Machine Learning in Production

From Models to Systems

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
- Define system goals and map them to goals for ML components
- Describe the typical components relating to AI in an AI-enabled system and typical design decisions to be made

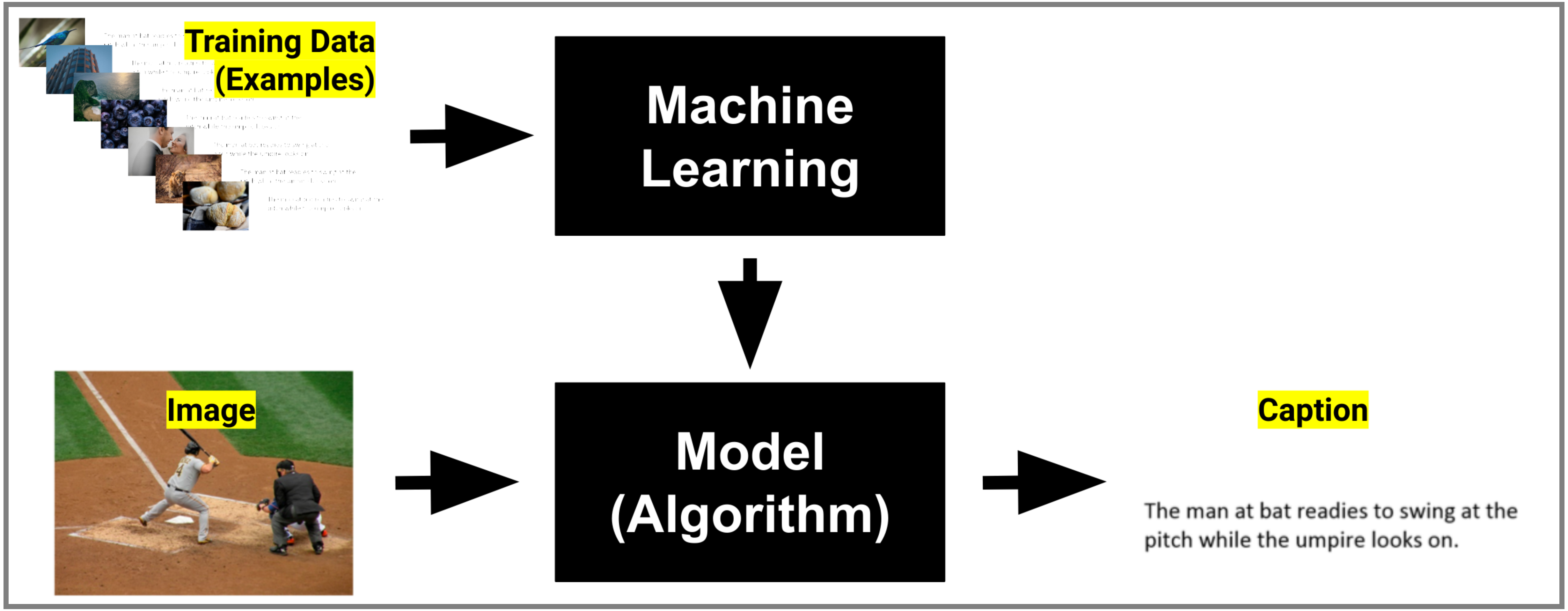
Required Readings

- Chapters 4 (Goals), 5 (Components), and 7 (Experiences) from the book "Building Intelligent Systems: A Guide to Machine Learning Engineering" by Hulten

ML Models as Part of a System

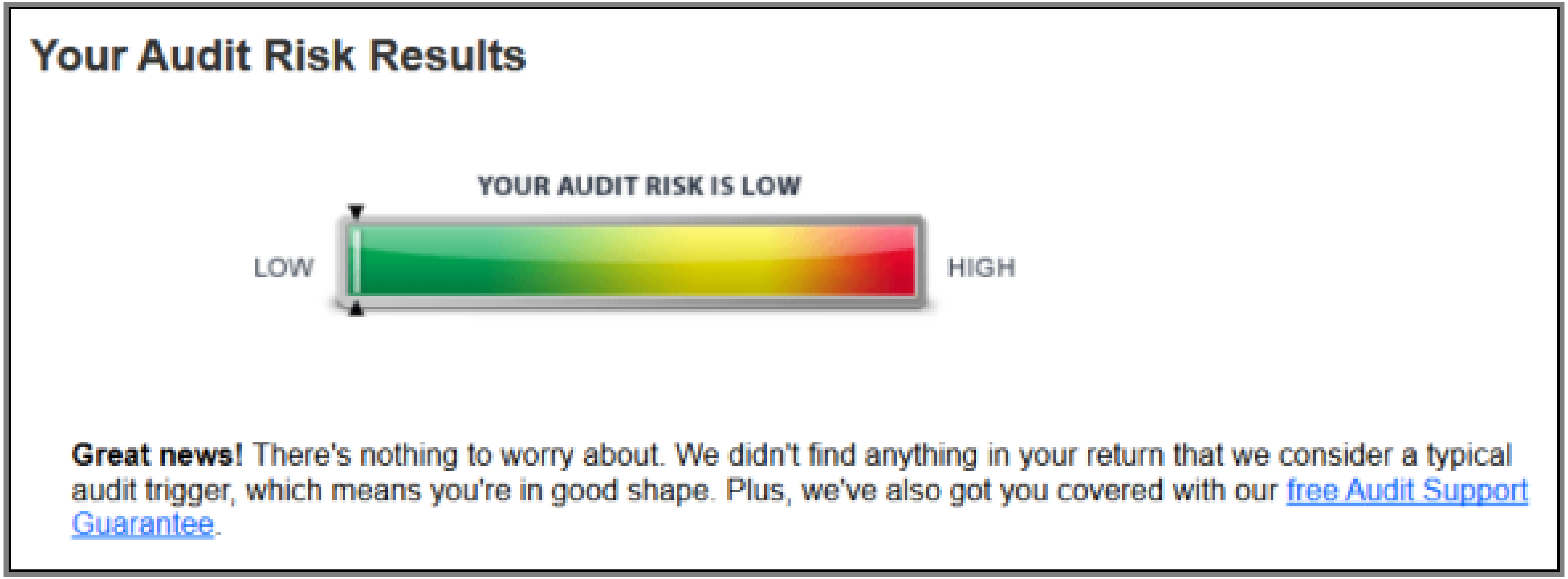
Example: Image Captioning Problem

Example: Image Captioning Problem



Why do we care about image captioning?

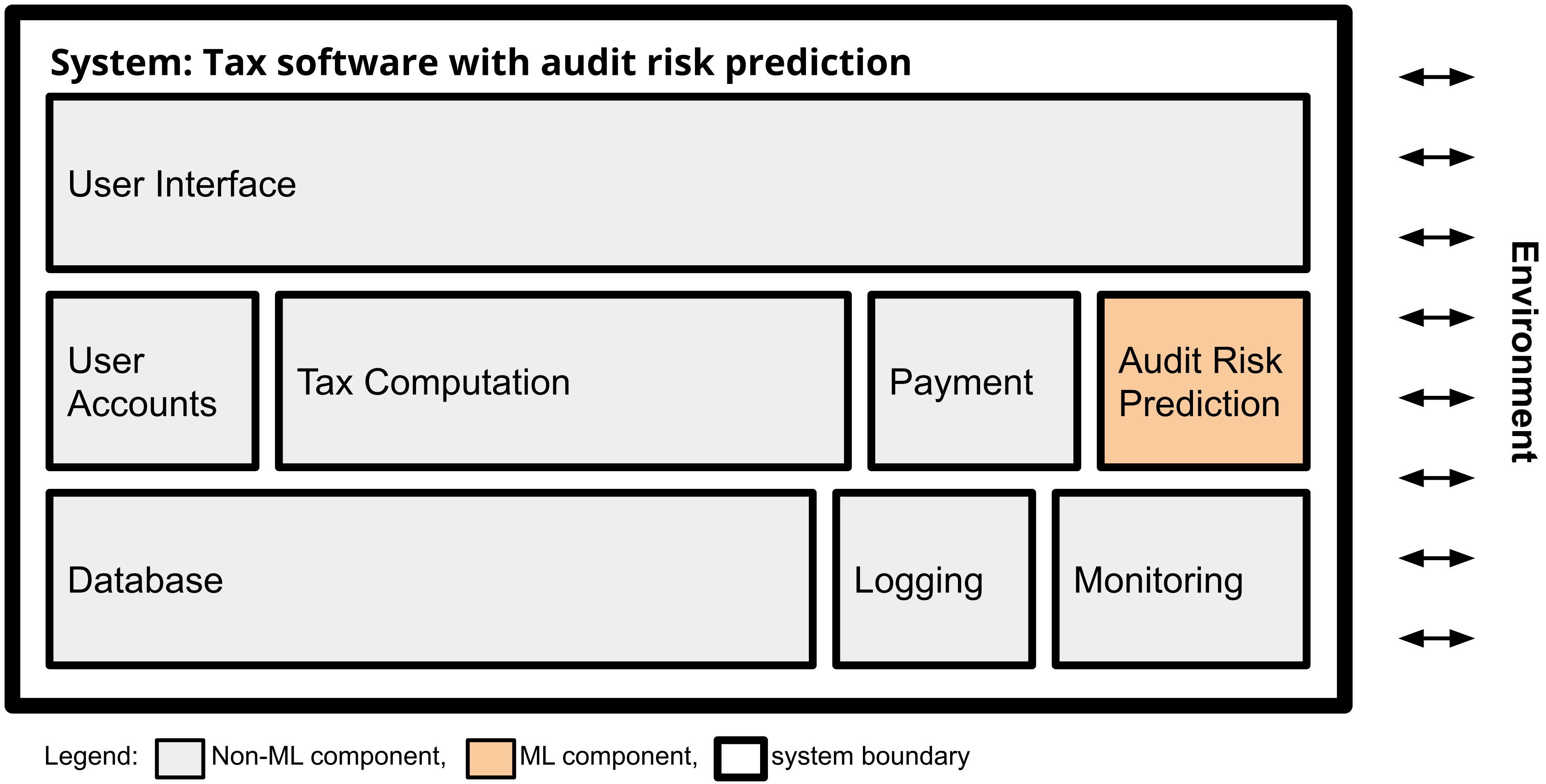
Machine learning as (small) component in a system



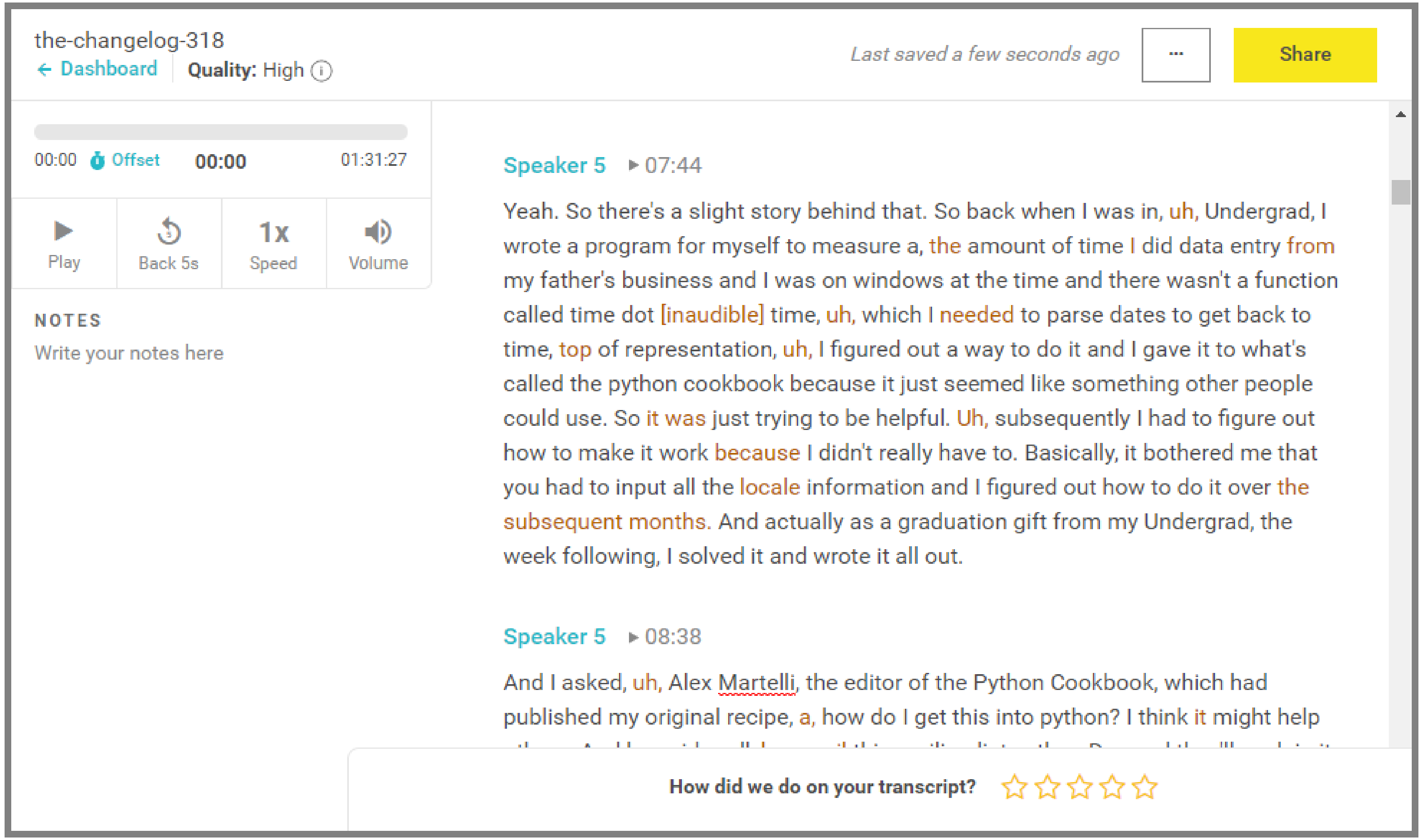
Speaker notes

Traditional non-ML tax software, with an added ML component for audit risk estimation

Machine learning as (small) component in a system



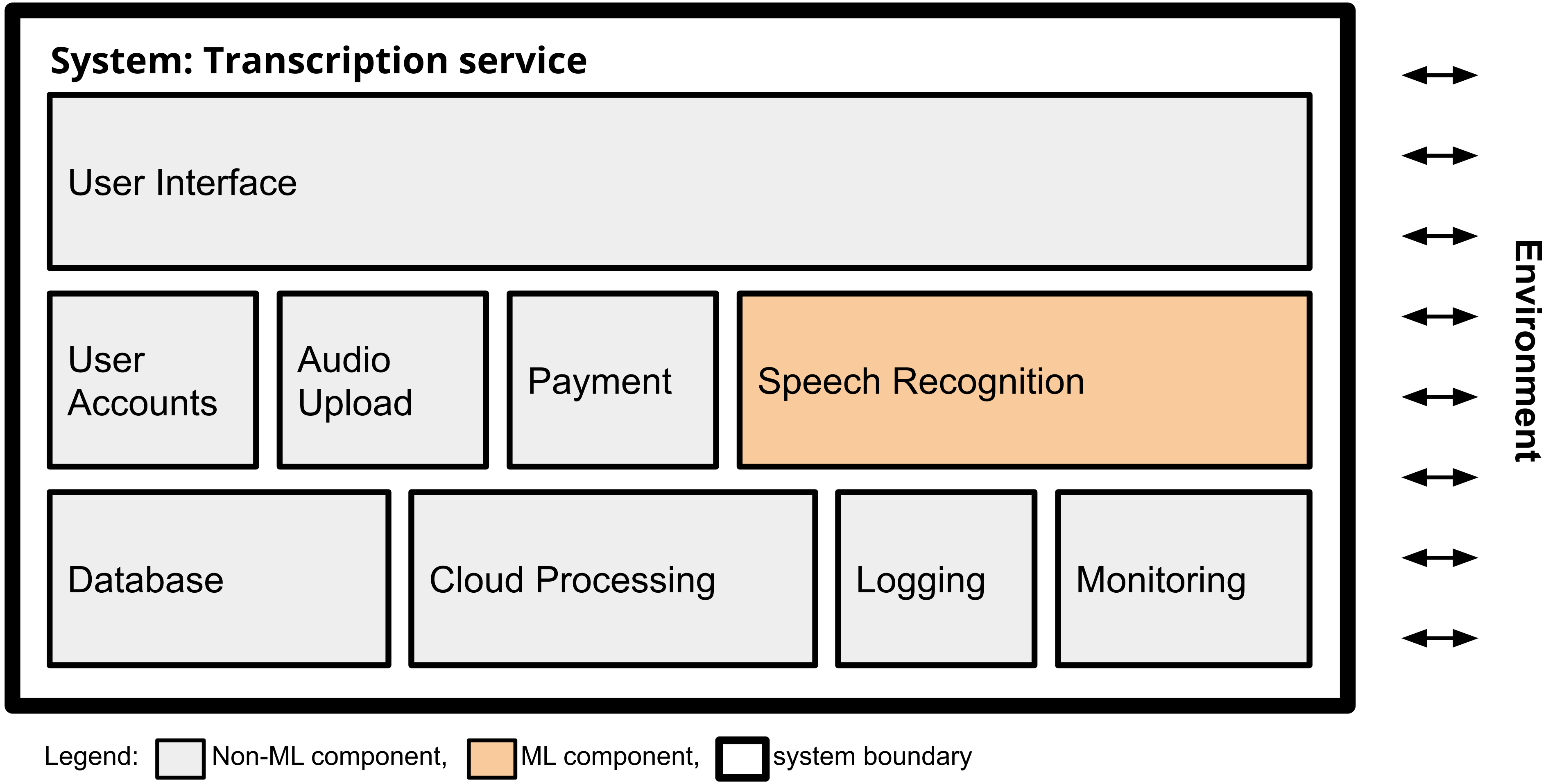
Machine learning as (core) component in a system



