SPRING 2024

DSE I2450 - 3GG

BIG Data &

Scalable Computation

Week 6

Data in the Cloud

Dall-E 3: “inside a big data system, use your imagination”

plan for today!

- **housekeeping, logistics**
- **lecture + demo: "the Cloud"**
- **(break)**
- **group work**
- **back together, until 7:20 👋**

🧹 housekeeping:

- 📚 **abstract + bibliography due tonight! 11:59pm**
- 📖 feedback on bibliography, as well as midterm grade, back from me by next week
- 📆 schedule groups for symposium: I will do this randomly via Discord! **let me know ASAP** if you have a conflict with 1 of the days (April 10th/17th) - 1 month away
- 📄 will share more guidelines and examples for symposium presentation, but know that you will have **30 min.**

🧹 housekeeping:

- 🖥️ **data center tour = on!**
  - John Licci, Director Data Operations @ CUNY (we will see CUNYFirst, clean room, racks, etc.)
  - “private cloud”

- **March 27th:**
  - anybody **cannot** make 3:30pm?
  - anybody **cannot** make 4:00pm?

- **March 25th**, backup - 3:30pm?

what is … the CLOUD?

- “The illusion of infinite computing resources available on demand” - Berkeley, 2009

- “Delivery of software, processing power, and storage over the internet” - NASA, 2009

- “Cloud … is the delivery of computing resources as services, meaning that the resources are owned and managed by the cloud provider rather than the end user.” - Digital Ocean, 2024

- “A model for enabling ubiquitous, convenient, on-demand network access to a shared pool of configurable computing resources (e.g., networks, servers, storage, applications, and services) that can be rapidly provisioned and released with minimal management effort or service provider interaction.”

- NIST, National Institute of Standards and Technology, US Dept. of Commerce

HISTORY:

- Resource pooling, remote jobs as old as 1960’s

image source: [computer history museum]

- Compatible Time-Sharing System (CTSS), MIT, IBM Mainframes
- In 1964: “probably more than 20k computers in use within the US" - Atlantic
- First use of “cloud” in 1994, marketing

☁️ = **<u>hardware</u>**-driven:

"The construction and operation of extremely large-scale, commodity-computer datacenter at low-cost locations was the key necessary enabler of cloud computing" - Berkeley (2009)

This scale + building happened rapidly …

Google Docs: 2006
2024: … what *isn't* the cloud?

If it’s in the cloud, it’s on someone else’s computer - and you’re renting it

☁️ = **hardware**-driven:

“...the ethereal metaphor of ‘the cloud’ for offsite data management and processing is in **complete contradiction with the physical realities** of the extraction of minerals from the Earth’s crust and dispossession of human populations that sustain its existence.” Mosco via Crawford, 2018

some characteristics of the cloud:

- highly, highly scalable - scale *out*
- pay-as-you-go, highly granular
- “on-demand,” no need to build, hire, manage
- little upfront commitment, low barrier to entry for making things (startups, etc.)
- hardware increasingly managed by BIG players, “economy of scale”


West Central US
The Dalles, OR
West US 2
Oregon
N. California
West US
US Gov Iowa
Central US
Council Bluffs, IA
Canada Central
Canada East
North Central US
AWS GovCloud
South Central US
Ohio
Berkeley County, SC
N. Virginia
East US
East US 2
US Gov Virginia
Ireland
North Europe
UK West
UK South
West Europe
Germany Northeast
Frankfurt
Germany Central
St. Ghislain, Belgium
China North
Beijing
Seoul
Japan East
Tokyo
Tokyo, Japan
Japan West
China East
East Asia
Changhua County, Taiwan
West India
Central India
Mumbai
South India
Southeast Asia
Singapore
São Paulo
Brasil South
Australia East
Sydney
Australia Southeast
Microsoft Azure
amazon web services
Google Cloud Platform
Atomia

abstraction, revisited

- hiding complexity, or things the programmer/engineer does not need to deal with

- Spark provides **abstraction** with RDDs, we don't have to manage the cluster or program partitions; HDFS as well

