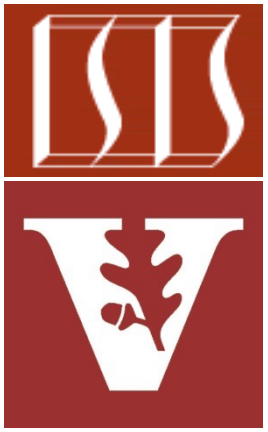
CS 891: Scalable Microservices: Overview (Part 1)

Douglas C. Schmidt

d.schmidt@vanderbilt.edu

www.dre.vanderbilt.edu/~schmidt



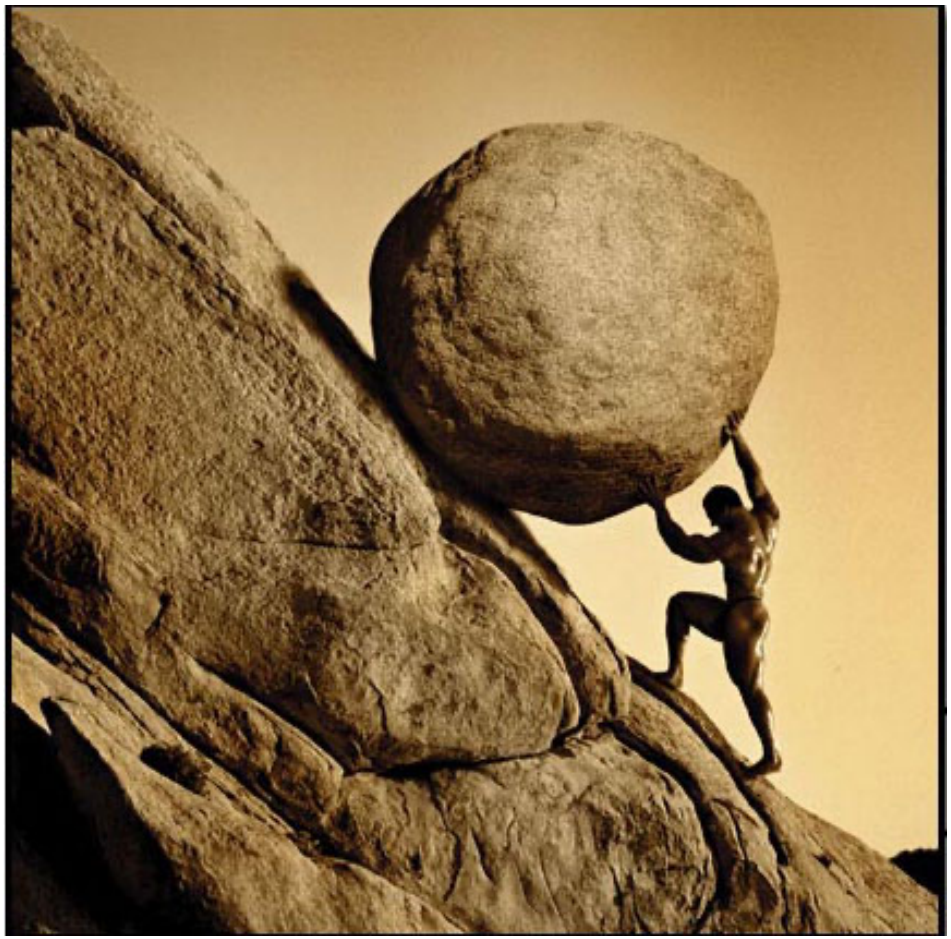
Professor of Computer Science

Institute for Software Integrated Systems

