Engineering Robust Server Software

Introduction

Duke
UNIVERSITY

Brian Rogers
Duke ECE
Used with permission from Drew Hilton

Welcome To ERSS!

- Welcome to Engineering Robust Server Software (ERSS)
- Introductions:
  - **Instructor:** Brian Rogers
    - Adjunct faculty at Duke, day job as a processor architect in industry
    - Also have recently taught ECE 650 and ECE 565
  - **Teaching Assistants:** Longhao Zhu, Evan Li, Zixu Geng, Kaixin Lu
- **Important** - course website is here:
  - https://people.duke.edu/~bmr23/ece568/
- **Ed**: questions/answers
  - Post all of your questions here
  - Questions must be "public" unless good reasons otherwise
  - **No code or copyable answers** in public posts!
- **Canvas**: submission of certain parts of assignments, grade book

Duke UNIVERSITY

Assumptions Going Into This Class

- I assume you want to be a software development professional
- I assume you are taking 650 (or have equivalent preparation)
  - You are competent C programmer (Mastery of 551 material)
  - You know basic systems concepts: caching, instructions, etc… (550)
  - If not in 650, you know or are learning:
    - Programming with pthreads
    - Networking
    - Relational databases (Postgresql, in particular)
- I assume you are eager to learn this material, and **write a bunch of code**
- I assume you can consult documentation, try things out, etc.
  - In lectures, we will learn cover concepts and practices
  - In assignments, you will be doing independent learning, experiments, etc.

Duke UNIVERSITY

What is this class about?

- Engineering Robust Server Software
  - **Software**: This class is all about software
    - Hardware may come up in regards to how it affects SW performance
  - **Engineering**: Designing and building systems
    - This is an engineering class, so expect to build a lot of software
    - Focus on useful things in real world
  - **Robust**: Stands up in the face of adversity
    - Badly formed user inputs, many requests at once, evil users…
  - **Server**: handles requests from clients
    - Different constraints from most programs you have written

Duke UNIVERSITY

Server Software

- Servers come in a wide range of "flavors"
- We are going to consider two major ones
  - UNIX daemons: sshd, httpd, …
    - C/C++, systems programming…
  - Web-sites: writing the server side logic for a website
    - Django, databases
- Three major themes
  - Security
  - Resilience
  - Scalability

Duke UNIVERSITY

Five Major Parts To Semester

- [1] Intro
  - Requirements/constraints/differences from other software
  - Protocols
  - Unix Daemons
  - DJango/website/AJAX basics
  - Containers

See course website for schedule details

Five Major Parts To Semester

- [2] Resilience
  - Error handling, exception models/safety
  - High-availability/disaster recovery

See course website for schedule details

Five Major Parts To Semester

- [3] Security
  - Cryptography basics
  - Common attacks/vulnerability types
    - (e.g., SQL injection, privilege escalation, …)
  - Famous vulnerabilities: Heartbleed, Dirty COW, Apple goto
  - Defense in Depth

See course website for schedule details

Interlude

- Midterm exam
- Spring break

See course website for schedule details

Five Major Parts To Semester

- [4] Performance/Scalability
  - Non-blocking IO
  - C++ atomics, memory model
  - Serialization bottlenecks
    - Locking granularity
    - "hidden" locks
  - Load balancing
  - Load testing
  - IO Scalability

See course website for schedule details

Duke UNIVERSITY

Five Major Parts To Semester

- [5] "Topics" Guest Lectures & Class Wrap-up
  - Intended to supplement the class topics
  - Also provide a perspective of real-world SW engineering

See course website for schedule details

Duke UNIVERSITY

What Will You Do?

- 4 Homeworks:
  - See course site for due dates
  - Programming
  - Different partner for each homework (groups of 2)
  - Thinking about and write down "dangers"
    - Revisit as semester progresses
- Example Assignments:
  - Simple Website (Django)
  - Caching Http Proxy (Unix Daemon in C)
  - Exchange Matching (Pair "Buy" with "Sell" orders)

Duke UNIVERSITY

What Will You Do? (cont'd)

- 1 Midterm
- 1 Final
- 1 Project
- Do in pairs (may select partner from prior homework)
  - Half class: e-commerce site ("Amazon")
  - Half class: shipping site ("UPS")
  - Systems have to interact

See course website for schedule details

"Danger" Log

- Critical programming skill: "spidey sense"
  - As you write, internal mental warning of danger
    - "What if the user ..."
    - "What if we run out of memory..."
    - "What if this fails..."
    - "What if..."
- As you code, think of these, write them down
  - Submit a text file with your thoughts
  - Particular focus on class themes (security, resilience, scalability)

THE USER MIGHT BE EVIL! OR WORSE, STUPID!
AND WHAT IF THE COMPUTER SUCKS?

