Must Serverless = Stateless?

Anurag Khandelwal
Assistant Professor, Computer Science Department
Yale University

WHY SHOULD YOU CARE?

WHY SHOULD YOU CARE?

Mainstream interest triggered by AWS Lambda service in 2014

SERVERLESS

WHY SHOULD YOU CARE?

Mainstream interest triggered by AWS Lambda service in 2014

Real-time monitoring	Interactive	Log processing	IoT
Streaming data	Real-time	Data	Event-driven

SERVERLESS

WHY SHOULD YOU CARE?

Mainstream interest triggered by AWS Lambda service in 2014

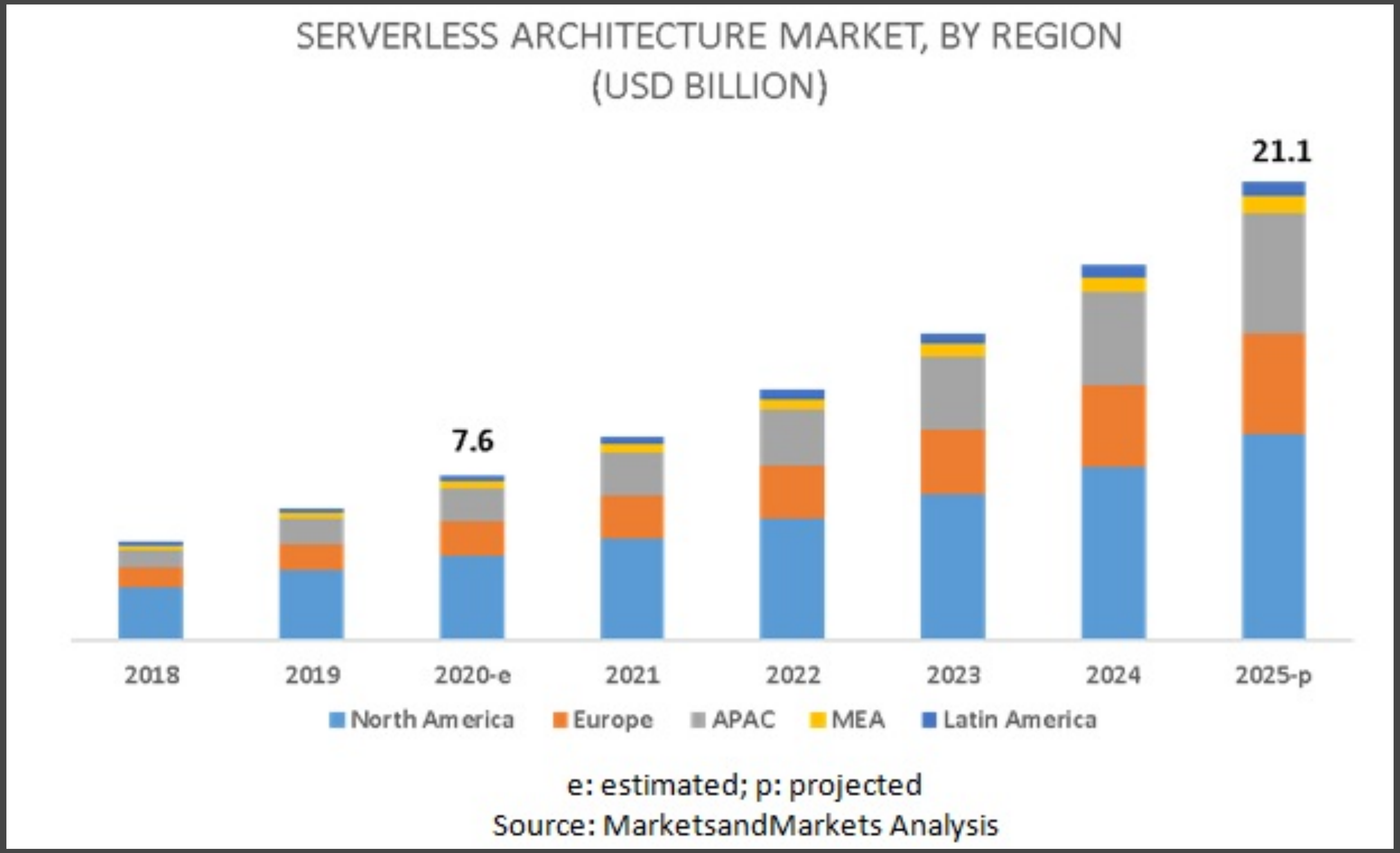
Real-time monitoring
Interactive
Log processing
IoT
Streaming data
Real-time
Data
Event-driven

SERVERLESS

All major cloud players now offer a wide range of serverless platforms

WHY SHOULD YOU CARE?

Mainstream interest triggered by AWS Lambda service in 2014

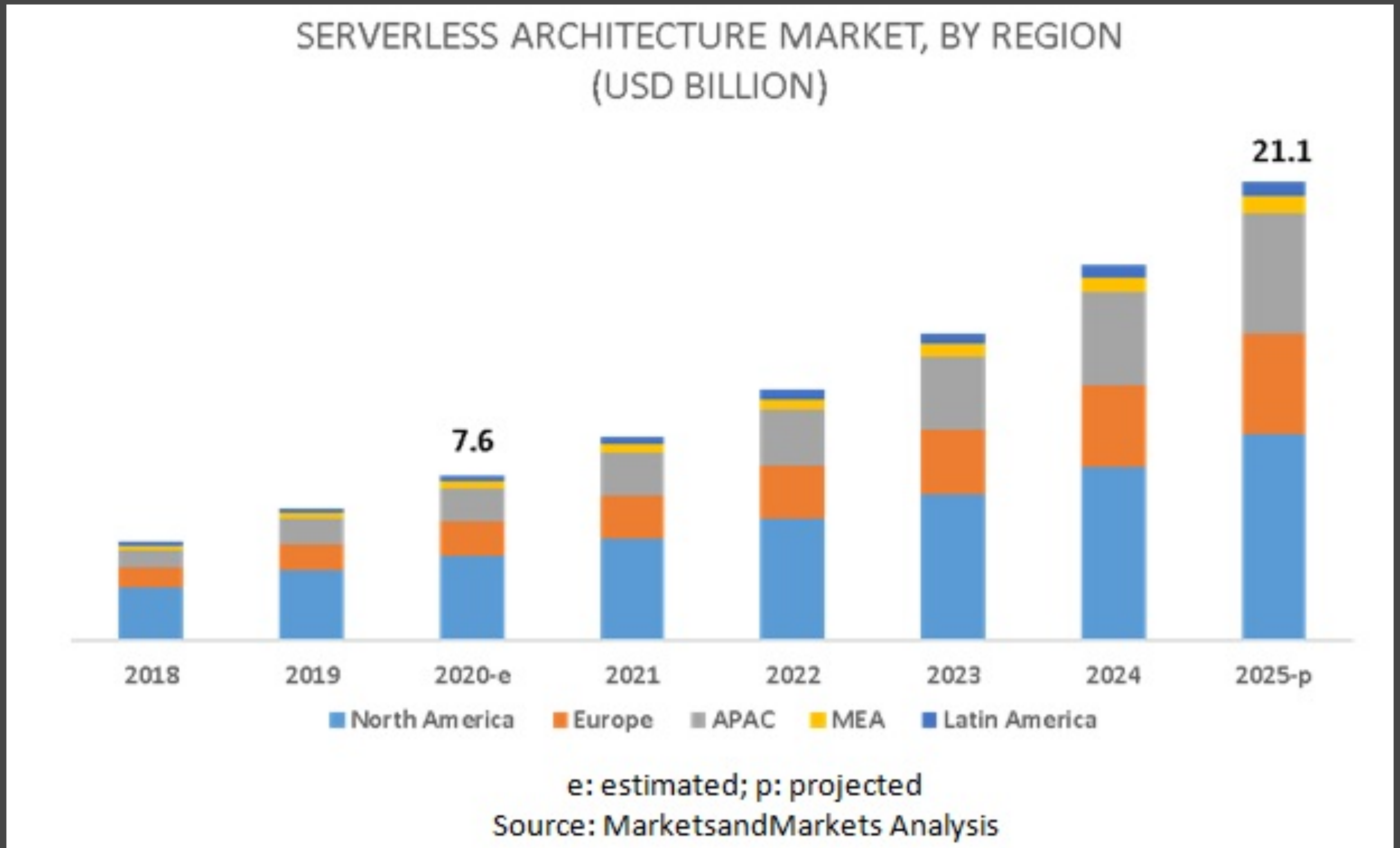


All major cloud players now offer a wide range of serverless platforms

... and the market is growing at a fast pace...

WHY SHOULD YOU CARE?

Mainstream interest triggered by AWS Lambda service in 2014



All major cloud players now offer a wide range of serverless platforms

... and the market is growing at a fast pace...

This talk:

- ✦ What & why of serverless
- ✦ Today's Serverless = Stateless
- ✦ Stateless → Stateful serverless

WHAT & WHY OF SERVERLESS

WHAT IS SERVERLESS?

WHAT IS SERVERLESS?

DEVELOPMENT

EASE OF USE

High-level language (e.g., Python, SQL)

Zero management (fault-tolerance, load-balancing)

WHAT IS SERVERLESS?

DEVELOPMENT

EASE OF USE

High-level language (e.g., Python, SQL)

Zero management (fault-tolerance, load-balancing)

COMPUTATION

DEMAND DRIVEN EXECUTION

Fine-grained resource elasticity

WHAT IS SERVERLESS?

DEVELOPMENT

EASE OF USE

High-level language (e.g., Python, SQL)

Zero management (fault-tolerance, load-balancing)

COMPUTATION

DEMAND DRIVEN EXECUTION

Fine-grained resource elasticity

BILLING

COST EFFICIENCY

Pay-as-you-go + fine-grained billing

SERVERLESS COMPUTING *SIMPLIFIES* CLOUD PROGRAMMING

SERVERLESS COMPUTING *SIMPLIFIES* CLOUD PROGRAMMING

Upload Code

Run at any scale

Pay for what you use

A REAL USE-CASE: HOW FINANCIAL ENGINES CUT COSTS 90% USING SERVERLESS[1]

- Financial Engines: Independent Investment Advisor
  - 9 million people across 743 companies, $1.8 trillion in assets

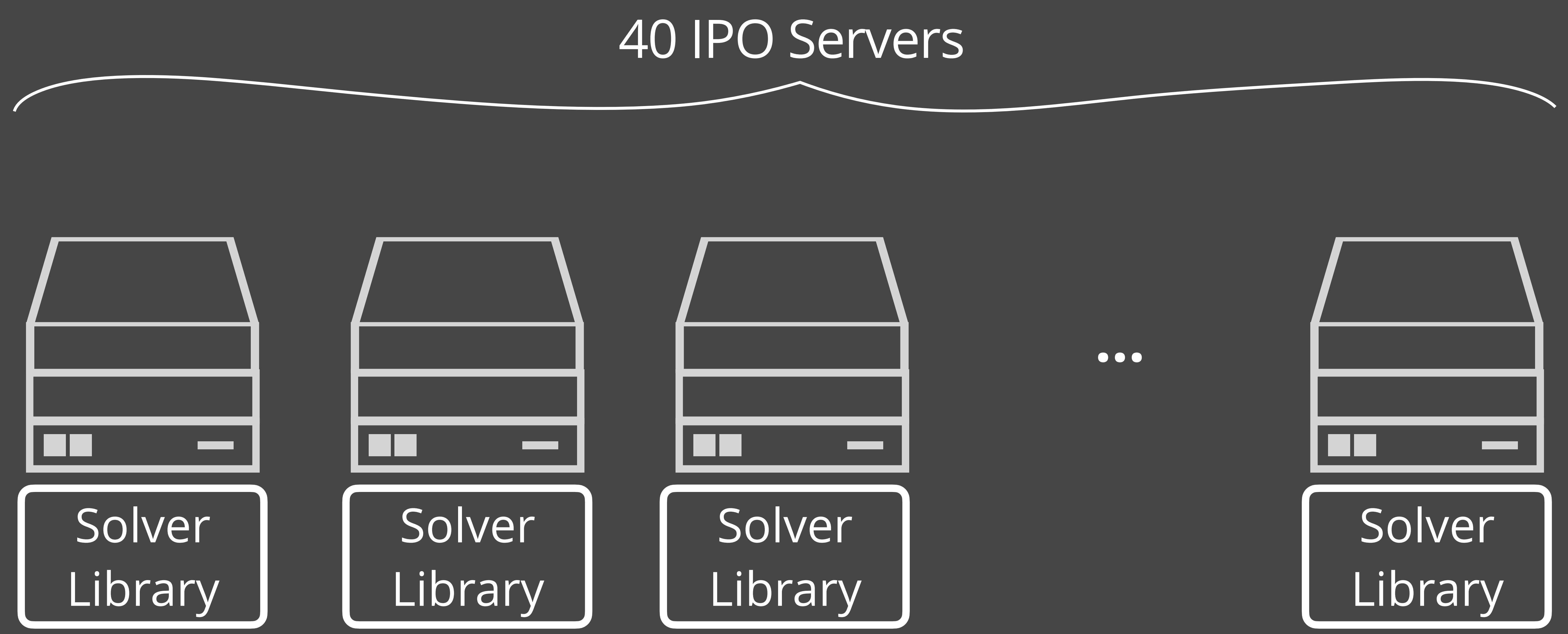
[1] Financial Engines Cuts Costs 90% Using AWS Lambda and Serverless Computing, https://aws.amazon.com/solutions/case-studies/financial-engines/

A REAL USE-CASE: HOW FINANCIAL ENGINES CUT COSTS 90% USING SERVERLESS[1]

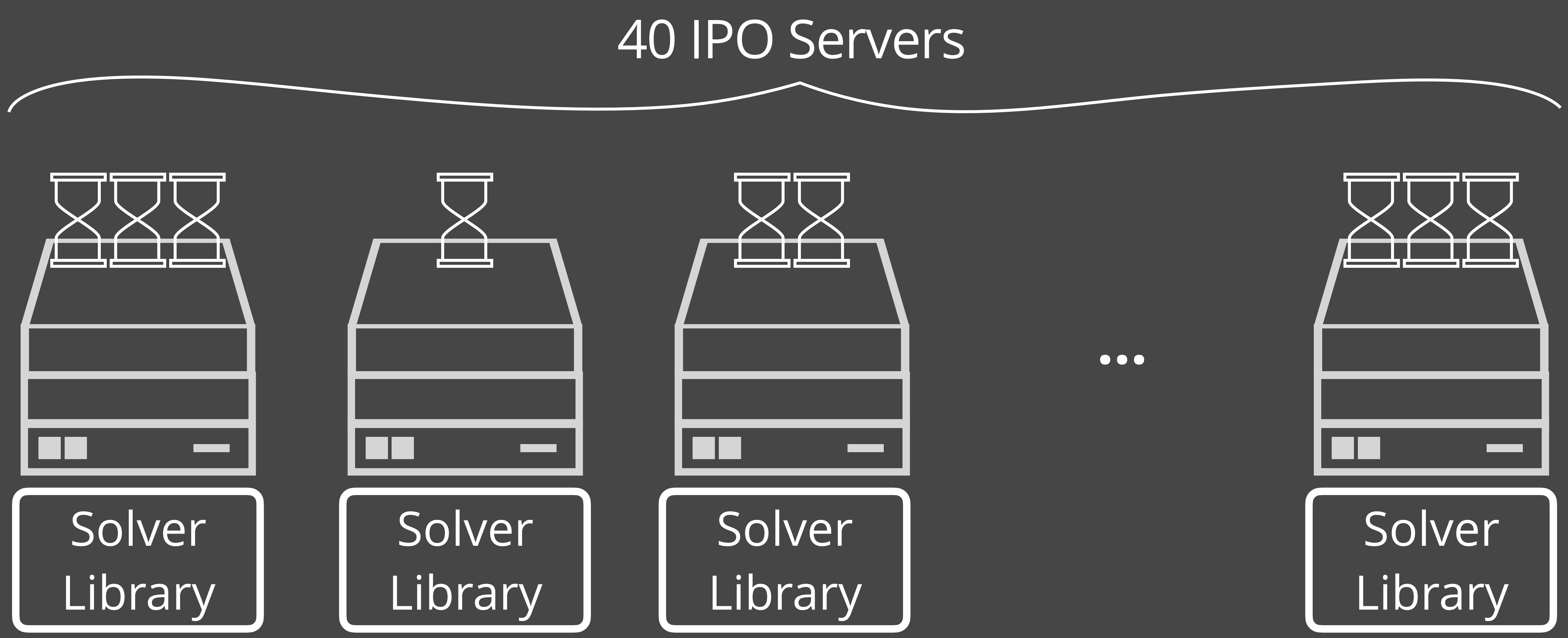
- Financial Engines: Independent Investment Advisor
  - 9 million people across 743 companies, $1.8 trillion in assets
- Automated portfolio management using computational engines
  - Core engine component: Integer programming optimizer (IPO)
  - Linear Programming to compute optimization/feasibility

[1] Financial Engines Cuts Costs 90% Using AWS Lambda and Serverless Computing, https://aws.amazon.com/solutions/case-studies/financial-engines/

IPO SERVER FARM

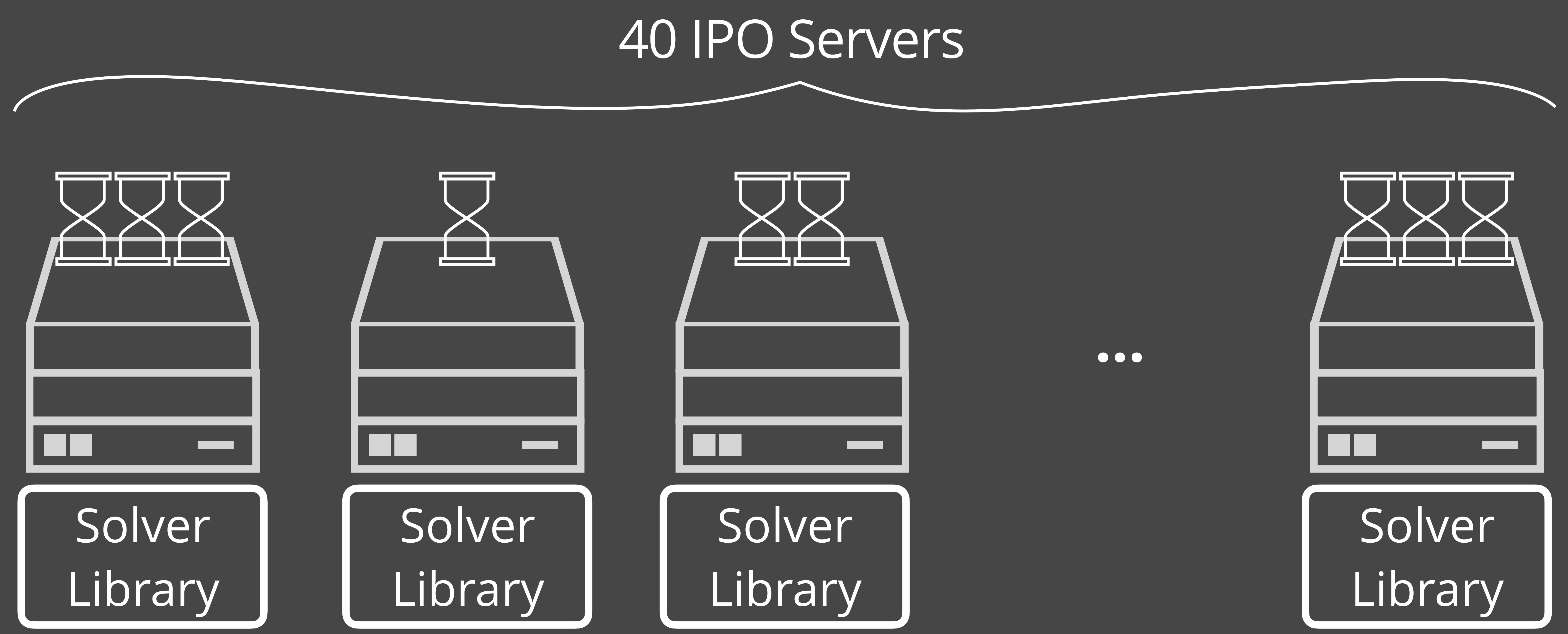


IPO SERVER FARM



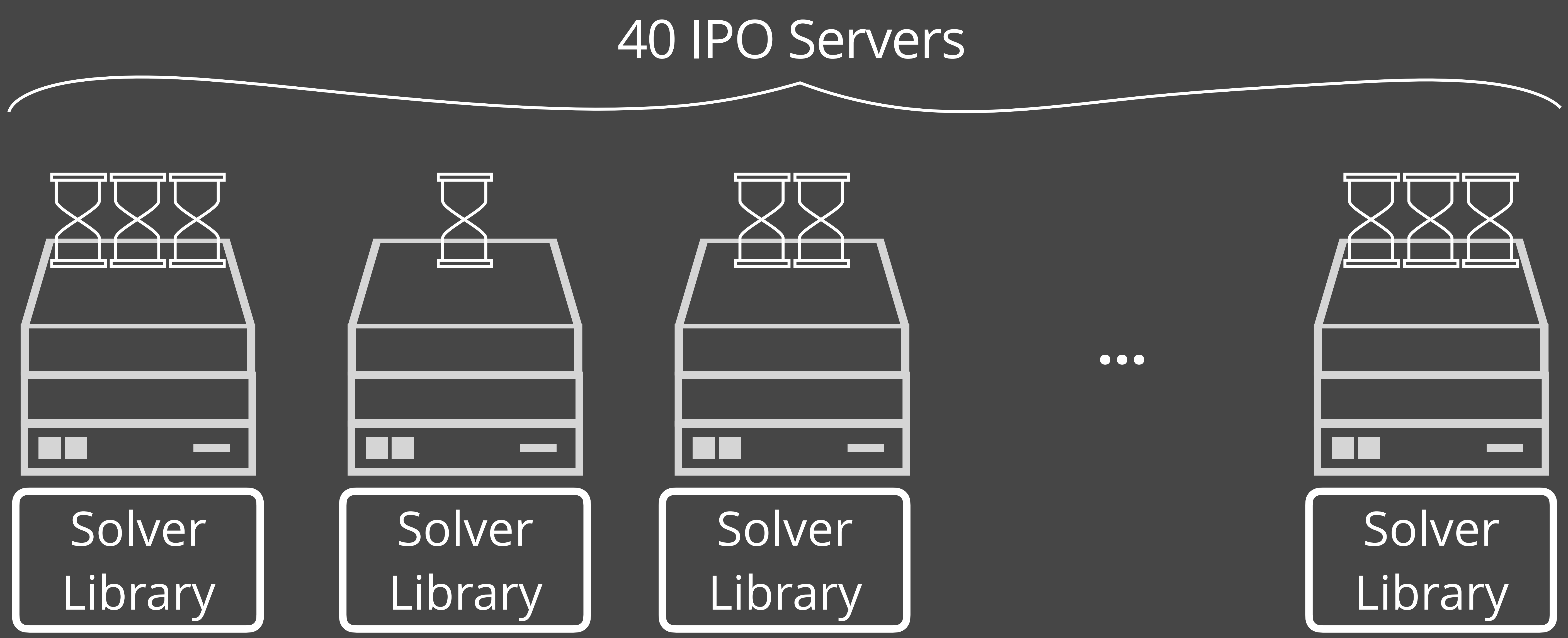
- IPO consumes > 30% of total CPU capacity

IPO SERVER FARM



- IPO consumes > 30% of total CPU capacity
  - Spikes of up to 1000 requests/s, 100ms per request

IPO SERVER FARM



- IPO consumes > 30% of total CPU capacity
  - Spikes of up to 1000 requests/s, 100ms per request
  - Capacity planning during marketing campaigns that produce large traffic spikes is hard...

NEED TO DO A LOT OF WORK…

NEED TO DO A LOT OF WORK…

- Scaling in response to load variations
- Request routing and load balancing
- Monitoring to respond to problems
- Provision servers based on budget, requirements
- System upgrades, including security patching
- Migration to new hardware as it becomes available

⋮

NEED TO DO A LOT OF WORK…

- Scaling in response to load variations
- Request routing and load balancing
- Monitoring to respond to problems
- Provision servers based on budget, requirements
- System upgrades, including security patching
- Migration to new hardware as it becomes available

Developer’s burden!