Vanderbilt University
Nashville, Tennessee, USA

Learning Objectives in this Lesson

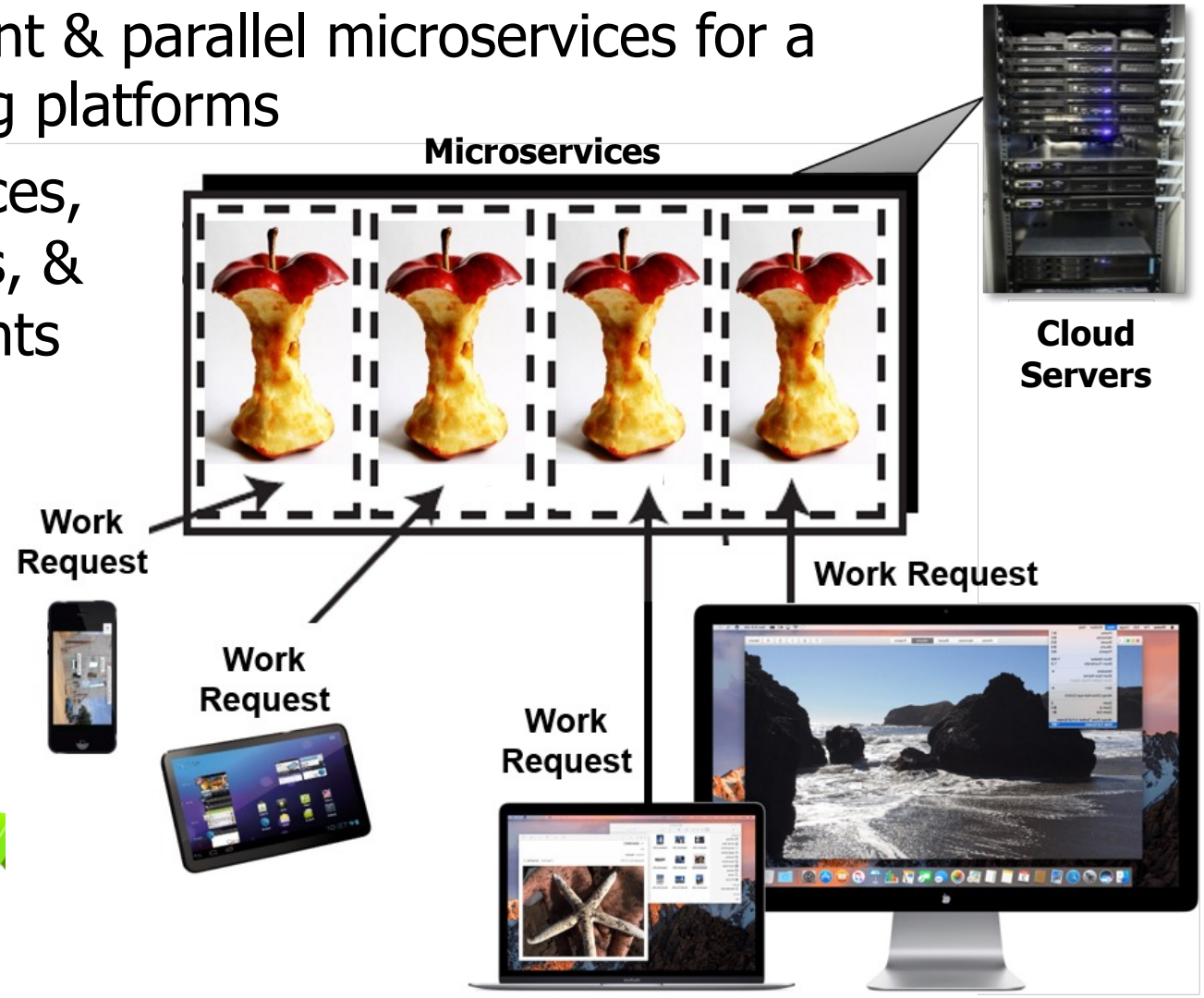
- Understand the course topics & logistics
  - Course philosophy
  - Course contents
  - Structure of the lecture material



