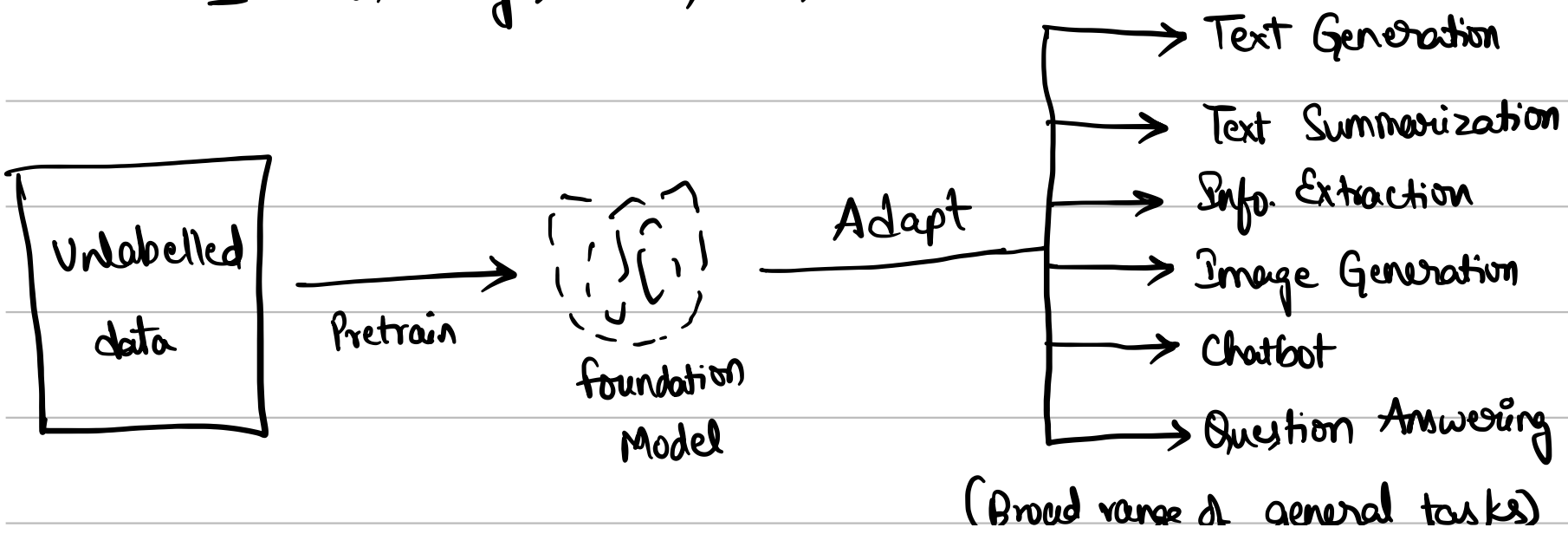
Amazon Bedrock & Generative AI

- **Generative AI (GenAI):**
  - ↳ A subset of Deep Learning
  - ↳ Used to generate new data that is similar to data it's trained on

    Ex: Text, Image, Audio, Code, Video...



- **Foundation Model:**

↳ To generate data, we must rely on foundation model.

↳ wide variety of input data & costs in tens of millions

↳ Some foundation models are: OpenAI, Meta(fb), Amazon, Google, Anthropic

Meta, Google BERT - free open sources

- **Large Language Model(LLM):**

↳ Type of AI designed to generate coherent human-like text.

↳ Trained on large corpus of text data

↳ It has billions of parameters

↳ Can perform translation, summarization, question answering, content creation

↳ We can interact with the LLM by giving a **prompt**.

↳ non-deterministic (generates different text for the same prompt).

- Working of GenAI:

1. for text:

