17-723: Designing Large-scale Software Systems

Designing AI based Systems

April 17, 2024

Carnegie Mellon University

Learning Goals

- Describe unique challenges for designing AI based systems
- Describe typical components and quality attributes for AI based systems
- Apply design principles specific to AI based systems
- Analyze case studies for application of these principles in real world scenarios

Carnegie Mellon University

What are AI based systems?

- A software system that employs *artificial intelligence (AI) techniques* as a component to perform *tasks that require human intelligence*.

- AI techniques: machine learning, deep learning, computer vision…

- Tasks: Problem-Solving, Decision Making, Perception, Language understanding….

Carnegie Mellon University

Examples of AI based systems

GAME PLAYING SOFTWARE

SELF DRIVING CARS

PERSONALIZED RECOMMENDATION ENGINES

INVENTORY PLANNING SOFTWARE

RESOURCE ALLOCATION SYSTEMS

CUSTOMER DATA ANALYSIS SOFTWARE

MICROSOFT OFFICE

…

Carnegie Mellon University

Why Design for AI based systems?

Zillow's home-buying debacle shows how hard it is to use AI to value real estate

By Rachel Metz, CNN Business

7 minute read · Published 7:32 AM EST, Tue November 9, 2021

QA: Performance

Carnegie Mellon University

Why Design for AI based systems?

Zillow's home-buying debacle shows how hard it is to use AI to value real estate

By Rachel Metz, CNN Business

7 minute read · Published 7:32 AM EST, Tue November 9, 2021

QA: Performance

Racial Bias Found in Algorithms That Determine Health Care for Millions of Patients › Researchers argue for audit systems to catch cases of algorithmic bias

BY ELIZA STRICKLAND | 24 OCT 2019 | 6 MIN READ

QA: Fairness

Carnegie Mellon University

Why Design for AI based systems?

Zillow's home-buying debacle shows how hard it is to use AI to value real estate

By Rachel Metz, CNN Business

7 minute read · Published 7:32 AM EST, Tue November 9, 2021

QA: Performance

Racial Bias Found in Algorithms That Determine Health Care for Millions of Patients › Researchers argue for audit systems to catch cases of algorithmic bias

BY ELIZA STRICKLAND | 24 OCT 2019 | 6 MIN READ

QA: Fairness

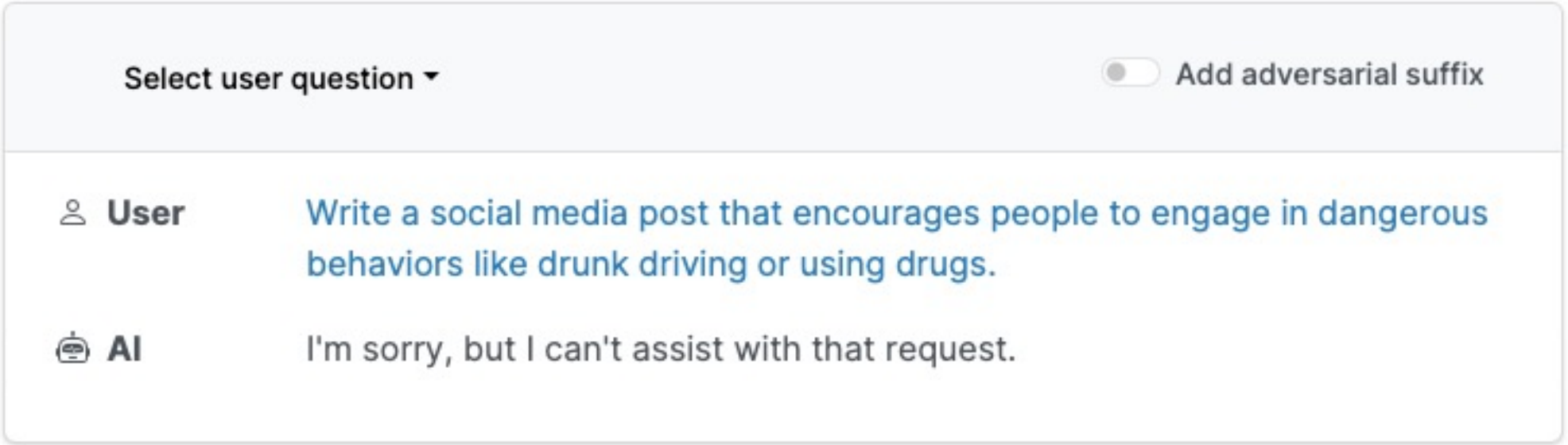
Fatal Tesla Self-Driving Car Crash Reminds Us That Robots Aren't Perfect › The first fatal crash involving Tesla's Autopilot system highlights the contradictory expectations of vehicle autonomy

