

# Machine Learning in Production Motivation, Syllabus, and Introductions



# Slack

We use Slack for this course, including during lectures

See signup link on Canvas

Setup the ability to read/post to Slack during lecture

# Catastrophic Success

# The Waitlist Situation



Best guess: Most people will get in, but it may take a few days

For those joining late:

- Ask us for recording of missed lectures on Slack
- Post introduction on Slack (#intro) when joining
- If joining after Jan 25, automatic 5 day extension for Homework I1
- Automatically excused for participation in missed lectures

# Learning Goals

- Understand how ML components are parts of larger systems
- Illustrate the challenges in engineering an ML-enabled system beyond accuracy
- Explain the role of specifications and their lack in machine learning and the relationship to deductive and inductive reasoning
- Summarize the respective goals and challenges of software engineers vs data scientists
- Explain the concept and relevance of "T-shaped people"

# Agenda Today

1. Preliminaries (just done)
2. Case Study
3. Syllabus
4. Introductions

# Case Study: A Transcription Service Startup

 GoTranscript education discount

 Place Your Order

 Login |  Sign Up

 Contact us



 Services

Cost Estimate

 Samples

Pricing

About Us

Transcriptions samples

Captions and Subtitles samples

# Academic Transcription Services

Our education transcription services have got you covered:

 Lectures

 Seminars

 Group discussions

 Interviews

 Presentations

20% discount for:



Chat with us

# Transcription services

Take audio or video files and produce text.

- Used by academics to analyze interview text
- Podcast show notes
- Subtitles for videos

State of the art a few years ago: Manual transcription, often mechanical turk (1.5 \$/min)

Recently: Many ML models for transcription (e.g., in Youtube, Alexa, Siri, Zoom)

# The startup idea

PhD research on domain-specific speech recognition, that can detect technical jargon

DNN trained on public PBS interviews + transfer learning on smaller manually annotated domain-specific corpus

Research has shown amazing accuracy for talks in medicine, poverty and inequality research, and talks at Ruby programming conferences; published at top conferences

Idea: Let's commercialize the software and sell to academics and conference organizers

# Breakout: Likely challenges in building commercial product?

As a group, think about challenges that the team will likely focus when turning their research into *a product*:

- One machine-learning challenge
- One engineering challenge in building the product
- One challenge from operating and updating the product
- One team or management challenge
- One business challenge
- One safety or ethics challenge

*Post answer to #lecture on Slack and tag all group members (skip if nobody in group has slack set up yet)*

# What qualities are important for a good commercial transcription product?



# ML in a Production System



# ML in a Production System



Last saved a few seconds ago

...

[Share](#)**NOTES**

Write your notes here

**Speaker 5** ► 07:44

Yeah. So there's a slight story behind that. So back when I was in, uh, Undergrad, I wrote a program for myself to measure a, the amount of time I did data entry from my father's business and I was on windows at the time and there wasn't a function called time dot [inaudible] time, uh, which I needed to parse dates to get back to time, top of representation, uh, I figured out a way to do it and I gave it to what's called the python cookbook because it just seemed like something other people could use. So it was just trying to be helpful. Uh, subsequently I had to figure out how to make it work because I didn't really have to. Basically, it bothered me that you had to input all the locale information and I figured out how to do it over the subsequent months. And actually as a graduation gift from my Undergrad, the week following, I solved it and wrote it all out.

**Speaker 5** ► 08:38

And I asked, uh, Alex Martelli, the editor of the Python Cookbook, which had published my original recipe, a, how do I get this into python? I think it might help

How did we do on your transcript?

## Speaker notes

Highlights challenging fragments. Can see what users fix inplace to correct. Star rating for feedback.





and Data engineers + Domain specialists + Operators + Business team + Project managers + Designers, UI Experts + Safety, security specialists + Lawyers + Social scientists + ...