- less complexity also = less **CONTROL**

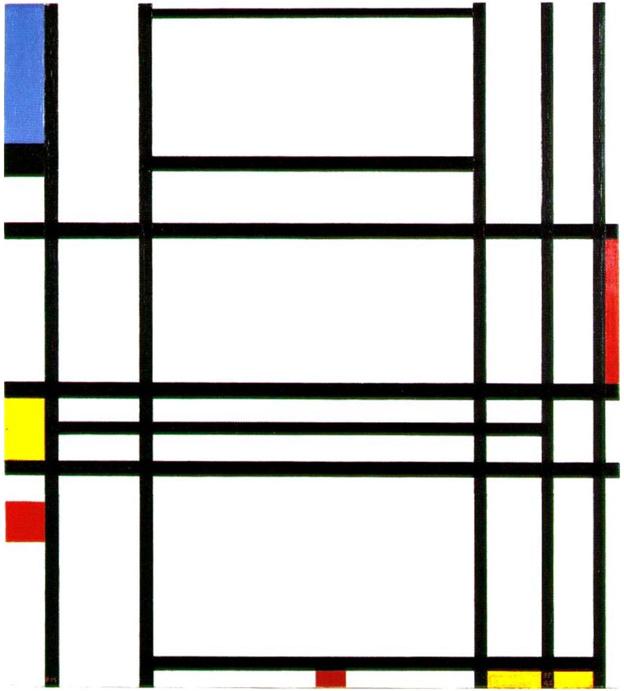


image source: Mondrian, wikimedia

Levels of abstraction in cloud services:

- IaaS, PaaS, SaaS, etc.

Levels of abstraction in cloud services:

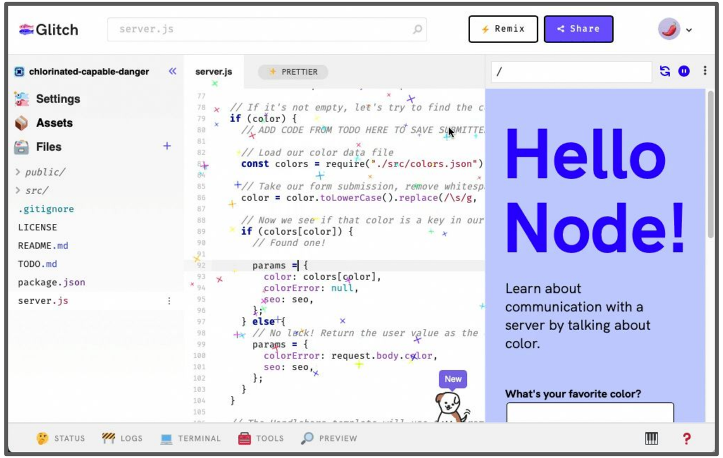
IaaS: **infrastructure** as a service

- virtual machine (VM)
- can install your own software on it, access shell
- Amazon EC2 (2006); Google Compute; Digital Ocean droplet
- server, network, hardware, etc. all handled by provider
- **LOW ABSTRACTION = HIGH CONTROL**

Levels of abstraction in cloud services:

PaaS: **platform** as a service

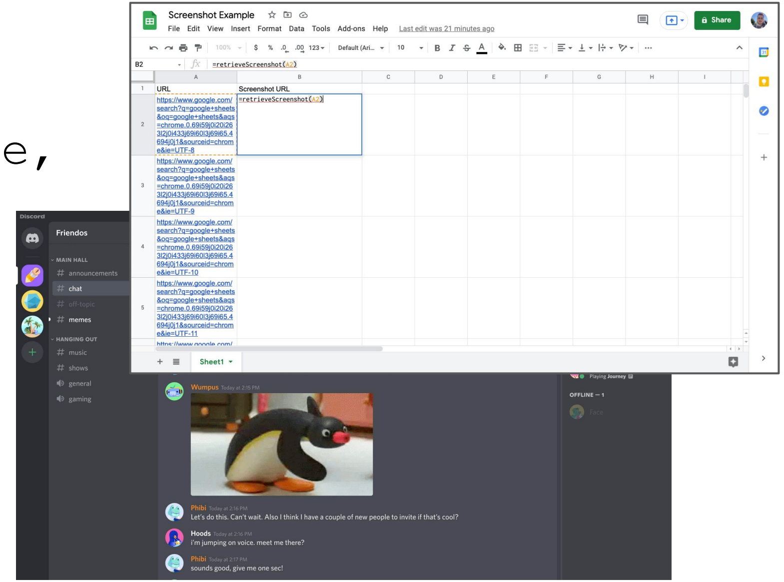
- more software installed, still the ability to code + customize
- goal = not to customize infrastructure, but build on top
- Glitch.com, Heroku, Databricks
- **MID abstraction, MID control**