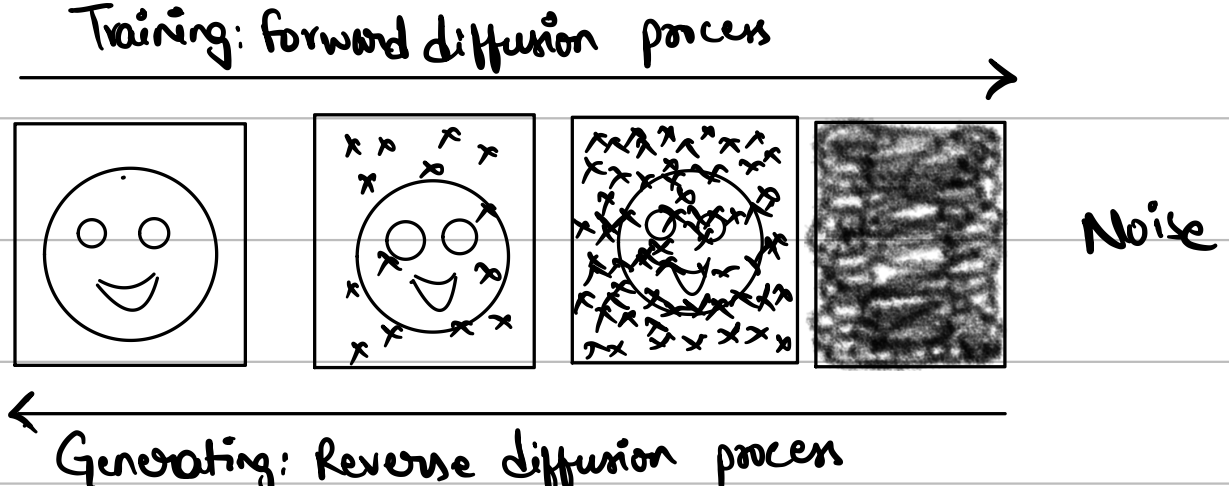
After the rains, roads are ____▽____

flooded (0.4)
blocked (0.25)
wet (0.03)

} one is selected & same process is repeated again & again.

↖ probability

2. for Images



- Amazon Bedrocks:

↳ Build Gen AI applications on AWS

↳ fully managed service & keep control on the training data.

↳ Pay-per-use pricing model & unified APIs

↳ Out-of-the-box features: RAG, LLM Agents....

↳ Security, privacy, governance and responsible AI features.

- foundation Models:

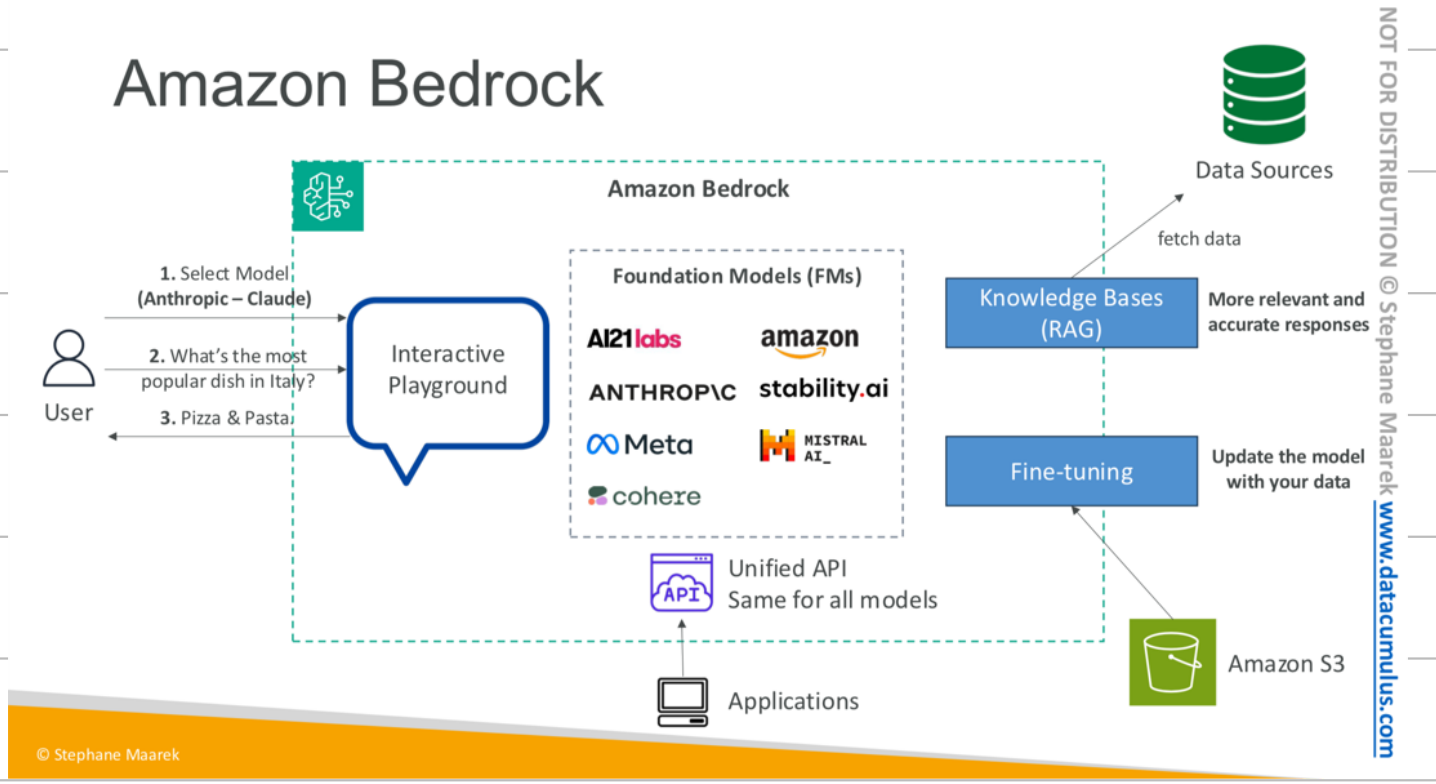
↳ Accès to wide range of them

AI21labs | cohere | stability.ai | amazon

anthropic | Meta | Mistral_ai

↳ Amazon makes copy of FM available only to you, which you can further

fine-tune with your own data.