# Data scientist

- Often fixed dataset for training and evaluation (e.g., PBS interviews)
- Focused on accuracy
- Prototyping, often Jupyter notebooks or similar
- Expert in modeling techniques and feature engineering
- Model size, updateability, implementation stability typically does not matter

# Software engineer

- Builds a product
- Concerned about cost, performance, stability, release time
- Identify quality through customer satisfaction
- Must scale solution, handle large amounts of data
- Detect and handle mistakes, preferably automatically
- Maintain, evolve, and extend the product over long periods
- Consider requirements for security, safety, fairness

# Likely collaboration challenges?



# What might Software Engineers and Data Scientists Focus on?



By Steven Geringer, via Ryan Orban. [Bridging the Gap Between Data Science & Engineer: Building High-Performance Teams](#). 2016

# T-Shaped People

*Broad-range generalist + Deep expertise*



"I-shaped"  
Expert at one thing



Generalist  
Capable in a lot of things  
but not expert in any



"T-shaped"  
Capable in a lot of things  
and expert in one of them

≡ Figure: Jason Yip. Why T-shaped people?. 2018

# T-Shaped People

*Broad-range generalist + Deep expertise*

Example:

- Basic skills of software engineering, business, distributed computing, and communication
- Deep skills in deep neural networks (technique) and medical systems (domain)

# Latest Buzzword: $\pi$ -Shaped People

$\pi$

# Examples for discussion

- What does correctness or accuracy really mean? What accuracy do customers care about?
- How can we see how well we are doing in practice? How much feedback are customers going to give us before they leave?
- Can we estimate how good our transcriptions are? How are we doing for different customers or different topics?
- How to present results to the customers (including confidence)?
- When customers complain about poor transcriptions, how to prioritize and what to do?
  
- What are unacceptable mistakes and how can they be avoided? Is there a safety risk?
- Can we cope with an influx of customers?
- Will transcribing the same audio twice produce the same result? Does it matter?
- How can we debug and fix problems? How quickly?

# Examples for discussion 2

- With more customers, transcriptions are taking longer and longer -- what can we do?
- Transcriptions sometimes crash. What to do?
- How do we achieve high availability?
- How can we see that everything is going fine and page somebody if it is not?
- We improve our entity detection model but somehow system behavior degrades... Why?
- Tensorflow update; does our infrastructure still work?
- Once somewhat successful, how to handle large amounts of data per day?
- Buy more machines or move to the cloud?
  
- Models are continuously improved. When to deploy? Can we roll back?
- Can we offer live transcription as an app? As a web service?
- Can we get better the longer a person talks? Should we then go back and reanalyze the beginning? Will this benefit the next upload as well?

# Examples for discussion 3

- How many domains can be supported? Do we have the server capacity?
- How specific should domains be? Medical vs "International Conference on Allergy & Immunology"?
- How to make it easy to support new domains?
- Can we handle accents?
- Better recognition of male than female speakers?
- Can and should we learn from customer data?
- How can we debug problems on audio files we are not allowed to see?
- Any chance we might private leak customer data?
- Can competitors or bad actors attack our system?

