Engineering ML Systems

17-313 Spring 2025

Foundations of Software Engineering

https://cmu-313.github.io

Michael Hilton, Austin Henley, and Nadia Nahar

Administrivia

- P3B (Final Deliverables) due on Thursday

Smoking Section

- Last full row

Learning Goals

- Identify the stages/tasks that comprise the typical ML development pipeline.
- Identify differences between traditional software development and development of ML systems.
- Understand the complexities of integrating ML into a software engineering process/system
- Identify challenges in handling unreliable ML components, and strategies to mitigate impact of mistakes
- Identify the architectural decisions to be taken and tradeoffs

What is one thing you remember from last class?

SE and ML: Connected in Two Ways

Using ML for engineering

How to use AI to help engineering processes?

Artificial intelligence for software engineering: AI4SE

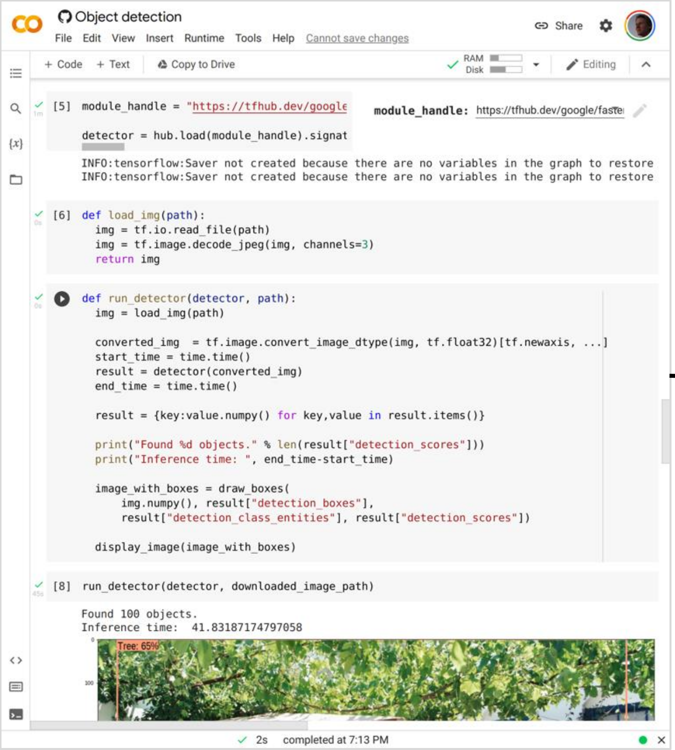
Engineering ML systems

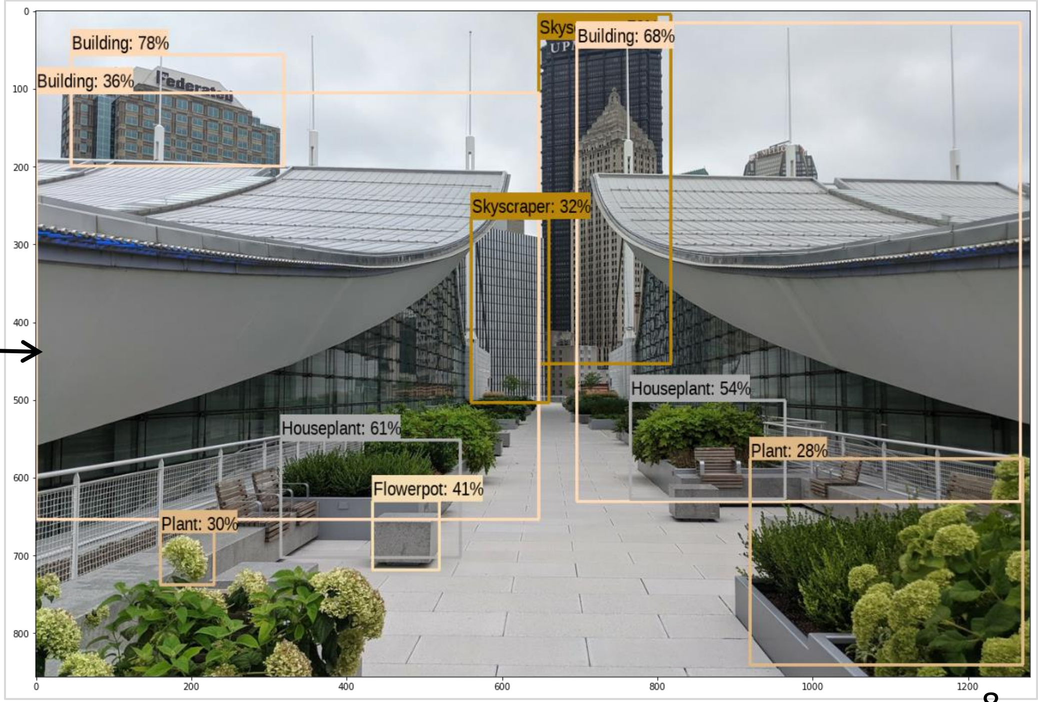
How to integrate AI components into engineering systems?

Software engineering for Artificial Intelligence: SE4AI

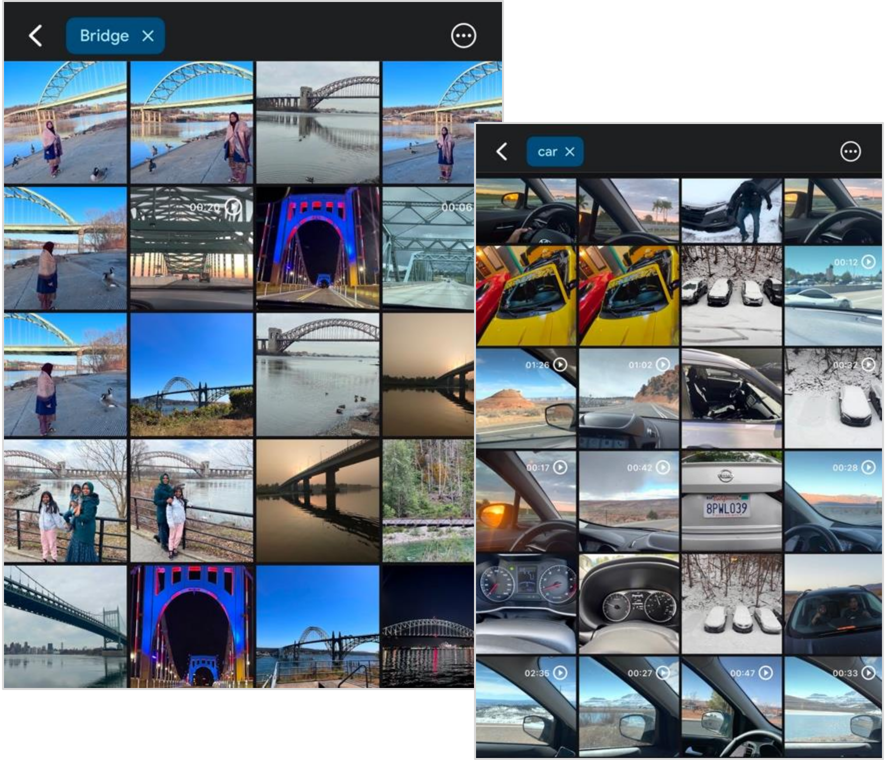
From Models to Systems

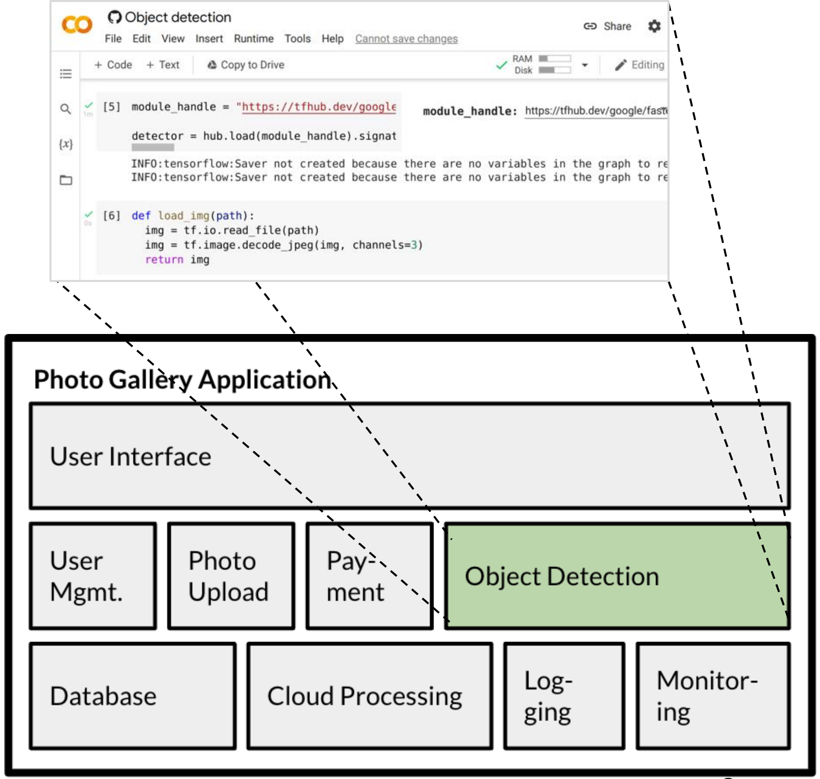
ML Model vs. ML System





ML Model vs. **ML System**





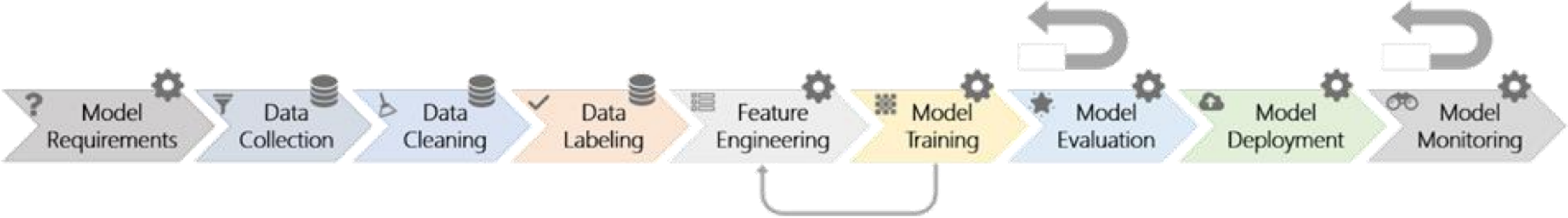
Apollo ML Models

Source: Zi Peng, Jinqiu Yang, Tse-Hsun (Peter) Chen, and Lei Ma. 2020. A First Look at the Integration of Machine Learning Models in Complex Autonomous Driving Systems: A Case Study on Apollo. In Proceedings of the 28th ACM Joint European Software Engineering Conference and Symposium on the Foundations of Software Engineering (ESEC/FSE '20)

Augmented Reality Smart Glasses

What apps do you use that have ML?

