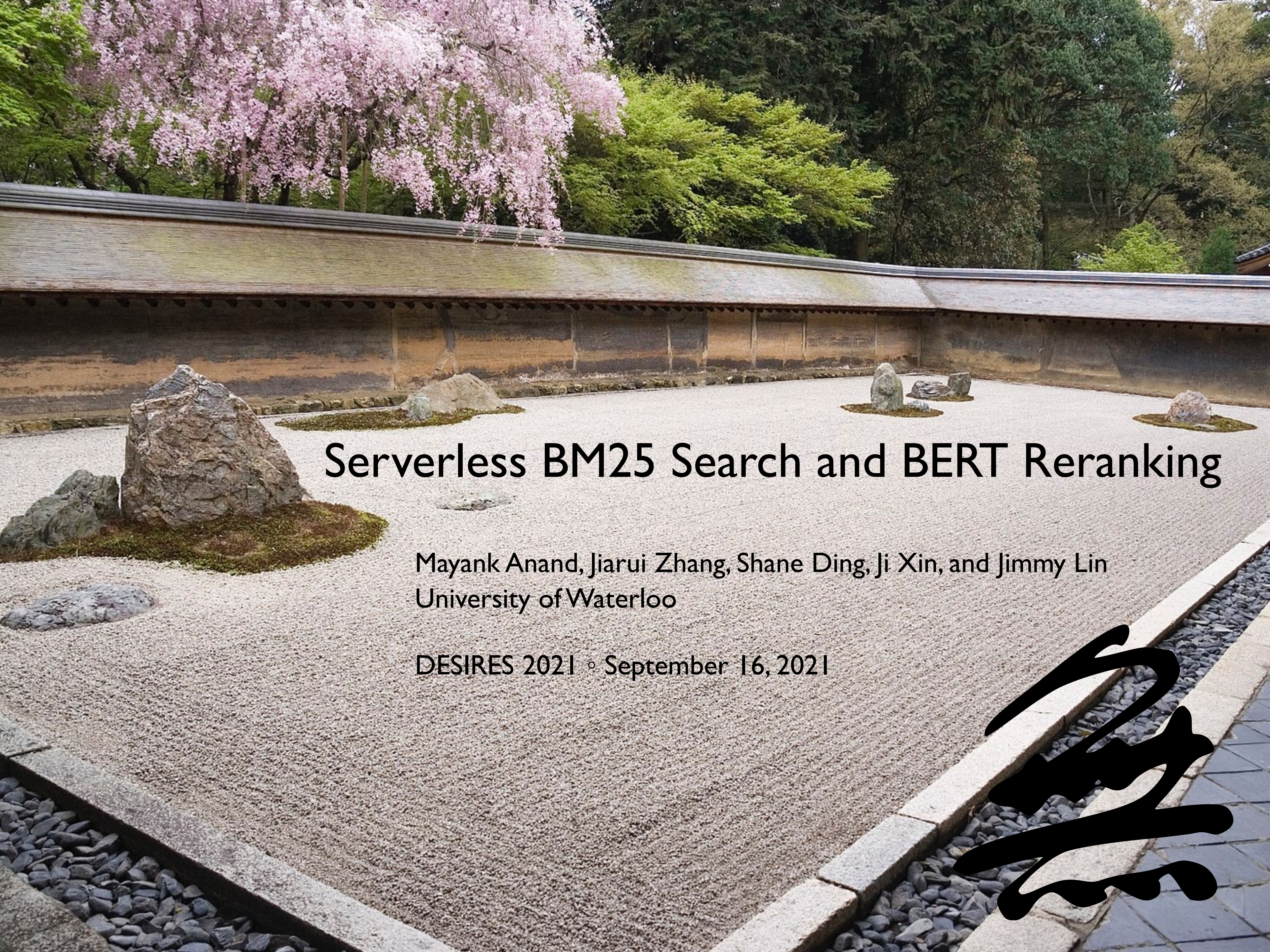
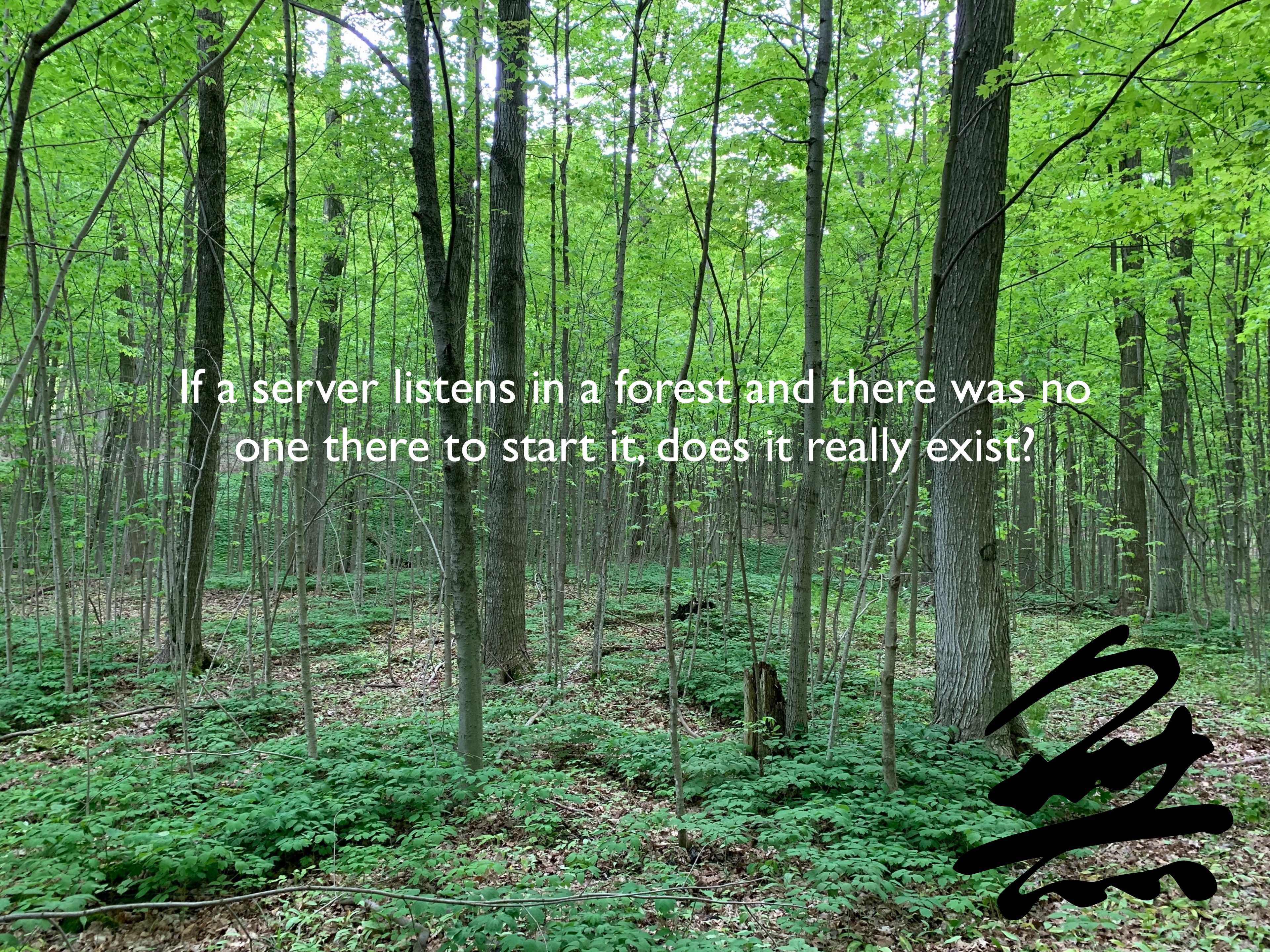


Serverless BM25 Search and BERT Reranking

Mayank Anand, Jiarui Zhang, Shane Ding, Ji Xin, and Jimmy Lin
University of Waterloo

DESIRES 2021 ◦ September 16, 2021



If a server listens in a forest and there was no one there to start it, does it really exist?

DESIRES

A systems-oriented biennial conference, complementary in its mission to the mainstream Information Access and Retrieval conferences, emphasizing the *innovative technological aspects* of search and retrieval systems.

It gathers researchers and practitioners from both *academia and industry* to discuss the latest innovative and visionary ideas in the field.

Computing without Servers, V8, Rocket Ships, and Other Batsh*t Crazy Ideas in Data Systems

Jimmy Lin
David R. Cheriton School of Computer Science
University of Waterloo

Wednesday, August 29, 2018

Source: Flickr (massimoleonardi/8064194094/)

DESIRES 2018
Desires 2018
Opening remarks
Omar Alonso and Gianmaria Silvello
BiCi
Bertinoro international Center for informatics
#DESIRES2018
UNIVERSITY OF WATERLOO

still 16 oz.
still 16 oz.
still 16 oz.
still 16 oz.
still 16 oz.
Vermont's Finest
BEN & JERRY'S
ICE CREAM
half baked
ONE PINT (473mL)
Chocolate & Vanilla Ice Creams mixed with Gobs of Chocolate Chip Cookie Dough & Fudge Brownies

We’ve done it!

Serverless BM25 Search and BERT Reranking

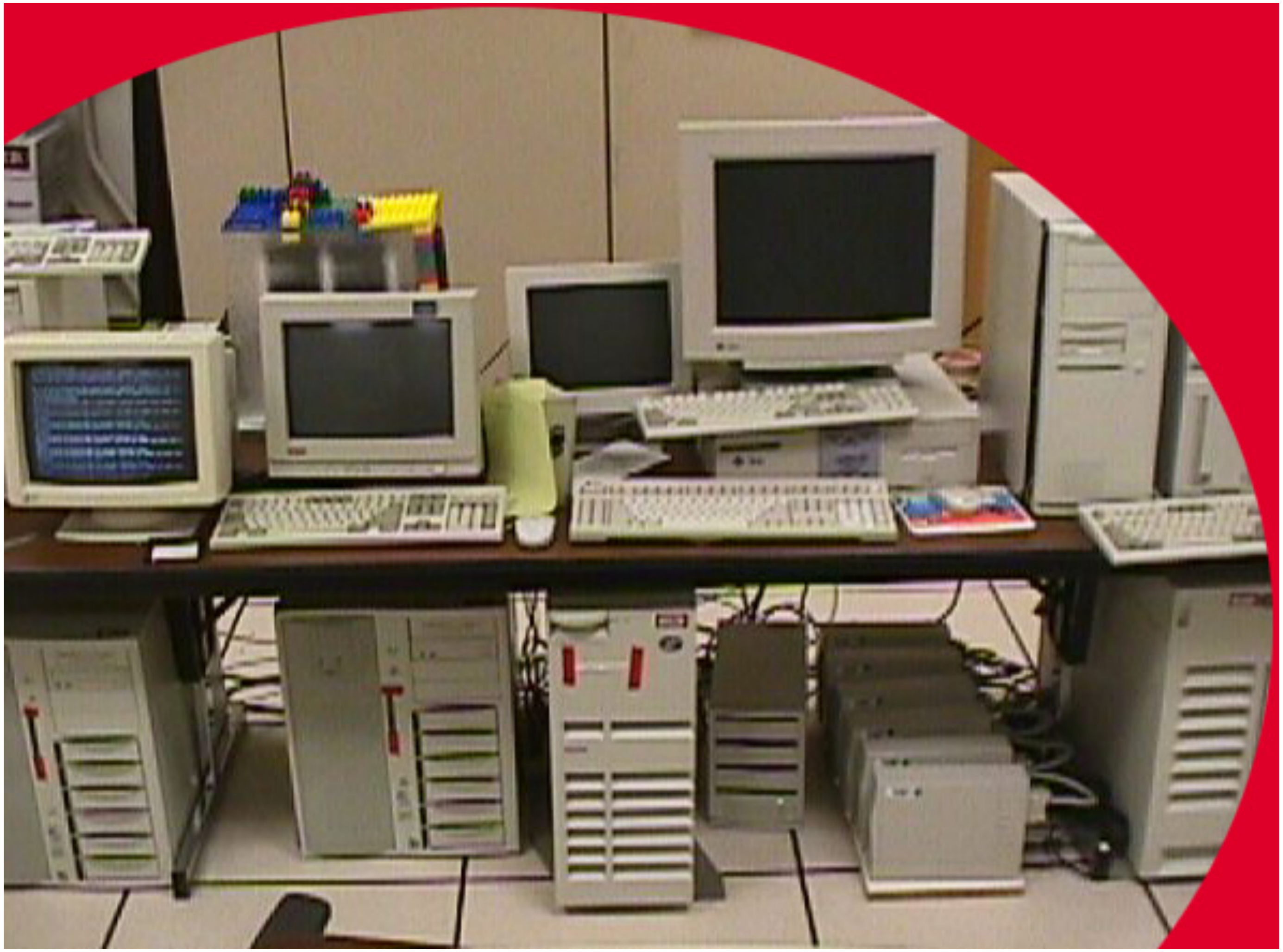
Servers

The most fundamental building blocks of IR systems
(both software and hardware)

