Site Reliability Engineering
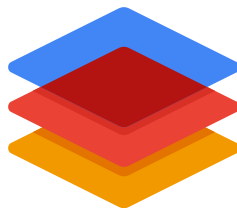
Distributed ImageServer

Non-Abstract Large System Design in 1 Hour

NALSD / SRE Classroom
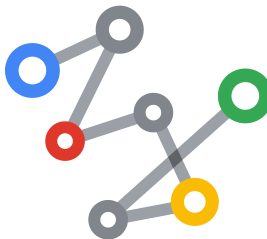
Hands-on Exercises

Abstract System Design

Non-Abstract Design

Planet Scale
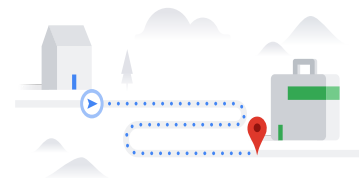
Distributed Architecture

Resilient Systems

Agenda

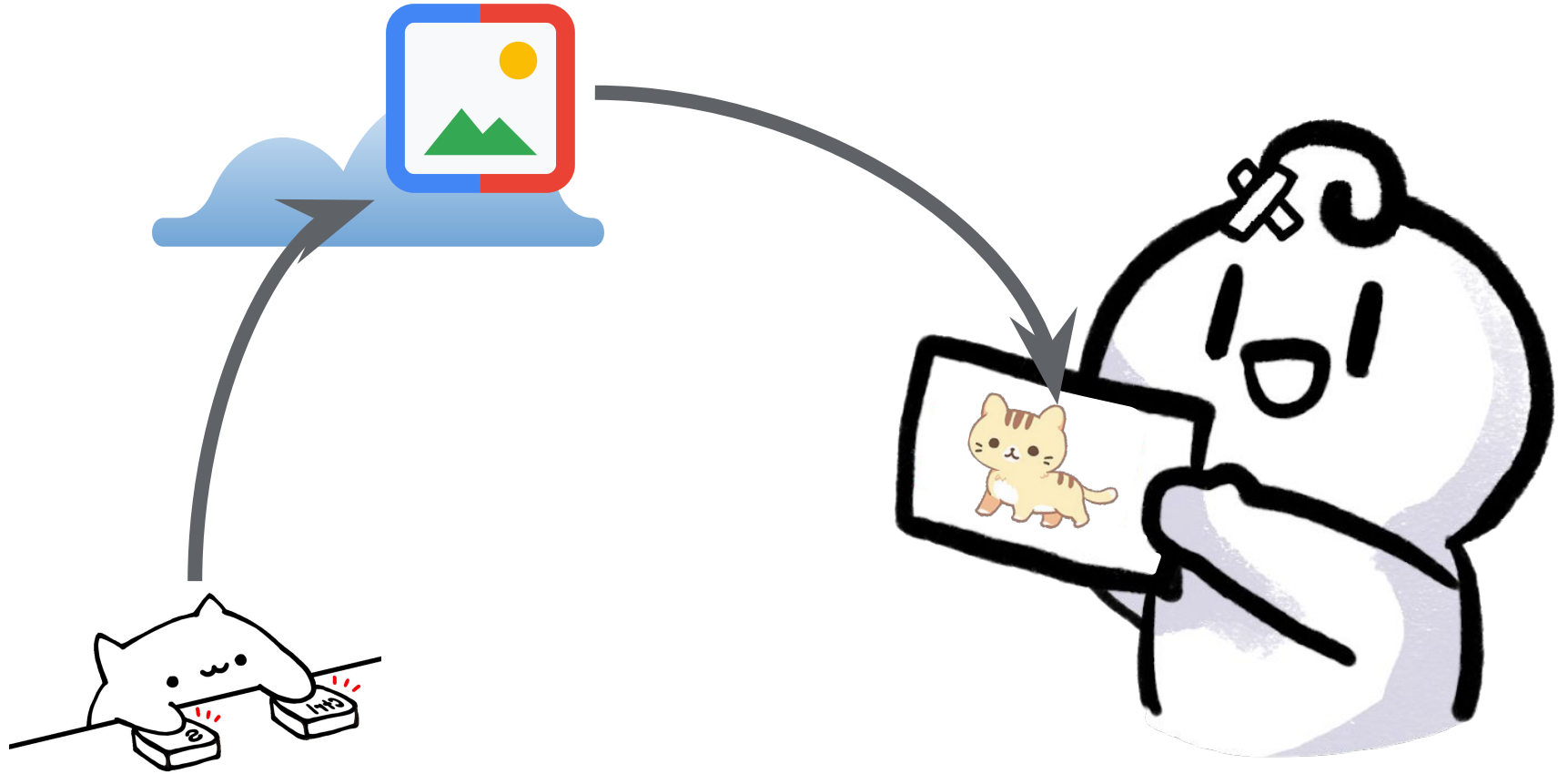Problem Statement

Sample Solution

Advanced Optimizations

Wrap Up & Conclusions

Problem Statement
Let's identify the problem at hand

The Problem Space

Our Mission:

Design a planet-scale image-serving system that users can use to upload, search for, and view images.

Gather Requirements

Let's identify what we know and what we need

"There are millions of images online! It's overwhelming to find the ones I care about. **I want to search images by keyword**."

"It is extremely painful when **I can't quickly and reliably access my favorite cute cat photos** when I need them."

"I like to share photos of my travels with my friends and family, but **I don't want those photos to linger on the internet forever!**"

Google

Key Product Requirements

	1	**2**	**3**
REQUIREMENTS	**Easy To Use**	**Reliable Service**	**Data Integrity**
	Simple and intuitive UX.	System Performance.	Data boundaries.
USER PAIN POINTS	Can't Find Features	Slow Service	Privacy & Security
	User Toil	Data Loss	Unexpected Data Expiration