Machine Learning Pipeline

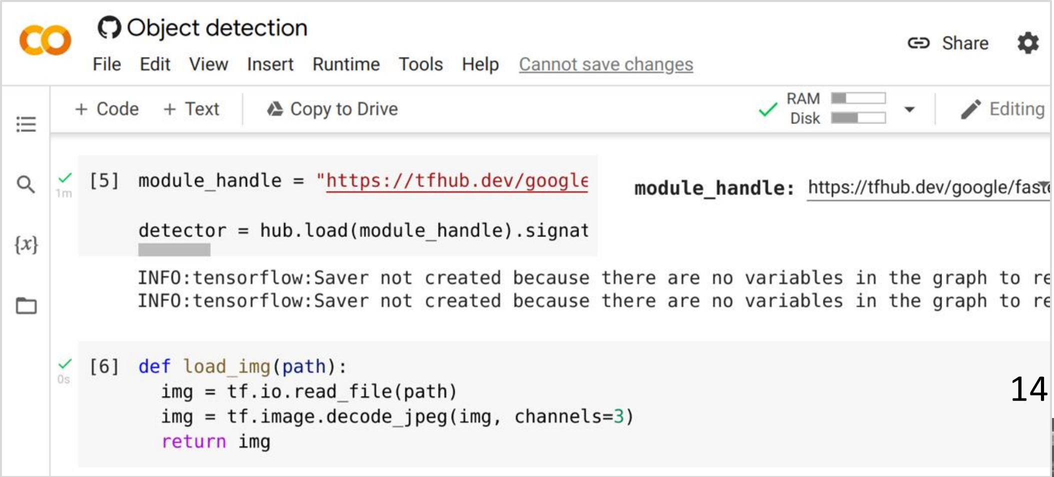


Source: “Software Engineering for Machine Learning: A Case Study” by Amershi et al. ICSE 2019

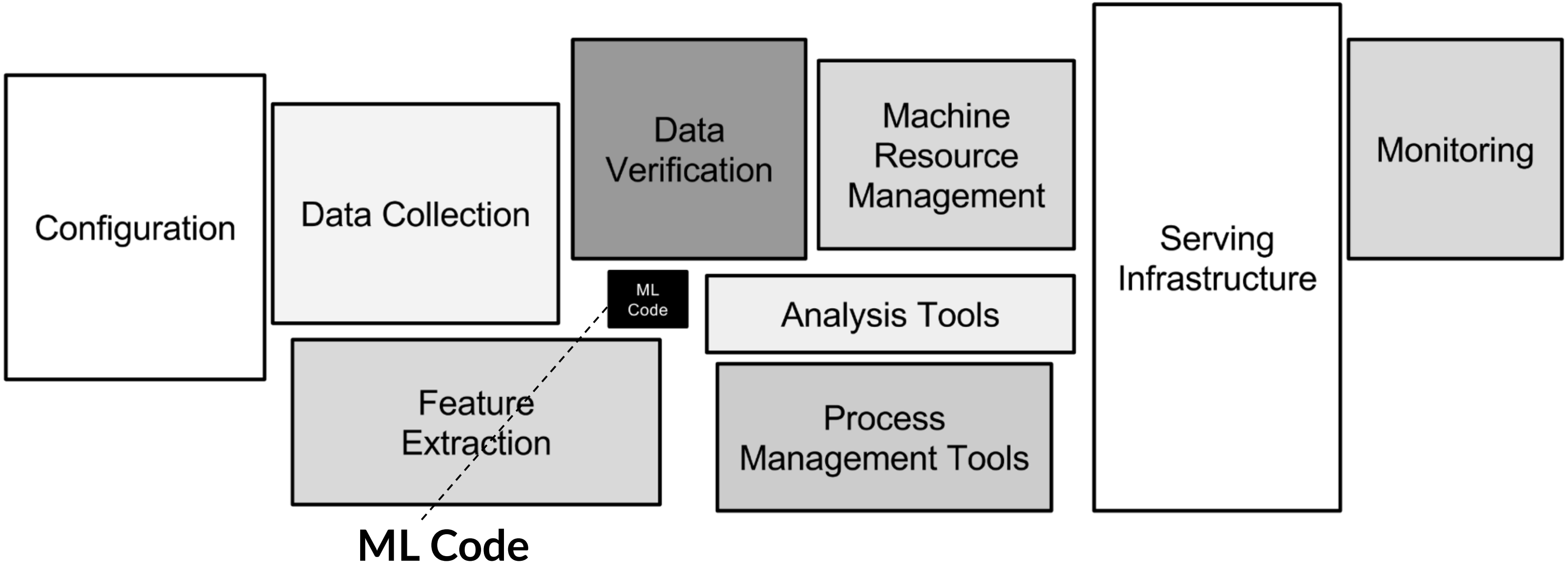
Let’s Take a Closer Look

Typical Machine Learning Book / Course

Focus: building models from given data, evaluating accuracy

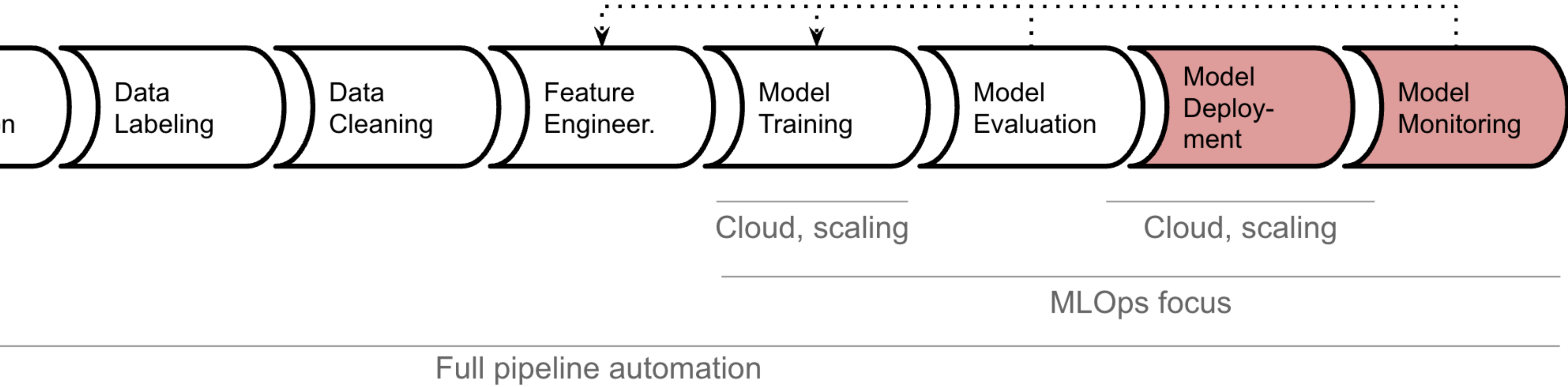


Only a fraction of real-world ML systems is ML code...



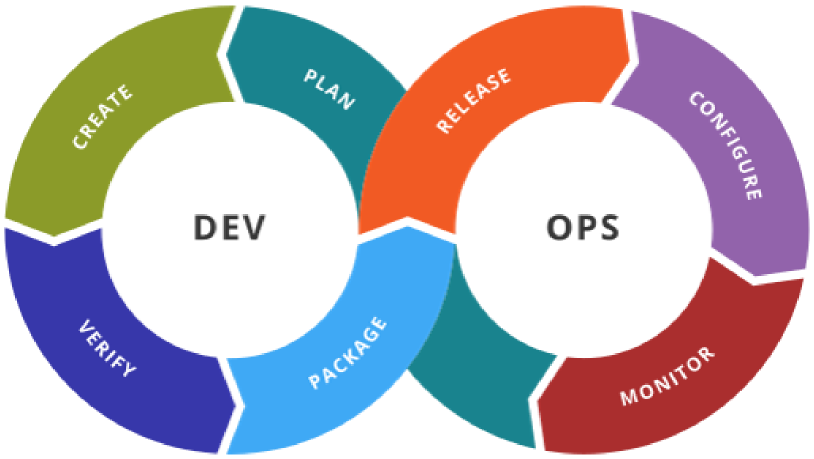
Sculley, et al. "*Hidden technical debt in machine learning systems.*" NeurIPS 28 (2015): 2503-2511.

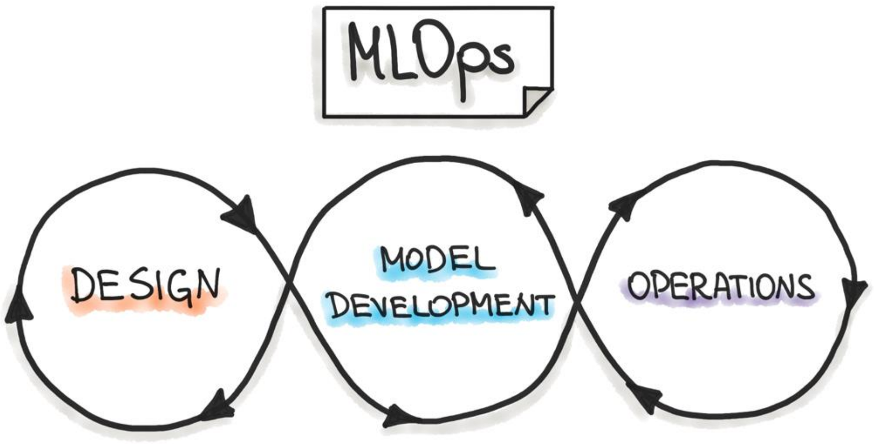
Pipeline Automation and MLOps



Focus: experimenting, deploying, scaling training and serving, model monitoring and updating

DevOps and MLOps



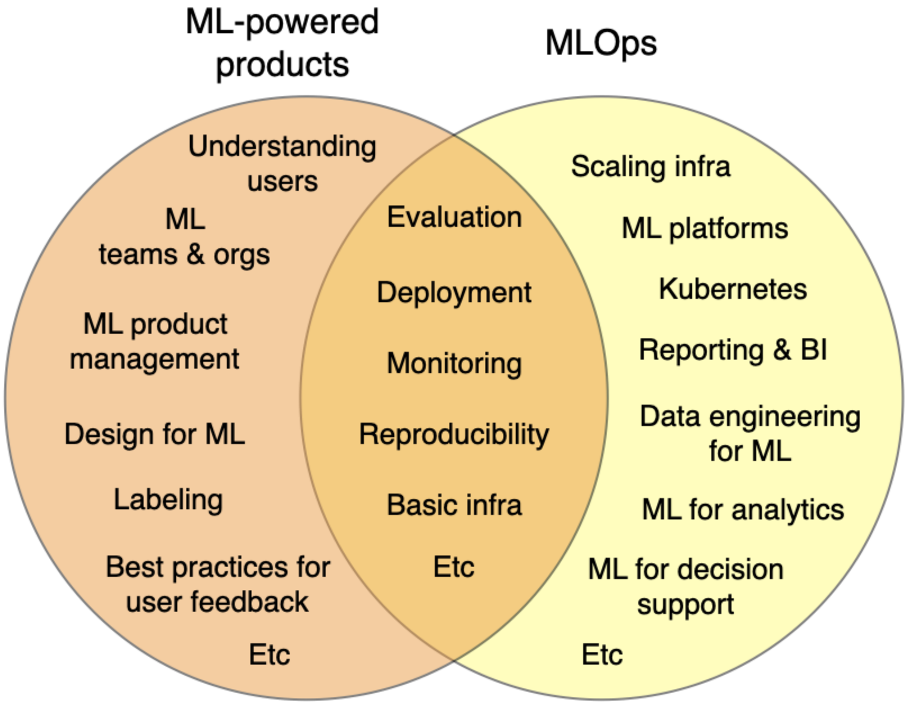


Set of practices for continuous delivery; relies on heavy automation, e.g., continuous delivery, monitoring

Automation around Machine Learning pipeline, including training, evaluation, versioning, and deployment

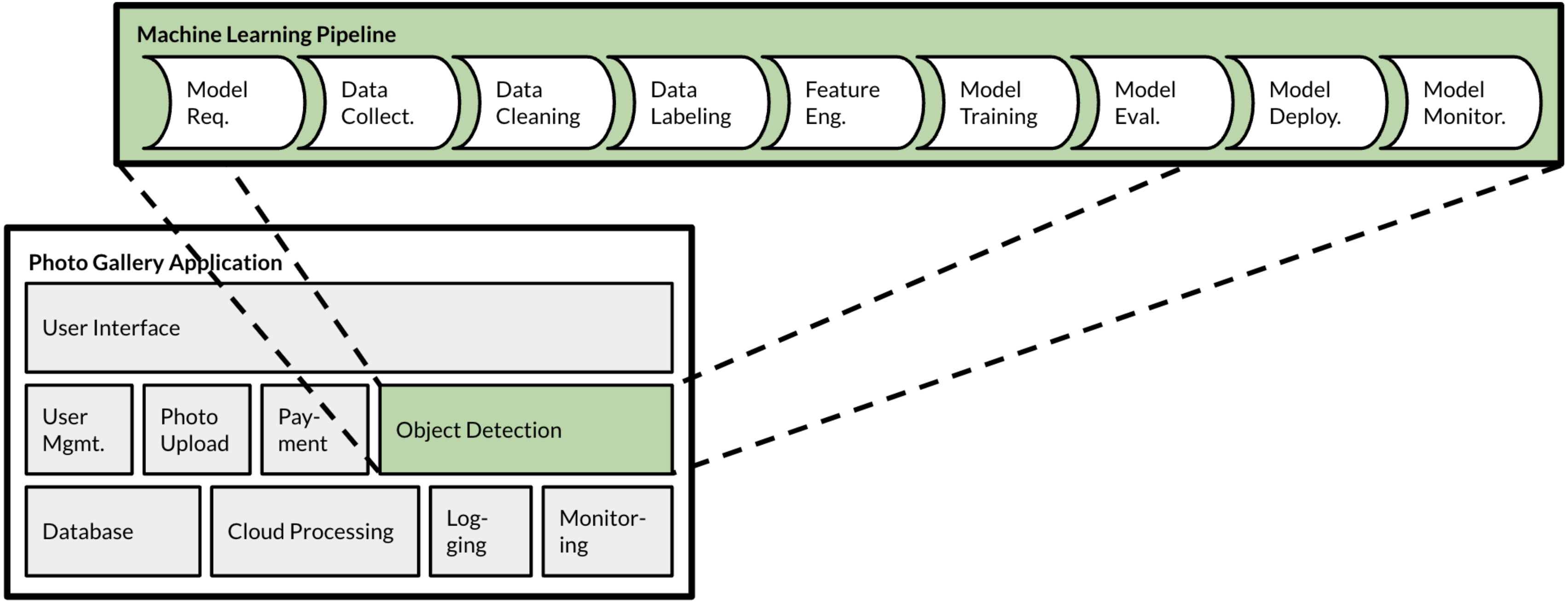
Think about MLOps as a specialized subset of DevOps for machine learning applications