Levels of abstraction in cloud services:

SaaS: **software** as a service

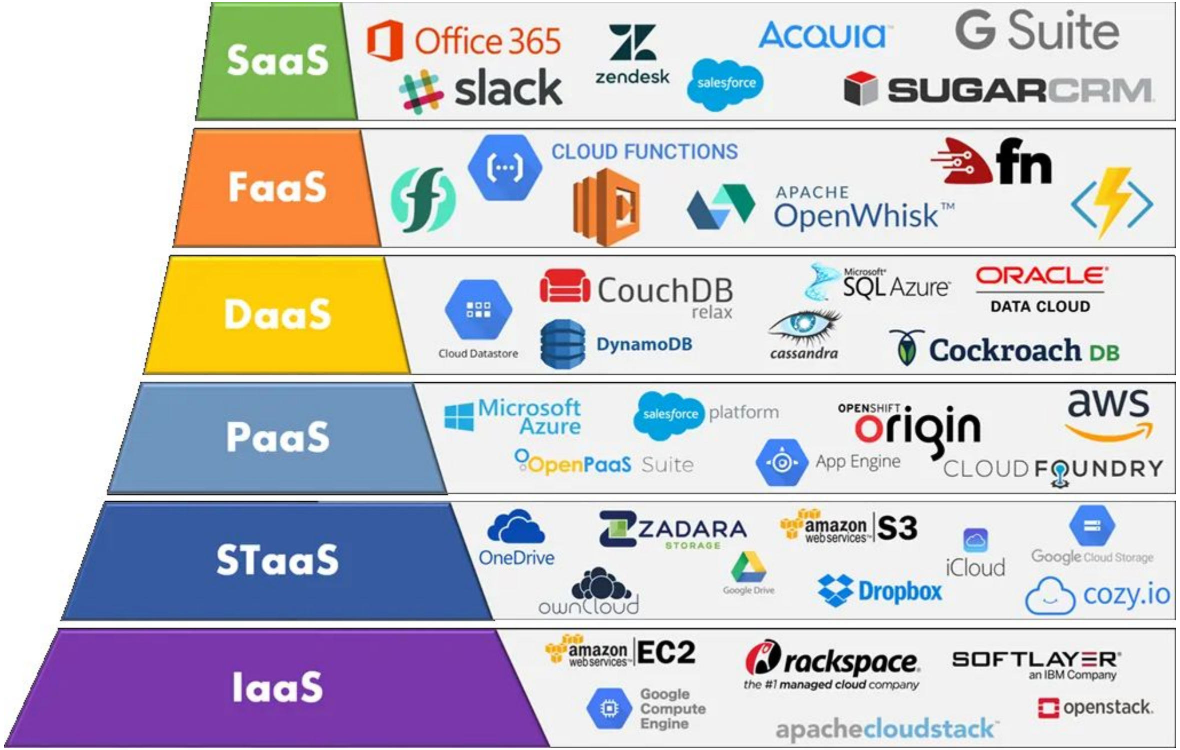
- Google Sheets, GDrive, Airtable, Gmail, Slack, Discord, Salesforce, Zoom, Dropbox … Netflix …
- So many more …
- **HIGH ABSTRACTION = LOW CONTROL**



