Building an Event Driven Gen-AI powered video summarizer API

Bonjour !

I am Zied Ben Tahar

AWS Community Builder / Principal Architect @ Aviv Group /

You can find me:

Blog: https://zied-ben-tahar.medium.com/
Linkedin: https://www.linkedin.com/in/zied-ben-tahar-5200913/
Twitter : @wow_sig

Agenda

01

GenAI on AWS - A quick overview

02

Let’s build: AI powered Video summarizer

03

Video summarizer - Scaling the architecture with EDA

04

Api-fying the solution

05

Demo Time

01

GenAI on AWS - A quick overview

Amazon Bedrock

Amazon Bedrock

A serverless Gen-AI service

Managed FMs

Access to foundation models from leading partners on AI space

Customizable

Capabilities to adapt and privately customize FMs to domain specific problem

Serverless

No infrastructure to manage,
Good integration with AWS ecosystem

AI21 labs AI21 Labs Jurassic | Amazon Titan | Anthropic's Claude

Cohere Command | Meta Llama 2 | Stability AI SDXL

MISTRAL AI_

Gen AI

Addressing some LLM challenges

How to solve issues related to knowledge cut-off, hallucinations, blackbox ?

➔ **Prompt engineering**
- Optimizing prompts for specific tasks and guiding the model to yield the expected answers

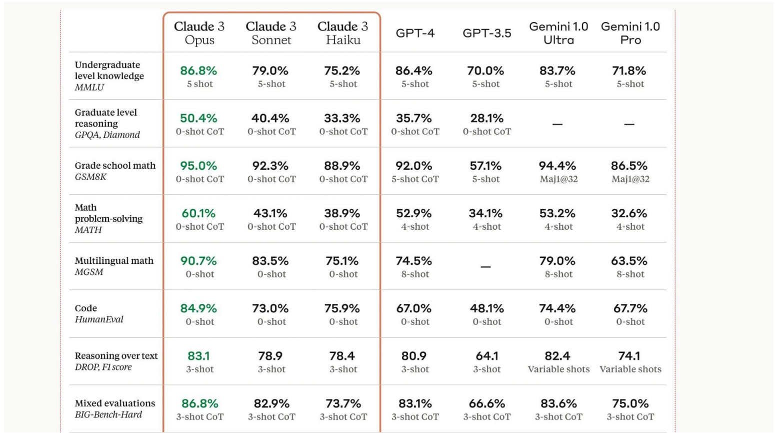
➔ **Retrieval augmented generation (RAG)**
- Allow the model to access to external/up to date information to enhance response.

➔ **Model fine-tuning**
- Additional training of a pre-trained model on task-specific datasets.

Anthropic's Claude ~~(2.1)~~ 3

A serious alternative to GPT 4

- ➔ Handles large context windows, 200K tokens (150k words)
- ➔ Able to maintain factual consistency when processing long prompts
- ➔ Can generate code and structured data
- ➔ Vision capabilities (new with Sonnet)



	Claude 3 Opus	Claude 3 Sonnet	Claude 3 Haiku	GPT-4	GPT-3.5	Gemini 1.0 Ultra	Gemini 1.0 Pro
Undergraduate level knowledge *MMLU*	86.8% 5 shot	79.0% 5-shot	75.2% 5-shot	86.4% 5-shot	70.0% 5-shot	83.7% 5-shot	71.8% 5-shot
Graduate level reasoning *GPQA, Diamond*	50.4% 0-shot CoT	40.4% 0-shot CoT	33.3% 0-shot CoT	35.7% 0-shot CoT	28.1% 0-shot CoT	—	—
Grade school math *GSM8K*	95.0% 0-shot CoT	92.3% 0-shot CoT	88.9% 0-shot CoT	92.0% 5-shot CoT	57.1% 5-shot	94.4% Maj1@32	86.5% Maj1@32
Math problem-solving *MATH*	60.1% 0-shot CoT	43.1% 0-shot CoT	38.9% 0-shot CoT	52.9% 4-shot	34.1% 4-shot	53.2% 4-shot	32.6% 4-shot
Multilingual math *MGSM*	90.7% 0-shot	83.5% 0-shot	75.1% 0-shot	74.5% 8-shot	—	79.0% 8-shot	63.5% 8-shot
Code *HumanEval*	84.9% 0-shot	73.0% 0-shot	75.9% 0-shot	67.0% 0-shot	48.1% 0-shot	74.4% 0-shot	67.7% 0-shot
Reasoning over text *DROP, F1 score*	83.1 3-shot	78.9 3-shot	78.4 3-shot	80.9 3-shot	64.1 3-shot	82.4 Variable shots	74.1 Variable shots
Mixed evaluations *BIG-Bench-Hard*	86.8% 3-shot CoT	82.9% 3-shot CoT	73.7% 3-shot CoT	83.1% 3-shot CoT	66.6% 3-shot CoT	83.6% 3-shot CoT	75.0% 3-shot CoT