Course Philosophy

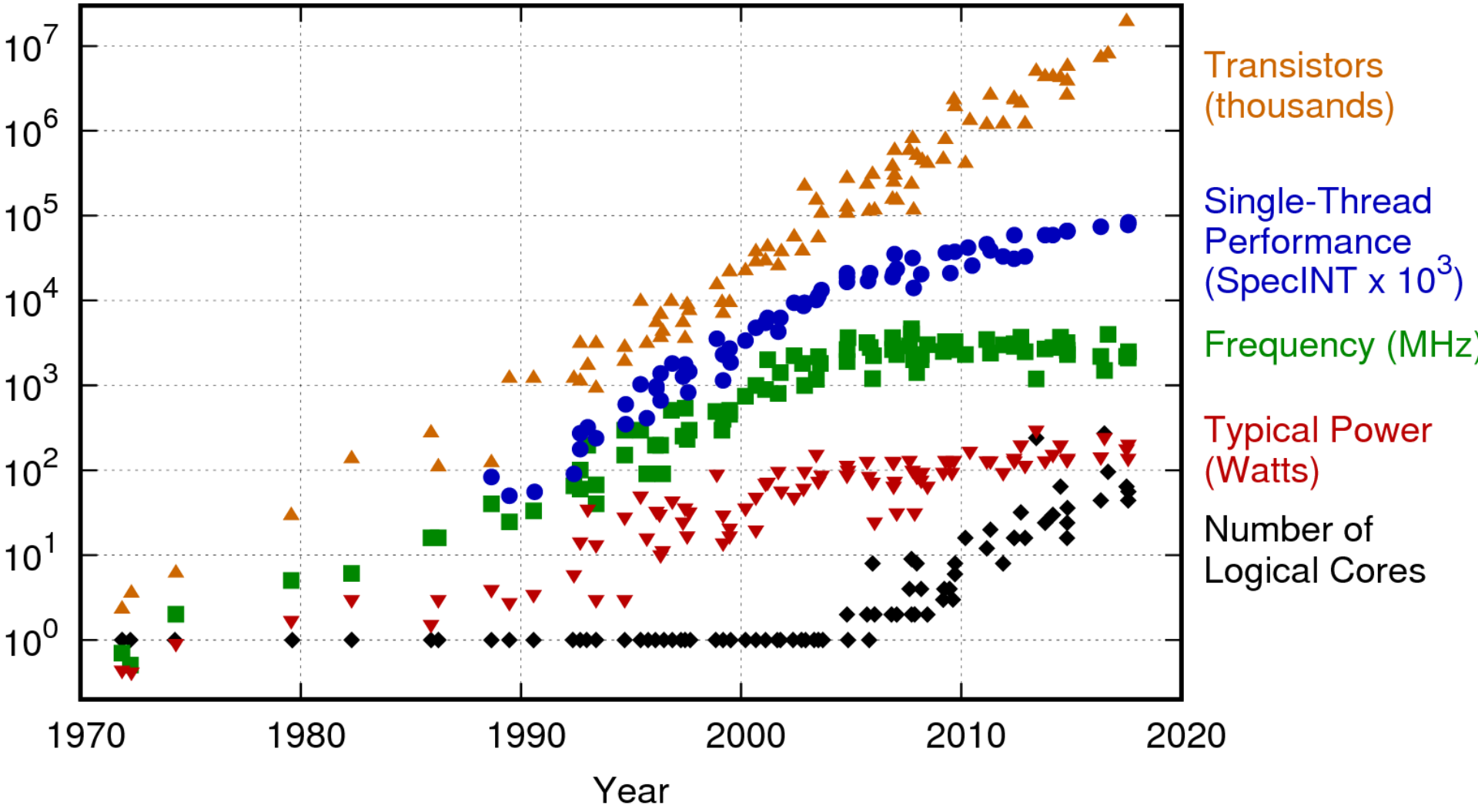
Course Philosophy

- There's a growing need for software developers who can write concurrent & parallel microservices for a range of computing platforms
  - e.g., mobile devices, laptops, desktops, & cloud environments



Course Philosophy

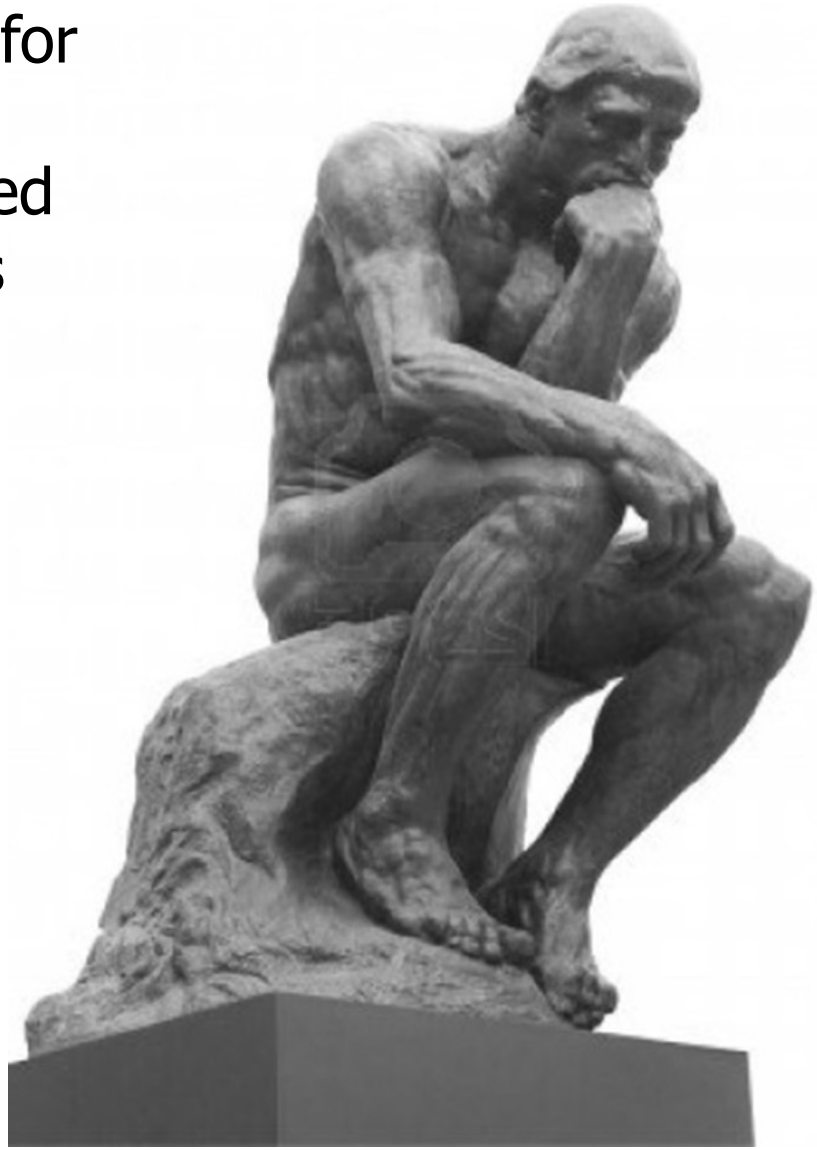
- Demand is driven by software/hardware infrastructure advances