Duke UNIVERSITY

"Danger" Log 2.0

- As you learn new things, revisit old assignments
  - Look at code:
    - What should you have worried about?
  - Look at danger logs:
    - What could you have done about these dangers?
- Update log with new thoughts ~weekly.

Pair Programming

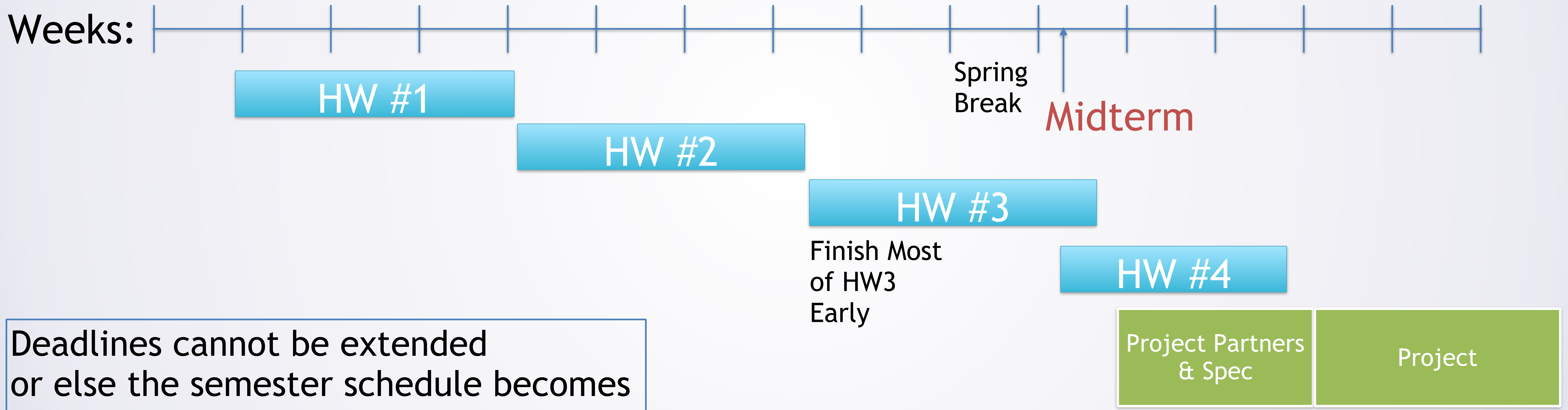
- Highly recommended development model: **pair programming**
  - Not just "doing assignment with a partner"
- Partners work on code at same time
  - One is "driver"
  - The other "navigator"
  - Switch roles frequently/as needed
- Driver: writes code
- Navigator: watches
  - Looks for errors, danger, thinks about bigger picture..

Pair Programming

- Useful tool: screen (or tmux)
    - Multiplex terminal session
    - Can have two terminals connected to one logical terminal
        - Both of you can look at, edit code from your own laptops
    - Facilitates switching driver/navigator
    - (Zoom is also fine too)
- Recommend to be on voice chat of some sort
    - Typing too slow (i.e. instant messaging)