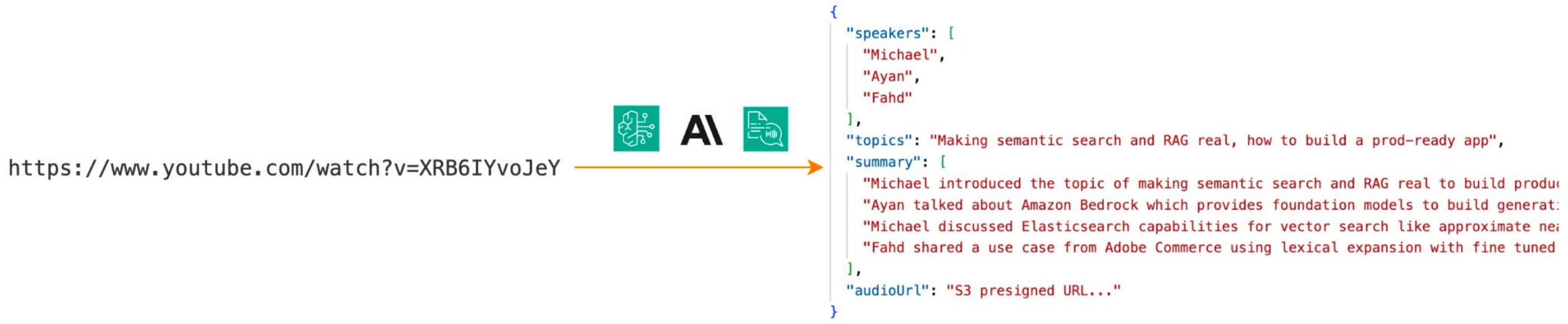
Claude 3 benchmarks

02

Let’s build!
AI powered Video summarizer

Solution overview

Summarization workflow, first attempt



https://www.youtube.com/watch?v=XRB6IYvoJeY

```
{
  "speakers": [
    "Michael",
    "Ayan",
    "Fahd"
  ],
  "topics": "Making semantic search and RAG real, how to build a prod-ready app",
  "summary": [
    "Michael introduced the topic of making semantic search and RAG real to build produ
    "Ayan talked about Amazon Bedrock which provides foundation models to build generat
    "Michael discussed Elasticsearch capabilities for vector search like approximate ne
    "Fahd shared a use case from Adobe Commerce using lexical expansion with fine tuned
  ],
  "audioUrl": "S3 presigned URL..."
}
```

01 Extract/Get Video transcript

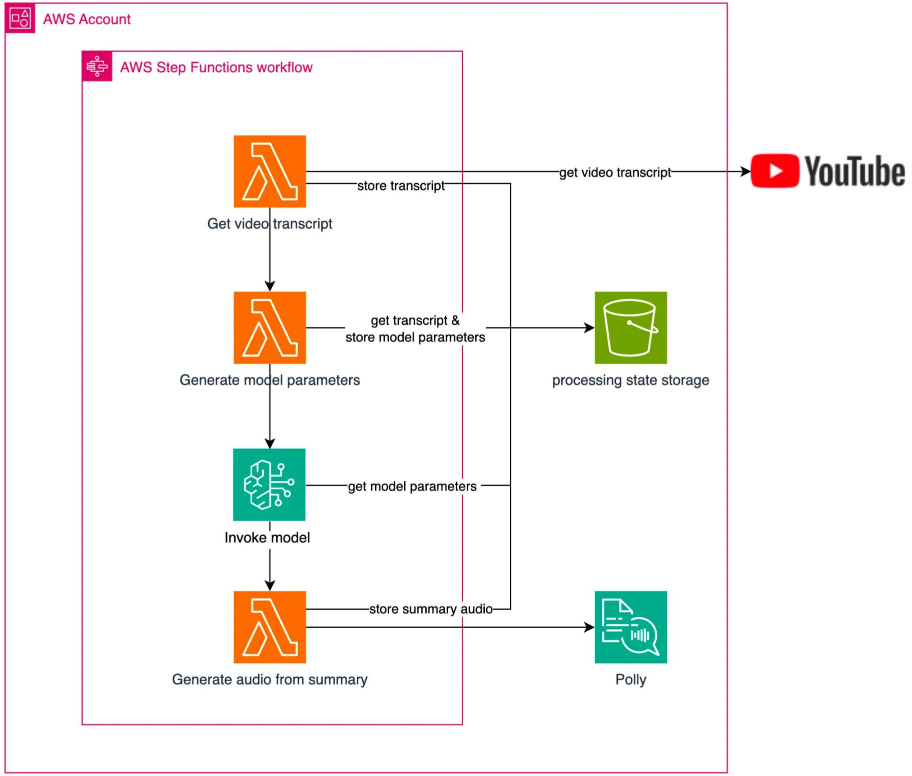
02 Summarize & extract structured data

03 Generate audio from summary

04 Profit !

Solution overview

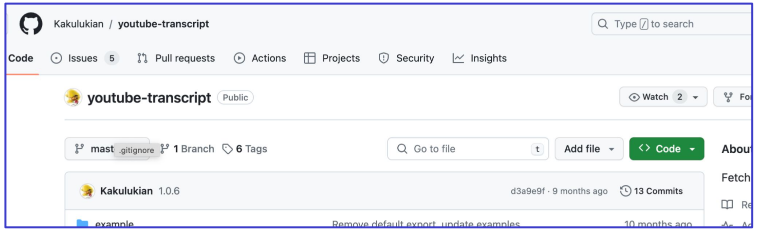
Serverless Summarizer workflow, a first “naïve” approach