See www.gotw.ca/publications/concurrency-ddj.htm

Course Philosophy

- Effective techniques & practices for developing concurrent & parallel microservices are *not* best learned through generalities & platitudes



"Sitting & thinking" is not sufficient…

Course Philosophy

- Instead, it's better to see *by example* how these programs can be made
  - *easier* to write & read,
  - *easier* to maintain & modify,
  - *more* efficient & resilient

  by applying time-proven software patterns & object-oriented, functional, & reactive design & programming techniques

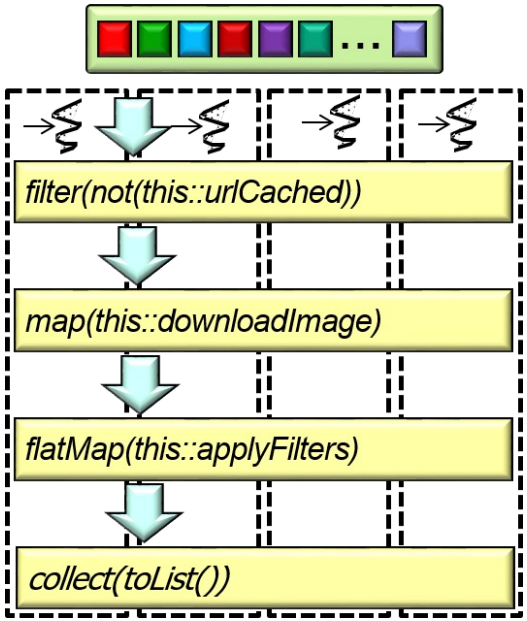
This course involves lots of hands-on software development & testing!

