Today is about System Design

To build things we need to design them first.

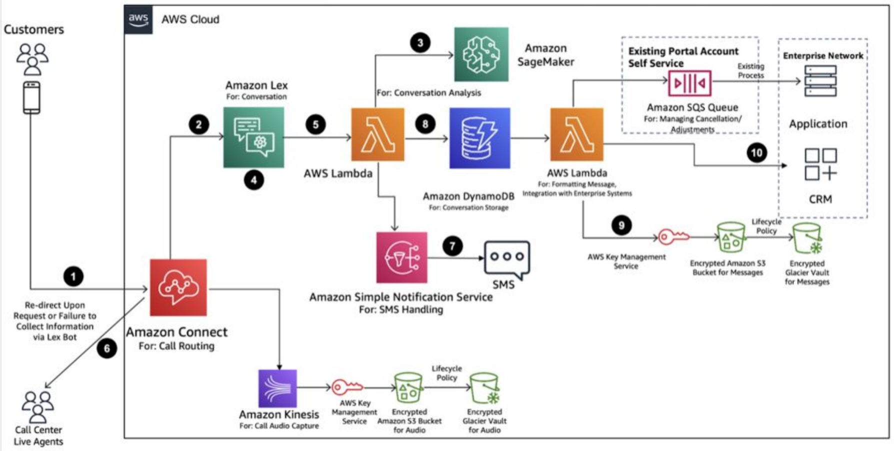
Obviously.

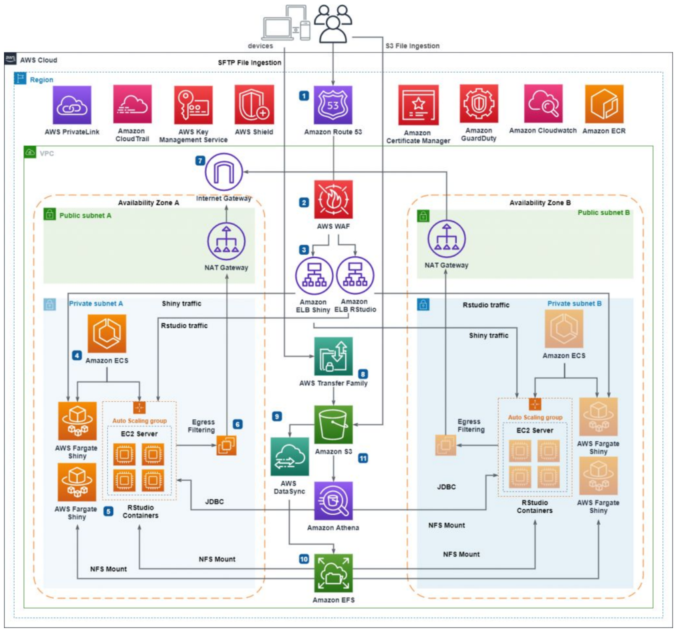
System Design

System Design
How we build a system — defining its structure, components, and how they interact to achieve a goal.

System Thinking
How we understand a system — focusing on the relationships, patterns, and the bigger picture beyond individual parts.

System Design





Note: these are AWS diagrams

System Design

Relevant resources

Machine-Learning-Interviews/src/MLSD/ml-system-design.md at main

Machine learning systems design

System Design (in the AI world)

Step 1	Problem Formulation
Step 2	Metrics (Offline and Online)
Step 3	Architectural Components (MVP Logic)
Step 4	Data Collection and Preparation
Step 5	Feature Engineering
Step 6	Model Development and Offline Evaluation
Step 7	Prediction Service
Step 8	Online Testing and Deployment
Step 9	Scaling, Monitoring, and Updates

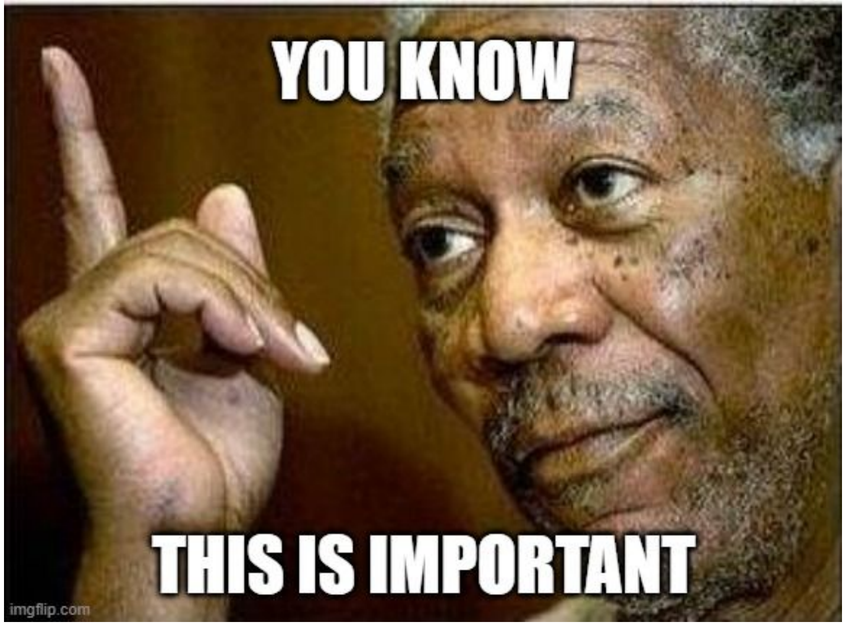
System Design (in the AI world)