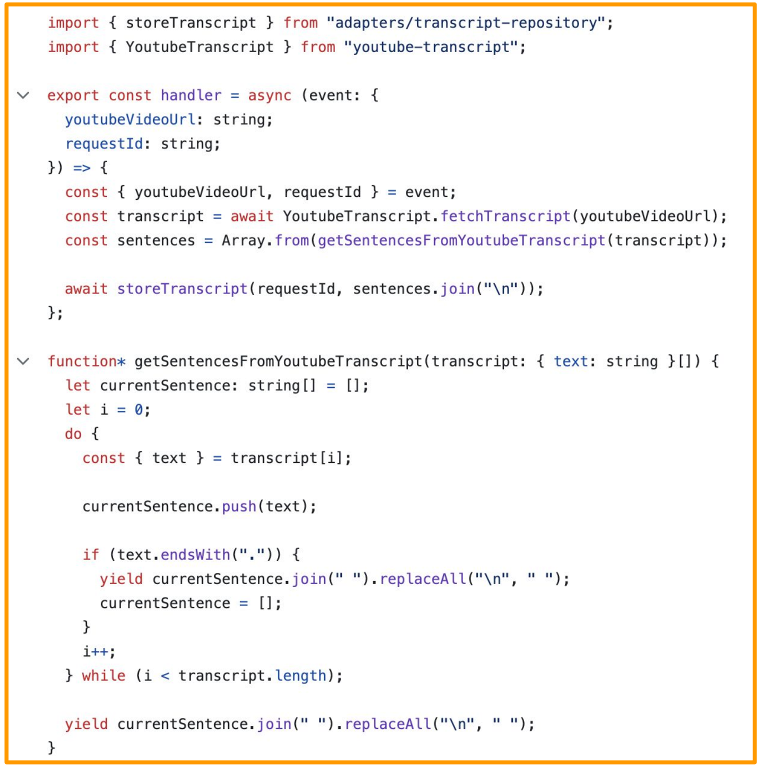
Deep dive

Getting video transcripts

Konami’ing your way into video transcripts



🔗 https://github.com/Kakulukian/youtube-transcript



```
import { storeTranscript } from "adapters/transcript-repository";
import { YoutubeTranscript } from "youtube-transcript";

export const handler = async (event: {
  youtubeVideoUrl: string;
  requestId: string;
}) => {
  const { youtubeVideoUrl, requestId } = event;
  const transcript = await YoutubeTranscript.fetchTranscript(youtubeVideoUrl);
  const sentences = Array.from(getSentencesFromYoutubeTranscript(transcript));

  await storeTranscript(requestId, sentences.join("\n"));
};

function* getSentencesFromYoutubeTranscript(transcript: { text: string }[]) {
  let currentSentence: string[] = [];
  let i = 0;
  do {
    const { text } = transcript[i];

    currentSentence.push(text);

    if (text.endsWith(".")) {
      yield currentSentence.join(" ").replaceAll("\n", " ");
      currentSentence = [];
    }
    i++;
  } while (i < transcript.length);

  yield currentSentence.join(" ").replaceAll("\n", " ");
}
```

~ Fast, as it retrieves already generated transcripts from youtube
~ ...but it’s brittle, based on a non official youtube API, does not work on some languages

Deep dive

Prompting techniques with Claude

```
You are a video transcript summarizer.
Here is a video transcription that you must summarize:

<transcript>{{transcript}}</transcript>

Summarize this transcript in a third person point of view in 10 sentences.
Identify the speakers and the main topics of the transcript and add them in
the output as well.
Do not add or invent speaker names if you not able to identify them.
You must output the summary JSON format conforming to this JSON schema:
{
 "type": "object",
 "properties": {
   "speakers": {
     "type": "array",
     "items": {
       "type": "string"
     }
   },
   "topics": {
     "type": "string"
   },
   "summary": {
     "type": "array",
     "items": {
       "type": "string"
     }
   }
 }
}
```

- ➔ Clear without ambiguity, no room for assumptions
- ➔ Don’t be polite, be straightforward
- ➔ Concrete example/ schema for structured outputs

Deep dive

Putting words in Claude's mouth

For structured output, we'll need to give hints to the Assistant by prefilling its response

With Claude's Completion API (Legacy)

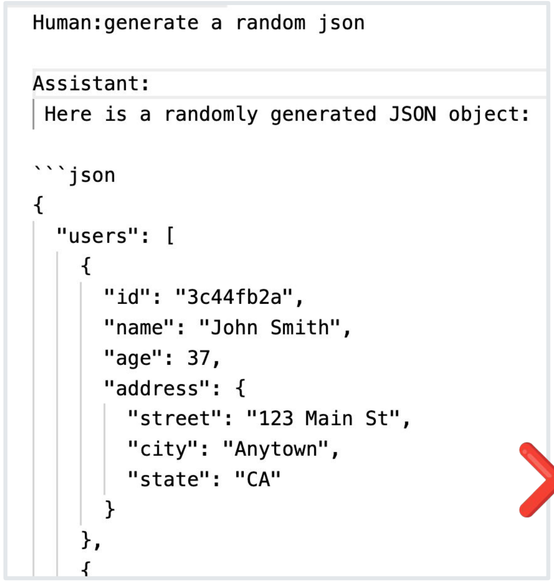
```
\n\nHuman:{{prompt}}\n\nAssistant:{
```