A REAL USE-CASE: HOW FINANCIAL ENGINES CUT COSTS 90% USING SERVERLESS[1]

λ

Solver Library

[1] Financial Engines Cuts Costs 90% Using AWS Lambda and Serverless Computing, https://aws.amazon.com/solutions/case-studies/financial-engines/

A REAL USE-CASE: HOW FINANCIAL ENGINES CUT COSTS 90% USING SERVERLESS[1]

- AWS Lambda function for each IPO request
  - Run as many copies of the IPO function as needed in parallel

[1] Financial Engines Cuts Costs 90% Using AWS Lambda and Serverless Computing, https://aws.amazon.com/solutions/case-studies/financial-engines/

A REAL USE-CASE: HOW FINANCIAL ENGINES CUT COSTS 90% USING SERVERLESS [1]

λ | Solver Library | λ | Solver Library | λ | Solver Library

- AWS Lambda function for each IPO request
  - Run as many copies of the IPO function as needed in parallel

[1] Financial Engines Cuts Costs 90% Using AWS Lambda and Serverless Computing, https://aws.amazon.com/solutions/case-studies/financial-engines/

A REAL USE-CASE: HOW FINANCIAL ENGINES CUT COSTS 90% USING SERVERLESS [1]

λ	Solver Library	λ	Solver Library	λ	Solver Library

- AWS Lambda function for each IPO request
  - Run as many copies of the IPO function as needed in parallel

[1] Financial Engines Cuts Costs 90% Using AWS Lambda and Serverless Computing, https://aws.amazon.com/solutions/case-studies/financial-engines/

A REAL USE-CASE: HOW FINANCIAL ENGINES CUT COSTS 90% USING SERVERLESS[1]

λ | Solver Library | λ | Solver Library | λ | Solver Library

- AWS Lambda function for each IPO request
  - Run as many copies of the IPO function as needed in parallel

- Serverless benefits
  - 200-300 M IPO requests/month, 60,000 per minute at peak
  - Up to 94% cost savings annually, *not* including operational savings
  - Increased reliability: just instantiate new lambda requests on crash

[1] Financial Engines Cuts Costs 90% Using AWS Lambda and Serverless Computing, https://aws.amazon.com/solutions/case-studies/financial-engines/

SERVERLESS TODAY

EVOLUTION[1]

OF CLOUD PLATFORMS

[1] Slide adapted from talk by Eric Jonas and Johann Schleier-Smith, “A Berkeley View on Cloud Computing”

EVOLUTION[1]

OF CLOUD PLATFORMS

On-prem virtualization

[1] Slide adapted from talk by Eric Jonas and Johann Schleier-Smith, “A Berkeley View on Cloud Computing”

EVOLUTION[1]

OF CLOUD PLATFORMS

On-prem virtualization

↓

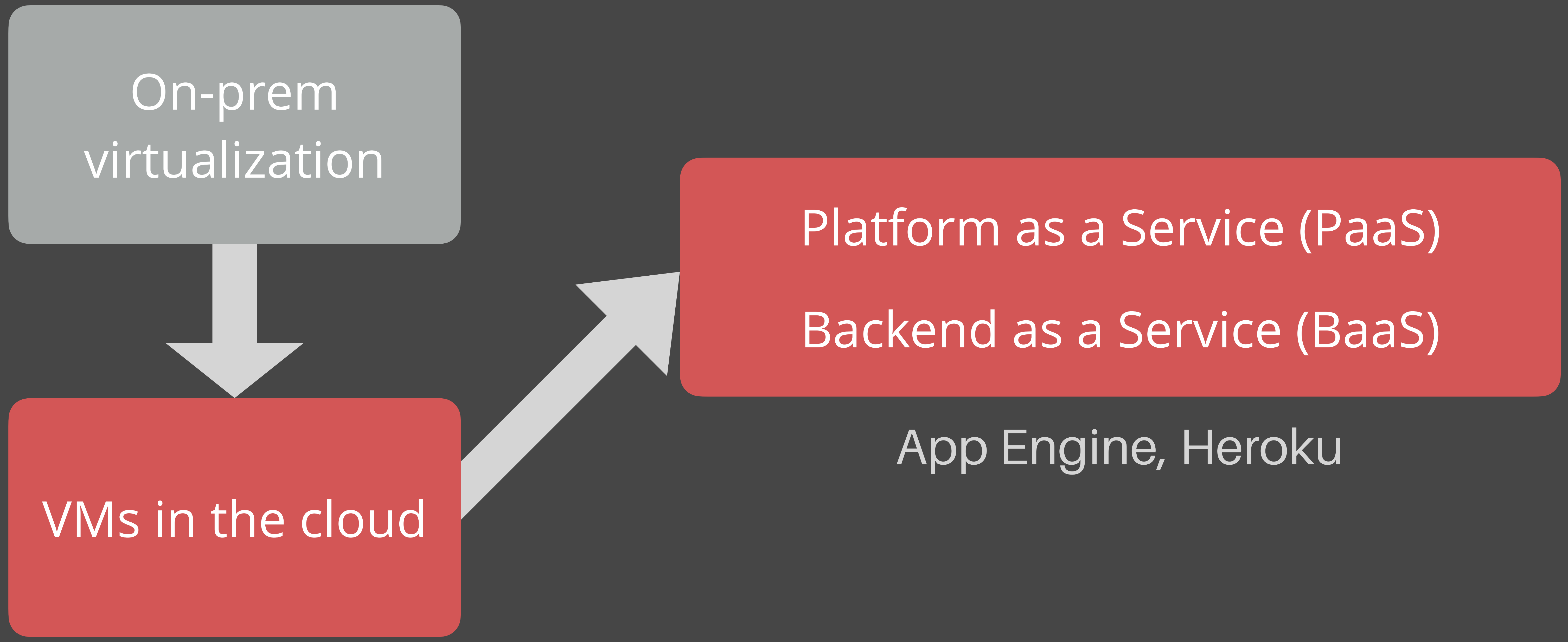
VMs in the cloud

- Easy switch from legacy infrastructure
- Added *cloud services* (e.g., storage, compute)

[1] Slide adapted from talk by Eric Jonas and Johann Schleier-Smith, "A Berkeley View on Cloud Computing"

EVOLUTION[1]

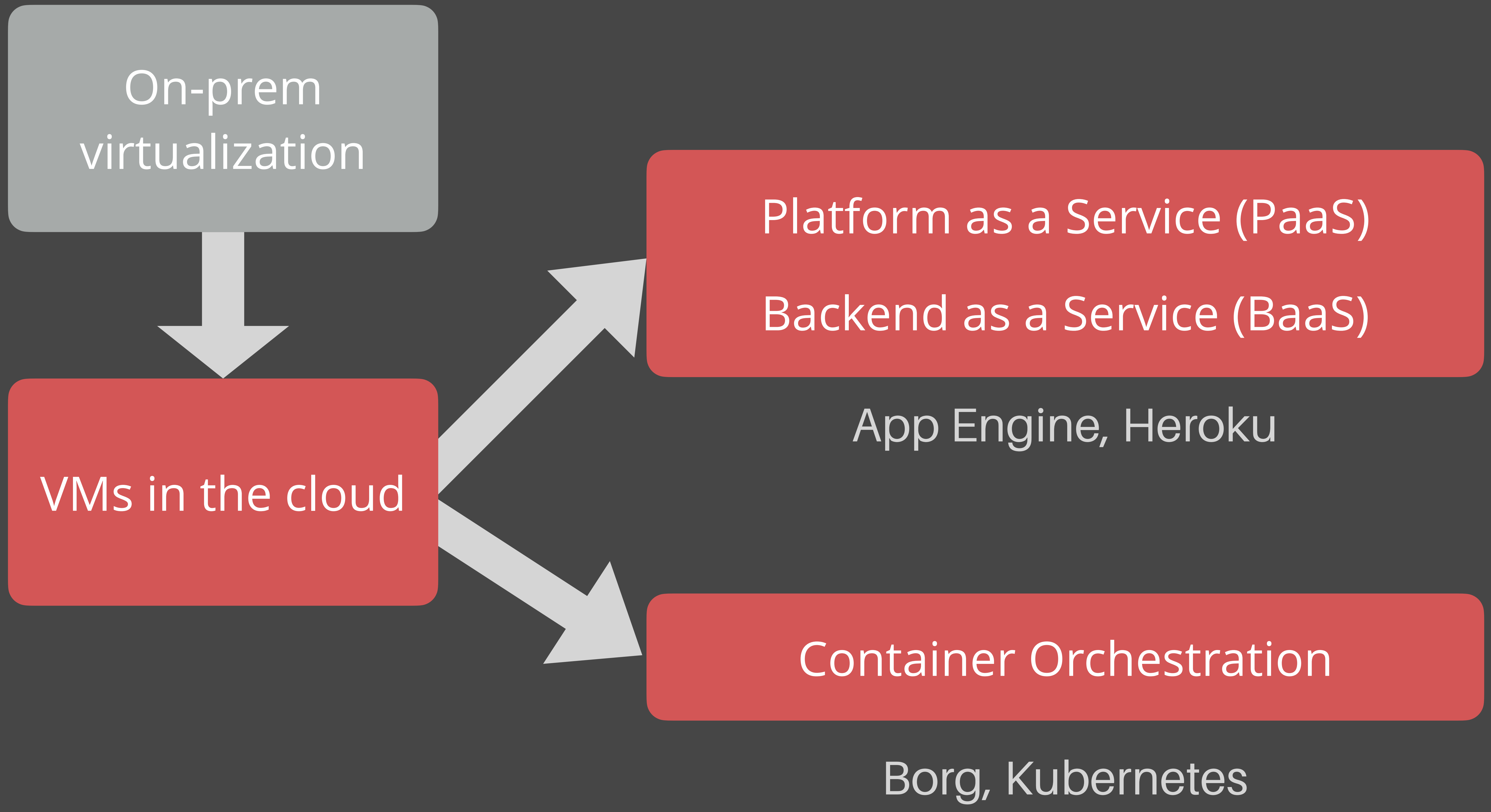
OF CLOUD PLATFORMS



[1] Slide adapted from talk by Eric Jonas and Johann Schleier-Smith, "A Berkeley View on Cloud Computing"

EVOLUTION[1]

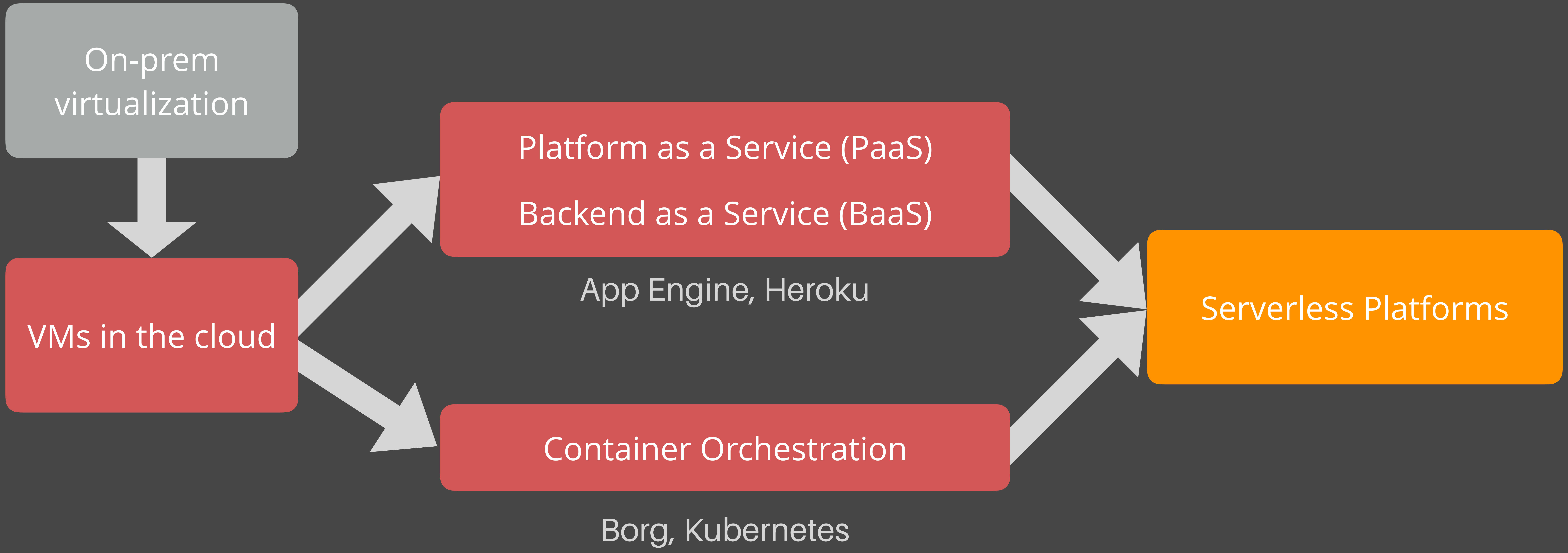
OF CLOUD PLATFORMS



[1] Slide adapted from talk by Eric Jonas and Johann Schleier-Smith, "A Berkeley View on Cloud Computing"

EVOLUTION[1]

OF CLOUD PLATFORMS



[1] Slide adapted from talk by Eric Jonas and Johann Schleier-Smith, "A Berkeley View on Cloud Computing"

SERVERLESS TODAY: FUNCTION-AS-A-SERVICE (FAAS)

[1] Peeking Behind the Curtains of Serverless Platforms, Wang et. al.
[2] Evaluation of Production Serverless Computing Environments, Lee et. al.

SERVERLESS TODAY: FUNCTION-AS-A-SERVICE (FAAS)

- Many FaaS platforms

AWS Lambda | IBM Cloud Functions | Google Cloud Functions | Cloudflare Workers | Azure Functions | Alibaba Function Compute

[1] Peeking Behind the Curtains of Serverless Platforms, Wang et. al.
[2] Evaluation of Production Serverless Computing Environments, Lee et. al.

SERVERLESS TODAY: FUNCTION-AS-A-SERVICE (FAAS)

- Many FaaS platforms

AWS Lambda | IBM Cloud Functions | Google Cloud Functions | Cloudflare Workers | Azure Functions | Alibaba Function Compute

- Many FaaS orchestration frameworks:

AWS Step Functions | Microsoft Azure — Azure Durable Functions | IBM Composer

[1] Peeking Behind the Curtains of Serverless Platforms, Wang et. al.
[2] Evaluation of Production Serverless Computing Environments, Lee et. al.

SERVERLESS TODAY: FUNCTION-AS-A-SERVICE (FAAS)

- Many FaaS platforms

AWS Lambda | IBM Cloud Functions | Google Cloud Functions | Cloudflare Workers | Azure Functions | Alibaba Function Compute

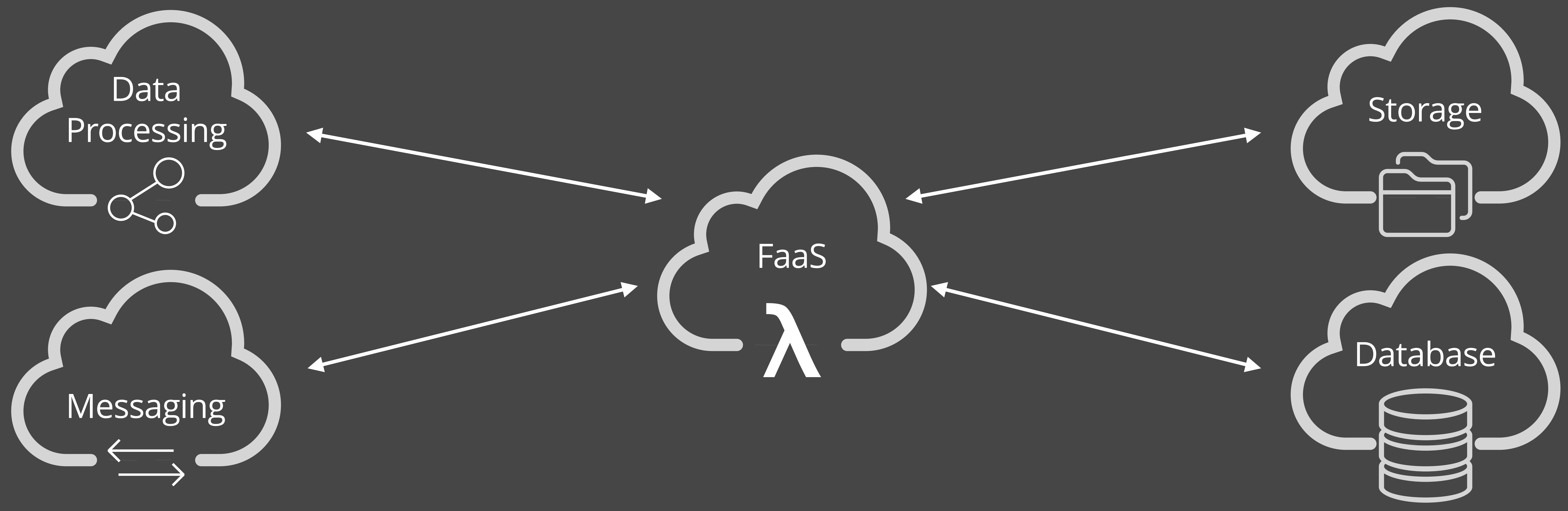
- Many FaaS orchestration frameworks:

AWS Step Functions | Microsoft Azure — Azure Durable Functions | IBM Composer

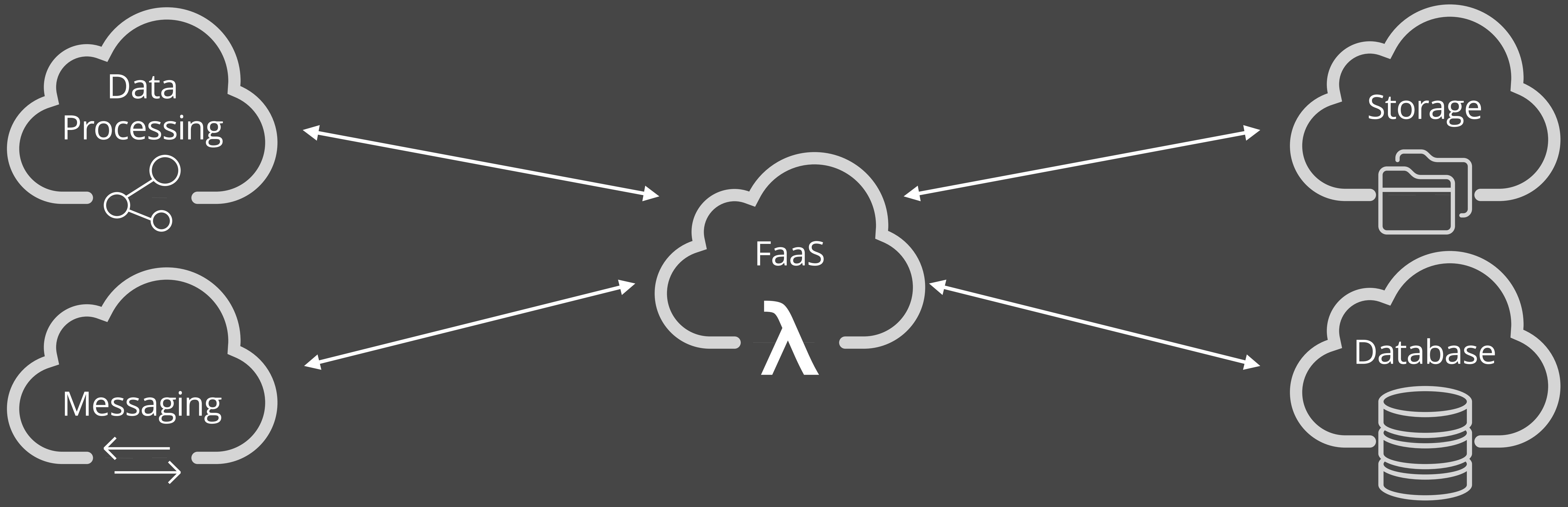
- Different pricing models, resource allocations, security, isolation, programming language support, OS support, etc.[1,2]

[1] Peeking Behind the Curtains of Serverless Platforms, Wang et. al.
[2] Evaluation of Production Serverless Computing Environments, Lee et. al.

SERVERLESS TODAY: BACKEND AS A SERVICE (BAAS)

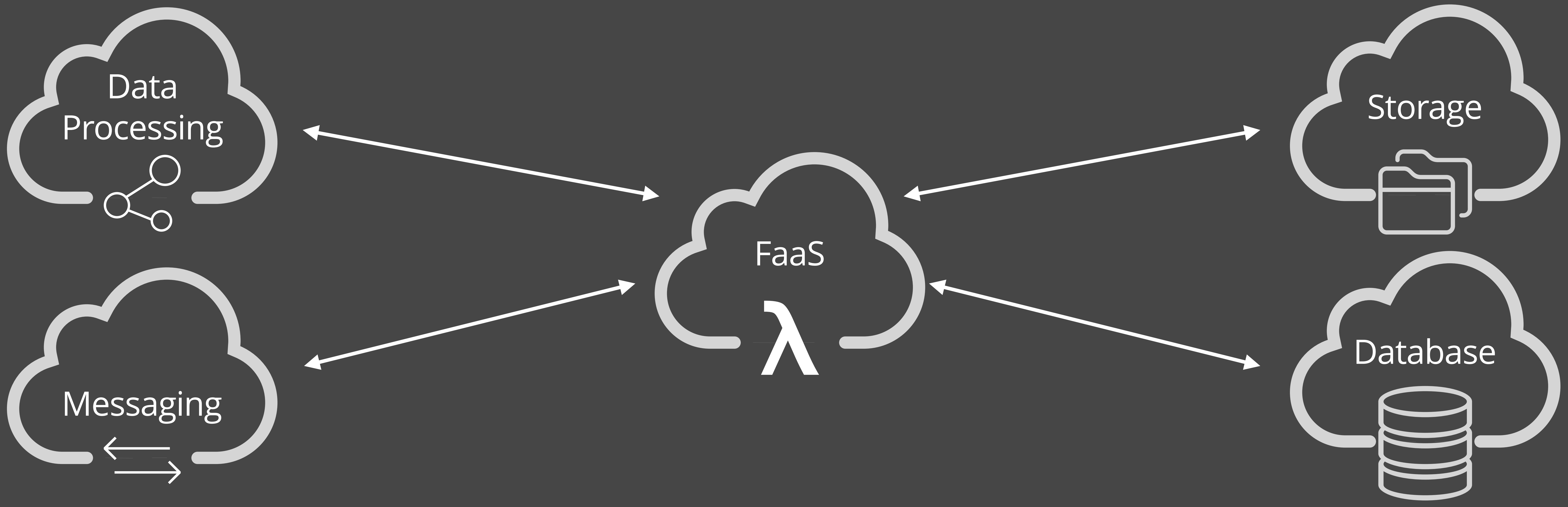


SERVERLESS TODAY: BACKEND AS A SERVICE (BAAS)