Levels of abstraction in cloud services:

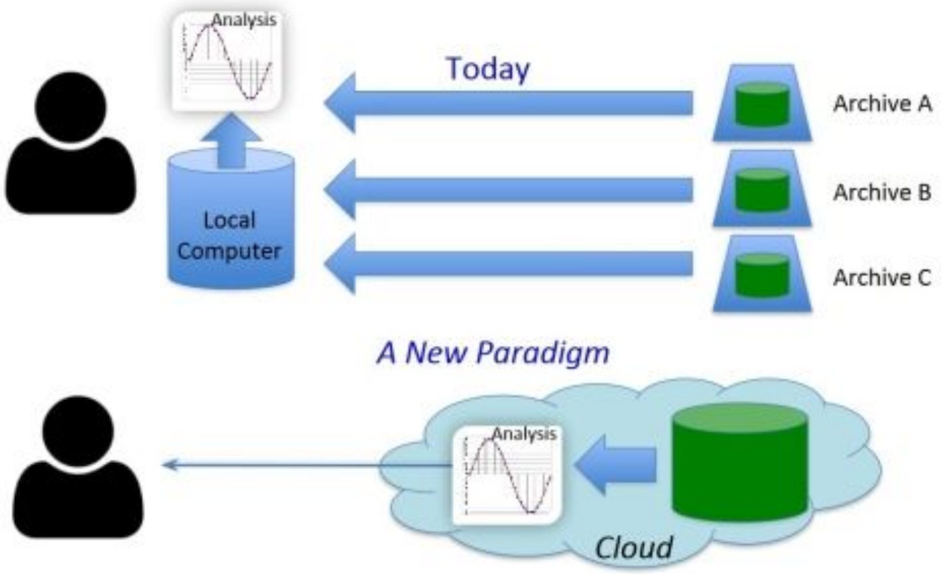
- hybrid; not cut on dry - no one agrees

- BaaS, CPaaS, **FaaS** … “x as a service”



NASA example: Nebula, 2009

- "Big data was the main reason to move to the cloud"



NASA example: Nebula, 2009

- Started building their own, OpenStack ..

- By 2012, NASA uses AWS

“cloud” might be right

- Transfer of risk
- Hardware, data failure?
  ON THEM (provider)
- Scalability: including low risk of scaling DOWN
- Cost: compute + storage might be cheaper

cloud “migration” can be a challenge

- Trust
- Change in org, training
- Proprietary providers
- Data confidentiality, security, persistence
- Data audits - regional issues!
- Vendor or data “lock-in”
- Cost … @ WHAT COST? SHEIN, TEMU

🎬 video from USGS, 2020:

https://www.usgs.gov/media/videos/landsat-data-cloud

(re: migration of LANDSAT imagery)

“cost associativity”

- Embarrassingly parallel
- Say it takes 1 machine 1000 hours to perform my task
- It takes 1000 machines 1 hour to perform my task
- It takes 2000 machines 30 minutes
- “unique opportunity for batch-processing and analytics jobs that analyze terabytes of data and can take hours to finish” - Berkeley, 2009