Google

Detailed Requirements

1 Million
Global Users

50 Uploads /
User / Day

Upload Page

Search

user or tag

...
**10
results
/ page**
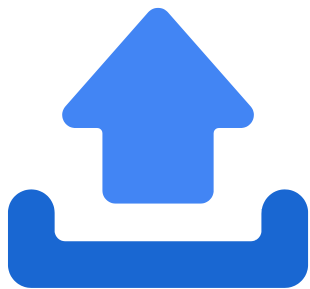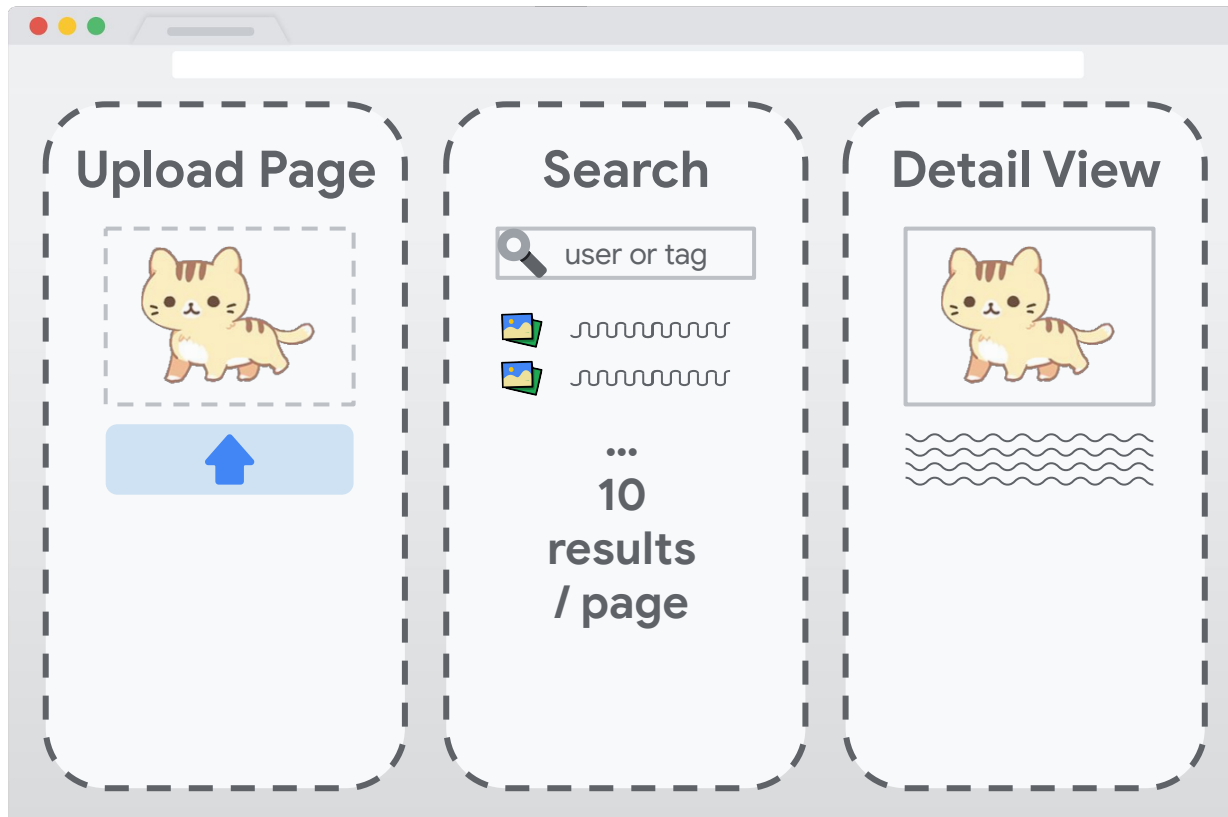
Detail View

*An **SLO** is a Service Level Objective, a measurable guide to specific system performance.*

Detailed Requirements: SLOs

Detail View page (*1 image, full-resolution*):

serve within 200ms at 99.9 percentile

(HTTP 200)

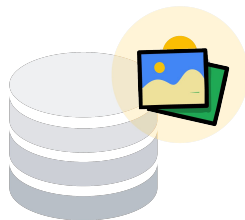**Search page** (*10 thumbnails*):

serve within 250 ms at 99.9 percentile

(HTTP 200)

SLOs only apply to data that is **30 days or fresher**.

Google

Stuff That's Available To Us Out Of The Box

Reliable Network

Globally Distributed Storage
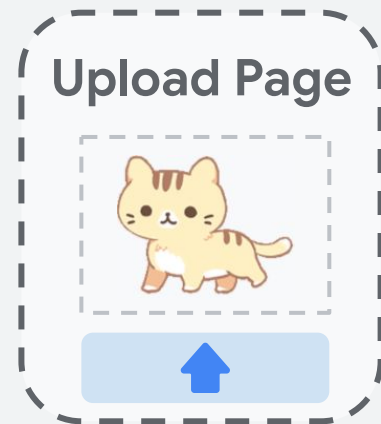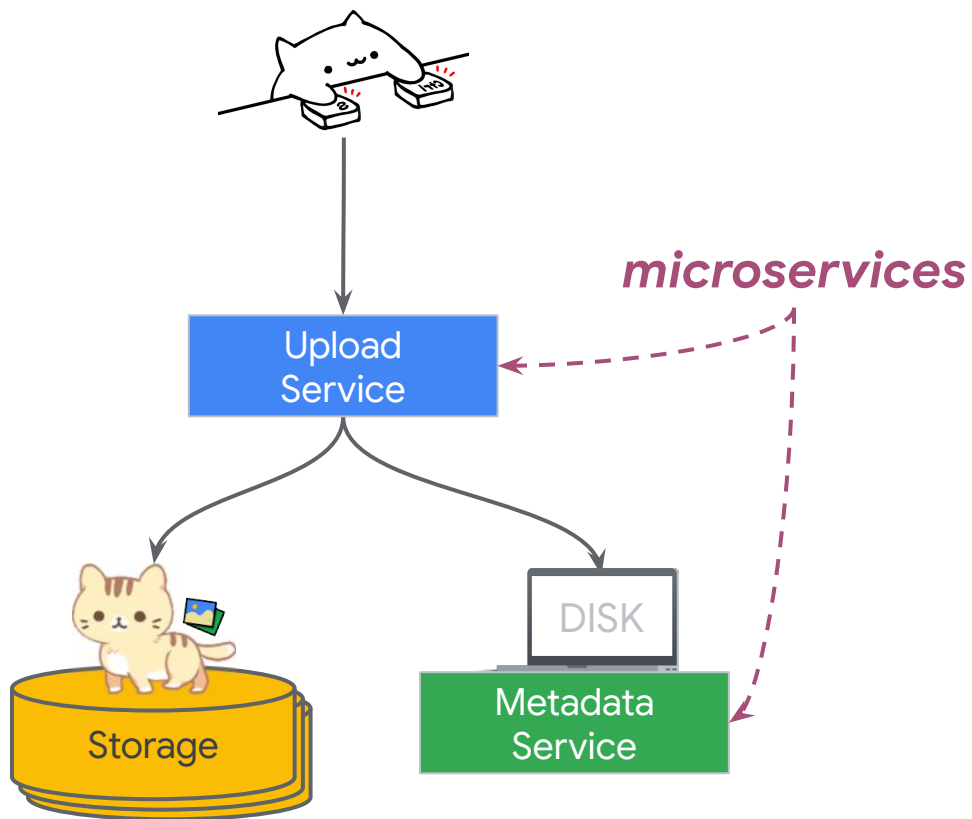
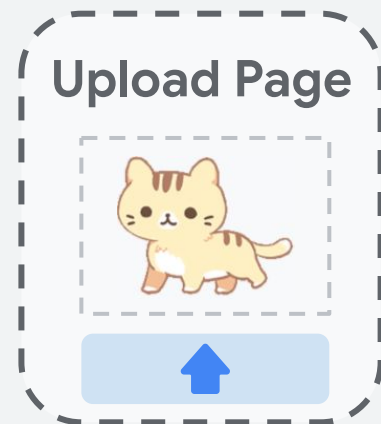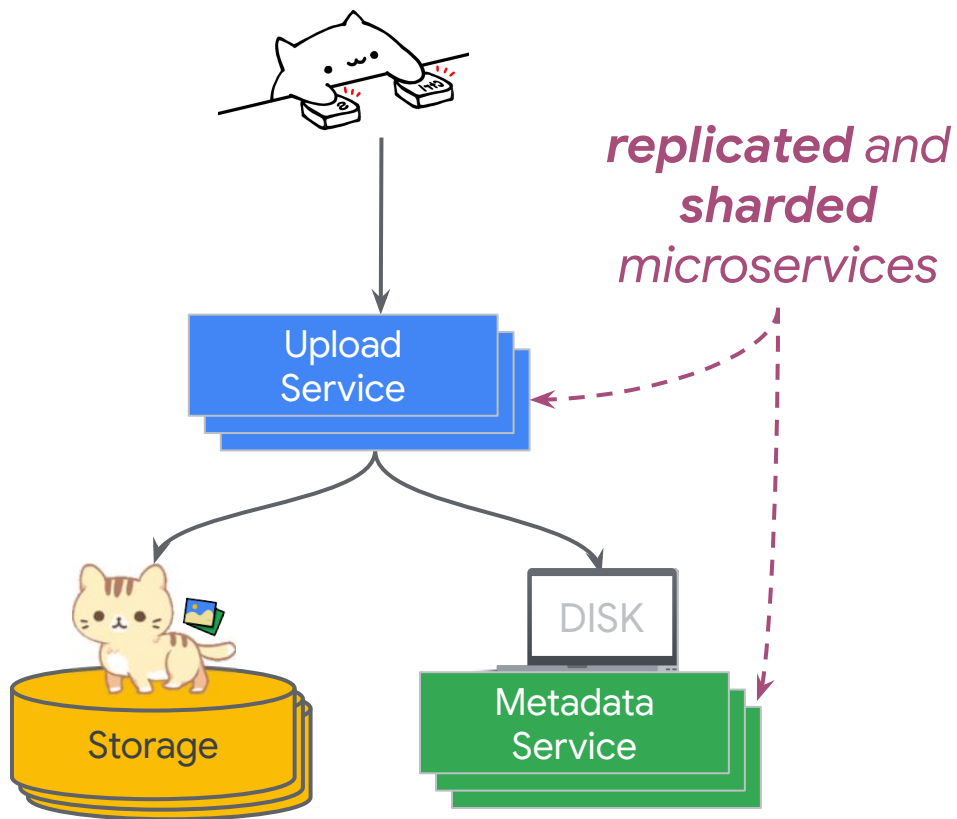Datacenters: 3 Global Regions