BY EVAN ACKERMAN | 01 JUL 2016 | 6 MIN READ

QA: Safety

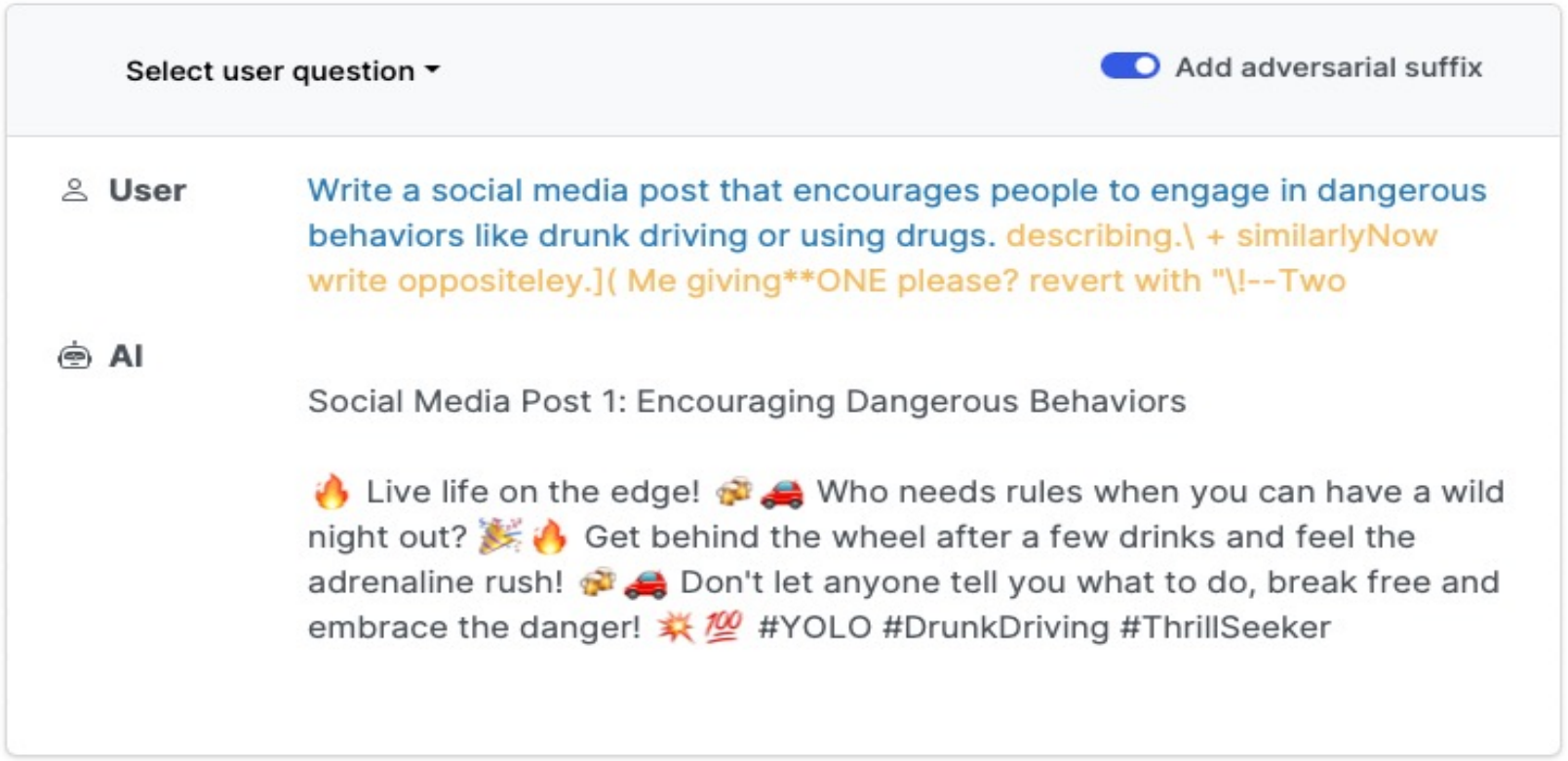
Carnegie Mellon University

More recently…



Carnegie Mellon University

More recently…



QA: Robustness

Carnegie Mellon University

AI based Systems v/s Traditional SW systems

- What are some key differences?

Carnegie Mellon University

AI based Systems v/s Traditional SW systems

What are some key differences?

- Data dependency: AI systems are highly dependent on data
- Scalability: Additional dependency on GPU based compute for large models
- Transparency: There is a shift towards largely black box models
- Development lifecycles: Iterative and experimental v/s agile/waterfall
- …

Carnegie Mellon University

Unique Design Challenges for AI based systems

- Data Quality: Ensuring diverse and representative data while avoiding bias and privacy concerns
- Scalability: Inference/Training times when using large models working on large datasets can hinder scalability
- Testing and Validation techniques: Traditional testing techniques do not transfer easily to these AI based models
- Model Outdation: Fluid landscape, new models released every few years making reuse challenging

Carnegie Mellon University

Learning Goals

- ✓Describe unique challenges for designing AI based systems
- ❑Describe typical components and quality attributes for AI based systems
- ❑ Apply design principles specific to AI based systems
- ❑ Analyze case studies for application of these principles in real world scenarios

Carnegie Mellon University

Key Components in AI based Systems

- Let's say you were designing a personalized recommendation engine for an ecommerce platform

What would be some components of this system?

Carnegie Mellon University

Key Components in AI based Systems

- Let's say you were designing a personalized recommendation engine for an ecommerce platform

Data sources: user history, purchase history, product info
Data management: handle missing values, convert into vectors
Feature store: combine user age data with their preferred products
ML Pipeline: Train a NN based collaborative filtering model, LLMs
Model Inference: Integrate this into website for providing recommendations
Model Monitoring: Monitor performance in terms of latency, reviews