Summary of the Course Contents

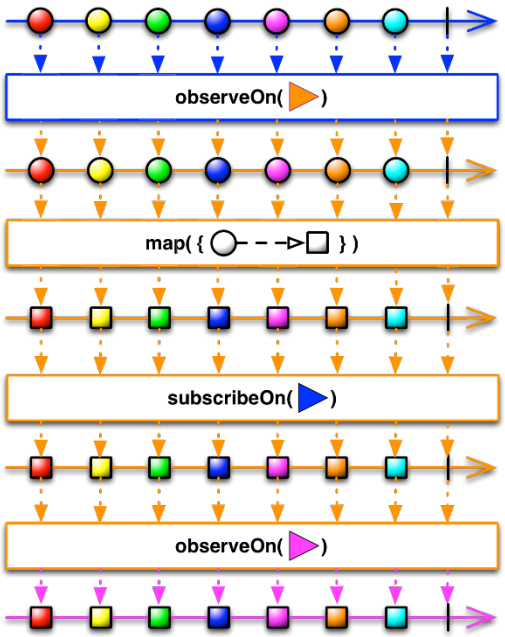
Summary of Course Contents

- Key Java frameworks

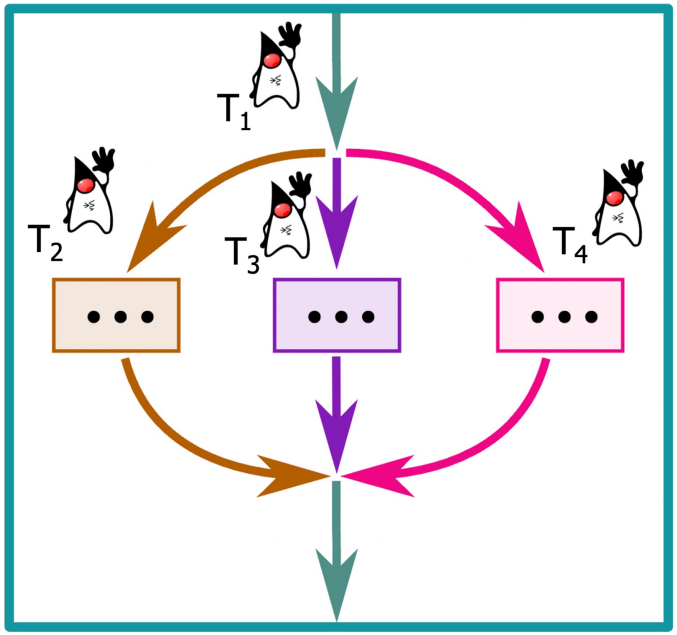
Streams



Reactive Streams



Structured Concurrency

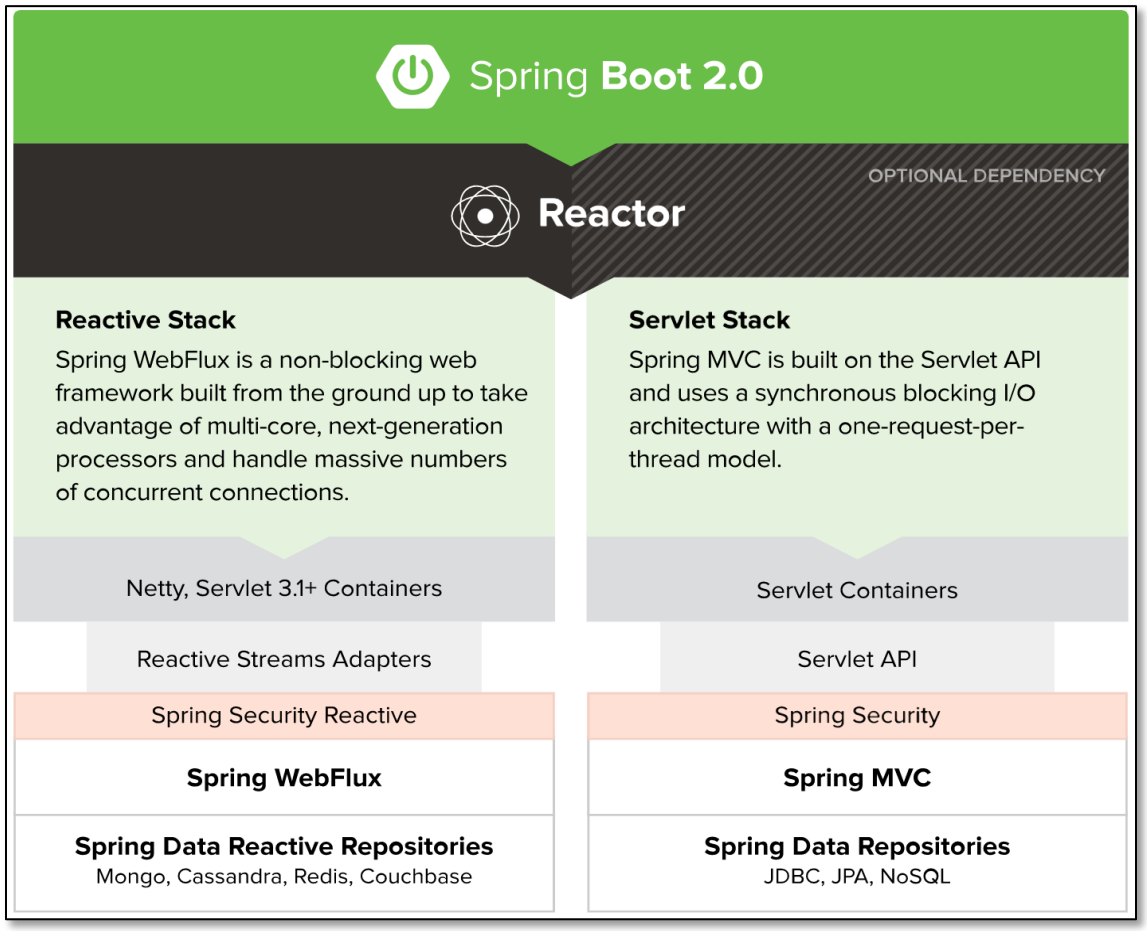


Assumes knowledge of Java object-oriented & functional language features

Summary of Course Contents

- Key Java frameworks
