Carnegie Mellon University
Master of
Software Engineering

17-647: Engineering Data Intensive Scalable Systems
MW 10:40 -12:00pm, Remote
Course Zoom Link:
https://cmu.zoom.us/j/99112551608?pwd=Wk9pRGNSelMzSnBreUNacVpVSFpBQT09
Meeting ID: 991 1255 1608
Passcode: 857930
[A4, Spring 2021, 6 Units]

Last Update: 3/20/21

Instructor	Email	Office Location & Hours
Matthew Bass	mbass@andrew.cmu.edu	SCR 267 Zoom ID: 3609899473 By appointment
Paulo Merson	pmerson@andrew.cmu.edu	

Teaching Assistants

Ariel Xiao	**yuweix@andrew.cmu.edu**	**TBD**

Course Description. Internet services companies such as Google, Yahoo!, Amazon, and Facebook, have pioneered systems that have achieved unprecedented scale while still providing high level availability and a high cost-performance. These systems differ from mainstream high performance systems in fundamental ways. They are data intensive rather than compute intensive as we see with mainstream super computers spending the bulk of their time performing data I/O and manipulation rather than computation. They need to inherently support scalability, typically having high reliability and availability demands as well. Given that they often operate in the commercial space the cost-performance of these systems needs to be such that the organizations dependent on such systems can turn a profit.
Designing and building these systems require a specialized set of skills. This course will cover the set of topics needed in order to design and build data intensive scalable systems. In this domain engineers not only need to know how to architect systems that are inherently scalable, but to do so in a way that also supports high availability, reliability, and performance. Given the large distributed nature of these systems basic distributed systems concepts such as consistency and time and synchronization are also important. These systems largely operate around the clock, placing an emphasis on operational concerns. This course will introduce students to these concerns with the intent that they understand the extent to which things like deploying, monitoring, and upgrading impact the design.

The course will be a hands-on project oriented course. The basic concepts will be given during the lectures and applied in the project. The students will gain exposure to the core concepts needed to design and build such systems as well as current technologies in this space. Class size will be limited.

Learning Objectives: Students in this class will learn what it takes to engineer systemic properties into data intensive distributed systems. Namely, we will focus on:

- Scalability
- Availability
- Response Time
- Distributed Data
- Consistency
- Compute Cost
- Intelligent Systems

In support of these goals students will be introduced to some basic distributed systems and software architecture theory as well as relevant technologies. This is not a technology or theoretical course, however. Students will not be tested on their knowledge of theory or technology, however, rather they will be expected to demonstrate the application of this knowledge through the construction of systems with increasing expectations.
In addition to learning how to optimize on these properties in isolation, students will learn about the interplay among these concerns. Managing tradeoffs appropriately is often one of the more difficult engineering challenges and will be the focus of later projects in this course.

Prior Knowledge. Students are expected to be familiar with programming in at least one, preferably object-oriented, programming language. Formal training in Software Architecture is helpful, but not required.

Students should have some experience writing small programs or software applications. Students in doubt regarding their experience should obtain instructor's permission.

Learning Resources. The course and all course materials will be distributed online and accessible with a CMU account via Canvas.

Assessments. Students learn more by applying and explaining ideas to others, thus, the course requires the following activities:

- **Lecture and Reading Assessments:** These are short online questions derived from the required readings and lectures
- **Class participation:** There are exercises students will do in recitations in pairs or small groups to practice applying the concepts learned in the course as well as participation in online discussions in Piazza

- **Individual Homework Assignments:** These will be primarily programming and reflection assignments based on the concepts learned throughout the course

Assessment	Final Grade %
Quizzes	30%
Individual Homework Assignments	60%
Class participation	10%

Course and Grading Policies

- **Late-work policy**: All work is expected to be handed in at the indicated due date and time. For fairness to the whole class, no late submissions or makeups will be accepted for the Quizzes. We will, however, drop the lowest grade. The penalty for turning in Individual Homeworks late is 10%/day. In the first week of classes, you should receive a course schedule for each course; please use them to plan ahead. If you have any questions you should raise them immediately rather than waiting for conflicts to arise.
- **Participation policy**. Class participation will be graded by in-class engagement, including asking relevant questions based on a critical review of required readings and lectures, preparation for any in-class exercises, and responses on the class discussion board. The lack of attendance and participation, will count against your participation grade.