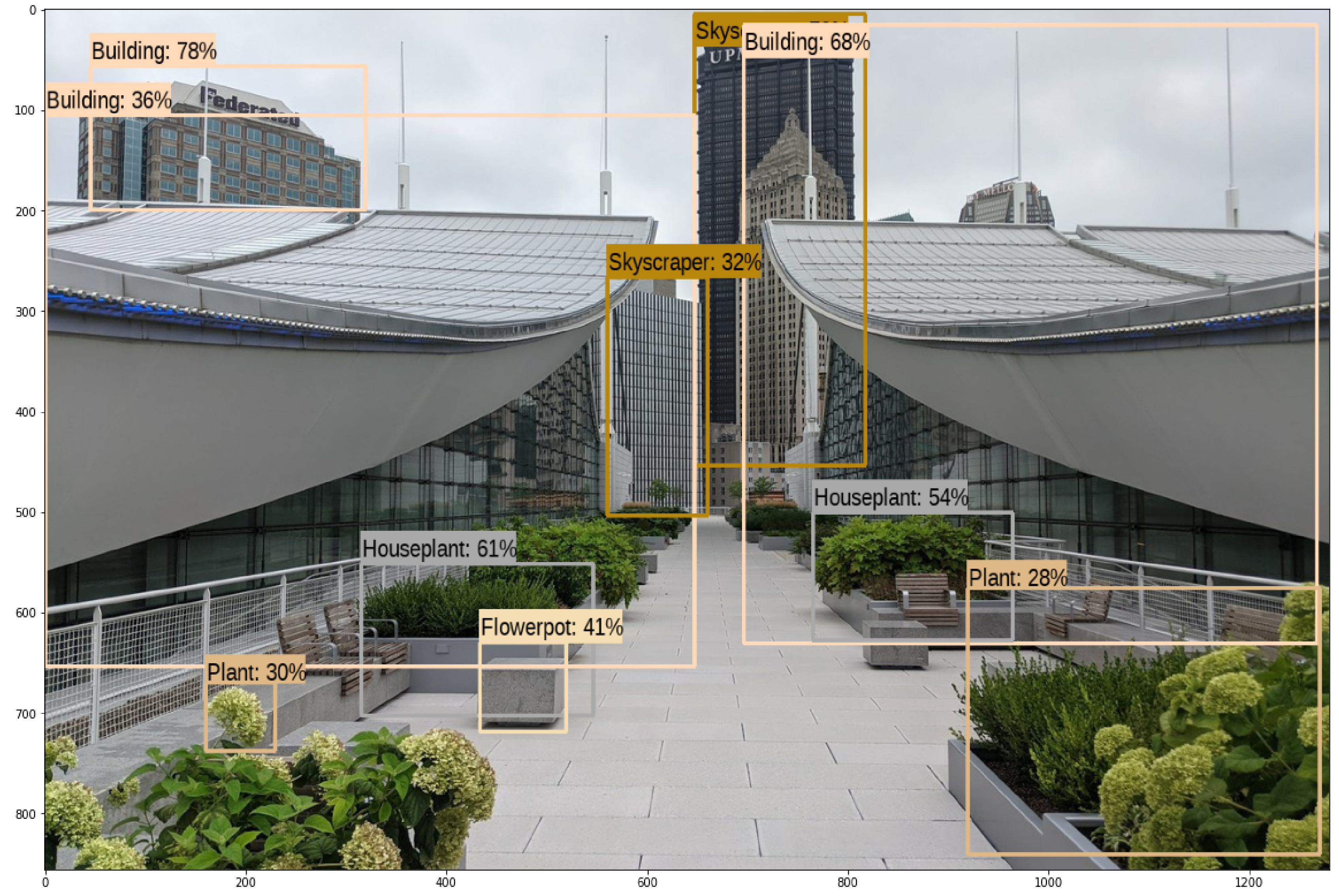
Speaker notes

Transcription service, where interface is all built around an ML component

Machine learning as (core) component in a system



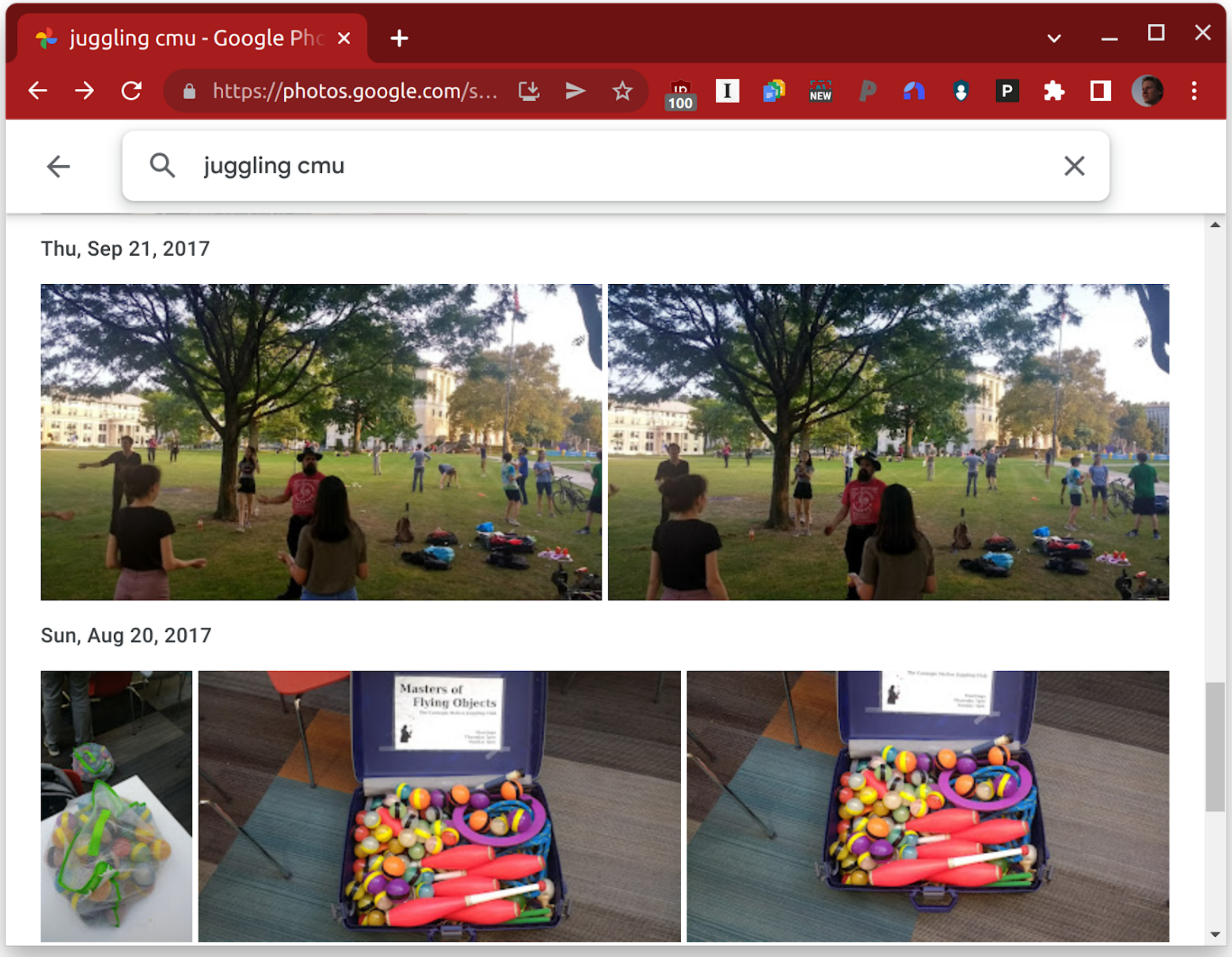
Products using Object Detection?



Products using Object Detection

What if Object Detection makes a Mistake?

Products using Object Detection

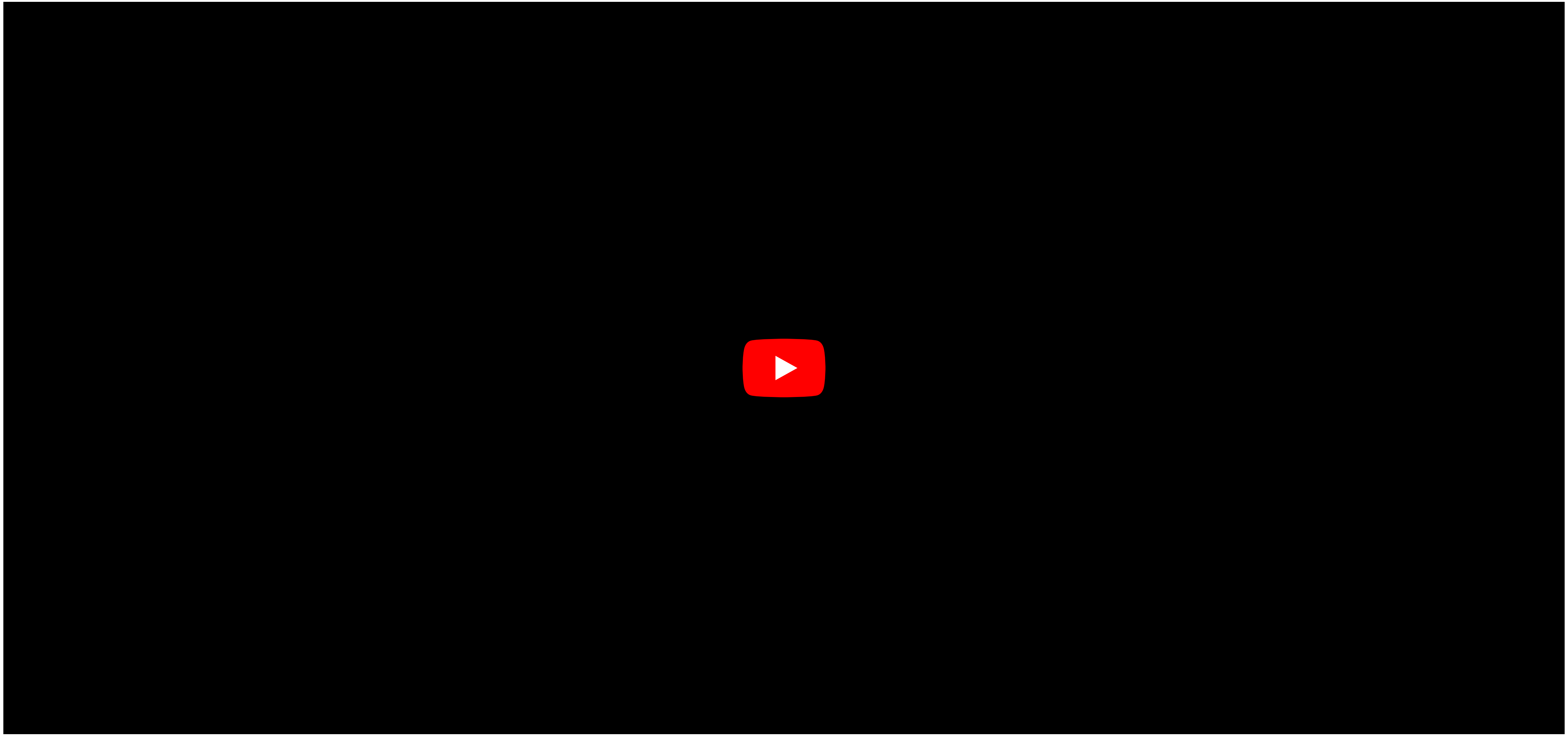


What if Object Detection makes a Mistake?

Products using Image Synthesis?

From https://openai.com/blog/dall-e/

Products using … a Juggling Robot?



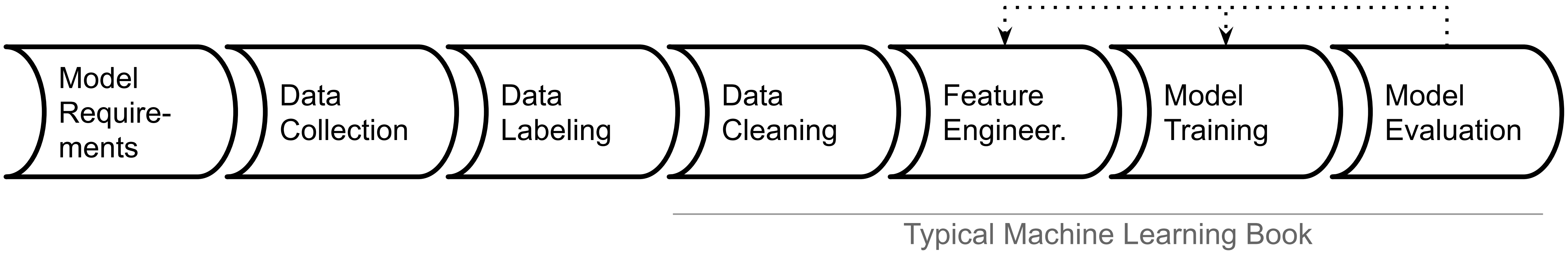
Many more examples of ML in products:

- Product recommendations on Amazon
- Surge price calculation for Uber
- Inventory planning in Walmart
- Search for new oil fields by Shell
- Adaptive cruise control in a car
- Smart app suggestion in Android
- Fashion trends prediction with social media data
- Suggesting whom to talk to in a presidential campain
- Tracking and predicting infections in a pandemic
- Adaptively reacting to network issues by a cell phone provider
- Matching players in a computer game by skill
- ...

- Some for end users, some for employees, some for expert users
- Big and small components of a larger system
- More or less non-ML code around the model

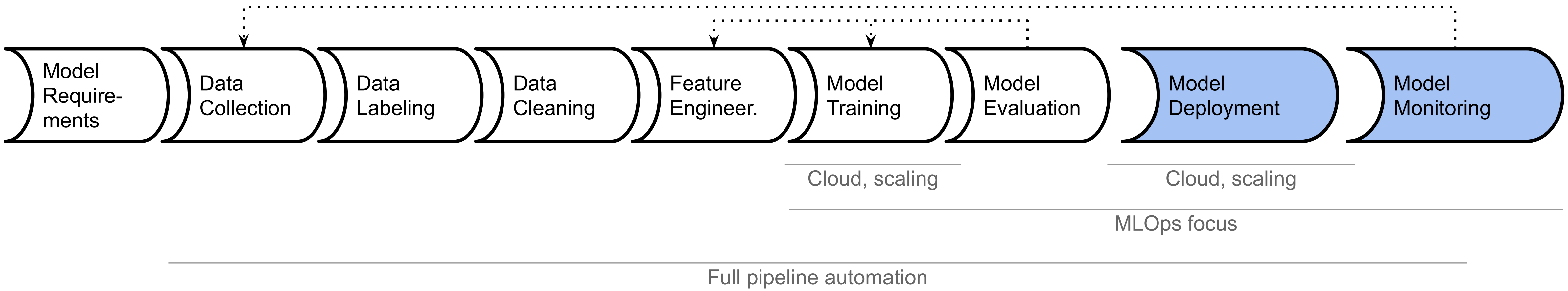
Model-Centric vs System-Wide Focus

Traditional Model Focus (Data Science)



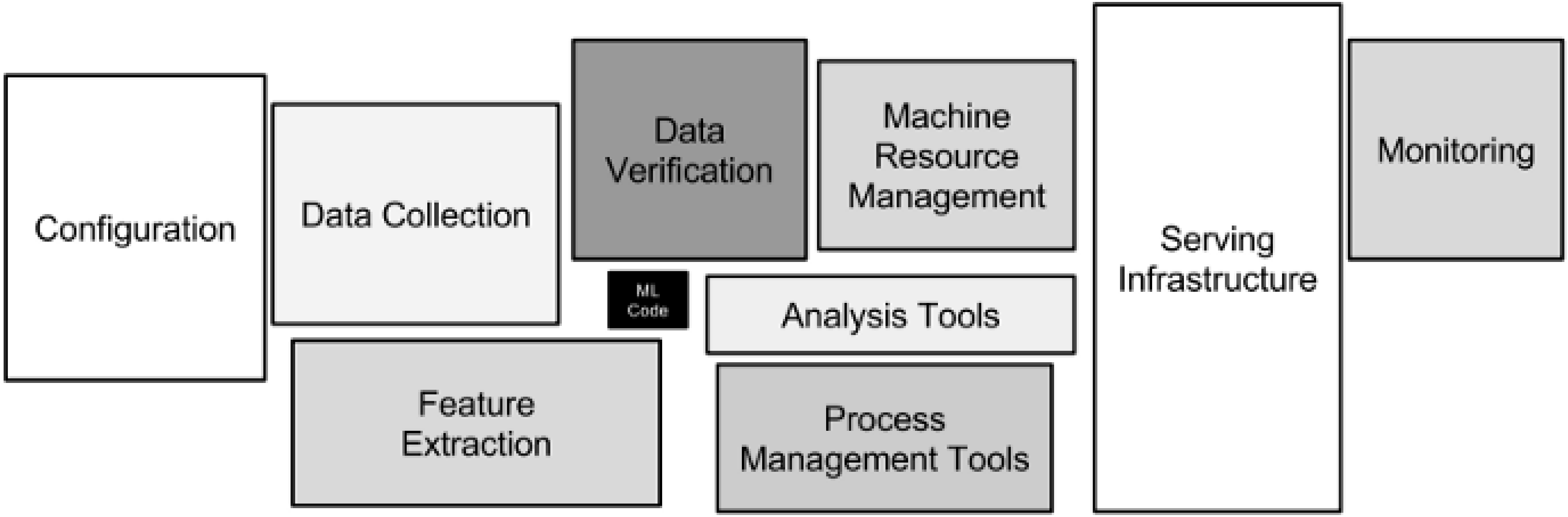
Focus: building models from given data, evaluating accuracy