Semester Timeline

- To help visualize how you will need to budget your time



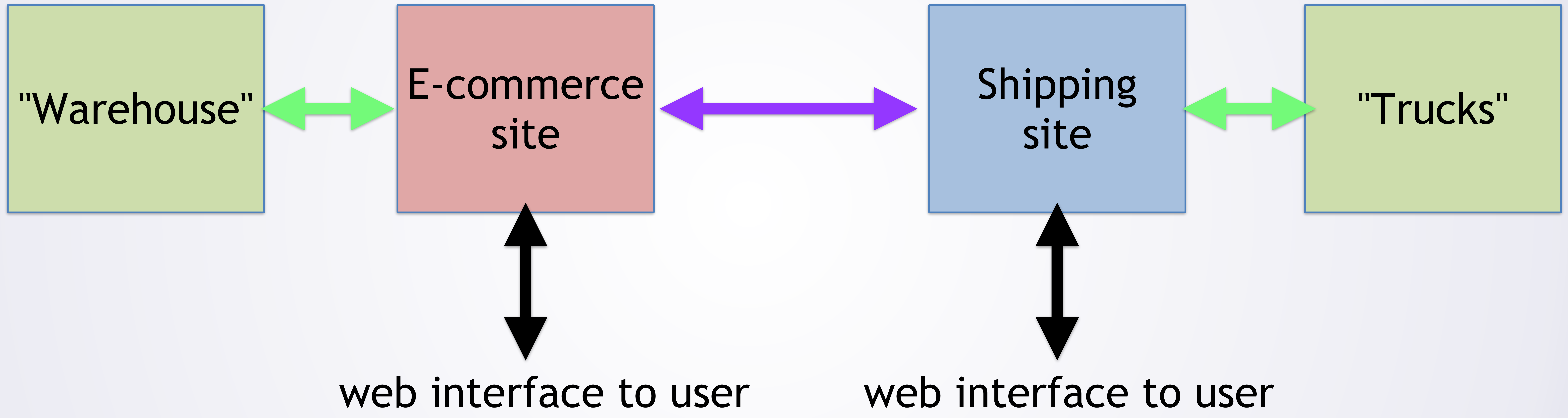
Deadlines cannot be extended
or else the semester schedule becomes
too compressed!

Duke UNIVERSITY

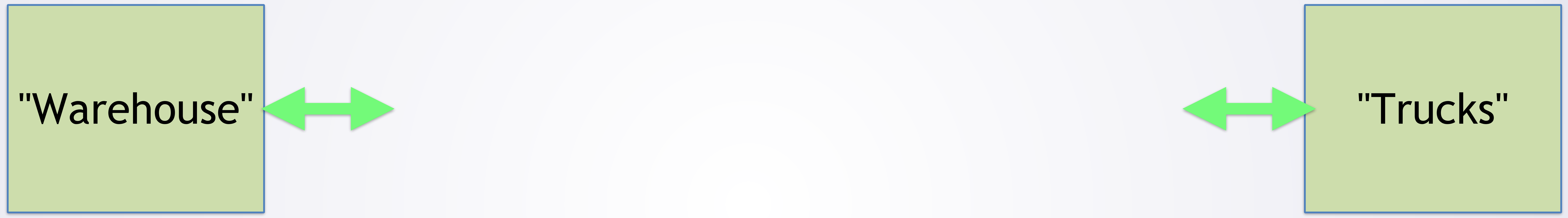
Recommended Assignment Practices

- Find partner & make a plan within 48 hours of assignment
  - Identify major pieces to implement or questions to answer
  - What steps are needed to architect, develop and test pieces?
  - What will testing strategy be? Creation of test cases in plan
- Make a schedule timeline based on this info
- Stick to the schedule!
  - Adjust timeline if things don't go exactly to plan
- **Learning happens best with steady, planned progress**
  - **You want to hit questions / issues early so you have time to think**
  - **You don't want to be tempted to take short-cuts**