In the beginning…

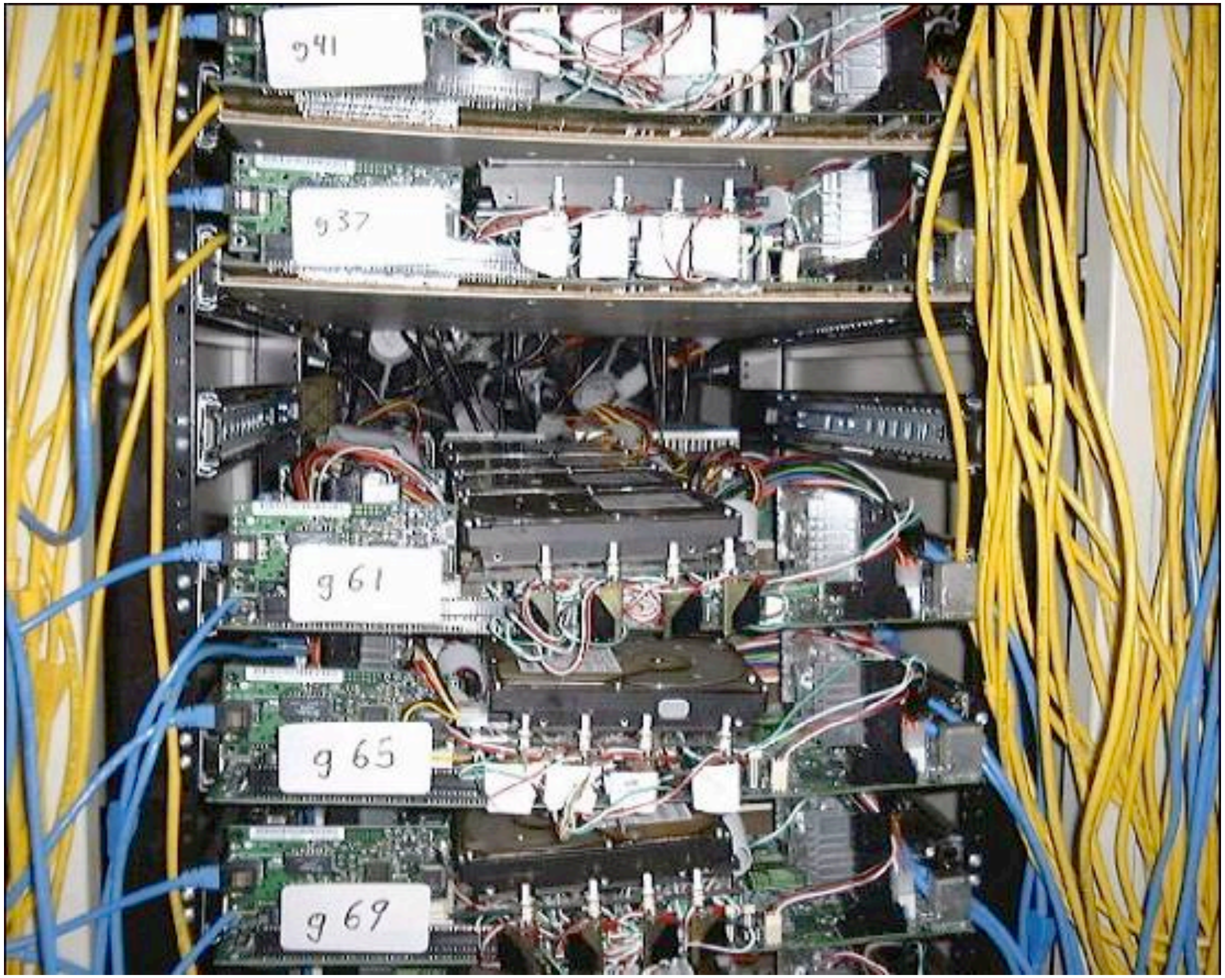
Source: Wikipedia

“Google” Circa 1997 (google.stanford.edu)



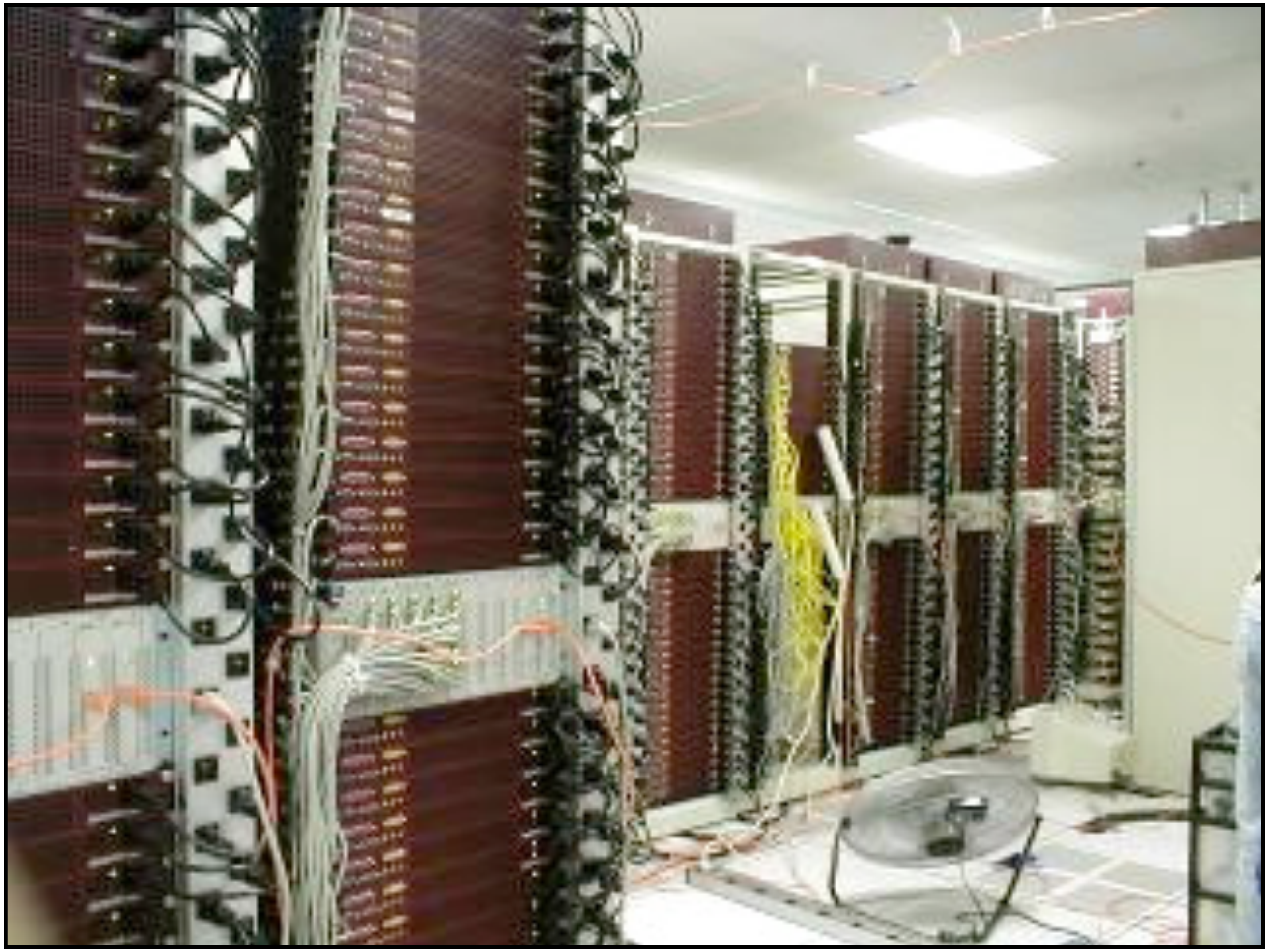
Slide from Jeff Dean, WSDM 2009 keynote

“Corkboards” (1999)



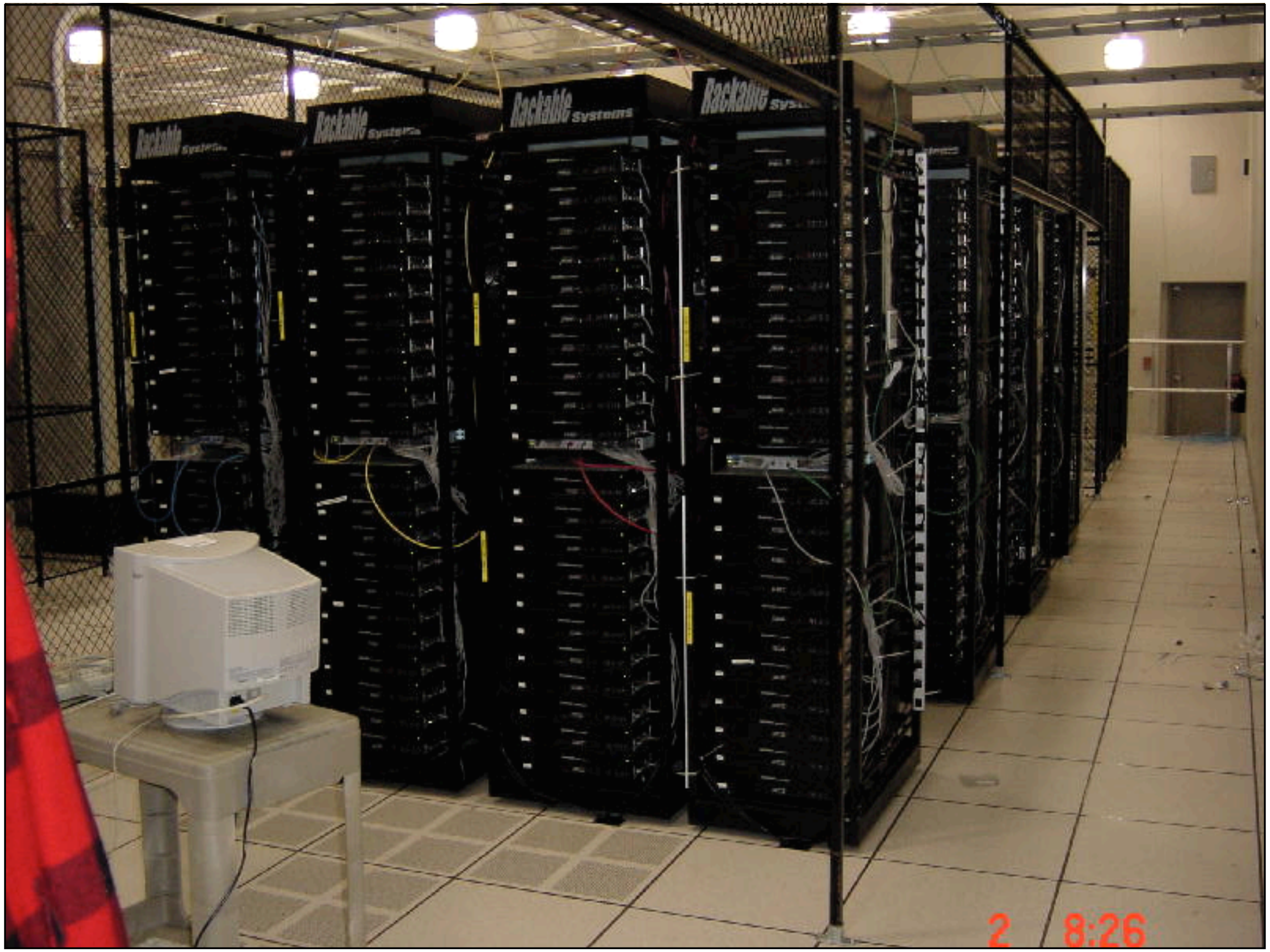
Slide from Jeff Dean, WSDM 2009 keynote

Google Data Center (2000)



Slide from Jeff Dean, WSDM 2009 keynote

Google Data Center (3 days later)



Slide from Jeff Dean, WSDM 2009 keynote

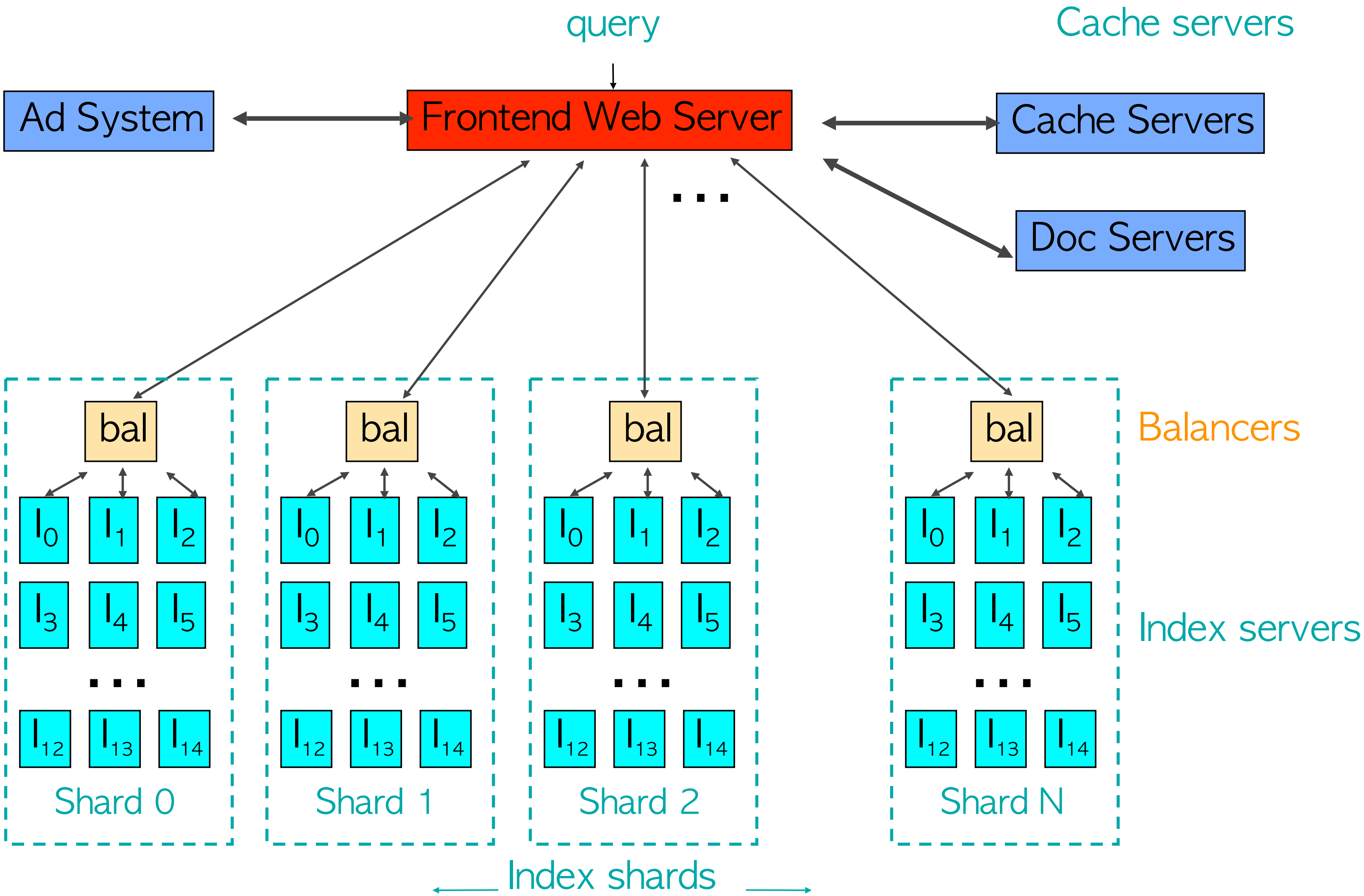
Servers

… in the cloud

Source: Google

Source: Google

Early 2001: In-Memory Index



Slide from Jeff Dean, WSDM 2009 keynote

Challenges Remain

(Especially if you're not Google)

Always on!