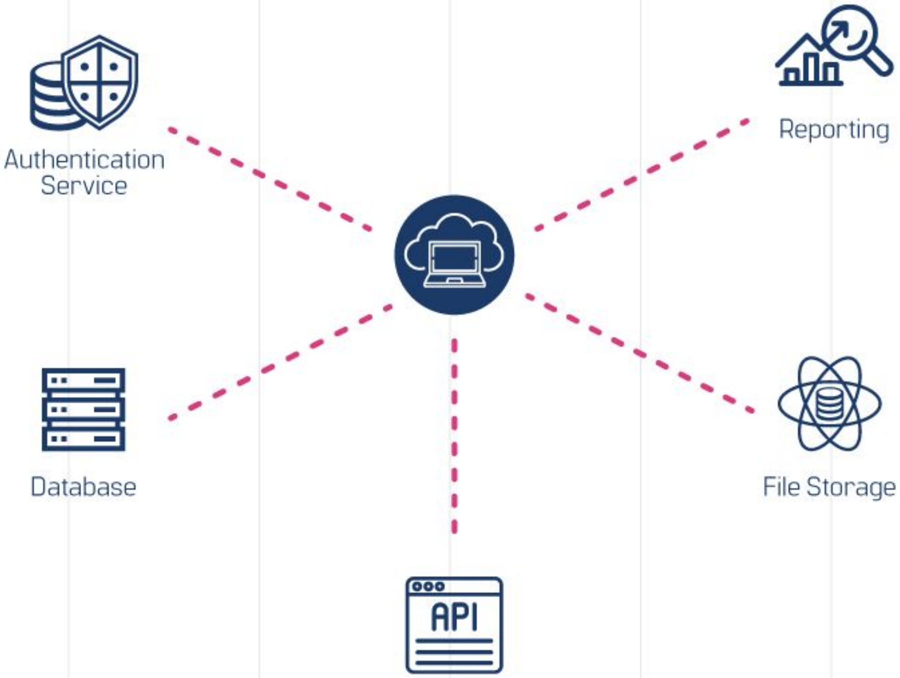
“pay-as-you-go”

- Down to the microsecond, or per invocation, or per bandwidth, etc.
- VERY DIFFERENT: don’t pay when you are not using
- Resources can be idle at off-peak times - examples?
- “Real world estimation of data center usage is 5% - 20%” - Berkeley 2009

BIG DATA?

- Batch tasks + parallelization
- **DISAGGREGATE STORAGE + COMPUTE, very different than HDFS/SPARK!**
- Both can scale up separately based on your needs
- Also can imagine other pipelines besides batch processing: streaming, event-driven, “on-the-fly” analysis

SERVERLESS ARCHITECTURE



1 example: **FaaS**, serverless

- Of course there are still servers (like … cloud), I just don't deal with them

- **unit of abstraction = FUNCTION**, 1 function in code, modular

- provider gives me coding environment, language installed, etc.

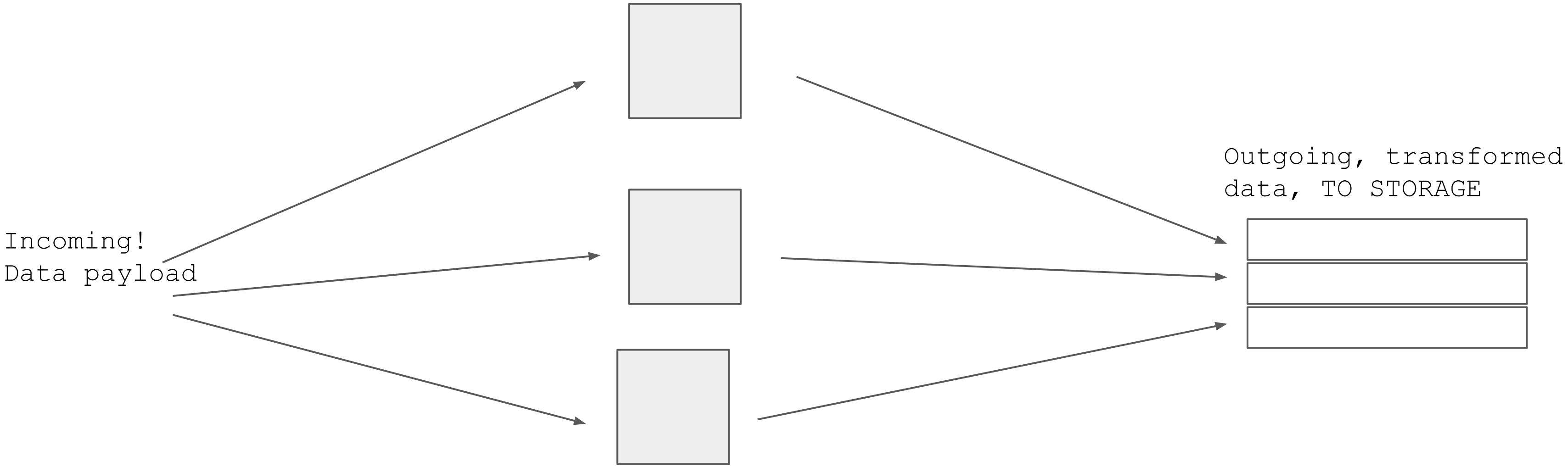
1 example: **<u>FaaS</u>**, serverless

- function "deployed," waiting for my signal to run (trigger, event, invocation)