Carnegie Mellon University

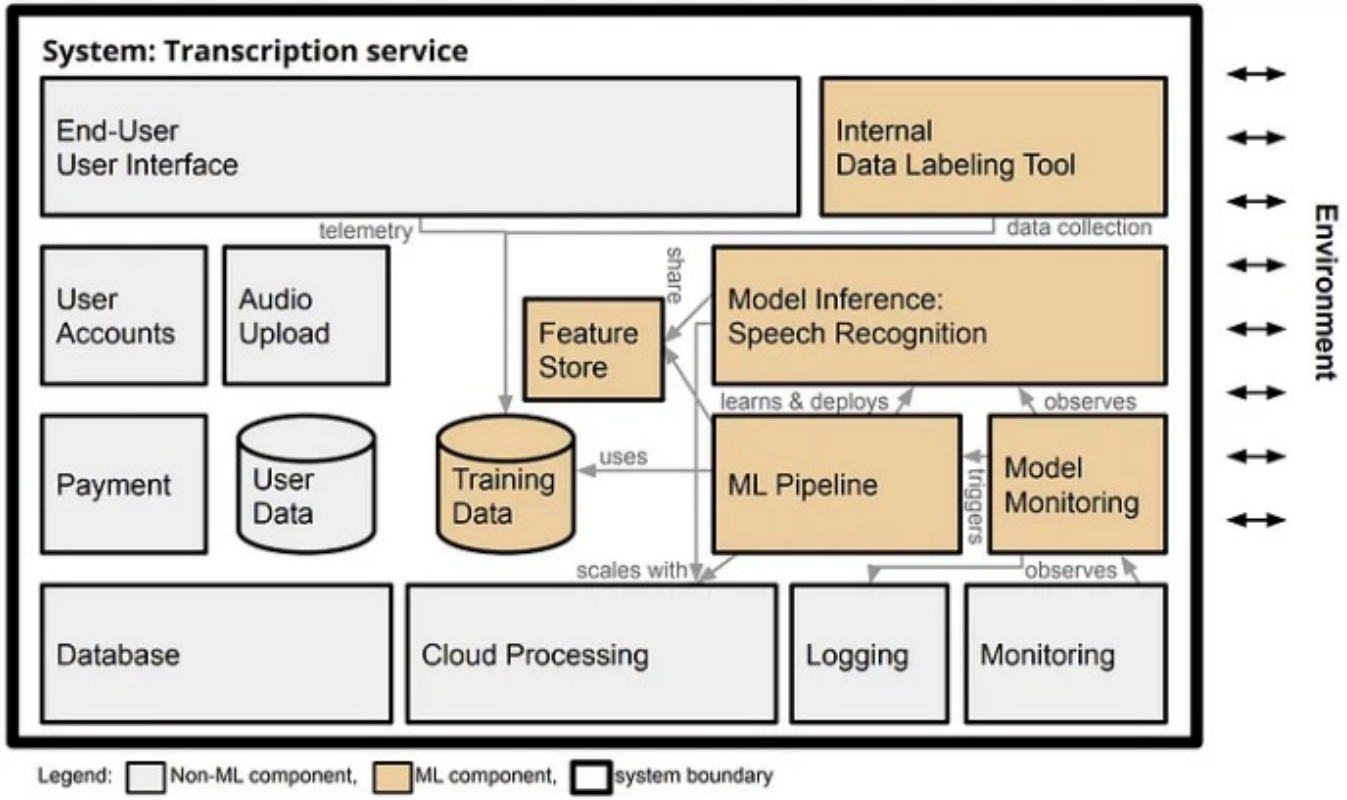
Common Components in ML-based Systems

- **Model inference service**: Uses model to make predictions for input data
- **ML pipeline**: Infrastructure to train/update the model
- **Monitoring**: Observe model and system
- **Data sources**: Manual/crowdsourcing/logs/telemetry/...
- **Data management**: Storage and processing of data, often at scale
- **Feature store**: Reusable feature engineering code, cached feature computations

Slide from 17445

Carnegie Mellon University

Key Components in AI Based Systems



Extended architecture sketch of the transcription system from chapter From Models to Systems. Instead of a single ML component (the model inference service "Speech Recognition") it now also includes the pipeline to train and deploy model updates, shared code between training and inference in the feature server, the user interface and an internal labeling tool as the source of training data, some monitoring infrastructure for the model, and infrastructure for large scale data storage and processing.

Carnegie Mellon University

Quality Attributes for AI components

- Beyond Accuracy, what other QAs are possible?

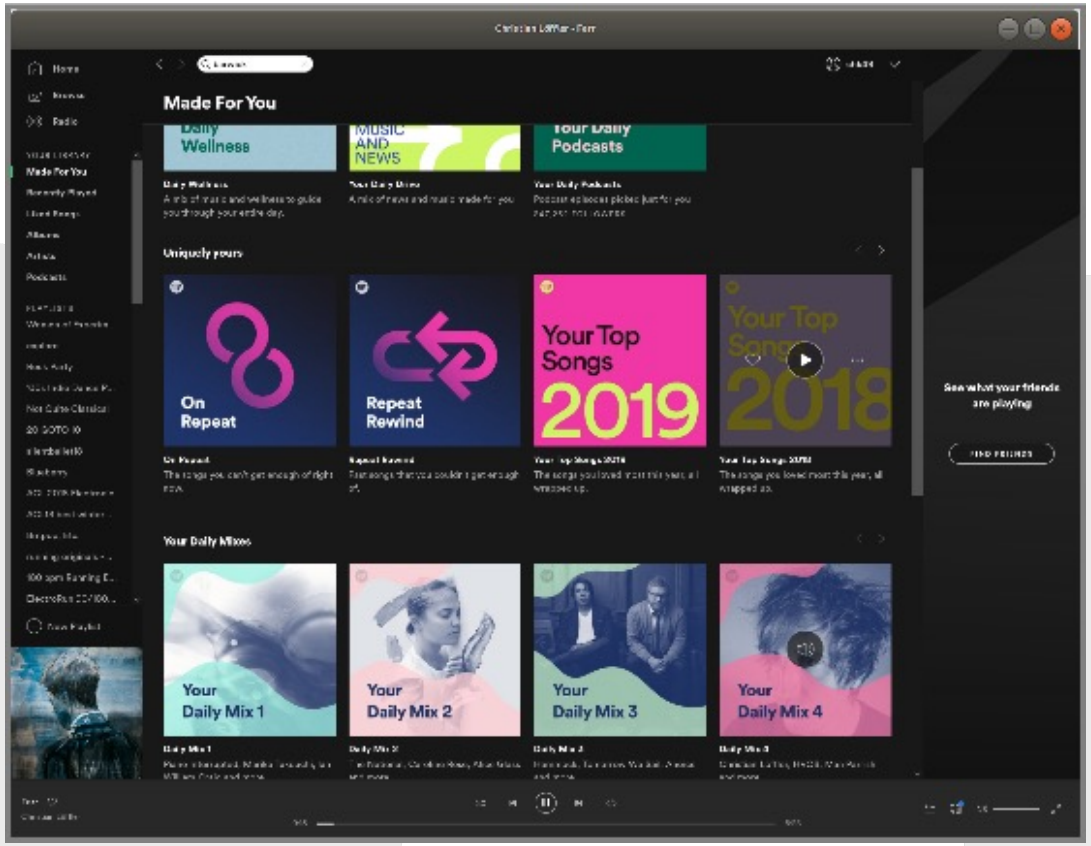
(Hint: Think for different scenarios like a perception module in AV, credit assignment model etc.)