There is more to ML systems than MLOps…

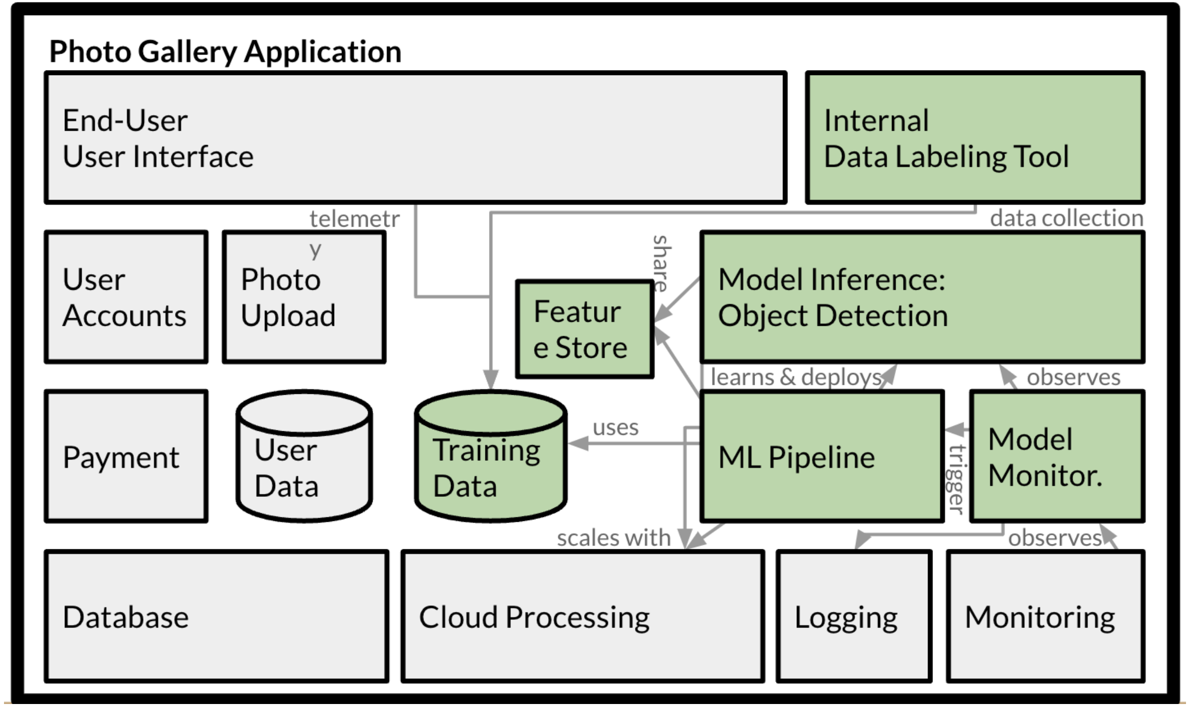


https://fullstackdeeplearning.com/course/2022/lecture-1-course-vision-and-when-to-use-ml/

ML is a Component in a System

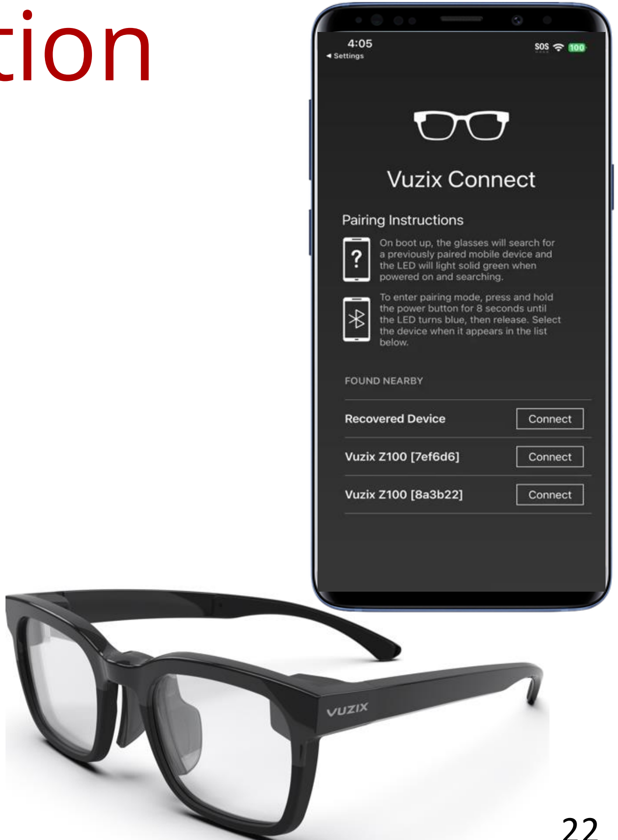


Or Many ML Components Actually



What are some ML vs non-ML components in the apps, you mentioned?

Case Study: Augmented Reality Smart Glasses for Navigation

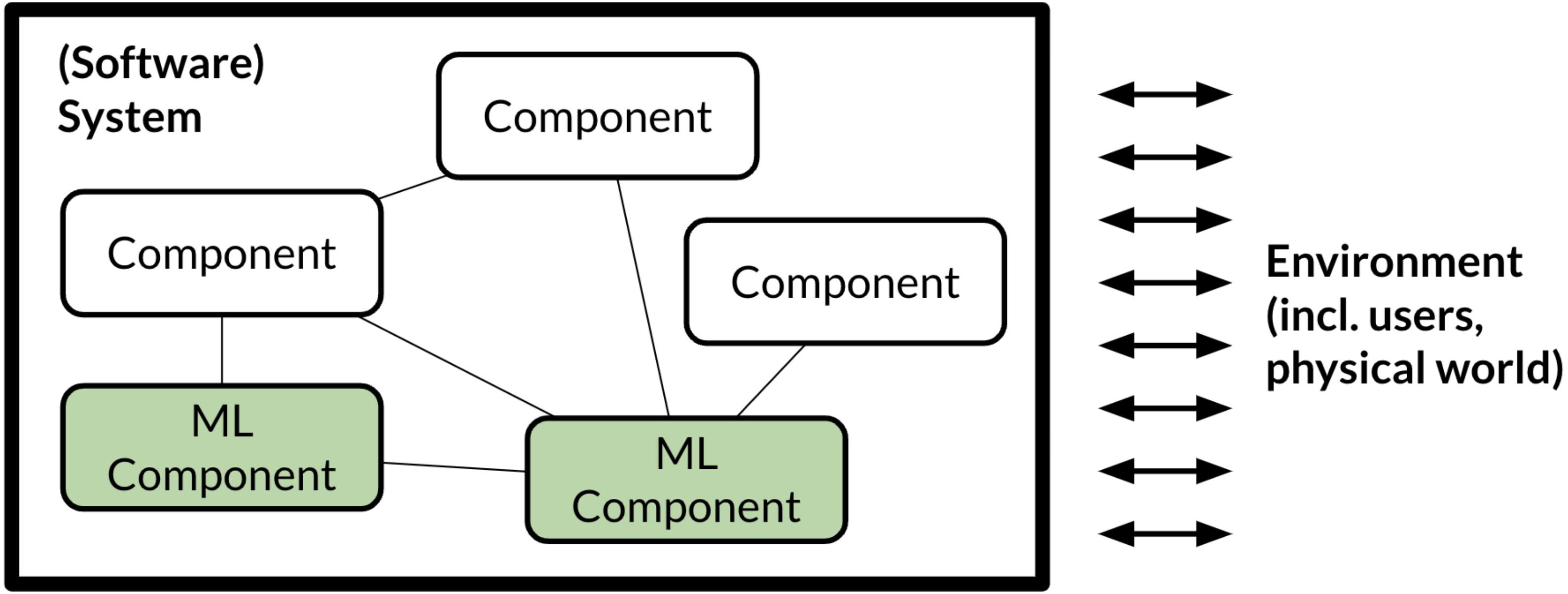


Activity: Draw Architectural Diagram with ML and non-ML Components

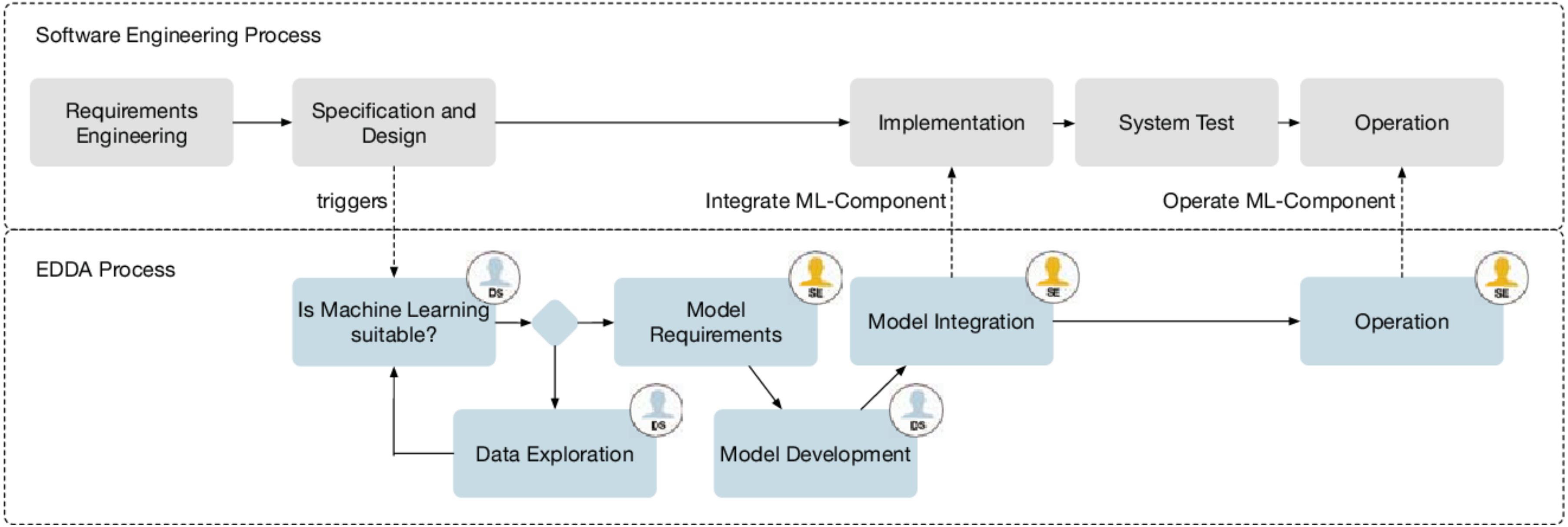
In a team of 2-3 students, consider the augmented reality navigation system to:

- identify the ML components
- identity the non-ML components
- draw an architectural diagram with the components with notations of your choice

Systems Thinking

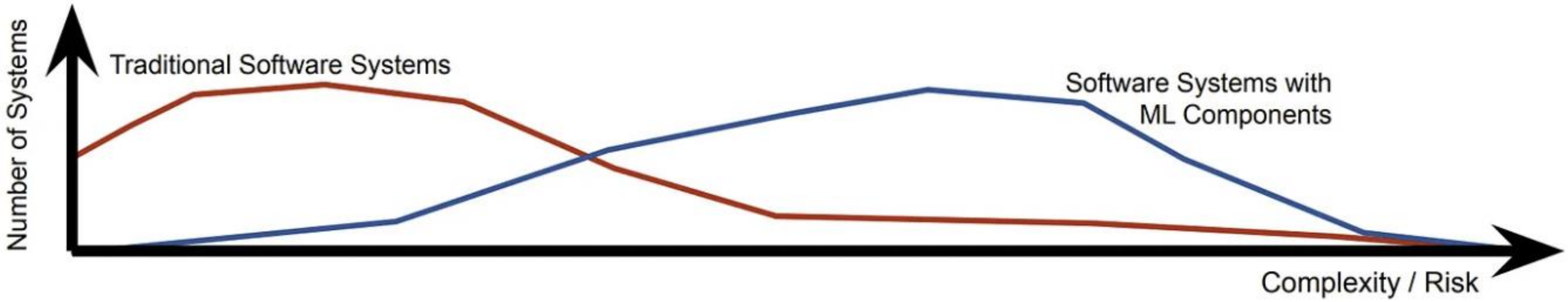


ML Introduces Additional Complexities in Software Systems



Hesenius, Marc, et al. "Towards a software engineering process for developing data-driven applications." 2019 IEEE/ACM 7th International Workshop on Realizing Artificial Intelligence Synergies in Software Engineering (RAISE). IEEE, 2019.