- **stateless**, it remembers + stores nothing, only receives input + sends output

1 example: **<u>FaaS</u>**, serverless

- O, and it **<u>SCALES,</u>** automatically - “function-level parallelism”
- Functions on many many machines - all over world ⛅🌎



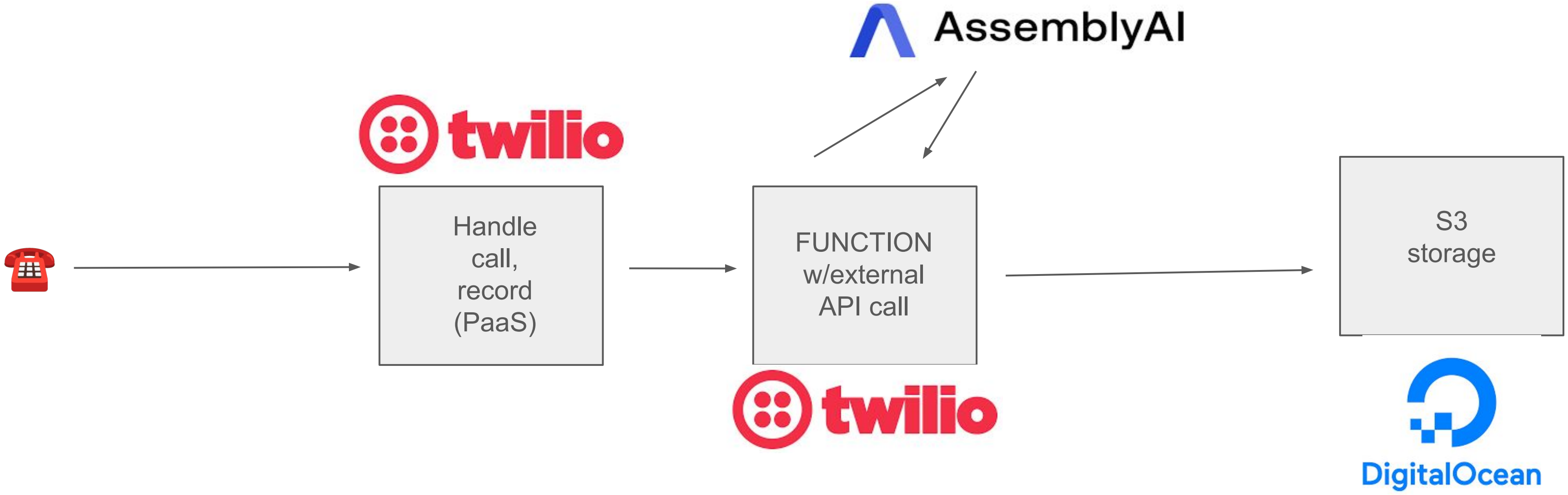
<u>DEMO: serverless data pipeline</u>

- ☎️ Imagine a call center for a business - “this call is being monitored and recorded” - or a crowd-sourced research project gathering voicemails, etc.