Week	Topics	Assignments	Notes

1	• SOA and Microservice architecture styles
	  ○ Characterization of these styles vis-a-vis other C&C styles
	  ○ Overall benefits and challenges (tradeoffs)
	• Types of components and connectors in the Microservice style
	  ○ REST vs component technology (gRPC, GraphQL, etc.)
	  ○ REST API design
	  ○ Asynchronous messaging (producers, consumers, queue/topics)
	  ○ Service interceptors
		• A1 given	Reading:
			• [Merson15b]
			• [Merson15a]

2	• Core principles for microservice design: “IDEALS” • API Gateway, BFF, service mesh, circuit breaker	• A1 Due (wednesday before classtime) • A2 Given	Reading: • [Merson21] • [Newman15b] • [Richardson 18] excerpt (Section 8.2 and subsections 8.2.1 and 8.2.2) Design and implementation assignment: REST services

3	• Messaging patterns used in microservice architectures
	  ○ Building blocks: point-to-point, pub-sub
	  ○ For reliability: store-and-forward, transactional outbox, dead-letter channel
		• A2 Due (wednesday before classtime)
		• A3 Given
			Reading:
			• [Hohpe03] excerpt about Request-Reply
			• [Hohpe03]

	○ Other: sync-over-async, push vs pull ● Event-Driven Architecture (EDA) and its tradeoffs ○ Choreography and orchestration ● Asynchronous without a message broker		excerpt about Dead Letter Channel ● [Richardson 18] excerpt (section 3.3.7) Design and implementation assignment: EDA using SNS or Kafka, transactional outbox, and dead-letter topic
4	● Scalability Defined ● Post-SQL Data Stores ● CAP theorem	● A3 Due (wednesday before classtime) ● A4 Given	
5	● Designing Post-SQL data models ● Data Processing Pipelines	● A4 Due (wednesday before classtime) ● A5 Given	
6	● Strategies for managing data ○ Service autonomy and distributed transactions ○ Saga pattern (compensation) ○ Database per Microservice pattern ○ Data replication and eventual consistency ● CQRS pattern	● A5 Due (wednesday before classtime) ● A6 Given	Reading: ● [Richardson 18] Section 4.1.3 and 4.2 (including 4.2.1) ● “Pattern: Database per service” Design and implementation assignment: event-driven Saga
7	● Service loose coupling	● A6 due	Reading:

	• Façade pattern (within the service) • Wrapper service • Moving from a monolithic to a microservice architecture ○ Strangler Fig pattern and its variations	(wednesday before classtime)	• [Yoder20] Only these sections: Introduction, Replace as Microservice, Extract Component and Add Façade, Proxy Monolith Components to Microservices

Course Schedule. The following schedule provides a general overview of topics and assignments. Please refer to the syllabus online in Canvas for specific lecture topics, reading assignments and due dates.

Accommodations for Students Disabilities. If you have a disability and have an accommodations letter form the Disability Resources office, I encourage you to discuss your accommodations and needs with me as early in the semester as possible. I will work with you to ensure that accommodations are provided as appropriate. If you suspect that you may have a disability and would benefit from accommodations but are not yet registered with the Office of Disability Resources, I encourage you to contact them at access@andrew.cmu.edu.

Academic Integrity. Honesty and transparency are important to good scholarship. Plagiarism and cheating, however, are serious academic offenses with serious consequences. If you are discovered engaging in either behavior in this course, you will earn a failing grade on the assignment in question, and further disciplinary action may be taken.

For a clear description of what counts as plagiarism, cheating, and/or the use of unauthorized sources, please see the University's Policy on Academic Integrity.

If you have any questions regarding plagiarism or cheating, please ask me as soon as possible to avoid any misunderstandings. For more information about Carnegie Mellon's standards with respect to academic integrity, you can also check out the Office of Community Standards & Integrity website.

Student Wellness. As a student, you may experience a range of challenges that can interfere with learning, such as strained relationships, increased anxiety, substance use, feeling down, difficulty concentrating and/or lack of motivation. These mental health concerns or stressful events may diminish your academic performance and/or reduce your ability to participate in daily activities. CMU services are available, and treatment does work. You can learn more about confidential mental health services available on campus at the Counseling and Psychological Services website. Support is always available (24/7) from Counseling and Psychological Services: 412-268-2922.

We must treat every individual with respect. We are diverse in many ways, and this diversity is fundamental to building and maintaining an equitable and inclusive campus community. Diversity can refer to multiple ways that we identify ourselves, including but not limited to race, color, national origin, language, sex, disability, age, sexual orientation, gender identity, religion, creed, ancestry, belief, veteran status, or genetic information. Each of these diverse identities, along with many others not mentioned here, shape the perspectives our students, faculty, and staff bring to our campus. We, at CMU, will work to promote diversity, equity and inclusion not only because diversity fuels excellence and innovation, but because we want to pursue justice. We acknowledge our imperfections while we also fully commit to the work, inside and outside of our classrooms, of building and sustaining a campus community that increasingly embraces these core values.
Each of us is responsible for creating a safer, more inclusive environment.
Unfortunately, incidents of bias or discrimination do occur, whether intentional or unintentional. They contribute to creating an unwelcoming environment for individuals and groups at the university. Therefore, the university encourages anyone who experiences or observes unfair or hostile treatment on the basis of identity to speak out for justice and support, within the moment of the incident or after the incident has passed. Anyone can share these experiences using the following resources:

- **Center for Student Diversity and Inclusion:** csdi@andrew.cmu.edu, (412) 268-2150
- **Report-It online anonymous reporting platform:** reportit.net username**:** *tartans* password**:** *plaid*

All reports will be documented and deliberated to determine if there should be any following actions. Regardless of incident type, the university will use all shared experiences to transform our campus climate to be more equitable and just.