Hardware: HDD Machines SSD Machines
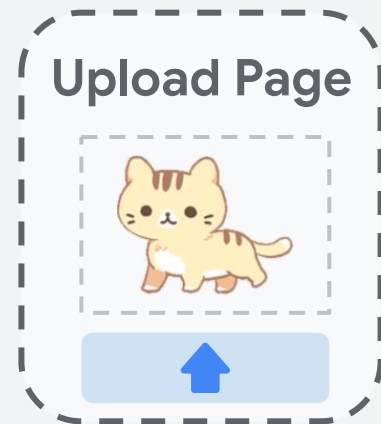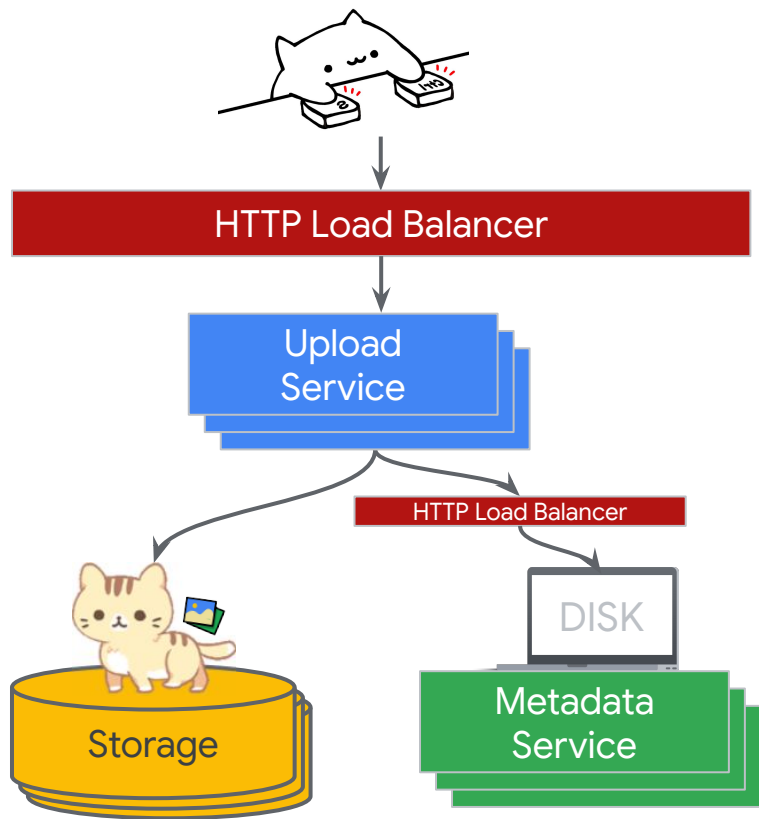
Google