Step 1	Problem Formulation
Step 2	Metrics (Offline and Online)
Step 3	Architectural Components (MVP Logic)
Step 4	Data Collection and Preparation
Step 5	Feature Engineering
Step 6	Model Development and Offline Evaluation
Step 7	Prediction Service
Step 8	Online Testing and Deployment
Step 9	Scaling, Monitoring, and Updates

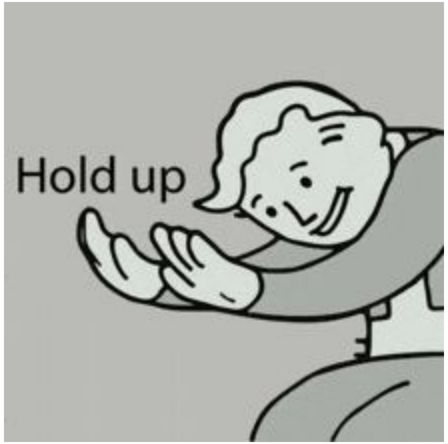
So far you've been doing mostly this.

System Design (in the AI world)

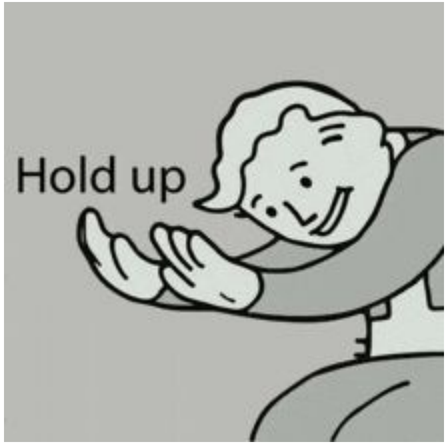
Step 1	Problem Formulation
Step 2	Metrics (Offline and Online)
Step 3	Architectural Components (MVP Logic)
Step 4	Data Collection and Preparation
Step 5	Feature Engineering
Step 6	Model Development and Offline Evaluation
Step 7	Prediction Service
Step 8	Online Testing and Deployment
Step 9	Scaling, Monitoring, and Updates



System Design (in the AI world)



System Design (in the AI world)



I barely know how to *really* program with python

System Design (in the AI world)

Possible (*technical*) ranks within a company

Director
Principal
Staff
Senior
Mid
Junior
Intern

System Design (in the AI world)

Possible (*technical*) ranks within a company

The higher you go, the better you should be at decomposing problems

Director
Principal
Staff
Senior
Mid
Junior
Intern

System Design

Let's build systems

System Design (in the AI world)

Design a machine learning system

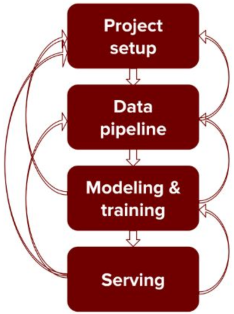
Designing a machine learning system is an iterative process. There are generally four main components of the process: project setup, data pipeline, modeling (selecting, training, and debugging your model), and serving (testing, deploying, maintaining).

The output from one step might be used to update the previous steps. Some scenarios:

- After examining the available data, you realize it's impossible to get the data needed to solve the problem you previously defined, so you have to frame the problem differently.
- After training, you realize that you need more data or need to re-label your data.
- After serving your model to the initial users, you realize that the way they use your product is very different from the assumptions you made when training the model, so you have to update your model.

When asked to design a machine learning system, you need to consider all of these components.

Machine learning project flow



Design a machine learning system

Project setup

Before you even say neural network, you should first figure out as much detail about the problem as possible.

- **Goals**: What do you want to achieve with this problem? For example, if you're asked to create a system to rank what activities to show first in one's newsfeed on Facebook, some of the possible goals are: to minimize the spread of misinformation, to maximize revenue from sponsored content, or to maximize users' engagement.
- **User experience**: Ask your interviewer for a step by step walkthrough of how end users are supposed to use the system. If you're asked to predict what app a phone user wants to use next, you might want to know when and how the predictions are used. Do you only show predictions only when a user unlocks their phone or during the entire time they're on their phone?
- **Performance constraints**: How fast/good does the prediction have to be? What's more important: precision or recall? What's more costly: false negative or false positive? For example, if you build a system to predict whether someone is vulnerable to certain medical problems, your system must not have false negatives. However, if you build a system to predict what word a user will type next on their phone, it doesn't need to be perfect to provide value to users.
- **Evaluation**: How would you evaluate the performance of your system, during both training and inferencing? During inferencing, a system's performance might be inferred from users' reactions, e.g. how many times they choose the system's suggestions. If this metric isn't differentiable, you need another metric to use during training, e.g. the loss function to optimize. Evaluation can be very difficult for generative models. For example, if you're asked to build a dialogue system, how do you evaluate your system's responses?
- **Personalization**: How personalized does your model have to be? Do you need one model for all the users, for a group of users, or for each user individually? If you need multiple models, is it possible to train a base model on all the data and finetune it for each group or each user?
- **Project constraints**: These are the constraints that you have to worry about in the real world but less so during interviews: how much time you have until deployment, how much compute power is available, what kind of talents work on the project, what available systems can be used, etc.

System Design

Let's build systems

“We need to estimate better and faster our tasks to streamline project management” - Your CEO demanded of you.

“I need to know what our customers are saying about our products”

- Your CEO demanded of you

“Mortality rate of preventable diseases is increasing, we need to do something about it” - The president demanded of you.

System Design

Let's build systems

Collaboratively

Don’t be shy.

System Design

Let's build systems

Collaboratively

Don’t be shy.

That's it for today