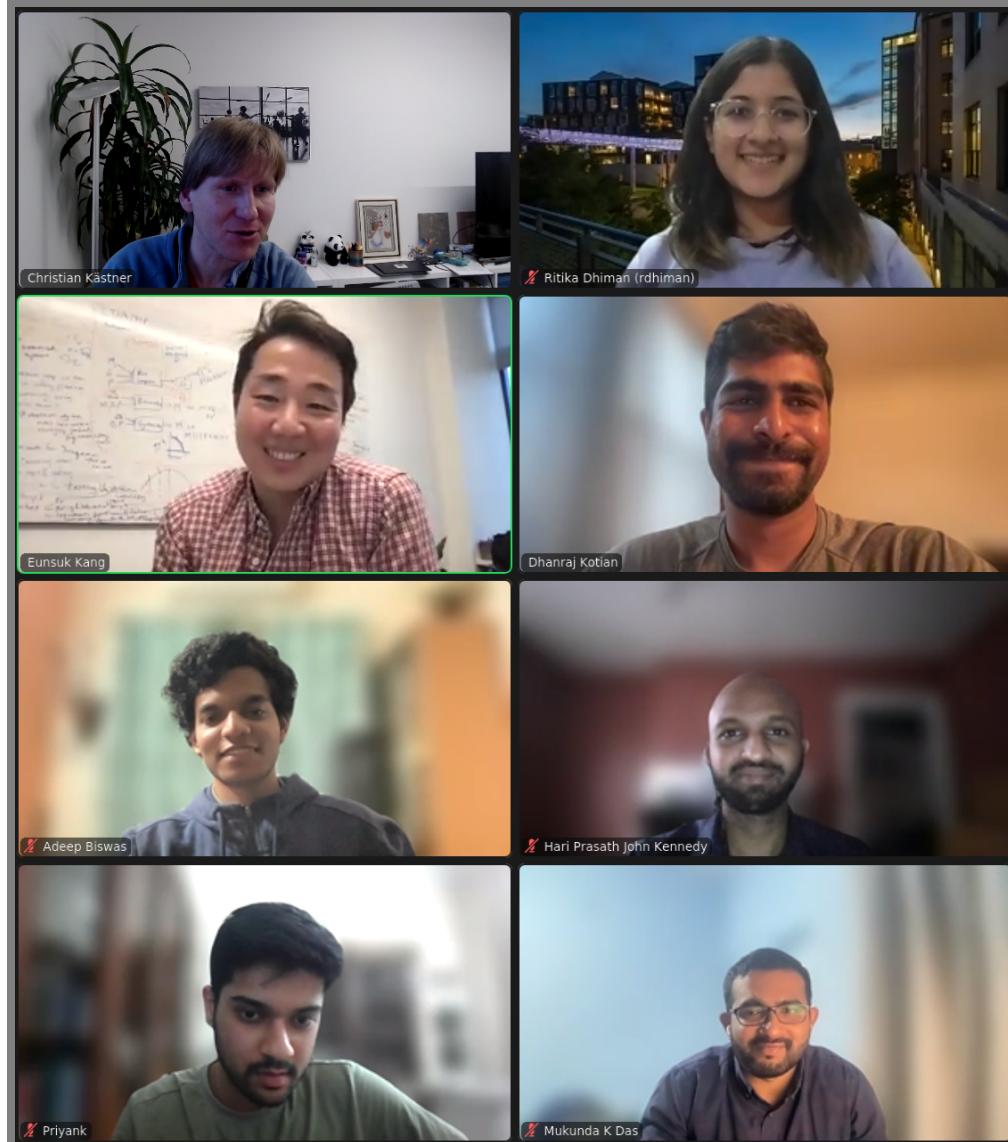
# Syllabus and Class Structure

17-445/17-645/17-745/11-695, Spring 2023, 12 units

Monday/Wednesdays 2-3:25pm

Recitation Fridays 10:00-10:50am / 12:00-12:50pm

# Instructors



# Communication

- Email us or ping us on Slack (invite link on Canvas)
- All announcements through Slack #announcements
- Weekly office hours (see Canvas for schedule)
- Post questions on Slack
  - Please use #general or #assignments and post publicly if possible; your classmates will benefit from your Q&A!
- All course materials (slides, assignments, old midterms) available on GitHub and course website: <https://mlip-cmu.github.io/s2023/>
  - Pull requests encouraged!

# Class with software engineering flavor

Focused on engineering judgment

Arguments, tradeoffs, and justification,  
rather than single correct answer

Practical engagement, building  
systems, testing, automation

Strong teamwork component

Both text-based and code-based  
homework assignments



# Prerequisites

## Some machine-learning experience required

- Basic understanding of data science process, incl. data cleaning, feature engineering, using ML libraries
- High level understand of machine-learning approaches
  - supervised learning
  - regression, decision trees, neural networks
  - accuracy, recall, precision, ROC curve
- Ideally, some experience with notebooks, sklearn or other frameworks

## Basic programming and command-line skills will be needed

## No further software-engineering knowledge required

- Teamwork experience in product team is useful but not required
- No required exposure to requirements, software testing, software design, continuous integration, containers, process management, etc
  - If you are familiar with these, there will be some redundancy -- sorry!

# First Homework Assignment I1

*"Coding warmup assignment"*

**Out now**, due Monday Jan 30

Enhance simple web *application*  
with ML-based features: Image  
search and automated captioning

Open ended coding assignment,  
change existing code, learn new  
APIs



# Active lecture

Case study driven

Discussions highly encouraged

Regular in-class activities,  
breakouts

Contribute your own experience!