Message API

```
{
    "role": "user",
    "content": "{{Prompt}}"
},
{
    "role": "assistant",
    "content": "{"
}
```

Deep dive

Putting words in Claude's mouth

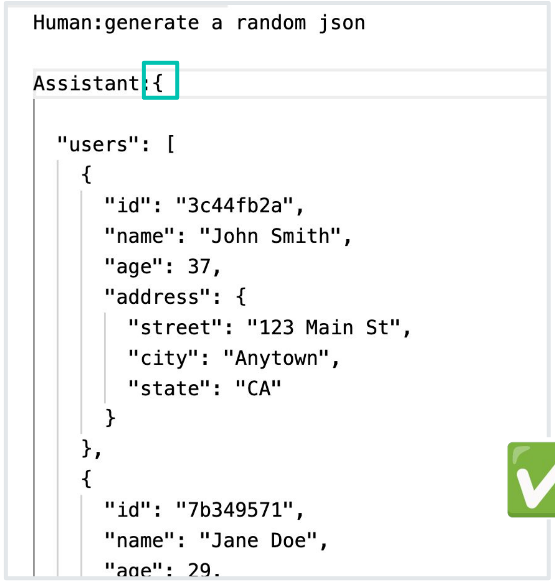


~~~
Human:generate a random json

Assistant:
Here is a randomly generated JSON object:

```json
{
  "users": [
    {
      "id": "3c44fb2a",
      "name": "John Smith",
      "age": 37,
      "address": {
        "street": "123 Main St",
        "city": "Anytown",
        "state": "CA"
      }
    },
    {
~~~

❌



```
Human:generate a random json

Assistant:{

  "users": [
    {
      "id": "3c44fb2a",
      "name": "John Smith",
      "age": 37,
      "address": {
        "street": "123 Main St",
        "city": "Anytown",
        "state": "CA"
      }
    },
    {
      "id": "7b349571",
      "name": "Jane Doe",
      "age": 29,
```

✅

Deep dive

Invoking the model - state machine definition

```
  new LambdaInvoke(this, "generate-model-parameters", {
    lambdaFunction: generateModelParameters,
    payload: TaskInput.fromObject({