ML Introduces Additional Complexities in Software Systems



Speculation based on our observations: Most systems with machine-learning components tend to fall toward the more complex or more risky end of the spectrum of possible software systems, compared to traditional systems without machine learning.

Christian Kästner. Machine Learning in Production: From Models to Products. 2022. https://ckaestne.medium.com/introduction-to-machine-learning-in-production-eef7427426f1

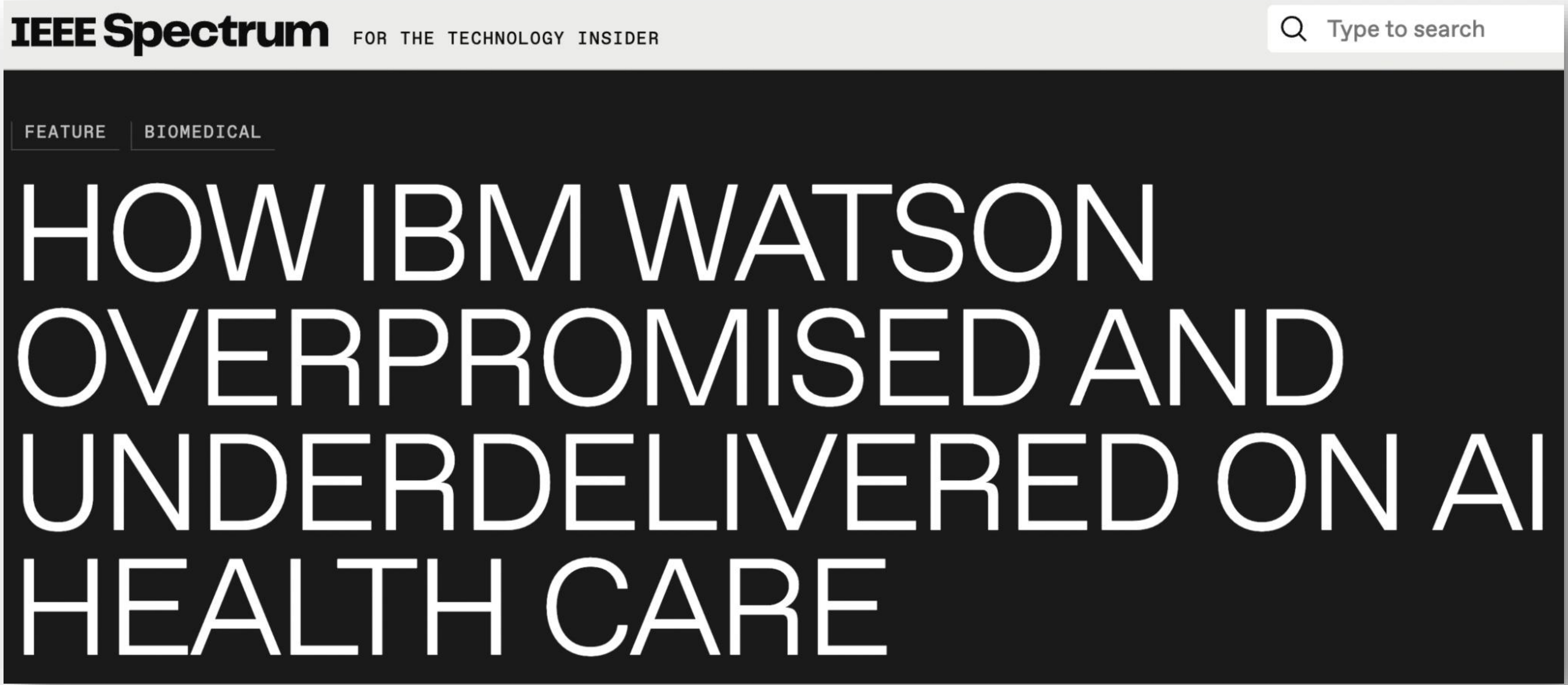
Why 85% of Machine Learning Projects Fail – How to Avoid This

According to Gartner, 85% of Machine Learning (ML) projects fail. Worse yet, the research company predicts that this trend will continue through 2022.

Does this point to some weakness in ML itself? No, it points to weaknesses in the way it's applied to projects.

The high failure rate of machine learning projects, often cited around 85%, can be attributed to factors like inadequate data quality, lack of skilled personnel, unrealistic expectations, and challenges in integrating machine learning into existing workflows.

https://www.iiot-world.com/industrial-iot/connected-industry/why-85-of-machine-learning-projects-fail



https://spectrum.ieee.org/how-ibm-watson-overpromised-and-underdelivered-on-ai-health-care

The New York Times

Apple Kills Its Electric Car Project

The car, which Apple spent billions of dollars researching, had been intended as a rival to Tesla's E.V.s, which include autonomous driving features.

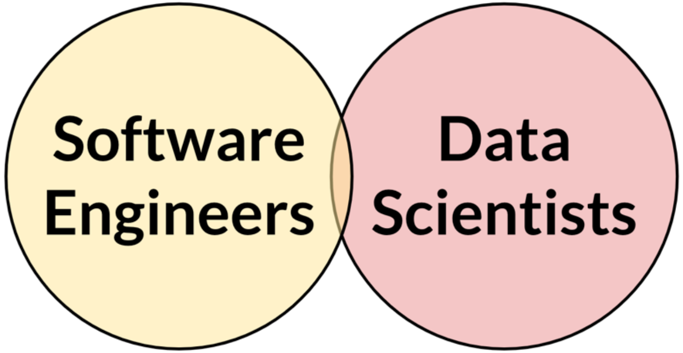
https://www.nytimes.com/2024/02/27/technology/apple-ends-electric-car-plan.html
https://www.nytimes.com/2024/02/28/technology/behind-the-apple-car-dead.html

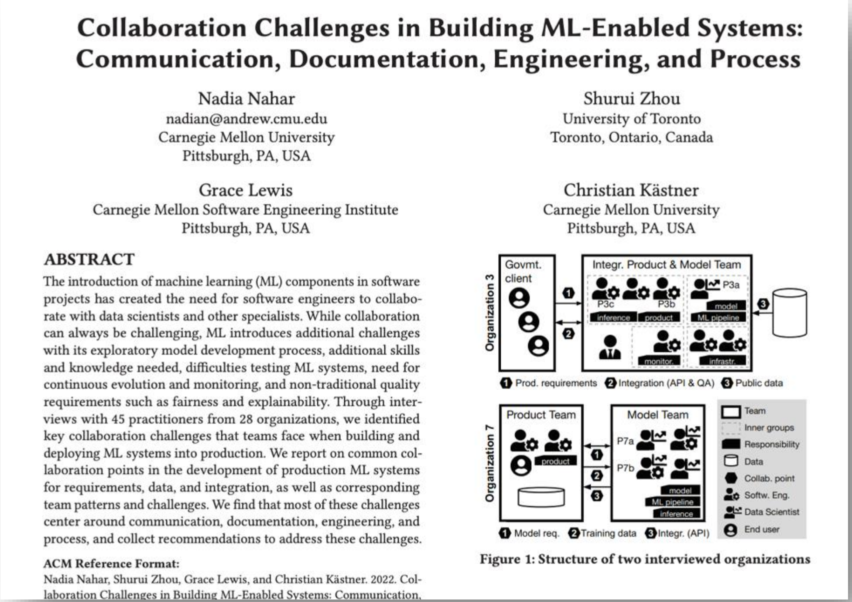
What Changes with ML

Contrast with SE

- **Experimental:** Experiment-driven with model training, testing, and refinement based on empirical data.
- **Data-Driven:** Relies heavily on data to train models; data preprocessing is crucial.
- **Algorithmic Focus:** Development of algorithms (e.g., supervised, unsupervised learning) for pattern recognition.
- **Model Evaluation:** Continuous refinement through metrics like accuracy, precision, and recall.

Change of process/ metrics/ mindsets needed…





Collaboration Challenges in Building ML-Enabled Systems: Communication, Documentation, Engineering, and Process

Nadia Nahar
nadian@andrew.cmu.edu
Carnegie Mellon University
Pittsburgh, PA, USA

Shurui Zhou
University of Toronto
Toronto, Ontario, Canada

Grace Lewis
Carnegie Mellon Software Engineering Institute
Pittsburgh, PA, USA

Christian Kästner
Carnegie Mellon University
Pittsburgh, PA, USA

ABSTRACT

The introduction of machine learning (ML) components in software projects has created the need for software engineers to collaborate with data scientists and other specialists. While collaboration can always be challenging, ML introduces additional challenges with its exploratory model development process, additional skills and knowledge needed, difficulties testing ML systems, need for continuous evolution and monitoring, and non-traditional quality requirements such as fairness and explainability. Through interviews with 45 practitioners from 28 organizations, we identified key collaboration challenges that teams face when building and deploying ML systems into production. We report on common collaboration points in the development of production ML systems for requirements, data, and integration, as well as corresponding team patterns and challenges. We find that most of these challenges center around communication, documentation, engineering, and process, and collect recommendations to address these challenges.

ACM Reference Format:
Nadia Nahar, Shurui Zhou, Grace Lewis, and Christian Kästner. 2022. Collaboration Challenges in Building ML-Enabled Systems: Communication,

Figure 1: Structure of two interviewed organizations

Nahar, Nadia, et al. "Collaboration challenges in building ml-enabled systems: Communication, documentation, engineering, and process." *Proceedings of the 44th international conference on software engineering*. 2022.

Specifications and Testing in SE

```
/**
 * Return the sum of all values
 * @ensures \result = \sum int i; 0 <= i < …
 */
int sum(int[] values);
```

```
@Test
void testSentence1() {
  assertEquals(9, sum({2, 3, 4}));
}
```

Lack of Specification in ML

```
/**
 * Detect objects visible in image
 * ????
 */
ObjectId[] detectObjects(File image);
```

Lack of Specification in ML

```
@Test
void testHomePhoto() {
  assertEquals({HOUSE, PLANT},
               detectObjects("img1.jpg"));
}
```

```
@Test
void testStreetPhoto() {
  assertEquals({PERSON, DOG, BICYCLE},
               detectObjects("img2.jpg"));
}
```

Lack of Specifications…

… breaks modular reasoning

… challenges quality assurance

… inhibits safety and fairness reasoning

… hinders coordination across teams

(though, we didn't need ML to build low quality, harmful, and unethical software)

All Models are Wrong!

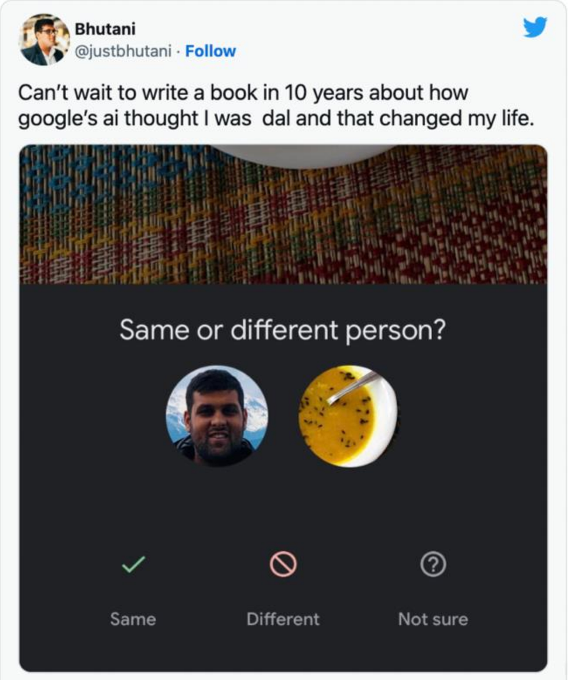
“*All models are approximations. Assumptions, whether implied or clearly stated, are never exactly true.*

All models are wrong, but some models are useful.

So the question you need to ask is not "Is the model true?" (it never is) but "Is the model good enough for this particular application?"

George Box

Model Makes Mistake



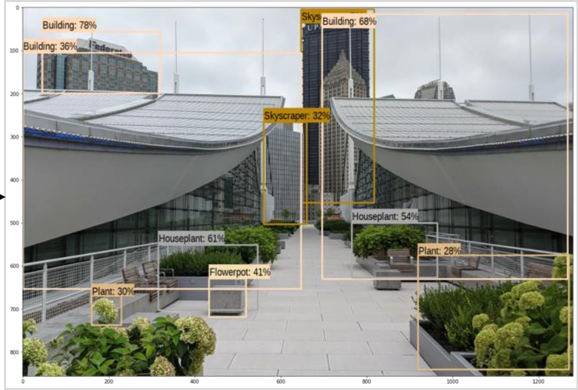
Mistakes Cause Harms

Self-driving Uber car that hit and killed woman did not recognize that pedestrians jaywalk

The automated car lacked "the capability to classify an object as a pedestrian unless that object was near a crosswalk," an NTSB report said.