Scaling up… scaling down…

Scaling to zero?

Serverless

Preliminaries

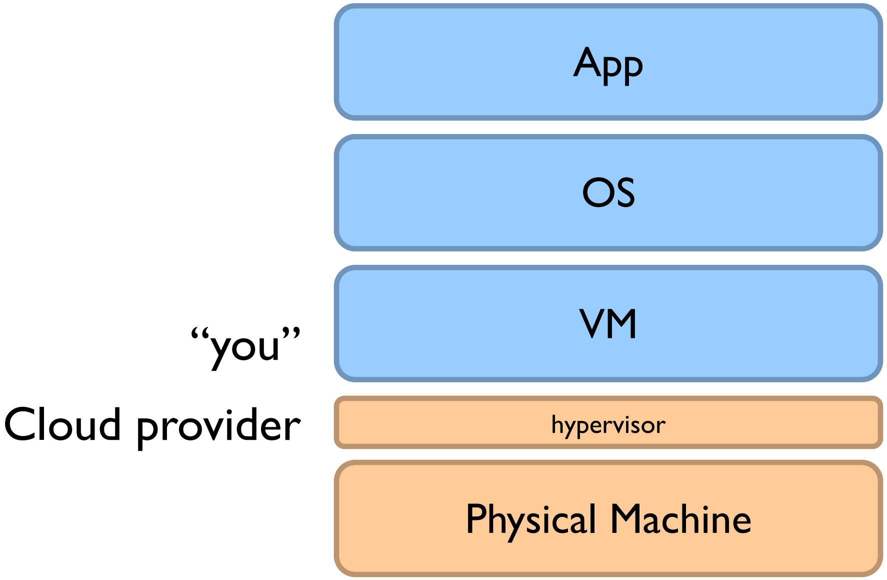
Cloud computing allows us to explore different abstractions and organizations of computing

(trend towards disaggregation)

In the beginning…

Source: Wikipedia

Infrastructure as a Service (IaaS)



App

OS

VM

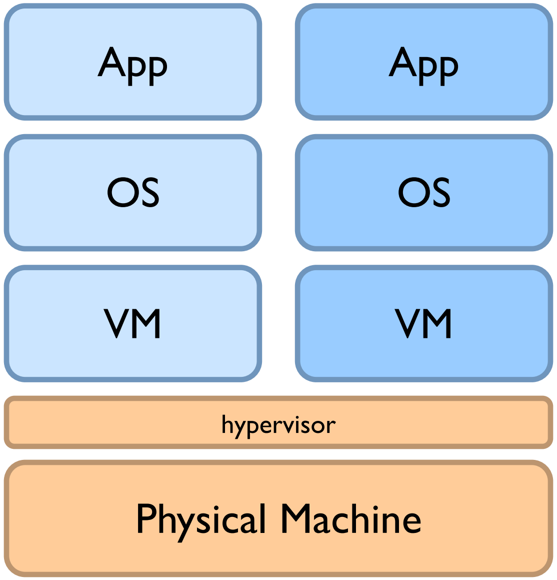
"you"

Cloud provider

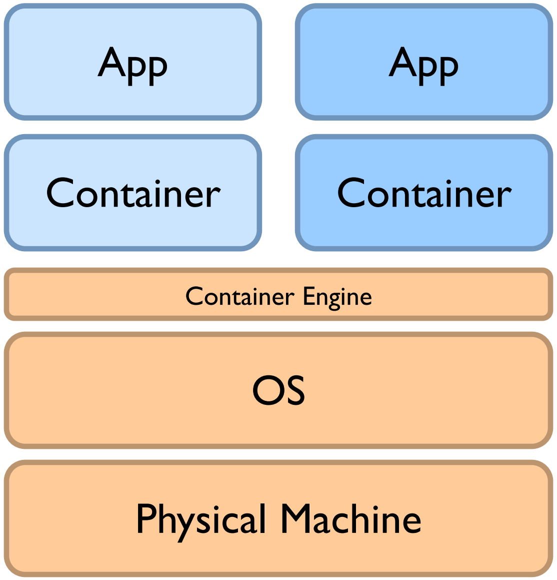
hypervisor

Physical Machine

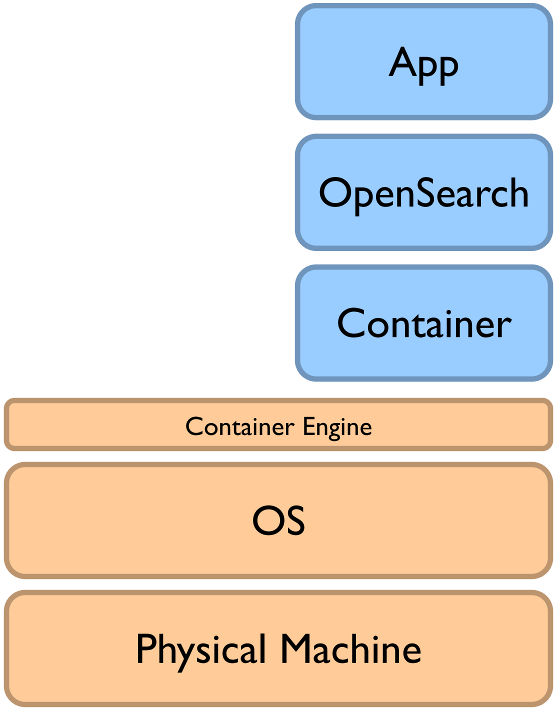
Multi-Tenancy



Containers >> VMs



Typical Stack



App

OpenSearch

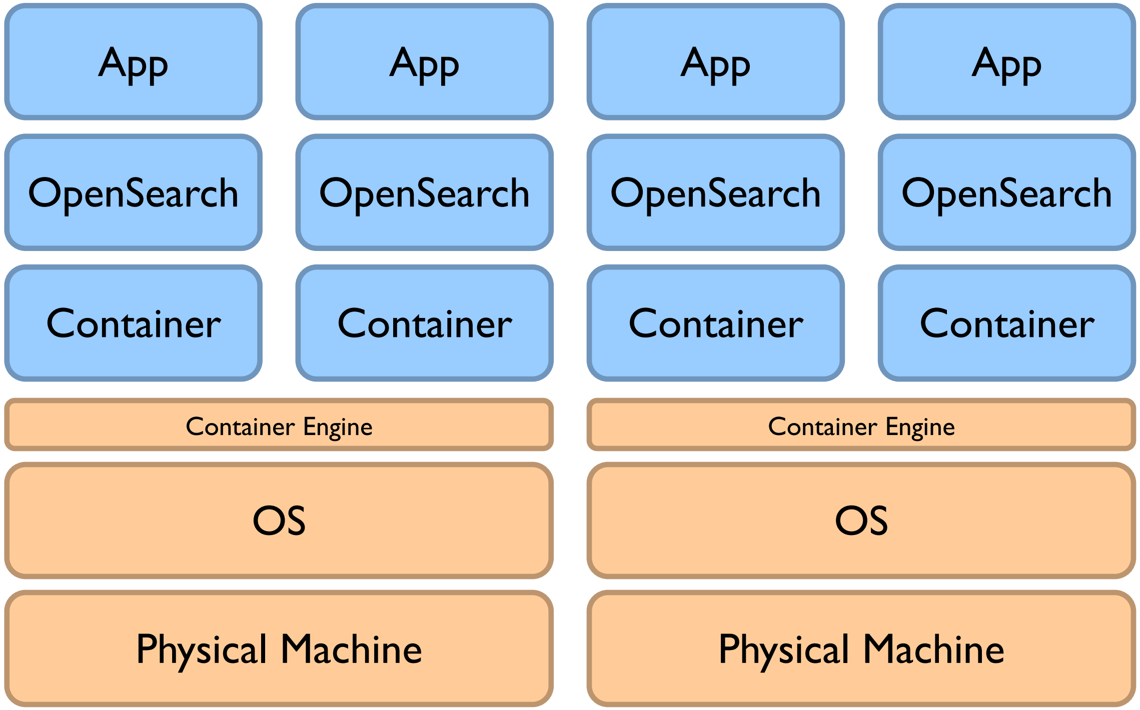
Container

Container Engine

OS

Physical Machine

Multi-Container Orchestration



App

App

App

App

OpenSearch

OpenSearch

OpenSearch

OpenSearch

Container

Container

Container

Container

Container Engine

Container Engine

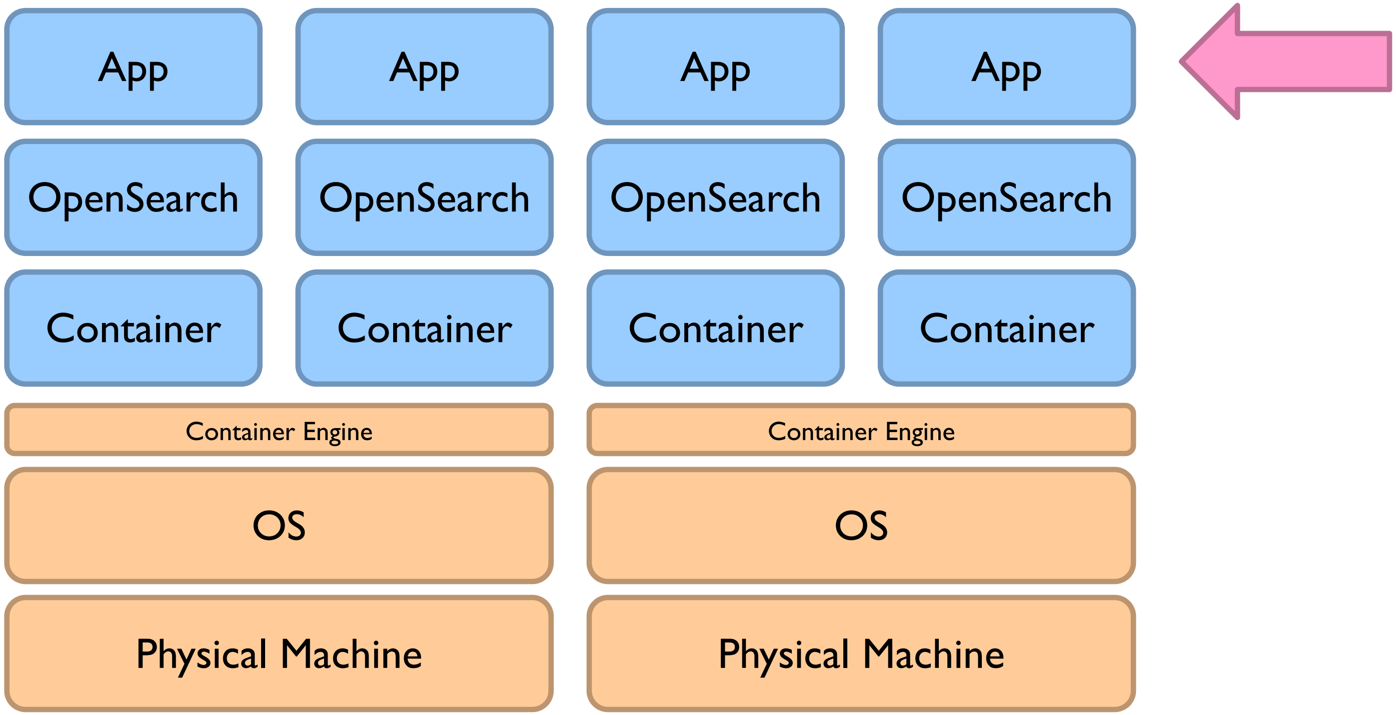
OS

OS

Physical Machine

Physical Machine

Source: Flickr (purplelight/14754940092/)