Discussions over definitions

the-changelog-318  
Dashboard | Quality: High ⓘ Last saved a few seconds ago ⋮ Share

00:00 Offset 00:00 01:31:27  
Play Back 5s 1x Volume

NOTES  
Write your notes here

**Speaker 5** ▶ 07:44  
Yeah. So there's a slight story behind that. So back when I was in, uh, Undergrad, I wrote a program for myself to measure a, the amount of time I did data entry from my father's business and I was on windows at the time and there wasn't a function called time dot [inaudible] time, uh, which I needed to parse dates to get back to time, top of representation, uh, I figured out a way to do it and I gave it to what's called the python cookbook because it just seemed like something other people could use. So it was just trying to be helpful. Uh, subsequently I had to figure out how to make it work because I didn't really have to. Basically, it bothered me that you had to input all the locale information and I figured out how to do it over the subsequent months. And actually as a graduation gift from my Undergrad, the week following, I solved it and wrote it all out.

**Speaker 5** ▶ 08:38  
And I asked, uh, Alex Martelli, the editor of the Python Cookbook, which had published my original recipe, a, how do I get this into python? I think it might help

How did we do on your transcript? ⭐⭐⭐⭐⭐

# Recordings and Attendance

Try to attend lecture -- discussions are important to learning

Participation is part of your grade

No lecture recordings, textbook and slides available

Contact us for accommodations (illness, interview travel, unforseen events) or have your advisor reach out. We try to be flexible

# Participation

Participation != Attendance

Grading:

- 100%: Participates actively at least once in most lectures by (1) asking or responding to questions or (2) contributing to breakout discussions
- 90%: Participates actively at least once in two thirds of the lectures
- 75%: Participates actively at least once in over half of the lectures
- 50%: Participates actively at least once in one quarter of the lectures
- 20%: Participates actively at least once in at least 3 lectures.

# Fundamentals of Engineering AI-Enabled Systems

**Holistic system view:** AI and non-AI components, pipelines, stakeholders, environment interactions, feedback loops

## Requirements:

- System and model goals
- User requirements
- Environment assumptions
- Quality beyond accuracy
- Measurement
- Risk analysis
- Planning for mistakes

## Architecture + design:

- Modeling tradeoffs
- Deployment architecture
- Data science pipelines
- Telemetry, monitoring
- Anticipating evolution
- Big data processing
- Human-AI design

## Quality assurance:

- Model testing
- Data quality
- QA automation
- Testing in production
- Infrastructure quality
- Debugging

## Operations:

- Continuous deployment
- Contin. experimentation
- Configuration mgmt.
- Monitoring
- Versioning
- Big data
- DevOps, MLOps

**Teams and process:** Data science vs software eng. workflows, interdisciplinary teams, collaboration points, technical debt

# Responsible AI Engineering

Provenance,  
versioning,  
reproducibility

Safety

Security and  
privacy

Fairness

Interpretability  
and explainability

Transparency  
and trust

Ethics, governance, regulation, compliance, organizational culture

# Reading Assignments & Quizzes

*Building Intelligent Systems* by Geoff Hulten

<https://www.buildingintelligentsystems.com/>

Most chapters assigned at some point in the semester

Supplemented with research articles, blog posts, videos, podcasts, ...

[Electronic version in the library](#)



# Reading Quizzes

Short essay questions on readings, due before start of lecture (Canvas quiz)

Planned for: about 30-45 min for reading, 15 min for discussing and answering quiz

# Book for the Class

*"Machine Learning in Production: From Models to Products"*

Mostly similar coverage to lecture

Not required, use as supplementary reading

Still evolving, feedback appreciated!

Published [online](#)

# Assignments

All [assignments](#) available on GitHub now

Series of 4 small to medium-sized **individual assignments**:

- Engage with practical challenges
- Analyze risks, fairness
- Reason about tradeoffs and justify your decisions
- Mostly written reports, a little modeling, some coding

**Large team project** with 4 milestones:

- Build and deploy a prediction (movie recommendation) service
- Testing in production, monitoring
- Final presentation

≡ Usually due Monday night; see schedule

# 17-745 PhD Research Project

Research project instead of individual assignments I3 and I4

Design your own research project and write a report

- A case study, empirical study, literature survey, etc.,

Very open ended: Align with own research interests and existing projects

See the [project description](#) and talk to us

First hard milestone: initial description due Feb 27

# Timeline

11-695/17-445/17-645



17-745



# Recitations

Introducing various tools, e.g., fastAPI (serving), Kafka (stream processing), Jenkins (continuous integration), MLflow (experiment tracking), Docker & Kubernetes (containers), Prometheus & Grafana (monitoring), CHAP (explainability)...