- for choosing a model, consider

↳ model types, performance requirements, capabilities, constraints, compliance

↳ level of customization, model size, inference options, licensing agreements.

↳ how to get o/p of model

context windows, latency.

↳ multimodal models (varied types of input and outputs)

↳ smaller models are more cost-effective

- Amazon Titan:

↳ high performing FM from AWS.

↳ image, text, multimodal choices via fully-managed APIs

↳ can be customized with our own data

- fine-Tuning a model:

- Adapt a copy of FM with your own data

- You can train the foundational models as well and for doing that the data should be present in Amazon S3 bucket & in specific format

FM → fine-tuned model -- trained only once

FM → Pre-trained model -- trained over multiple times

- You can change the hyperparameters as well.
- The outputs are also stored in S3 bucket
- To write to S3 on our behalf, we have to create a service role in which we give S3 access to bedrock.
- We have to purchase provisioned throughput → different pricing model to use a fine-tuned model
- Not all models can be fine-tuned.
- fine-tuning will change the weight of base foundation model

⚠ Instruction-Based fine tuning:

↳ Improves the performance of a pre-trained FM on domain-specific tasks

↳ further trained on a particular field of area of knowledge

↳ Uses labelled examples that are prompt-response pairs.

{ "prompt" : - - - - - - - - - ,

"completion" : - - - - - - - - - }

→ labelled data

ii) Continued Pre-training:

↳ provide unlabelled data to continue training of FM.

↳ also called domain-adaptation fine tuning (Ex: feed AWS documentation so that it becomes an AWS expert).

↳ Good to feed industry specific terminology into a model.

↳ can continue to train as more data becomes available.

{ "input" : - - - - - }

iii. Single-Turn Messaging:

↳ Similar to working of a chatbot.

↳ part of instruction-based fine-tuning.

(optional) context of conversation

array of msg objects

'user' or 'assistant'

text content of the msg

```
{
  "system": "You are an helpful assistant.",
  "messages": [
    {
      "role": "user",
      "content": "what is AWS"
    },
    {
      "role": "assistant",
      "content": "it's Amazon Web Services."
    }
  ]
}
```

iv. Multi-turn Messaging:

- provide instruction-based fine tuning for a conversation.

- chatbots = multi-turn environment.

```
{
"system": "You are an AI assistant specializing in AWS services.",
 "messages": [
  { "role": "user",      "content": "Tell me about Amazon SageMaker." },
  { "role": "assistant", "content": "Amazon SageMaker is a fully managed service for building, training, and deploying machine learning models at scale." },
  { "role": "user",      "content": "How does it integrate with other AWS services?" },
  { "role": "assistant", "content": "SageMaker integrates with AWS services like S3 for data storage, Lambda for event-driven computing, and CloudWatch for monitoring." }
 ]
}
```

- You must alternate between "user" and "assistant" roles.

- Retraining an FM requires higher budget.

- Instruction based fine-tuning is usually cheaper as computations are less intense and the amount of data required is usually less.

- Prepare the data → do the fine tuning → evaluate the model.

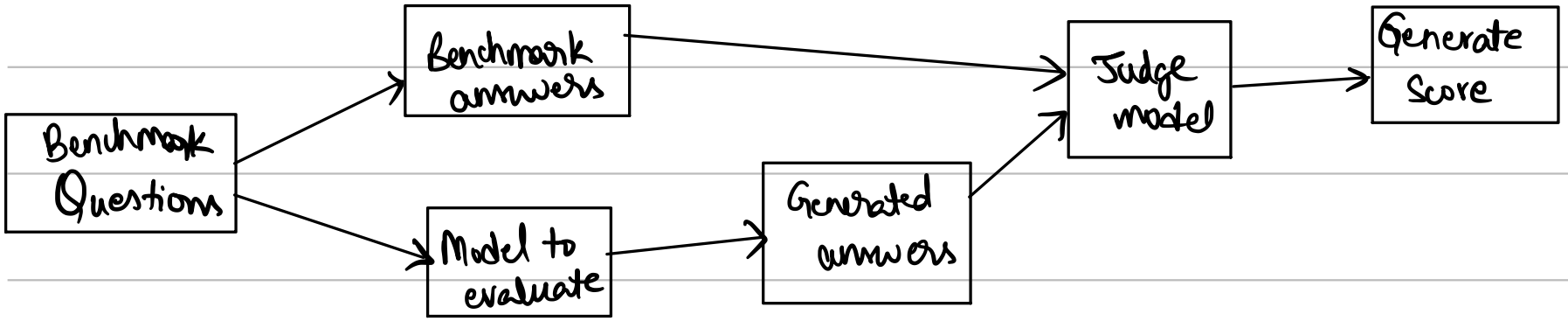
- Running a fine-tuned model is also expensive (provisioned throughput).