App App App App

OpenSearch OpenSearch OpenSearch OpenSearch

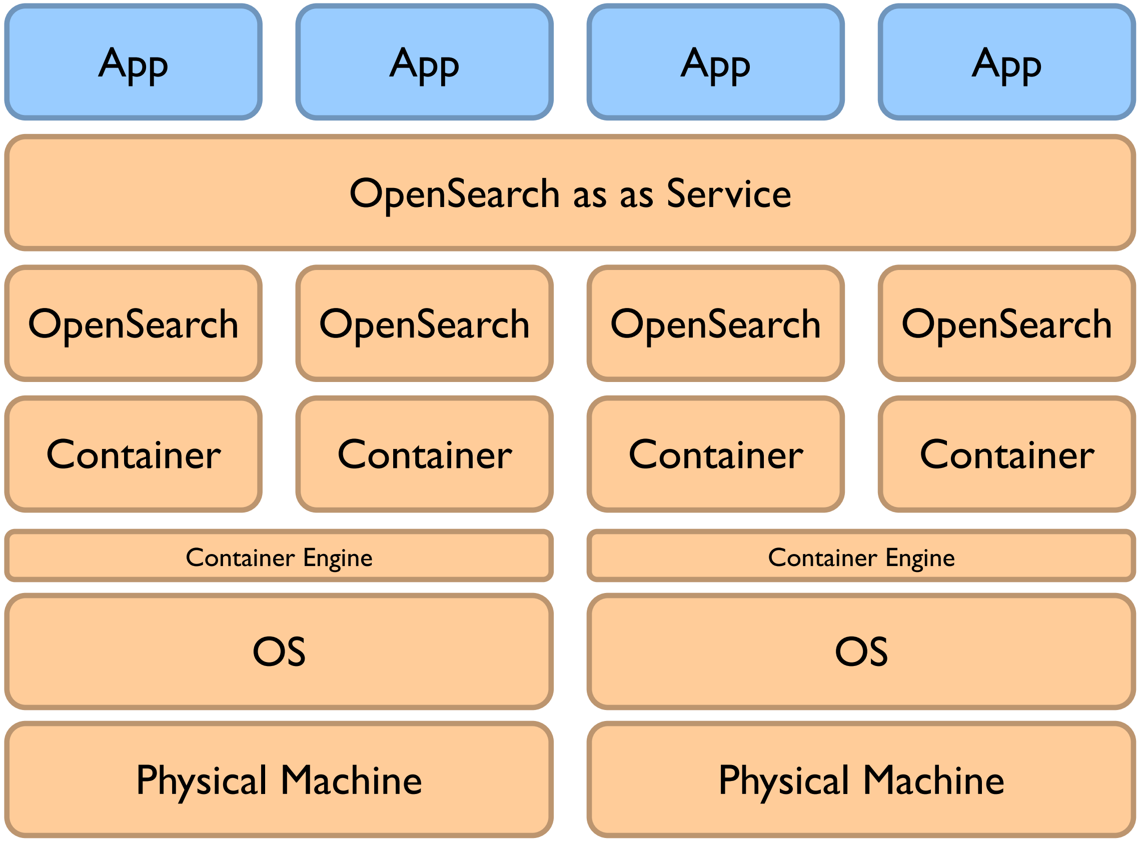
Container Container Container Container

Container Engine Container Engine

OS OS

Physical Machine Physical Machine

Platform as a Service



App

App

App

App

OpenSearch as as Service

OpenSearch

OpenSearch

OpenSearch

OpenSearch

Container

Container

Container

Container

Container Engine

Container Engine

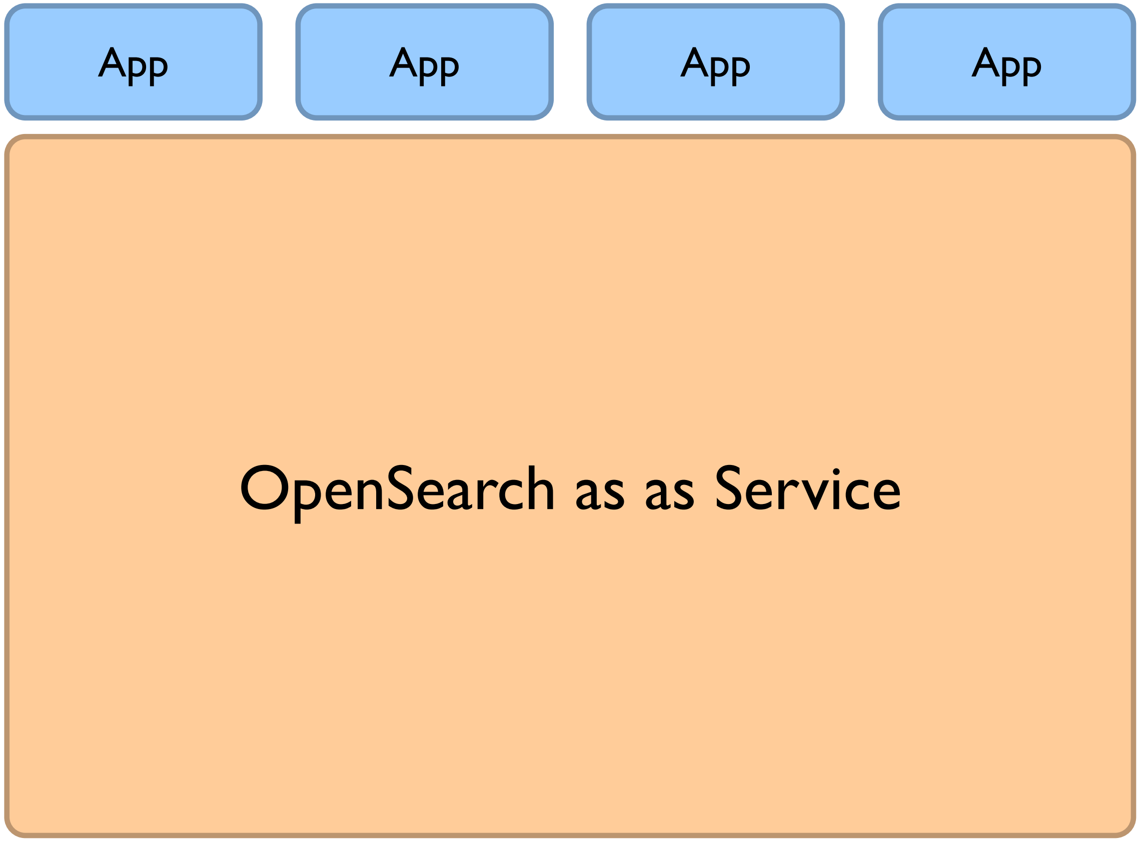
OS

OS

Physical Machine

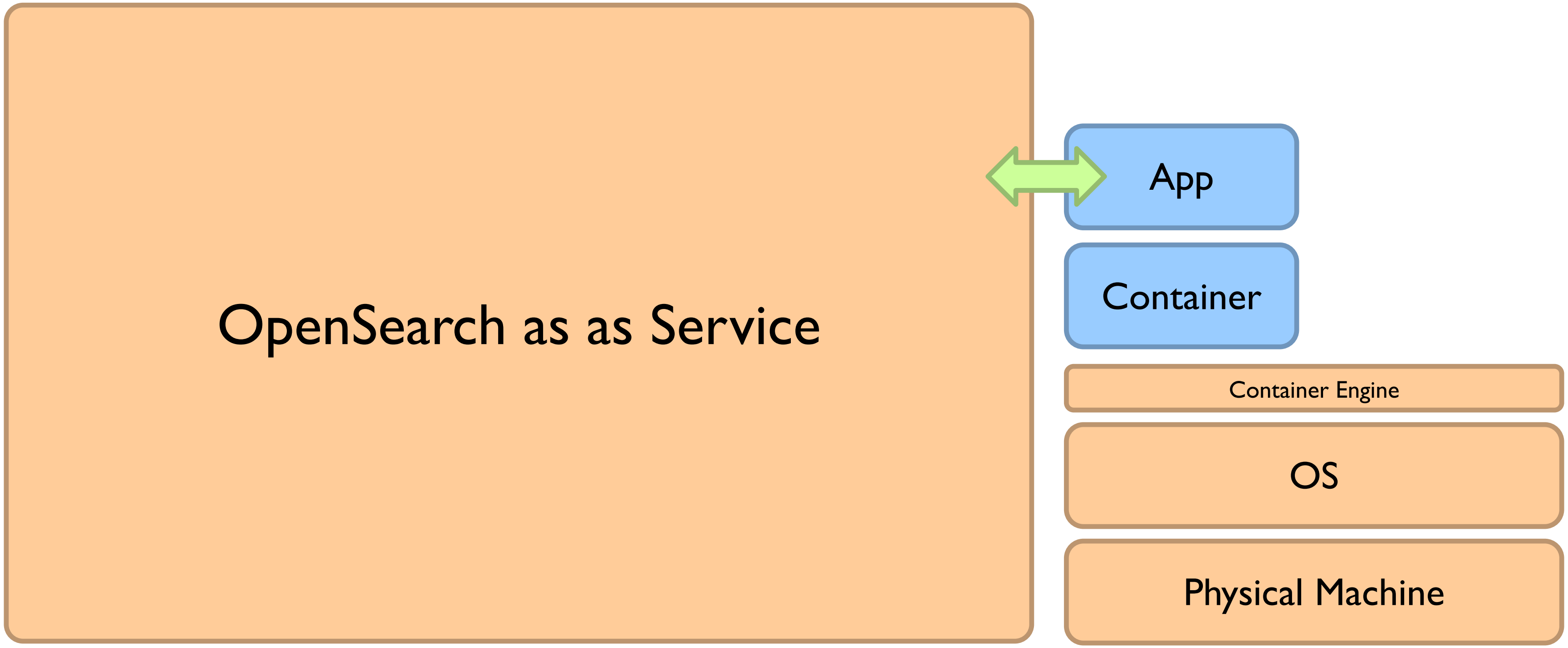
Physical Machine

Platform as a Service

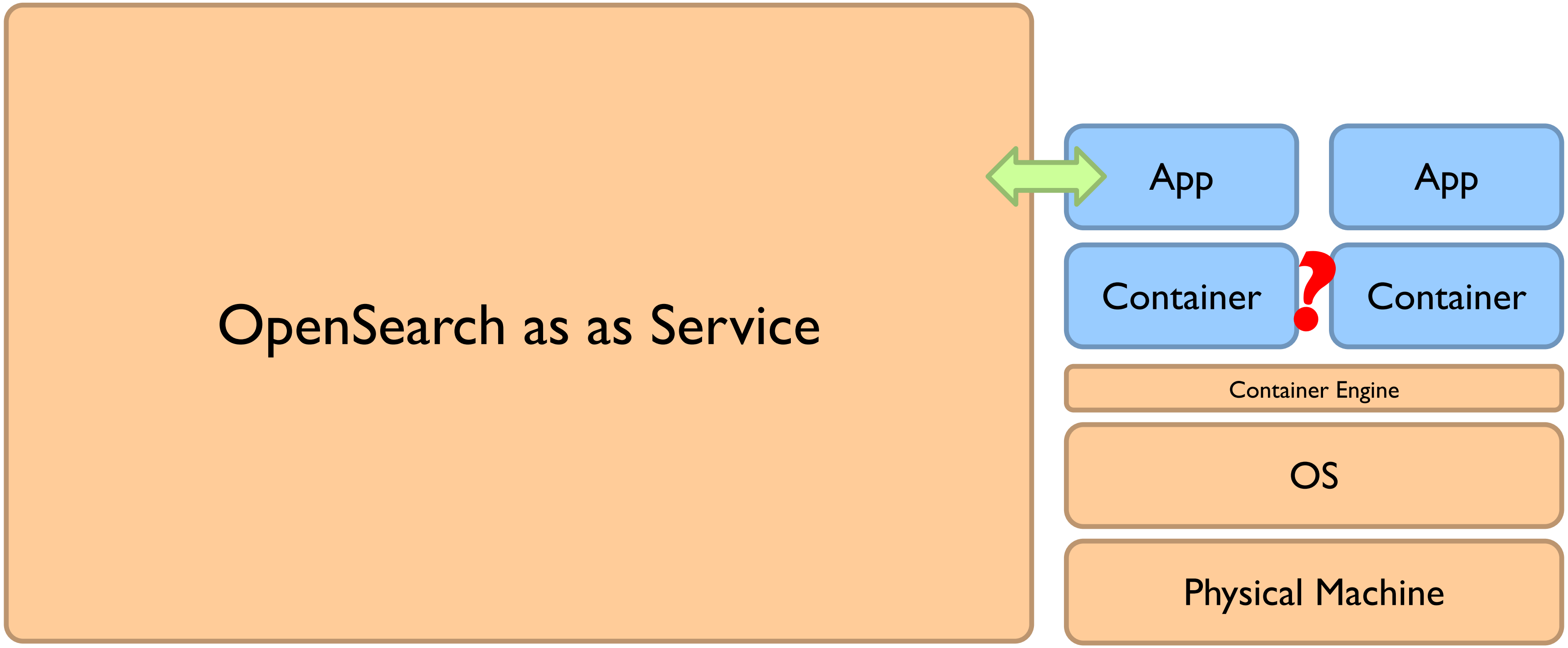


Service-level agreements (SLA) in terms
of latency, capacity, scalability, etc.

What about the apps?