- Modern web programming platforms

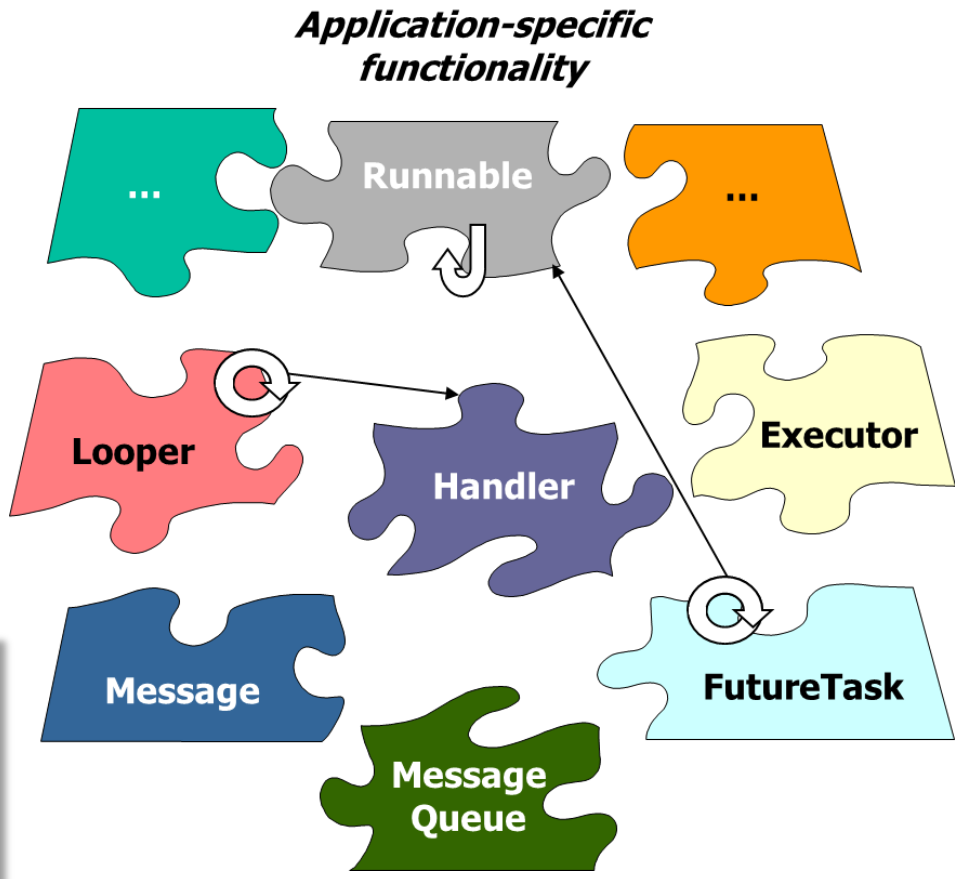
Spring WebMVC & WebFlux

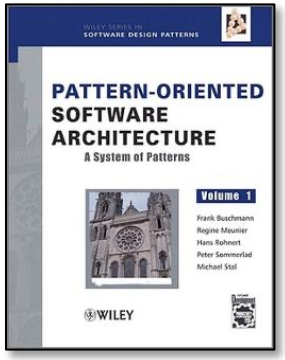


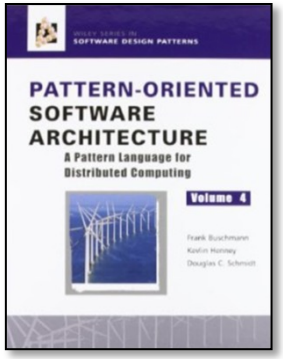
See spring.io/projects/spring-boot

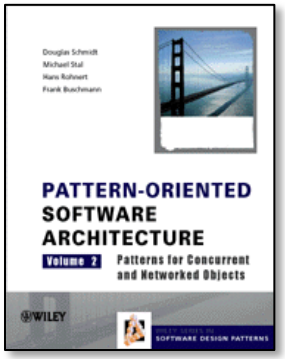
Summary of Course Contents

- Key Java frameworks
- Modern web programming platforms
- Patterns for concurrent & parallel programming