Automating Pipelines and MLOps (ML Engineering)



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

MLOps Infrastructure

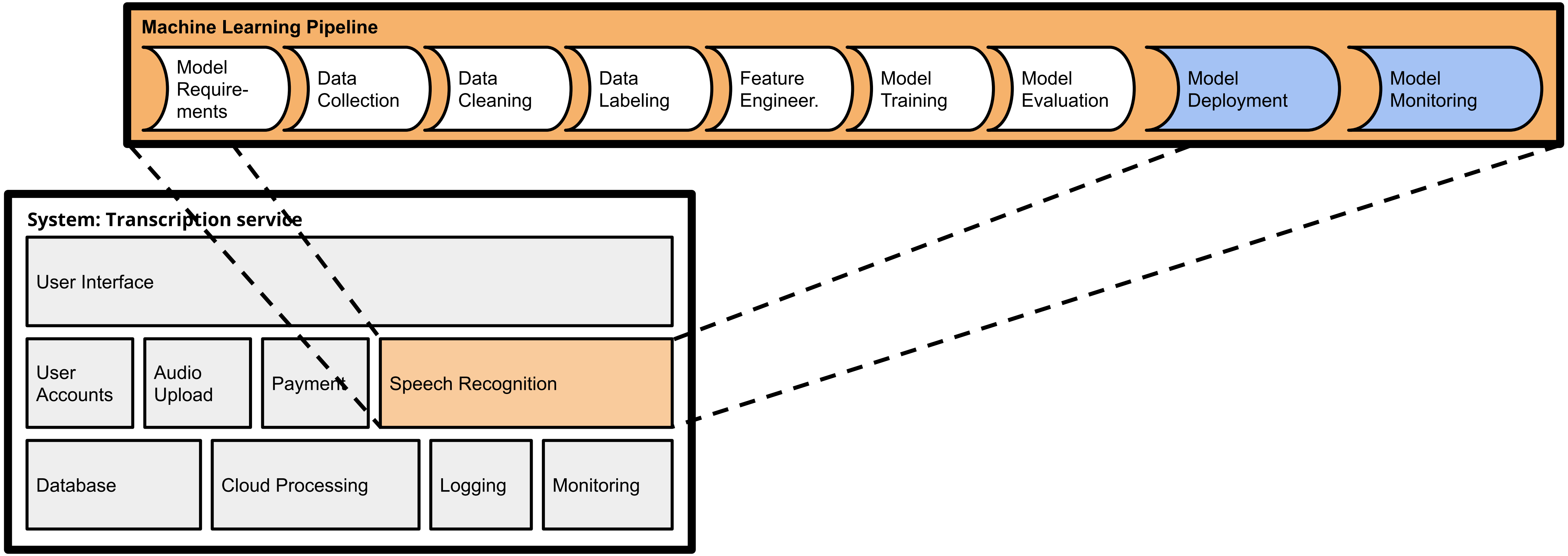


From: Sculley, David, et al. "Hidden technical debt in machine learning systems." NIPS 28 (2015).

Speaker notes

Figure from Google’s 2015 technical debt paper, indicating that the amount of code for actual model training is comparably small compared to lots of infrastructure code needed to automate model training, serving, and monitoring. These days, much of this infrastructure is readily available through competing MLOps tools (e.g., serving infrastructure, feature stores, cloud resource management, monitoring).

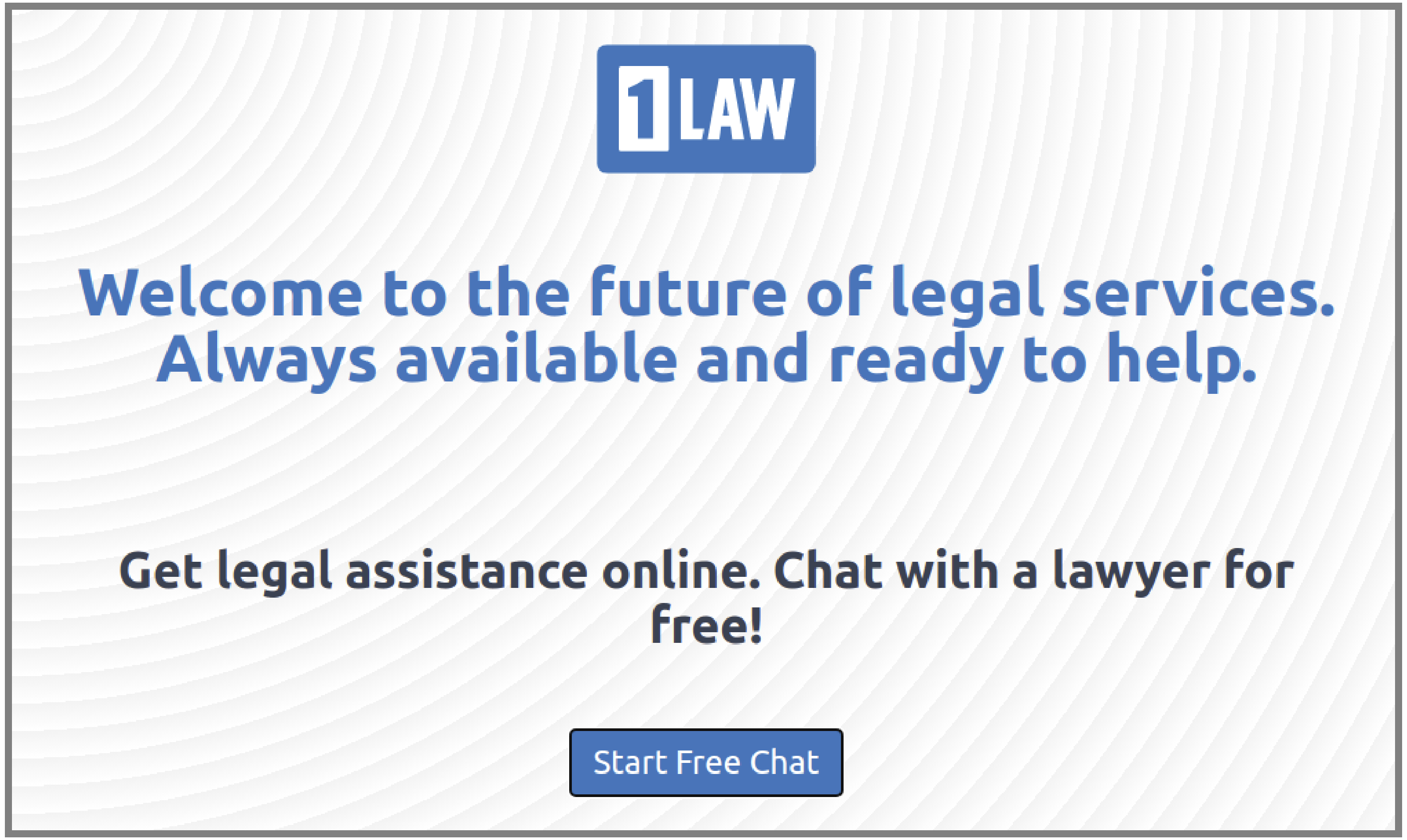
ML-Enabled Systems (ML in Production)



Interaction of ML and non-ML components, system requirements, user interactions, safety, collaboration, delivering products

Model vs System Goals

Case Study: Self-help legal chatbot

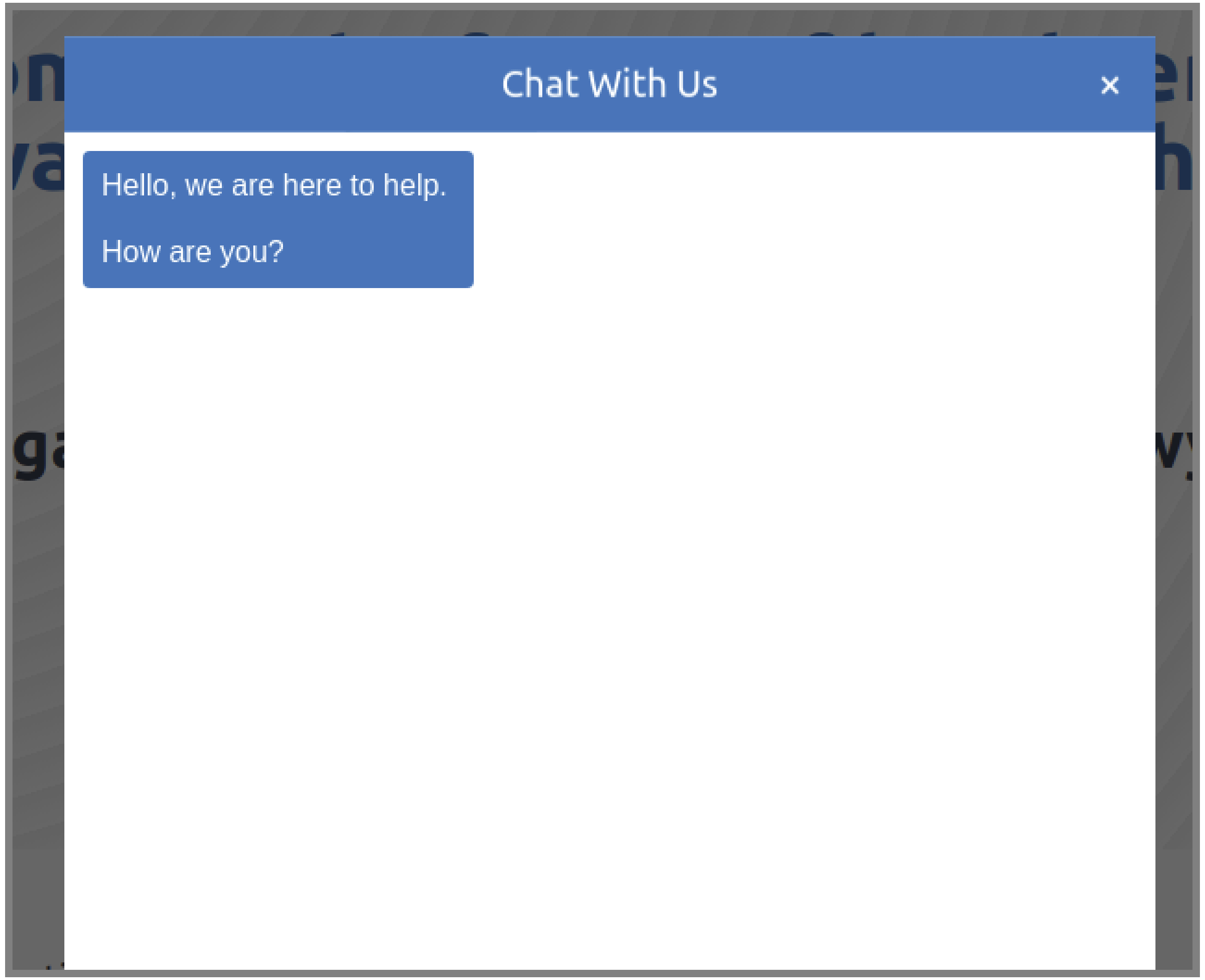


Based on the excellent paper: Passi, S., & Sengers, P. (2020). Making data science systems work. Big Data & Society, 7(2).

Speaker notes

Screenshots for illustration purposes, not the actual system studied

Case Study: Self-help legal chatbot



Previous System: Guided Chat

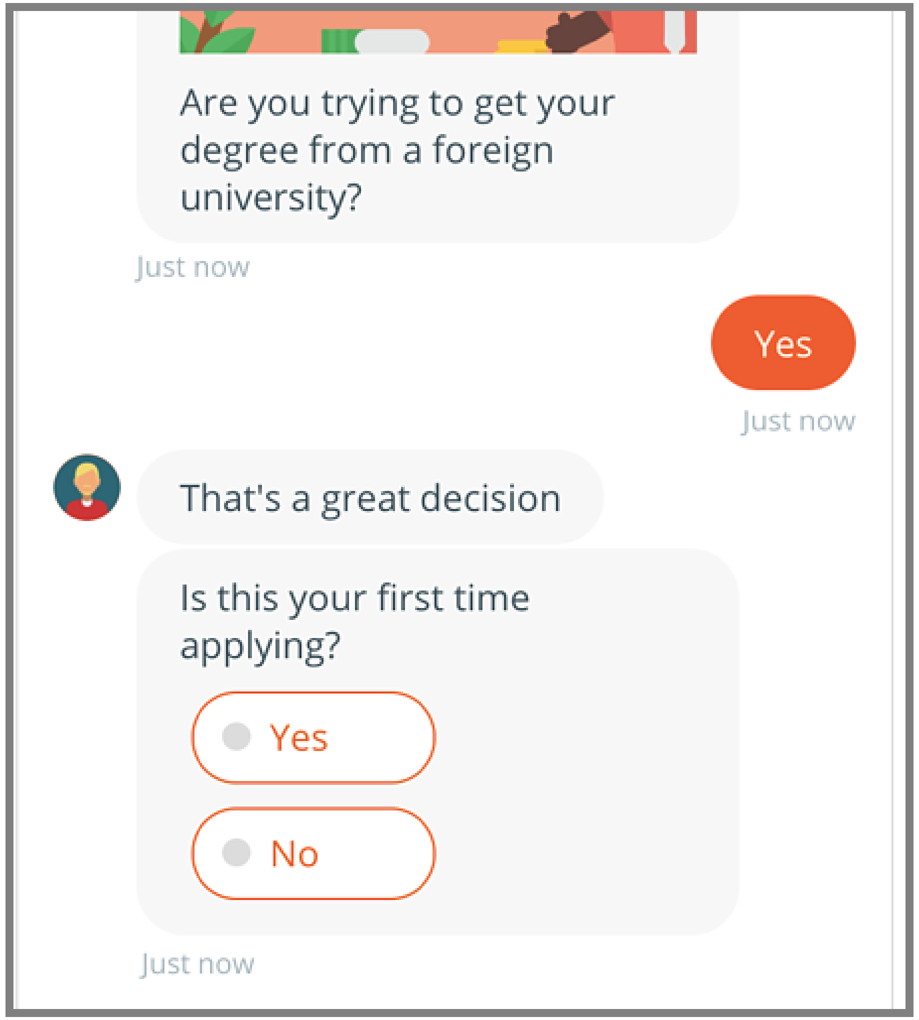


Image source: https://www.streamcreative.com/chatbot-scripts-examples-templates

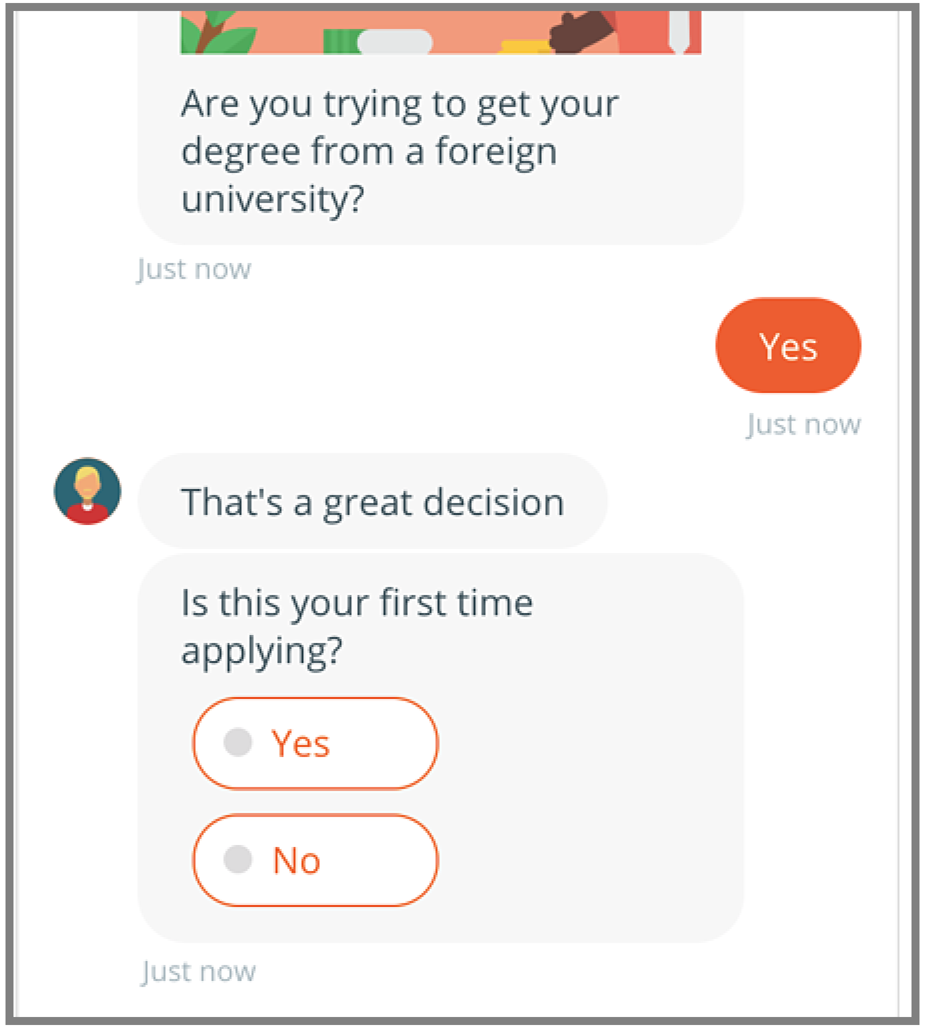
Problems with Guided Chats

Non-AI guided chat was too limited

- Cannot enumerate problems
- Hard to match against open entries ("I want to file for bankruptcy" vs "I have no money")