  Ex: chatbots designed with particular persona

- training with more up-to-date information
- training with exclusive data
- targeted use cases

- Evaluating a model:

i. Automatic Evaluation:



- Evaluate the model for quality control
- We can use our own prompt dataset & use built-in curated prompt datasets.

- Scores are calculated automatically using various statistical methods.

ii, Human Evaluation:

- Replace the judge model with humans. (employees / subject matter experts (SMEs))
- Define the metrics of evaluation
- Choose built-in tasks / custom tasks.
- Automated metrics for evaluating an FM:

i, ROUGE (Recall-Oriented Understudy for Gisting Evaluation):

↳ Evaluating automatic summarization & machine translating systems.

↳ ROUGE-N: measure the no. of matching n-grams

↳ ROUGE-L: longest common subsequence b/w reference & generated text.

ii, BLEU (Bilingual Evaluation Understudy):

↳ quality evaluation of generated text mainly for translation.

↳ considers both precision & penalizes too much brevity.

↳ looks at the combination of n-grams.

iii, BERT Score:

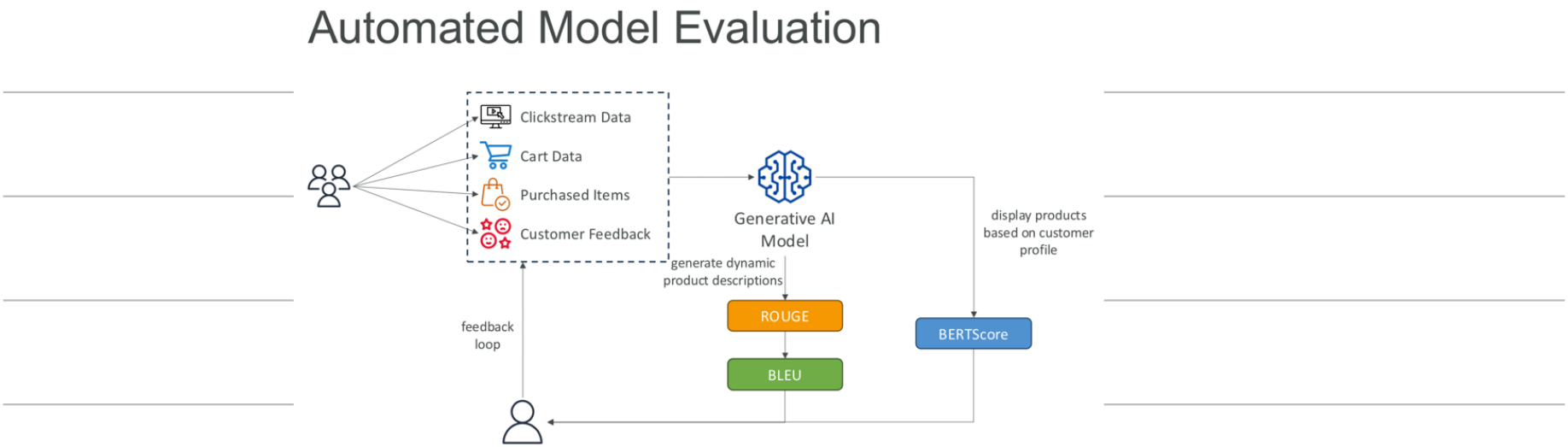
↳ Semantic similarity b/w generated text.

↳ uses pre-trained BERT models to compare contextual embeddings of both texts and computes cosine similarity b/w them.

↳ Capable of capturing more nuance b/w the texts

iv, Perplexity:

↳ how well the model predicts the next token (lower is better)



- Business metrics to evaluate the model:

i, User Satisfaction: gather feedback & access their satisfaction

ii, Average Revenue per User (ARPU): ARPU attributed to Gen-AI app. (more is better)

iii, Cross-Domain Performance: measure model's ability to perform tasks of different domains.

iv. Conversion Rate: generate recommended desired outcomes such as purchases.

v, Efficiency: evaluate model's efficiency in computation, resource utilization etc.