Project: High-level View

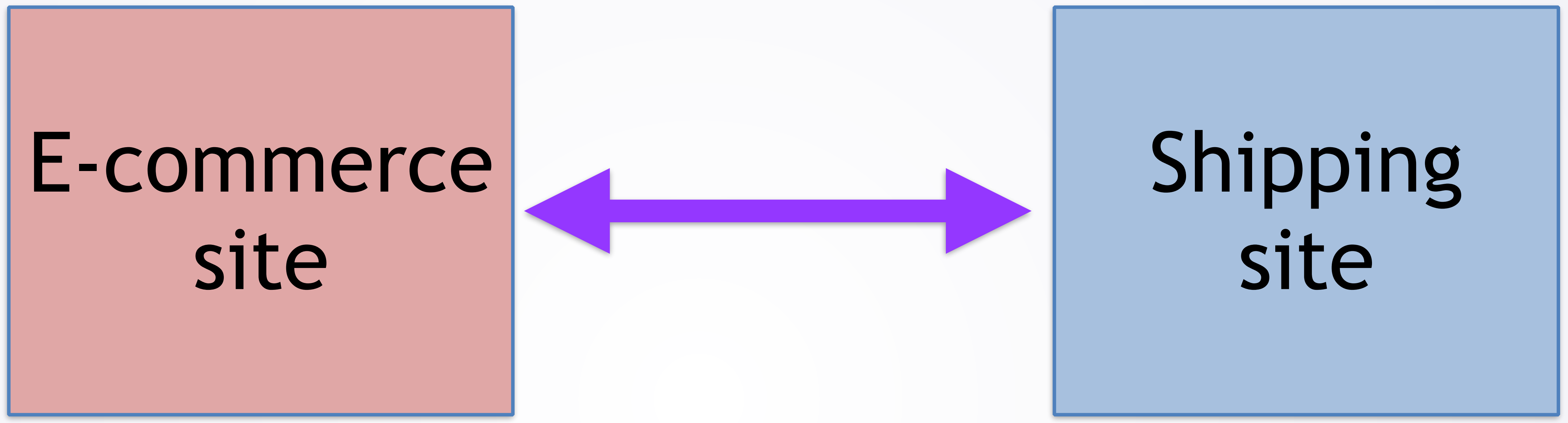


Project: High-level View



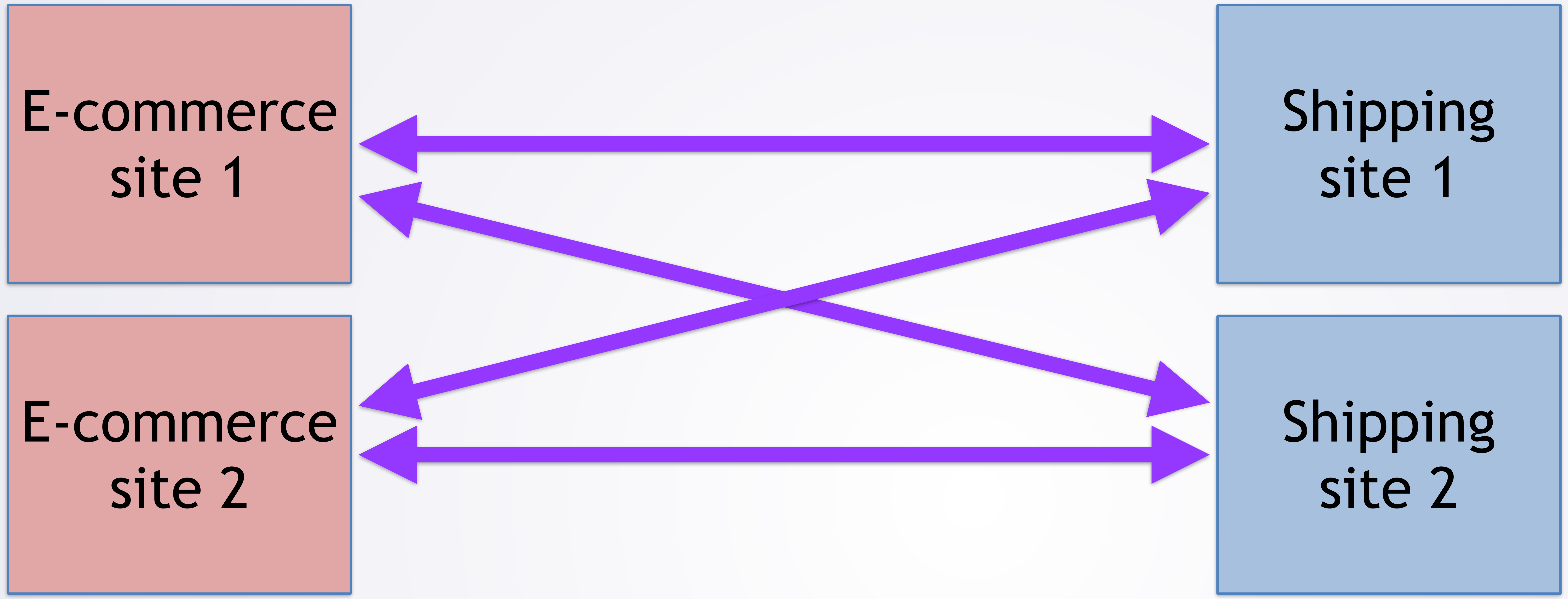
- I will define these protocols/implement these parts…
  - I'll give you a protocol spec
  - …but you should be resilient to **anything**
    - After all, that is a goal of this class

Project: High-level View



- You will do either the red (e-commerce) or the blue (shipping)
  - Protocol between them? Defined by your **interoperability** group