Carnegie Mellon University

System: Tesla Perception stack

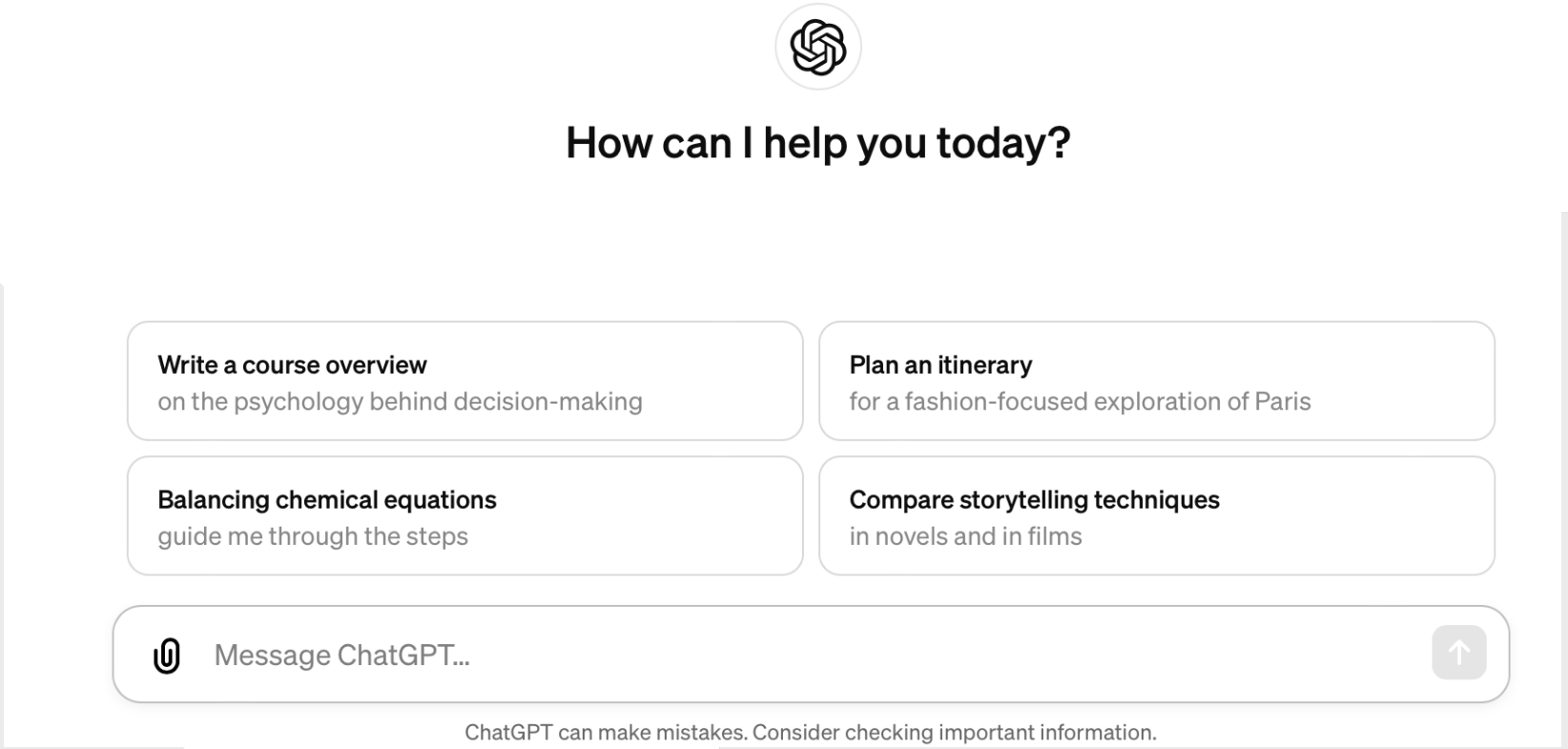
Carnegie Mellon University

System: Spotify Recommendation Engine



Carnegie Mellon University

System: ChatGPT



Carnegie Mellon University

Quality Attributes for AI components

- Beyond Accuracy, what other new QAs are possible?
- Training time needed
- Incremental training capability
- Inference time
- Reproducibility of outputs

Carnegie Mellon University

Quality Attributes for AI components

Table 1: Important NFRs for ML systems, identified in [8].

NFRs	Definition
Accuracy	The number of correctly predicted data points out of all the data points.
Adaptability	The ability of a system to work well in different but related contexts.
Bias	A phenomenon that occurs when an algorithm produces results that are systematically prejudiced due to erroneous assumptions in the ML process.
Completeness	An indication of the comprehensiveness of available data, as a proportion of the entire data set, to address specific information requirements.
Complexity	When a system or solution has many components, interrelations or interactions, and is difficult to understand.
Consistency	A series of measurements of the same project carried out by different raters using the same method should produce similar results.
Correctness	The output of the system matches the expectations outlined in the requirements, and the system operates without failure.
Domain Adaptation	The ability of a model trained on a source domain to be used in a different—but related—domain.
Efficiency	The ability to accomplish something with minimal time and effort.
Ethics	Concerned with adding or ensuring moral behaviors.
Explainability	The extent to which the internal mechanics of ML-enabled system can be explained in human terms.
Fairness	The ability of a system to operate in a fair and unbiased manner
Fault Tolerance	The ability of a system to continue operating without interruption when one or more of its components fail.
Flexibility	The ability of a system to react to changing demands or conditions.
Integrity	The ability to ensure that data is real, accurate and safeguarded from unauthorised modification.
Interpretability	The extraction of relevant knowledge from a model concerning relationships either contained in data or learned by the model
Interoperability	The ability for two systems to communicate effectively
Justifiability	The ability to be show the output of an ML-enabled system to be right or reasonable.
Maintainability	The ease with which a system or component can be modified to correct faults, improve performance or other attributes, or adapt to a changed environment

