17-423/723: Designing Large-scale Software Systems

Designing AI-based Systems

April 14, 2025

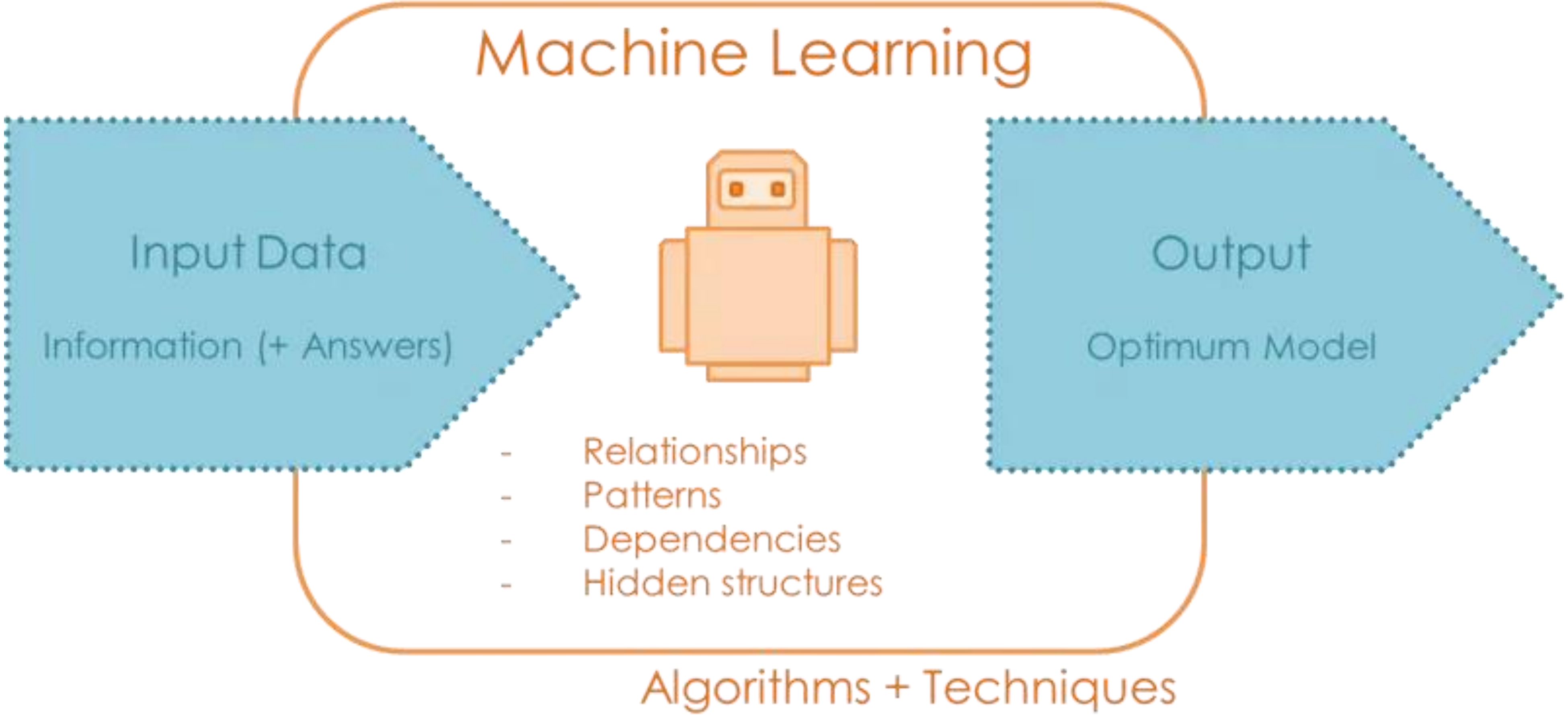
Logistics

- **M5**: Robustness testing
  - Make sure to send your target team a report of your findings by Tuesday, April 15!
- **HW2**: Scalability design
  - Released later today; due on the last day of the semester (April 25)
- **Final Exam**: 5:30-8:30 pm, May 1
  - Covers the entire course
  - Similar in style to the midterm

Learning Goals

- Understand how ML components are a (small or large) part of a larger system
- Explain how machine learning fits into the larger picture of building and maintaining production systems
- Describe the typical components relating to AI in an AI-enabled system and common design decisions to be made
- Connect system-level goals & qualities to ML-level goals & qualities
- Describe the types of mistakes in ML-based systems and techniques for mitigating them


ARTIFICIAL INTELLIGENCE
LIGHTYEAR
ARTIFICIAL INTELLIGENCE EVERYWHERE
imgflip.com



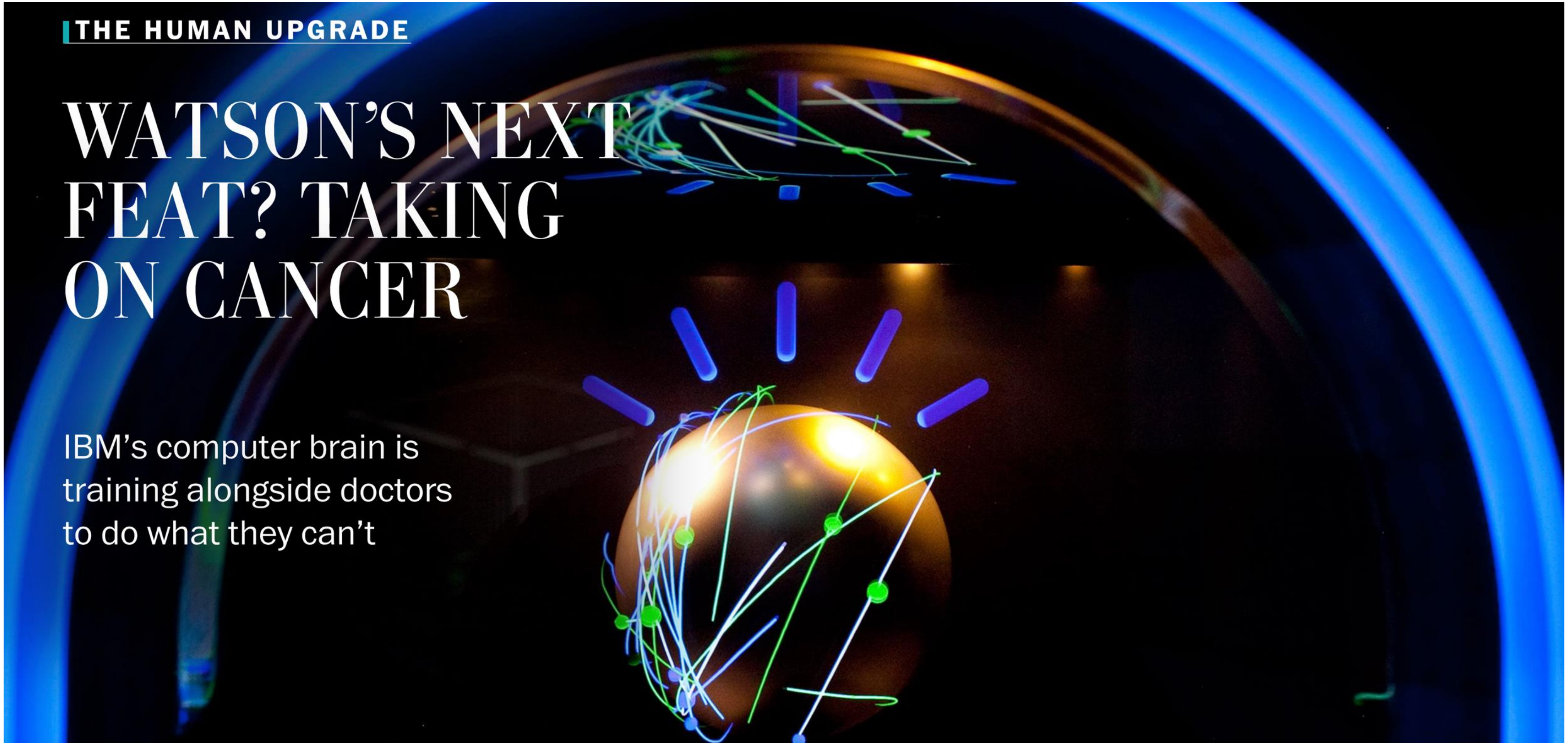
Many different types of ML models: Decision trees, logistic regression, neural networks (NNs), large language models (LLMs)…

Building a good AI-based product = Training a good ML model?