Involving human operators very expensive

Old-fashioned



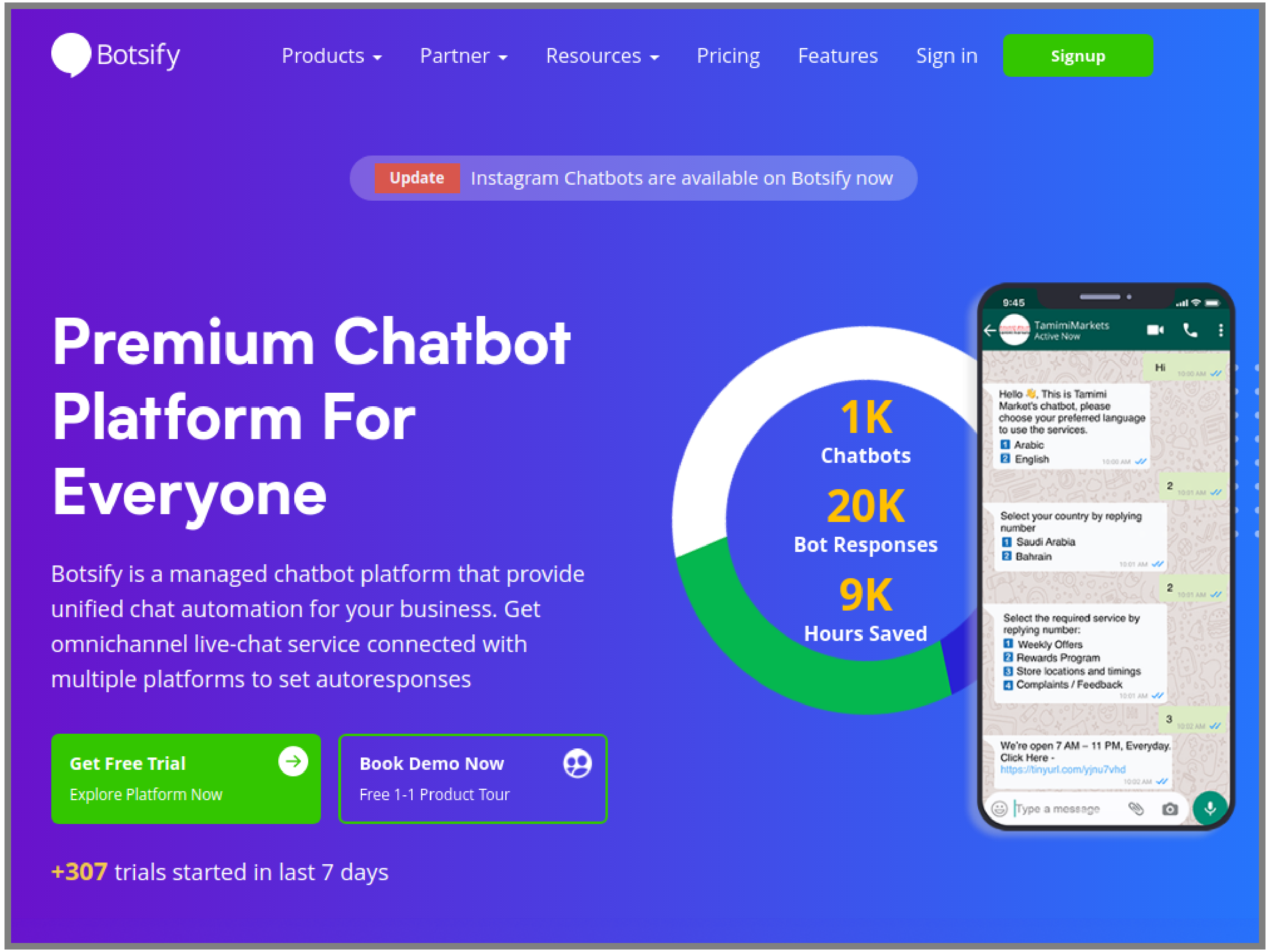
Initial Goal: Better Chatbot

Help users with simple task

Connect them with lawyers when needed

Modernize appearence; "future of digital marketing"

Buy or Build?


Botsify
Products
Partner
Resources
Pricing
Features
Sign in
Signup
Update Instagram Chatbots are available on Botsify now
Premium Chatbot Platform For Everyone
Botsify is a managed chatbot platform that provide unified chat automation for your business. Get omnichannel live-chat service connected with multiple platforms to set autoresponses
Get Free Trial
Explore Platform Now
Book Demo Now
Free 1-1 Product Tour
+307 trials started in last 7 days
1K
Chatbots
20K
Bot Responses
9K
Hours Saved

Speaker notes

One of many commercial frameworks for building AI chatbots

Data scientists' challenges

Infrastructure: Understand chat bot infrastructure and its capabilities

Knowing topics: Identify what users talk about, train/test concepts with past chat logs

- "*We fed VocabX a line deliberately trying to confuse it. We wrote, 'I am thinking about chapter 13 in Boston divorce filing.' VocabX figured out the two topics: (1) business and industrial/company/bankruptcy (2) society/social institution/divorce.*"

Guiding conversations: Supporting open-ended conversations requires detecting what's on topic and finding a good response; intent-topic modeling

- *Is talk about parents and children on topic when discussing divorce?*
- Data gathering/labeling very challenging -- too many corner cases

Stepping Back: What are the goals of the system?

Status meeting with (inhouse) Customer

The chatbot performed better than before but was far from ready for deployment. There were "too many edge cases" in which conversations did not go as planned.

> ***Customer:*** *"Maybe we need to think about it like an 80/20 rule. In some cases, it works well, but for some, it is harder. 80% everything is fine, and in the remaining 20%, we try to do our best."*
> ***Data science lead:*** *The trouble is how to automatically recognize what is 80 and what is 20.*
> ***Data scientist:*** *It is harder than it sounds. One of the models is a matching model trained on pairs of legal questions and answers. 60,000 of them. It seems large but is small for ML.*
> ***Customer:*** *That's a lot. Can it answer a question about say visa renewal?*
> ***Data scientist:*** *If there exists a question like that in training data, then yes. But with just 60,000, the model can easily overfit, and then for anything outside, it would just fail.*
> ***Customer:*** *I see what you are saying. Edge cases are interesting from an academic perspective, but for a business the first and foremost thing is value. You are trying to solve an interesting problem. I get it. But I feel that you may have already solved it enough to gain business value.*

Speaker notes

Adapted from Passi, S., & Sengers, P. (2020). Making data science systems work. Big Data & Society, 7(2).

System Goal for Chatbot

- Collect user data to sell to lawyers
- Signal technical competency to lawyers
- Acceptable to fail: Too complicated for self-help, connect with lawyer
- Solving edge cases not important

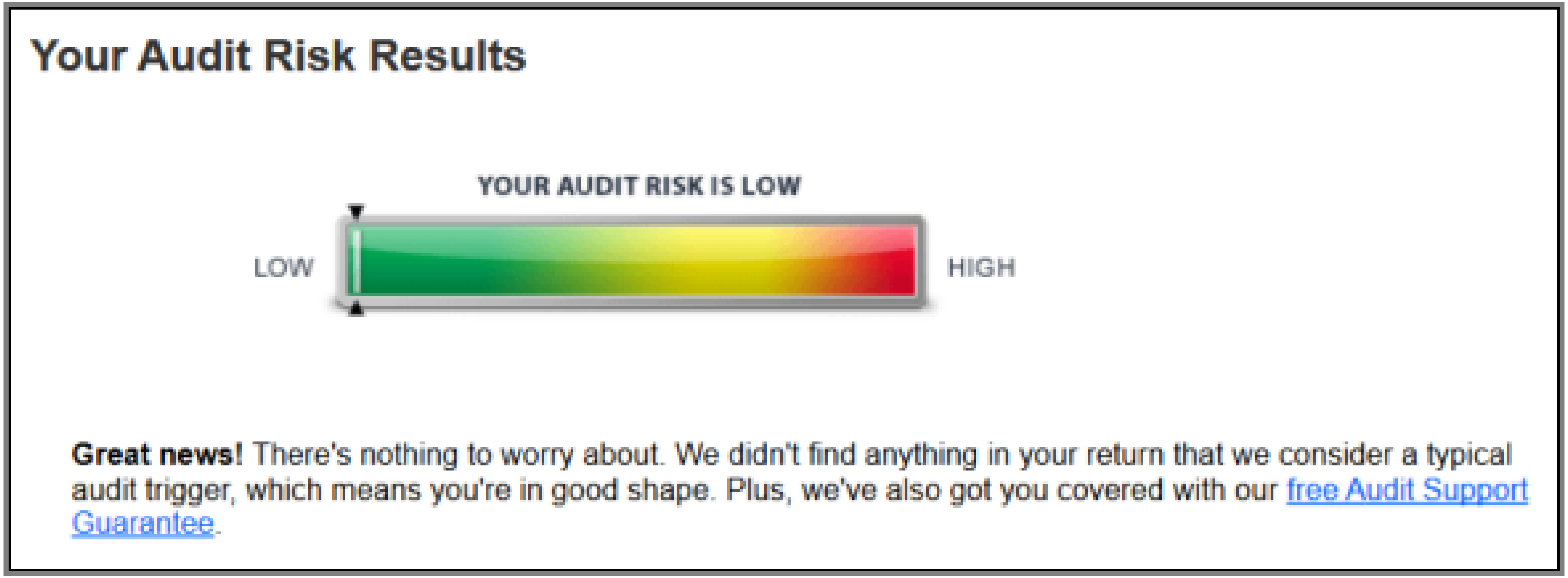
> *"Edge cases are important, but the end goal is user information, monetizing user data. We are building a legal self-help chatbot, but a major business use case is to tell people: 'here, talk to this lawyer.' We do want to connect them with a lawyer. Even for 20%, when our bot fails, we tell users that the problem cannot be done through self-help. Let us get you a lawyer, right? That is what we wanted in the first place."*

Speaker notes

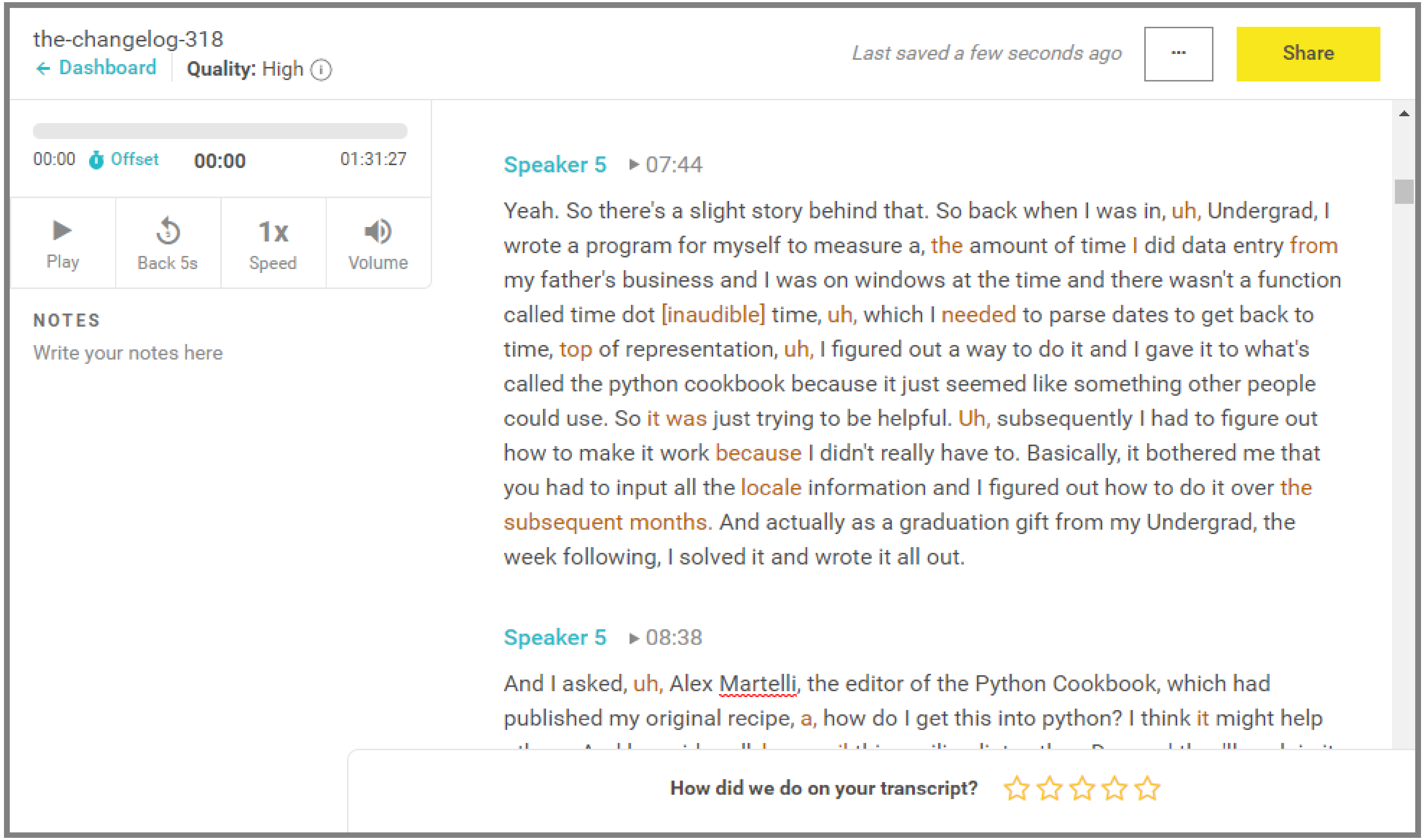
See Passi, S., & Sengers, P. (2020). Making data science systems work. Big Data & Society, 7(2).

Model vs System Goal?

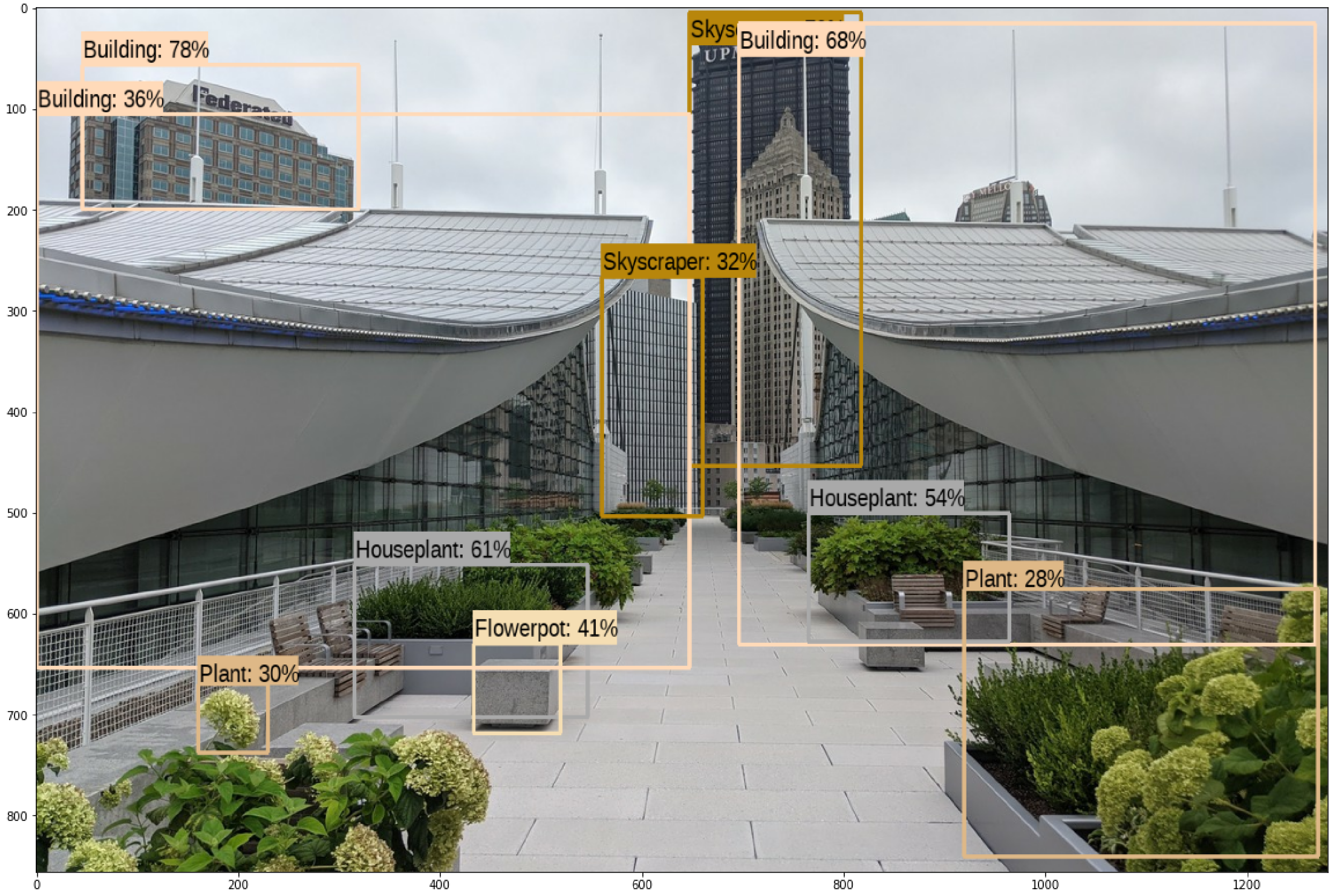
Model vs System Goal?