Sample Solution: Abstract Design

Google

System Diagram: Upload

microservices

Upload Service

Storage

DISK

Metadata Service

Upload Page

Upload: Scaling Services

replicated and
sharded
microservices

Upload
Service

DISK

Storage

Metadata
Service

Upload Page

Upload: Scaling Services

HTTP Load Balancer

Upload
Service

HTTP Load Balancer

DISK

Storage

Metadata
Service

Upload Page

Upload: Scaling Services

HTTP Load Balancer

Upload Service

DISK

Storage

Metadata Service

Upload Page
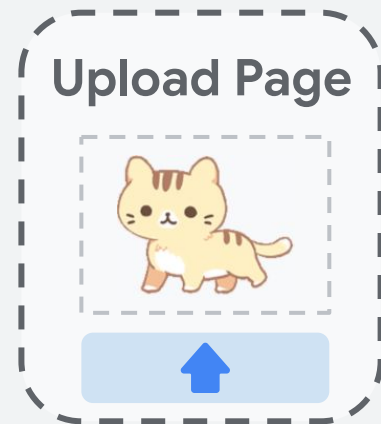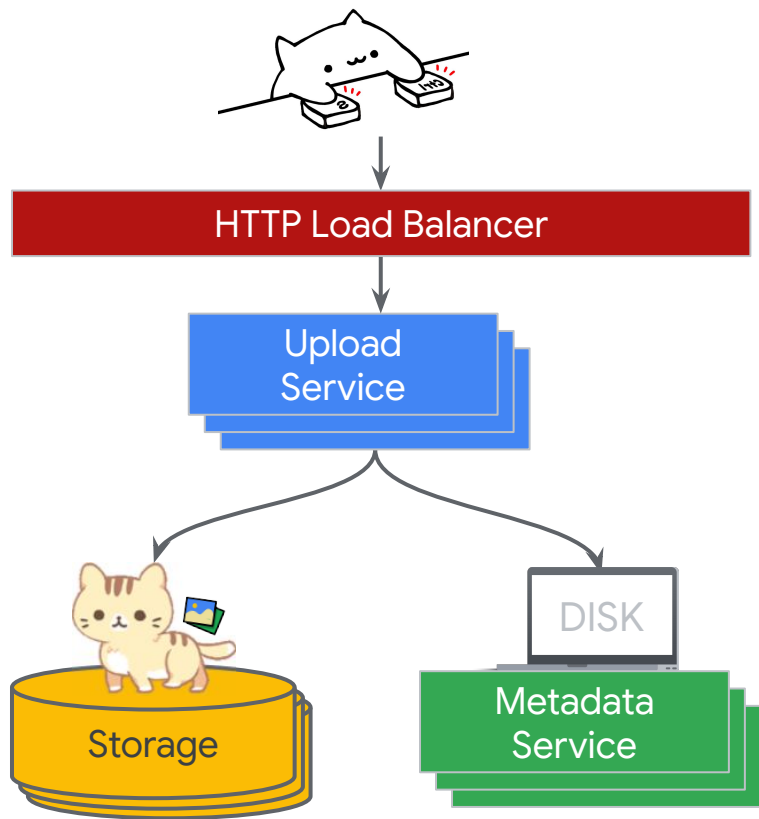
System Diagram: Download

HTTP Load Balancer

Download Service

speed up retrieval using caches

DISK

Storage

Metadata Service

Detail View

Why separate services?