ML Model = Unreliable Function

Object Detection Model



Building 99%
Path 97%
Plants 98%
Flowerpot 41%
Tree 4%

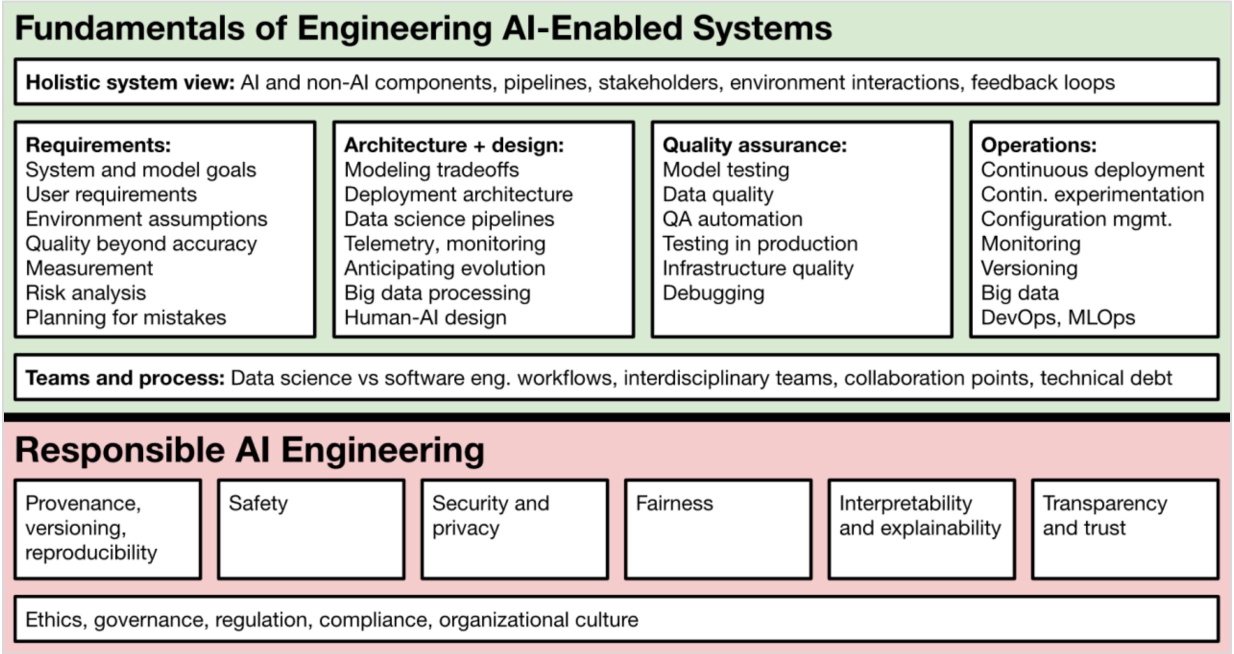
No guarantees, may make mistakes, confidence unreliable

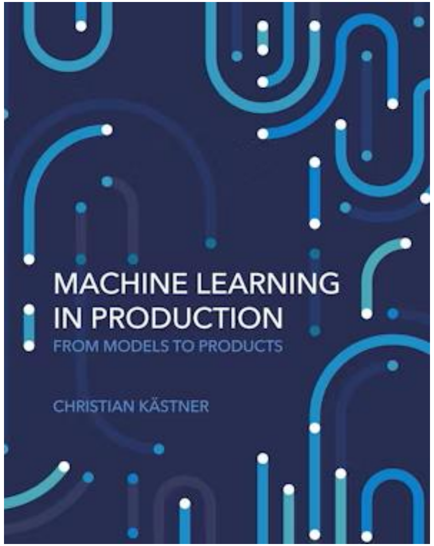
Model often inscrutable, opaque

Evaluated in terms of accuracy, not correctness

Building ML Systems

CMU 17-645: Machine Learning in Production





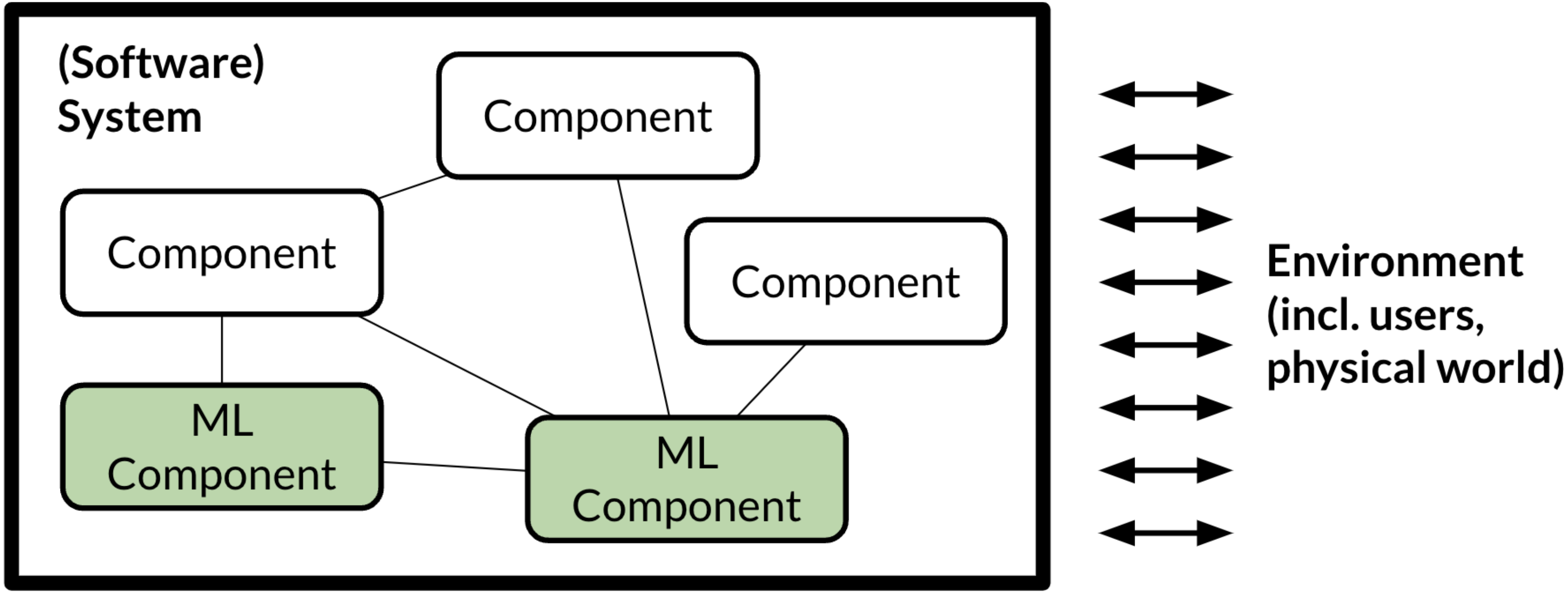
Christian Kästner, Machine Learning in Production, MIT Press, 2025.
https://mlip-cmu.github.io/book/

https://ckaestne.github.io/seai/

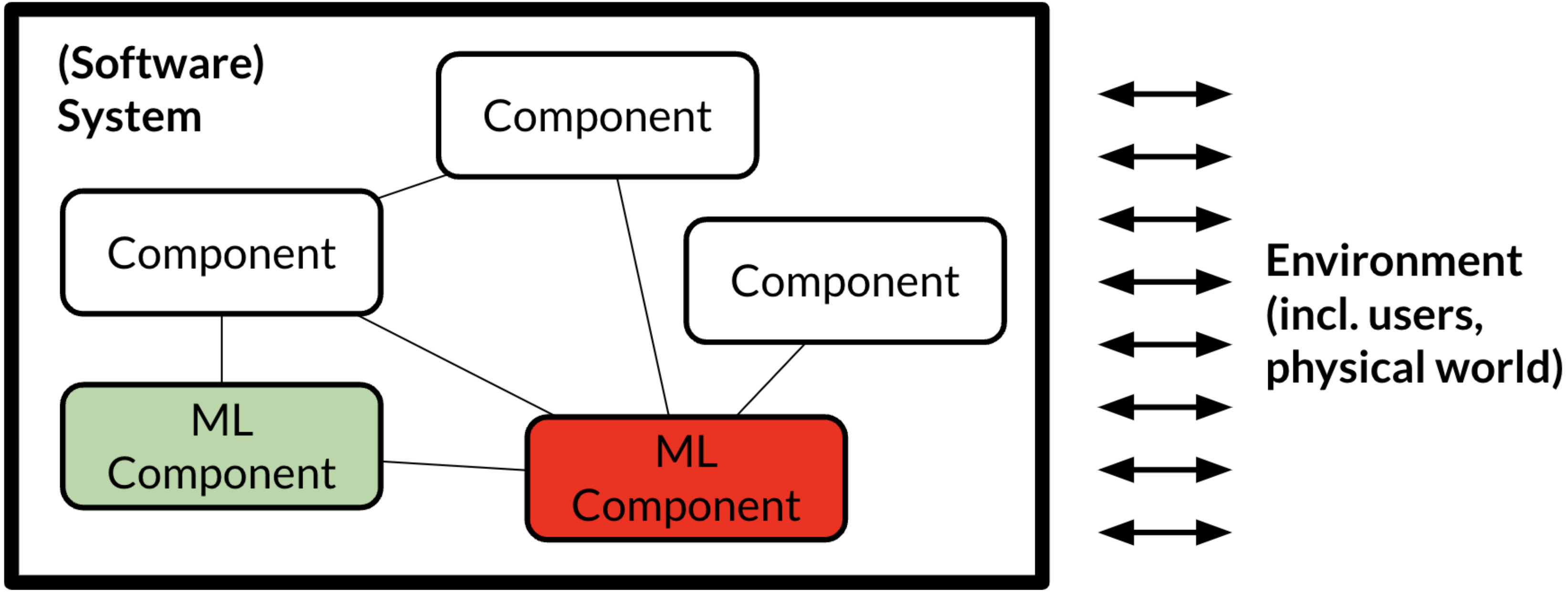
Systems Thinking

- Understand system needs and goals and interactions with environment
- Designing components and integrating ML and non-ML parts into a system
- Many roles and stakeholders, interdisciplinary endeavour

Systems Thinking



What to do when the ML component makes mistake?



Planning for Mistakes

Example: Smart Toaster

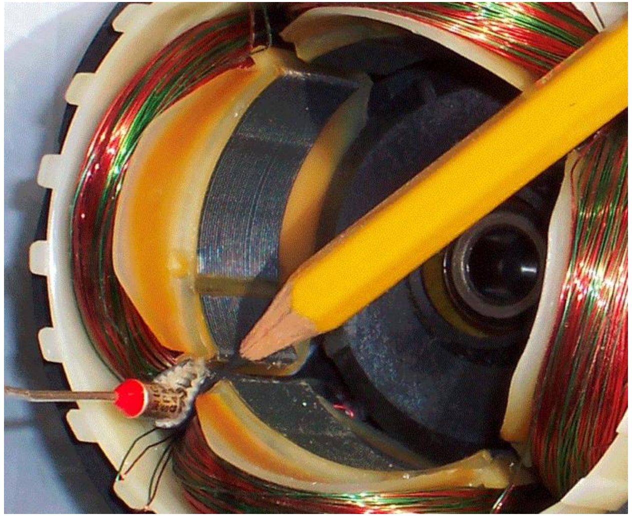
Let’s try to brainstorm:

How can you ensure that smart toaster does not burn the kitchen?