Project: High-level View



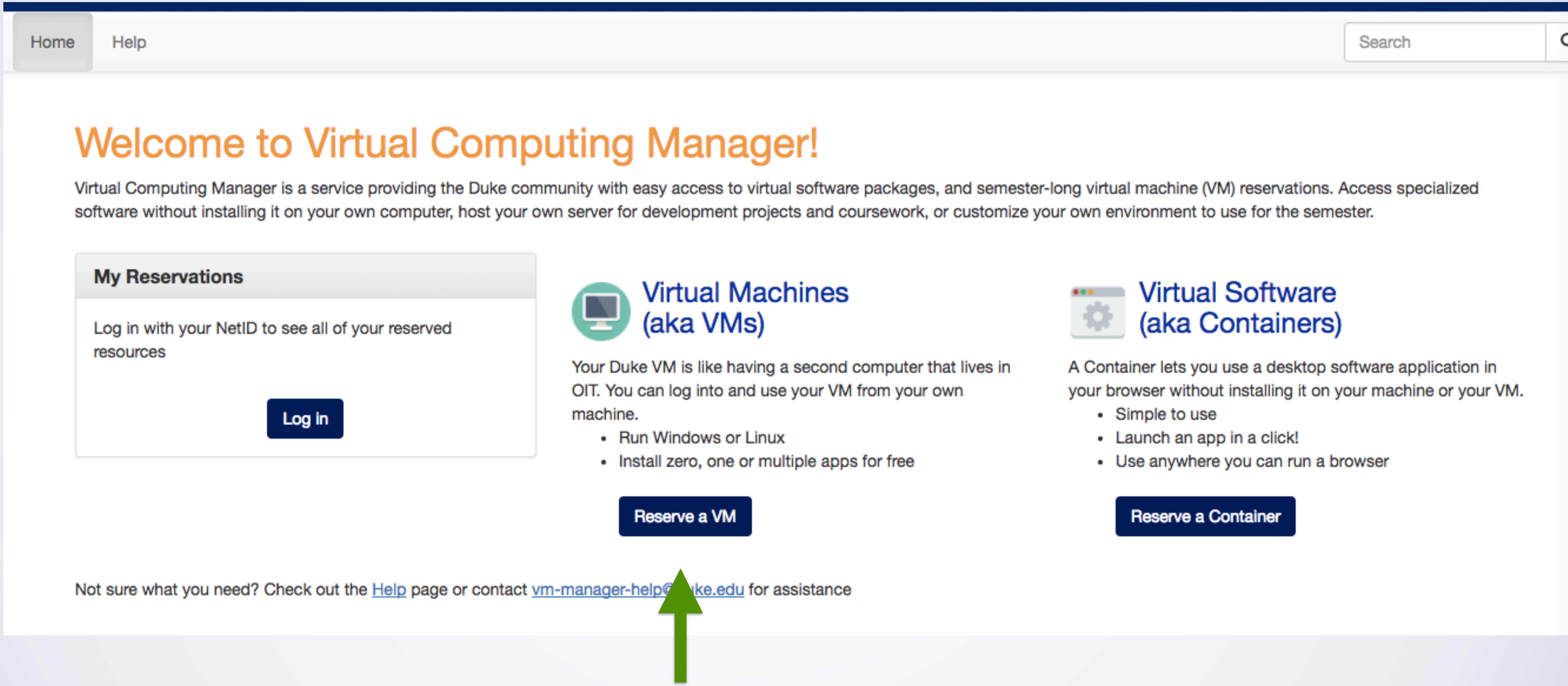
- 4 groups (8 people) = 1 interoperability group
  - Both e-commerce sites must work with both/either shipping site.
  - 8 of you define protocol

Where will you do it?

- You will each have your own server
  - You get root on it, you administer it
- OIT VMs
  - Go to https://vcm.duke.edu/
  - Select an Ubuntu machine (Ubuntu 20.04)
  - Login with netid
- Initially will use the VM you create
  - Later in the semester (HW #4 and Project) we'll use some multi-core VMs allocated for course (you'll select "ECE 568 Spring 2025")

Duke UNIVERSITY

Choose Reserve a VM in Lower Middle



Welcome to Virtual Computing Manager!

Virtual Computing Manager is a service providing the Duke community with easy access to virtual software packages, and semester-long virtual machine (VM) reservations. Access specialized software without installing it on your own computer, host your own server for development projects and coursework, or customize your own environment to use for the semester.

My Reservations

Log in with your NetID to see all of your reserved resources

Virtual Machines (aka VMs)

Your Duke VM is like having a second computer that lives in OIT. You can log into and use your VM from your own machine.