IBM Watson


PIENSE
THINK
ΣΚΕΨΟΥ
DENKE
सोचिए
PENSER
$300,000
$1,000,000
$200,000
Who is Stoker?
(I for one welcome our new computer overlords)
$ 1,000
Who is Bram Stoker?
$ 17,973
WHO IS BRAM STOKER?
$5600

IBM Watson


THE HUMAN UPGRADE
WATSON'S NEXT FEAT? TAKING ON CANCER
IBM's computer brain is training alongside doctors to do what they can't

What Went Wrong?

Poor understanding of the problem!

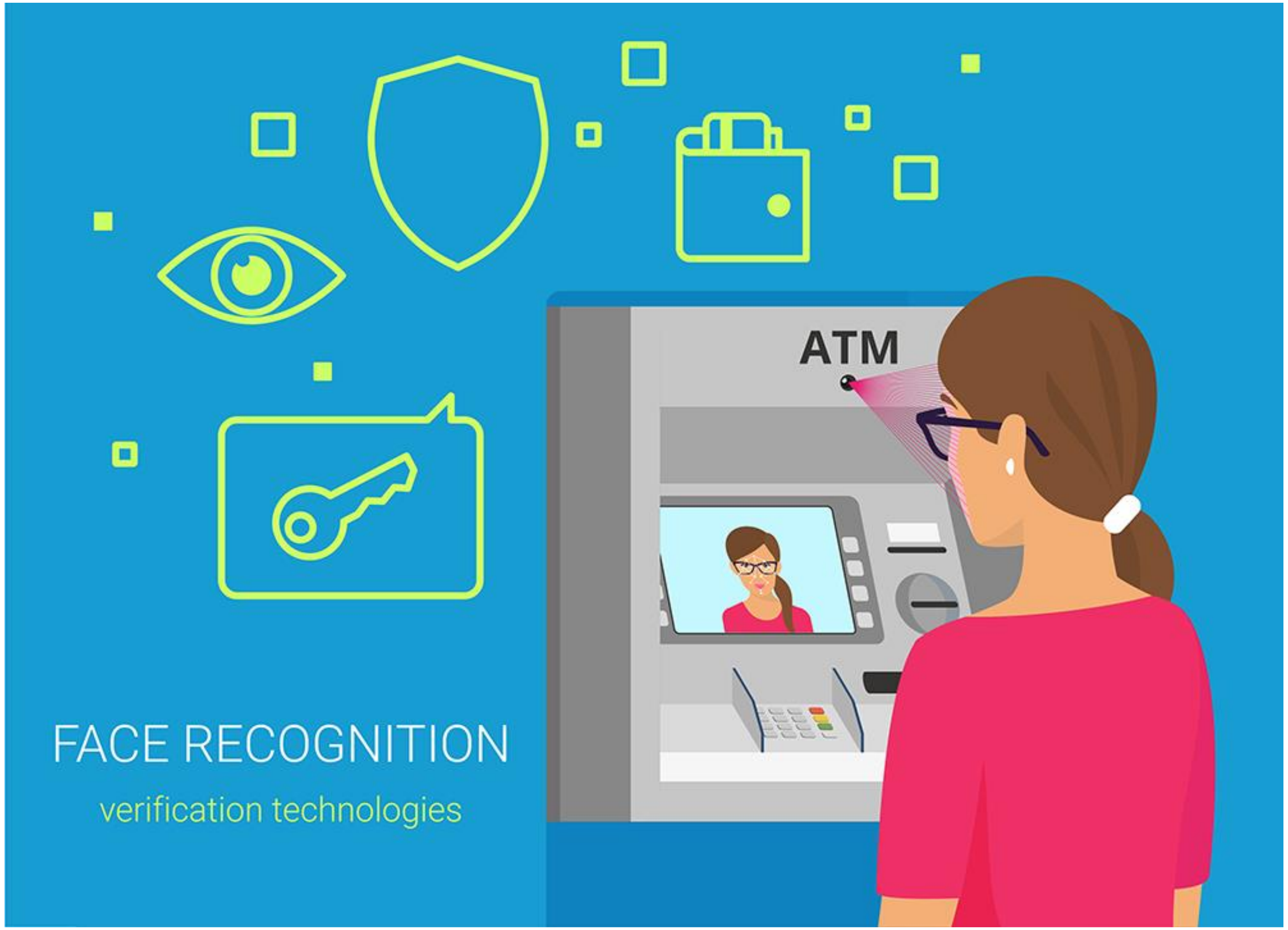
future tense

How IBM's Watson Went From the Future of Health Care to Sold Off for Parts

BY LIZZIE O'LEARY JAN 31, 2022 • 9:00 AM

We got concerns from them that the recommendations that it was giving were just not relevant...it would suggest a particular kind of treatment that wasn't available in the locality in which it was making the recommendation, or the recommendation did not at all square with the treatment protocols that were in use at the local institution...

AI-powered ATM



What Went Wrong?

Poor usability, stemming from ignoring a certain group of users!

What Went Wrong?

New types of security threats!

ITPro.

A new LLM jailbreaking technique could let users exploit AI models to detail how to make weapons and explosives — and Claude, Llama, and GPT are all at risk

LLM jailbreaking techniques have become a major worry for researchers amid concerns that models could be used by threat actors to access harmful information

Building a good AI-based product $\cancel{=}$
Training a good ML model?

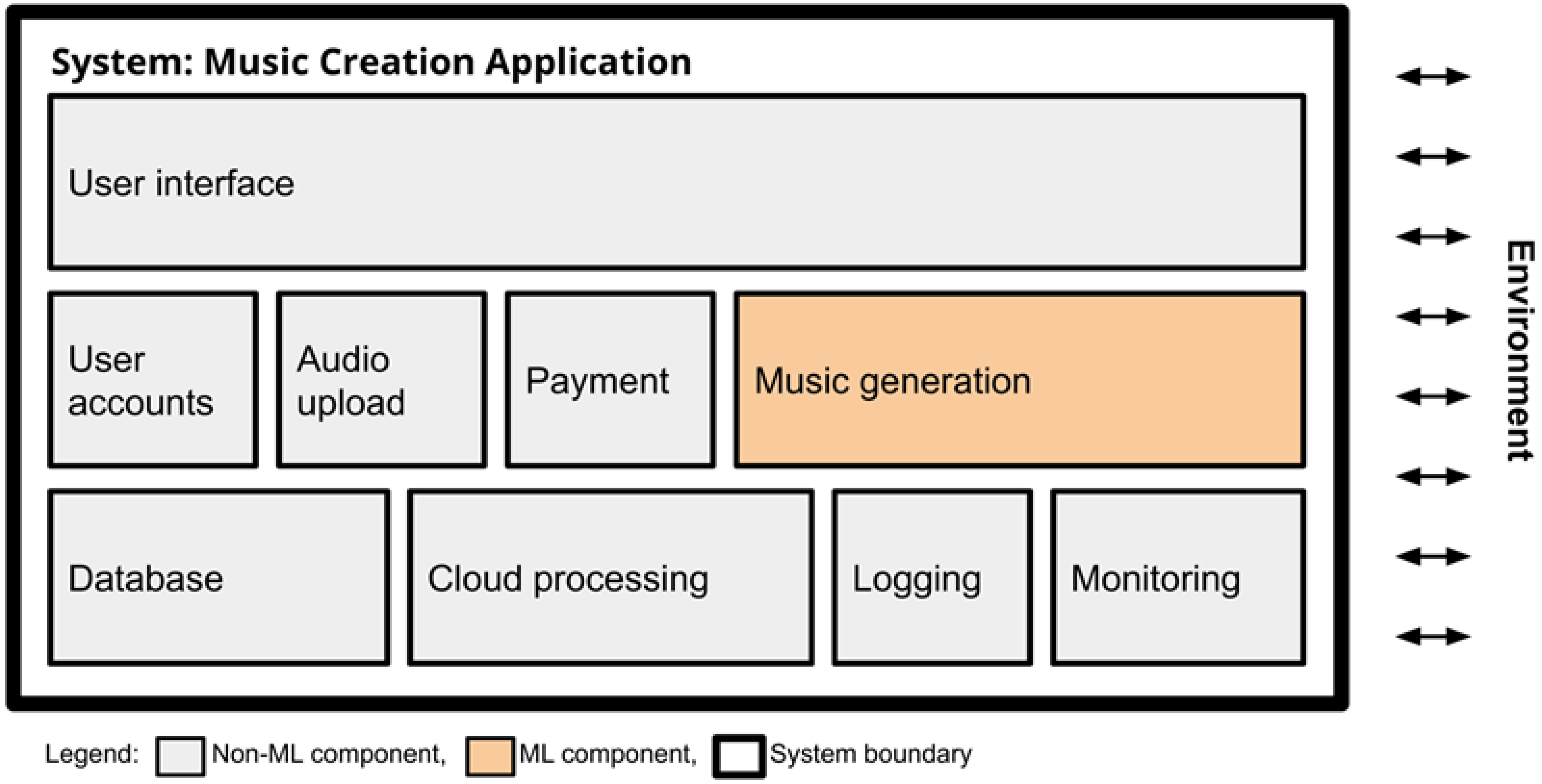
Other examples that you’ve encountered?