      "requestId.$": "$$.Execution.Name",
    }),
  }).addCatch(failState)
)
.next(
  new CustomState(this, "bedrock-invoke-model", {
    stateJson: {
      Type: "Task",
      Resource: "arn:aws:states:::bedrock:invokeModel",
      Parameters: {
        ModelId: "anthropic.claude-v2:1",
        Input: {
          "S3Uri.$": `$.Payload.modelParameters`,
        },
        ContentType: "application/json",
      },
      ResultSelector: {
        "requestId.$": "$$.Execution.Name",
        "summaryTaskResult.$":
          "States.StringToJson(States.Format('\\{{}', $.Body.completion))",
      },
    },
  })
    .addCatch(failState)
    .next(
      new LambdaInvoke(this, "generate-audio-from-summary", {
        lambdaFunction: generateAudioFromSummary,
      }).addCatch(failState)
    )
```

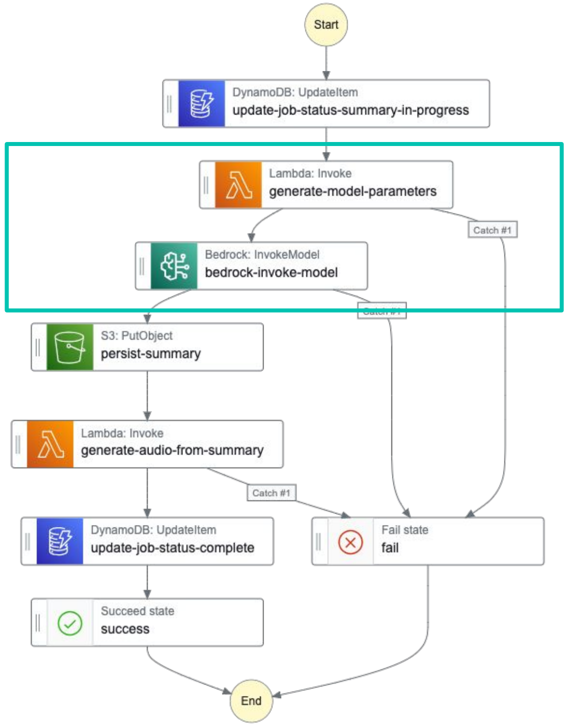
1- Generate model parameters and write to an s3 bucket

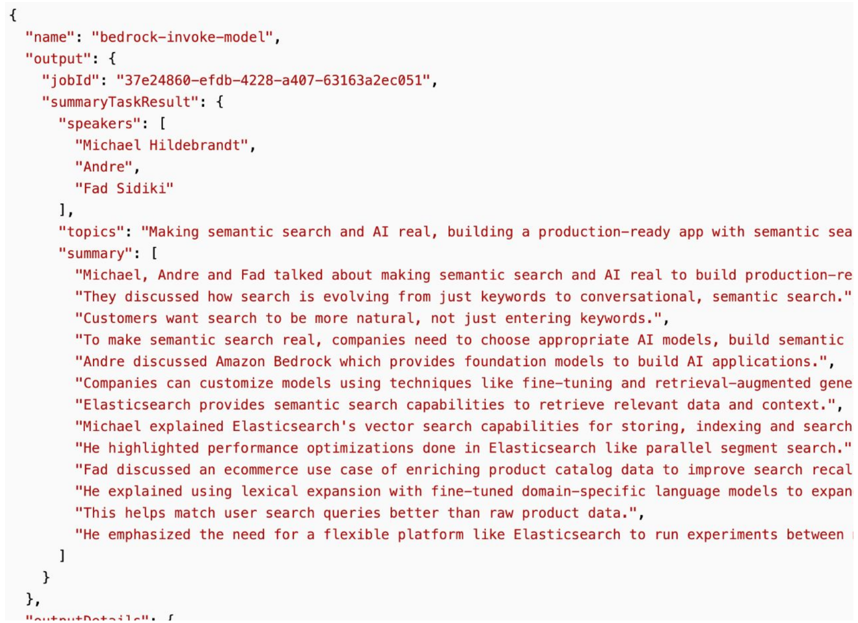
2- Direct invoke to claude model

3- Making sure that JSON is well formed

Deep diving

Invoking the model



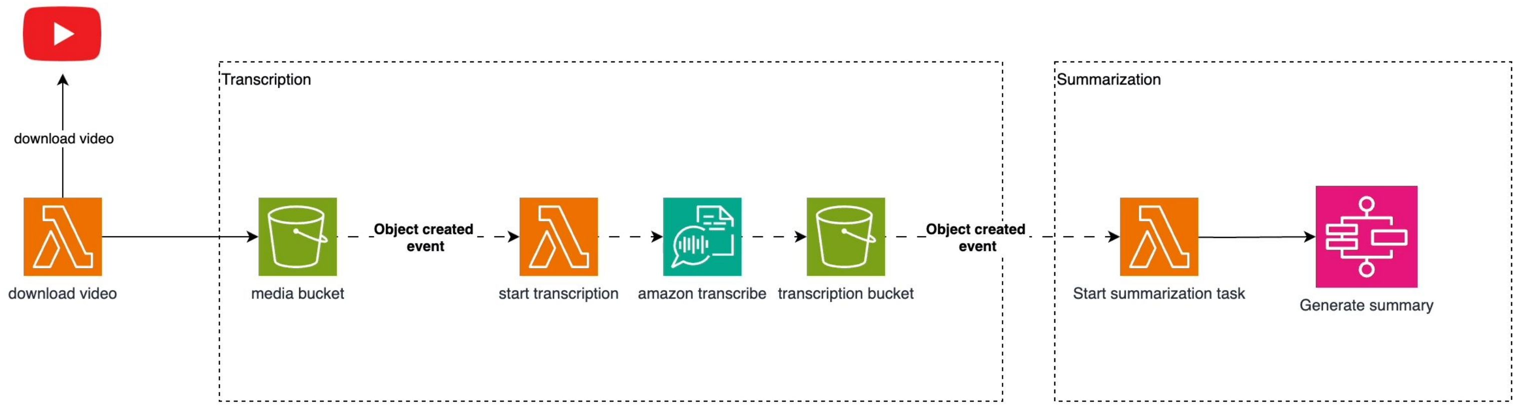