HTTP Load Balancer

Upload Service

Download Service

Storage

DISK

Metadata Service

Upload Page

Detail View

System Diagram: Search

HTTP Load Balancer

Search Service

speed up retrieval using caches

DISK

Storage

Metadata Service

Search

🔍 user or tag

10 results / page
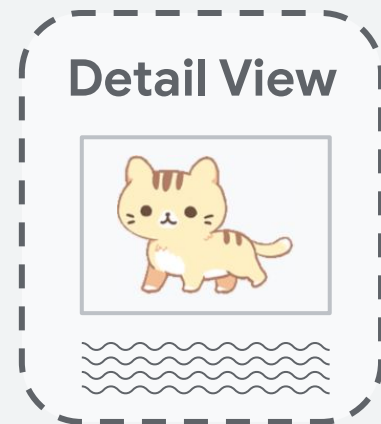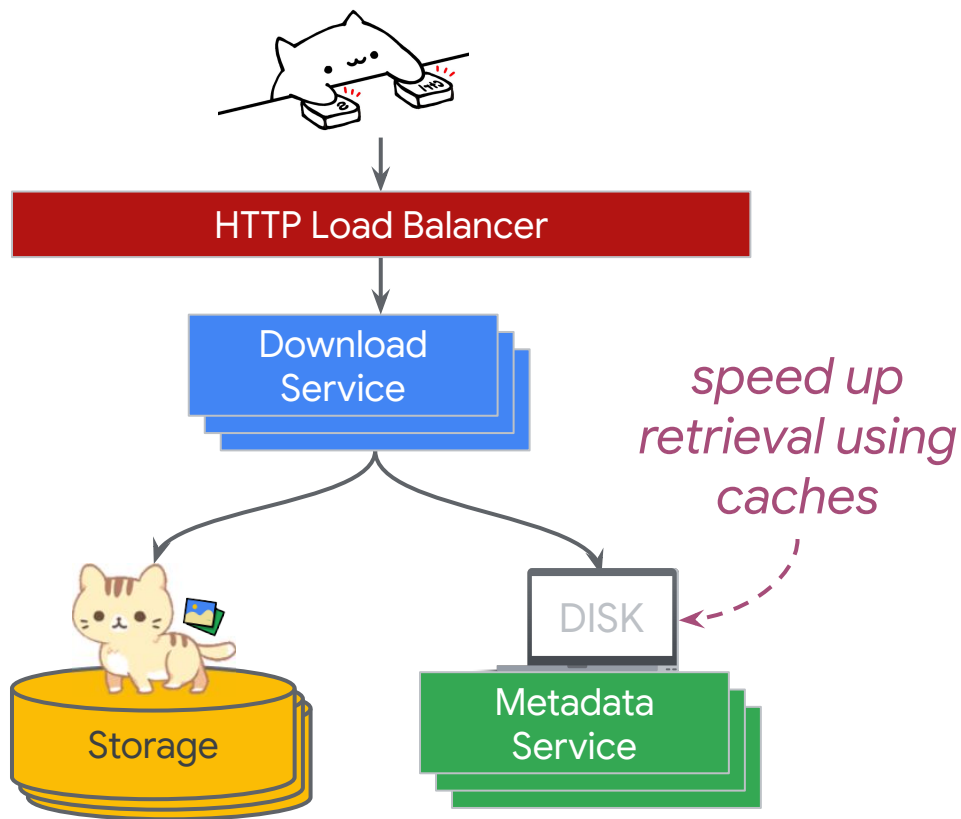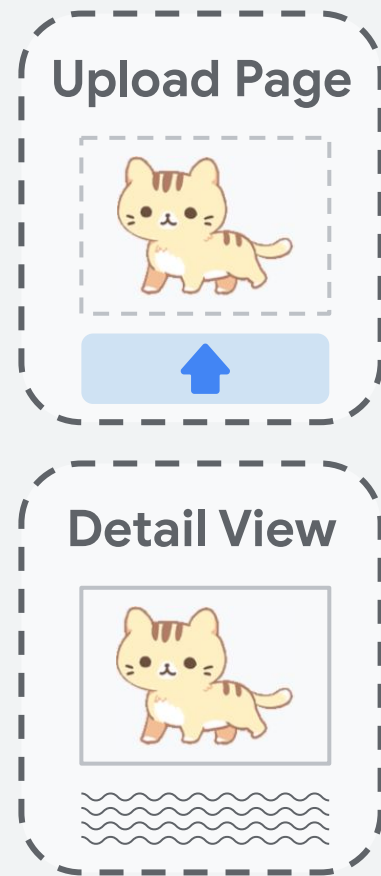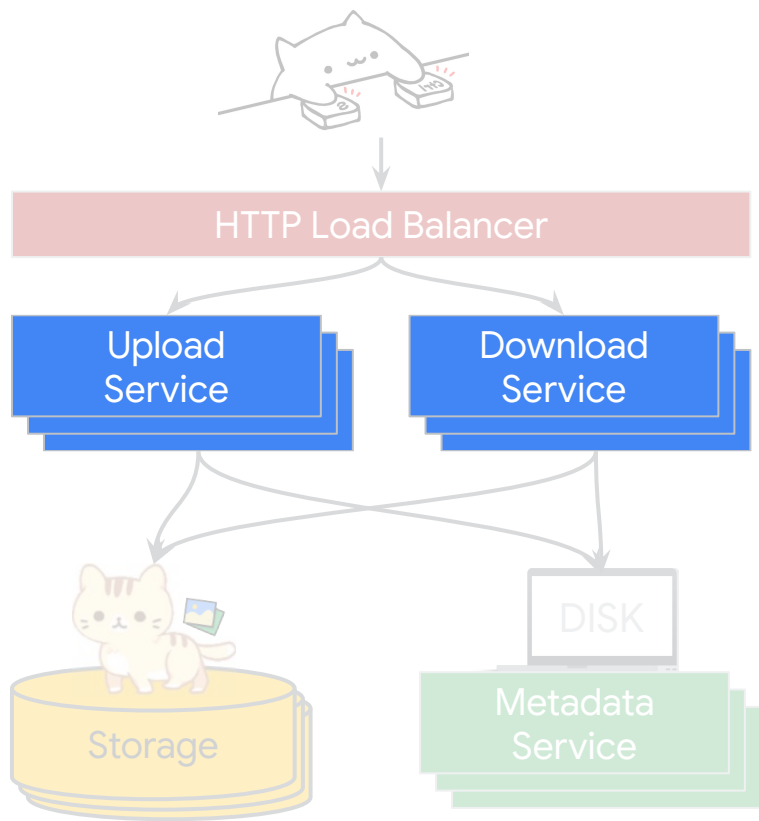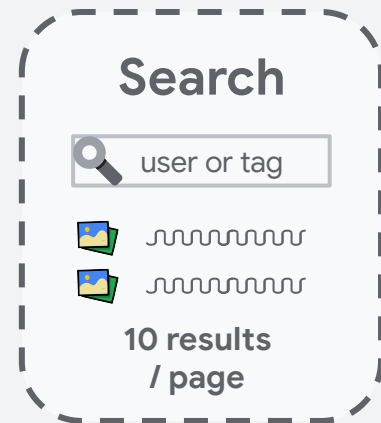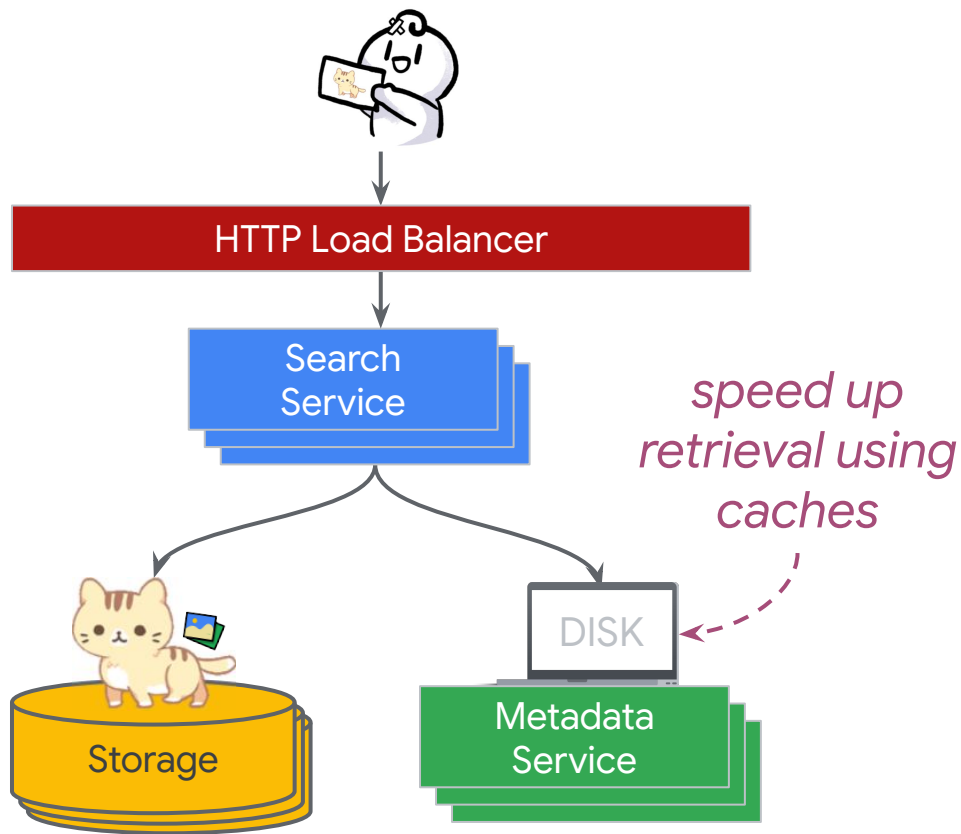
A Closer Look At The Metadata Service
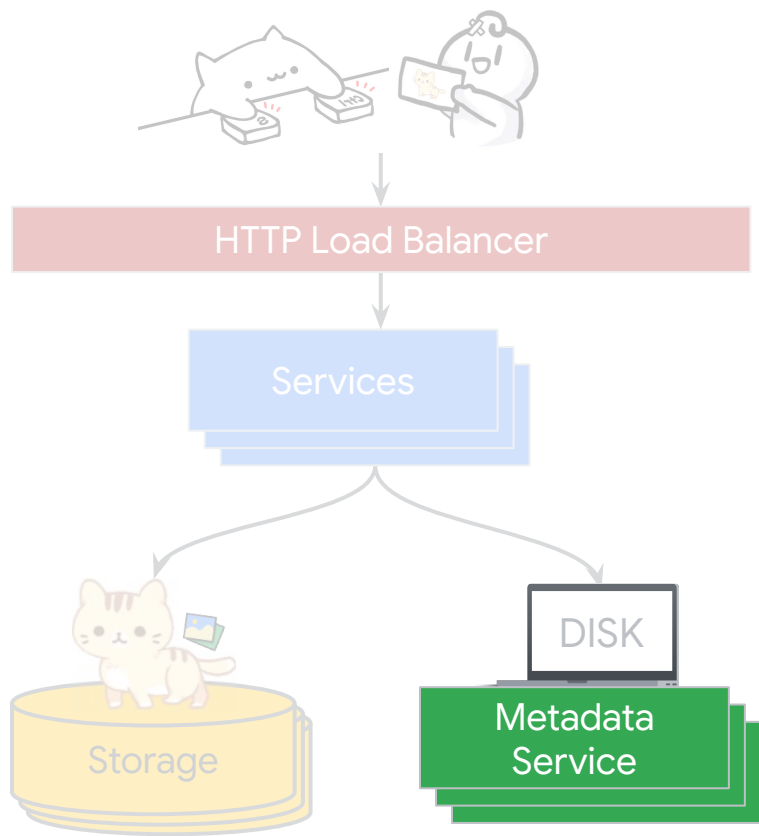
Stores Metadata:

- Uploading User

- Tag(s)

- StorageID

- Description

Size: 8 KB / Image

ImageServer System Diagram

HTTP Load Balancer

| Upload Service | Search Service | Download Service |

Storage

DISK

Metadata Service

Upload Page

Detail View

Search

🔍 user or tag

...
10 results / page

Google

Reminder: Questions

Google
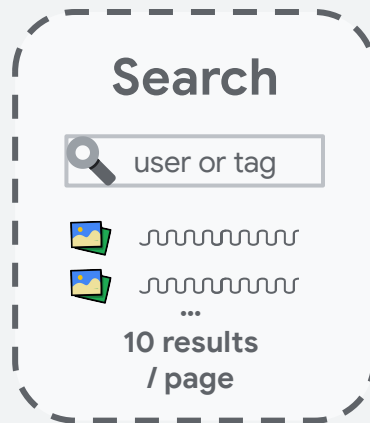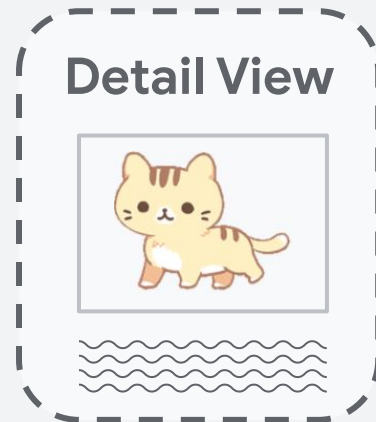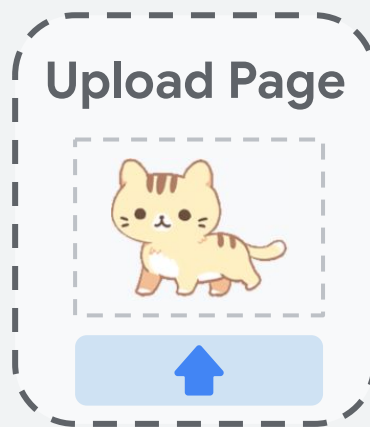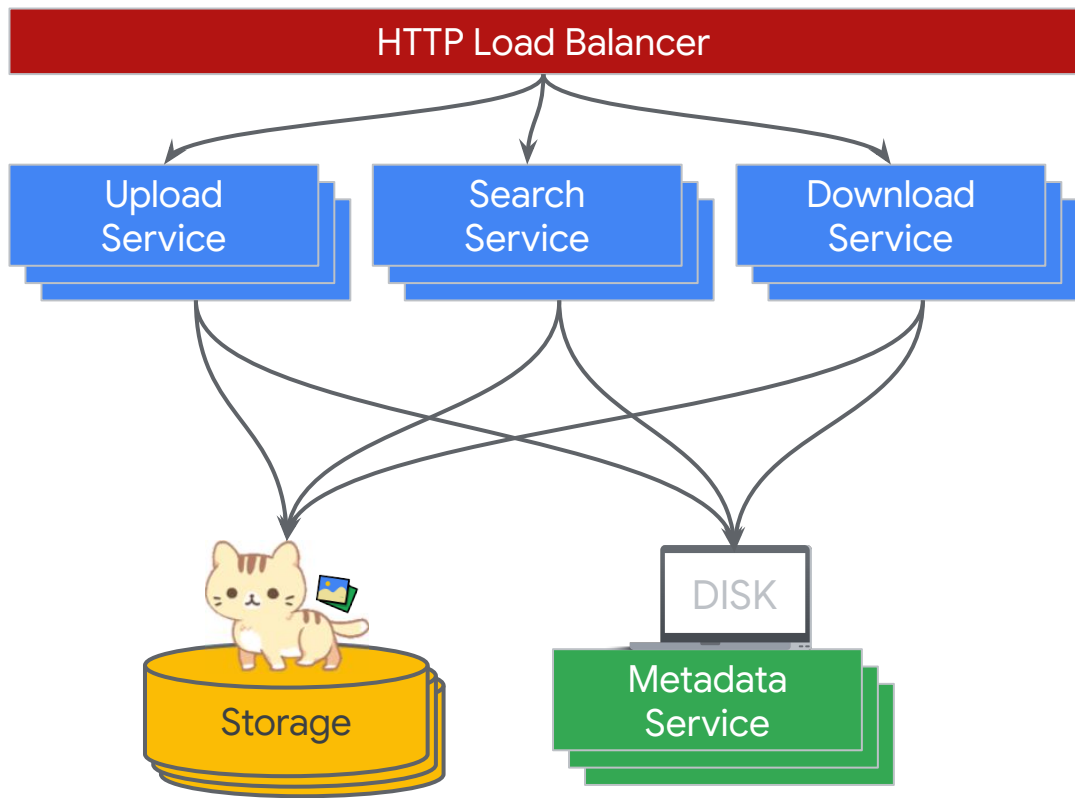
Sample Solution: Provision the System & Evaluate against SLOs

Google

Questions To Answer

How many machines do we need to allocate to each microservice?

Can we meet our SLO latency requirements?

SLOs

Download:

200ms at 99.9 percentile

Search:

250ms at 99.9 percentile

Coverage:

30 days or fresher data

Google

26

Provisioning

- Provisioning is an art.