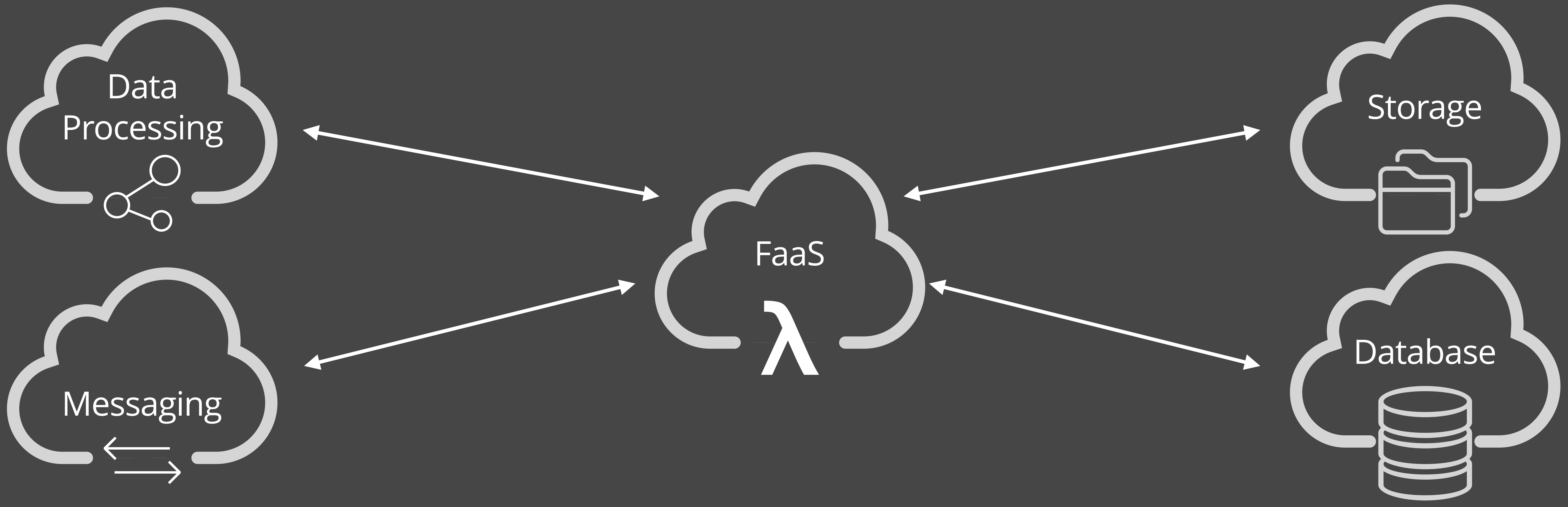
Serverless

SERVERLESS TODAY: BACKEND AS A SERVICE (BAAS)



Serverless = FaaS + **BaaS**

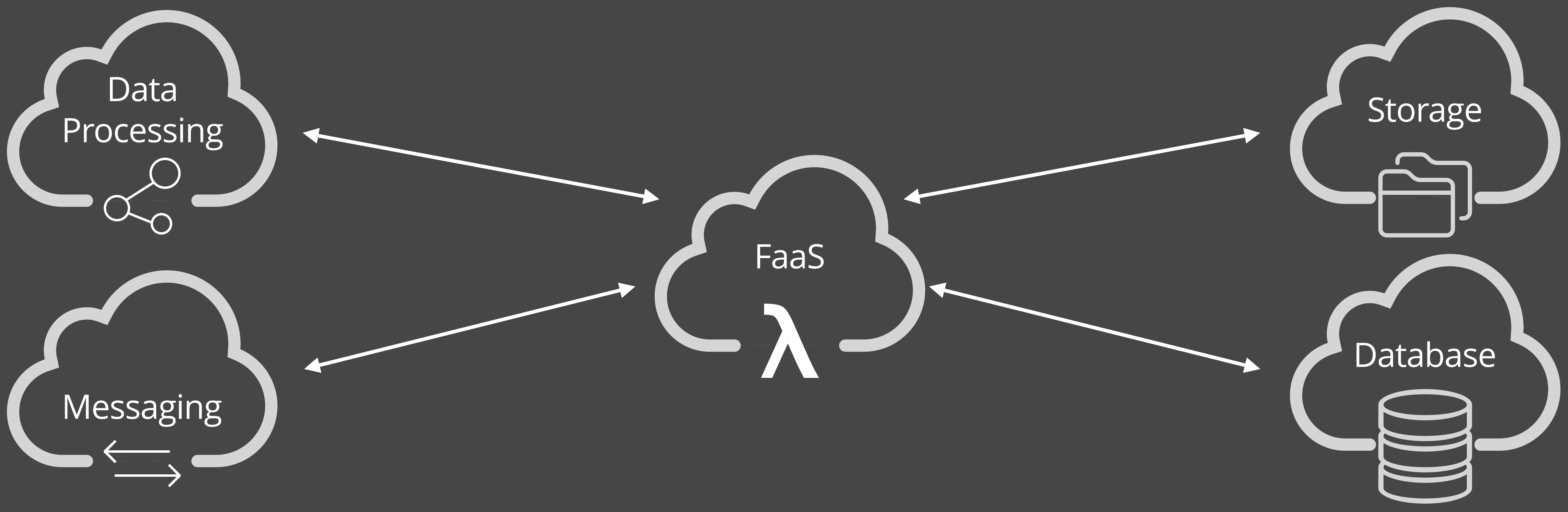
SERVERLESS TODAY: BACKEND AS A SERVICE (BAAS)



Serverless = FaaS + **BaaS**

- Object Storage (e.g., S3)
- Key-Value Stores (e.g., DynamoDB)
- Database (e.g., Cloud Firestore)
- Data Processing (e.g., Cloud Dataflow)

SERVERLESS TODAY: BACKEND AS A SERVICE (BAAS)



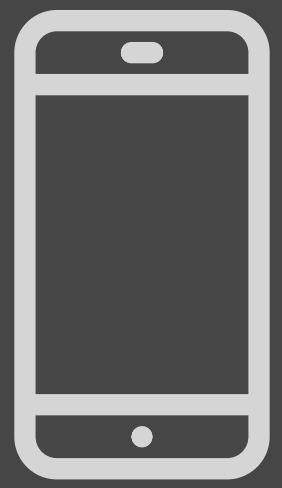
Serverless = FaaS + **BaaS**

- Complexity Hiding
- Consumption based billing
- Automatic scaling

- Object Storage (e.g., S3)
- Key-Value Stores (e.g., DynamoDB)
- Database (e.g., Cloud Firestore)
- Data Processing (e.g., Cloud Dataflow)

SERVERLESS TODAY: EVENT-DRIVEN & STATELESS

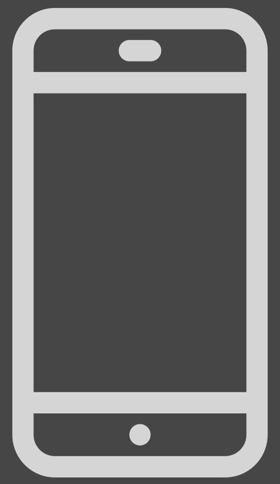
Image resizing example:



[1] Slide adapted from talk by Eric Jonas and Johann Schleier-Smith, “A Berkeley View on Cloud Computing”

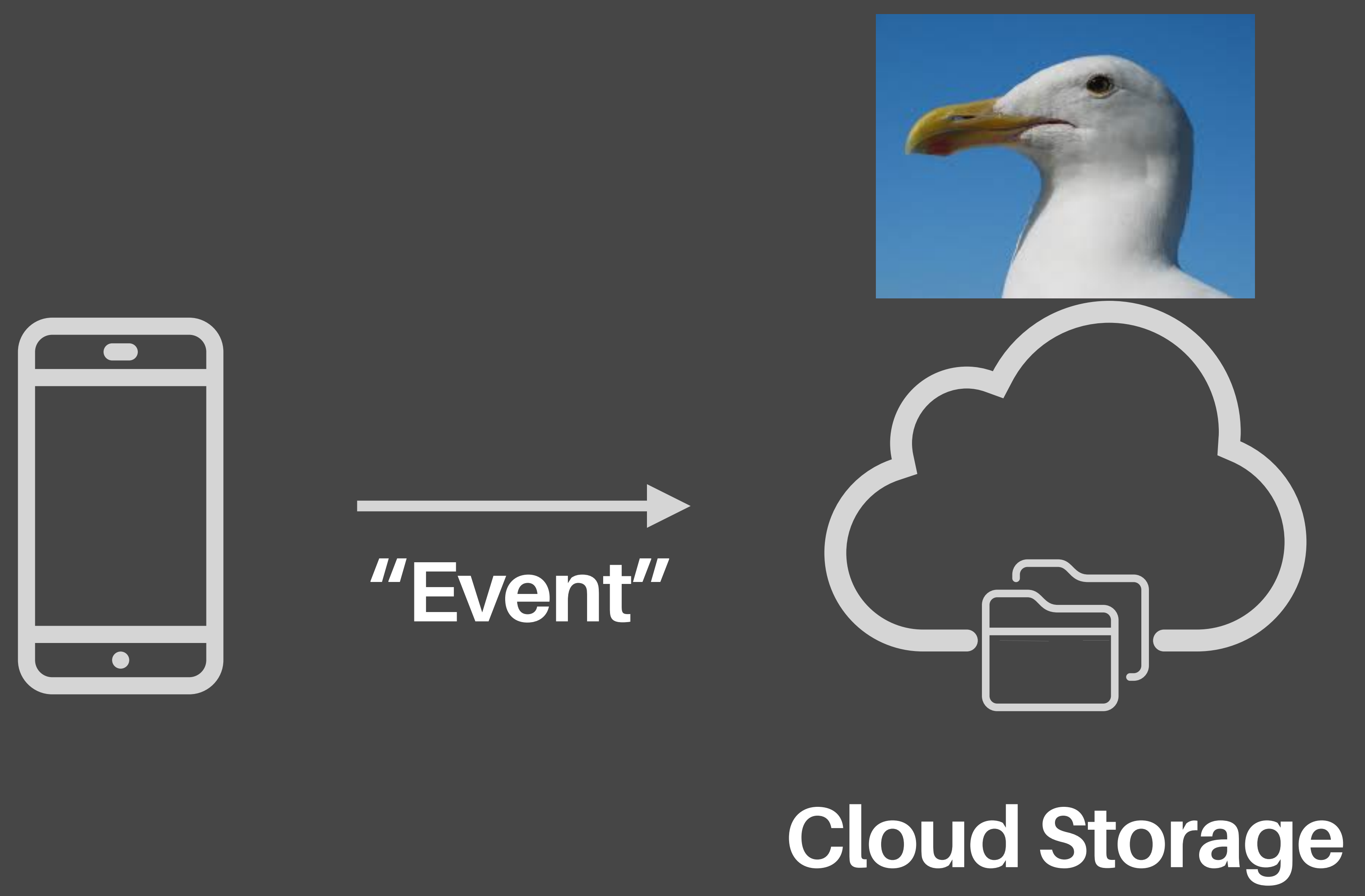
SERVERLESS TODAY: EVENT-DRIVEN & STATELESS

Image resizing example:



SERVERLESS TODAY: EVENT-DRIVEN & STATELESS

Image resizing example:



[1] Slide adapted from talk by Eric Jonas and Johann Schleier-Smith, “A Berkeley View on Cloud Computing”

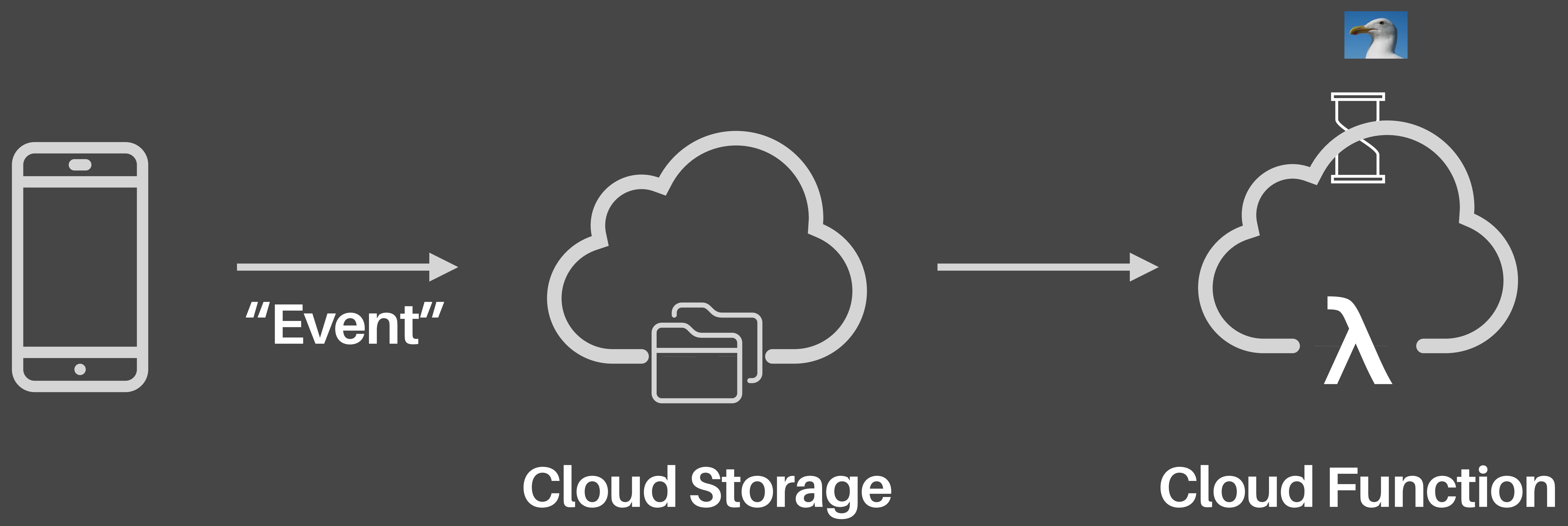
SERVERLESS TODAY: EVENT-DRIVEN & STATELESS

Image resizing example:

[1] Slide adapted from talk by Eric Jonas and Johann Schleier-Smith, "A Berkeley View on Cloud Computing"

SERVERLESS TODAY: EVENT-DRIVEN & STATELESS

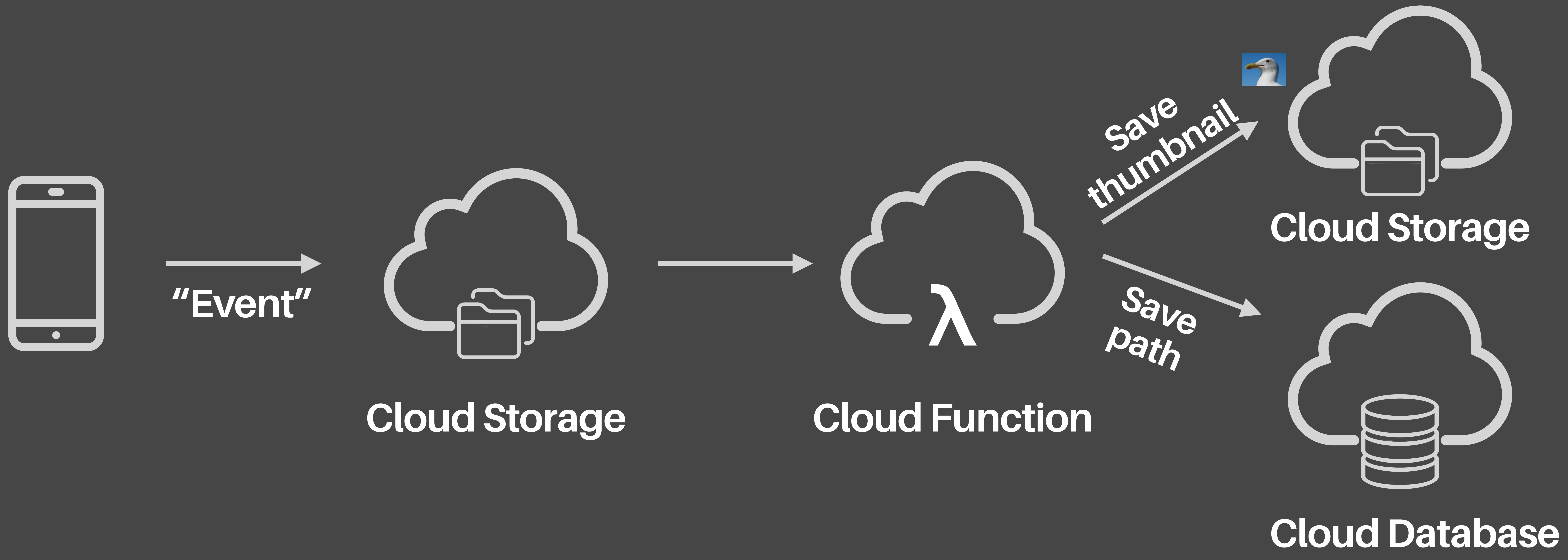
Image resizing example:



[1] Slide adapted from talk by Eric Jonas and Johann Schleier-Smith, "A Berkeley View on Cloud Computing"

SERVERLESS TODAY: EVENT-DRIVEN & STATELESS

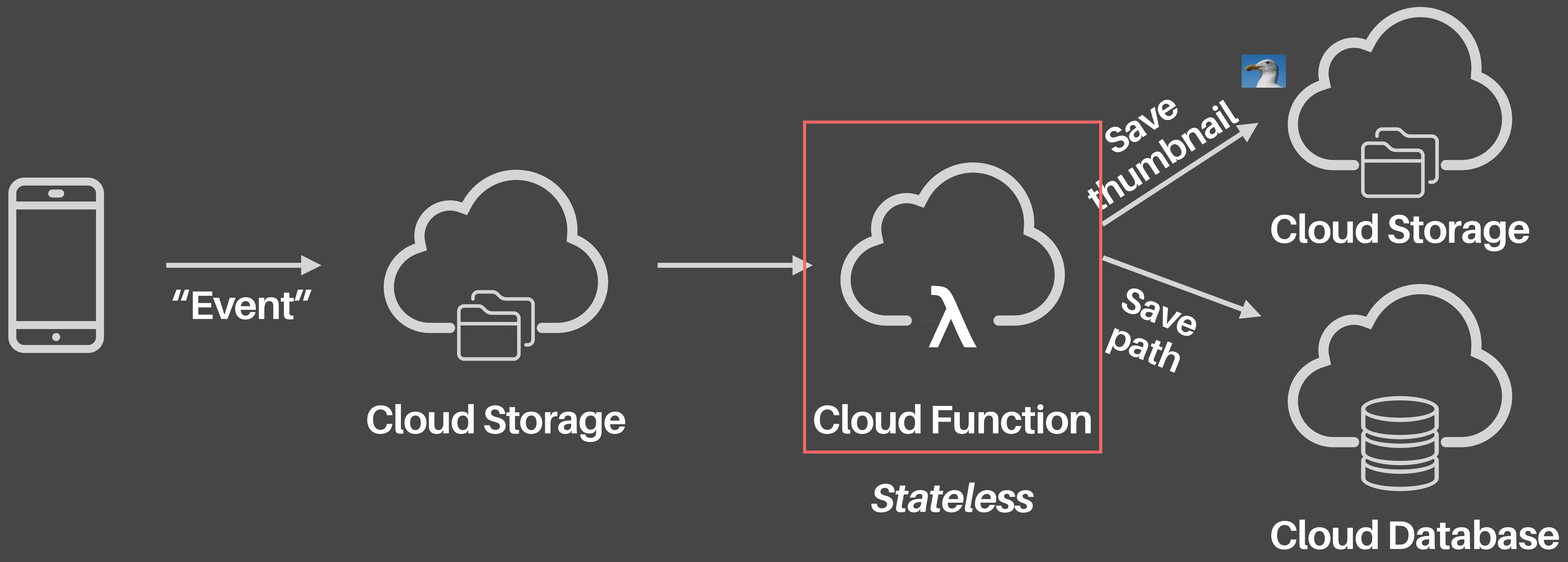
Image resizing example:



[1] Slide adapted from talk by Eric Jonas and Johann Schleier-Smith, "A Berkeley View on Cloud Computing"

SERVERLESS TODAY: EVENT-DRIVEN & STATELESS

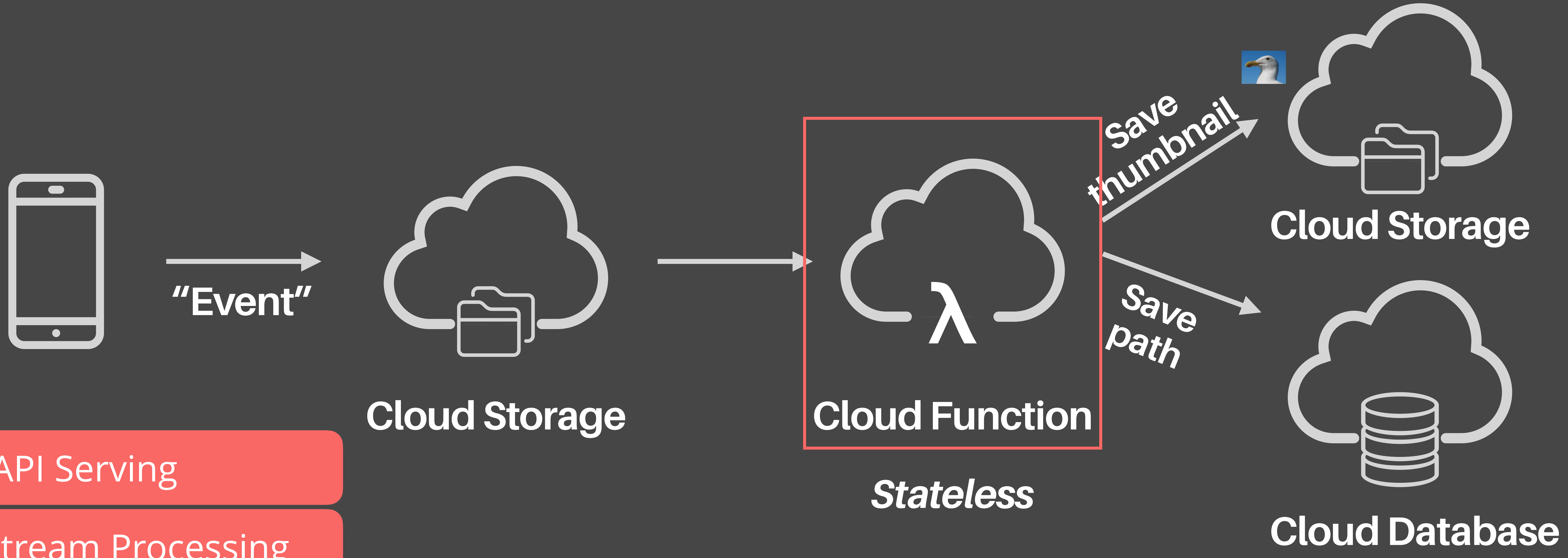
Image resizing example:



[1] Slide adapted from talk by Eric Jonas and Johann Schleier-Smith, "A Berkeley View on Cloud Computing"

SERVERLESS TODAY: EVENT-DRIVEN & STATELESS

Image resizing example:



API Serving

Event Stream Processing

ETL

[1] Slide adapted from talk by Eric Jonas and Johann Schleier-Smith, "A Berkeley View on Cloud Computing"

WHAT ABOUT *STATEFUL* APPLICATIONS?

WHAT ABOUT *STATEFUL* APPLICATIONS?

Generate, exchange and consume *intermediate data*

WHAT ABOUT *STATEFUL* APPLICATIONS?

Generate, exchange and consume *intermediate data*

Streaming Analytics

SQL Analytics

Video Analytics

⋮

WHAT ABOUT *STATEFUL* APPLICATIONS?

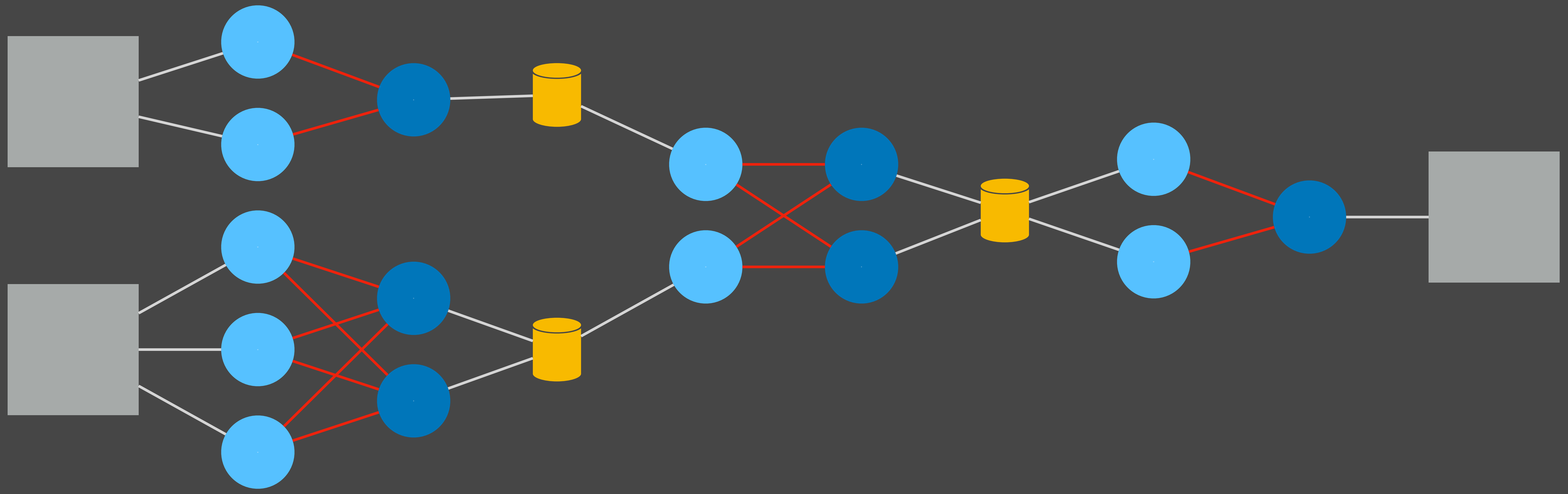
Generate, exchange and consume *intermediate data*

Streaming Analytics

SQL Analytics

Video Analytics

⋮



WHAT ABOUT *STATEFUL* APPLICATIONS?