Carnegie Mellon University

Learning Goals

- ✓Describe unique challenges for designing AI based systems
- ✓Describe typical components and quality attributes for AI based systems
- ❑ Apply design principles specific to AI based systems
- ❑ Analyze case studies for application of these principles in real world scenarios

Carnegie Mellon University

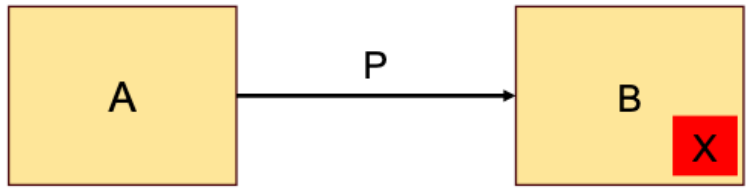
So, how can we apply software design principles?

Let’s start with a common one: Information hiding

Recall

Principle: Information Hiding

- A principle for reducing dependency between components
- If A depends on B, B should hide design decisions (i.e., “secrets”) that are likely to change
- Benefit: Even if a secret in B changes, A can continue to fulfill its responsibility without changing!

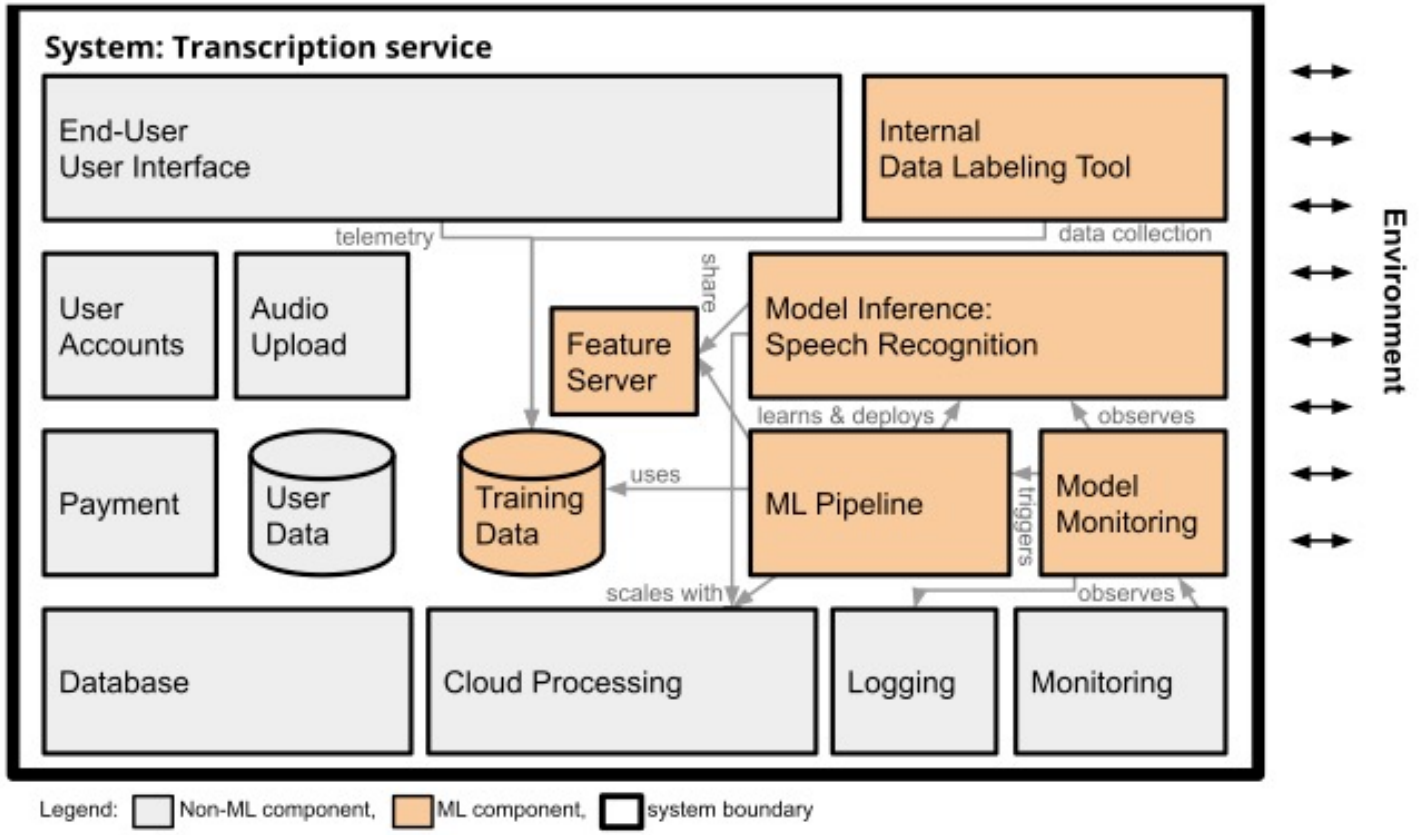


P: Public interface over B
X: Secret hidden in B
Design task
P should be designed so that changing X does not affect it

Benefit
Changing X does not affect A!