- Run Windows or Linux
- Install zero, one or multiple apps for free

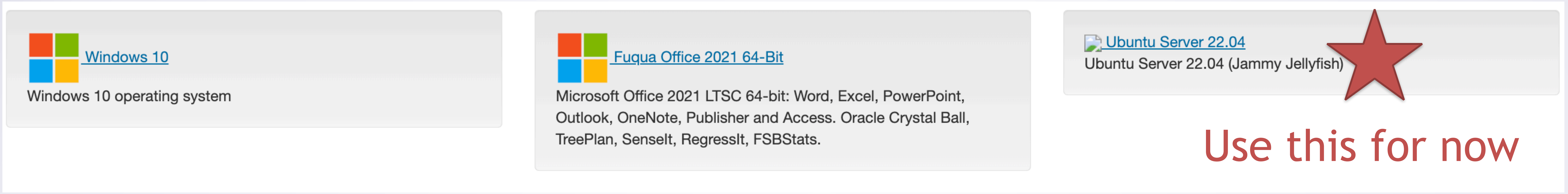
Virtual Software (aka Containers)

A Container lets you use a desktop software application in your browser without installing it on your machine or your VM.

- Simple to use
- Launch an app in a click!
- Use anywhere you can run a browser

Not sure what you need? Check out the Help page or contact vm-manager-help@duke.edu for assistance

Reserve a VM



Windows 10

Windows 10 operating system

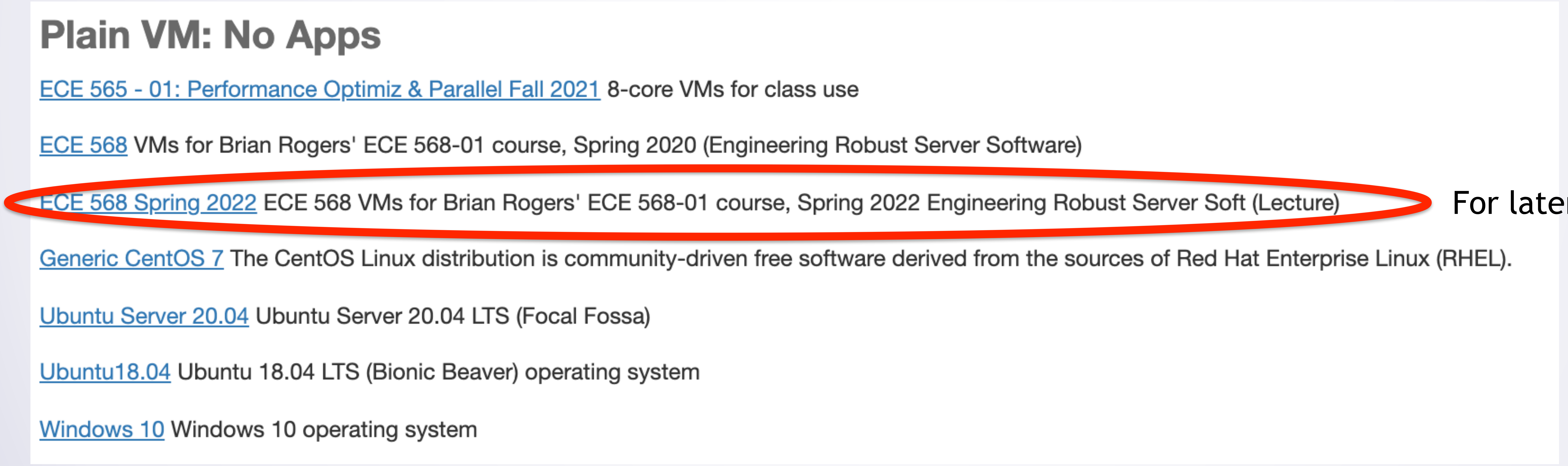
Fuqua Office 2021 64-Bit

Microsoft Office 2021 LTSC 64-bit: Word, Excel, PowerPoint, Outlook, OneNote, Publisher and Access. Oracle Crystal Ball, TreePlan, SenseIt, RegressIt, FSBStats.