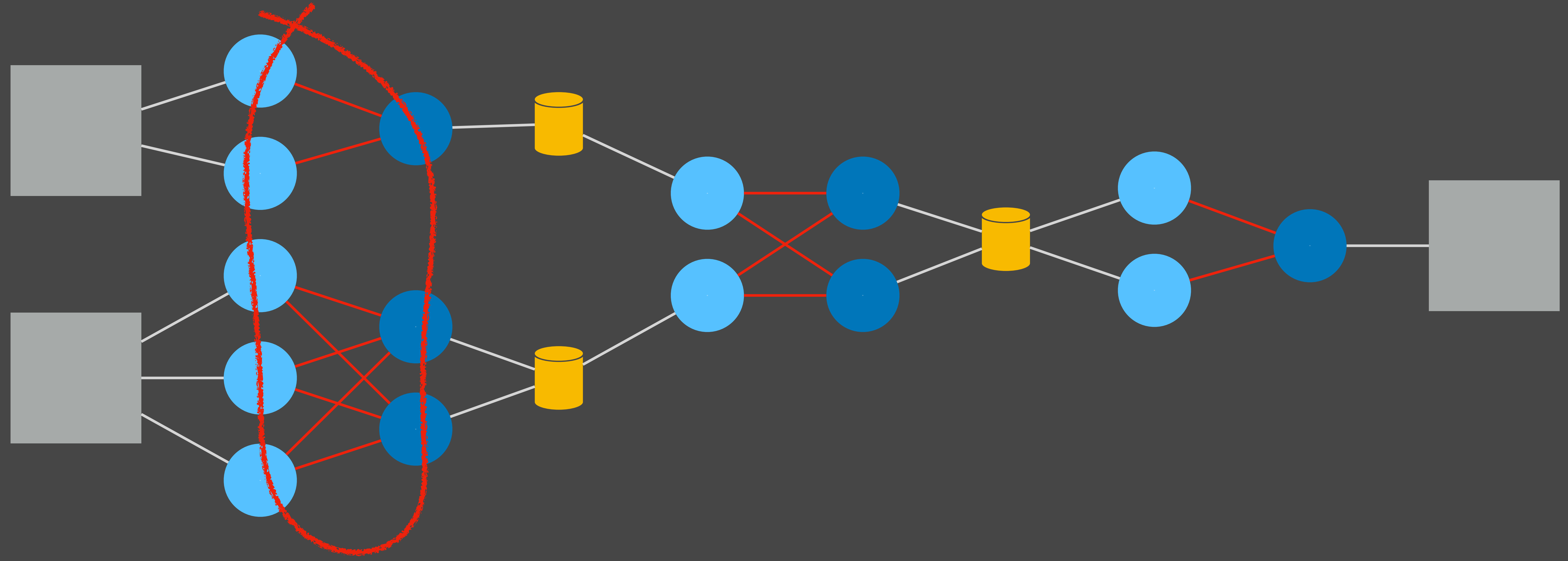
Generate, exchange and consume *intermediate data*

Streaming Analytics

SQL Analytics

Video Analytics

⋮



WHAT ABOUT *STATEFUL* APPLICATIONS?

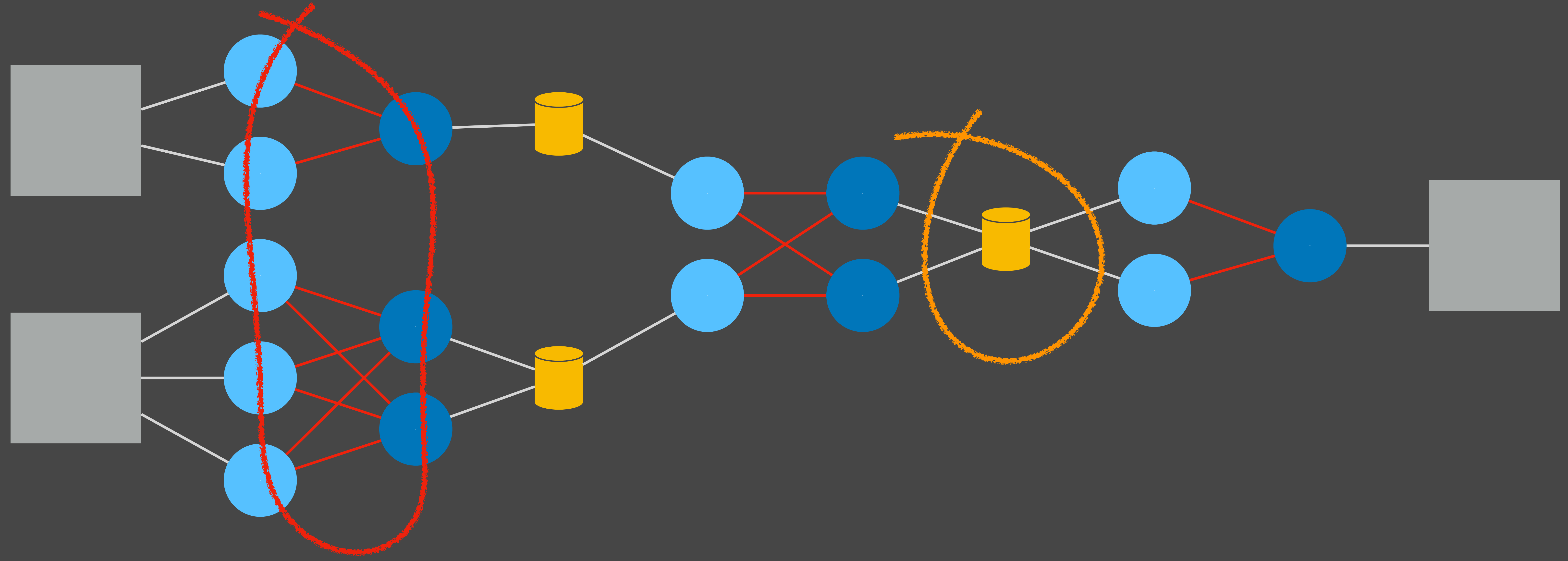
Generate, exchange and consume *intermediate data*

Streaming Analytics

SQL Analytics

Video Analytics

⋮



WHAT ABOUT *STATEFUL* APPLICATIONS?

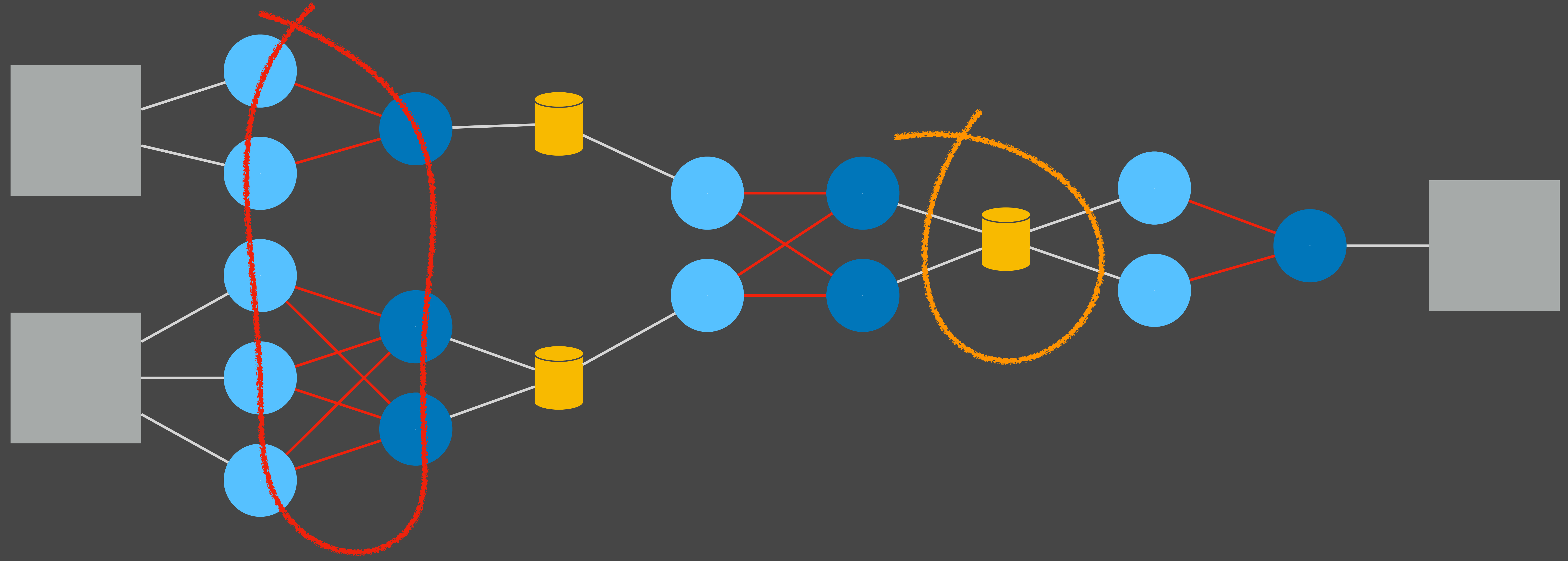
Generate, exchange and consume *intermediate data*

Streaming Analytics

SQL Analytics

Video Analytics

⋮

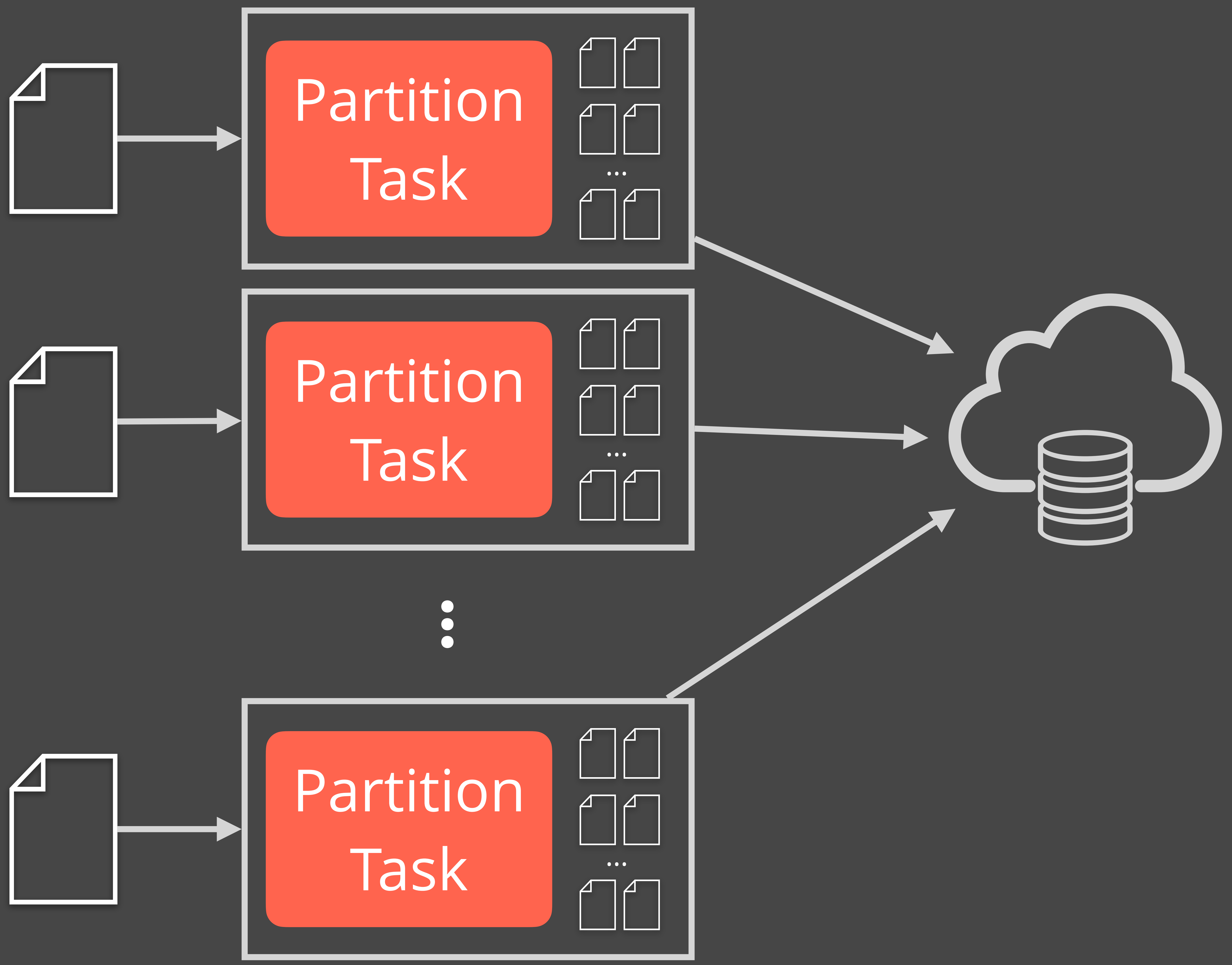


How can we efficiently support stateful applications on serverless?