Safety Assurance in/outside the Model

In the model

- Ensure maximum toasting time
- Use heat sensor and past outputs for prediction
- Hard to make guarantees



Outside the model

- Simple code check for max toasting time
- Non-ML rule to shut down if too hot
- Hardware solution: thermal fuse

Human in the Loop

to me

Hey Nadia,

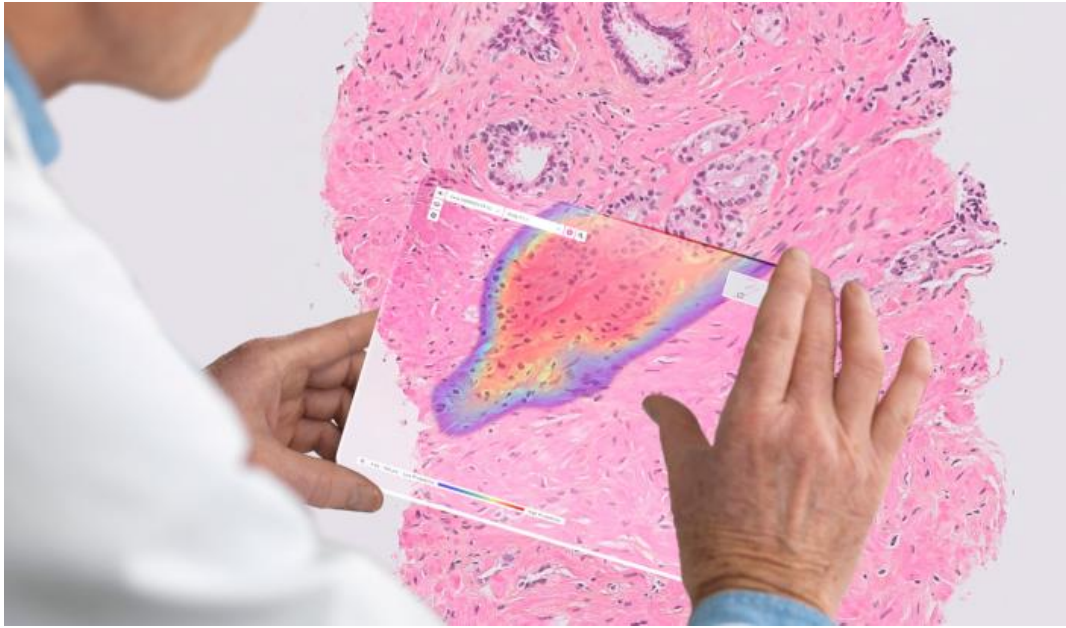
Does Wednesday work for you?

--

Sure, what time? | Yes, what time? | No, it doesn't.

Reply | Forward

Human in the Loop



AI powered diagnostic systems for cancer does not replace pathologists

Human in the Loop

Food delivery robot pauses operations after Monday incident

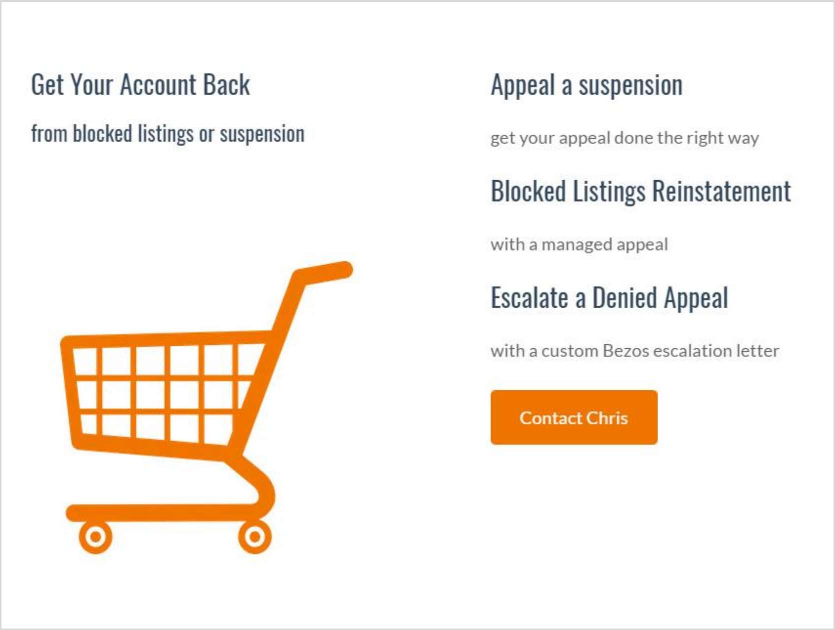
Emily Ackerman relies on a wheelchair for mobility and was trapped on Forbes Avenue when robot wouldn't move

Many different strategies

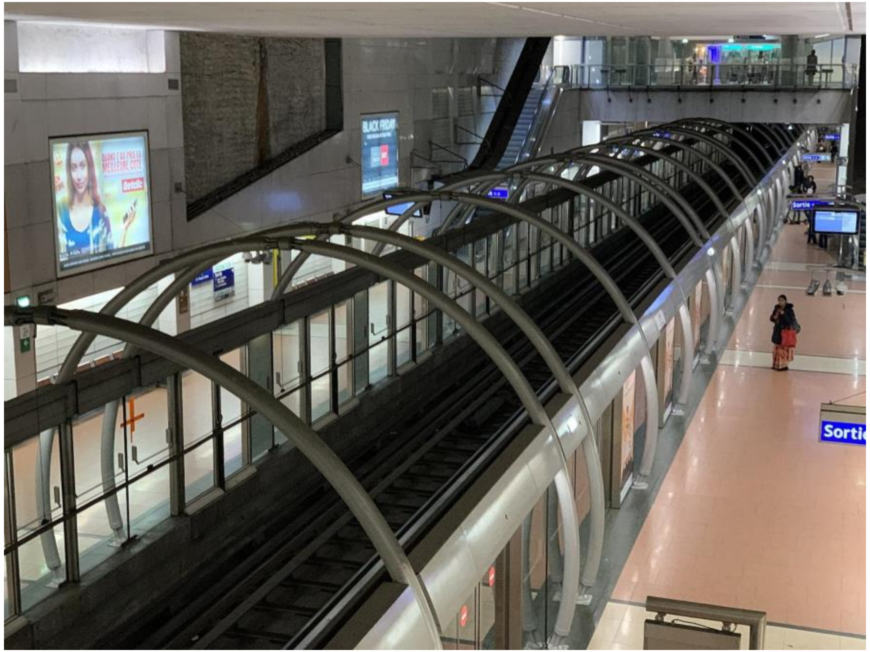
Based on fault-tolerant design, assuming that there will be software/ML mistakes or environment changes violating assumptions

- Human in the loop
- Undoable actions
- Guardrails
- Mistake detection and recovery (monitoring, doer-checker, fail-over, redundancy)
- Containment and isolation

Undoable Actions



Guardrails



Code check for max toasting time
Non-ML rule to shut down if too hot
Thermal fuse

Hazard Analysis

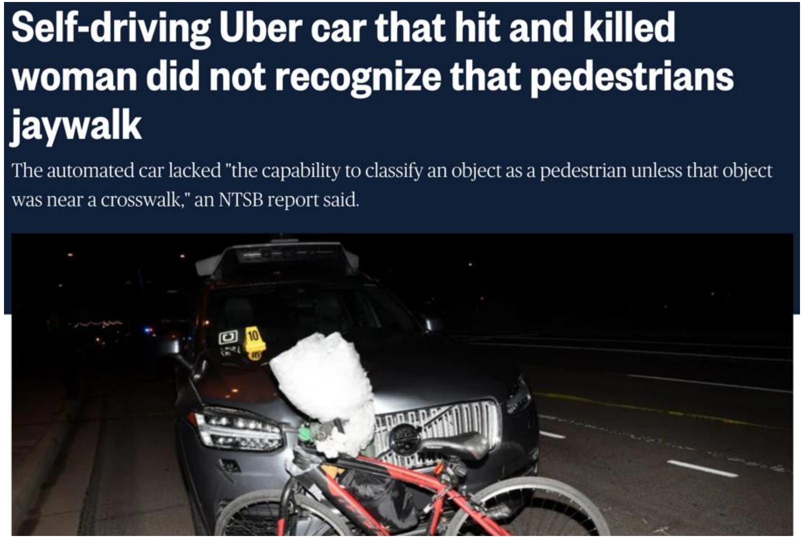
- Anticipate mistakes and their consequences.
  - Worst thing that can happen?

- Backup strategy? Undoable? Nontechnical compensation?

Fault Tree Analysis (FTA)

- Top-down, systematic method used to identify and analyze potential causes of system failures
- Visualized as a "fault tree" diagram
- Helps understand how component failures can lead to system-wide failures.

Fault Tree Analysis (FTA)



Self-driving Uber car that hit and killed woman did not recognize that pedestrians jaywalk

The automated car lacked "the capability to classify an object as a pedestrian unless that object was near a crosswalk," an NTSB report said.

Requirement:
The autonomous car shall not hit pedestrians.

Fault Tree Analysis (FTA)

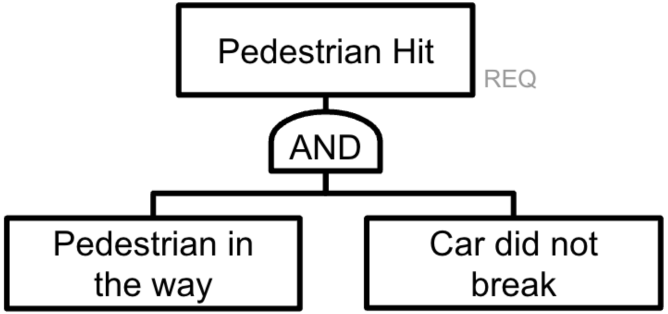
Pedestrian Hit

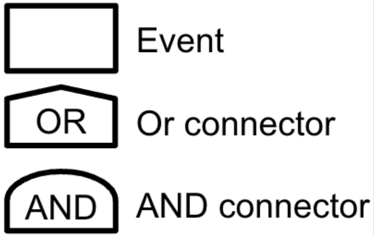
REQ

Notation:

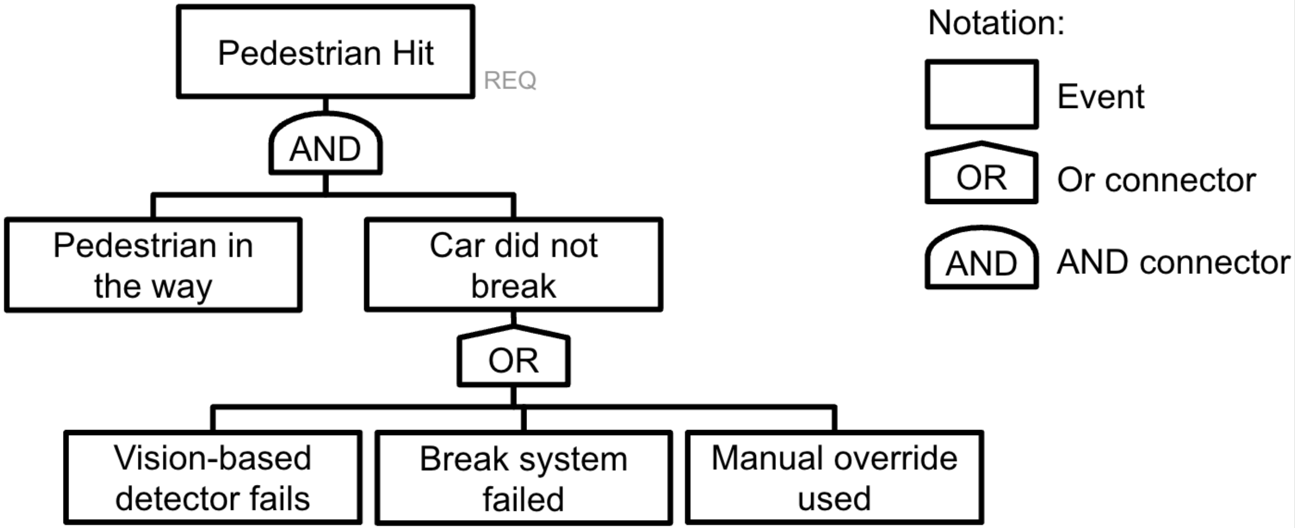
Event

Fault Tree Analysis (FTA)