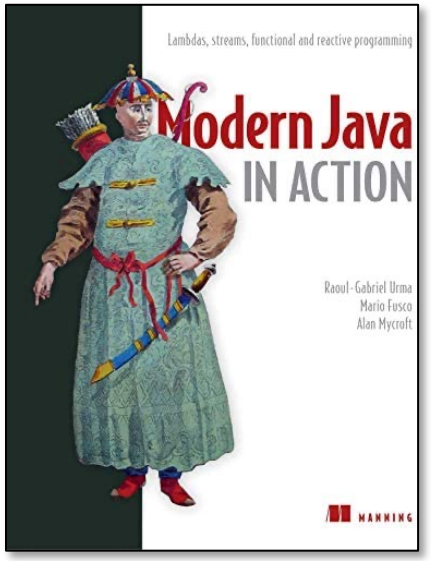


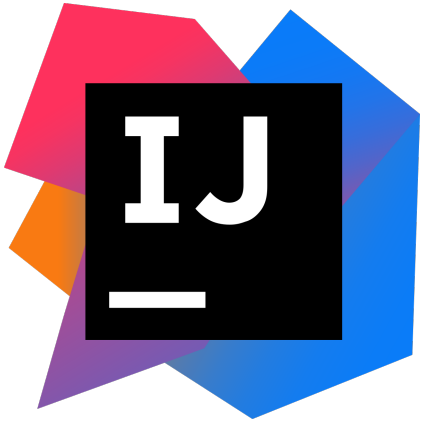


See www.dre.Vanderbilt.edu/~Schmidt/POSA

Summary of Course Contents

- Key Java frameworks
- Modern web programming platforms
- Patterns for concurrent & parallel programming
- We assume you know (or can quickly learn) modern Java, IntelliJ, & Git





See item #12 at github.com/douglascraigschmidt/CS891/wiki/CS-891-FAQ

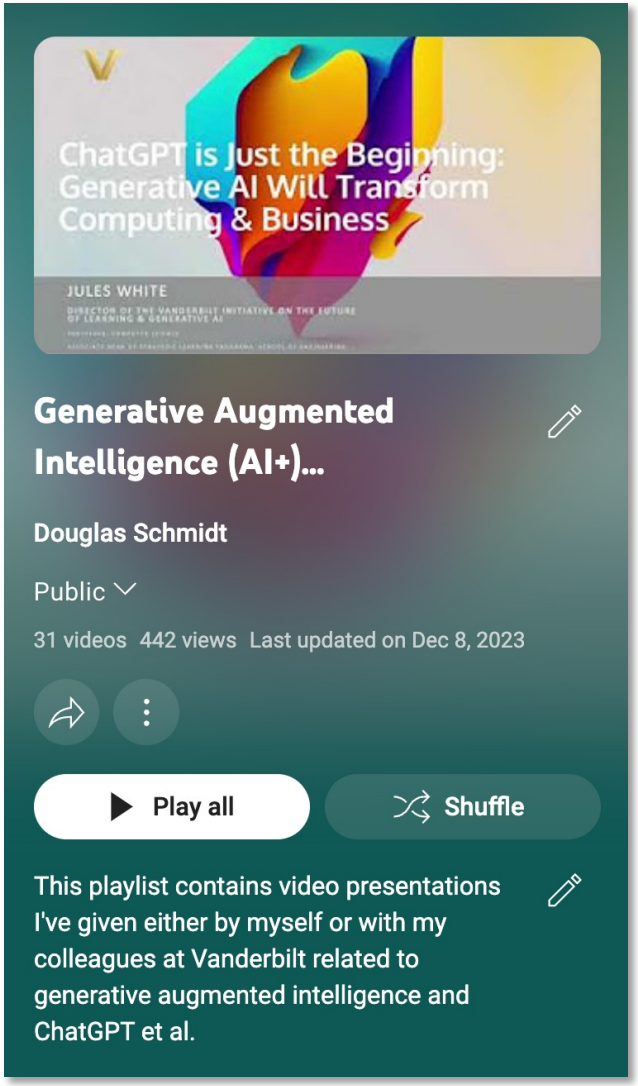
Summary of Course Contents

- Key Java frameworks
- Modern web programming platforms
- Patterns for concurrent & parallel programming
- We assume you know (or can quickly learn) modern Java, IntelliJ, & Git
- We'll apply large-language models (LLMs) throughout this course where appropriate

See cdn.vanderbilt.edu/vu-URL/wp-content/uploads/sites/241/2023/08/16143452/Vanderbilt-University-Academic-Affairs-Guidance-for-Artificial-Intelligence.pdf

Summary of Course Contents

- Key Java frameworks
- Modern web programming platforms
- Patterns for concurrent & parallel programming
- We assume you know (or can quickly learn) modern Java, IntelliJ, & Git
- We'll apply large-language models (LLMs) throughout this course where appropriate
  - LLMs are having a massive impact on education & workforce productivity



