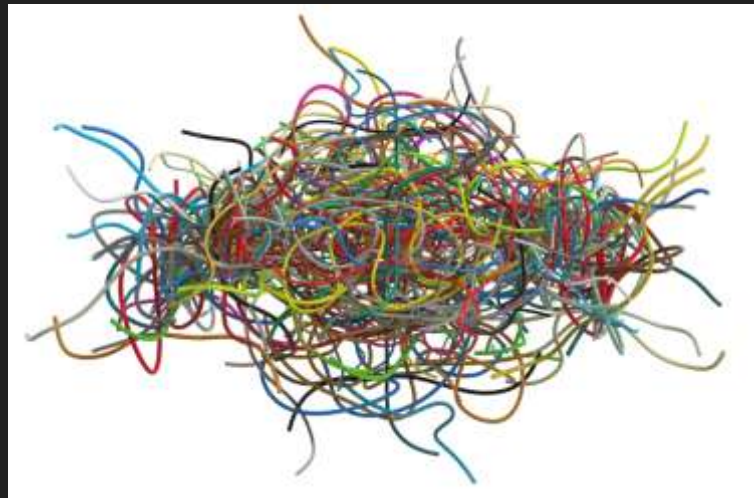
# Monolith Architecture

- Benefits
  - Simple to develop
    - All your code in one place
  - Simple to do end-to-end testing
    - Run and fuzz test the UI
  - Simple to deploy
    - Package + build the program and run multiple instances



# Monolith Architecture

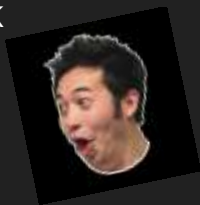
- Drawbacks
  - Complexity (!!)
    - Size of the codebase only gets larger
  - Maintenance
  - Updating requires full restart
  - Dependency Issues
  - A single bug can take down the entire program





# Microservice Architecture

- Key Idea
  - Break down products by functionality
  - Implement each functionality in its program with its own set of dependencies
- Examples
  - Google
  - Facebook
  - BearChat

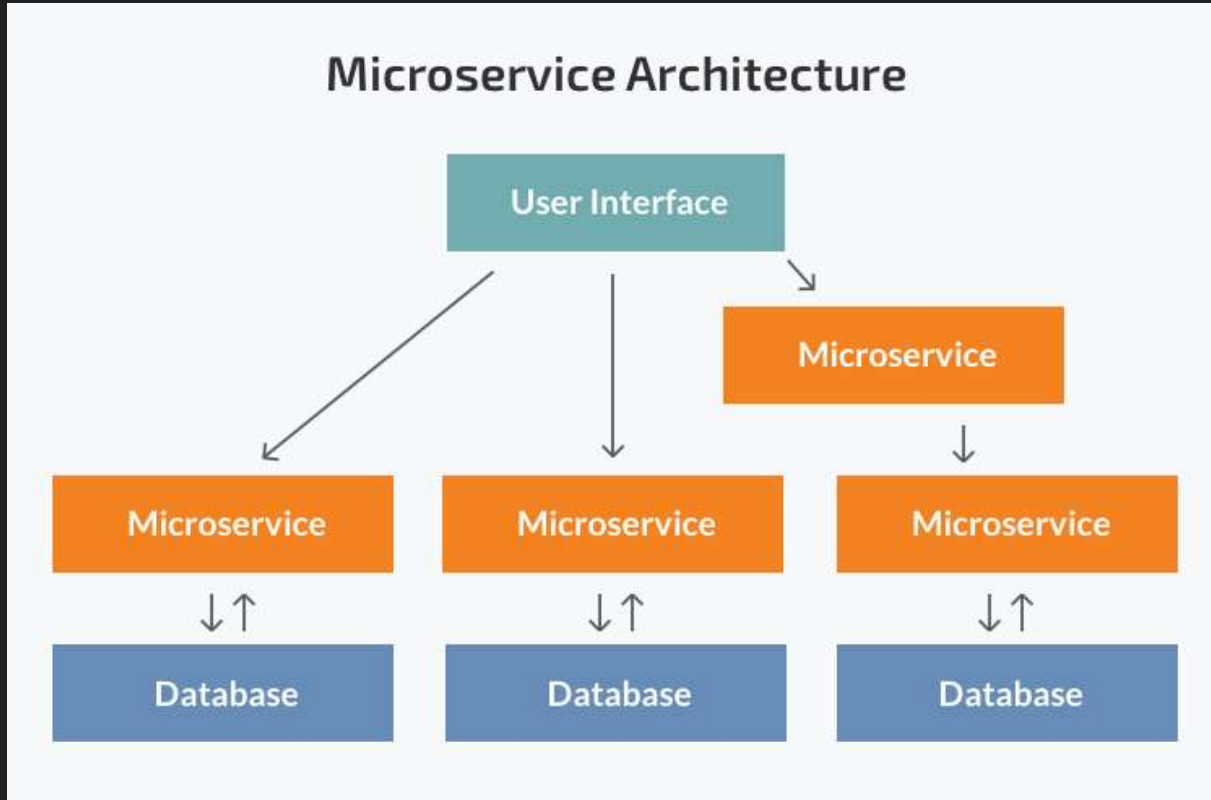


# Microservice Architecture

- Benefits
  - Easier to unit tests, services are smaller
  - Easier to deploy, services can be deployed independently
  - Easier to collaborate on, assign each team a microservice
  - Fault Isolation, errors in one services don't affect others
  - Easier to pivot tech stacks for different functionalities



# Microservice Architecture



# Microservice Architecture

- Drawbacks
  - Distributed System => also complex
    - Microservices can depend on the functionalities of other services
  - End-to-End testing more complex, multiple fault points
  - More containers = more memory overhead
  - Latency (minimal but worthy of discussion)



# Microservices vs Monoliths

- Monoliths
  - Small team
  - Simple Application
  - Quick launch
- Microservices
  - Complex and scalable application
  - Microservice expertise

# Project Microservice Structure

- Authorization Microservice
  - Sign-up, Sign-in, Logout, Verification
- Friends Microservice
  - Add/Remove Friends, Check mutual friends
- Posts Microservice
  - Create, Delete, and View Posts
- Profiles Microservice
  - Create, Update, and View Profiles