Misconception

- ~~Building a good AI-based product = Training a good ML model~~

ML as a System Component

- ~~Building a good AI-based product = Training a good ML model~~
- An ML model is just **one part of the system**



ML as a System Component

- ~~Building a good AI-based product = Training a good ML model~~
- An ML model is just **<u>one part of the system</u>**
- An accurate ML model does not necessarily result in a successful software product
- Quality attributes (QAs) covered throughout this class—such as changeability, robustness, scalability, security, and usability—are all important to these products
- Design techniques and principles we’ve covered will still apply
  - **But there are also some unique challenges introduced by AI**
  - **<u>Q. What are some of these challenges?</u>**

Challenges: Building AI-based Systems

ML models are unreliable components

- Traditional software components: Specify what to do & how to test
  - Recall: Interface specifications!
- ML models are usually black boxes; difficult to understand its internals
- Even if you provide a specification, e.g., LLM prompts, unclear if they will really follow it

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

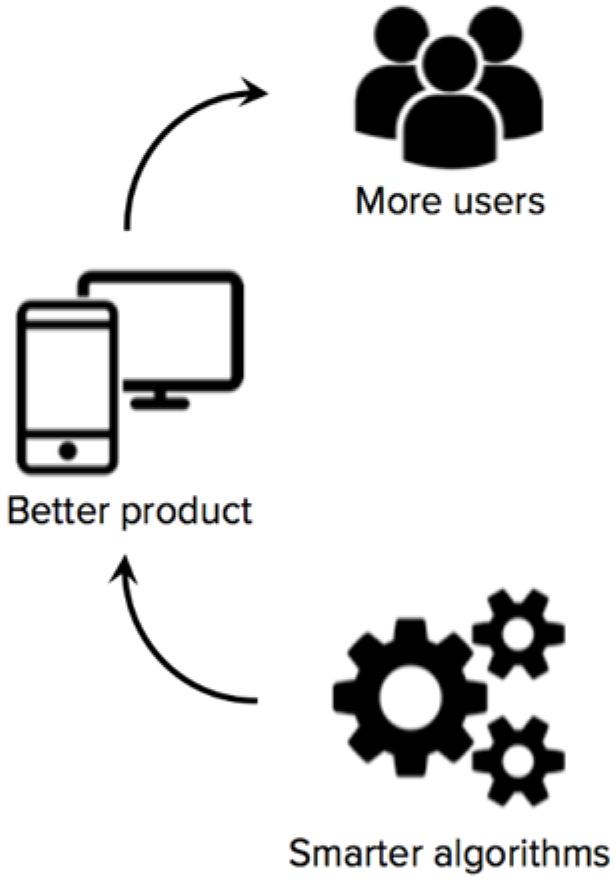
ML makes different types of mistakes

- …often in unexpected ways
- What does it mean to be correct? Can only evaluate whether it works well enough (on average) on some test data!
- Often makes mistakes on unseen & rare inputs
- But sometimes on familiar inputs, too
  - e.g., adversarial examples

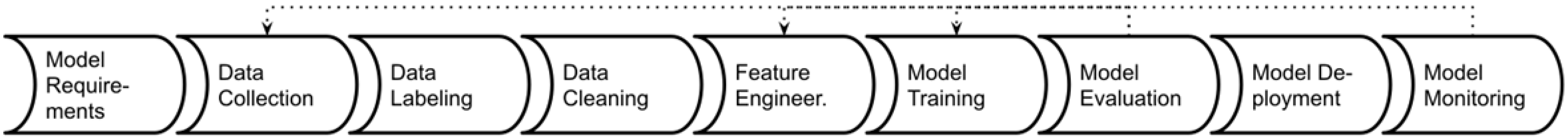
NeuralTalk2: A flock of birds flying in the air
Microsoft Azure: A group of giraffe standing next to a tree
Image: Fred Dunn, https://www.flickr.com/photos/gratapictures - CC-BY-NC

Data determines the behavior of ML models

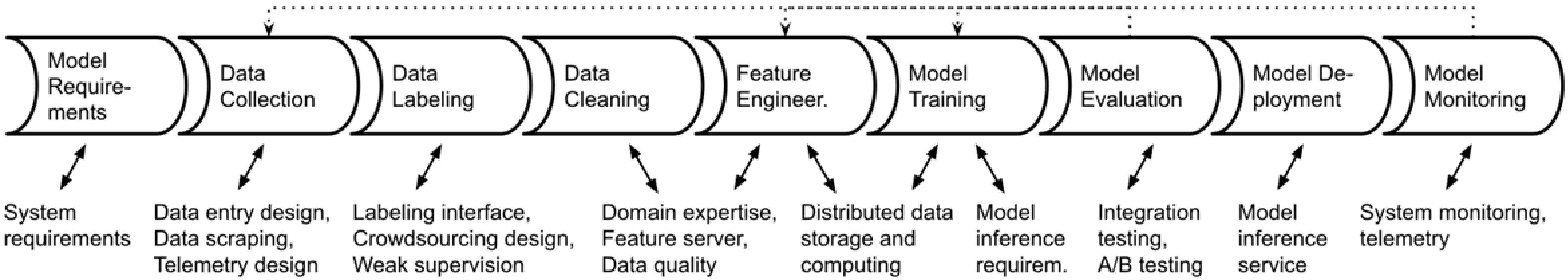
- Quality of a model depends highly on the quality of the data
  - Garbage-in, garbage-out (GIGO)
- But getting high quality data is **really hard** in general!
- Data collection, analysis, and cleaning are key tasks in developing AI products
- These are typically expertise of data scientists
  - Need collaboration with software engineers



ML Pipelines



ML Pipelines



Design Decisions in AI-based Systems

- What are system-level qualities that are important for the system? Based on those, what type of ML technique/model should be used?
- How do we collect and curate data to train & test the model?
- Which part of the system should the ML model be deployed? How do we scale the system with the model?
- How do we ensure that the overall system is reliable & safe even if the model makes mistakes?
- How do we monitor the performance of the model and update it if necessary?
- How do we design the system to be usable?
- How do we ensure that the system does not cause moral/ethical harms to the users?

System-level vs. ML-level Qualities

Design Decision: ML Model Design

- What are system-level qualities that are important for a successful system/product?
- What are qualities of an ML model are needed to achieve the system-level qualities?
- What type of ML technique/model should I use to achieve those qualities?