- Each call comes in via a cloud service (Twilio) - recorded and store in cloud, generates **a lot of data/metadata**

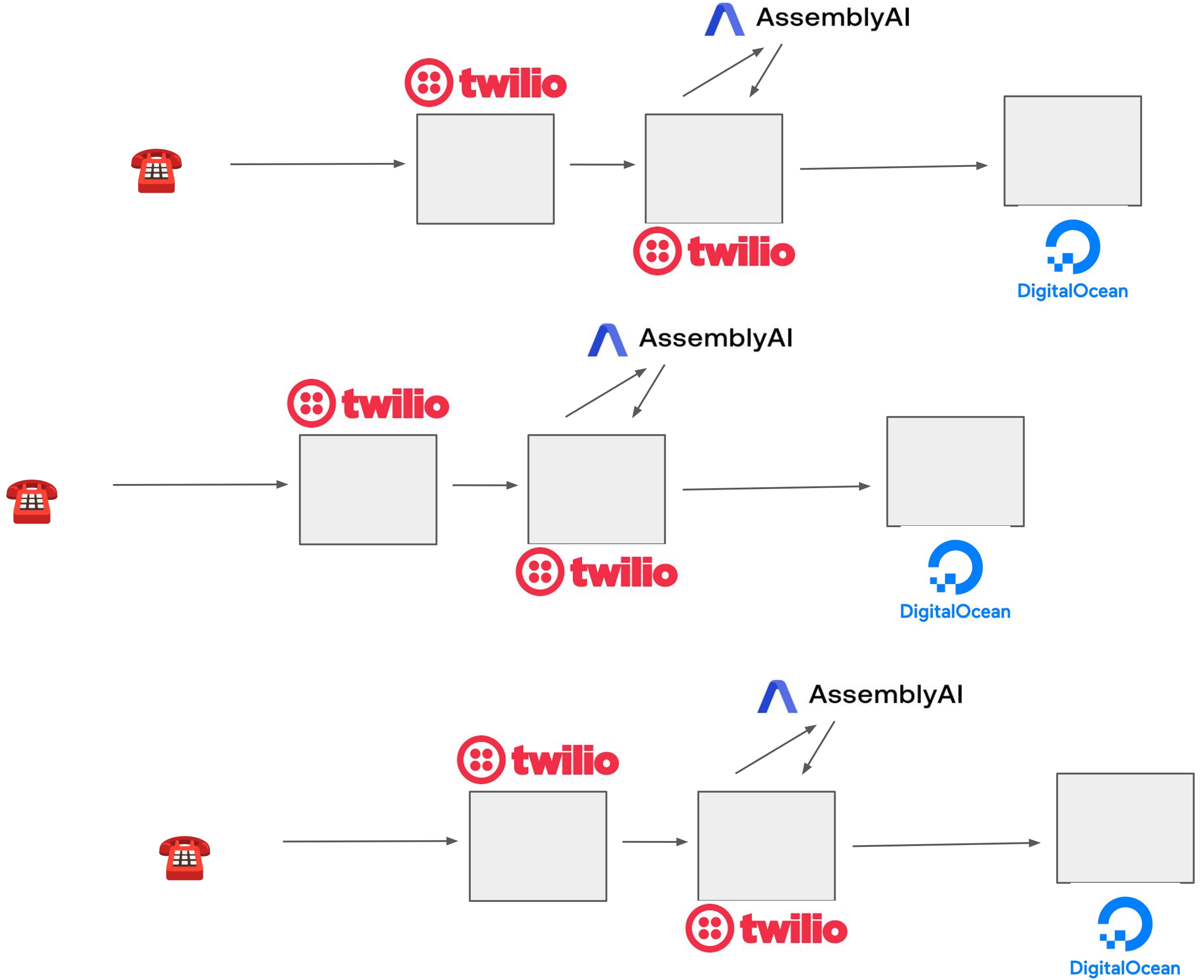
- I want to take that data **<u>1 call at a time</u>**, write a function that integrates it with a transcription and its metadata (language code, duration)