```
{
  "name": "bedrock-invoke-model",
  "output": {
    "jobId": "37e24860-efdb-4228-a407-63163a2ec051",
    "summaryTaskResult": {
      "speakers": [
        "Michael Hildebrandt",
        "Andre",
        "Fad Sidiki"
      ],
      "topics": "Making semantic search and AI real, building a production-ready app with semantic sea
      "summary": [
        "Michael, Andre and Fad talked about making semantic search and AI real to build production-re
        "They discussed how search is evolving from just keywords to conversational, semantic search."
        "Customers want search to be more natural, not just entering keywords.",
        "To make semantic search real, companies need to choose appropriate AI models, build semantic
        "Andre discussed Amazon Bedrock which provides foundation models to build AI applications.",
        "Companies can customize models using techniques like fine-tuning and retrieval-augmented gene
        "Elasticsearch provides semantic search capabilities to retrieve relevant data and context.",
        "Michael explained Elasticsearch's vector search capabilities for storing, indexing and search
        "He highlighted performance optimizations done in Elasticsearch like parallel segment search."
        "Fad discussed an ecommerce use case of enriching product catalog data to improve search recal
        "He explained using lexical expansion with fine-tuned domain-specific language models to expan
        "This helps match user search queries better than raw product data.",
        "He emphasized the need for a flexible platform like Elasticsearch to run experiments between
      ]
    }
  },
  "outputDetails": {
```

Can we do better ?

A better solution

Enters Amazon Transcribe



- A more stable solution that supports many languages and more media types
- Some important architectural side effects: The system is now **event based** and **asynchronous**

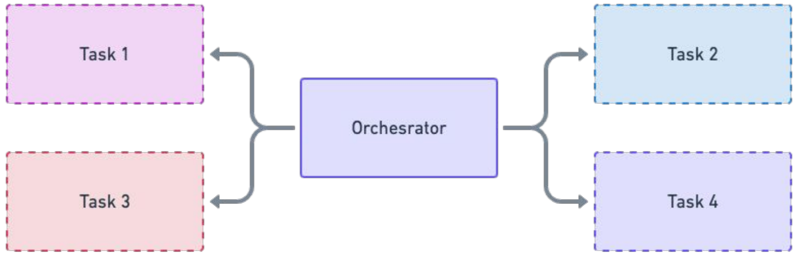
This is nice…but how can I scale my architecture ?

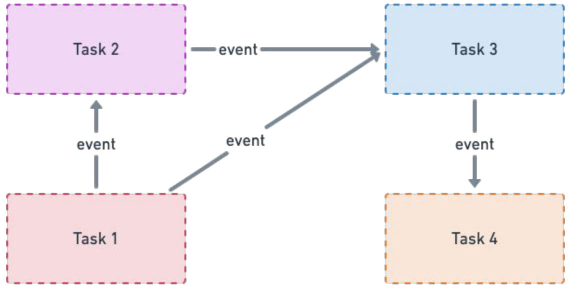
03

Scaling the architecture with EDA

Scaling the architecture

Orchestration or Choreography ?



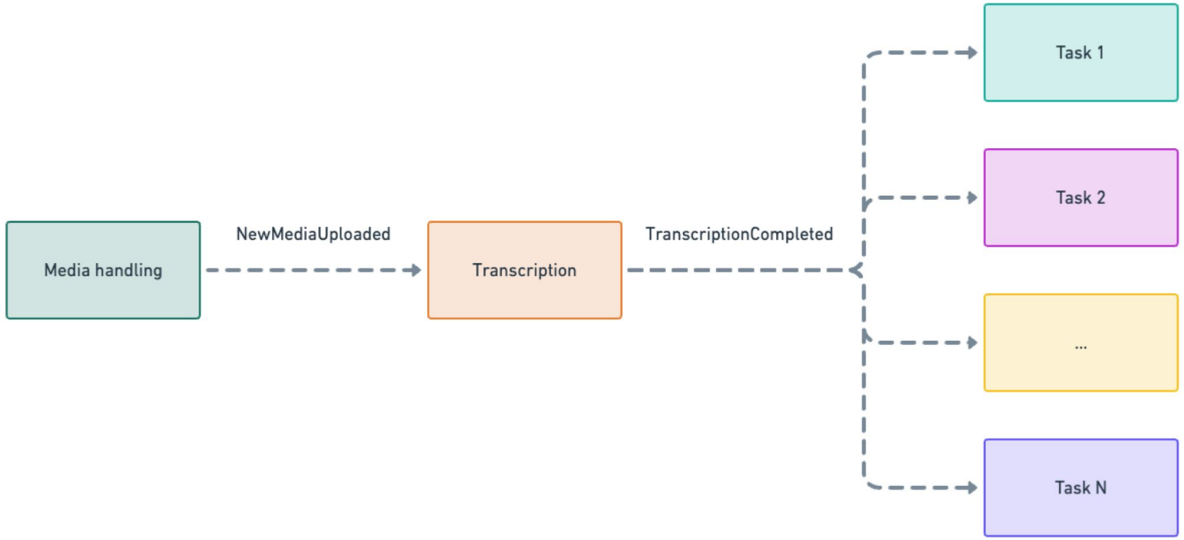


~ Orchestration: Greater control over complex workflows. Complex to evolve when collaborating with external contexts

~ Choreography: Flexible and and easier to extend but harder to Observe

Which pattern works best ?

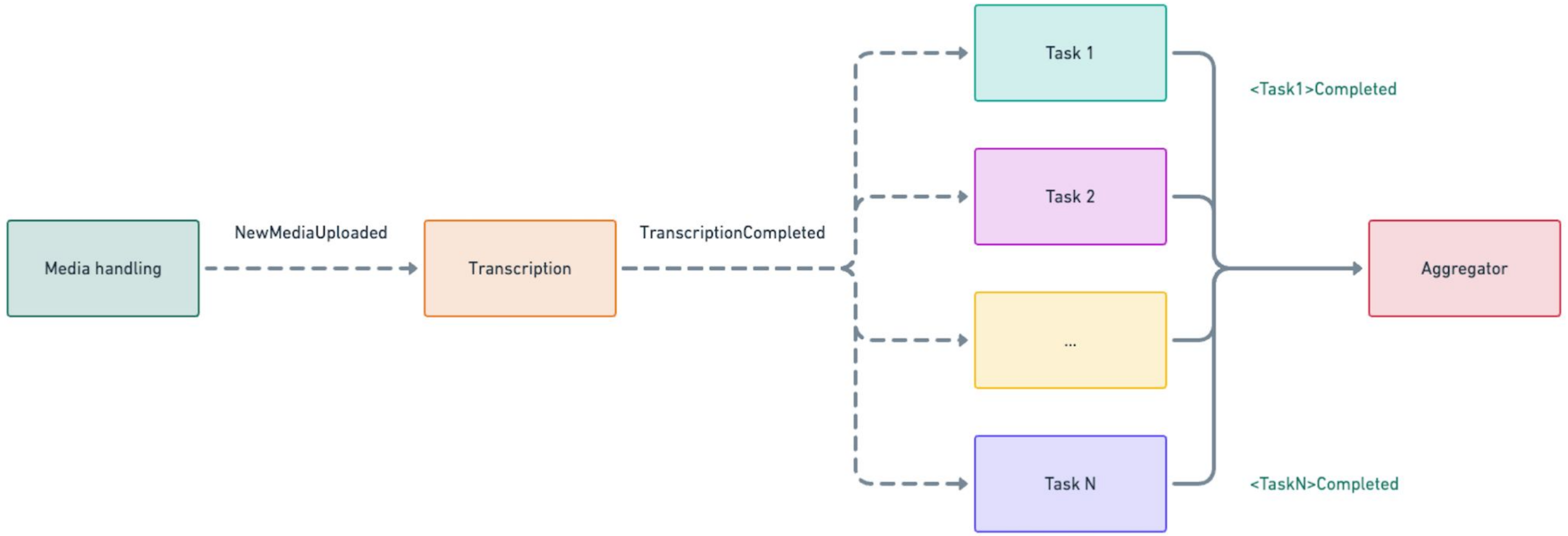
Scaling the architecture
Let’s try with Choreography



- Extensible: Tasks can be added or removed without impacting the entire system
- Resilient: If a task fails, it does not impact the entire application