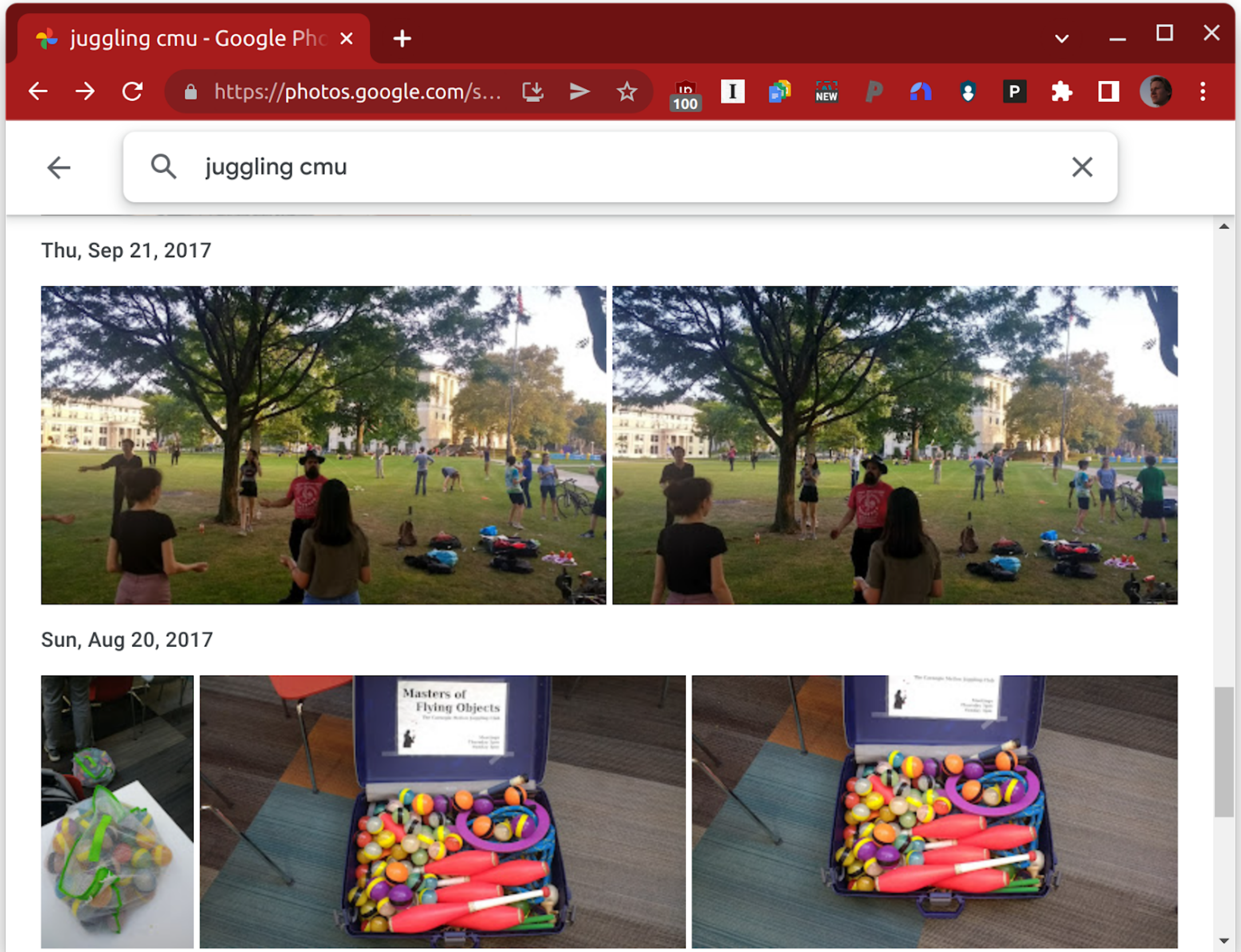
Model vs System Goal?



Model vs System Goal?



Model vs System Goal?



Model vs System Goal?

Model vs System Goal?

More Accurate Predictions may not be THAT Important

- "Good enough" may be good enough
- Prediction critical for system success or just an gimmick?
- Better predictions may come at excessive costs
  - need way more data, much longer training times
  - privacy concerns
- Better user interface ("experience") may mitigate many problems
  - e.g. explain decisions to users
- Use only high-confidence predictions?

Machine learning that matters

- 2012(!) essay lamenting focus on algorithmic improvements and benchmarks
  - focus on standard benchmark sets, not engaging with problem: Iris classification, digit recognition, ...
  - focus on abstract metrics, not measuring real-world impact: accuracy, ROC
  - distant from real-world concerns
  - lack of follow-through, no deployment, no impact
- Failure to *reproduce* and *productionize* paper contributions common
- Ignoring design choices in how to collect data, what problem to solve, how to design human-AI interface, measuring impact, ...
- *Argues: Should focus on making impact -- requires building systems*

Wagstaff, Kiri. "Machine learning that matters." In Proceedings of the 29 th International Conference on Machine Learning, (2012).

On Terminology

- There is no standard term for referring to building systems with AI components
- **ML-Enabled Systems**, *Production ML Systems*, *AI-Enabled Systems*, or *ML-Infused Systems*; *SE4AI*, *SE4ML*
- sometimes **AI Engineering** / **ML Engineering** -- but usually used with a ML-pipeline focus
- **MLOps** ~ technical infrastructure automating ML pipelines
- sometimes **ML Systems Engineering** -- but often this refers to building distributed and scalable ML and data storage platforms
- "AIOps" ~ using AI to make automated decisions in operations; "DataOps" ~ use of agile methods and automation in business data analytics
- My preference: **Production Systems with Machine-Learning Components**

Setting and Untangling Goals

Step 1 of Requirements...

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

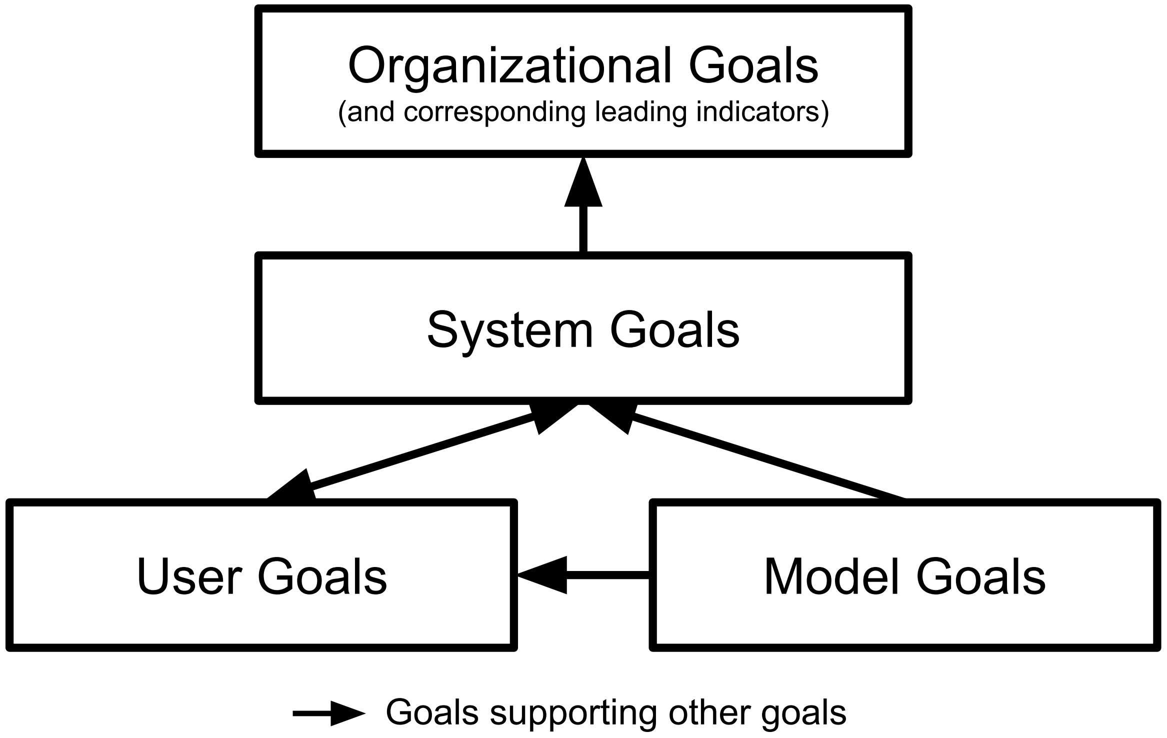
Ethics, governance, regulation, compliance, organizational culture

Layers of Success Measures

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

- **Leading indicators:** Short-term proxies for long-term measures, typically for organizational objectives

Ideally, these goals should be aligned with each other



Organizational Goals

Innate/overall goals of the organization

- Business
  - Current/future revenue, profit
  - Reduce business risks
- Non-Profits
  - Lives saved, animal welfare increased, CO2 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, ratings)
- 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/Feature Goals

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

Relates to system requirements

Examples:

- Detect cancer in radiology scans
- Provide and recommend music to stream
- Make personalized music recommendations
- Transcribe audio files
- Provide legal help with a self-service chatbot

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

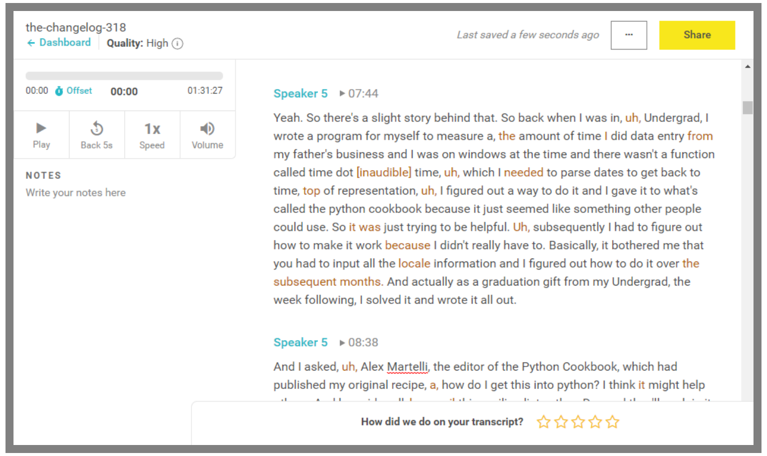
Model Goals

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

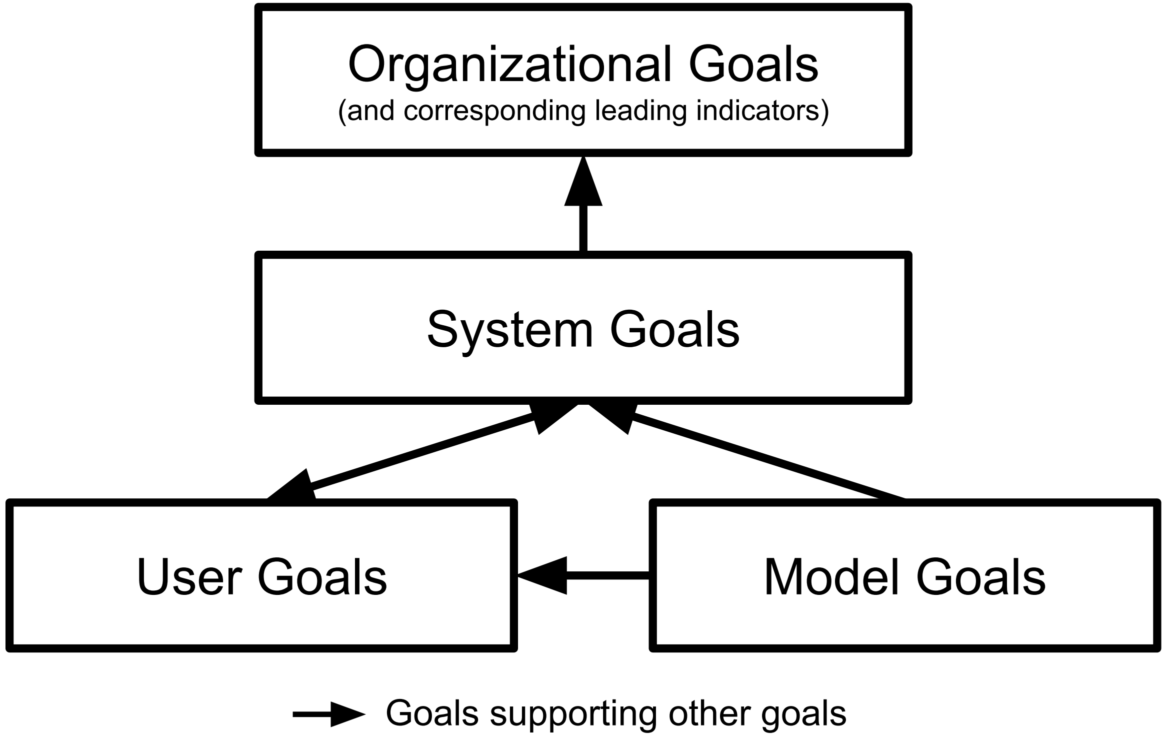
- Model accuracy
- Rate and kinds of mistakes
- Successful user interactions
- Inference time
- Training cost

Often not directly linked to organizational/system/user goals

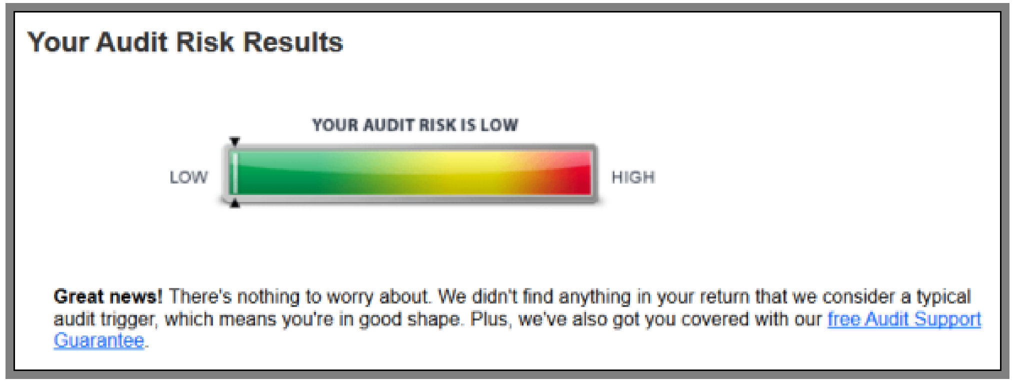
Success Measures in the Transcription Scenario?