Hands on exercises, bring a laptop

Often introducing tools relevant for assignments

First recitation on **this Friday**: Calling, securing, and creating APIs

# Grading

- 40% individual assignment
- 30% group project with final presentation
- 10% midterm
- 10% participation
- 10% reading quizzes
- No final exam (final presentations will take place in that timeslot)

Expected grade cutoffs in syllabus (>82% B, >94 A-, >96% A, >99% A+)

# Grading Philosophy

Specification grading, based in adult learning theory

Giving you choices in what to work on or how to prioritize your work

We are making every effort to be clear about expectations (specifications), will clarify if you have questions

Assignments broken down into expectations with point values, each graded pass/fail

Opportunities to resubmit work until last day of class

≡ [Example]

# Token System for Flexibility

7 individual tokens per student:

- Submit individual assignment 1 day late for 1 token (after running out of tokens 15% penalty per late day)
- Redo individual assignment for 3 token
- Resubmit or submit reading quiz late for 1 token
- Remaining tokens count toward participation
- 1 bonus token for attending >66% of recitations

7 team tokens per team:

- Submit milestone 1 day late for 1 token (no late submissions accepted when out of tokens)
- Redo milestone for 3 token

# How to use tokens

- No need to tell us if you plan to submit very late. We will assign 0 and you can resubmit
- Instructions and form for resubmission on Canvas
- We will automatically use remaining tokens toward participation and quizzes at the end
- Remaining individual tokens reflected on Canvas, for remaining team tokens ask your TA.

# Group project

Instructor-assigned teams

Teams stay together for project throughout semester, starting Feb 6

Fill out Catme Team survey before Feb 6 (3pt)

Some advice in lectures; we'll help with debugging team issues

TA assigned to each team as mentor; mandatory debriefing with mentor and peer grading on all milestones (based on citizenship on team)

- ☰ Bonus points for social interaction in project teams

# Academic honesty

See web page

In a nutshell: do not copy from other students, do not lie, do not share or publicly release your solutions

In group work, be honest about contributions of team members, do not cover for others

If you feel overwhelmed or stressed, please come and talk to us (see syllabus for other support opportunities)

# Thoughts on Generative AI for Homework?



GPT3, ChatGPT, ...? Reading quizzes, homework submissions, ...?

# Our Position on Generative AI for Homework.

This is a course on responsible building of ML products. This includes questions of how to build generative AI tools responsibly and discussing what use is ethical.

Feel free to use them and explore whether they are useful. Welcome to share insights/feedback.

Warning: They are *bullshit generators*! Requires understanding to check answers. We test them ourselves and they often generate bad/wrong answers for reading quizzes.

You are still responsible for the correctness of what you submit!