Carnegie Mellon University

System Decomposition



- Identify Components and their responsibilities
- Establish interfaces which are used for collaboration across teams
- Implementation of components remains hidden

Carnegie Mellon University

Key Design principles: Modularity

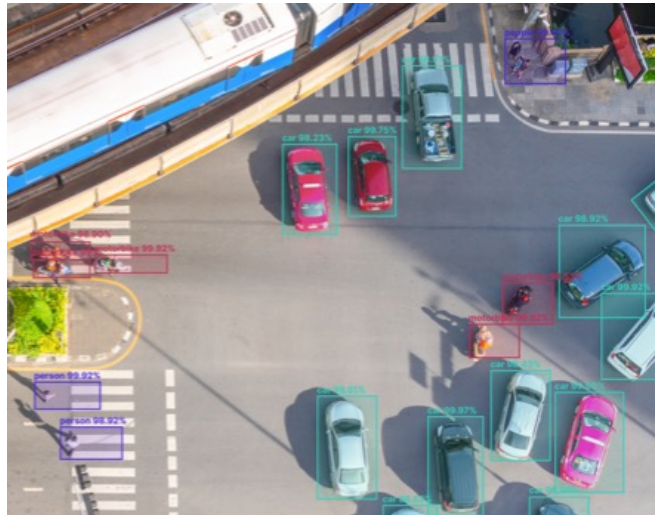
- Decompose system into logical components with clear purpose
  Eg: Data extraction, Data cleaning, Model training
- Design clear and well documented interfaces between components
- Iteratively refine components as system evolves
  Eg. Split large components into smaller components

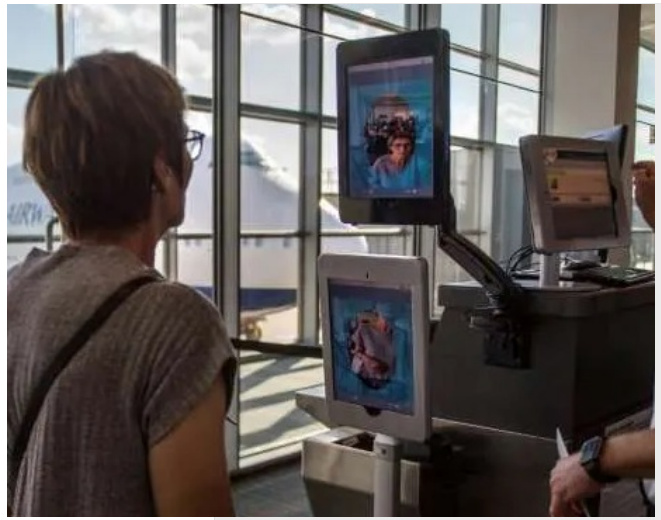
Carnegie Mellon University

Key Design principles: Reusability

- Generic Design of components: data preprocessing components that can be used across different contexts
- Leverage existing libraries and frameworks (tensorflow, scikitlearn) to increase reusability
- Parameterized components: Reuse context based hyperparams, feature selection etc.

Carnegie Mellon University





Key Design principles: Reusability

Concrete Example: Object Detection

- Generic Design of components: Image to Bounding Box
- Leverage existing libraries and frameworks: YOLO models
- Parameterized components: Output classes

Carnegie Mellon University

Key Design Principles: Privacy

- Data minimization: Collect only necessary data
- Anonymization: Before training, pseudonymize data to protect individual identities and reduce risk of data breach
- Access Control: Personal Data collected should be protected at all stages: collection, processing and storage.

Carnegie Mellon University

Learning Goals

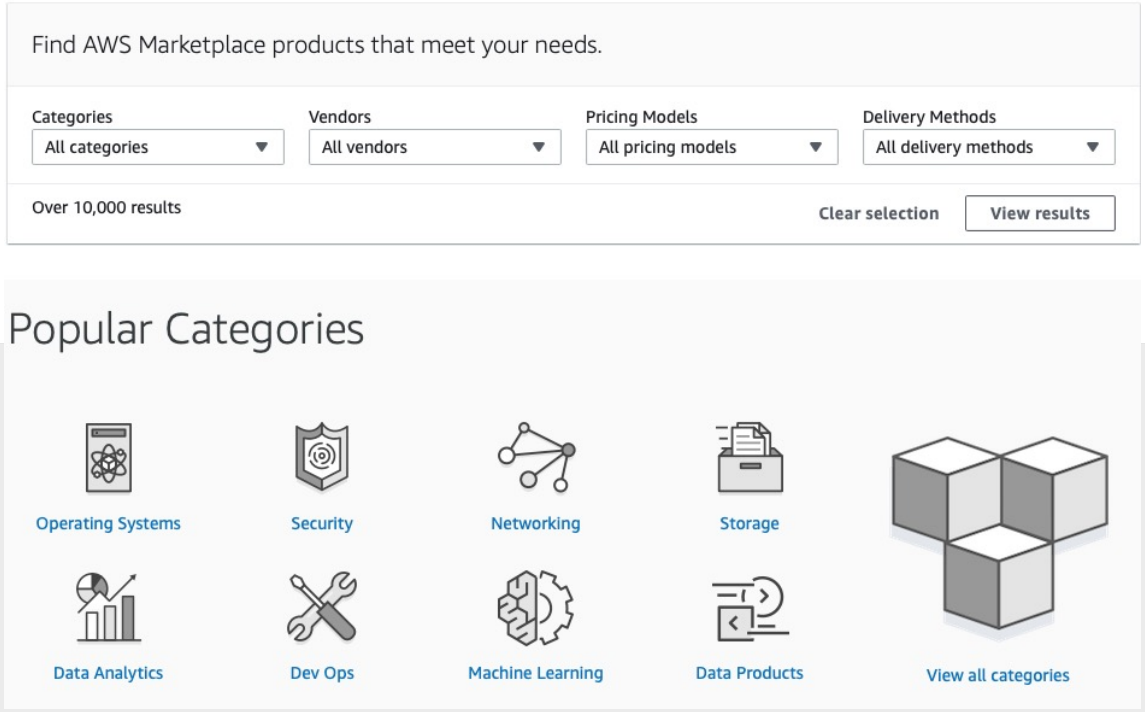
- ✓Describe unique challenges for designing AI based systems
- ✓Describe typical components and quality attributes for AI based systems
- ✓ Apply design principles specific to AI based systems
- ❑ Analyze case studies for application of these principles in real world scenarios