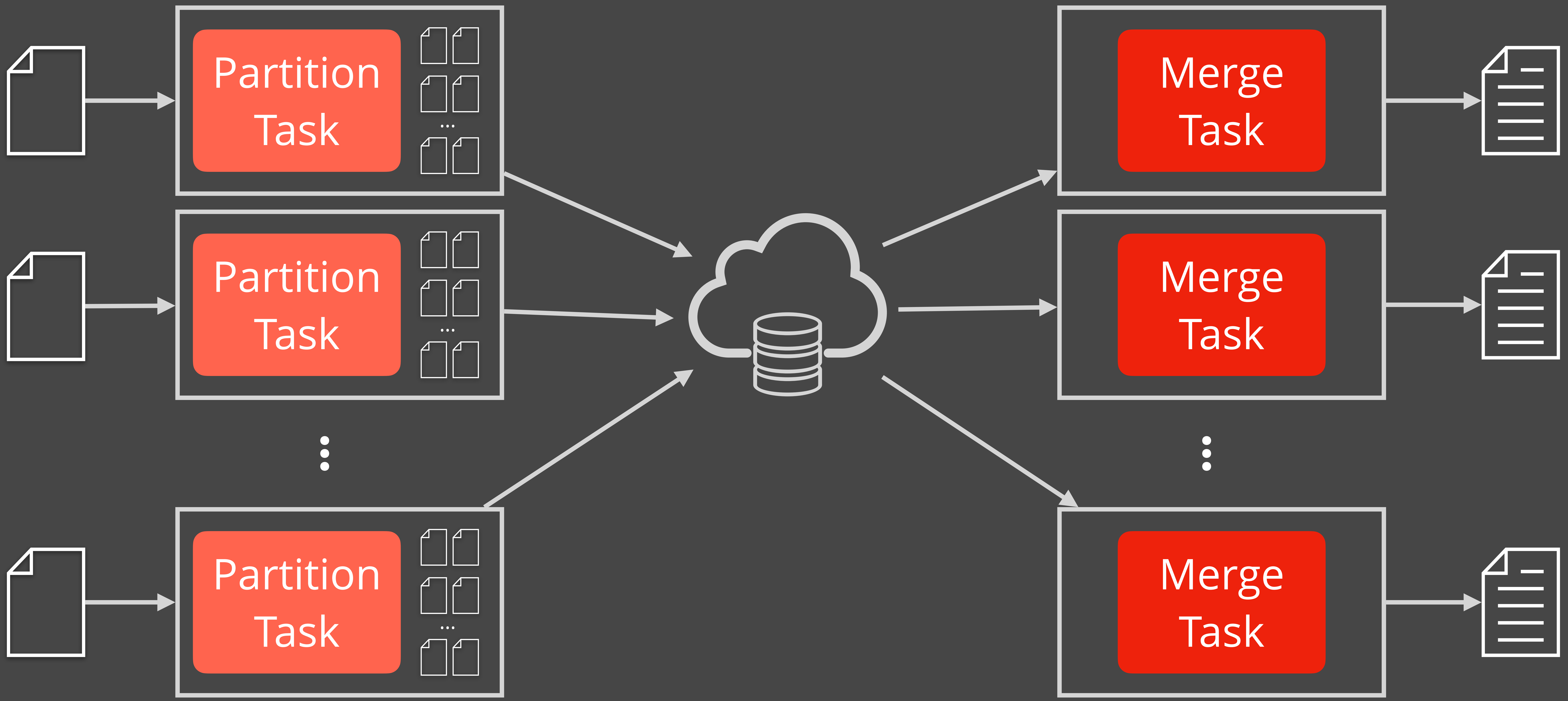
EMERGING & FUTURE SUPPORT FOR STATEFUL APPLICATIONS

A SIMPLE ANALYTICS EXAMPLE: SORT

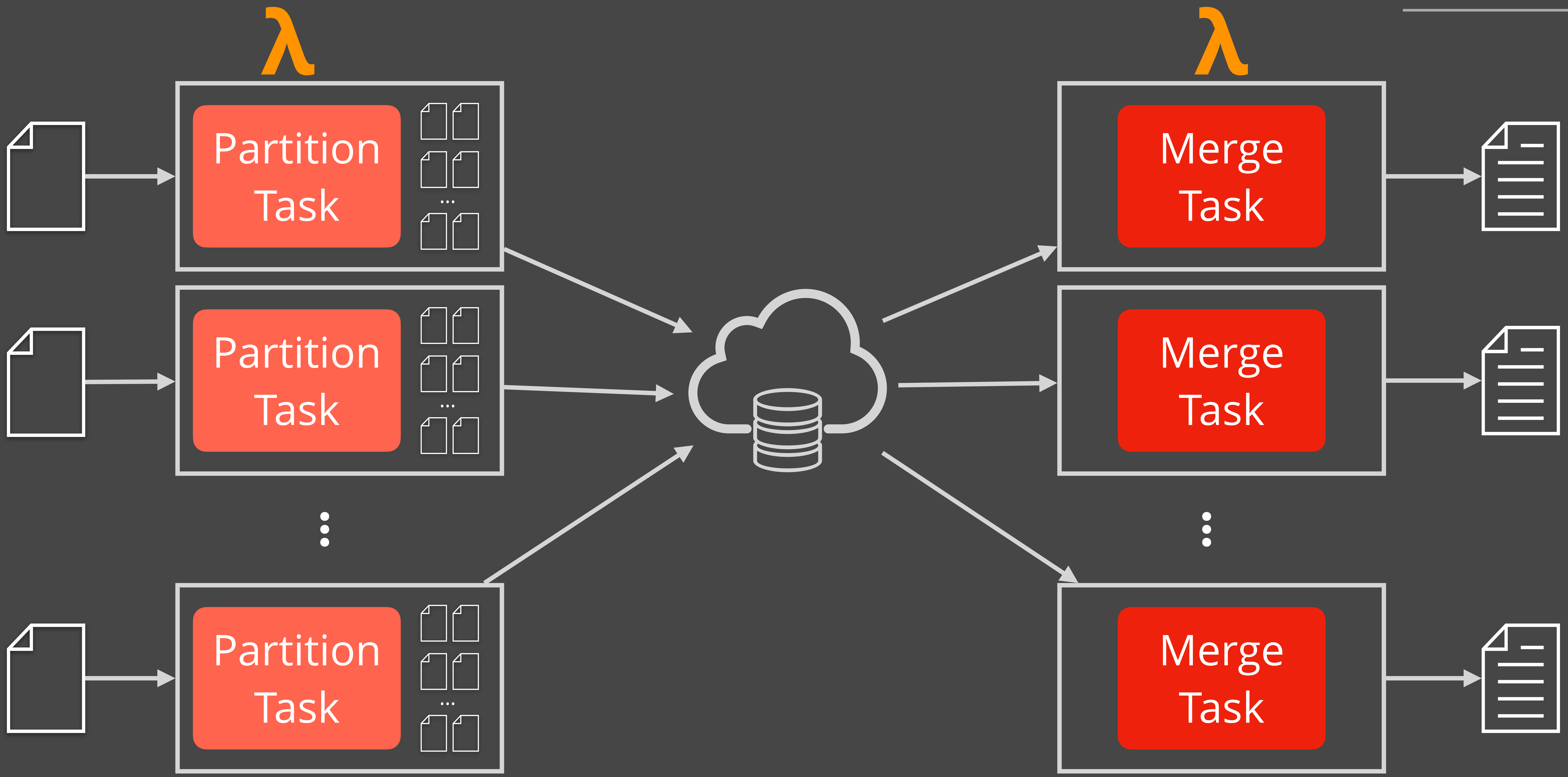
A SIMPLE ANALYTICS EXAMPLE: SORT



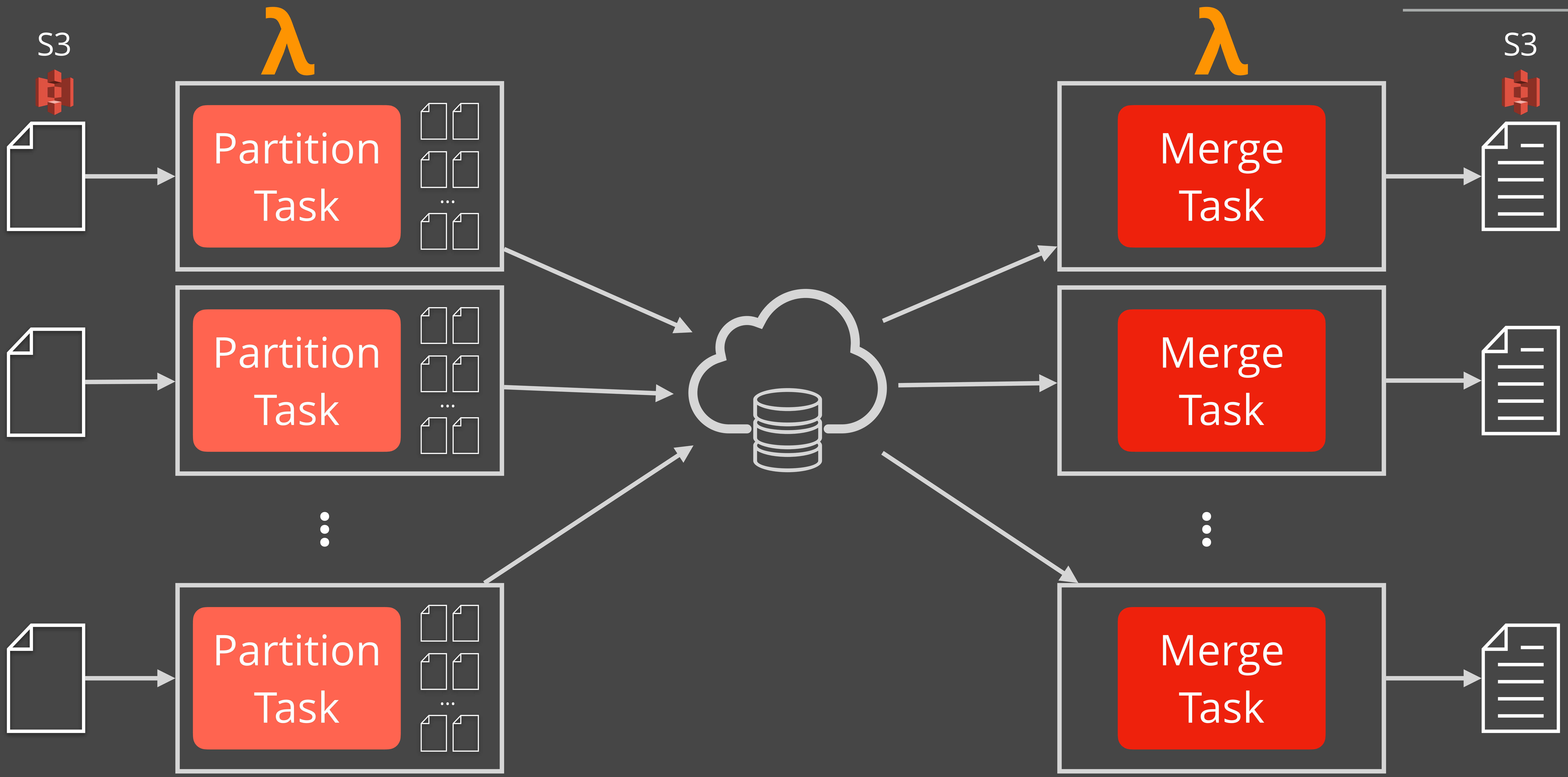
A SIMPLE ANALYTICS EXAMPLE: SORT



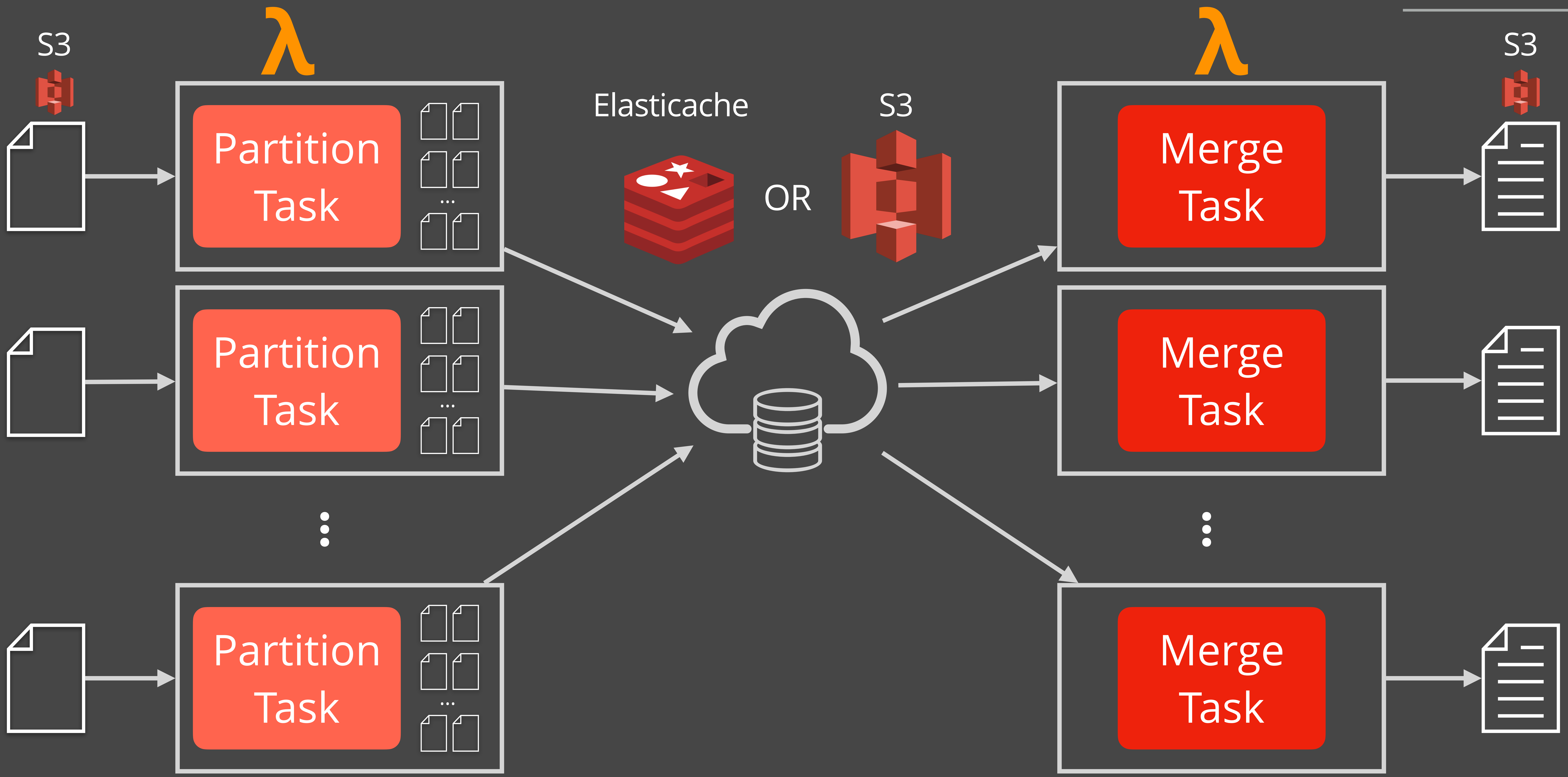
A SIMPLE ANALYTICS EXAMPLE: SORT



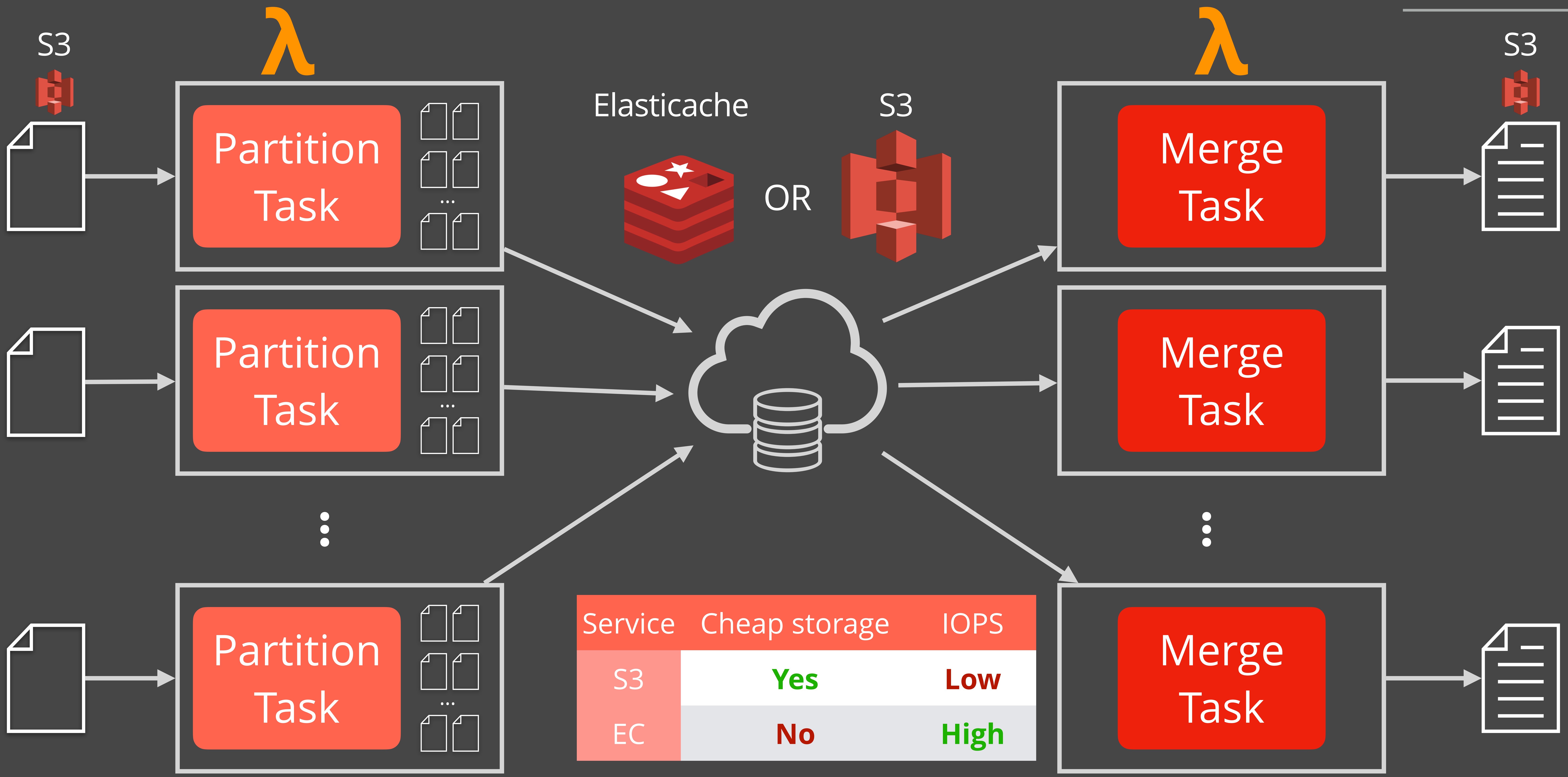
A SIMPLE ANALYTICS EXAMPLE: SORT



A SIMPLE ANALYTICS EXAMPLE: SORT



A SIMPLE ANALYTICS EXAMPLE: SORT

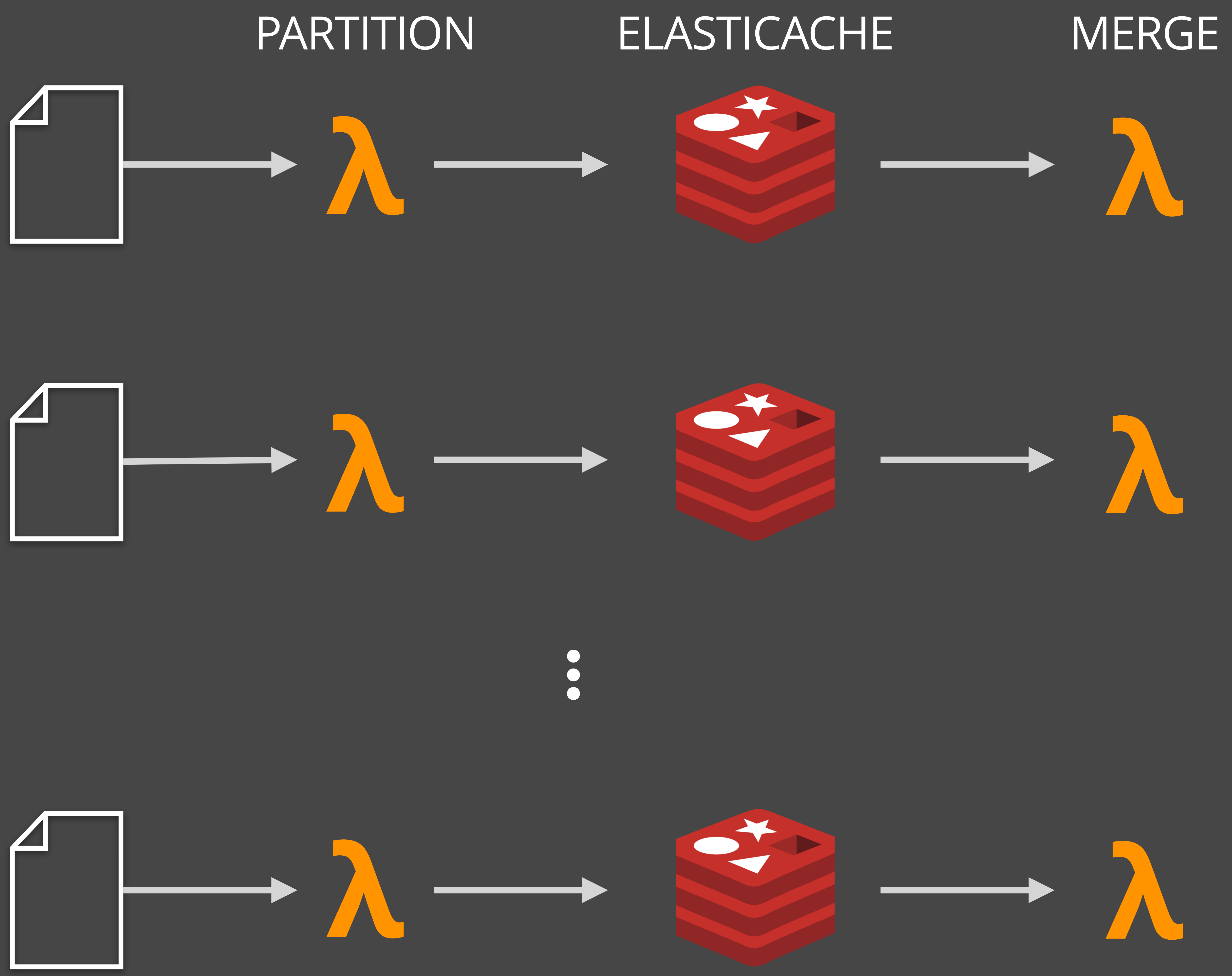


Service	Cheap storage	IOPS
S3	Yes	Low
EC	No	High

SERVERLESS ANALYTICS: LOCUS

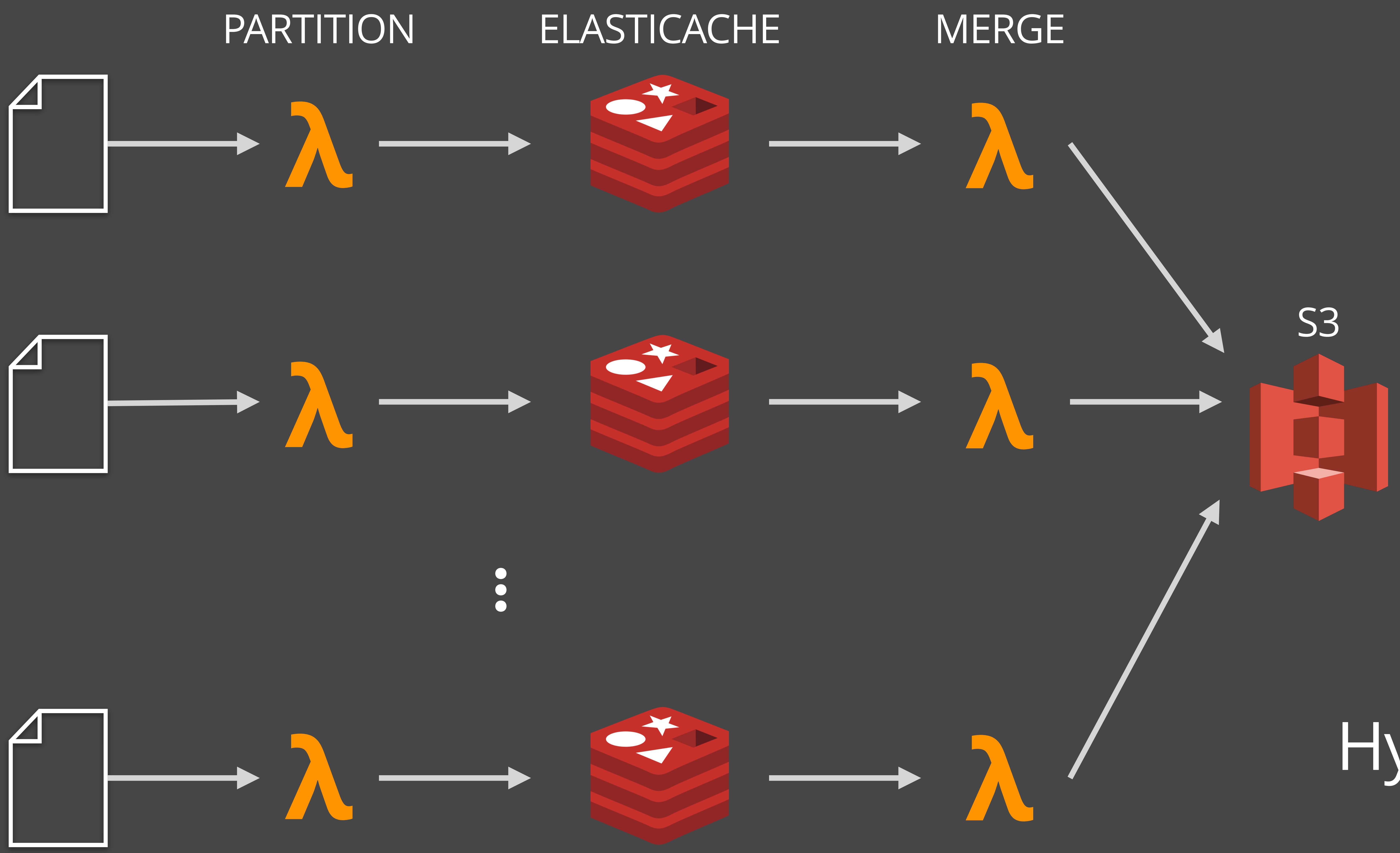
Hybrid Sort

SERVERLESS ANALYTICS: LOCUS



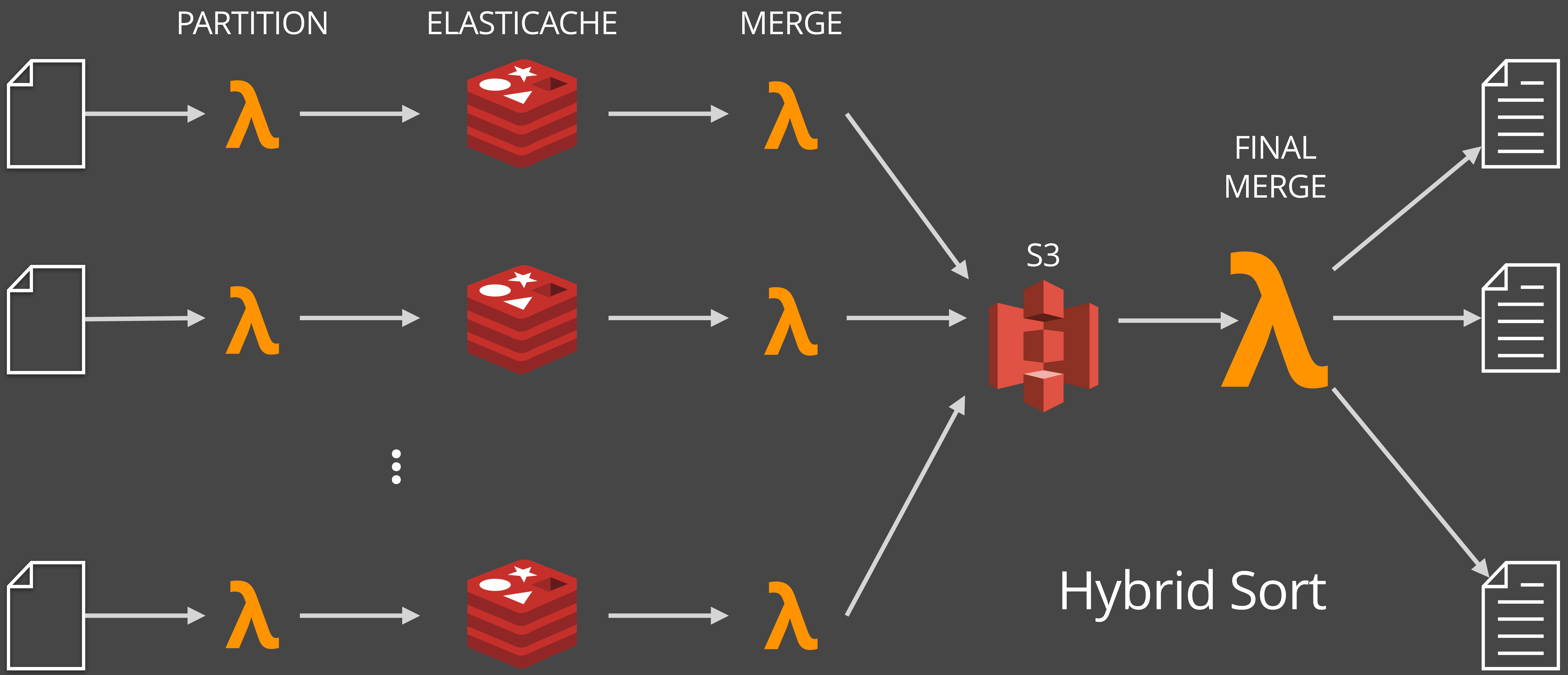
Hybrid Sort

SERVERLESS ANALYTICS: LOCUS



Hybrid Sort

SERVERLESS ANALYTICS: LOCUS



Hybrid Sort

VIDEO ENCODING/DECODING

VIDEO ENCODING/DECODING

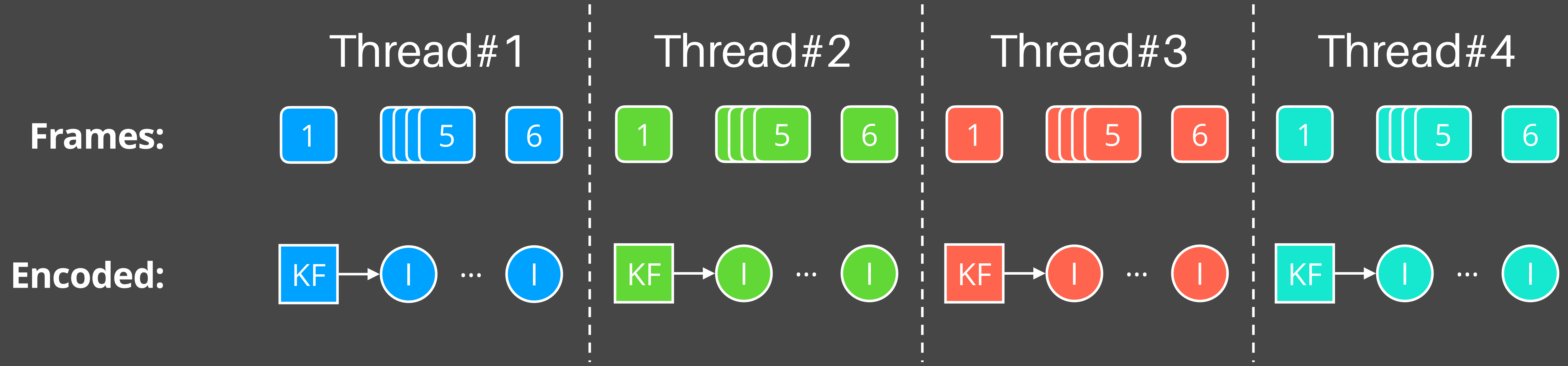
How is it done today?

- Video = Series of Chunks
  - Chunk = KeyFrame (large) + InterFrames (small deltas from KeyFrame)

VIDEO ENCODING/DECODING

How is it done today?

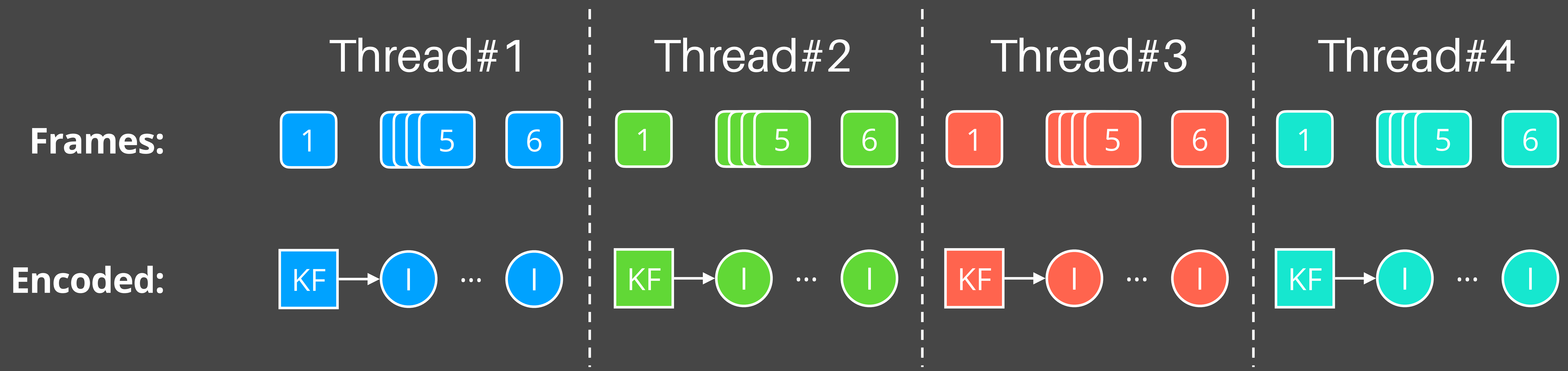
- Video = Series of Chunks
  - Chunk = KeyFrame (large) + InterFrames (small deltas from KeyFrame)



VIDEO ENCODING/DECODING

How is it done today?