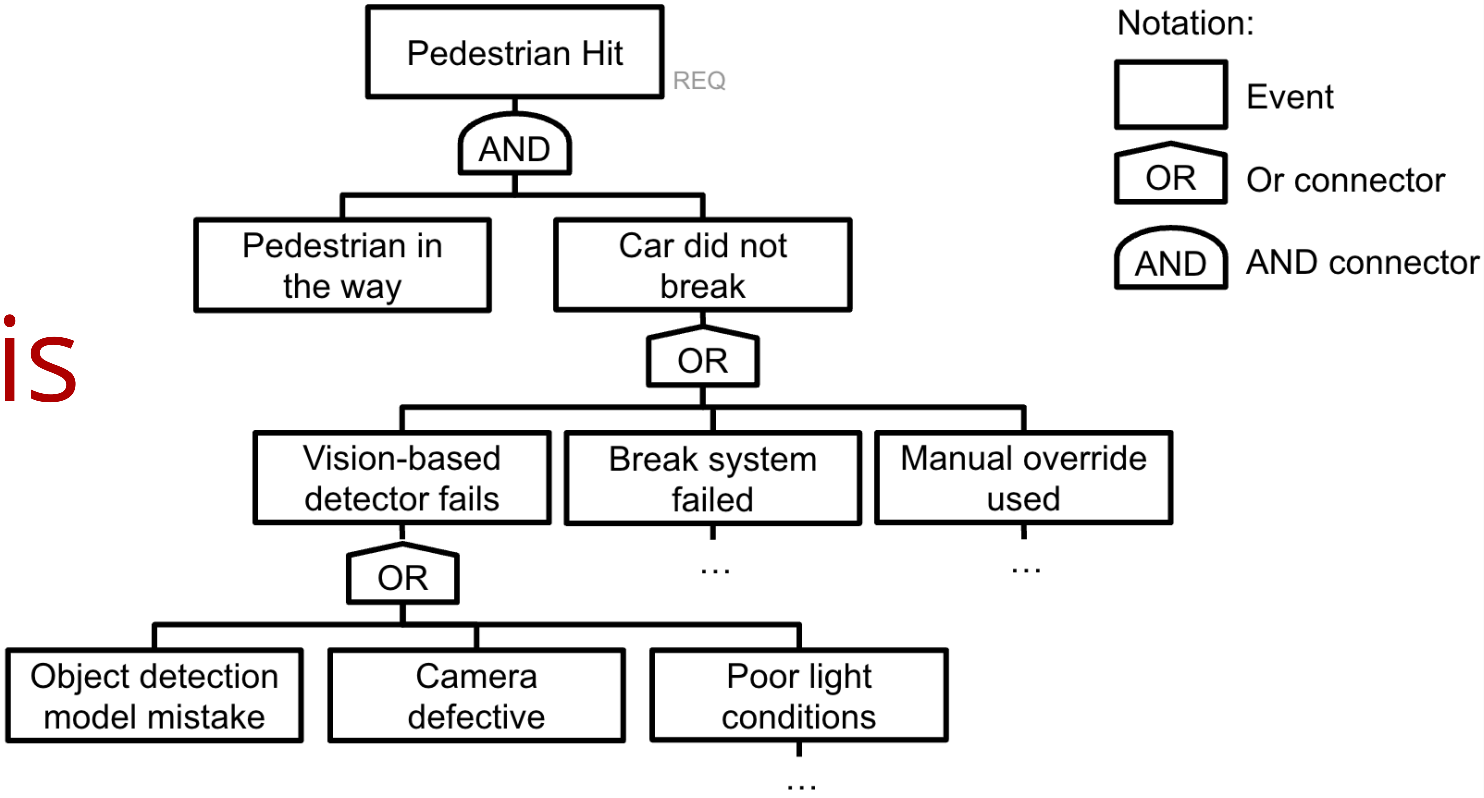




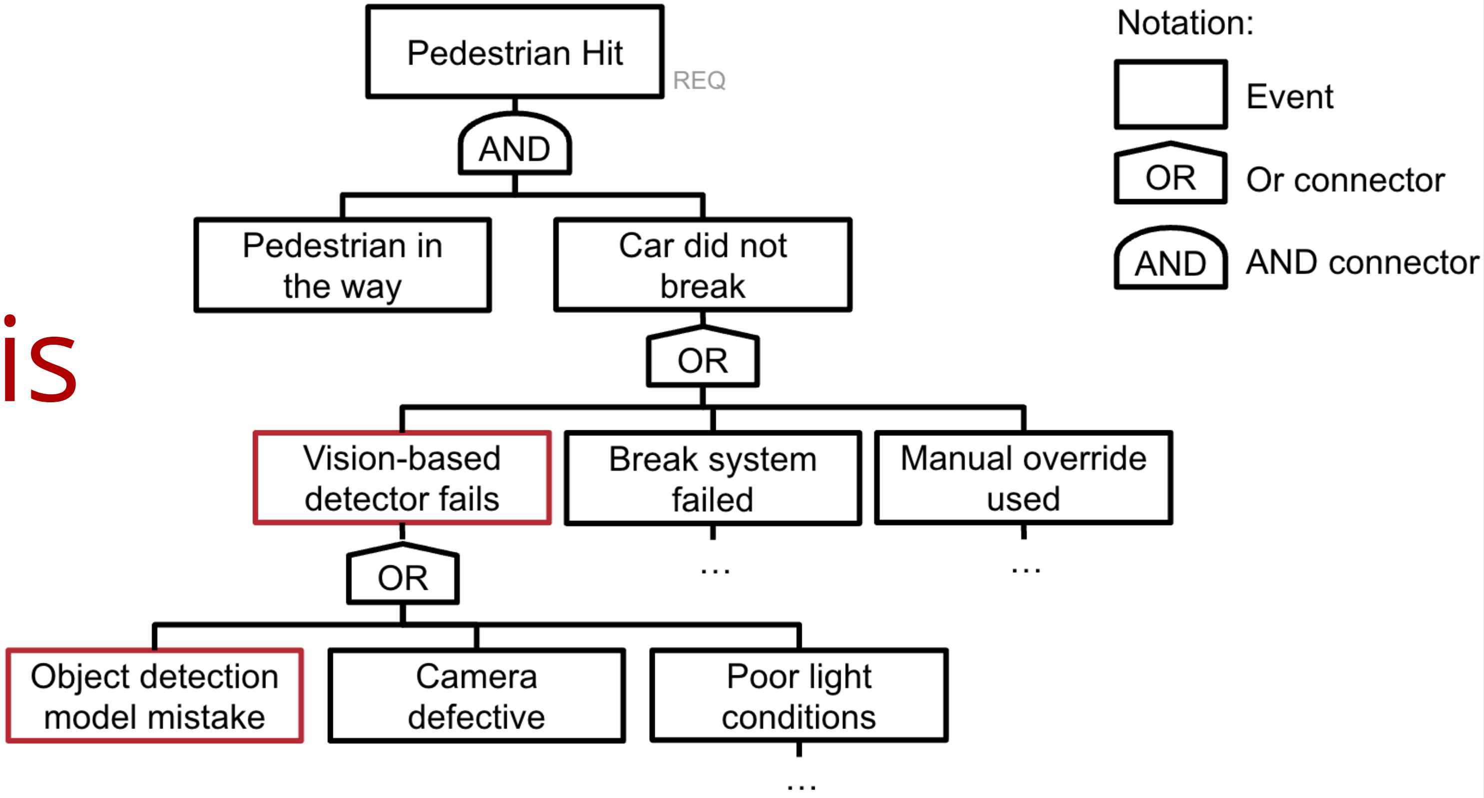
Fault Tree Analysis (FTA)



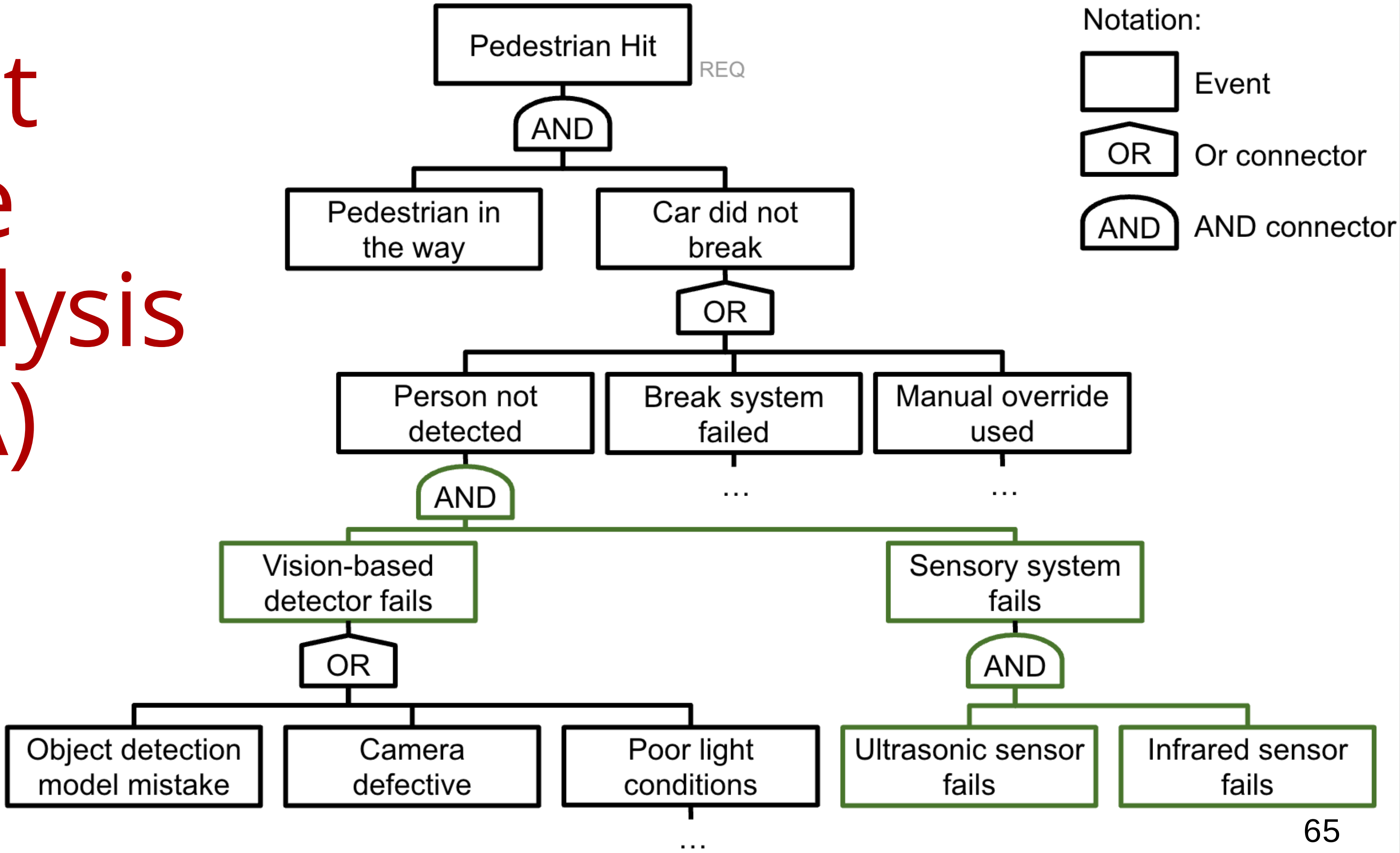
Fault Tree Analysis (FTA)



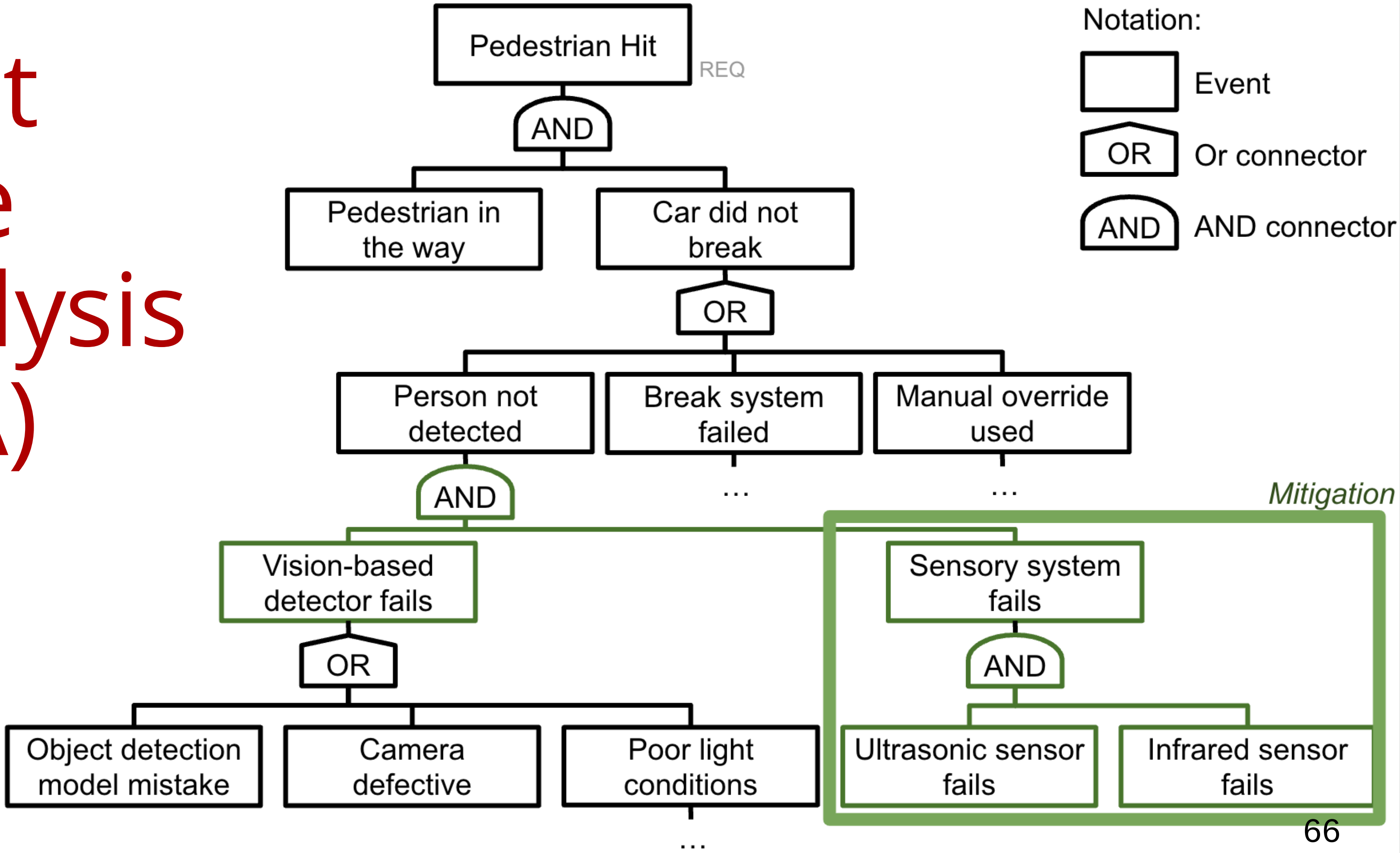
Fault Tree Analysis (FTA)