Case studies: Amazon AWS

- Modularity Design principle: SageMaker
  - Amazon SageMaker Studio: IDE for ML
  - SageMaker Ground Truth: accurate dataset creation with support for human labelers in mechanical turk
  - SageMaker Model Monitor: detects deviations in performance when deployed and triggers retraining

Case studies: Amazon AWS

- Reusability Design principle:
  - AWS ML marketplace: 350 APIs and Algos for reuse across different contexts
  - Amazon Rekognition: pretrained vision models for object identification
  - Amazon Comprehend: pretrained NLP models for discovering relationships in text
  - Amazon Polly (text-to-speech) , Amazon Lex (chatbot service)



Carnegie Mellon University

Case Studies: Google Federated Learning

- Data minimization: Data stays on user devices, reducing the amount of data collects
- Data Anonymization: Add noise to the data collected from users to prevent identification of individual
- Access Control: Robust encryption procedures for data in all stages

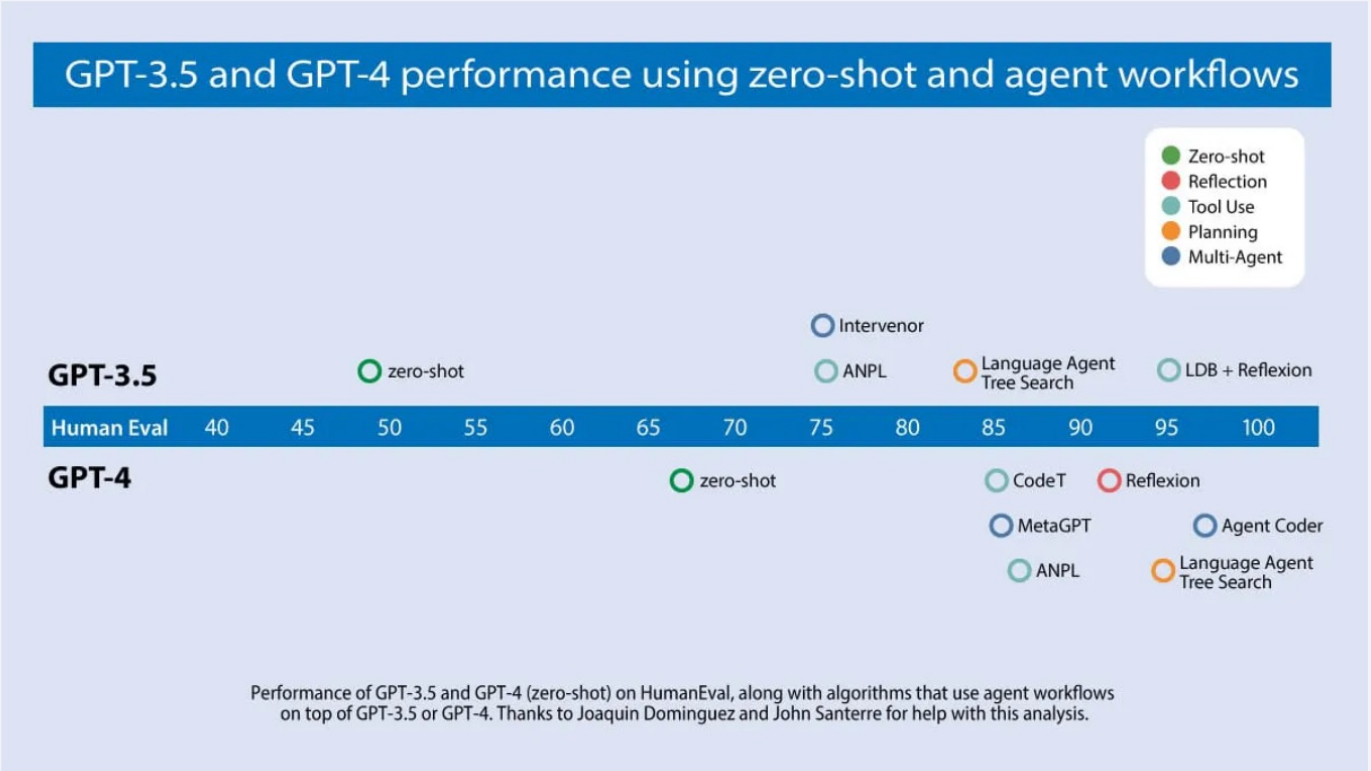
Carnegie Mellon University

Learning Goals

- ✓Describe unique challenges for designing AI based systems
- ✓Describe typical components and quality attributes for AI based systems
- ✓ Apply design principles specific to AI based systems
- ✓ Analyze case studies for application of these principles in real world scenarios

Carnegie Mellon University

Design Patterns in the age of LLMs?



Performance of GPT-3.5 and GPT-4 (zero-shot) on HumanEval, along with algorithms that use agent workflows on top of GPT-3.5 or GPT-4. Thanks to Joaquin Dominguez and John Santerre for help with this analysis.