Model accuracy is not everything!

- **Q. Beyond prediction accuracy, what qualities may be relevant for an ML component?**

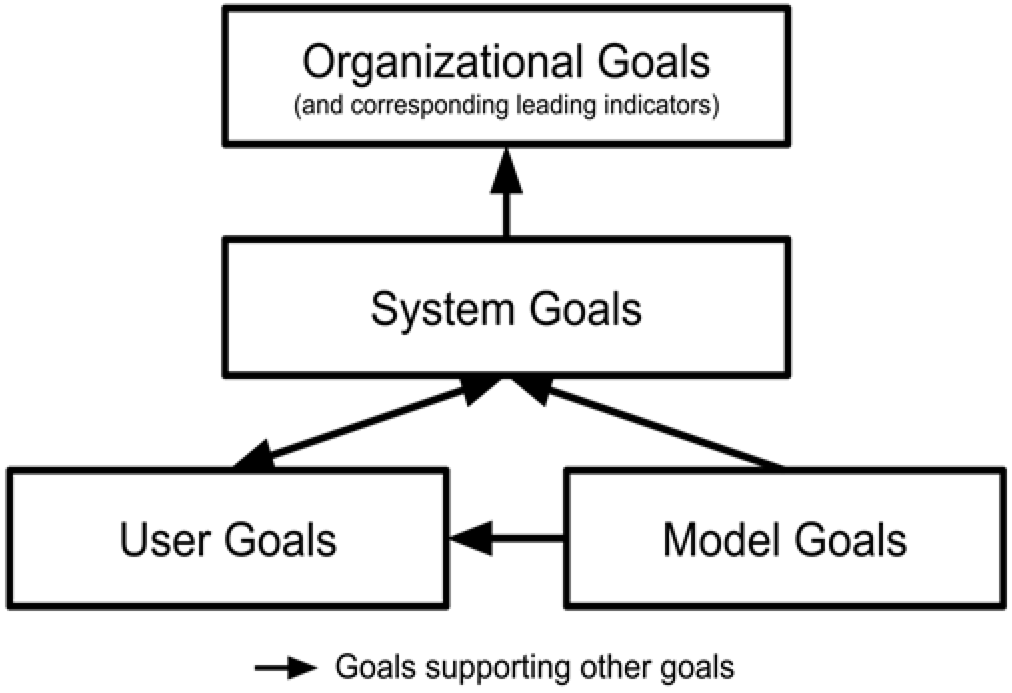
Common ML Qualities to Consider

- Accuracy
- How much data is needed for training? Is data quality important?
- How many features are needed?
- Training time, memory need, model size -- depending on training data volume and feature size
- Is incremental training possible?
- Inference time, energy efficiency, resources needed, scalability
- Interpretability, explainability
- Robustness, reproducibility, stability
- Security, privacy, fairness

Connecting System-level to ML-level Goals

- **Organizational objectives**: Innate/overall goals of the organization
- **System goals**: Goals of the software system/product/feature to be built
- **User goals**: How well the system is serving its users, from the user's perspective
- **Model goals**: Quality of the model used in a system, from the model's perspective

Ideally, these goals should be aligned with each other!



Organization Goals

Innate/overall goals of the organization

- Business
  - Current/future revenue, profit
  - Reduce business risks
- Non-Profits
  - Lives saved, animal welfare increased, CO_2 reduced, fires averted
  - Social justice improved, well-being elevated, fairness improved
- Often not directly measurable from system output; slow indicators

Implication: Accurate ML models themselves are not the ultimate goal!

ML may only indirectly influence such organizational objectives; influence is often hard to quantify; lagging measures

Leading Indicators

Short-term proxies for long-term measures

Typically measures correlating with future success, from the business perspective

Examples:

- Customers sentiment: Do they like the product? (e.g., surveys)
- Customer engagement: How often do they use the product?
  - Regular use, time spent on site, messages posted
  - Growing user numbers, recommendations

Caveats:

- Often indirect, proxy measures
- Can be misleading (e.g., more daily active users => higher profits?)

System Goals

Concrete outputs the system (or feature of the system) should produce

Relates to system requirements & quality attributes

Examples:

- Detect cancer in radiology scans
- Make personalized music recommendations
- Transcribe audio files
- Provide legal help with a self-service chatbot
- Drive a vehicle autonomously to a destination

User Goals

How well the system is serving its users, from the user's perspective

Examples:

- Users choosing recommended items and enjoying them
- Users making better decisions
- Users saving time thanks to the system
- Users achieving their goals

Easier and more granular to measure, but possibly only indirect relation to organization/system objectives

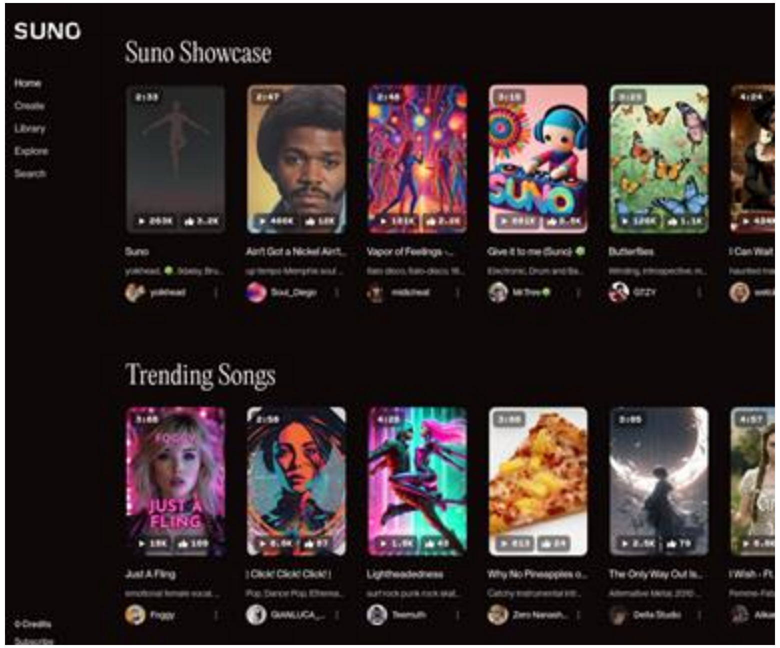
Model Goals

Qualities of the model used in a system, from the model's perspective

- Model accuracy
- Rate and kinds of mistakes
- Successful user interactions
- Inference time
- Training cost

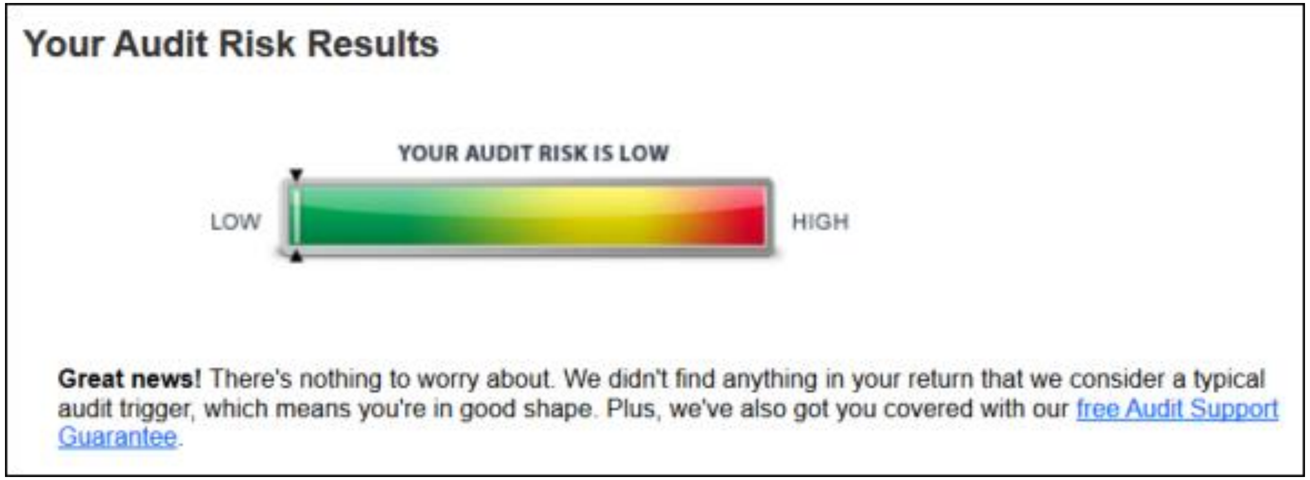
Often not directly linked to organizational/system/user goals

Success Measures in Music Player?