Scaling the architecture

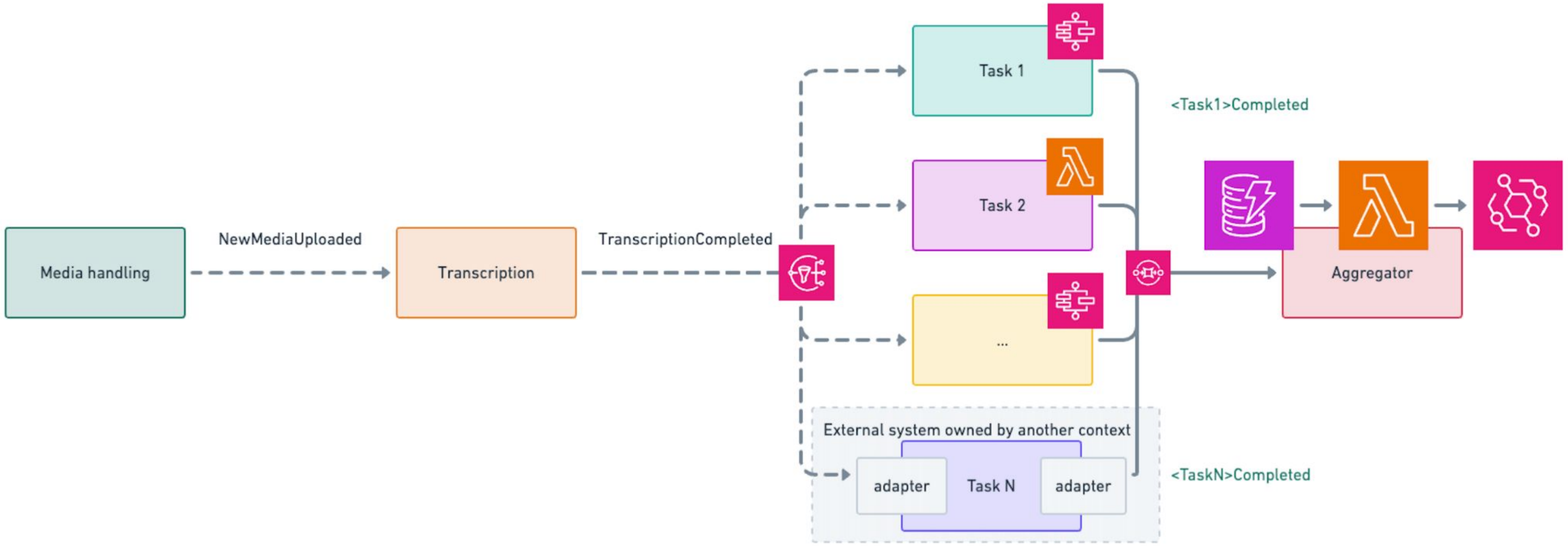
Scatter-gather pattern



- Aggregator: A central component handling the events from the scatter phase.

Scaling the architecture

Scatter-gather pattern



…Yes, but

Scaling the architecture

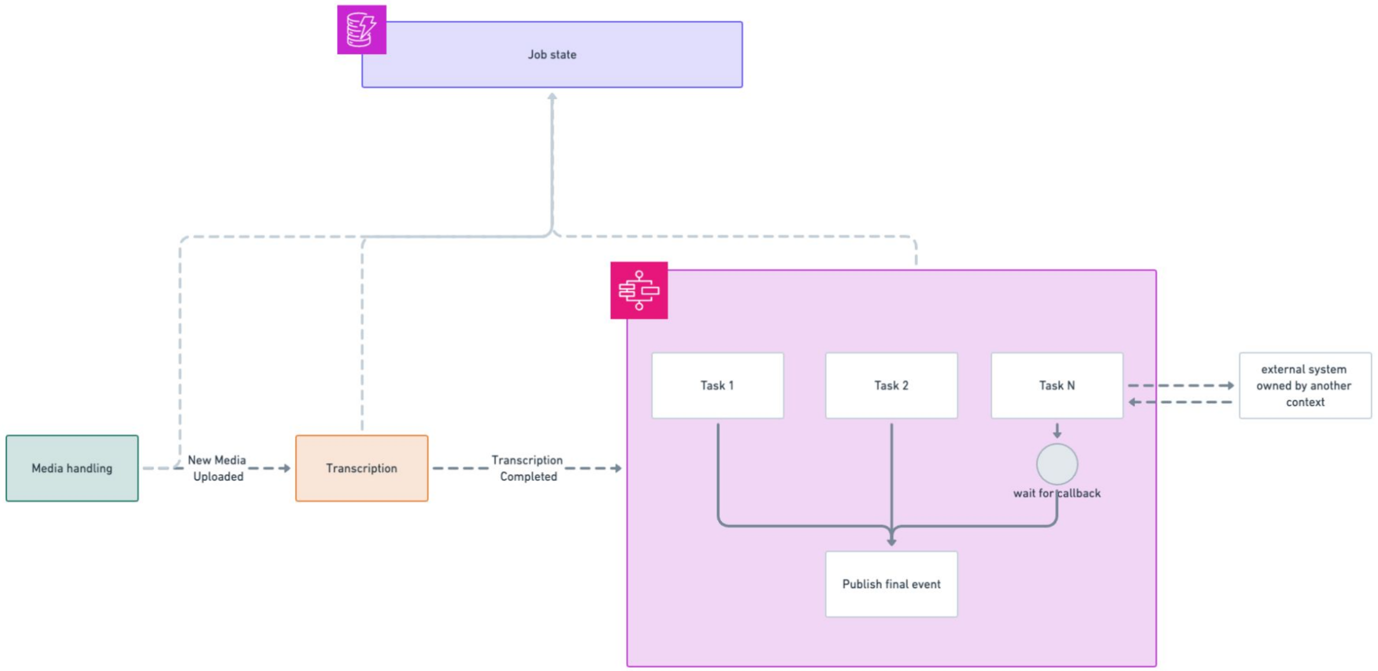
Aggregator pattern can be hard to tame

- What if…
  - a task fails
  - or never publishes the expected event
- How to make sure that failure is handled properly ?
- How to monitor the overall progress of a given Job ?
- How to make sure that my system is observed properly

So...Orchestration or Choreography ?

Scaling the architecture

Orchestration or Choreography...Why not both ?



04

Apify-ing the solution

APIs, like diamonds, are forever

Joshua Bloch

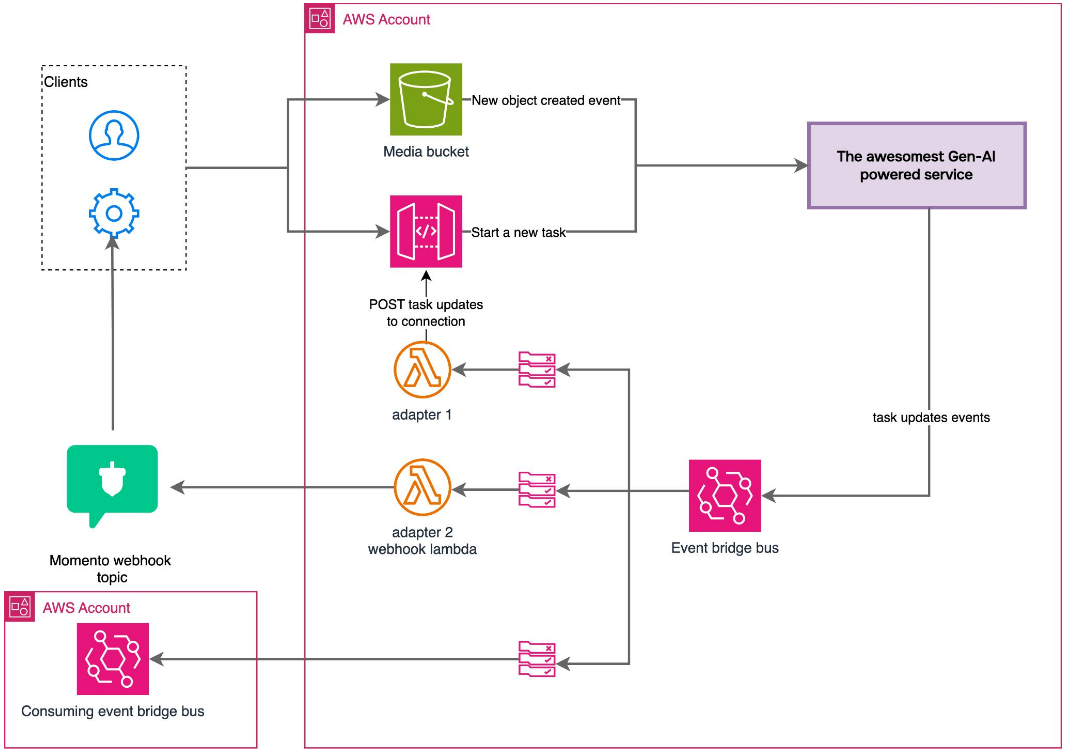
Apify-ing the solution

Principles on designing good “public” Event-Driven APIs

- Be **conservative** on what you send (Postel’s Law)
- Be mindful of the **observed behavior** of your APIs (Hyrum’s law)
- Make clear distinction between **your private** and **public events**
- Adopt a **sane** versioning strategy
- Adopt standards: **CloudEvents** & **AsyncAPI**

Apify-ing the solution

The awesomest GenAI powered video summarizer



05

Demo

Thanks !

Questions ?

Source code👇