Scaling out the apps…



OpenSearch as as Service

App

App

Container

Container

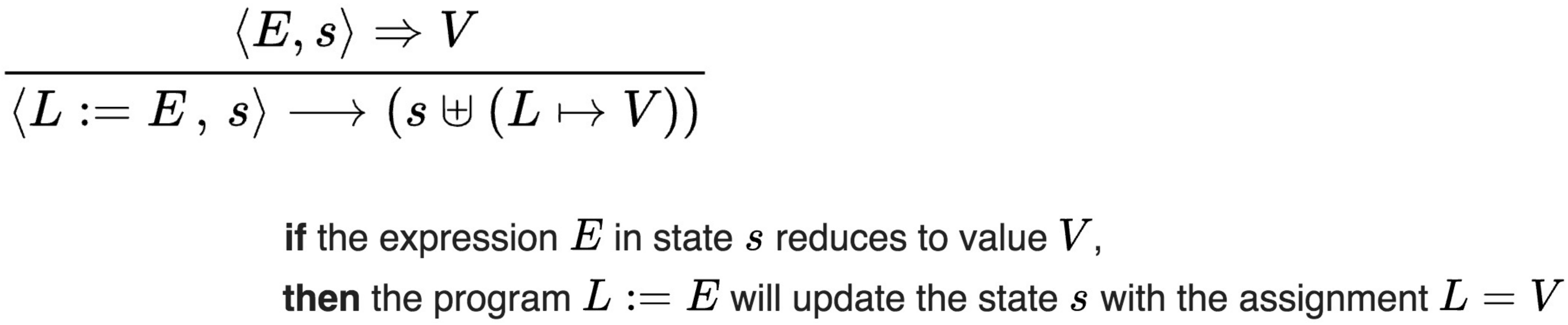
Container Engine

OS

Physical Machine

Source: Flickr (purplelight/14754940092/)

Operational Semantics of Computing…



$$\frac{\langle E, s\rangle \Rightarrow V}{\langle L := E\,,\ s\rangle \longrightarrow (s \uplus (L \mapsto V))}$$

if the expression E in state s reduces to value V,

then the program $L := E$ will update the state s with the assignment $L = V$

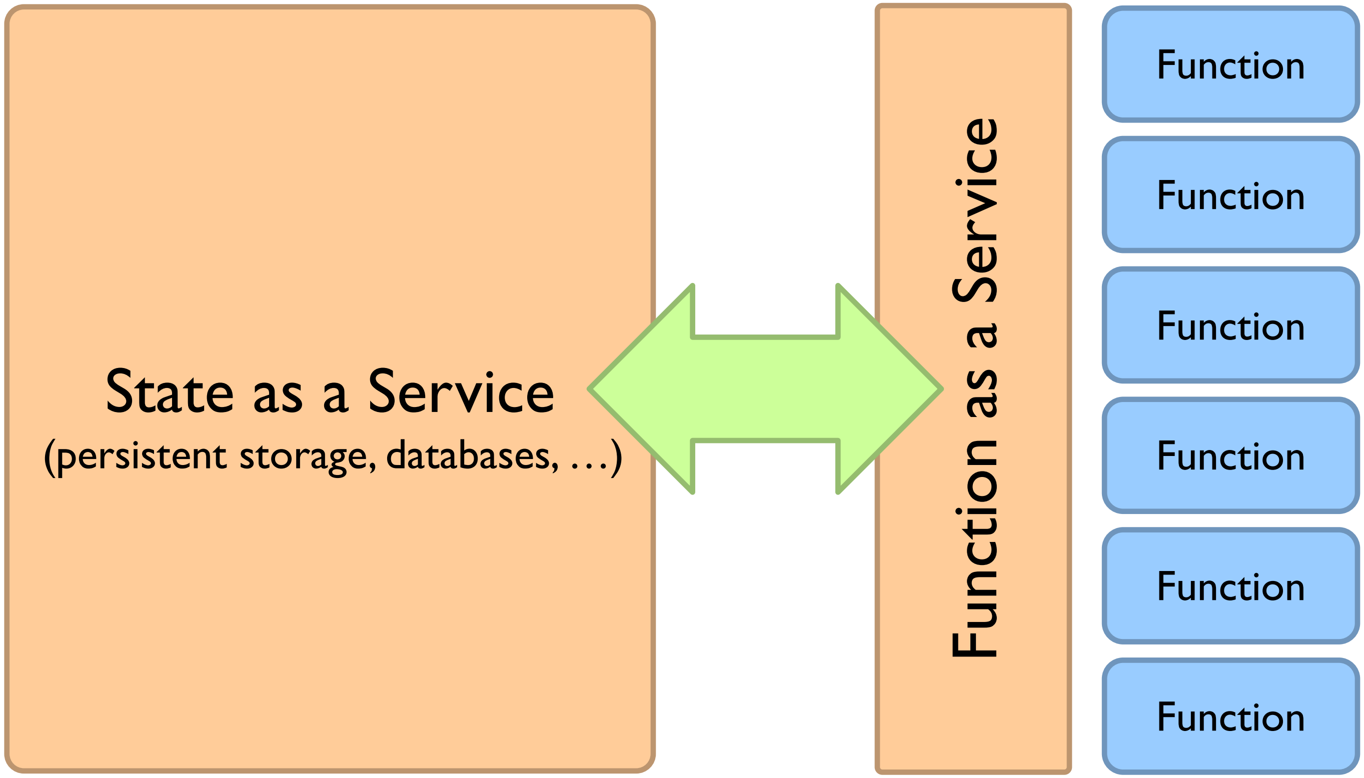
State	State as a Service (persistent storage, databases, message queues, …)
Transitions	Function as a Service (blocks of code with a well-defined entry and exit points)

Source: Wikipedia

Computing without Servers

Developer: Write a bunch of functions

typically – read state, perform some computation, update state



Cloud provider handles everything else!

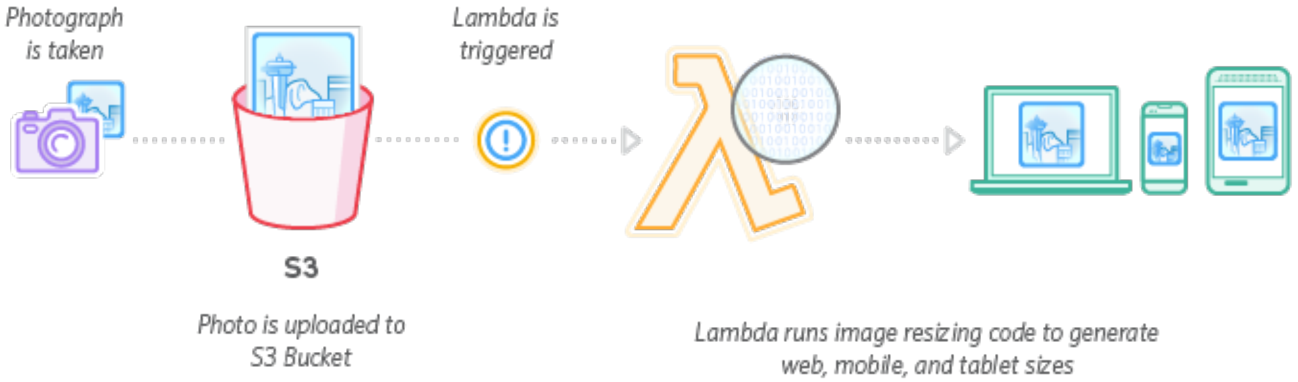
allocation of resources for execution, scaling up and down, load balancing, cleaning up, etc.

Cost model: pay per function invocation

AWS
Lambda

Serverless Examples

Example: *Image Thumbnail Creation*



Photograph is taken

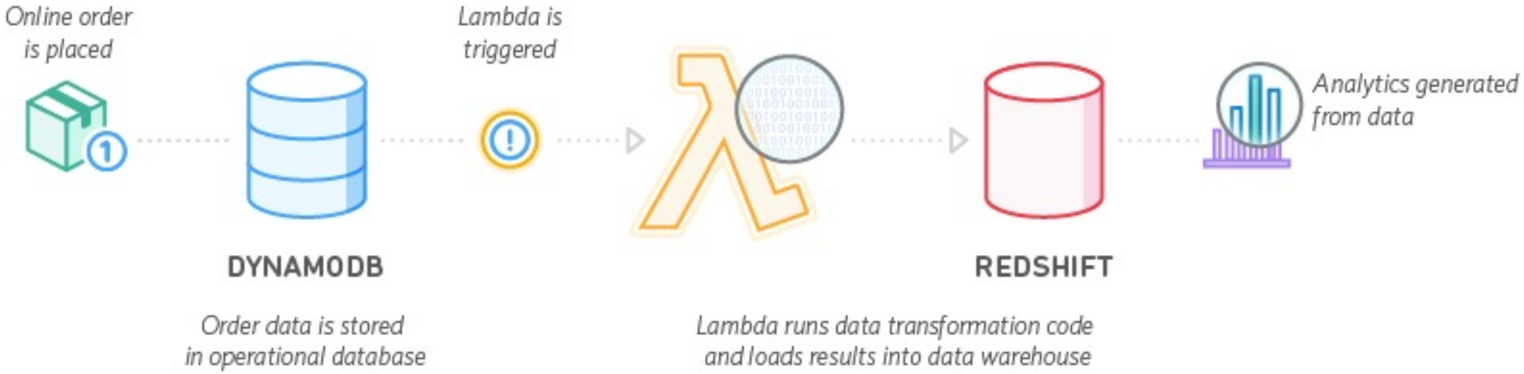
S3

Photo is uploaded to S3 Bucket

Lambda is triggered

Lambda runs image resizing code to generate web, mobile, and tablet sizes

Example: *Retail Data Warehouse ETL*



Online order is placed

DYNAMODB

Order data is stored in operational database

Lambda is triggered

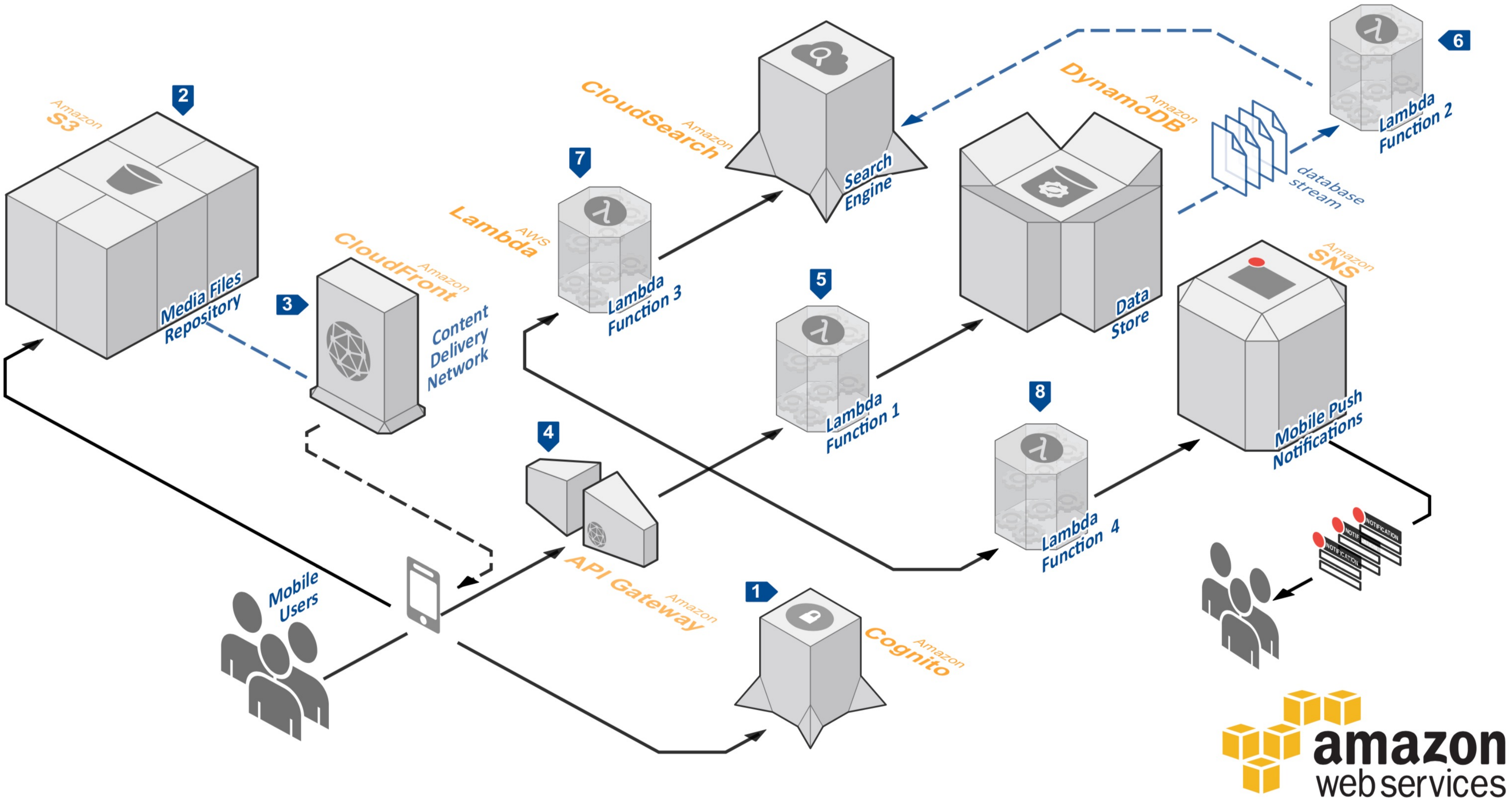
Lambda runs data transformation code and loads results into data warehouse