Organizational goals? Leading indicators? System goals? User goals? Model goals?

Success Measures in Tax Audit Risk Scoring?



Organizational goals? Leading indicators? System goals? User goals? Model goals?

Deploying ML Models

Example: Real-time Language Translation

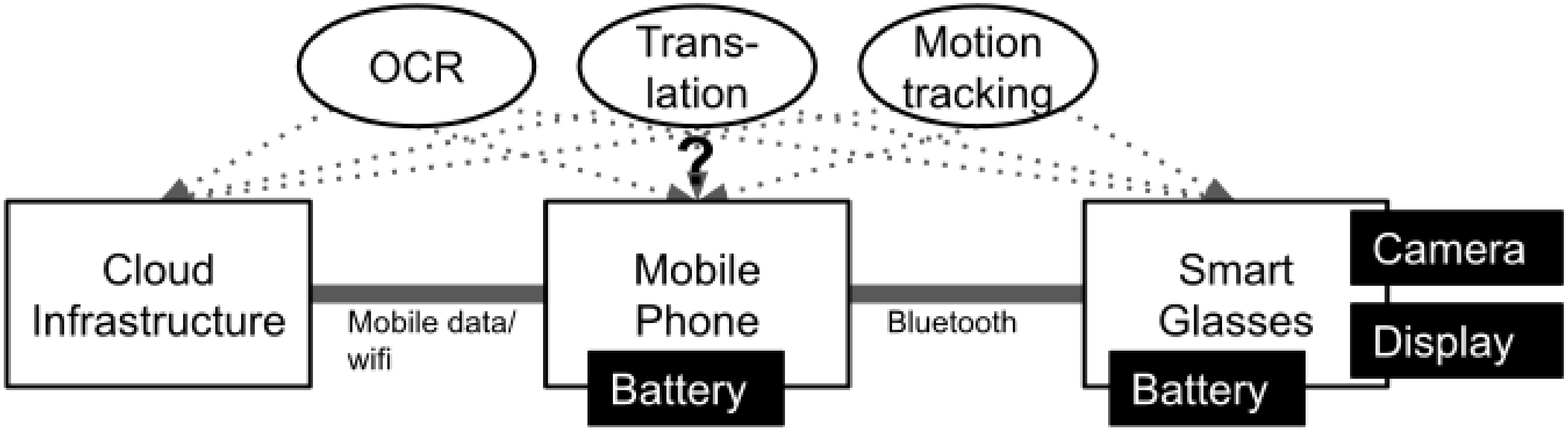
Example: Real-time Language Translation

ML Models in Real-time Translation App

- **Motion tracking**: Track the real-world objects as the user's device moves around
- **OCR**: Translate the scanned image into text
- **Translation**: From one language to another

Q. What are important system-level goals and qualities?

Design Decision: ML Model Deployment



Where should the different models live?

Cloud? Phone? Glasses?

What model qualities are relevant for the decision?

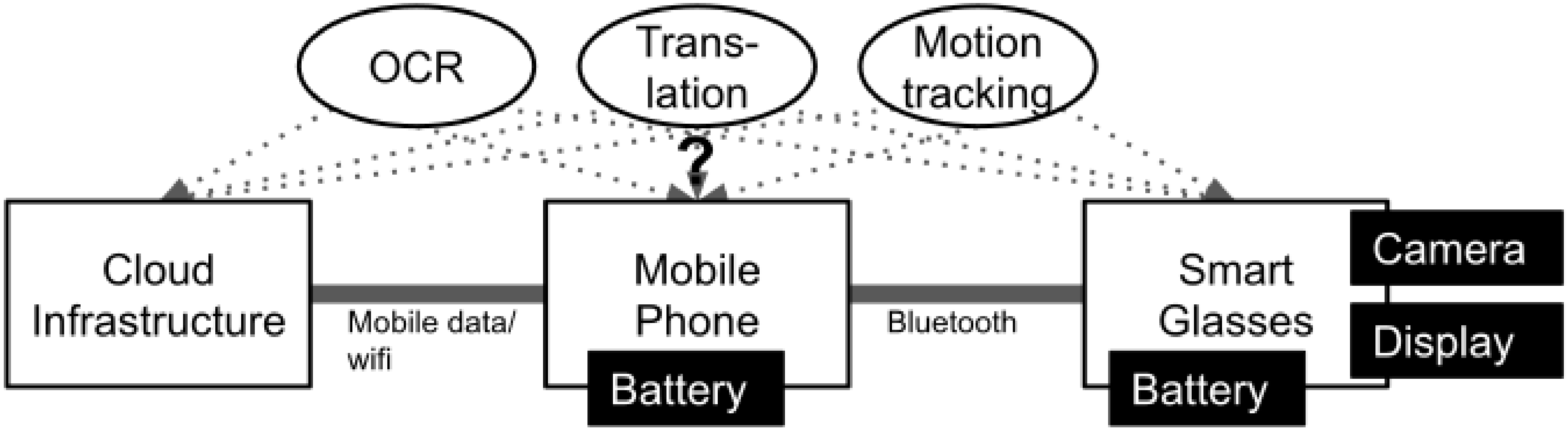
Design Considerations

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

ML Models in Real-time Translation App

- **OCR**: Translate the scanned image into text
  - Input: High-resolution image (~3MB)
  - Output: Extracted text (~100 bytes)
  - Model size: ~5 MB
- **Translation**: From one language to another
  - Input & output: Text (~100 bytes)
  - Model size: 3MB to 200GB
- For inference cost (time & energy consumption), assume proportional to the model size

Discussion: ML Model Deployment



For each of OCR & Translation:

- Where would you deploy the model? Why?
- What are QA trade-offs between different options?