- Simplify where possible

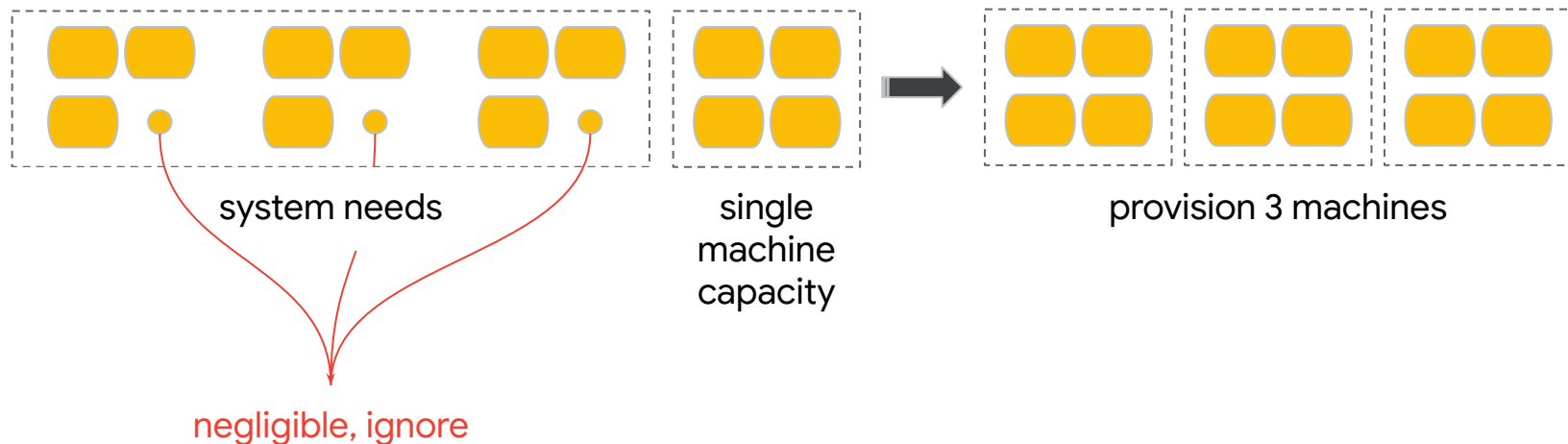- Over-provision by default

- Granularity: units of one machine

system needs

single machine capacity

provision 3 machines

negligible, ignore

Storage: Images

Uploaded image content:
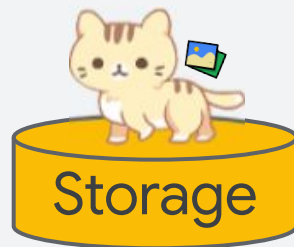
1 million users

  * 50 img / user / day

  * 4 MB / img

~= 200 TB / day

or, for 30 days retention:

~= 6 PB / 30 days

Key: **Image StorageID**
Value: **Thumbnail & Image**

average image size = 4 MB

thumbnail size = 256 KB

data time to live = 30 days

Storage: Metadata

Uploaded image metadata:

1 million users

  * 50 img / user / day

  * 8 KB / img

~= 0.5 TB / day

or, for 30 days retention:

~= 15 TB / 30 days

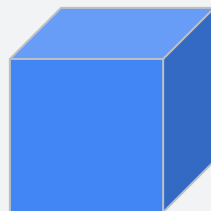> **→ 4 HDD machines or**
>
> **8 SSD machines**
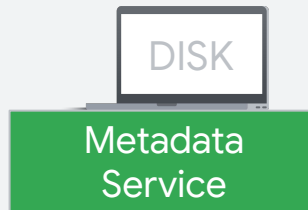
DISK

Metadata
Service

Stores image metadata,
Indexed for efficient searches

average img metadata size = 8 KB

data time to live = 30 days

Machine:
24GB RAM
2TB SSD *or* 4TB HDD
10Gbps ethernet

29

Bandwidth: Upload

Avg load

  = 200 TB / day

  = ~2500 MB / s

Peak load = 1.25x avg load

  = ~3500 MB / s

  = ~30 Gbps

30 Gbps inbound,

30 Gbps outbound

→ 3 machines

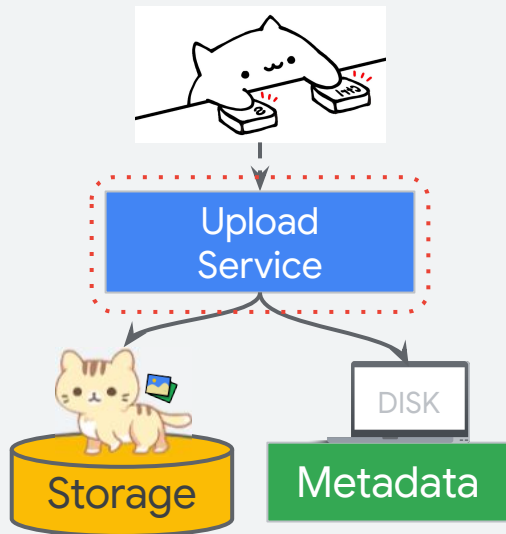
Upload Service

Storage

DISK

Metadata

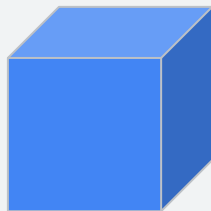image uploads = ~200 TB / day

Machine:
24GB RAM
2TB SSD *or* 4TB HDD
10Gbps ethernet

Bandwidth: Download

Avg load

= 400 TB / day

= ~5000 MB / s

Peak load = 1.25x avg load

= ~6500 MB / s

= ~60 Gbps

60 Gbps inbound,

60 Gbps outbound

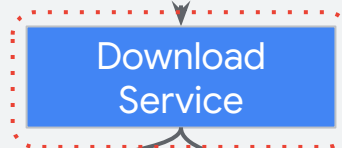**→ 6 machines**

Download Service
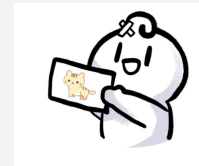
Storage

DISK

Metadata

image downloads = ~400 TB / day

Machine:
24GB RAM
2TB SSD *or* 4TB HDD
10Gbps ethernet

Bandwidth: Search

Avg load = ~2000 MB / s

Peak load = ~20 Gbps

20 Gbps inbound,

20 Gbps outbound

→ 2 machines
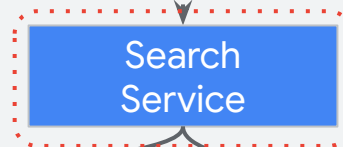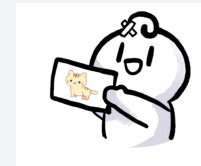
search queries = ~? TB / day

Machine:
24GB RAM
2TB SSD *or* 4TB HDD
10Gbps ethernet

32

Bandwidth: Metadata

Upload, Download, and Search each call Metadata Service.

Each call → read or write image metadata.