REDSHIFT

Analytics generated from data

Source: Amazon Web Services

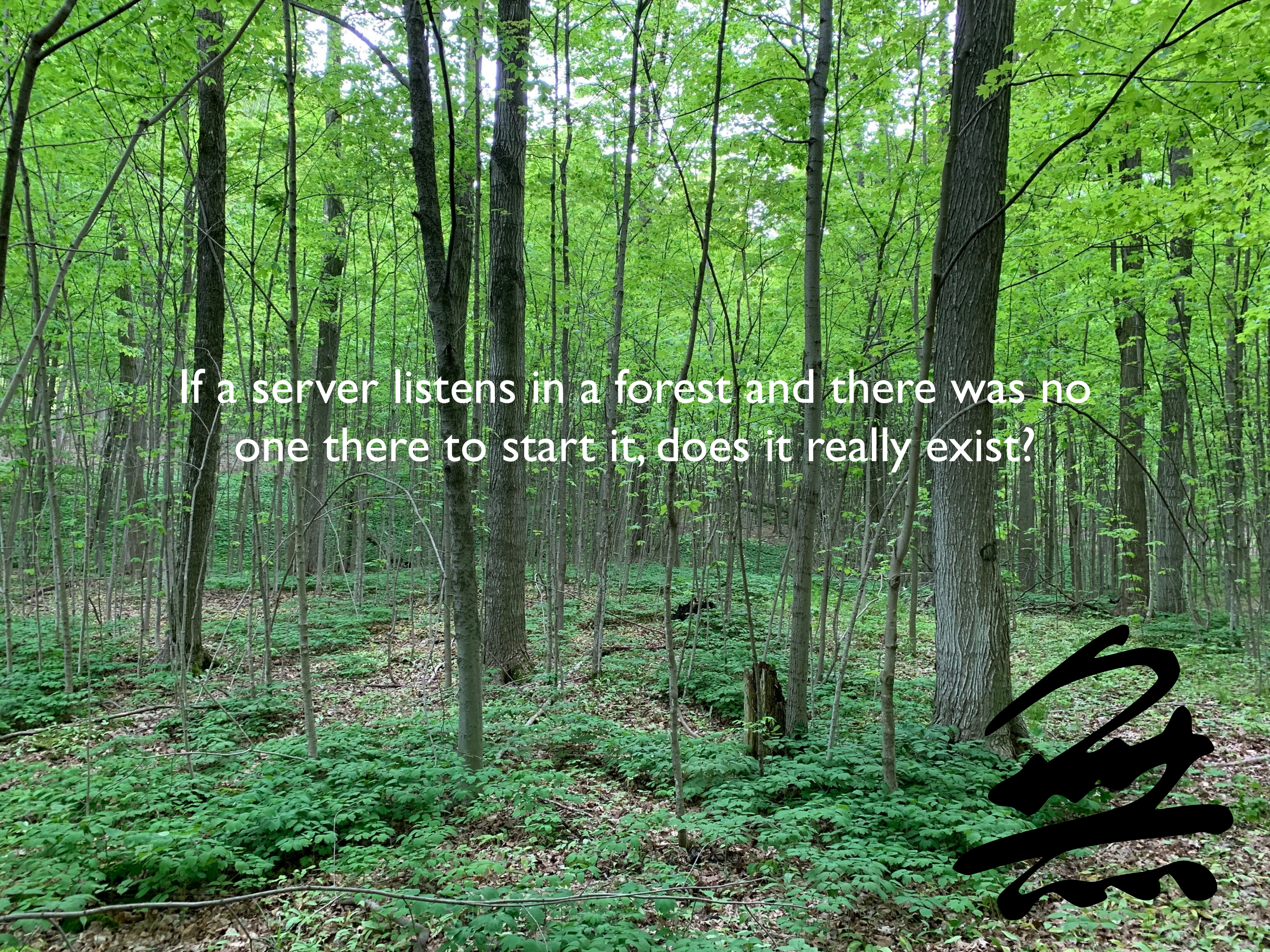
Serverless Examples



Source: Amazon Web Services

Serverless computing isn't actually computing without servers…

It's just that servers become someone else's problem!

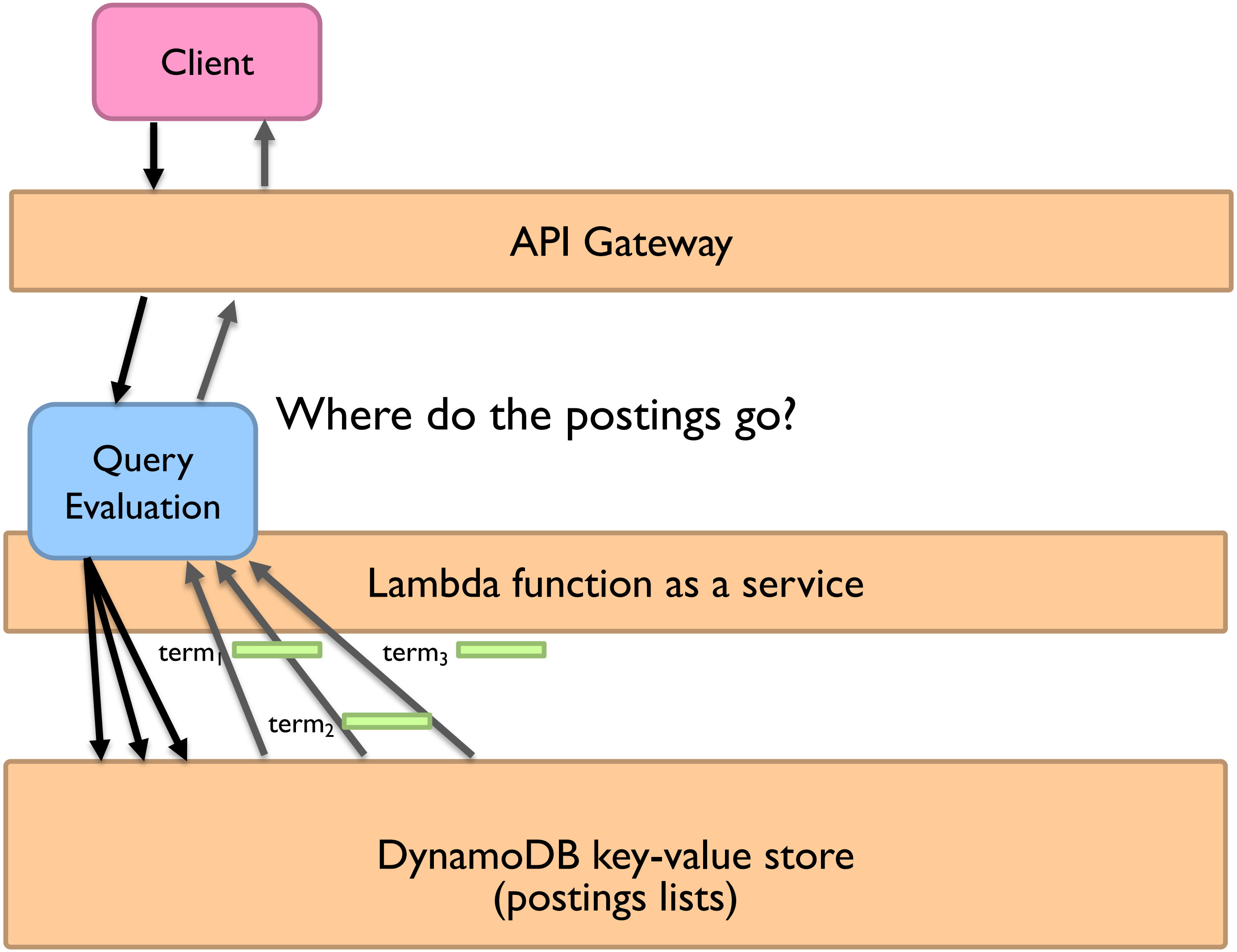


If a server listens in a forest and there was no one there to start it, does it really exist?

[redacted]

What would a serverless search engine look like?

Serverless Search



Client

API Gateway

Query Evaluation

Where do the postings go?

Lambda function as a service

$term_1$

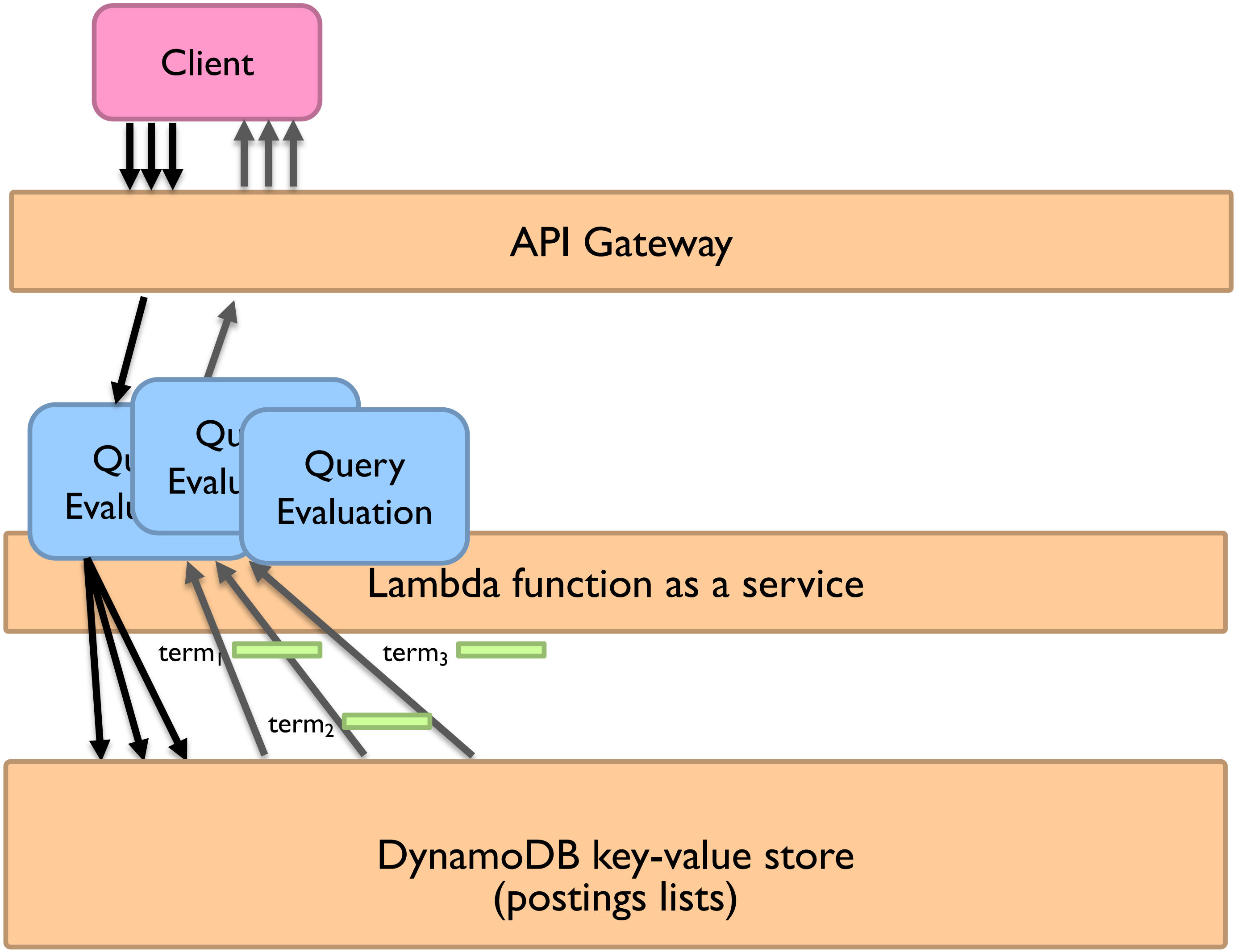
$term_2$

$term_3$

DynamoDB key-value store
(postings lists)

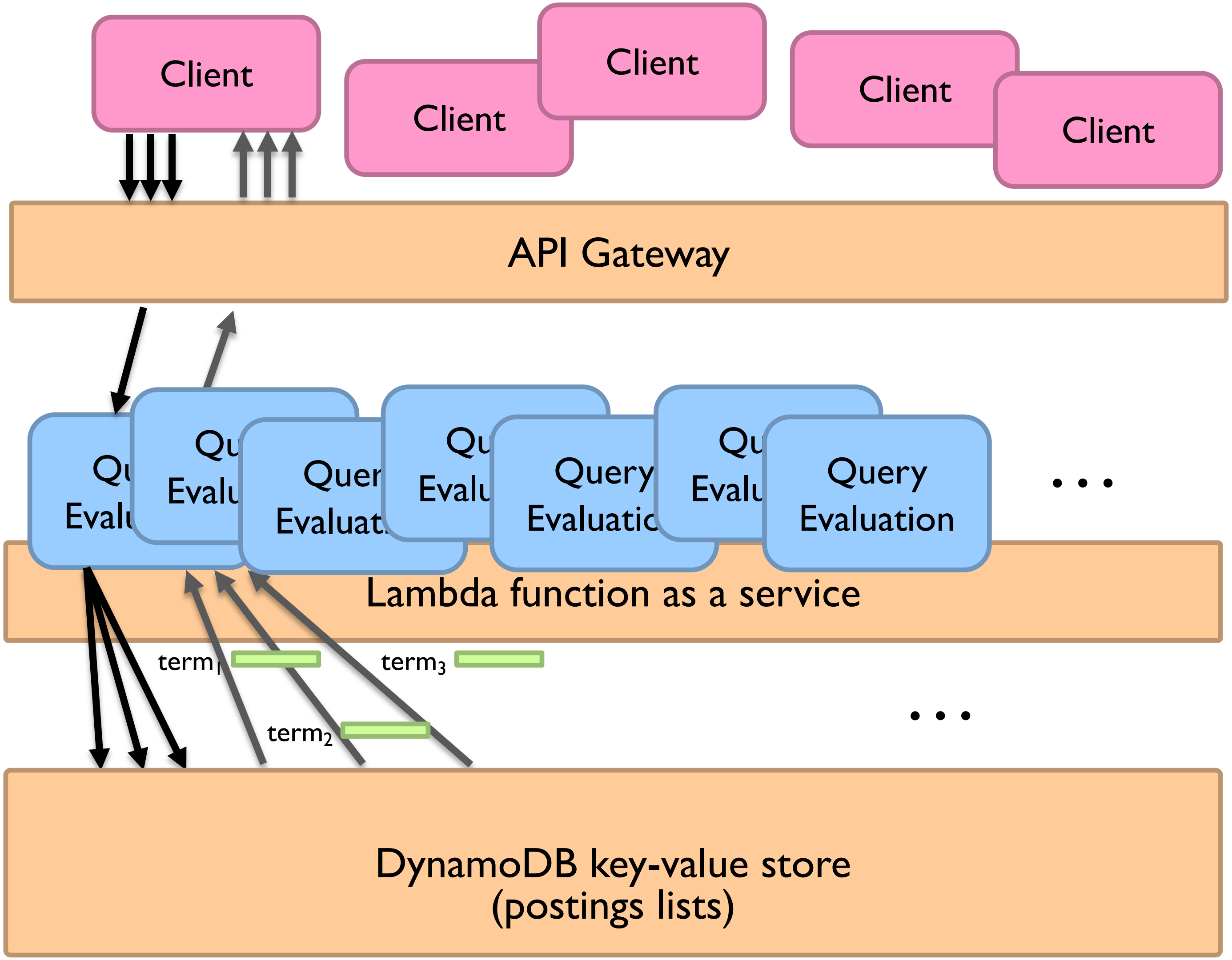
Matt Crane and Jimmy Lin. An Exploration of Serverless Architectures for Information Retrieval. *ICTIR 2017.*