Fault Tree Analysis (FTA)



Fault Tree Analysis (FTA)



Architecting ML Systems

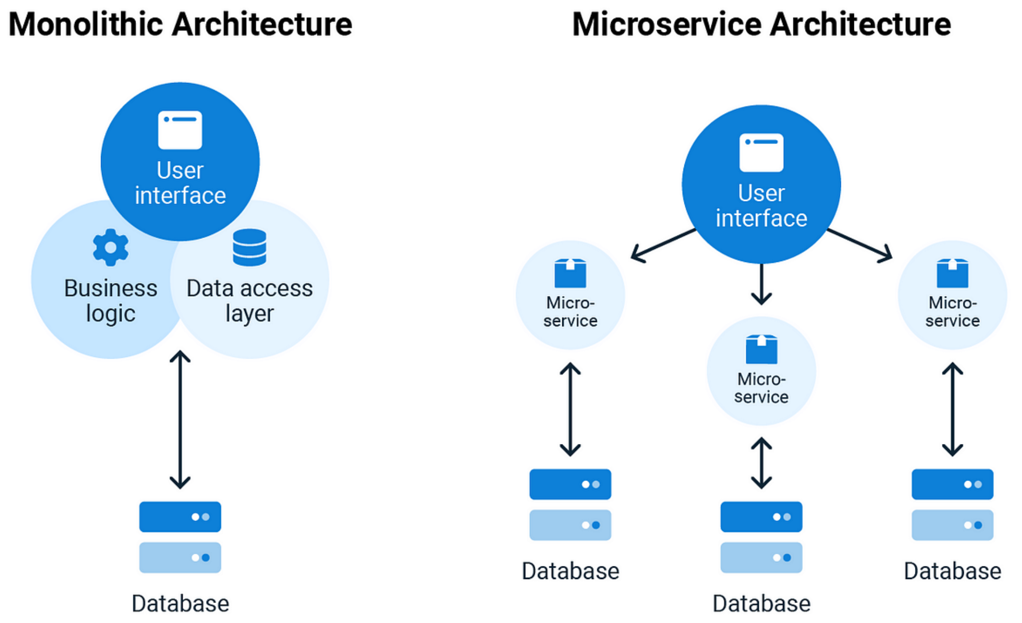
Architecture Decisions

- What are the major components in the system? What does each component do?
- Where do the components live? Monolithic vs microservices?
- How do components communicate to each other? Synchronous vs asynchronous calls?
- What API does each component publish? Who can access this API?
- Where does the ML inference happen? Client-side or server-side?
- Where is the telemetry data collected from the users stored?
- How large should the user database be? Centralized vs decentralized?
- ...and many others

Quality Requirements Drive Architecture Design

- Development cost, operational cost, time to release
- Scalability, availability, response time, throughput
- Security, safety, usability, fairness
- Ease of modifications and updates
- ML: Accuracy, ability to collect data, training latency
- ...

Architecture Design Involves Quality Trade-offs



Architecture Decision: ML Model Selection

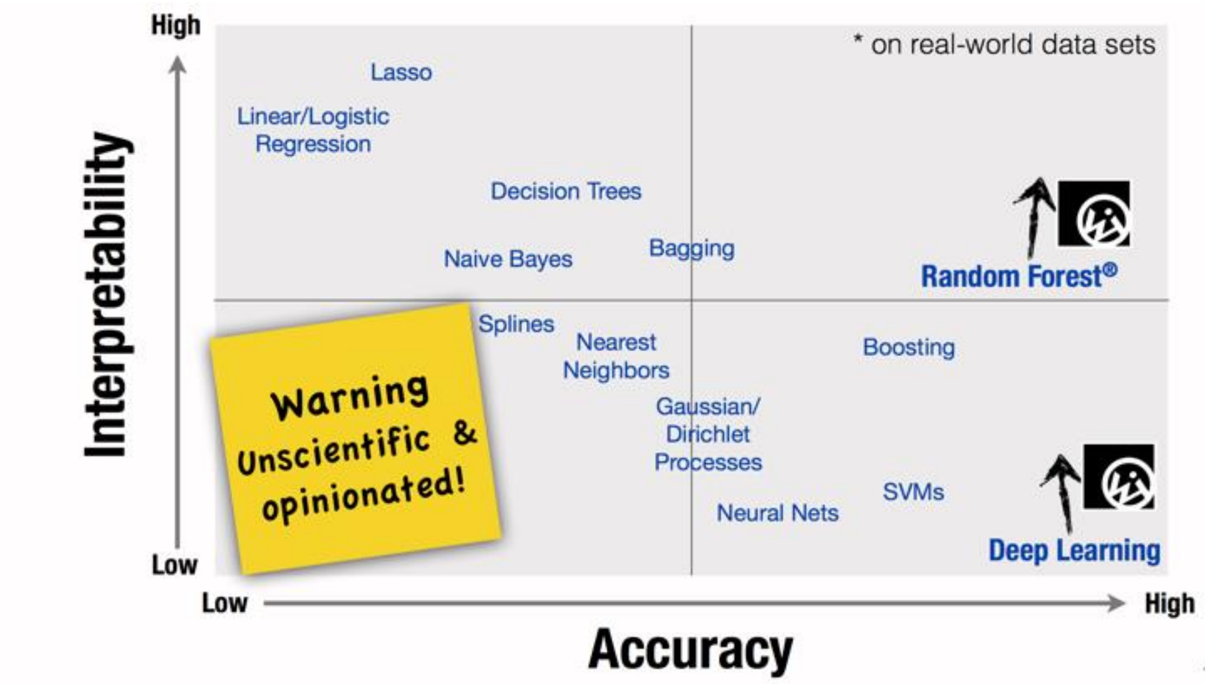
Accuracy is not Everything

ML != DL

Quality Tradeoffs

- Accuracy
- Capabilities (e.g. classification, recommendation, clustering…)
- Amount of training data needed
- Inference latency
- Learning latency
- Model size
- Explainable
- …

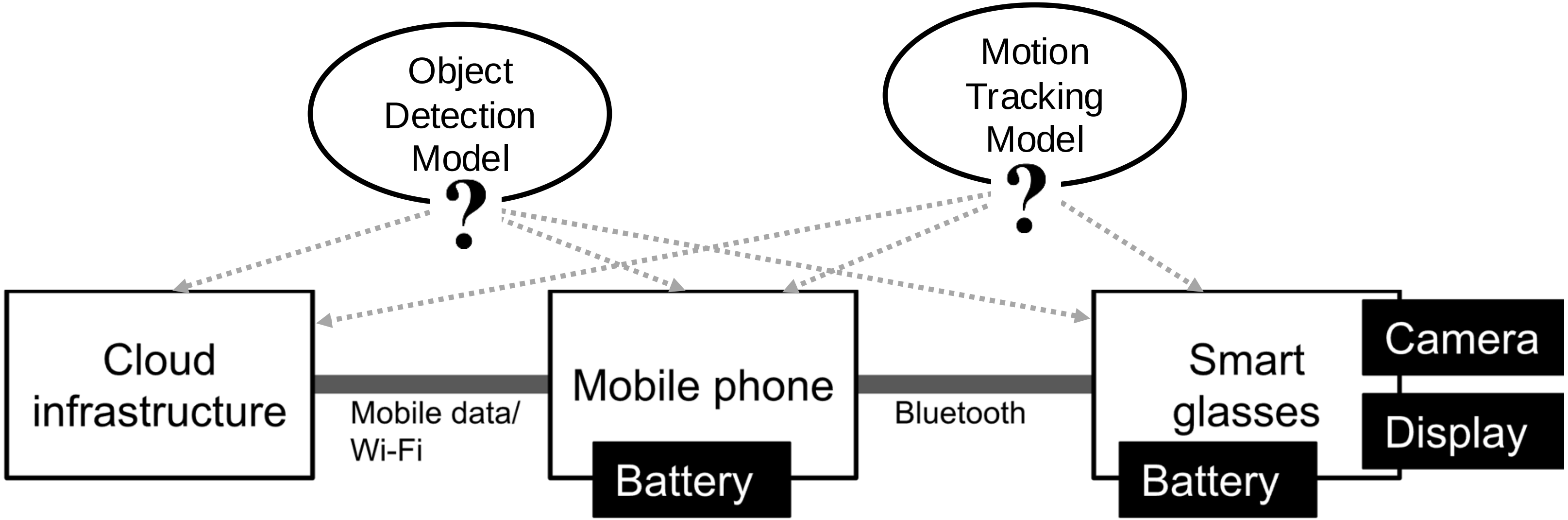
Tradeoffs: Accuracy vs Interpretability



What Qualities are Important?

Accuracy? Latency? Model Size?

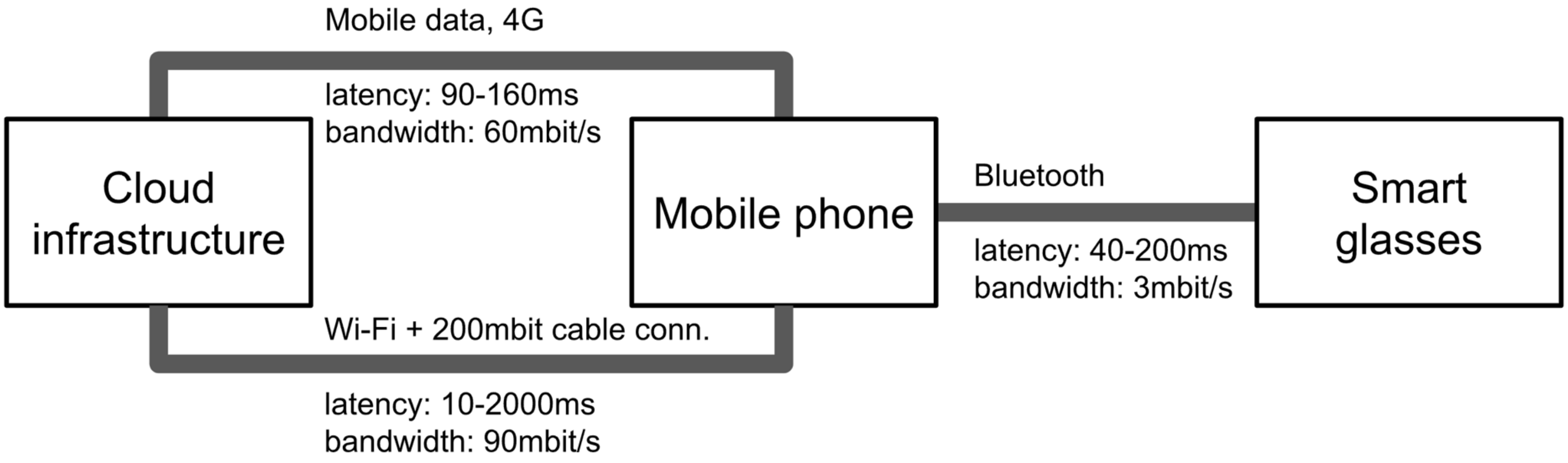
Architecture Decision: Where Should the Model Live?



Considerations

- How much data is needed as input for the model?
- How much output data is produced by the model?
- How fast/energy consuming is model execution?
- What latency is needed for the application?
- How big is the model? How often does it need to be updated?
- Cost of operating the model? (distribution + execution)
- What happens if users are offline?
- …

Latency and Bandwidth Analysis



Activity: Where should the model live?

- Discuss and decide
  - Where should the **Object Detection** component live?
    - Cloud? Phone? Glasses?
  - Where should the **Motion Tracking** component live?
    - Cloud? Phone? Glasses?
- Justify your choice
  - What qualities are relevant for the decision?