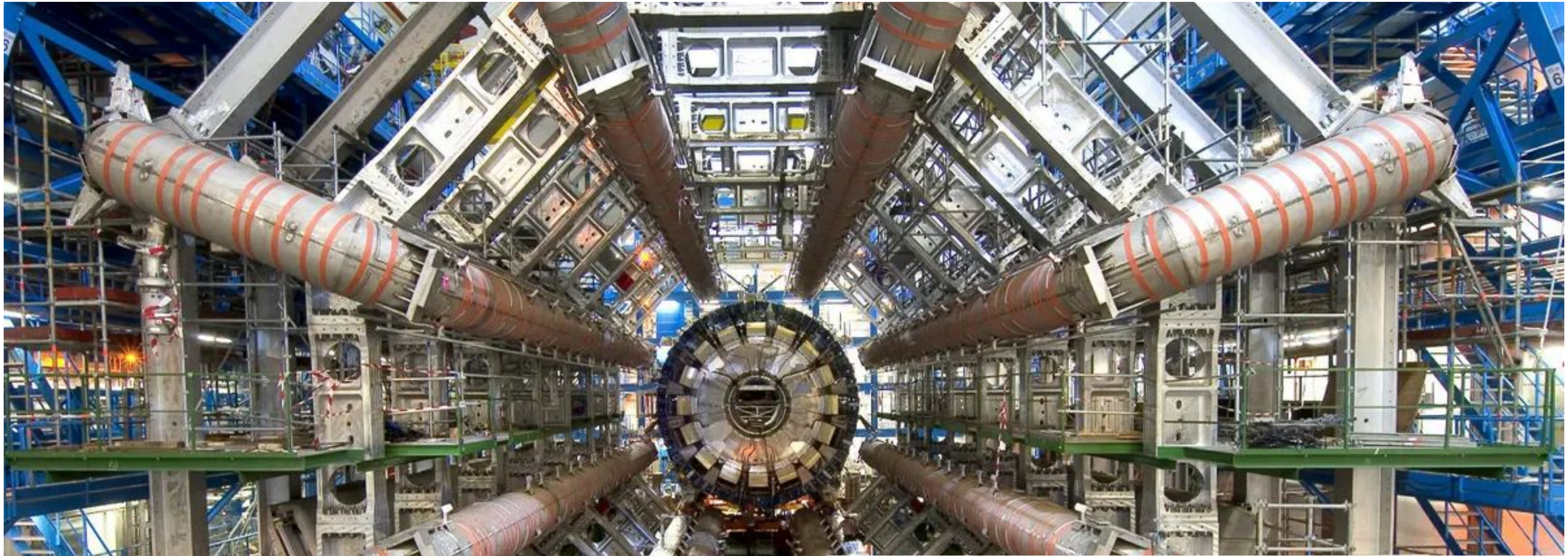
- And then store my results in the cloud (Digital Ocean)


AssemblyAI
twilio
Handle call, record (PaaS)
FUNCTION w/external API call
twilio
S3 storage
DigitalOcean

SCALE!

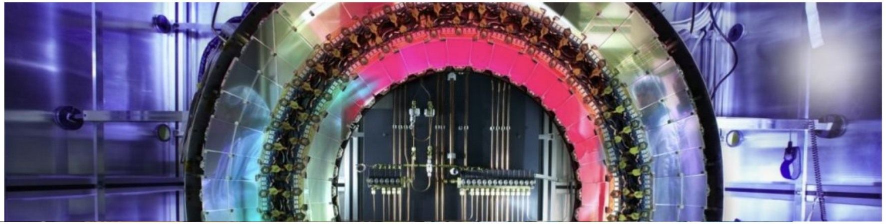
these all might be on different machines, in different datacenters, or regions(!!), as needed





Less Than 1% of Large Hadron Collider Data Ever Gets Looked at

PHYSICS 31 October 2018 By MICHELLE STARR



The Journal of Supercomputing (2023) 79:8940–8965
https://doi.org/10.1007/s11227-022-05016-y

Leveraging an open source serverless framework for high energy physics computing

Vincenzo Eduardo Padulano[1,2] · Pablo Oliver Cortés[1] · Pedro Alonso-Jordá[1] · Enric Tejedor Saavedra[2] · Sebastián Risco[3] · Germán Moltó[3]

Accepted: 16 December 2022 / Published online: 2 January 2023

Abstract

CERN (Centre Europeen pour la Recherce Nucleaire) is the largest research centre for high energy physics (HEP). It offers unique computational challenges as a result of the large amount of data generated by the large hadron collider. CERN has devel-