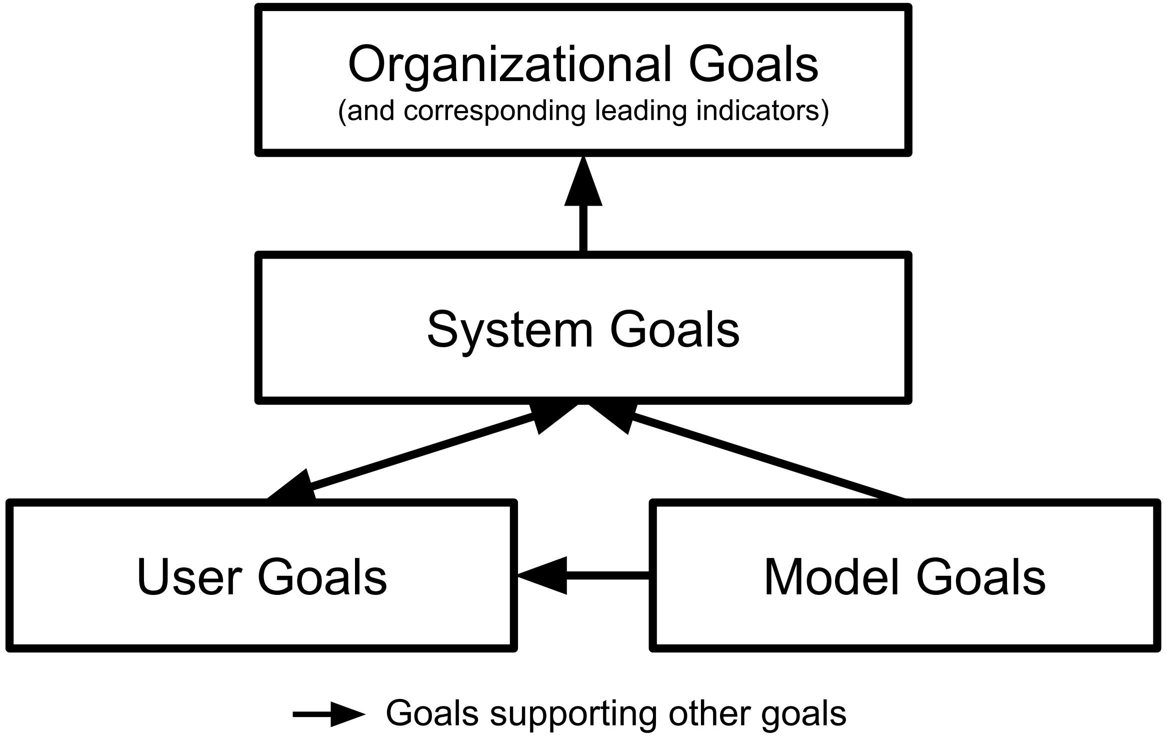


Organizational goals? Leading indicators? System goals? User goals? Model goals?



Success Measures in the Audit Risk Scenario?





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

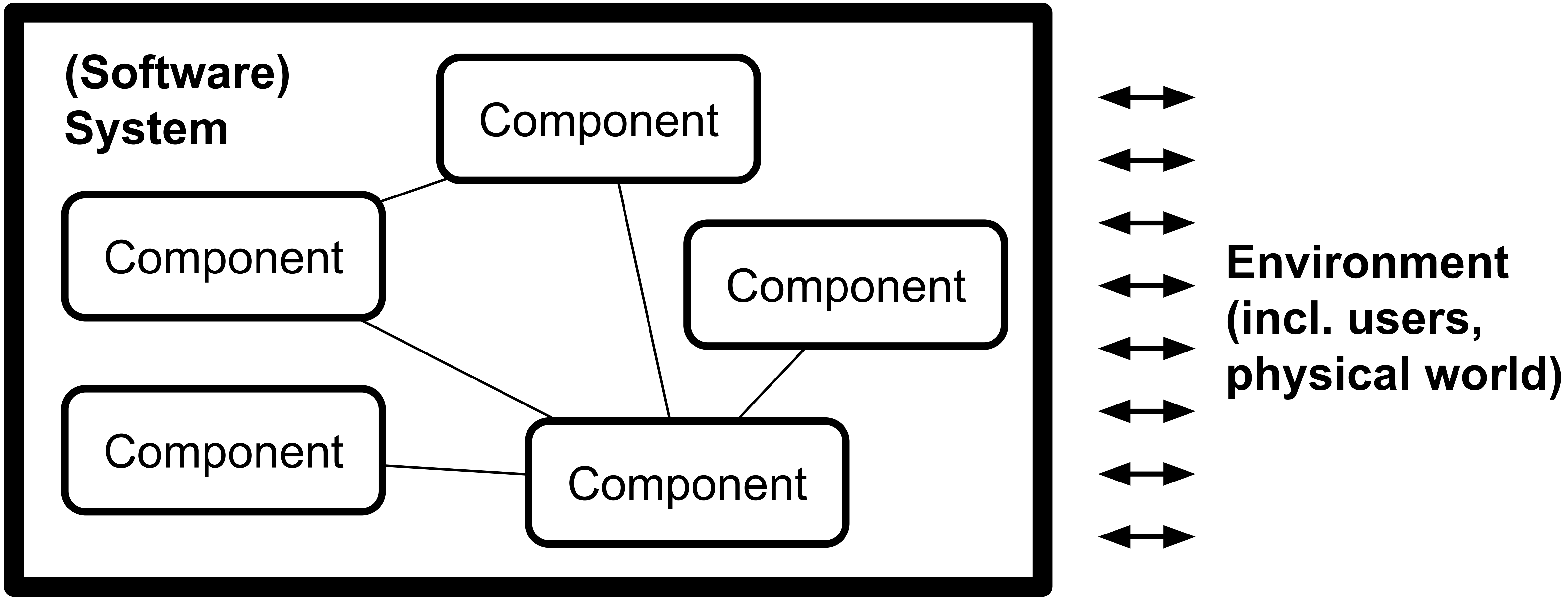
Breakout: Automating Admission Decisions

What are different types of goals behind automating admissions decisions to a Master's program?

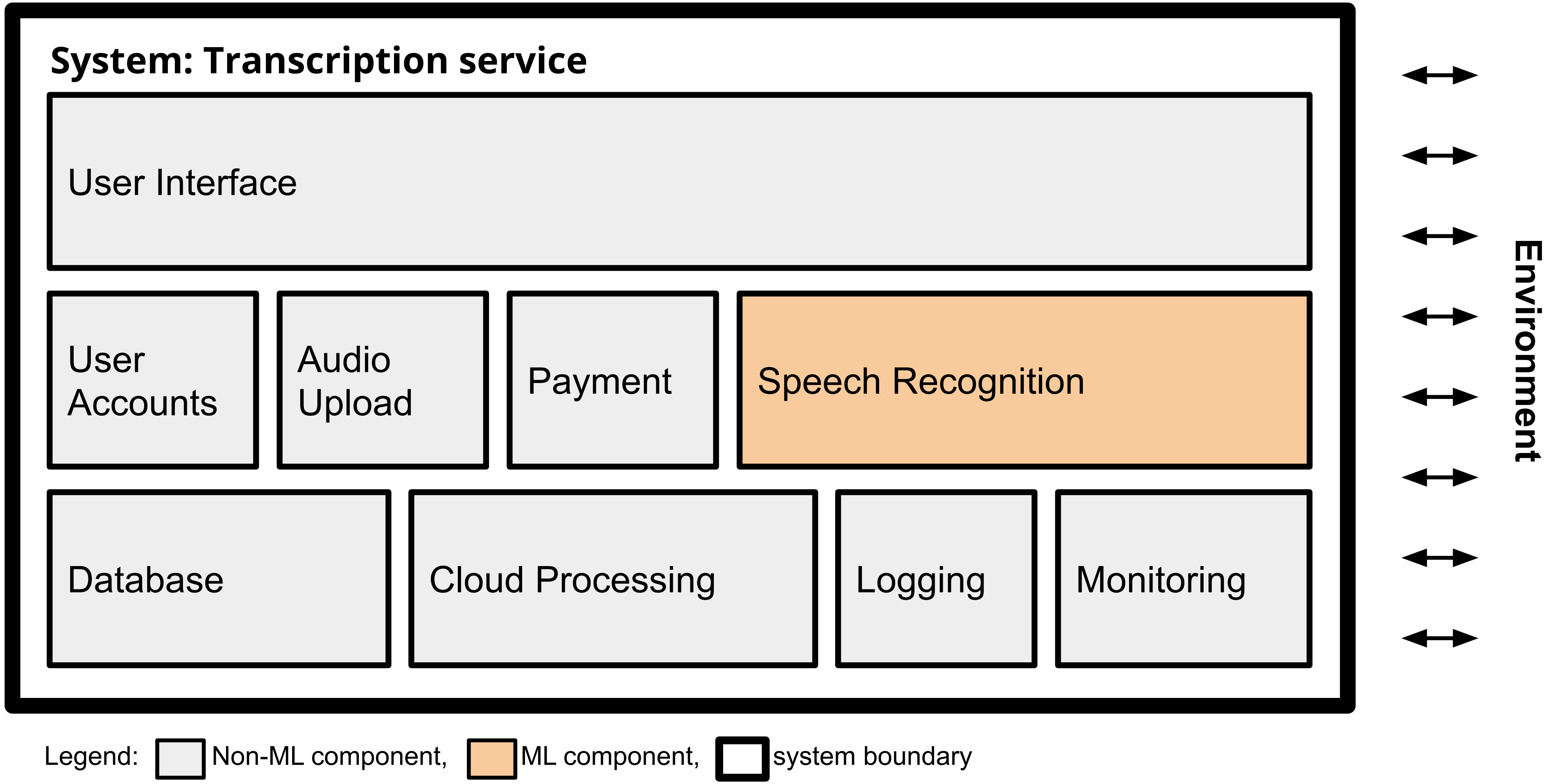
As a group post answer to `#lecture` tagging all group members using template:

> *Organizational goals: ...*
> *Leading indicators: ...*
> *System goals: ...*
> *User goals: ...*
> *Model goals: ...*

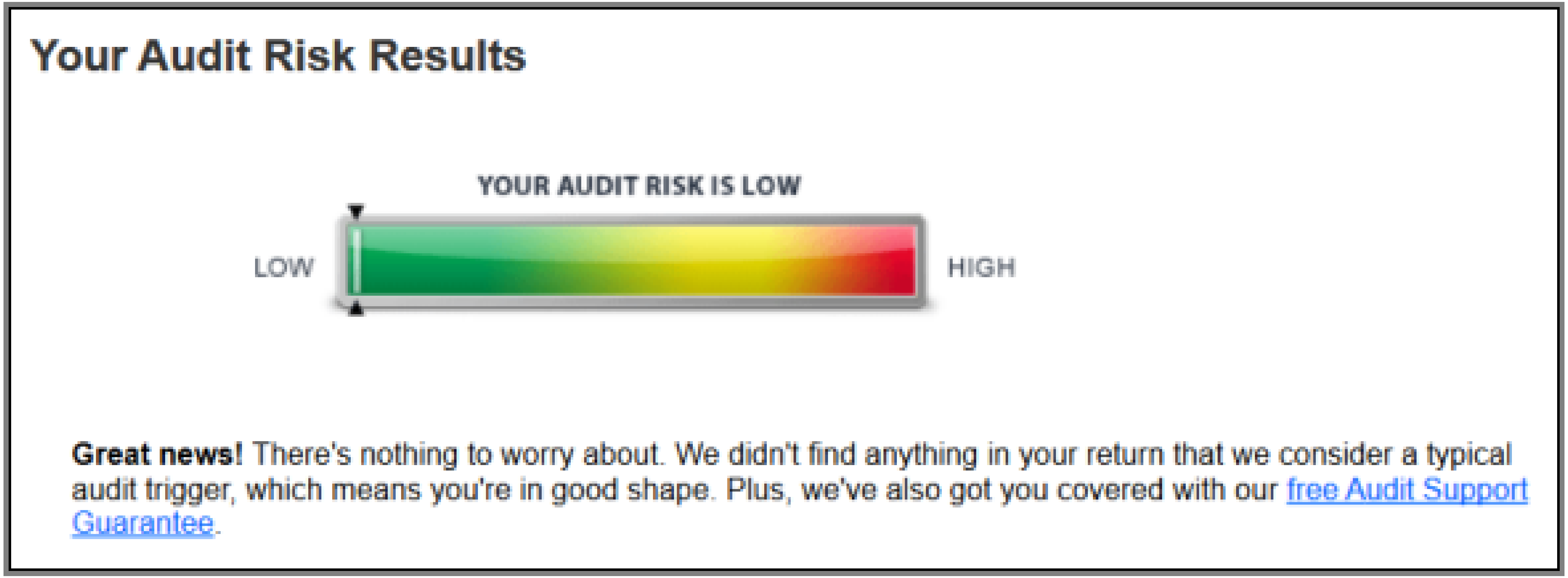
Systems Thinking



Repeat: Machine learning as component in a system



The System Interacts with Users

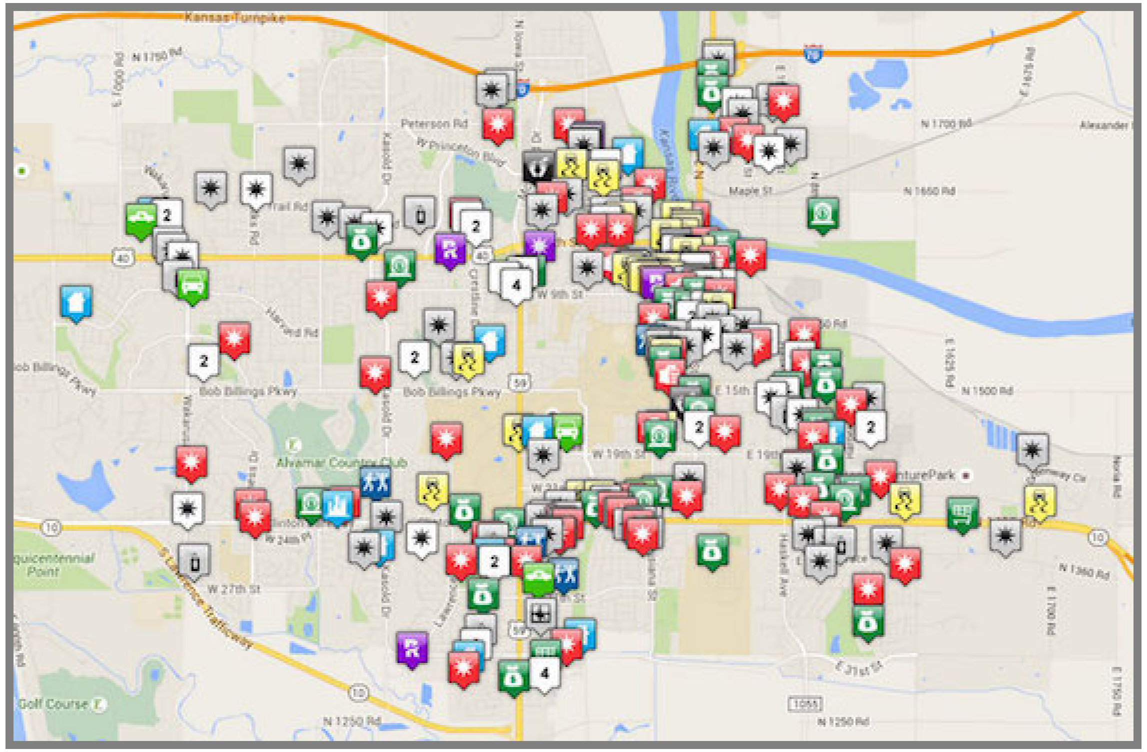


Speaker notes

Audit risk meter from Turbo-Tax

The System Interacts with the World

The System Interacts with the World



- Model: Use historical data to predict crime rates by neighborhoods
- Used for predictive policing: Decide where to allocate police patrol

User Interaction Design

Often: System interact with the world through *by influencing people* ("human in the loop")

Automate: Take action on user's behalf

Prompt: Ask the user if an action should be taken

Organize/Annotate/Augment: Add information to a display

Hybrids of these

Factors to Consider (from Reading)

Forcefulness: How strongly to encourage taking an action (or even automate it)?

Frequency: How often to interact with the user?

Value: How much does a user (think to) benefit from the prediction?

Cost: What is the damage of a wrong prediction?