# What makes software with ML challenging?

# ML Models Make Mistakes



NeuralTalk2: A flock of birds flying in the air

Microsoft Azure: A group of giraffe standing next to a tree

*Image: Fred Dunn, <https://www.flickr.com/photos/gratapictures> - CC-BY-NC*

## Speaker notes

Source: <https://www.aiweirdness.com/do-neural-nets-dream-of-electric-18-03-02/>



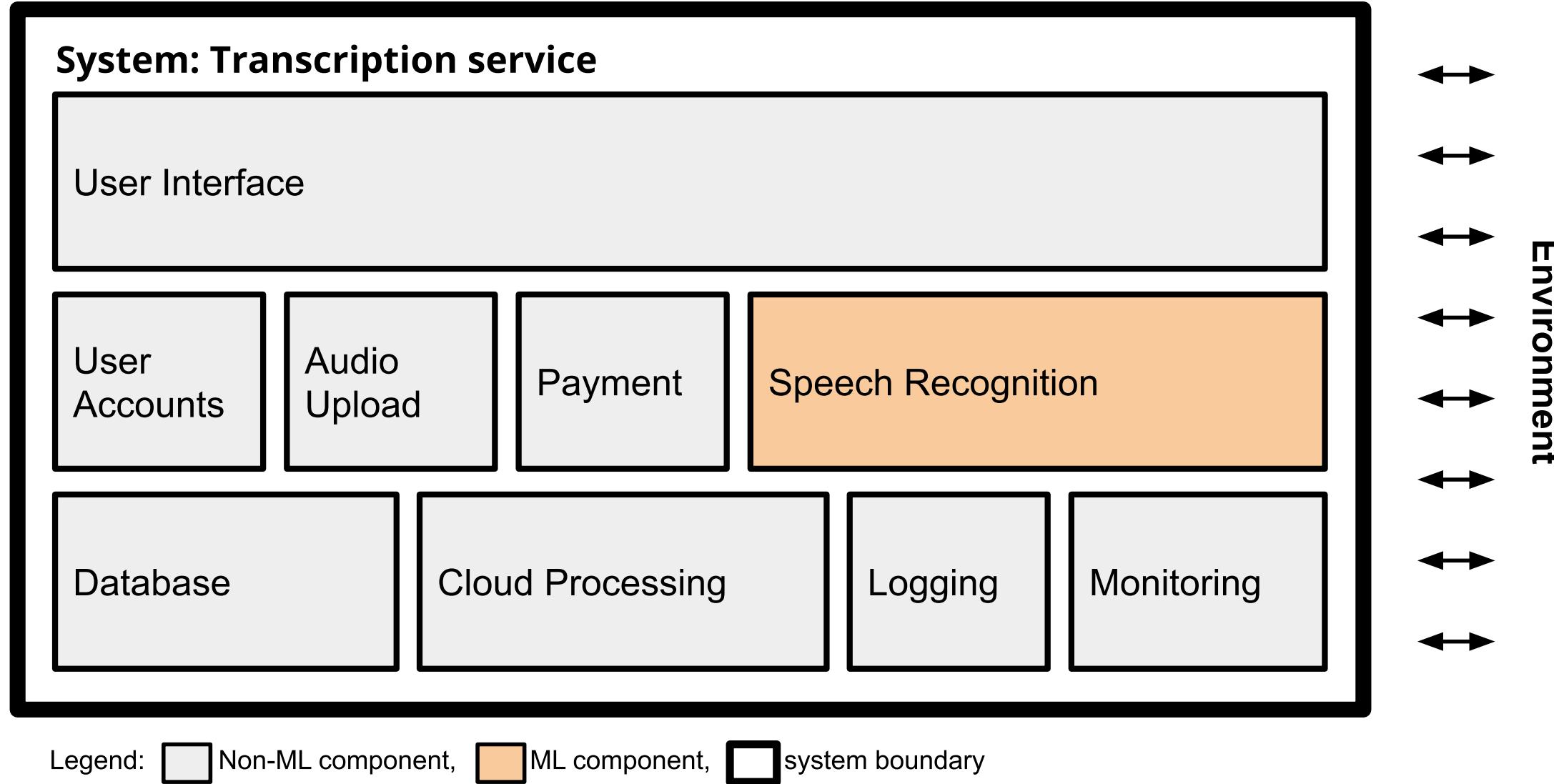
# Lack of Specifications

```
/**  
 * Return the text spoken within the audio file  
 * ???  
 */  
String transcribe(File audioFile);
```

# Data Focused and Scalable



# Interaction with the environment



# It's not all new

We routinely build:

- Safe software with unreliable components
- Cyberphysical systems
- Non-ML big data systems, cloud systems
- "Good enough" and "fit for purpose" not "correct"

ML intensifies our challenges

# Complexity



*Restaurant website  
Video streaming service  
Podcast hosting  
Conference website*

*Electric scooters  
Medical records  
Payment software*

*Nuclear power plant  
Space exploration*

# Complexity



# Introductions

Before the next lecture, introduce yourself in Slack channel #social:

- Your (preferred) name
- In 1~2 sentences, your data science background and goals (e.g., coursework, internships, work experience)
- In 1~2 sentences, your software engineering background, if any, and goals (e.g., coursework, internships, work experience)
- One topic you are particularly interested in learning during this course?
- A hobby or a favorite activity outside school

# Summary

Machine learning components are part of larger systems

*Data scientists and software engineers have different goals and focuses*

- Building systems requires both
- Various qualities are relevant, beyond just accuracy

Machine learning brings new challenges and intensifies old ones