Serverless Search

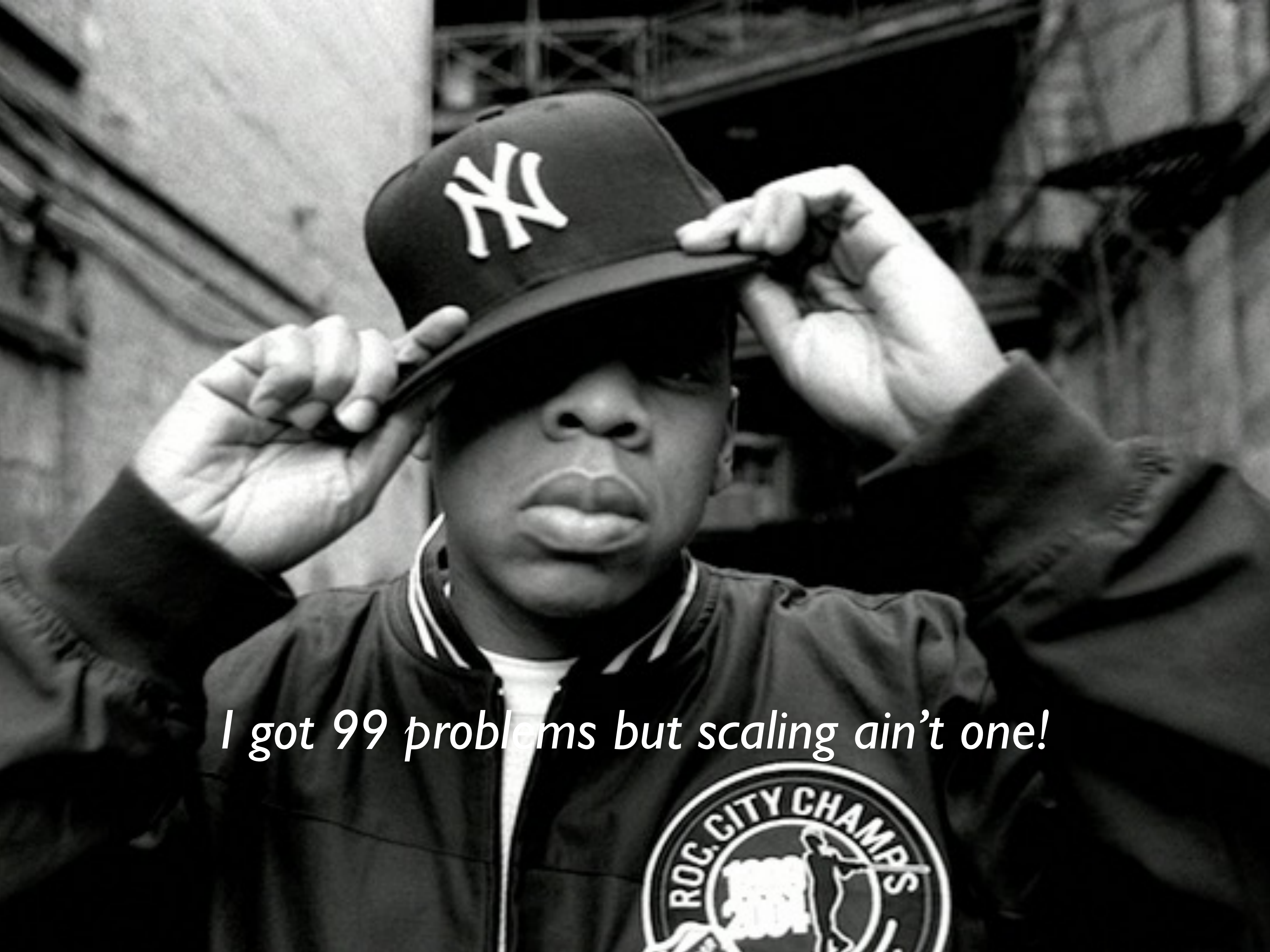


Matt Crane and Jimmy Lin. An Exploration of Serverless Architectures for Information Retrieval. *ICTIR 2017.*

Serverless Search



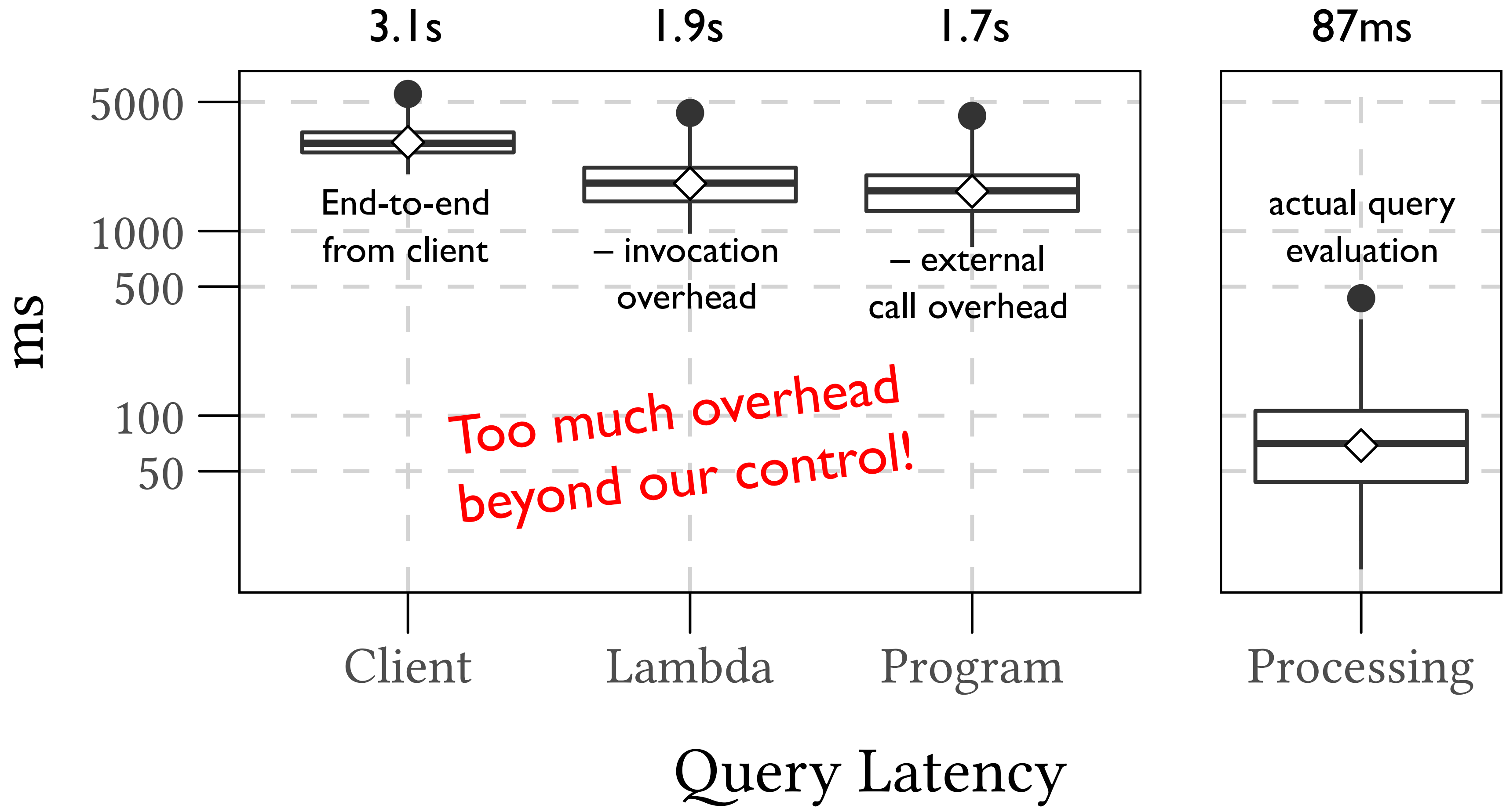
Matt Crane and Jimmy Lin. An Exploration of Serverless Architectures for Information Retrieval. *ICTIR 2017.*



I got 99 problems but scaling ain’t one!

How well does serverless search work?

tl; dr – *not very well…*



Gov2 collection (25m docs), topic 701-850

Matt Crane and Jimmy Lin. An Exploration of Serverless Architectures for Information Retrieval. *ICTIR 2017.*

Take 2

(This work)

Technical Highlights

tl;dr – Serverless Lucene

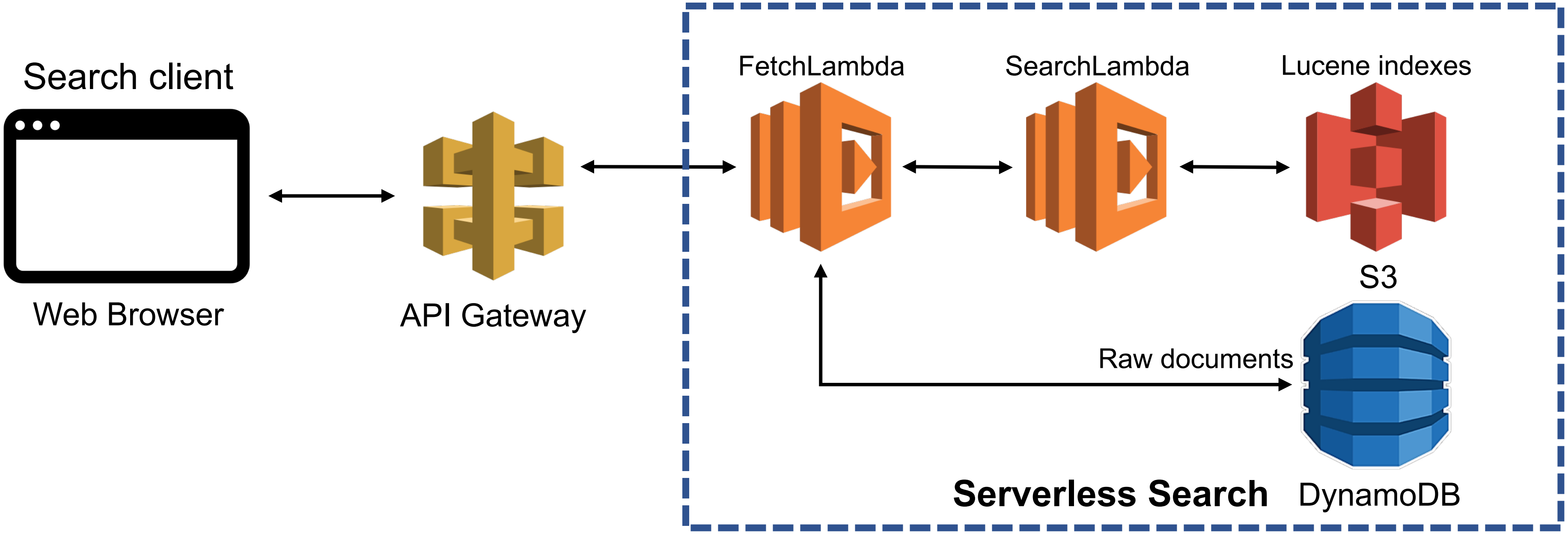
Index structures stored on S3
Query evaluation in AWS Lambda

Minimal modifications to "vanilla" Lucene!

What about query latency?

"cold instance" startup – loads indexes in memory
"warm instance" execution – indexes already in memory

No different from "standard" in-memory search!

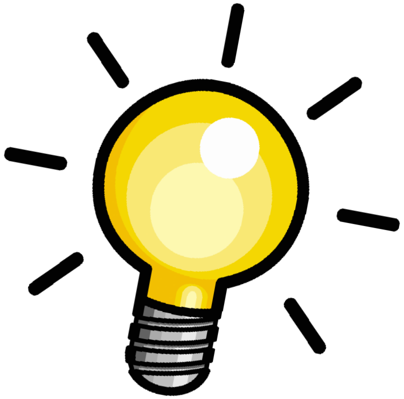

Search client
Web Browser
API Gateway
FetchLambda
SearchLambda
Lucene indexes
S3
Raw documents
DynamoDB
Serverless Search

Serverless Lucene isn’t enough!