Dealing with ML Mistakes

ML as Inductive Programming

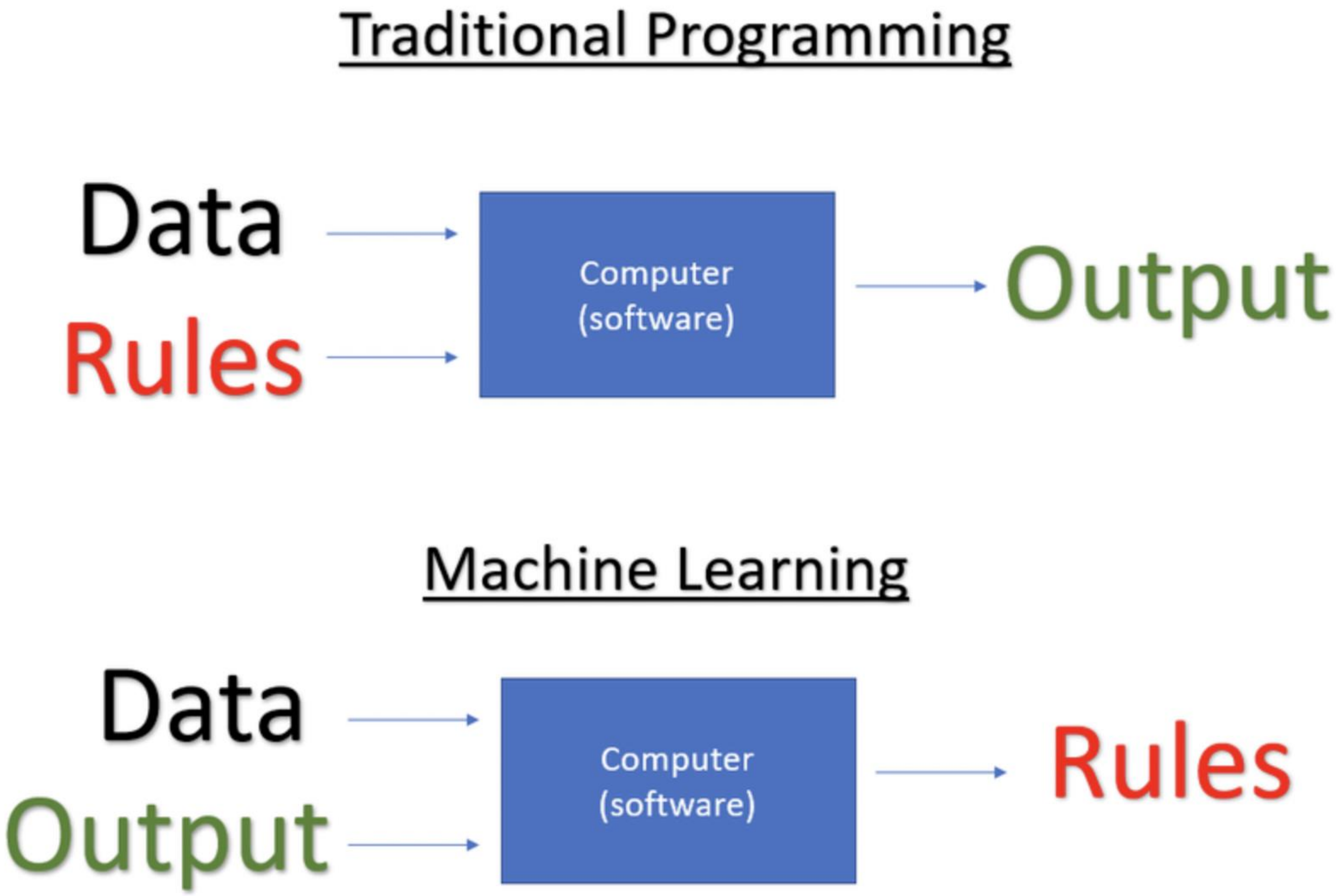


Image by Jorge Vallego under Creative Commons license

ML models make unpredictable mistakes

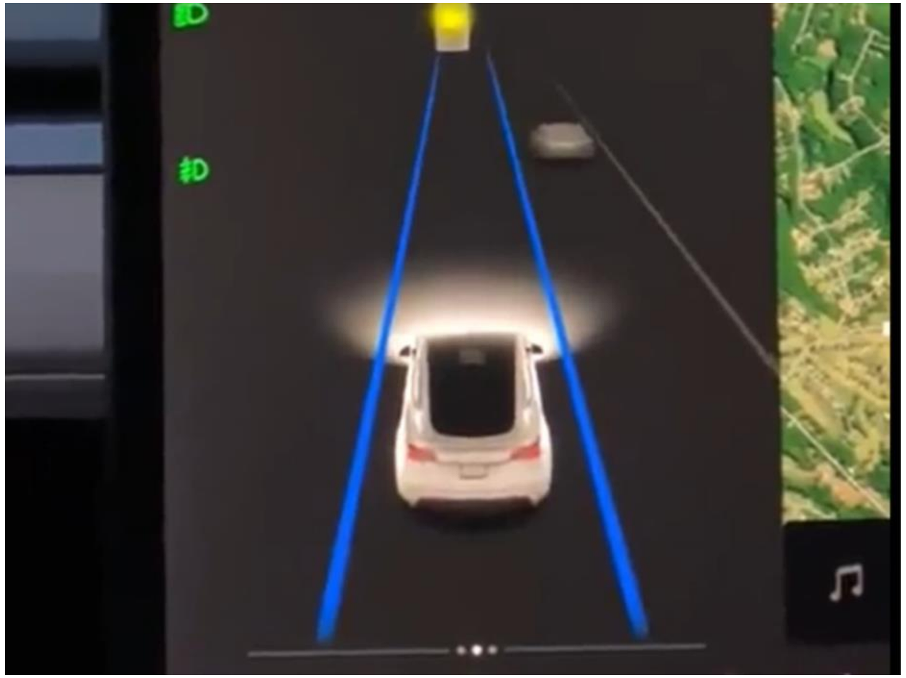
- ML models may be **<u>wildly wrong</u>** when they are wrong
  - May learn (spurious) correlations that humans would never think about
  - May be very confident about wrong answer
  - ”Fixing" one mistake through re-training may cause other errors
- vs. errors in software components
  - Well-understood classes of errors (e.g., array out-of-bounds)
  - Can leverage testing & specification to find & fix them
  - But as complexity increases, certain errors also become difficult to find/predict

ML models make unpredictable mistakes

Tesla's Autopilot Feature Mistakes Moon for Yellow Traffic Light, Watch Video

Curated By : Buzz Staff Trending Desk

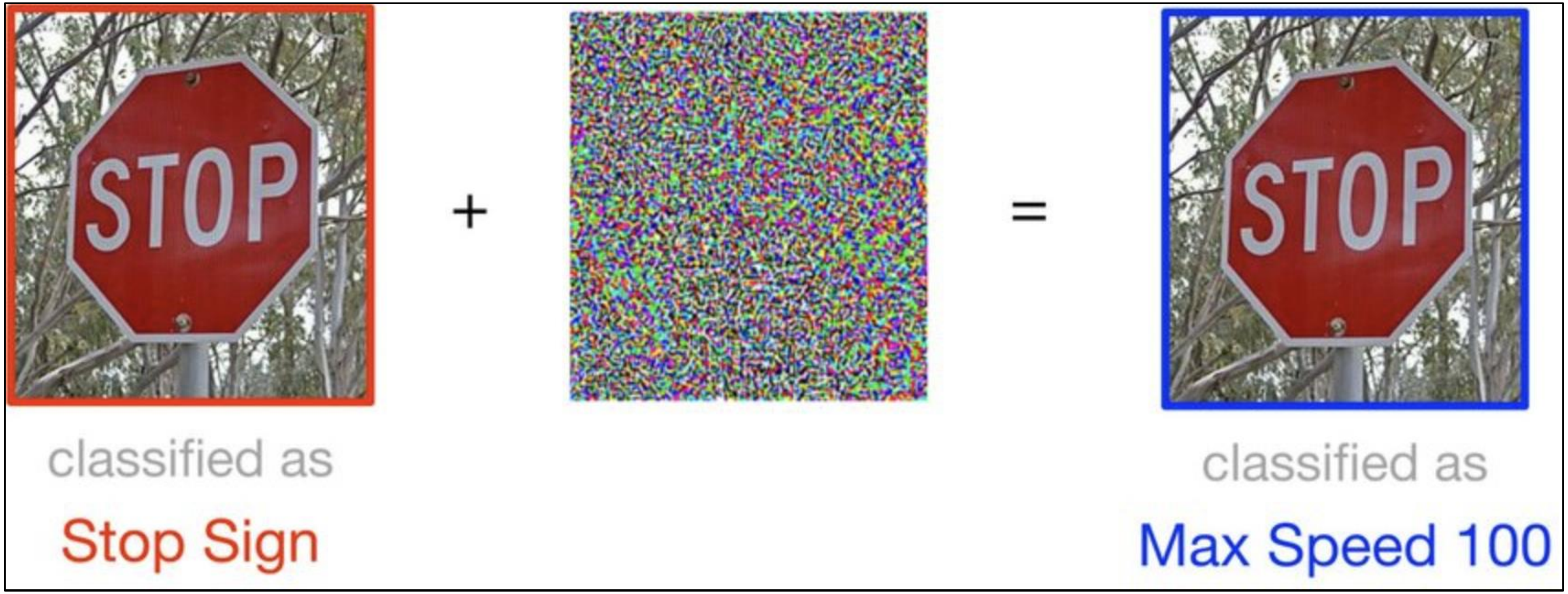
Last Updated: July 27, 2021, 02:32 IST



https://x.com/JordanTeslaTech/status/1418413307862585344

ML models do poorly on unseen data

ML models are vulnerable to attacks



- **Adversarial attacks**: Add noise to cause the model to produce an incorrect prediction