Ubuntu Server 22.04

Ubuntu Server 22.04 (Jammy Jellyfish)

Use this for now



Plain VM: No Apps

ECE 565 - 01: Performance Optimiz & Parallel Fall 2021 8-core VMs for class use

ECE 568 VMs for Brian Rogers' ECE 568-01 course, Spring 2020 (Engineering Robust Server Software)

ECE 568 Spring 2022 ECE 568 VMs for Brian Rogers' ECE 568-01 course, Spring 2022 Engineering Robust Server Soft (Lecture)

For later in the semester

Generic CentOS 7 The CentOS Linux distribution is community-driven free software derived from the sources of Red Hat Enterprise Linux (RHEL).

Ubuntu Server 20.04 Ubuntu Server 20.04 LTS (Focal Fossa)

Ubuntu18.04 Ubuntu 18.04 LTS (Bionic Beaver) operating system

Windows 10 Windows 10 operating system

Next Steps

- Login to your server
  - Username/password: NetID of the creator of it
- Setup a user account w/ sudo
  - sudo adduser *name*
  - sudo adduser *name* sudo
- Now you can ssh in as *name*
- Recommended: setup ssh key pair
  - https://vcm.duke.edu/help/23

Install Software!

.ssh — drew@ubuntu14-generic-template-01: ~ — ssh — 90×28

```
drew@ubuntu14-generic-template-01:~$ gcc
The program 'gcc' is currently not installed. You can install it by typing:
sudo apt install gcc
drew@ubuntu14-generic-template-01:~$
```

- Your server: fresh image, not much software installed
  - sudo apt-get install *package*

Packages you probably want to install

- For C development: **gcc g++ make valgrind**
- For editing: **emacs screen**
- For source control: **git**
- Database: **postgresql libpq-dev**
- For Django: **python python3-pip**
  - Then do: **sudo pip3 install django psycopg2**
  - Then **django-admin --version** should give 5.1.4
- Libraries: **libssl-dev libxerces-c-dev libpqxx-dev**
- Documentation: **manpages-posix-dev**

Note when working through the Django tutorial, where it mentions using ‘python’ commands, you may need to use ‘python3’

Recommended Server Setup [Optional]

- Set up your "dot files"
  - ~/.emacs : emacs configuration
  - ~/.profile : commands read on login

```
export EDITOR='emacs -nw'
export VISUAL='emacs -nw'
```

- Setup ssh key pair(s)
  - Login without password: private key authenticates
- Pick somewhere to backup your work
  - Keep a git remote on **another** computer

Duke UNIVERSITY

Grading

- Grade Breakdown:
  - Homeworks: 28%
  - Project: 22%
  - Midterm: 20%
  - Final: 30%

Regrades:
* All regrade requests must be in writing
* After getting feedback with the TA, if you still have concerns, contact the instructor
* All regrade requests must be submitted no later than 1 week after the assignment was returned to you.

- Letter grade:

A- [90,93)	A [93,97)	A+ [97,∞)
B- [80,83)	B [83,87)	B+ [87,90)
C- [70,73)	C [73,77)	C+ [77,80)
D- [60,63)	D [63,67)	D+ [67,70)
	F (-∞,70)	

Late Assignment Submission

- Increasing penalty for each day late
  - 0-24 hours late: score * 0.9
  - 24-48 hours late: score * 0.8
  - 48-72 hours late: score * 0.6
- Allows moderate penalty for small amount of lateness
- No submissions accepted after 3 days late (72 hours)
  - We need to have all assignments by then for grading timeliness
  - You'll need to be free to move on to the next assignment

Duke
UNIVERSITY

Academic Integrity

- Your work is expected to be your own (for midterm & final exam)
- Or yours and your partner's (for homework / project)
- Do not be tempted to use or look for unreasonable help (i.e. code of other groups or code found online)
  - If unsure, ask me
- Do not look at or use code you may find online which implements the same thing or something very close to what you are implementing
  - We use code similarity checking tools to do automated checking for this
  - These tools are very good at flagging irregularities
  - This is another reason to start early and make steady progress!
    - Do not be tempted to take shortcuts

RFCs

- Many standards are in the form of RFCs
- You SHOULD spend some time reading RFCs this semester
  - …and may effectively write one during your project
- Start with this one (describes MUST/MAY/SHOULD etc in RFCs)
  - https://tools.ietf.org/html/rfc2119

Next Time..

- Let’s look at class page…
- Wrap up for this time:
  - Questions?
  - Find partners for homework 1
    - Use “find a teammate” pinned post on Ed Discussion
  - Look at Django and Docker tutorials
- Next time:
  - Start talking about server software

Duke UNIVERSITY