1.5 Gbps inbound,
1.5 Gbps outbound
→ 1 machine

Stores image metadata,
Indexed for efficient searches

average img metadata size = 8 KB

data time to live = 30 days

Latency: Metadata

1. Receive image metadata over the network

   = <1 ms

2. Write image metadata

   = ~1 ms on SSD
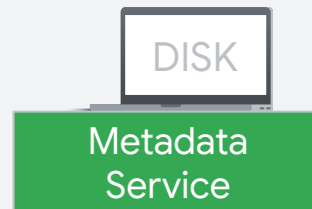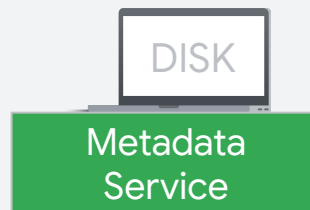
Total latency = ~2 ms

Reminders:

- HDD time: ~10 ms / 8 KB

- SSD time: <1 ms / 8 KB

- Network time: <1 ms / 8 KB

DISK

Metadata
Service

Stores image metadata,
Indexed for efficient searches

average img metadata size = 8 KB
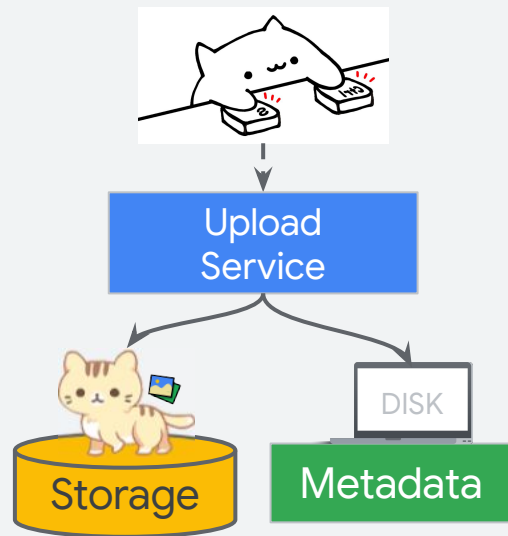
34

Latency: Upload

1. Write image metadata

   = ~2 ms

2. Write image to storage

   = ~200 ms

3. Send image to UI

   = ~5 ms

Total latency = ~210 ms

Reminders:

- Metadata time: ~2 ms / img

- Storage time: ~200 ms / img

- Network time: ~5 ms / img
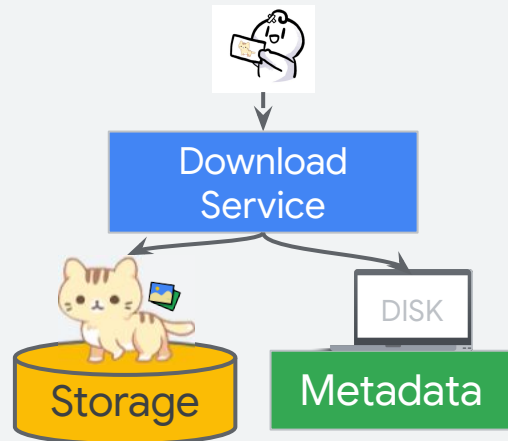
Latency: Download

1. Read image metadata

   = ~2 ms

2. Read image from storage

   = ~100 ms

3. Send image to UI

   = ~5 ms

Total latency = ~110 ms

Meets the SLO requirement.

Reminders:

- 99.9% ops finish in <200ms

- Metadata time: ~2 ms / img

- Storage time: ~100 ms / img

- Network time: ~5 ms / img
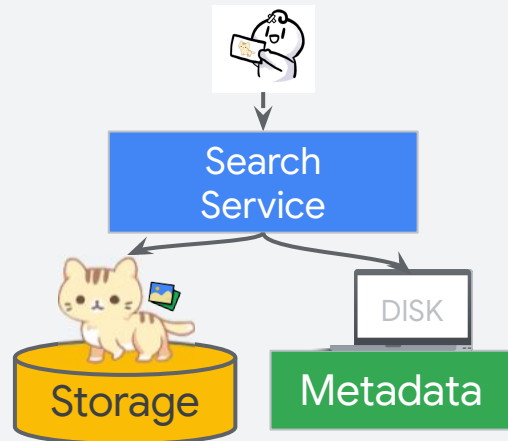
Latency: Search

1. Get query matches

   = ~2 ms

2. Read thumbnails from storage

   = ~100 ms

3. Send results to UI

   = ~1 ms

Total latency = ~105 ms

Meets the SLO requirement.

Reminders:

- 99.9% ops finish in <250ms

- Metadata time: ~2 ms / img

- Storage time: ~100 ms / img

- Network time: ~1 ms / search

Search Service

Storage

DISK

Metadata

Reminder: Questions

How many machines do we need?

	Bandwidth	Storage	Machines Required
Upload	3	-	3
Download	6	-	6
Search	2	-	2
Metadata	1	8	8

Google

Bill of Materials

Final count of machines:

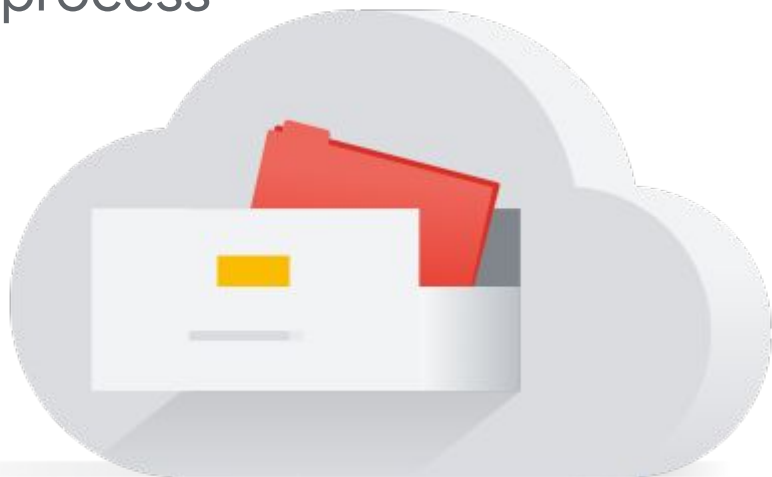3 upload +

6 download +

2 search +

8 metadata

= 19 per DC * 3 DCs * 1.25 (for infra tax + more load spikes)

= 72 machines

Advanced Optimizations

- Caching

- Storage backend degradation

- Capacity growth (per year, retention)

- Privacy requirements (GDPR anyone?)

- Toil (rollout, maintenance) - more a process

  thing though

Google

Last thoughts

- Start simple and iterate

- Flexibility vs.

  Premature future-proofing

- Make data-driven decisions

Take breaks and enjoy

the process!

Site Reliability Engineering

Distributed ImageServer

Non-Abstract Large System Design in 1 Hour

More material like this at
https://googlesre.page.link/sre-classroom!