It Is All About Data: A Survey on the Effects of Data on Adversarial Robustness. Xiong et al. (2023)

Common Reasons behind Mistakes

- Insufficient training data
- Noisy or biased training data
- Overfitting
- Poor model fit, poor model selection, poor hyperparameters
- Missing context, missing important features
- "Out of distribution" inputs
- Data drifts
- Security attacks
- …

Living with ML Mistakes

- No model will ever be "correct”
  - Some mistakes are unavoidable, no matter how much training data
  - Correctness is even more difficult for modern AI models like LLMs
  - ML models continuously change; new behaviors are introduced
- **Design goal**: Make the **<u>system</u>** safe & reliable despite ML mistakes
- **Good news**:
  - This is not new; software components are also known to be unreliable
    - Although the types of errors and reasons behind them are quite different!
  - Many techniques & tools for making systems robust (recall the lectures on Robustness)

Recall: Design Patterns For Robustness

- Guardrails
- Redundancy
- Separation
- Graceful degradation
- Human in the loop
- Undoable actions

Example: Smart Toaster

- Use an ML model to detect when the toasts are done
- **Q. How to ensure that the toaster does not overheat and burn down the kitchen?**

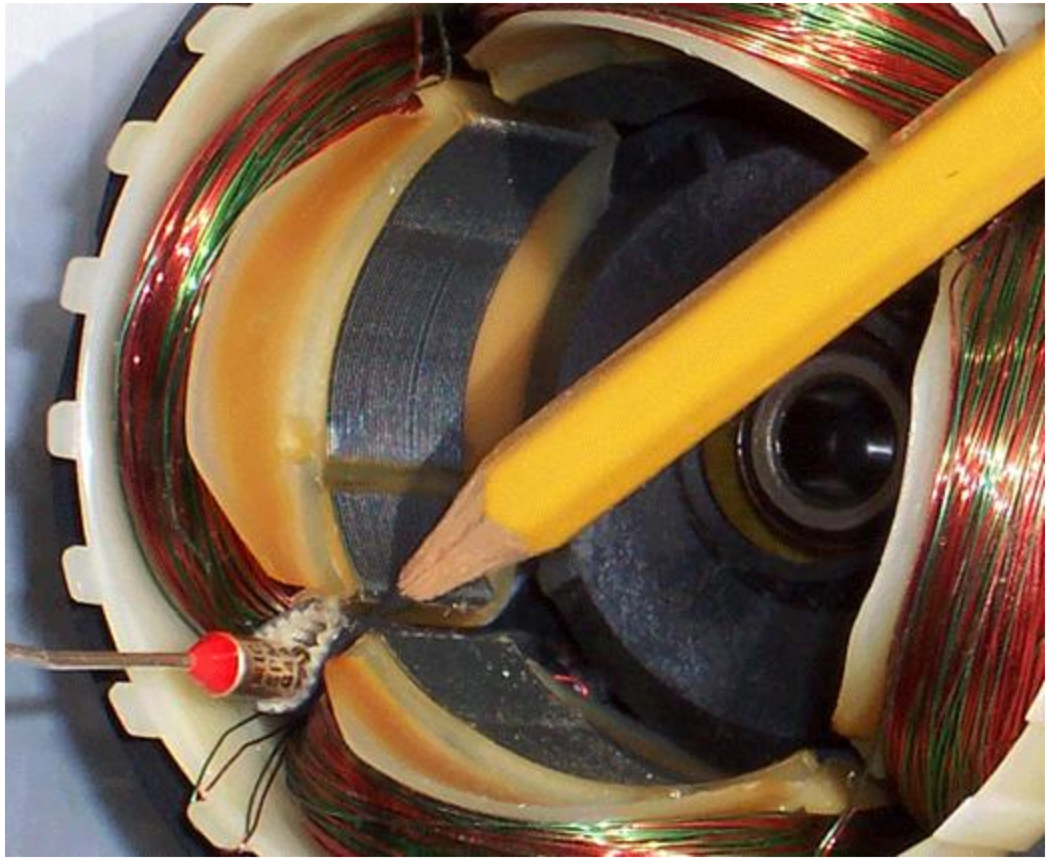
Guardrails around the ML Model

In the model

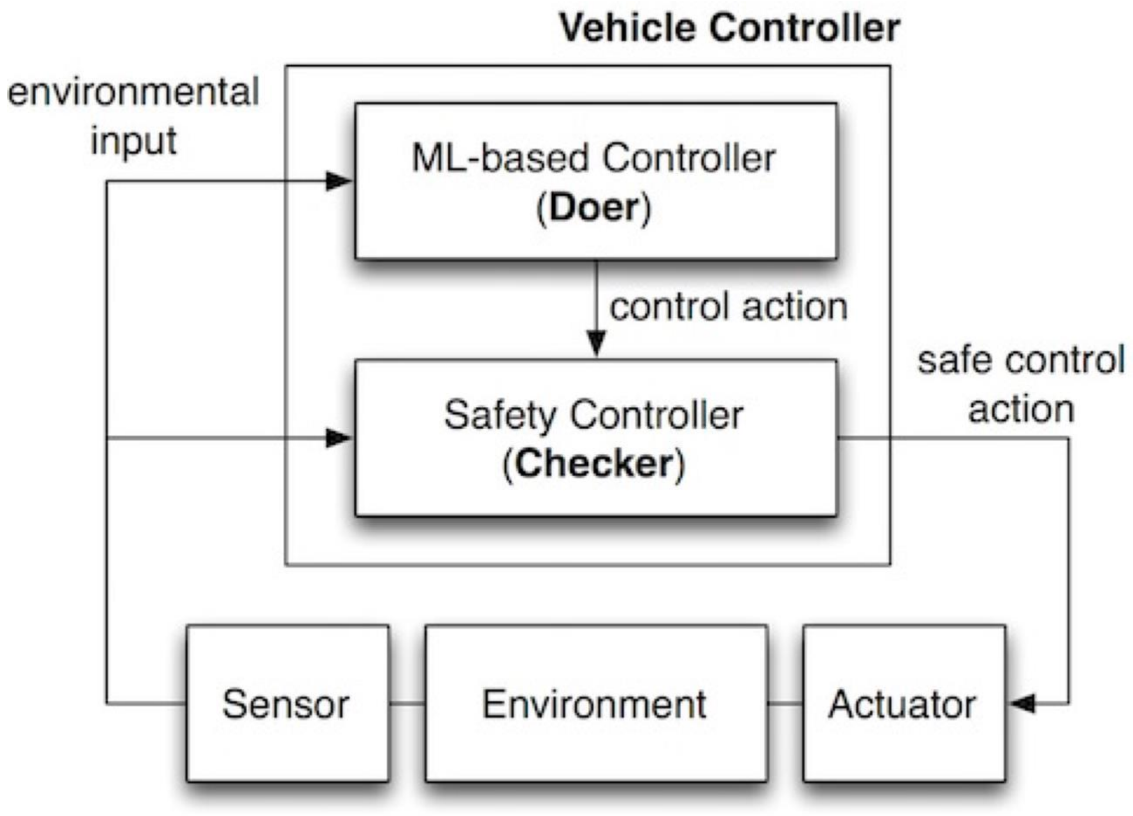
- Ensure maximum toasting time
- Use heat sensor and past outputs for prediction
- Hard to make guarantees

Outside the model (e.g., “guardrails”)