Discussion: Safe Browsing

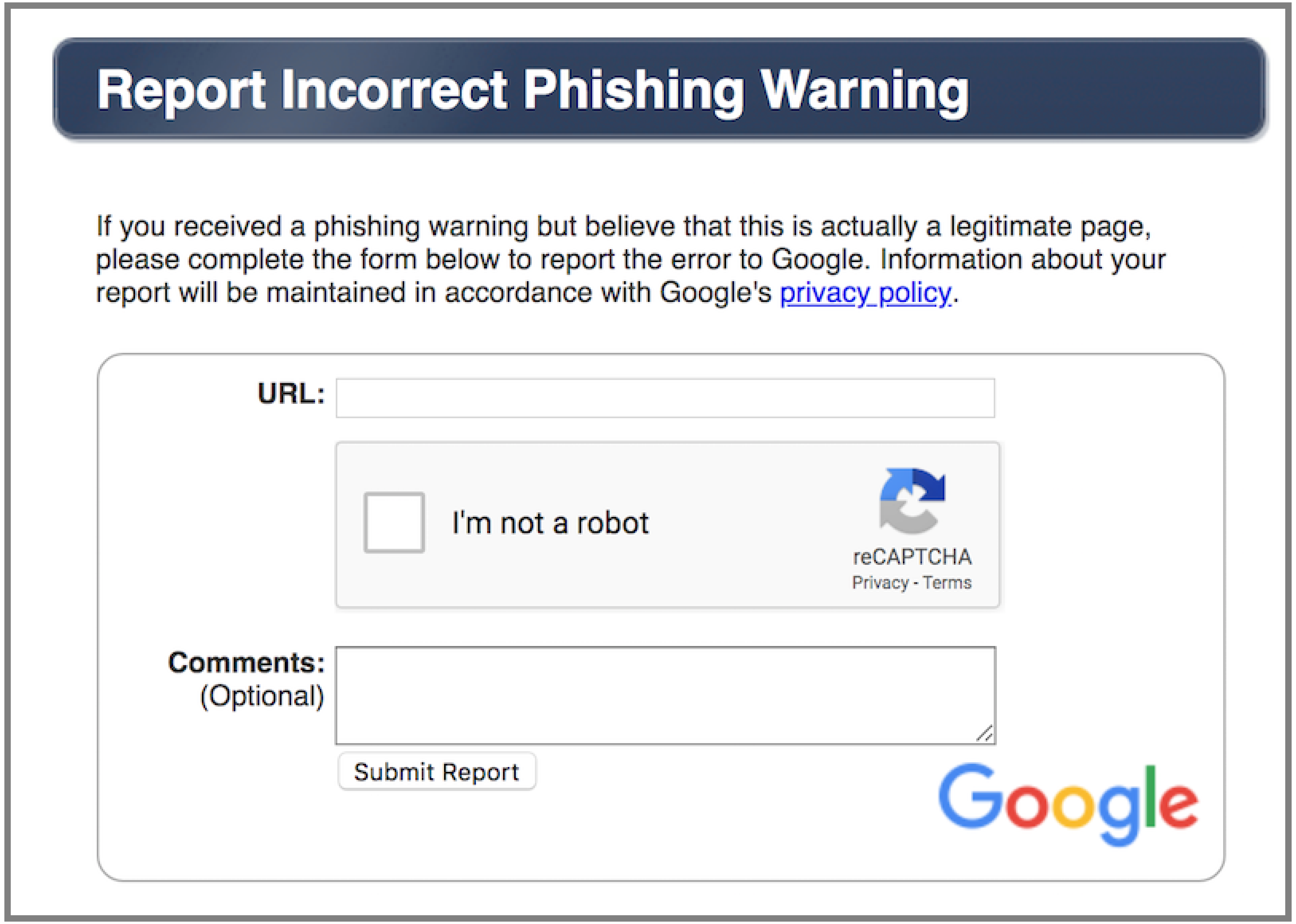
(1) How do we present the intelligence to the user?

(2) Justify in terms of system goals, forcefulness, frequency, value of correct and cost of wrong predictions

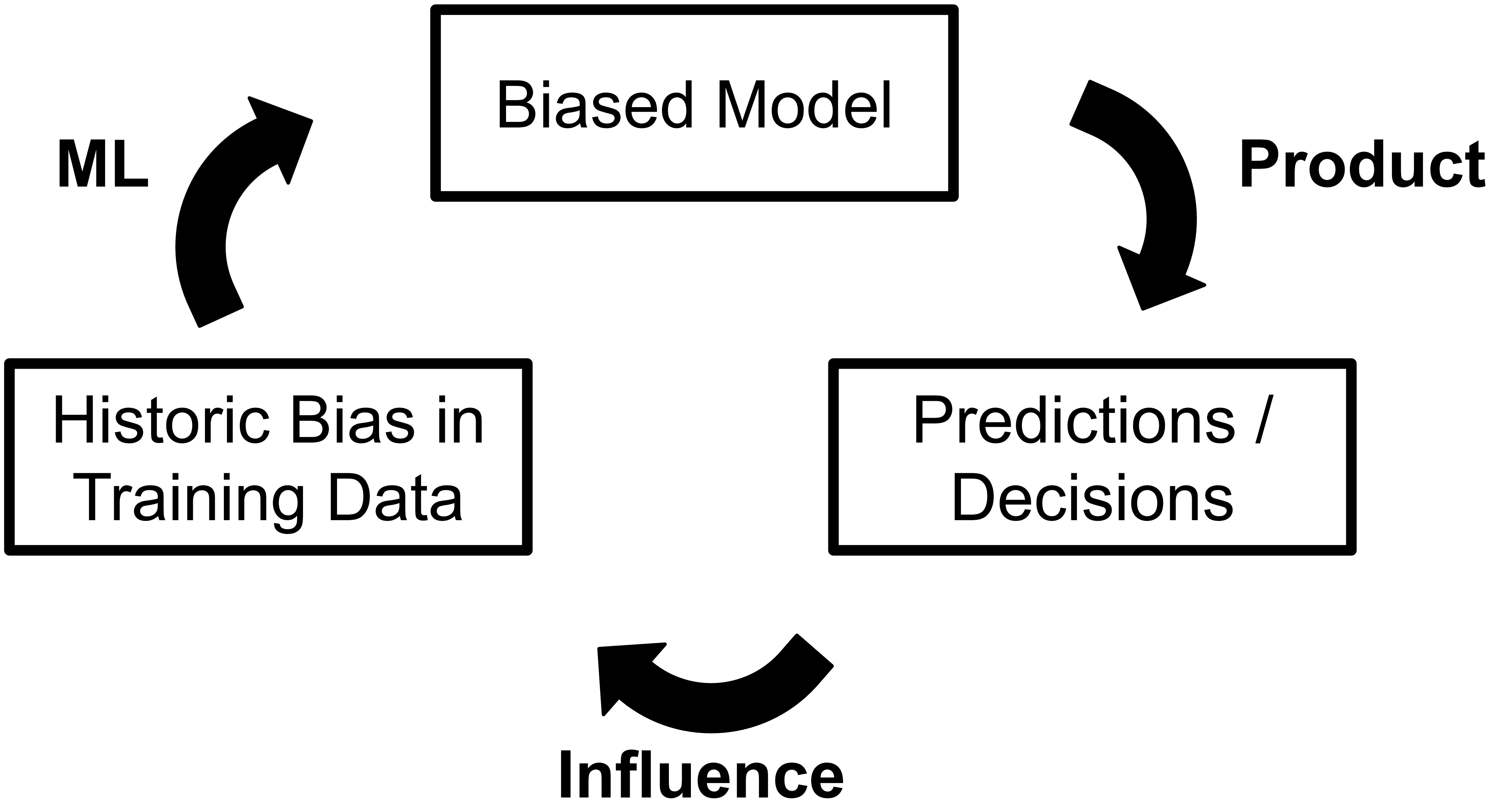
Speaker notes

Devices for older adults to detect falls and alert caretaker or emergency responders automatically or after interaction. Uses various inputs to detect falls. Read more: How fall detection is moving beyond the pendant, MobiHealthNews, 2019

Collecting Feedback



Feedback Loops



The System Interacts with the World

MIT Technology Review — Topics

Artificial intelligence

Predictive policing algorithms are racist. They need to be dismantled.

Lack of transparency and biased training data mean these tools are not fit for purpose. If we can't fix them, we should ditch them.

by **Will Douglas Heaven** — July 17, 2020

ML Predictions have Consequences

Assistance, productivity, creativity

Manipulation, polarization, discrimination

Feedback loops

➤ Need for **responsible engineering**

Safety is a System Property

- Code/models are not unsafe, cannot harm people
- Systems can interact with the environment in ways that are unsafe

Safety Assurance in/outside the Model

Goal: Ensure smart toaster does not burn the kitchen

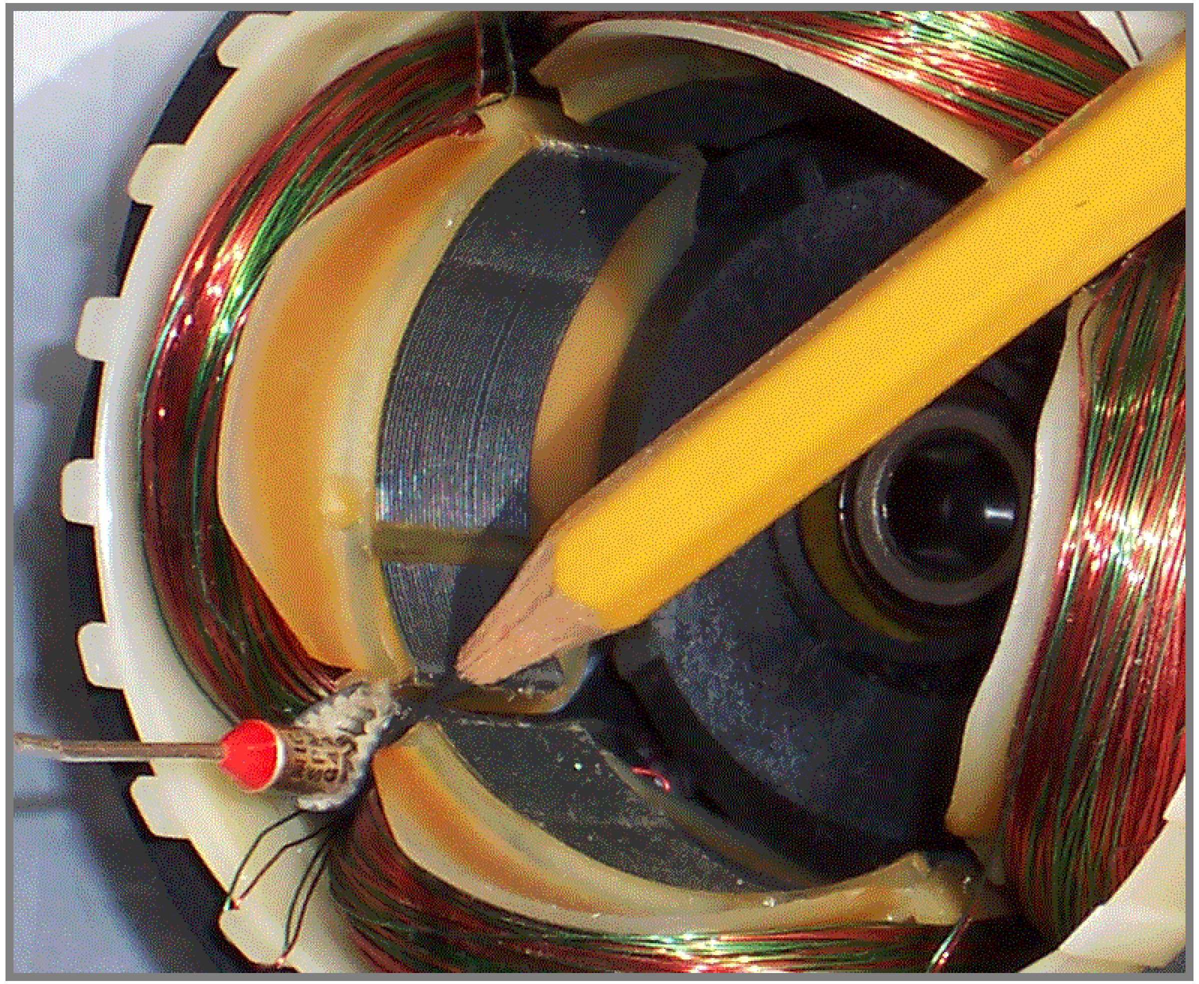
Safety Assurance in/outside the Model

In the model

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

Outside the model (e.g., "guardrails")

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



Model vs System Properties

Similar to safety, many other qualities should be discussed at model **and** system level

- Fairness
- Security
- Privacy
- Transparency, accountability
- Maintainability
- Scalability, energy consumption
- Impact on system goals
- ...

Thinking about Systems

- Holistic approach, looking at the larger picture, involving all stakeholders
- Looking at relationships and interactions among components and environments
  - Everything is interconnected
  - Combining parts creates something new with emergent behavior
  - Understand dynamics, be aware of feedback loops, actions have effects
- Understand how humans interact with the system

> *A system is a set of inter-related components that work together in a particular environment to perform whatever functions are required to achieve the system's objective -- Donella Meadows*

Leyla Acaroglu. "Tools for Systems Thinkers: The 6 Fundamental Concepts of Systems Thinking." Blogpost 2017

System-Level Challenges for AI-Enabled 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, nonlocal effects
- Deployment and maintenance
- Versioning, debugging and incremental improvement
- Keeping training and operating cost manageable
- Interdisciplinary teams
- Setting system goals, balancing stakeholders and requirements
- ...

Operating Production ML Systems

(deployment, updates)

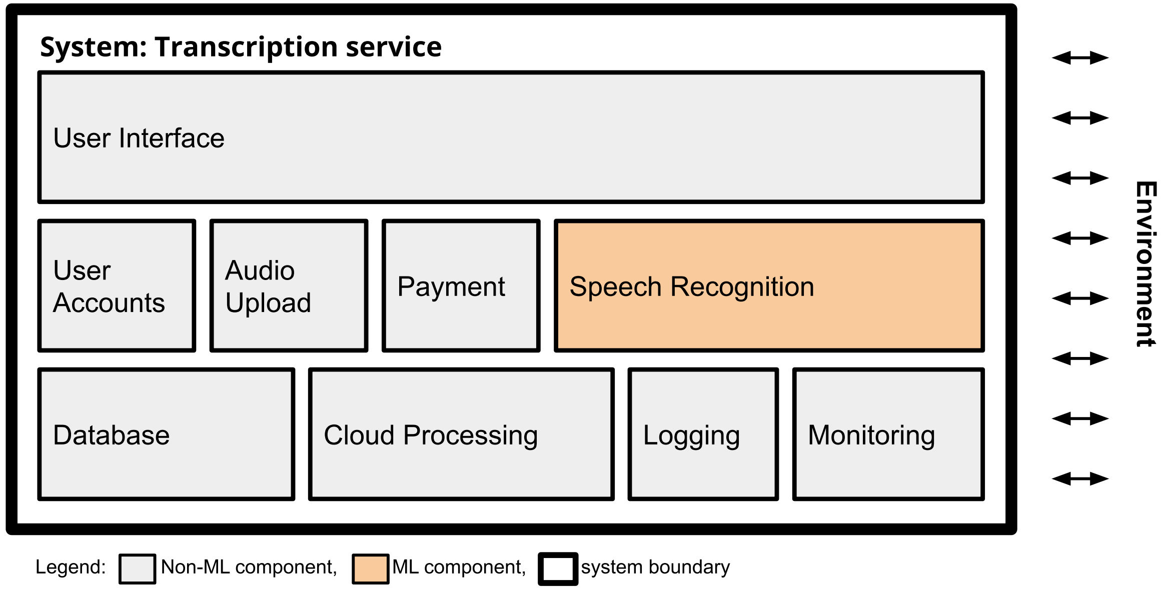
Things change...

Newer better models released (better model architectures, more training data, ...)

Goals and scope change (more domains, handling dialects, ...)

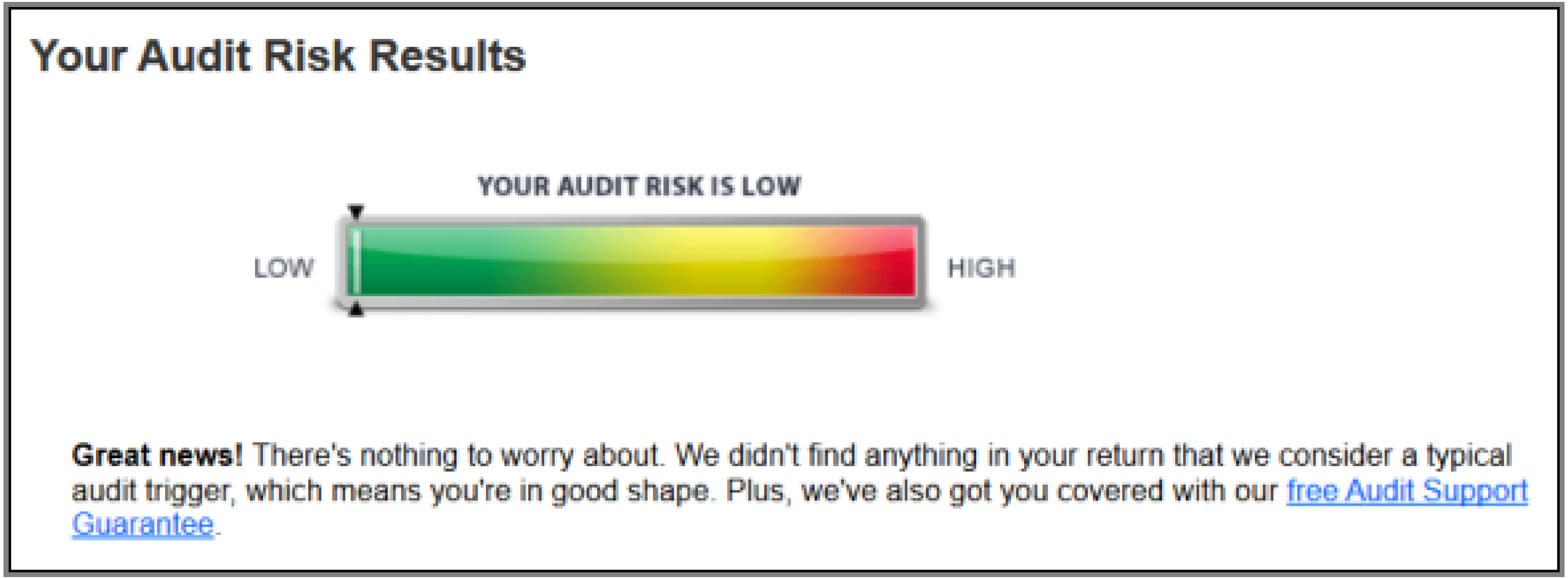
The world changes (new products, names, slang, ...)