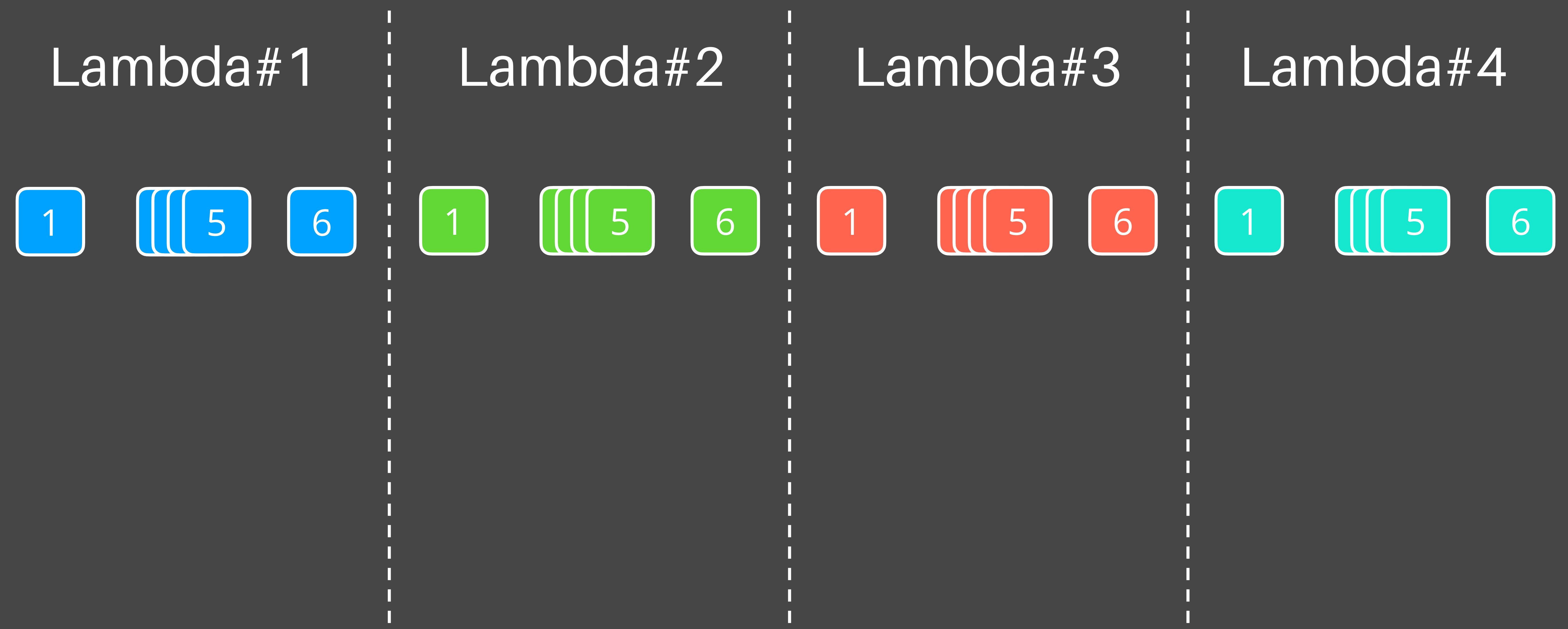
- Video = Series of Chunks
  - Chunk = KeyFrame (large) + InterFrames (small deltas from KeyFrame)



- High parallelism = worse compression (more KeyFrames)