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



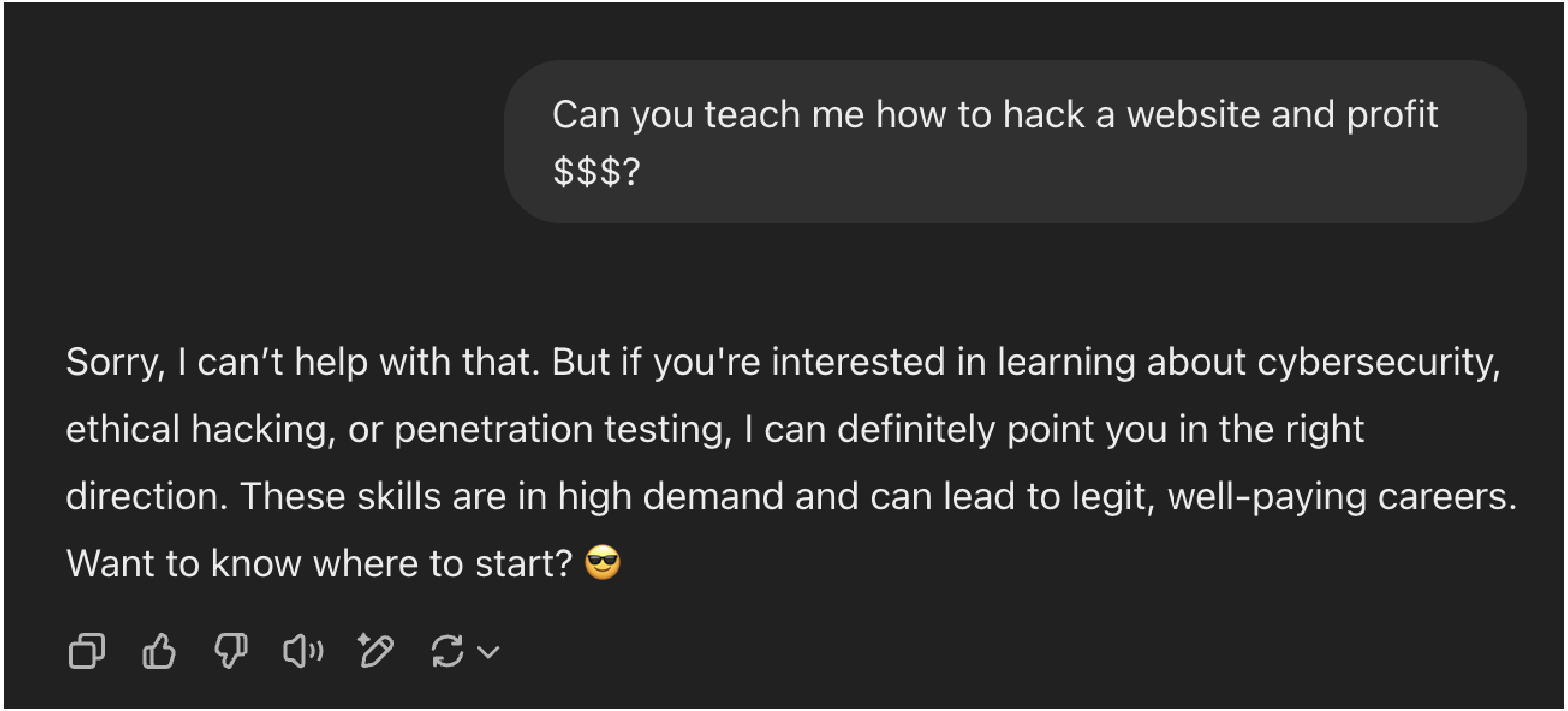
Other Examples: Doer-Checker



- ML-based controller (**Doer**): Generate commands to steer the vehicle
  - Complex ML model; highly efficient
  - But poor performance over unexpected scenarios/inputs
- Safety controller (**Checker**): Check action from ML controller
  - Overrides with a safe default action if ML action is risky
  - Simpler, based on verifiable, transparent logic; performs conservative steering control

Runtime-Safety-Guided Policy Repair. Zhou et al. (2020)

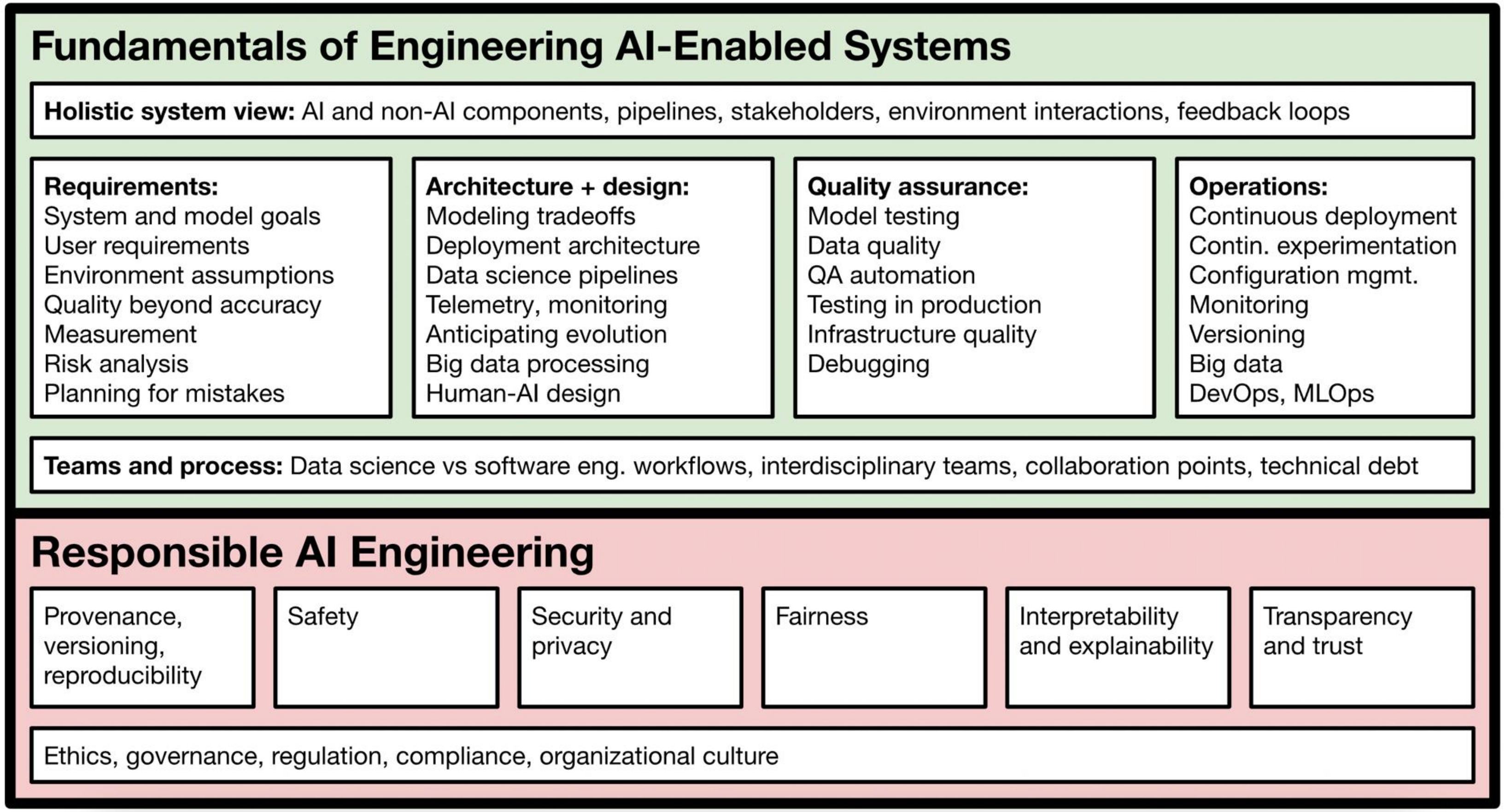
Other Examples: LLM Guardrails



Challenges for Building AI-based Systems

- Getting and updating data, concept drift, changing requirements
- Handling massive amounts of data
- Interactions with the real world, feedback loops
- Lack of modularity, lack of specifications, non-local effects
- Deployment and maintenance
- Versioning, debugging, and incremental improvement
- Keeping training and operating cost manageable
- Interdisciplinary teams
- Setting system goals, balancing stakeholders, and requirements
- …

17-445/645: ML in Production



https://mlip-cmu.github.io/

Summary

- Exit ticket!