Online experimentation



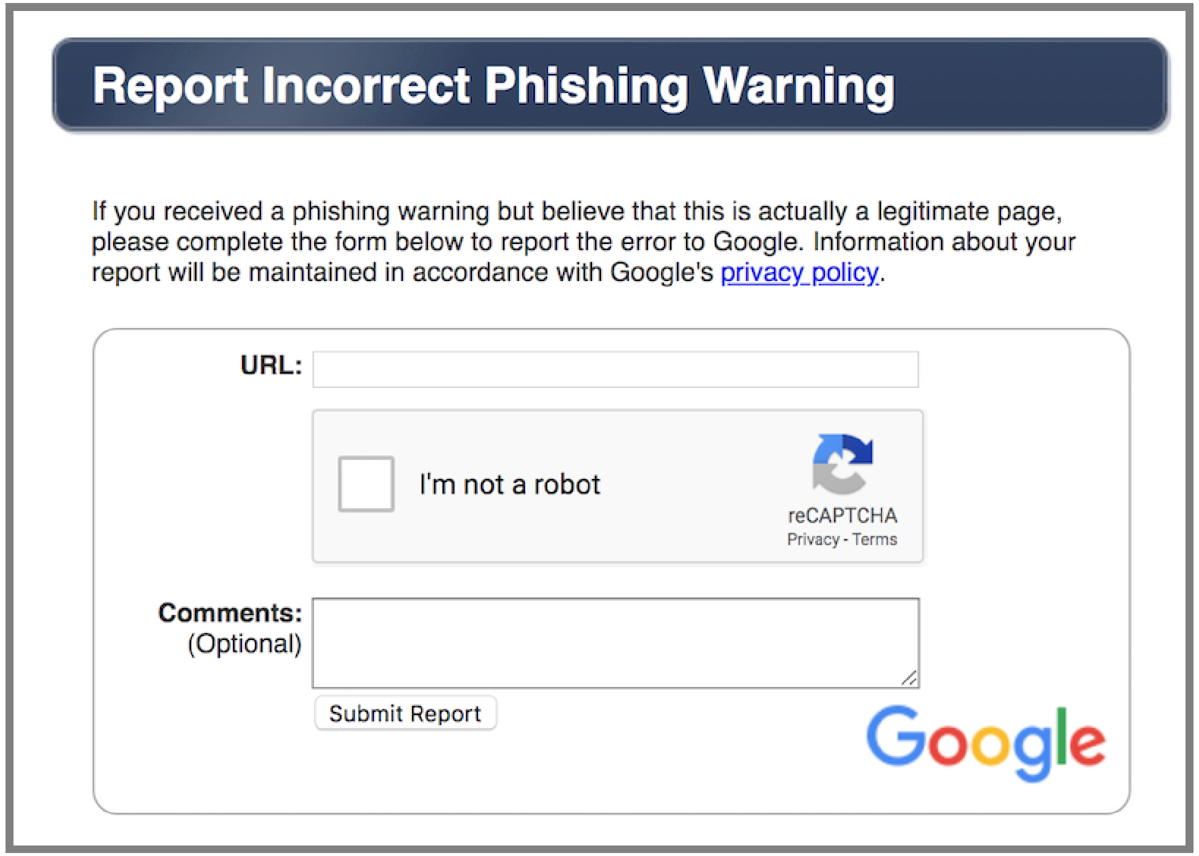
Things change...

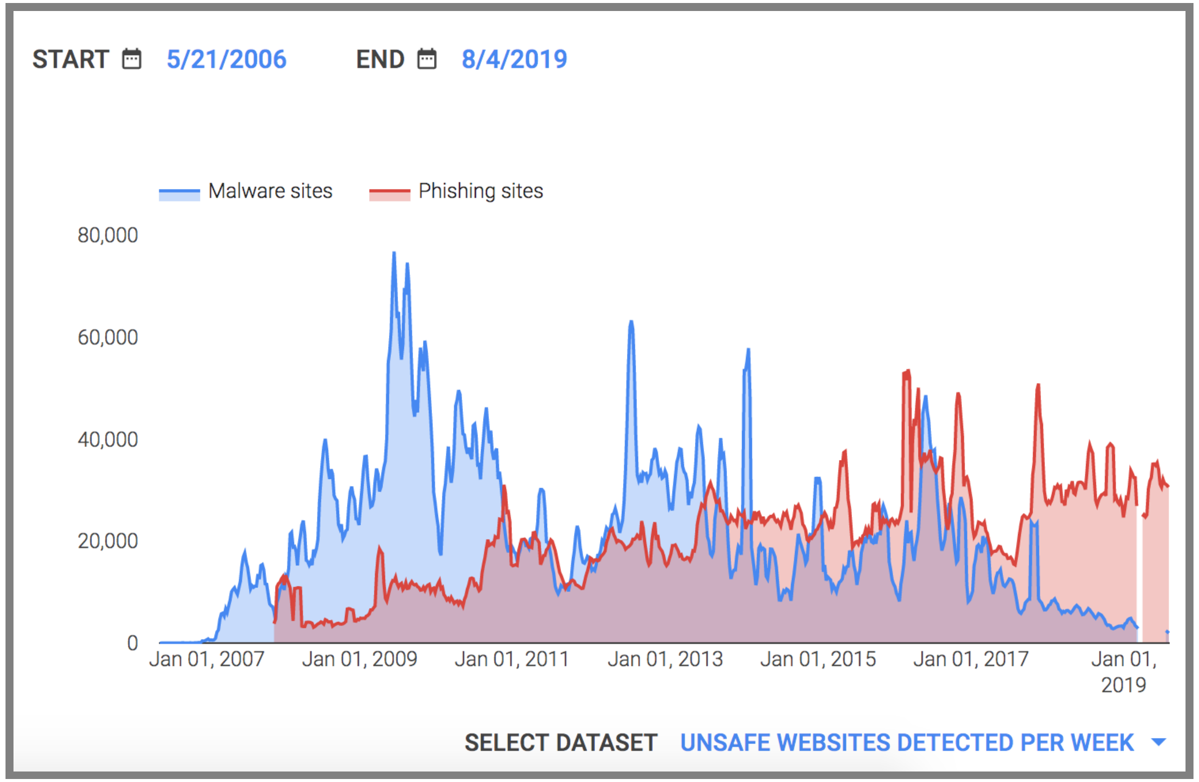
Reasons for change in audit risk prediction model?



Monitoring in Production

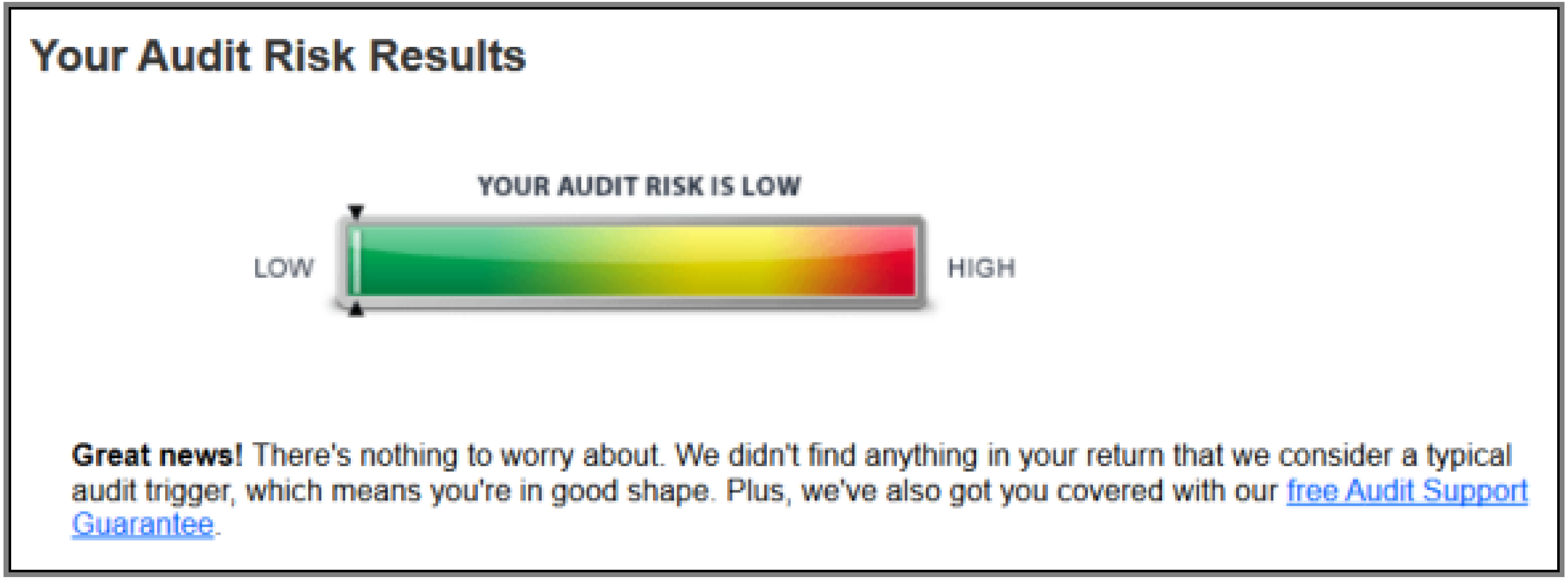
Design for telemetry



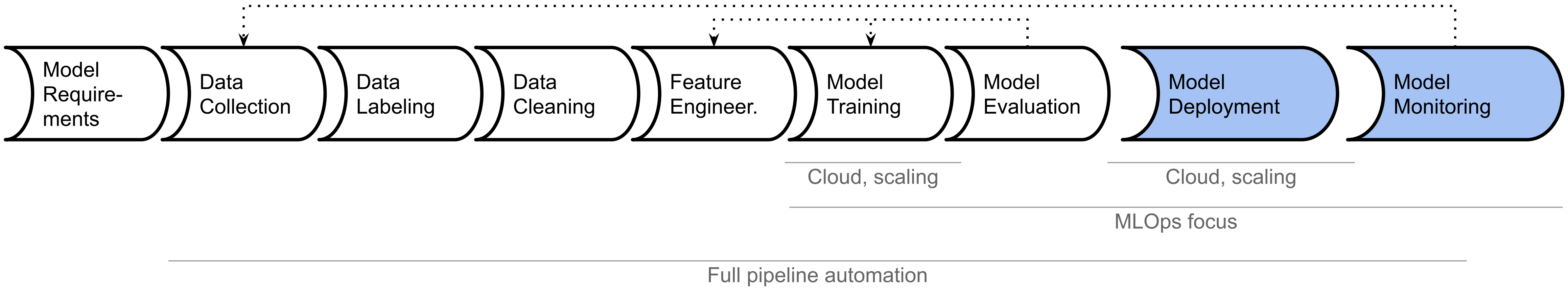


Monitoring in Production

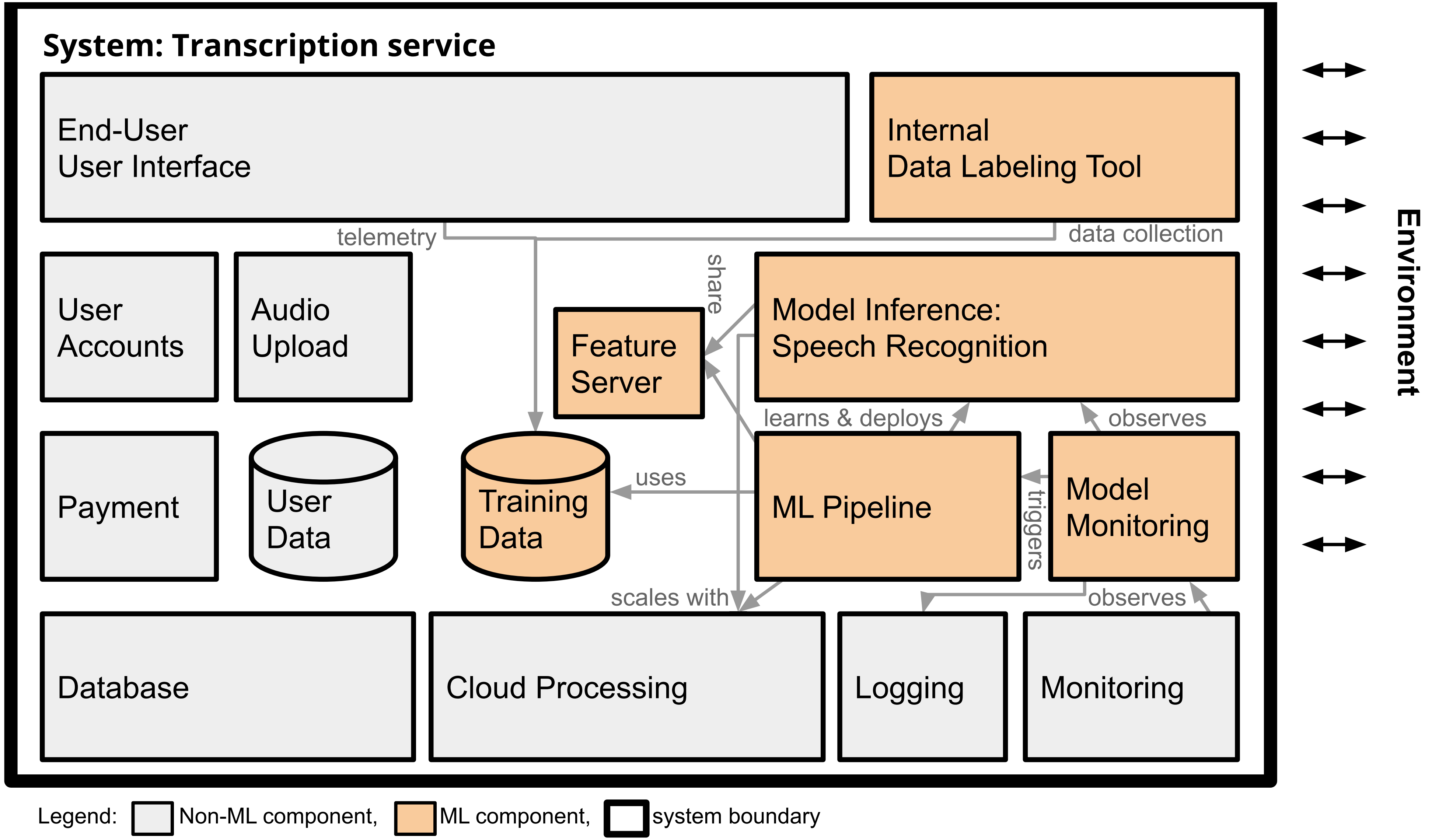
What and how to monitor in audit risk prediction?



Pipeline Thinking



Design with Pipeline and Monitoring



Pipelines Thinking is Challenging

In enterprise ML teams:

- Data scientists often focus on modeling in local environment, model-centric workflow
- Rarely robust infrastructure, often monolithic and tangled
- Challenges in deploying systems and integration with monitoring, streams etc

Shifting to pipeline-centric workflow challenging

- Requires writing robust programs, slower, less exploratory
- Standardized, modular infrastructure
- Big conceptual leap, major hurdle to adoption

Summary

Production AI-enabled systems require a *whole system perspective*, beyond just the model or the pipeline

Distinguish goals: organization, system, user, model goals

Quality at a *system* level: safety beyond the model, beyond accuracy

Large design space for user interface (intelligent experience): forcefulness, frequency, telemetry

Plan for operations (telemetry, updates)

Recommended Readings

- Passi, S., & Sengers, P. (2020). Making data science systems work. Big Data & Society, 7(2).
- Wagstaff, Kiri. "Machine learning that matters." In Proceedings of the 29th International Conference on Machine Learning, (2012).
- Sculley, David, Gary Holt, Daniel Golovin, Eugene Davydov, Todd Phillips, Dietmar Ebner, Vinay Chaudhary, Michael Young, Jean-Francois Crespo, and Dan Dennison. "Hidden technical debt in machine learning systems." In Advances in neural information processing systems, pp. 2503-2511. 2015.
- O'Leary, Katie, and Makoto Uchida. "Common problems with Creating Machine Learning Pipelines from Existing Code." Proc. Third Conference on Machine Learning and Systems (MLSys) (2020).
- Nushi, Besmira, Ece Kamar, Eric Horvitz, and Donald Kossmann. "On human intellect and machine failures: troubleshooting integrative machine learning systems." In *Proceedings of the Thirty-First AAAI Conference on Artificial Intelligence*, pp. 1017-1025. 2017.
- Nahar, Nadia, Shurui Zhou, Grace Lewis, and Christian Kästner. "Collaboration Challenges in Building ML-Enabled Systems: Communication, Documentation, Engineering, and Process." In Proceedings of the 44th International Conference on Software Engineering (ICSE), May 2022.
- Yang, Qian. "The role of design in creating machine-learning-enhanced user experience." In 2017 AAAI Spring Symposium Series. 2017.
- Sambasivan, Nithya, Shivani Kapania, Hannah Highfill, Diana Akrong, Praveen Paritosh, and Lora M. Aroyo. "“Everyone wants to do the model work, not the data work”: Data Cascades in High-Stakes AI". In proceedings of the 2021 CHI Conference on Human Factors in Computing Systems, pp. 1-15. 2021.
- Bernardi, Lucas, Themistoklis Mavridis, and Pablo Estevez. "150 successful machine learning models: 6 lessons learned at Booking.com." In Proceedings of the 25th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining, pp. 1743–1751. 2019.