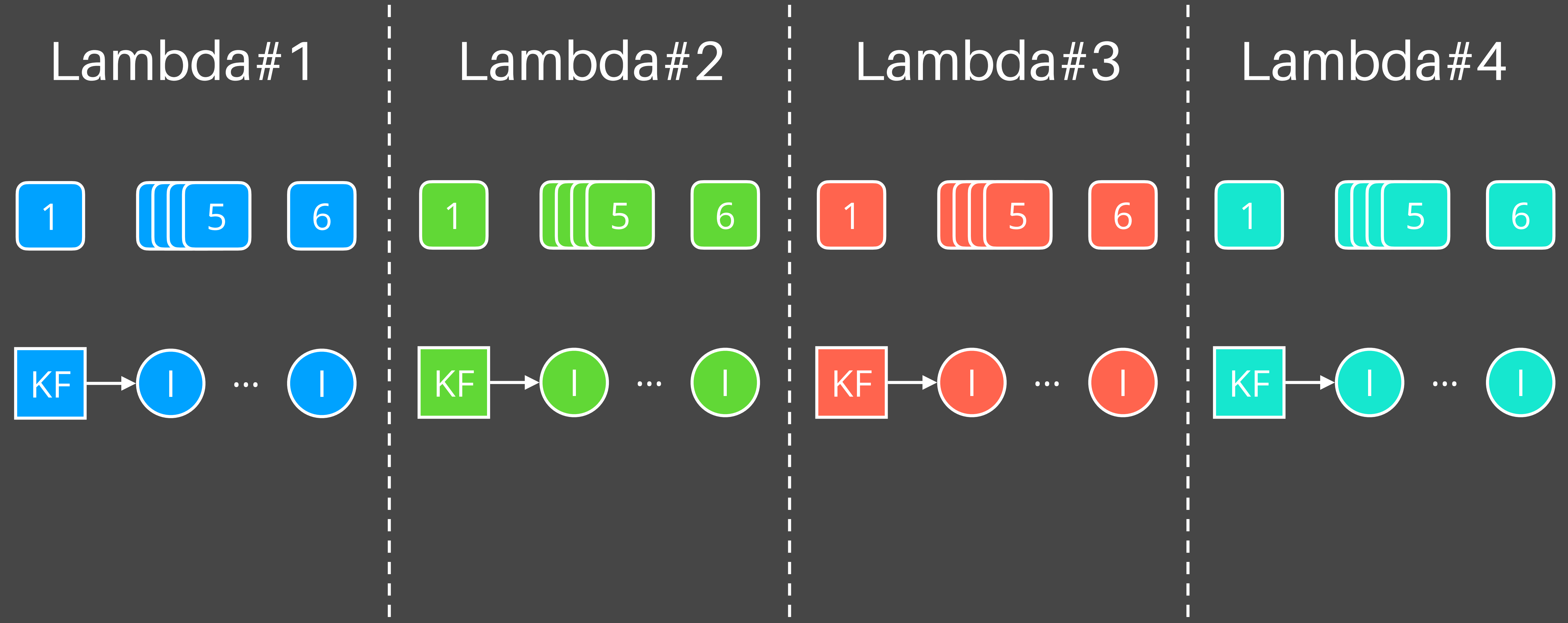
VIDEO ANALYTICS: EXCAMERA

VIDEO ENCODING/DECODING ON AWS LAMBDA



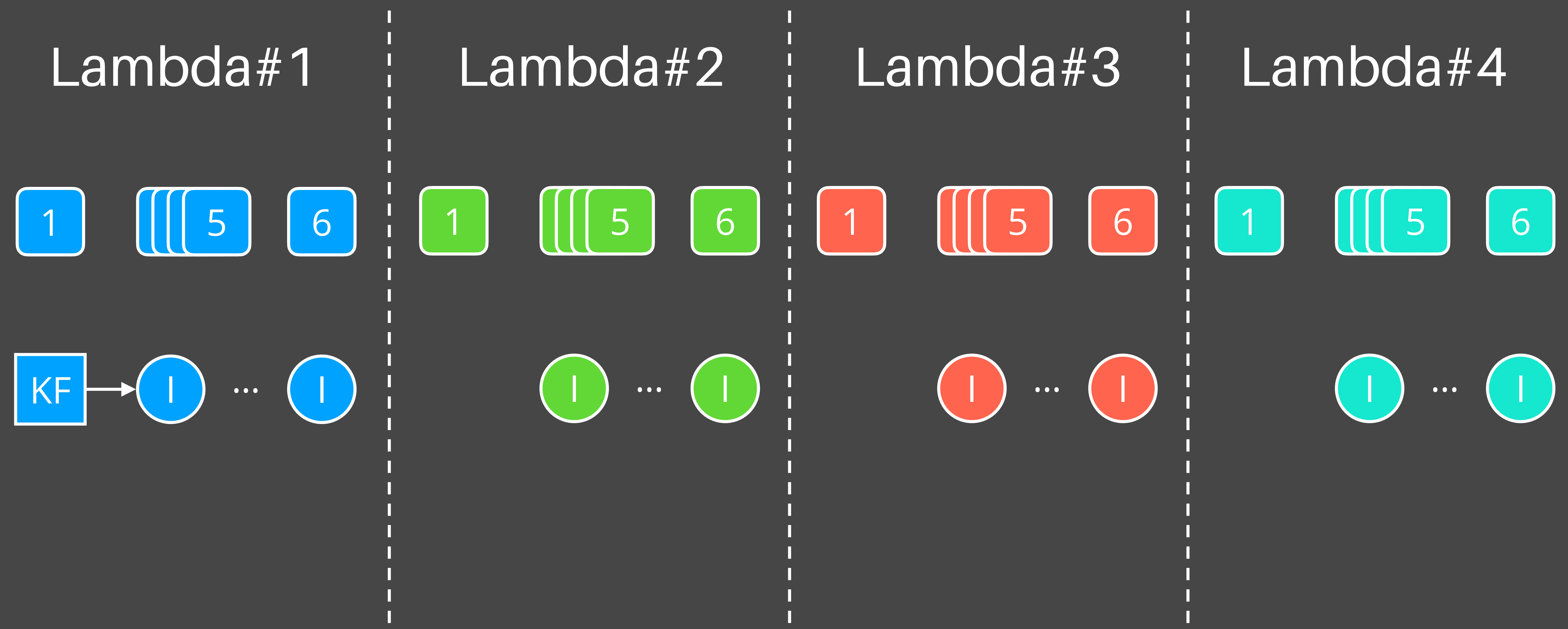
VIDEO ANALYTICS: EXCAMERA

VIDEO ENCODING/DECODING ON AWS LAMBDA



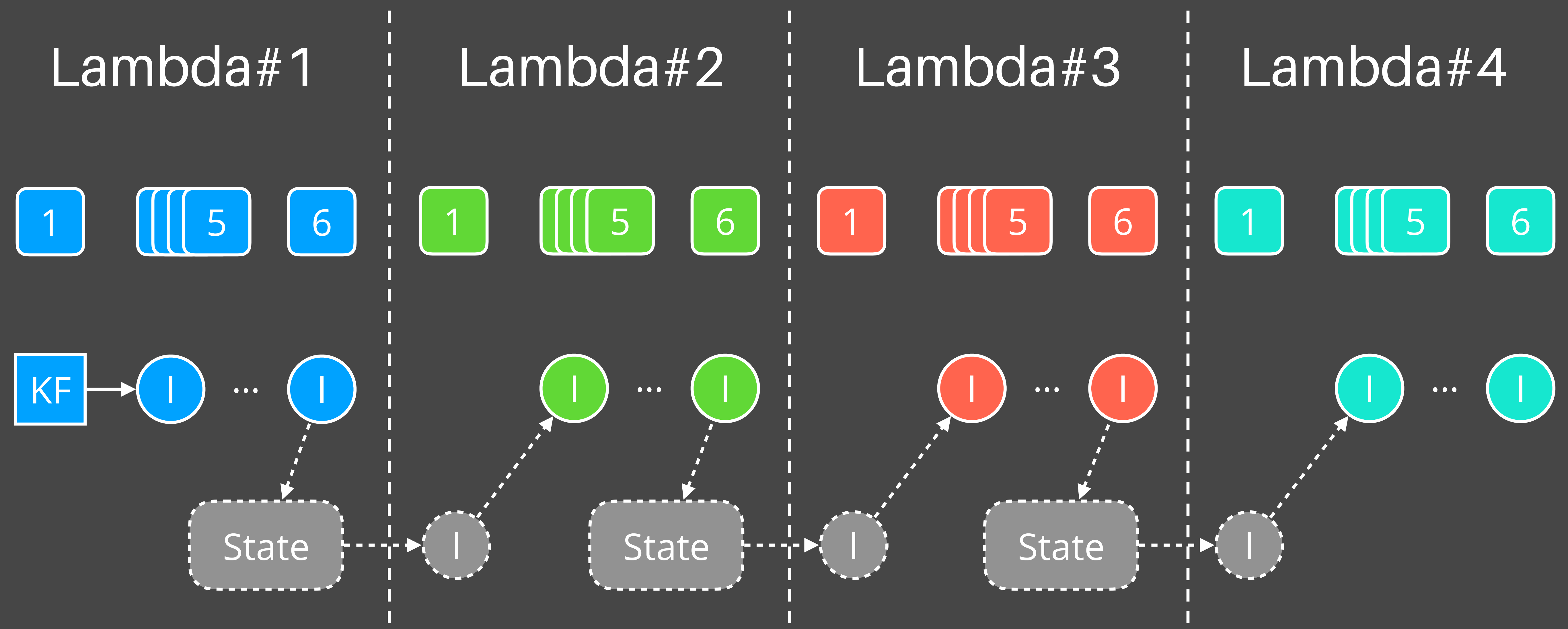
VIDEO ANALYTICS: EXCAMERA

VIDEO ENCODING/DECODING ON AWS LAMBDA



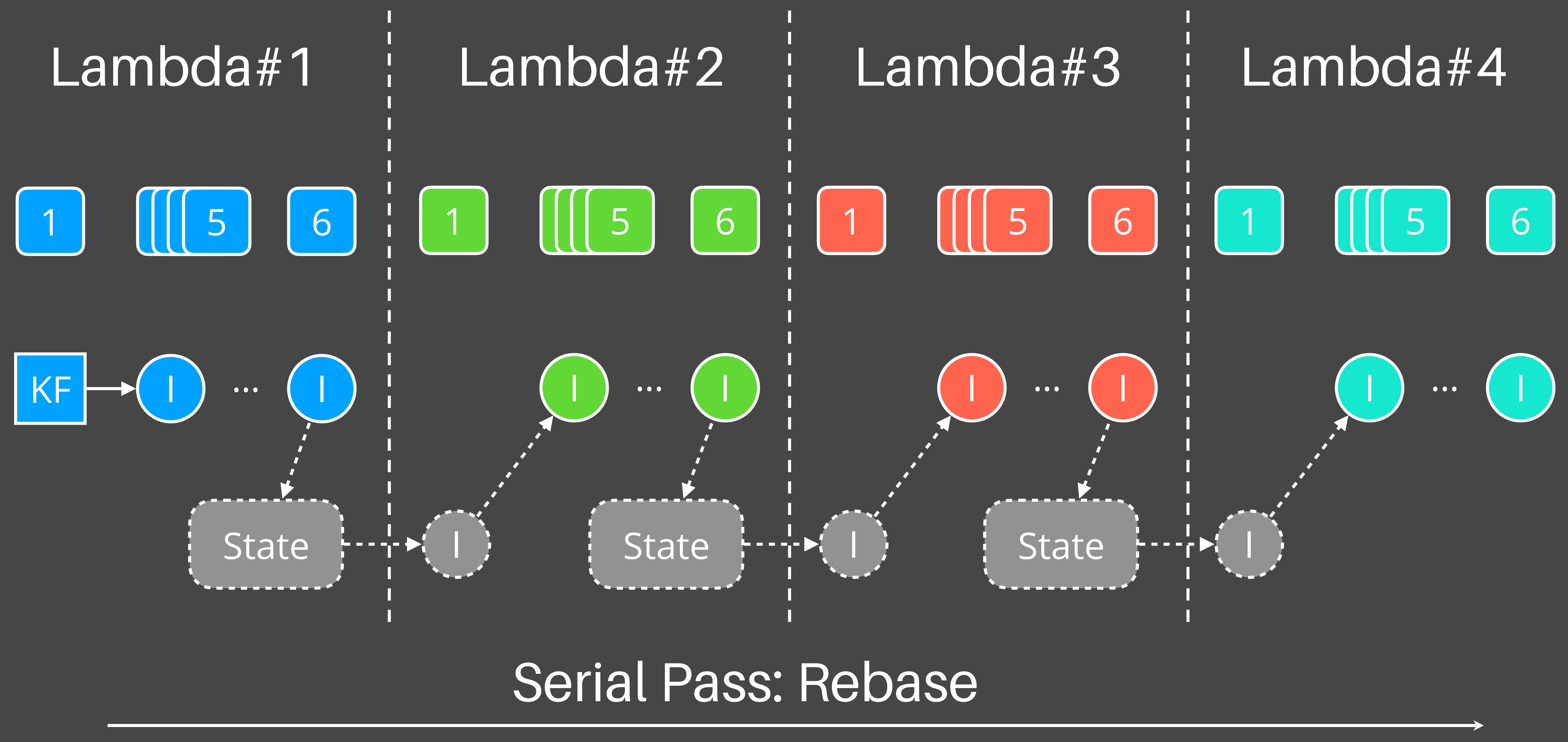
VIDEO ANALYTICS: EXCAMERA

VIDEO ENCODING/DECODING ON AWS LAMBDA



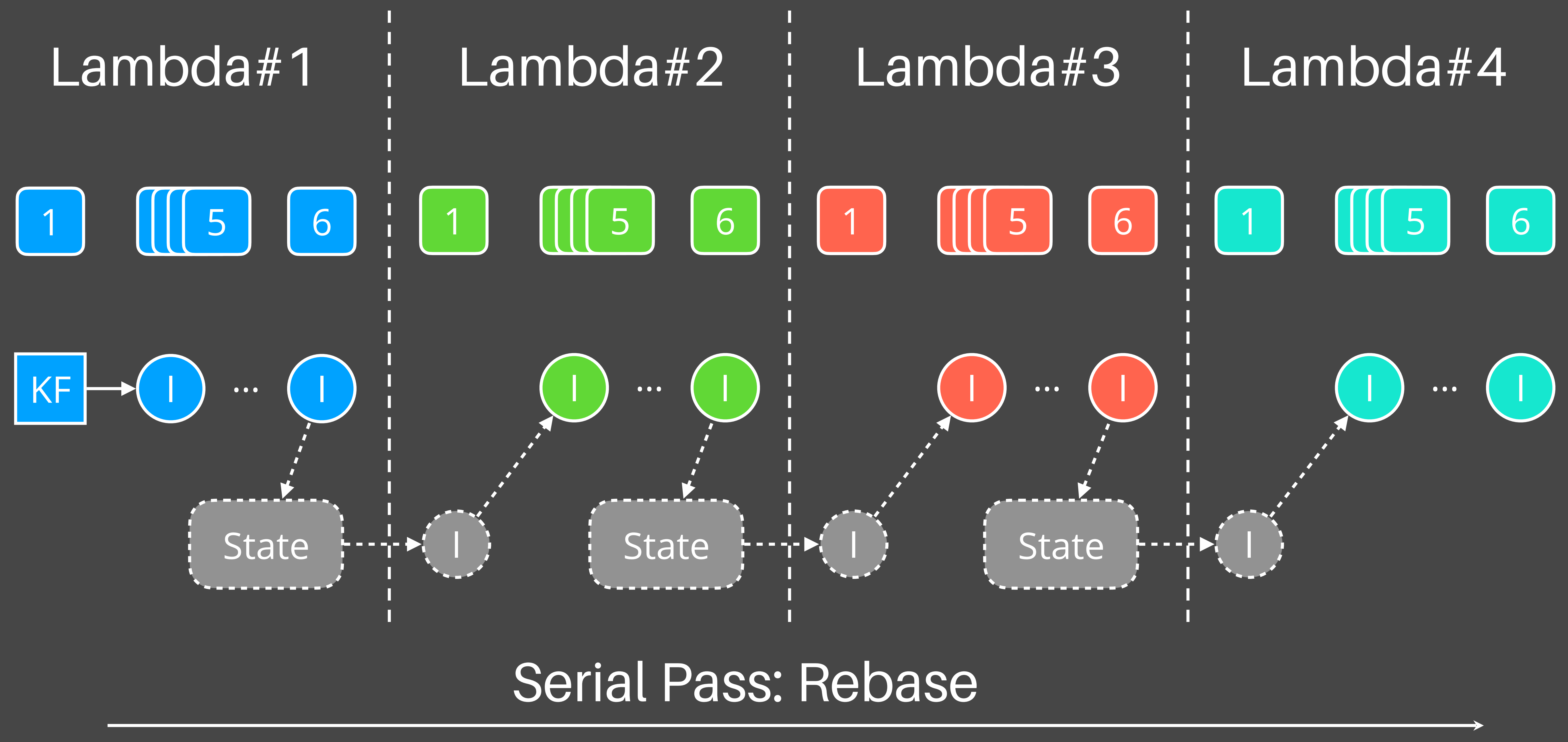
VIDEO ANALYTICS: EXCAMERA

VIDEO ENCODING/DECODING ON AWS LAMBDA



VIDEO ANALYTICS: EXCAMERA

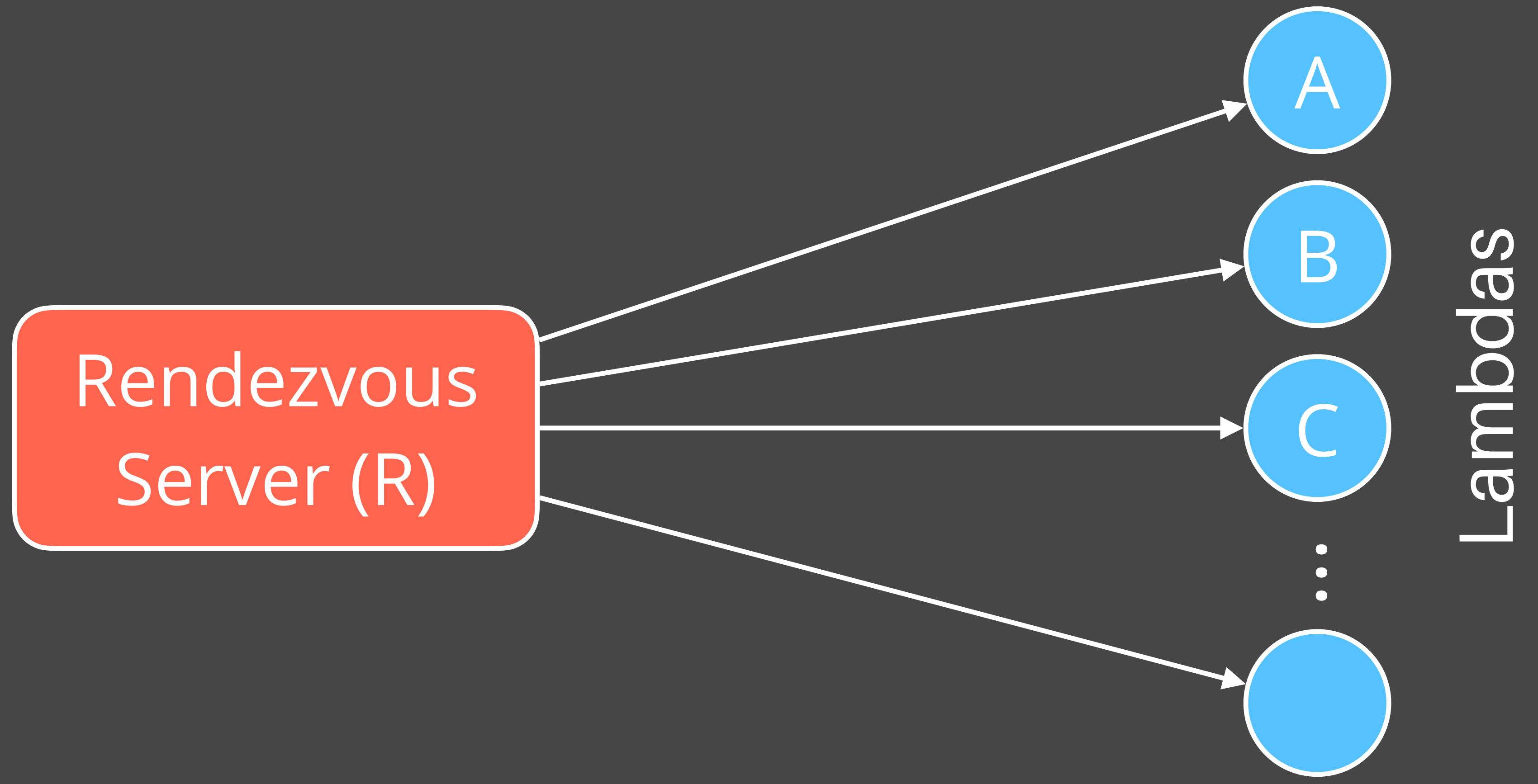
VIDEO ENCODING/DECODING ON AWS LAMBDA



- 60X faster and 6x cheaper than Google's **`vpxenc`** on 128 cores

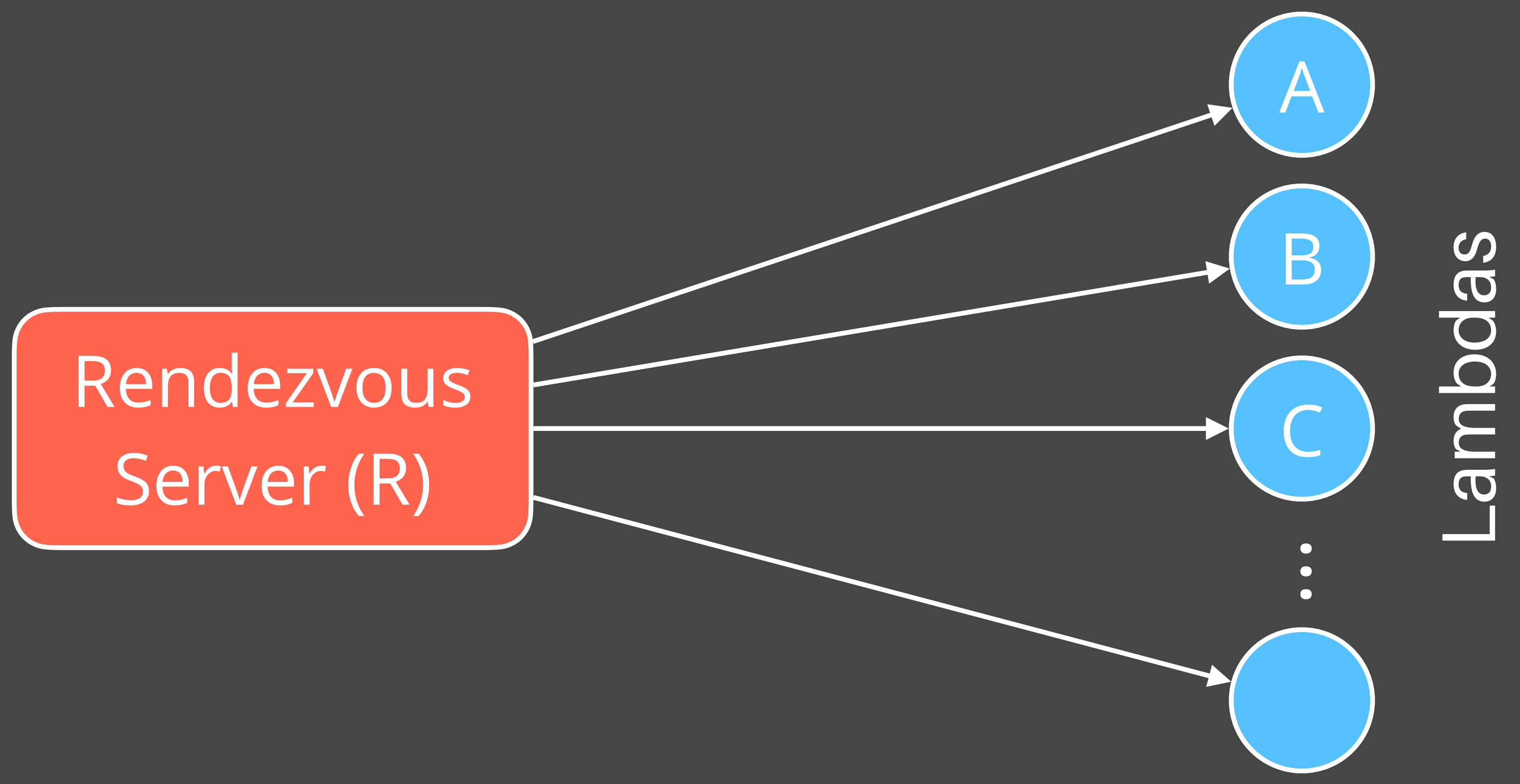
VIDEO ANALYTICS: EXCAMERA

Making lambdas talk to each other



VIDEO ANALYTICS: EXCAMERA

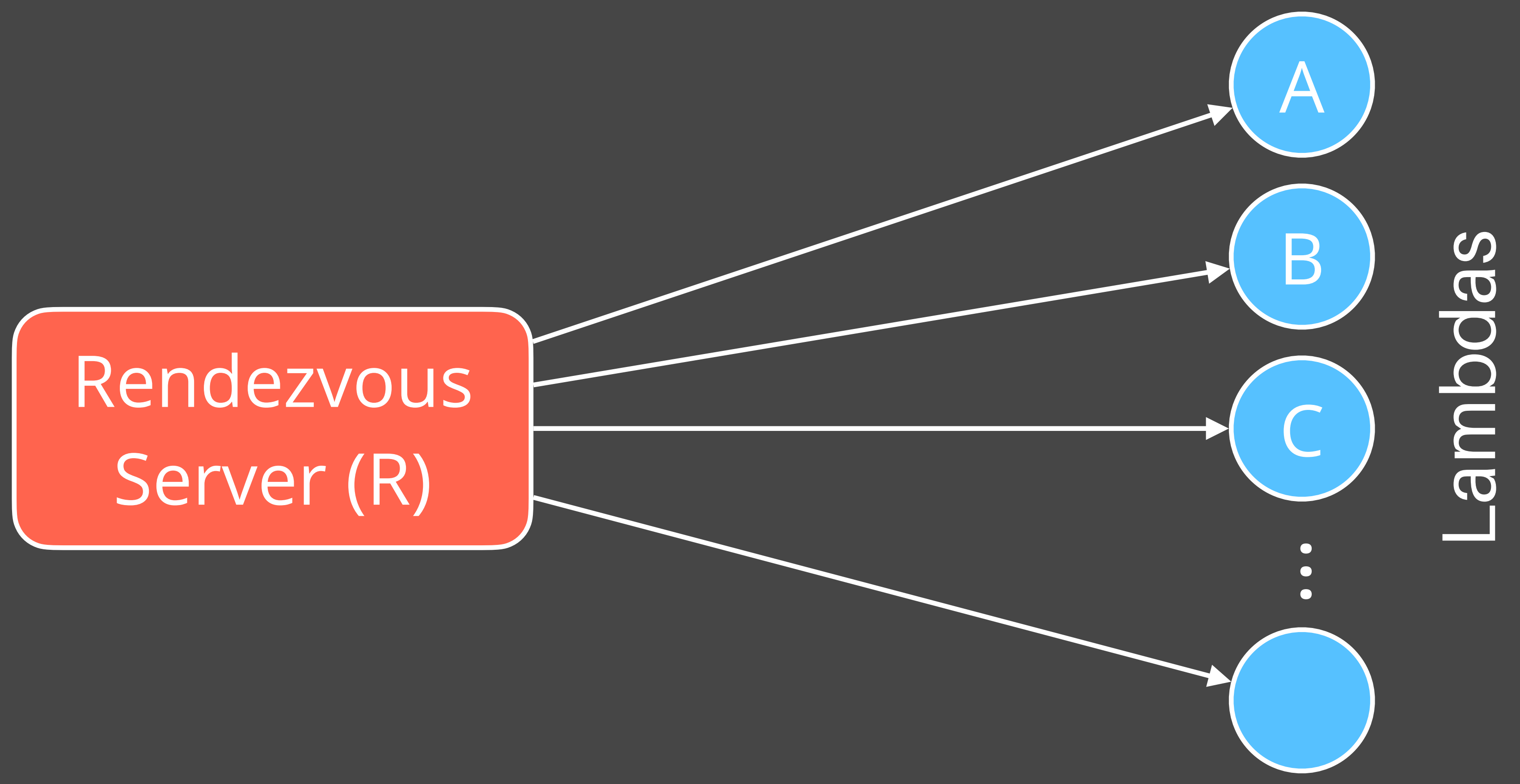
Making lambdas talk to each other



- Lambdas are only permitted ***outbound*** TCP/IP connections

VIDEO ANALYTICS: EXCAMERA

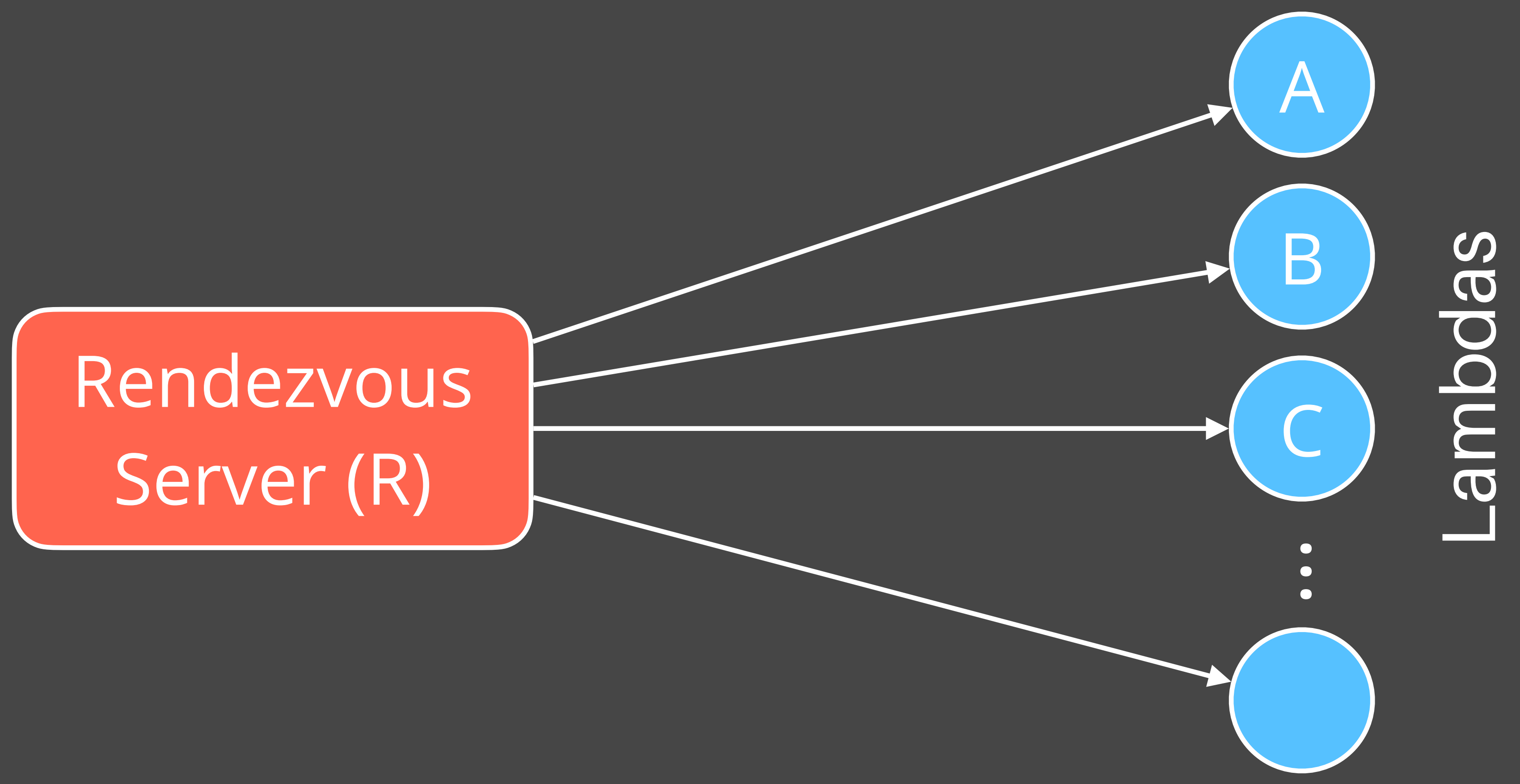
Making lambdas talk to each other



- Lambdas are only permitted ***outbound*** TCP/IP connections
- Establish outbound cxns to rendezvous server (R) at init

VIDEO ANALYTICS: EXCAMERA

Making lambdas talk to each other



- Lambdas are only permitted ***outbound*** TCP/IP connections
- Establish outbound cxns to rendezvous server (R) at init
- If A wants to talk to B, it sends R an init msg connect(A, B)
  - R forwards all of A's subsequent msgs to B

STATE STORAGE: CURRENT SOLUTIONS

Requirements

STATE STORAGE: CURRENT SOLUTIONS

Requirements

Low Latency,
High IOPS

STATE STORAGE: CURRENT SOLUTIONS

Requirements

Low Latency, High IOPS	Lifetime Management

STATE STORAGE: CURRENT SOLUTIONS

Requirements	Low Latency, High IOPS	Lifetime Management	Fine-grained Elasticity

STATE STORAGE: CURRENT SOLUTIONS

Requirements

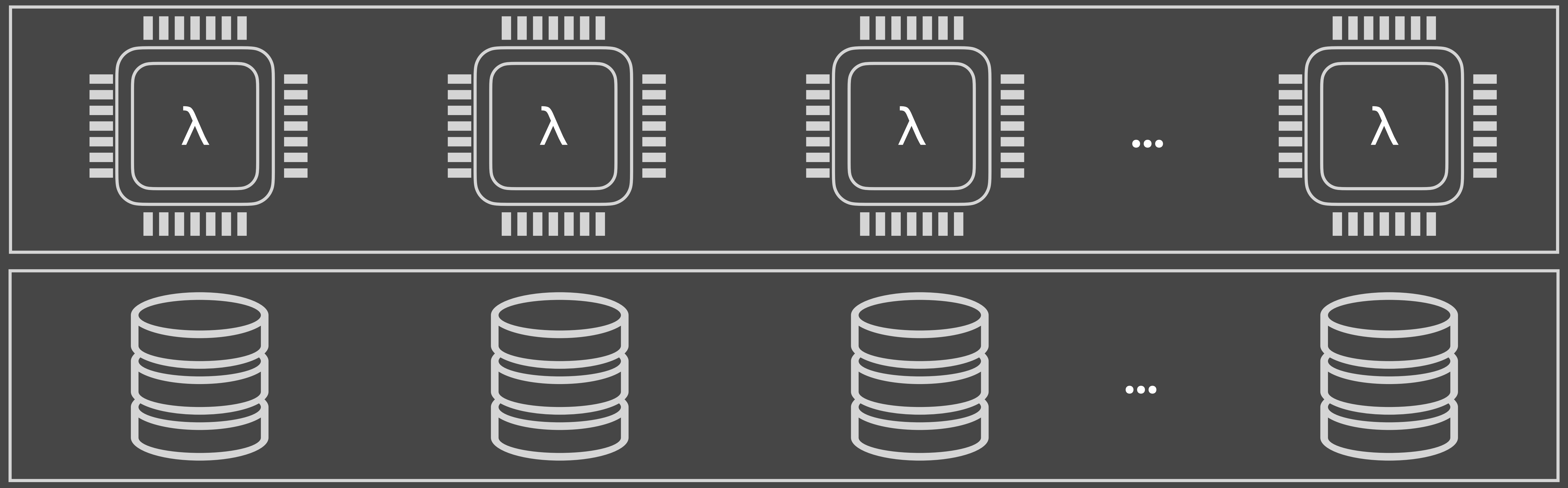
Low Latency, High IOPS

Lifetime Management

Fine-grained Elasticity

Stateful Tasks

Remote Persistent Storage (e.g., S3)



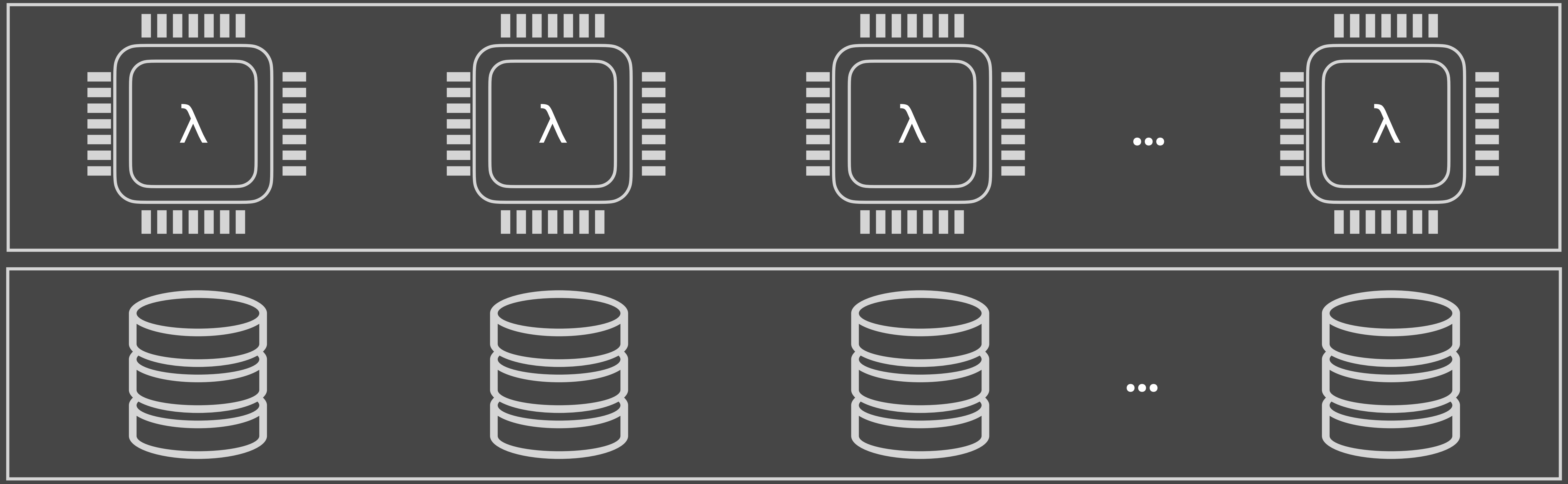
STATE STORAGE: CURRENT SOLUTIONS

Requirements

- ~~Low Latency, High IOPS~~
- ~~Lifetime Management~~
- Fine-grained Elasticity

Stateful Tasks

Remote Persistent Storage (e.g., S3)



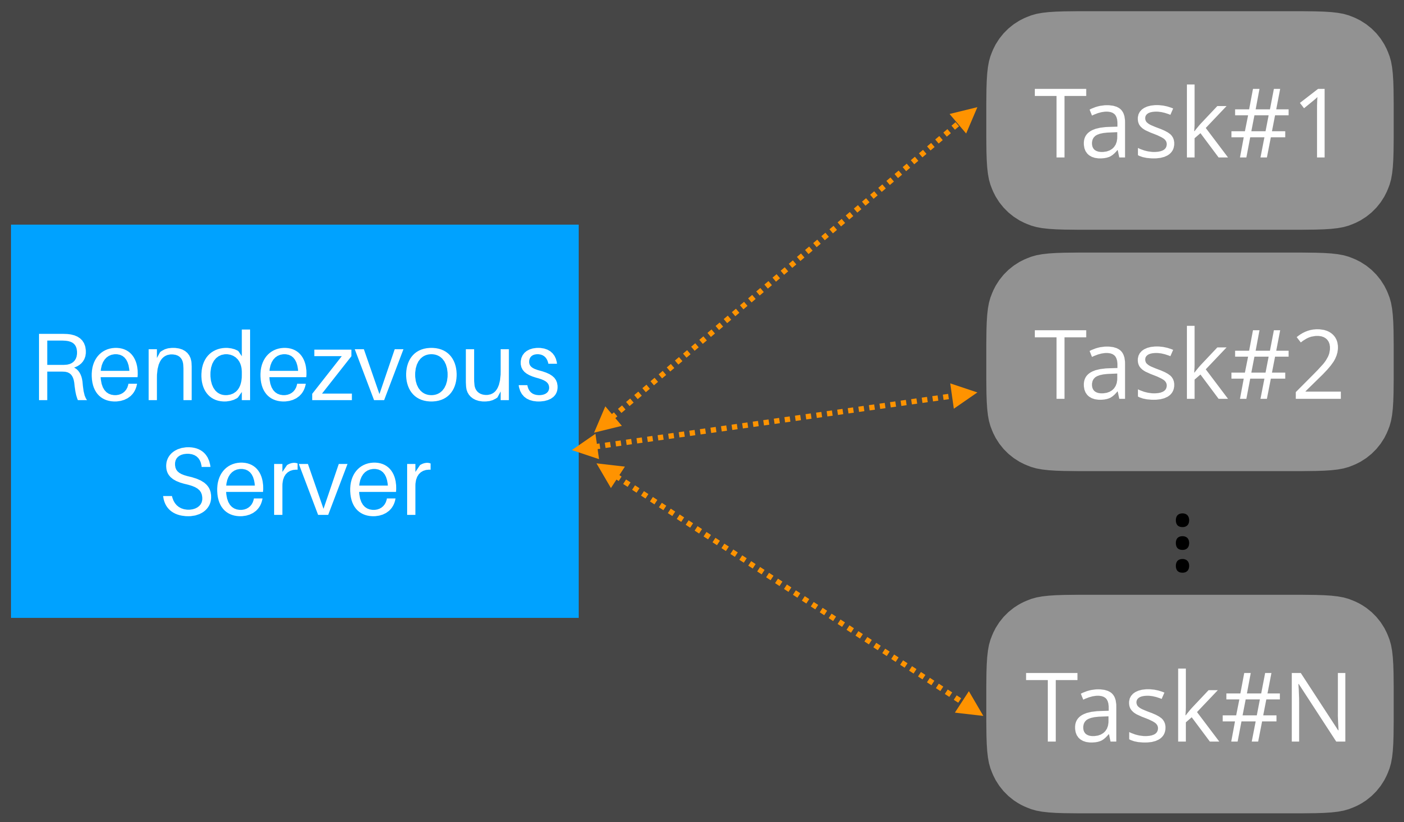
STATE STORAGE: CURRENT SOLUTIONS

Requirements

- Low Latency, High IOPS
- Lifetime Management
- Fine-grained Elasticity

Adhoc

Video Encoding in ExCamera [NSDI'17]