See www.youtube.com/playlist?list=PLZ9NgFYEMxp72Zo0yrTNS6utAXxYpqNGl

Structure of the Lecture Material

Structure of the Lecture Material

- This course has three main modules

Section	Topics

Java Structured Concurrency & Reactive Streams	• Coverage of modern Java concurrency & parallelism frameworks, e.g.
	  • Java structured concurrency
	  • Reactive streams (e.g., Project Reactor)

Structure of the Lecture Material

- This course has three main modules

Section	Topics

Java Structured Concurrency & Reactive Streams	• Coverage of modern Java concurrency & parallelism frameworks, e.g.
	  • Java structured concurrency
	  • Reactive streams (e.g., Project Reactor)

Sync & Async Web Communication	• Spring WebMVC & WebFlux

Structure of the Lecture Material

- This course has three main modules

Section	Topics

Java Structured Concurrency & Reactive Streams	• Coverage of modern Java concurrency & parallelism frameworks, e.g.
	  • Java structured concurrency
	  • Reactive streams (e.g., Project Reactor)

Sync & Async Web Communication	• Spring WebMVC & WebFlux
Software Patterns	• Concurrent & parallel programming & communication patterns

Structure of the Lecture Material

- This course has three main modules
  - Each module is composed of lessons

Structure of the Lecture Material

- This course has three main modules
  - Each module is composed of lessons
  - Each lesson is composed of parts

Structure of the Lecture Material

- This course has three main modules
  - Each module is composed of lessons
  - Each lesson is composed of parts
  - Each part is a single lecture

Screencasts of each lesson "part" & PDF versions of the slides will be uploaded to www.dre.vanderbilt.edu/~schmidt/cs891#lectures

Structure of the Lecture Material

- This course has three main modules
  - Each module is composed of lessons
  - Each lesson is composed of parts
  - Each part is a single lecture
    - Each part is composed of segments

Structure of the Lecture Material

- There will be bi-weekly quizzes on material covered in the lectures

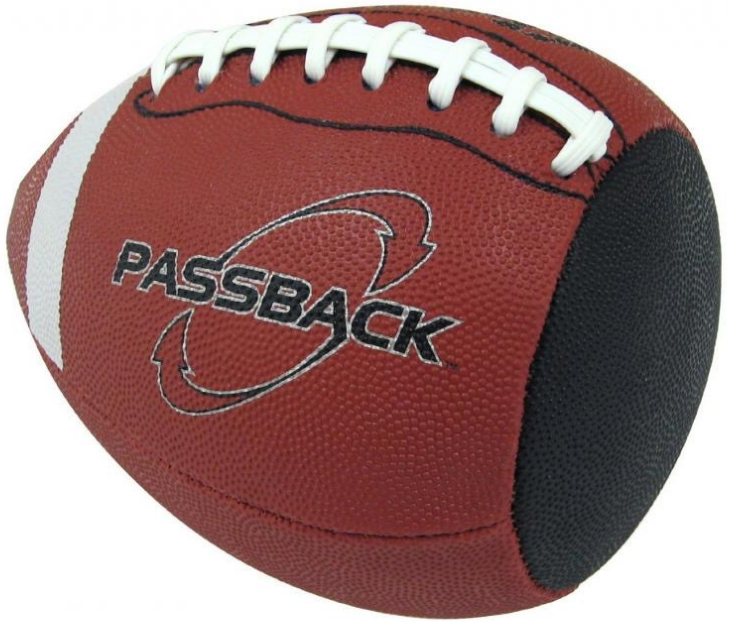
Structure of the Lecture Material

- There will be bi-weekly quizzes on material covered in the lectures
  - 1st quiz will be on Wednesday, January 17th

All quizzes are "closed book/note/Internet" & are given on Brightspace

Structure of the Lecture Material

- There will be bi-weekly quizzes on material covered in the lectures
  - 1st quiz will be on Wednesday, January 17th
  - We strive to hand back & review quizzes at the start of next class



One of the benefits of a smaller class ;-)

Structure of the Lecture Material

- There will be bi-weekly quizzes on material covered in the lectures
  - 1st quiz will be on Wednesday, January 17th
  - We strive to hand back & review quizzes at the start of next class

I recommend that you study for quizzes by reviewing slides & watching screencasts available at www.dre.vanderbilt.edu/~schmidt/cs891#lectures

Structure of the Lecture Material

- There *may* be a cumulative final exam that covers all the lectures
  - The focus will be on the last week(s) of the semester

The final exam *may* be held 9am-12pm, Wednesday, April 24th via Brightspace

CS 891: Scalable Microservices: Overview (Part 1)