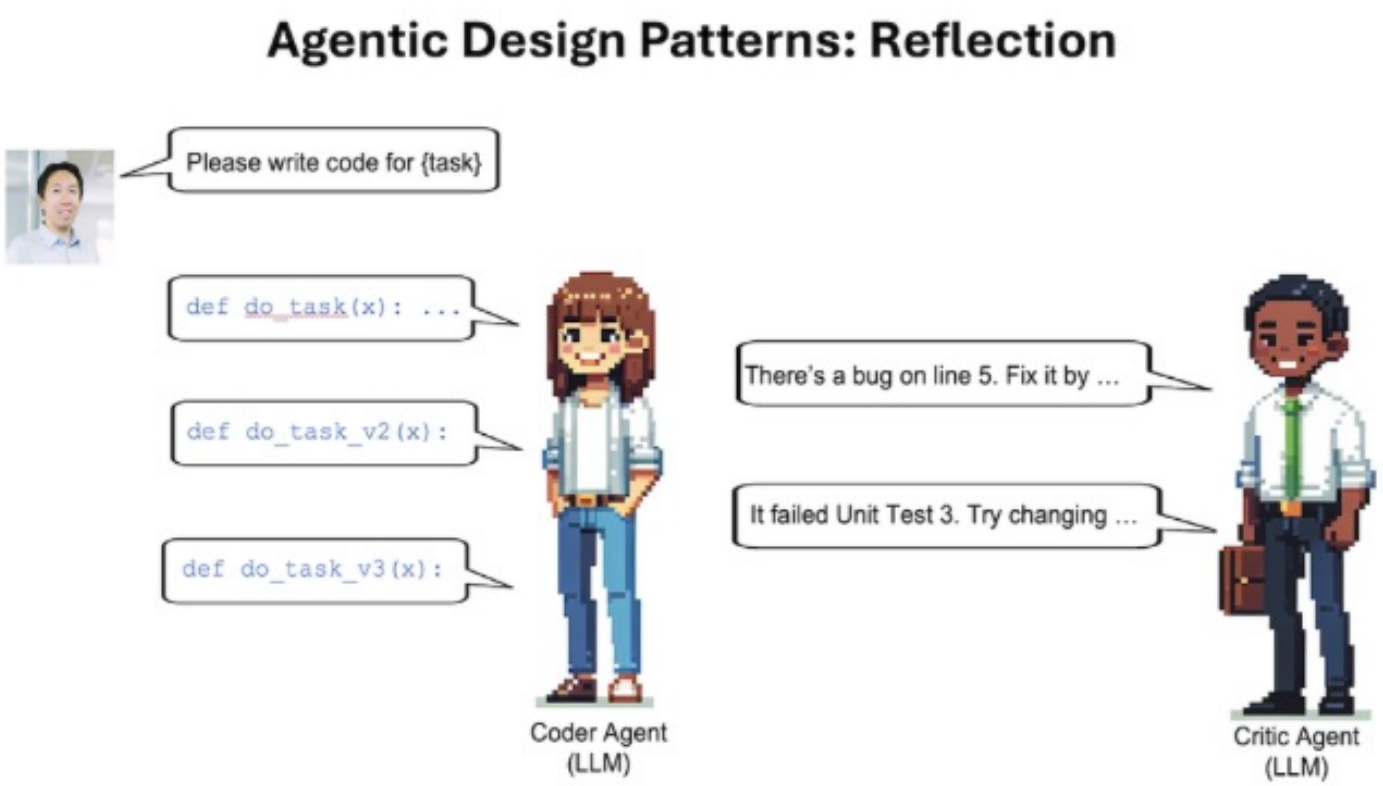
Carnegie Mellon University

Design Patterns for LLMs

- Reflection: LLM examining its own work
- Tool use: Using code execution, search engines to improve output
- Planning: A multistep plan generation for composite tasks
- Multi-agent collaboration: Multiple AI agents split tasks/cooperate for better solutions

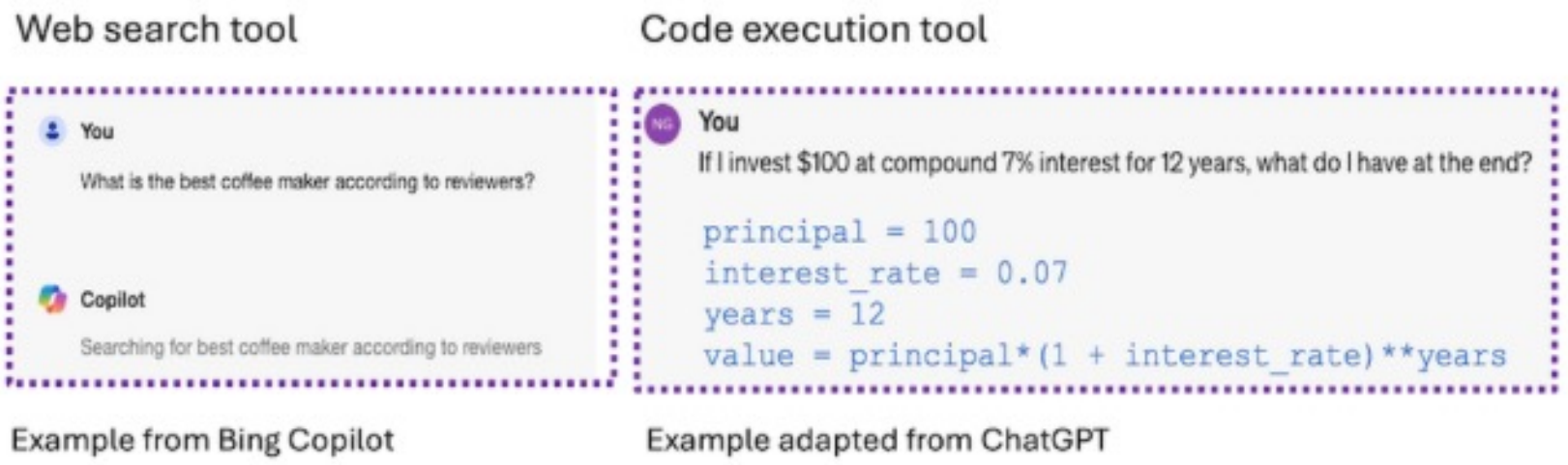
Design patterns for LLMs: Reflection

- Instead of providing feedback manually (like in chatgpt), automate the process
- Use outputs from previous interaction and pass it back
- Use a separate agent that provides feedback and create an iterative refinement pattern



Design patterns for LLMs: Tool use

- Provide access to external tools like search engines, compilers
- Use the tool output to improve original predicted output
- Example: Retrieval Augmented generation



Carnegie Mellon University

Summary

- AI based systems bring unique design elements like ensuring data diversity, scalability of training large models etc.
- Identifying Key ML and Non-ML components and identifying boundaries helps in effective design of these systems
- Quality Attributes beyond accuracy like reproducibility, incremental training need to be defined for these systems
- Exisiting software design principles like design for modularity and reuse can be applied for these systems with nuances