Retrieve + Rerank

Select some promising texts

Rerank selected texts



Lucene

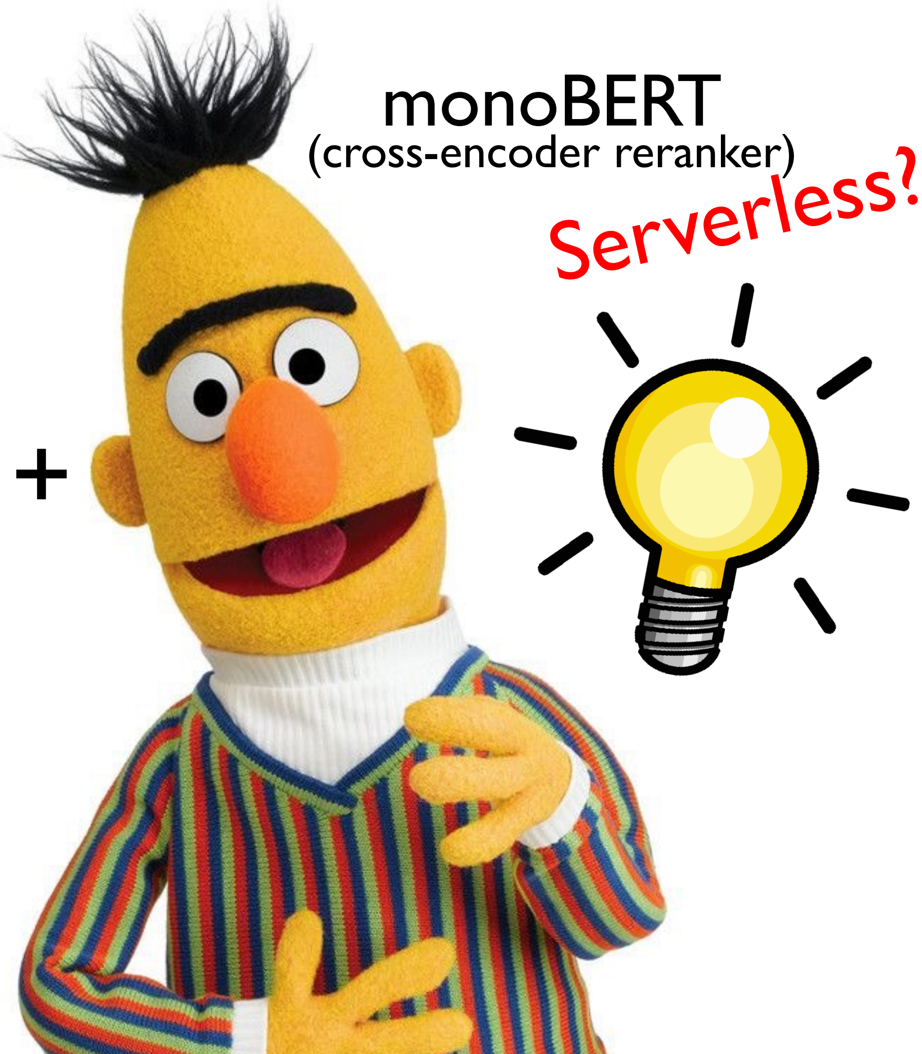
Retrieve + Rerank

Select some promising texts

Serverless!

Lucene

+



monoBERT
(cross-encoder reranker)

Serverless?

Yea, so what about serverless BERT?

(Good thing it's embarrassingly parallel!)

Technical Highlights

tl;dr – Serverless BERT inference in AWS Lambda

"Early-Exit" optimizations* to reduce inference latency

Main technical challenge is model size – solved by Elastic Container Registry

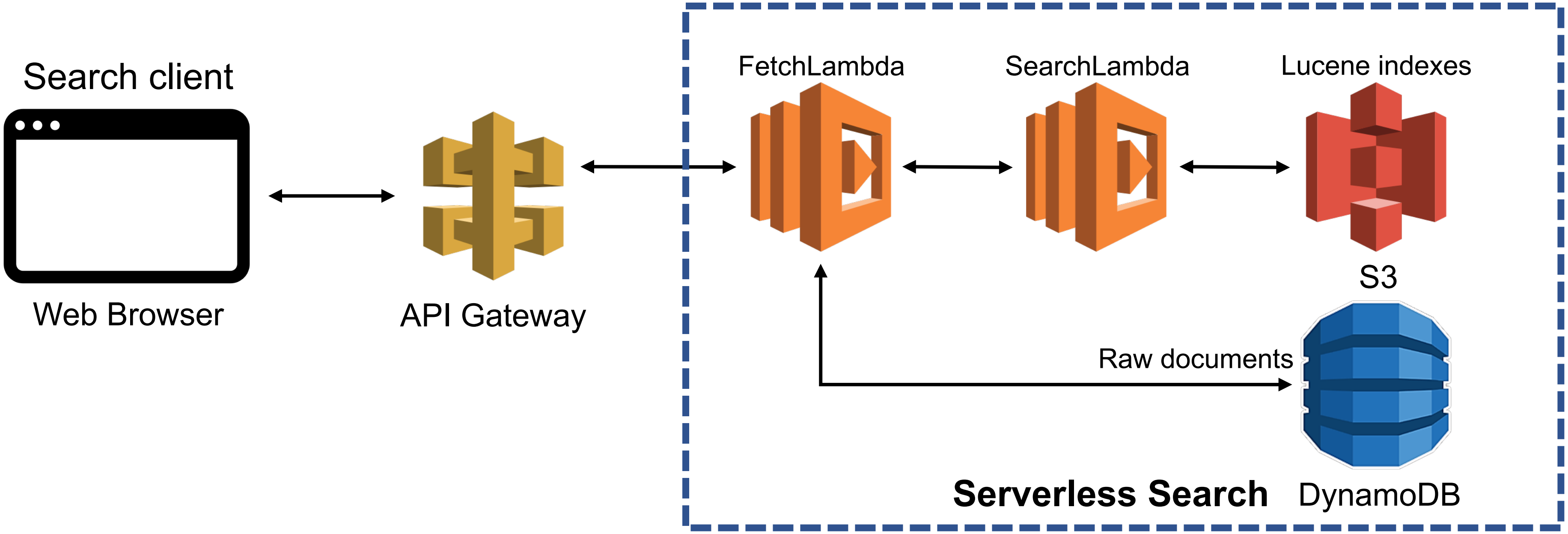
It's SMOP!

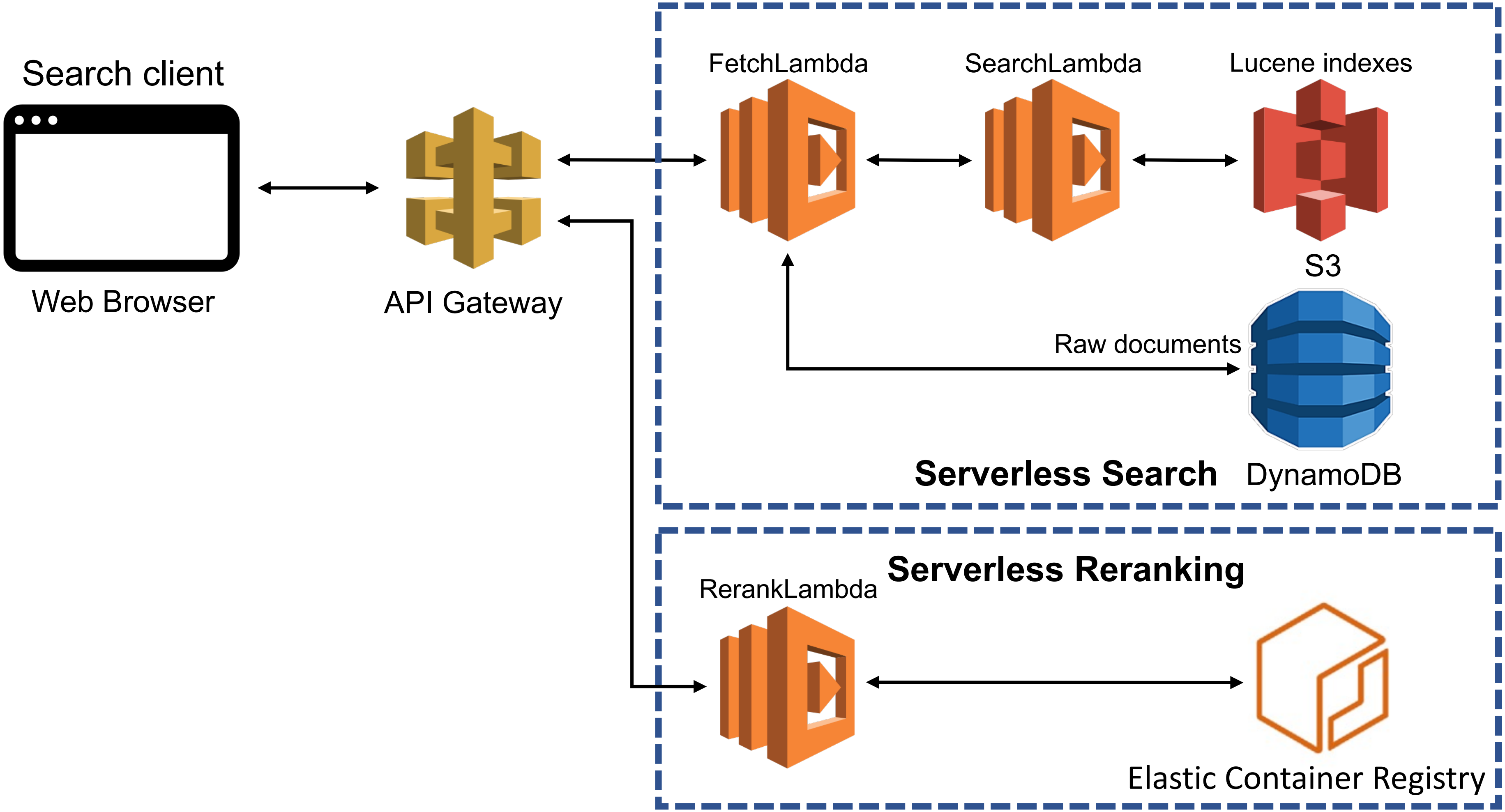
The bad: CPU inference

The good: Massive parallelism (100-way parallelism)

The ugly: Image packaging is a pain!

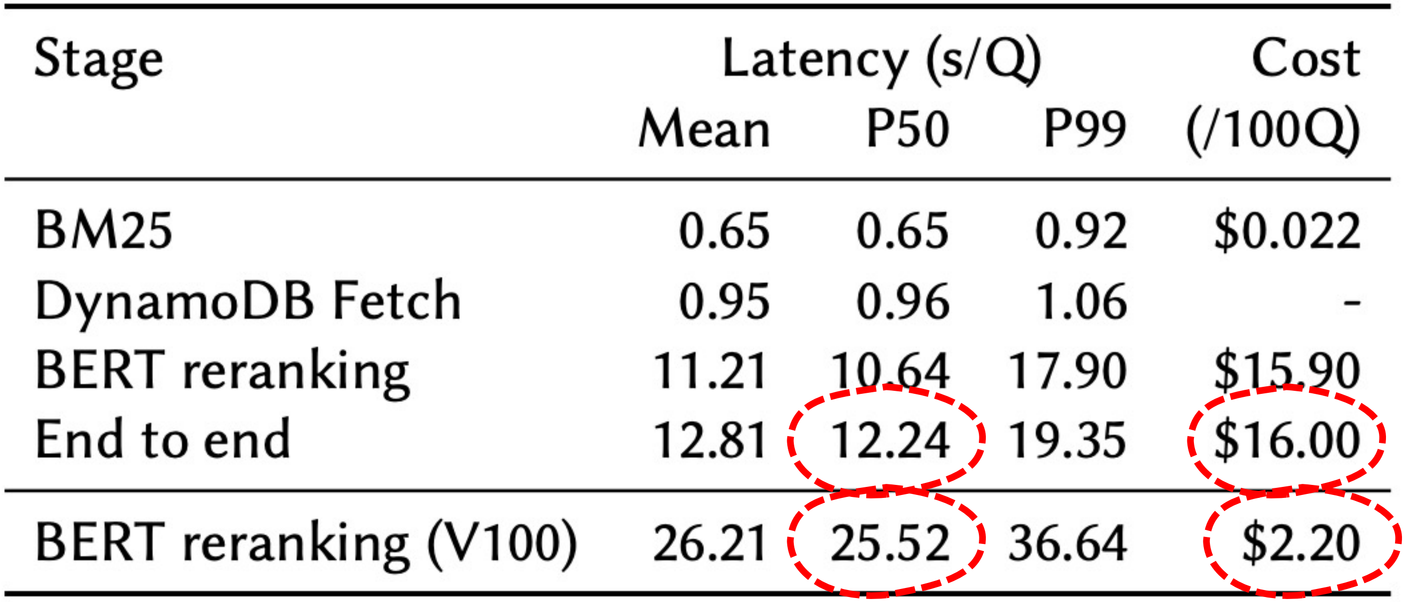
* Xin et al. Early Exiting BERT for Efficient Document Ranking. *SustaiNLP 2020*.


Search client
Web Browser
API Gateway
FetchLambda
SearchLambda
Lucene indexes
S3
Raw documents
DynamoDB
Serverless Search


Search client
Web Browser
API Gateway
FetchLambda
SearchLambda
Lucene indexes
S3
Raw documents
DynamoDB
Serverless Search
Serverless Reranking
RerankLambda
Elastic Container Registry

Results
(MS MARCO passage ranking)

Setup: reranking 1000 hits from Lucene with monoBERT



Stage	Latency (s/Q)			Cost
	Mean	P50	P99	(/100Q)
BM25	0.65	0.65	0.92	$0.022
DynamoDB Fetch	0.95	0.96	1.06	-
BERT reranking	11.21	10.64	17.90	$15.90
End to end	12.81	12.24	19.35	$16.00
BERT reranking (V100)	26.21	25.52	36.64	$2.20

(We confirmed that effectiveness is the same for serverless vs. server-based deployments)

Wait, how can CPU be faster than GPU?

7-8× more expensive = breakeven at 85%-90% idle

Objections

Source: https://www.flickr.com/photos/usdagov/22484527807/

It’s still too slow! (12s end-to-end)
It’s still too expensive! ($0.16 per query)

Agreed… but this is only the beginning!

Serverless infrastructure will become more efficient.

Lots of neural inference acceleration techniques to try.

The price of minimal management overhead and infinite scaling?

Serverless search is worth considering?!

Questions?