STATE STORAGE: CURRENT SOLUTIONS

Requirements

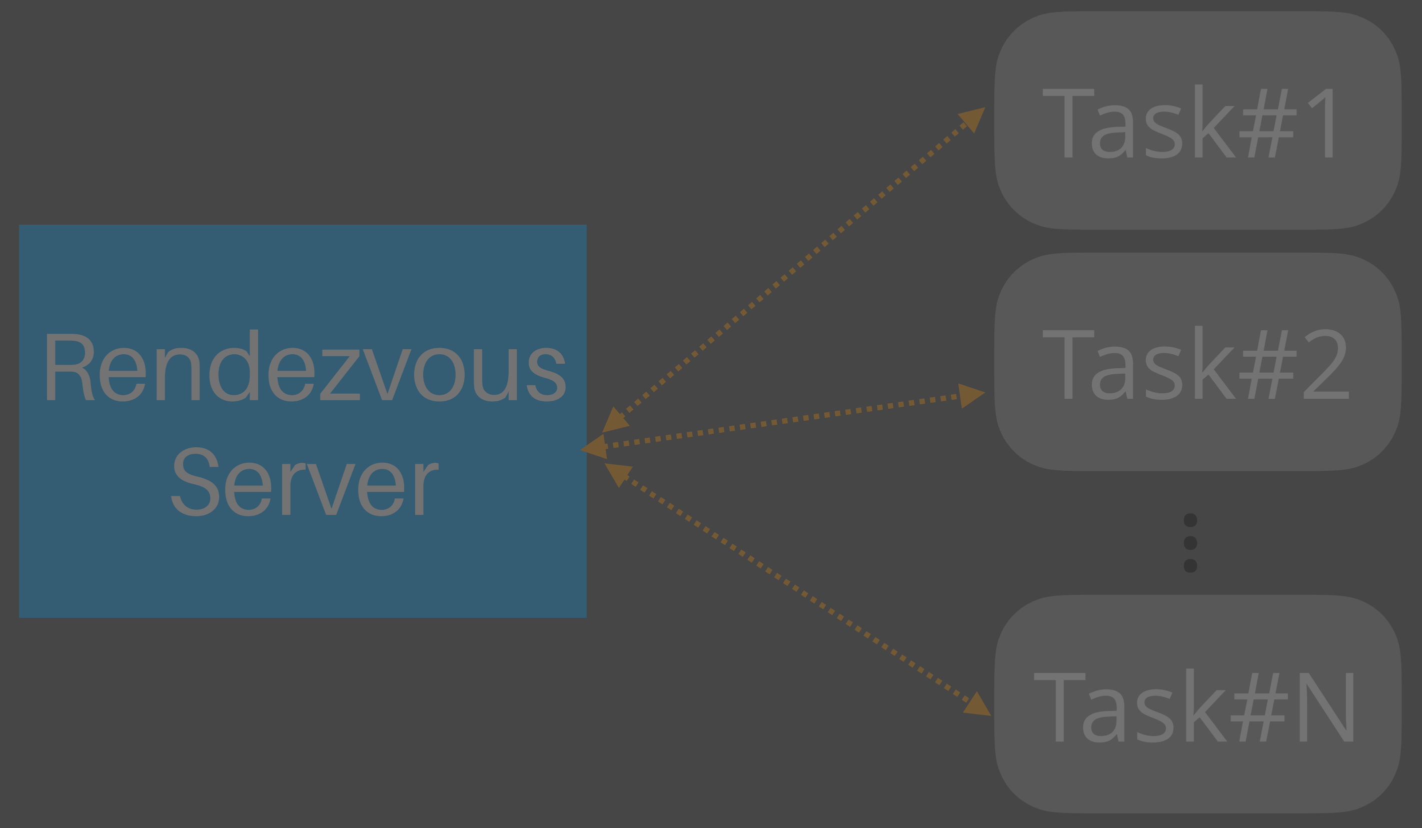
Low Latency, High IOPS

Lifetime Management

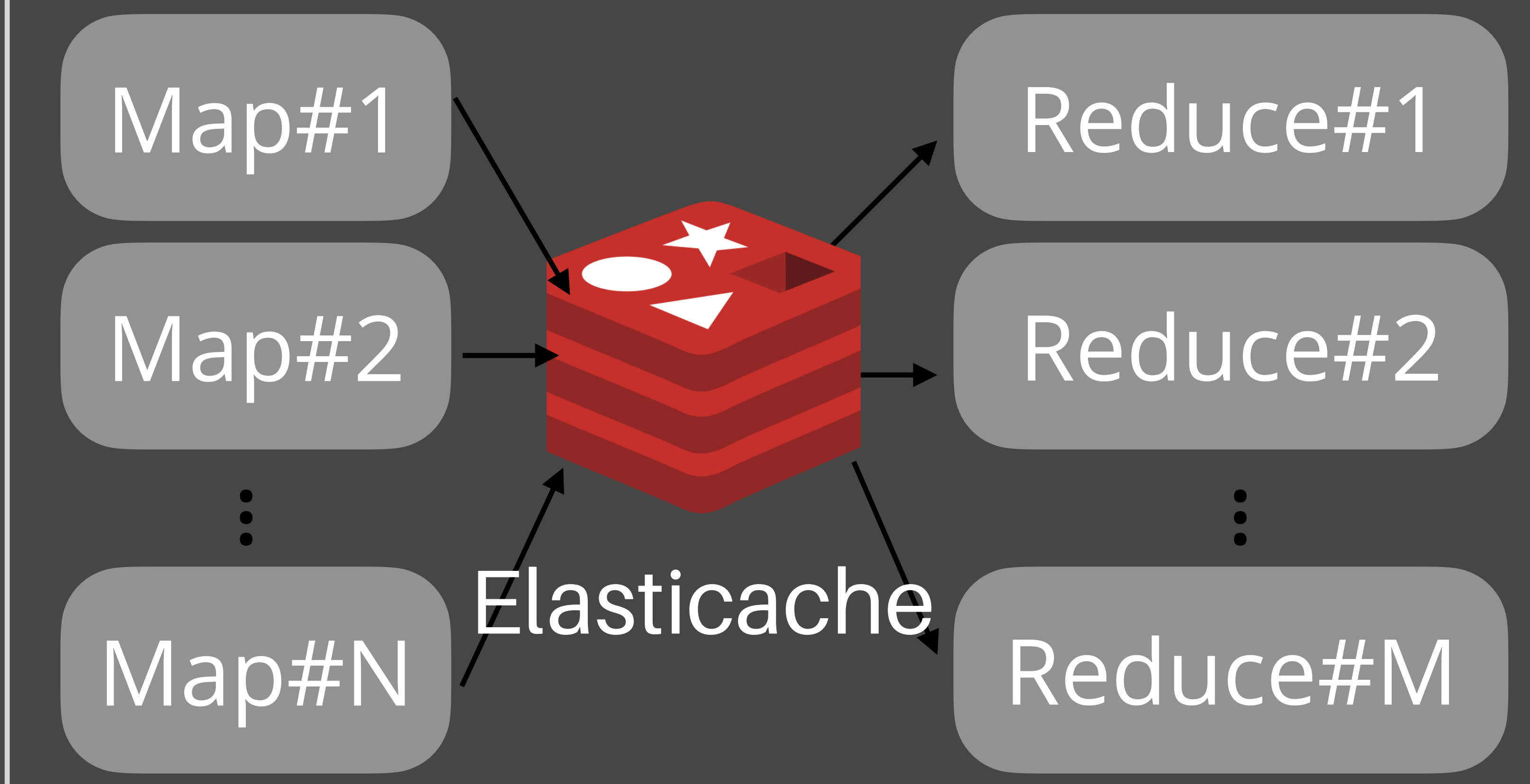
Fine-grained Elasticity

Adhoc

Video Encoding in ExCamera [NSDI'17]



Sorting data on PyWren using Locus [NSDI'19]



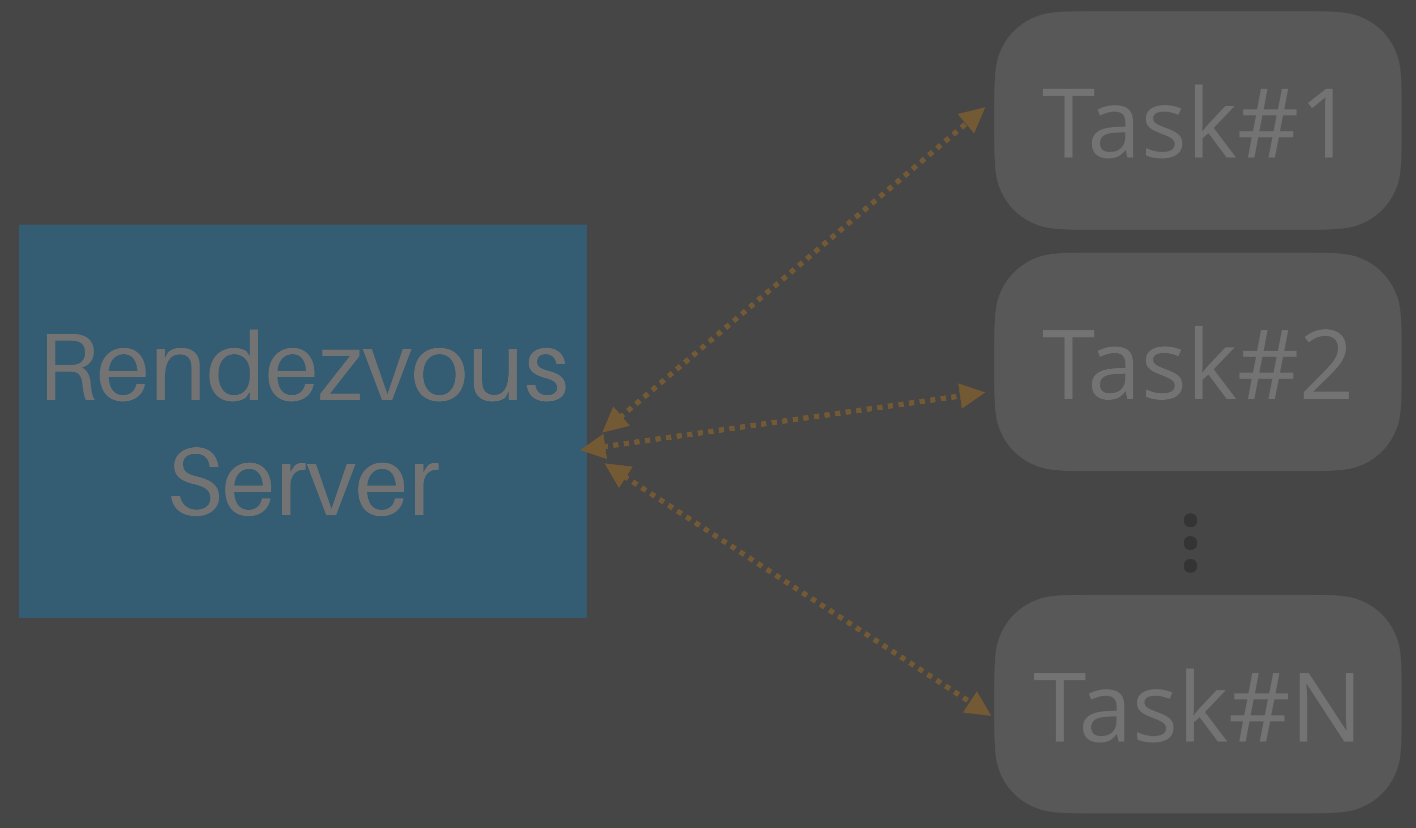
STATE STORAGE: CURRENT SOLUTIONS

Requirements

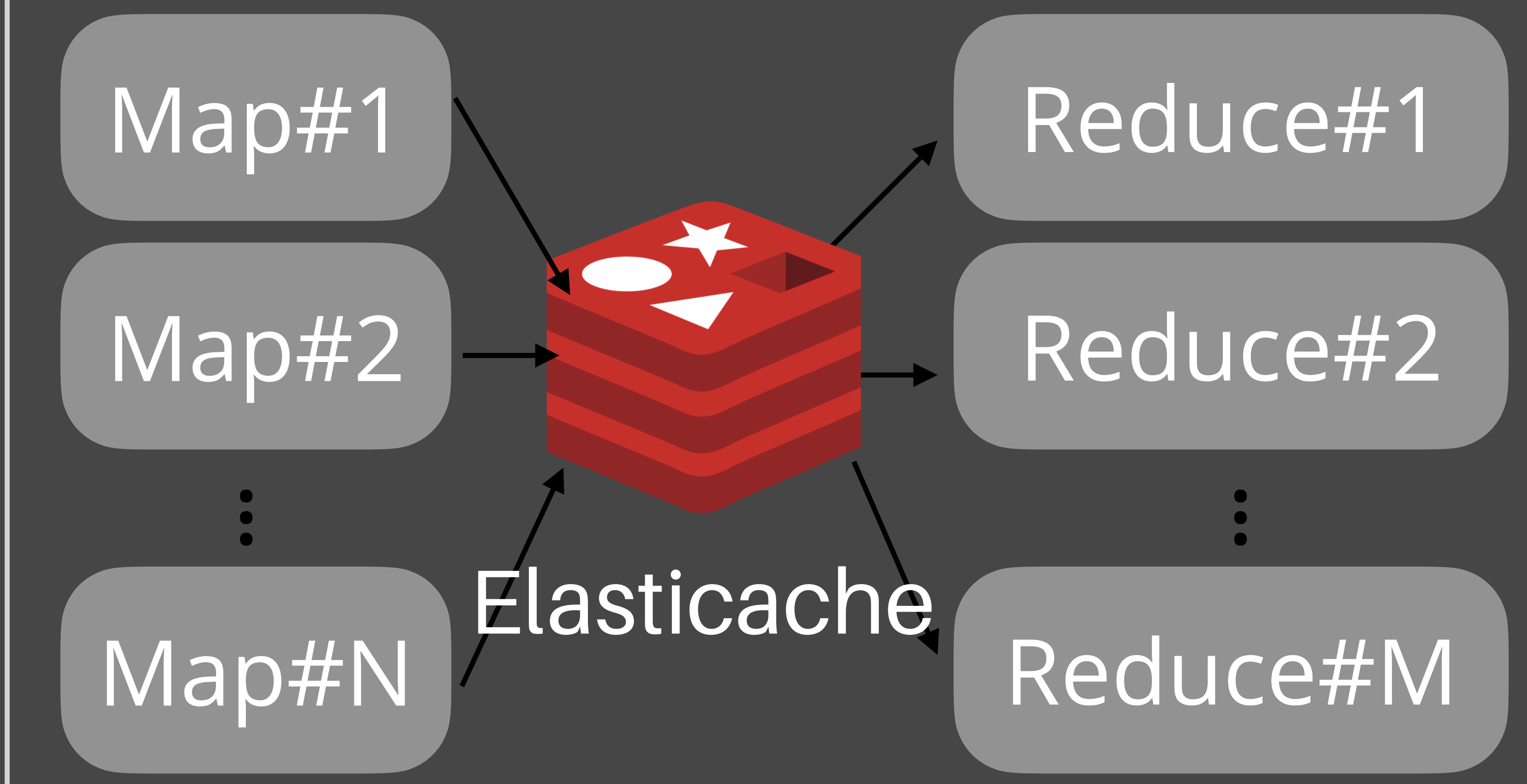
- Low Latency, High IOPS
- ~~Lifetime Management~~
- ~~Fine-grained Elasticity~~

Adhoc

Video Encoding in ExCamera [NSDI'17]



Sorting data on PyWren using Locus [NSDI'19]



STATE STORAGE: CURRENT SOLUTIONS

Requirements

Low Latency, High IOPS

Lifetime Management

Fine-grained Elasticity

Adhoc

General

Video Encoding in ExCamera [NSDI'17]

Sorting data on PyWren using Locus [NSDI'19]

STATE STORAGE: CURRENT SOLUTIONS

Requirements

- Low Latency, High IOPS
- ~~Lifetime Management~~
- Fine-grained Elasticity

Adhoc

Video Encoding in ExCamera [NSDI'17]

Sorting data on PyWren using Locus [NSDI'19]

General

Anna [VLDB'19, IEEE TKDE'19]

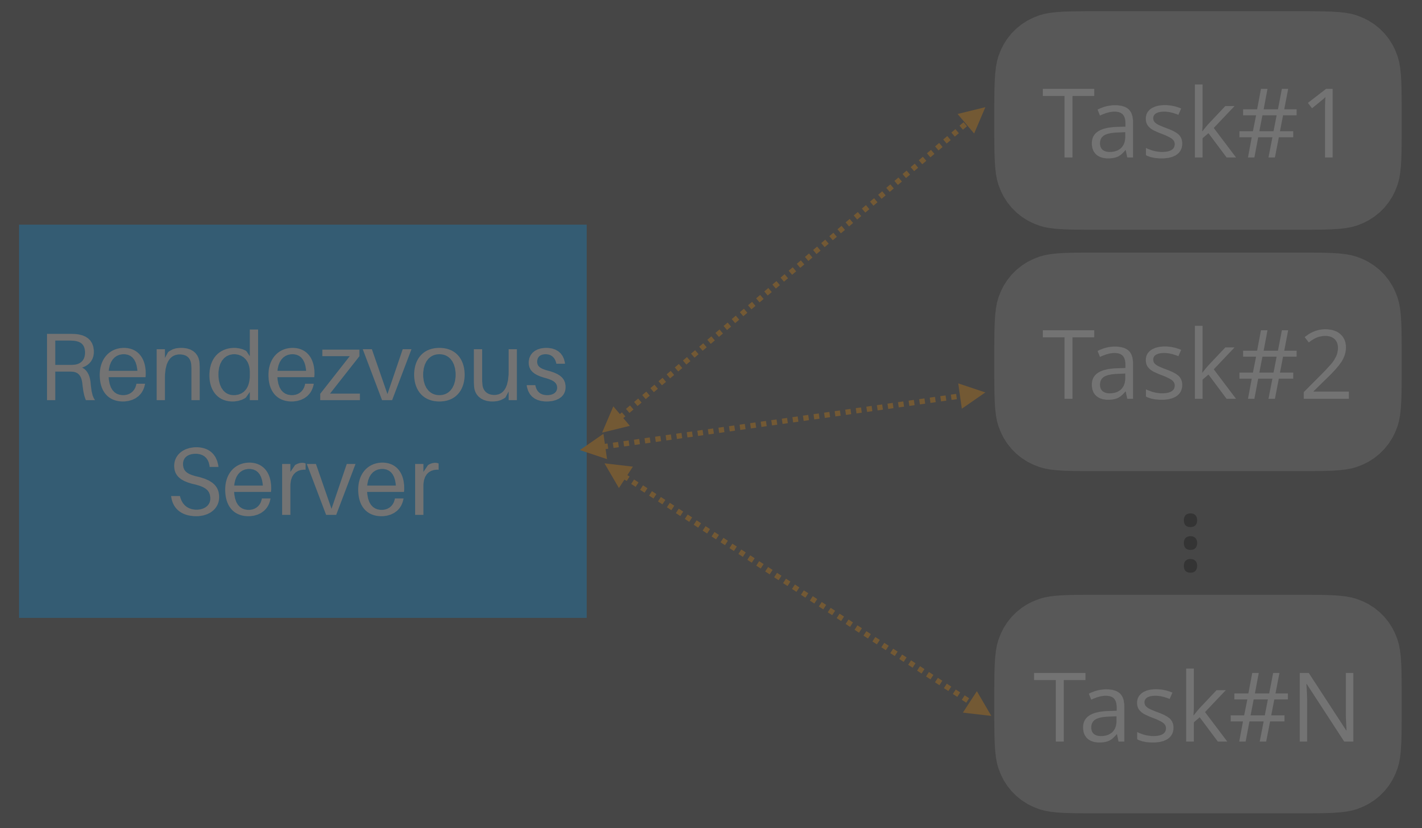
STATE STORAGE: CURRENT SOLUTIONS

Requirements

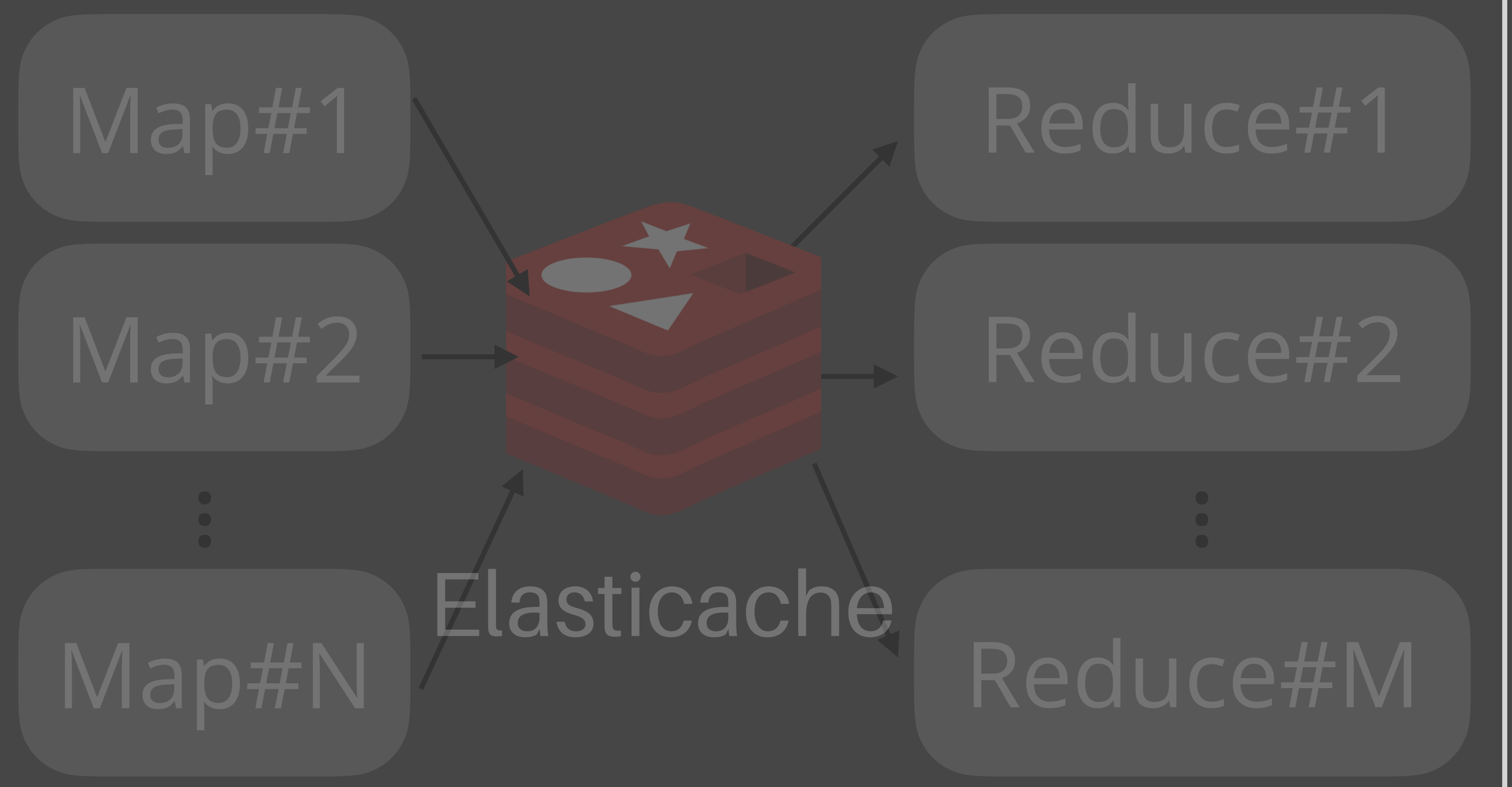
- Low Latency, High IOPS
- Lifetime Management
- ~~Fine-grained Elasticity~~

Adhoc

Video Encoding in ExCamera [NSDI'17]



Sorting data on PyWren using Locus [NSDI'19]

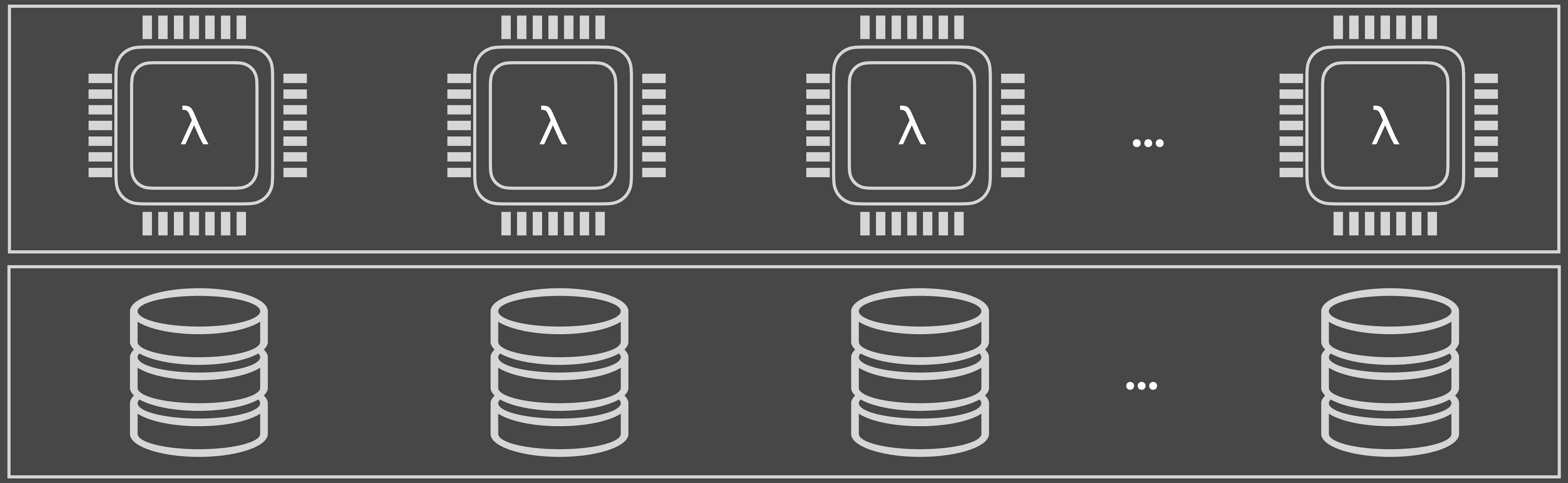


General

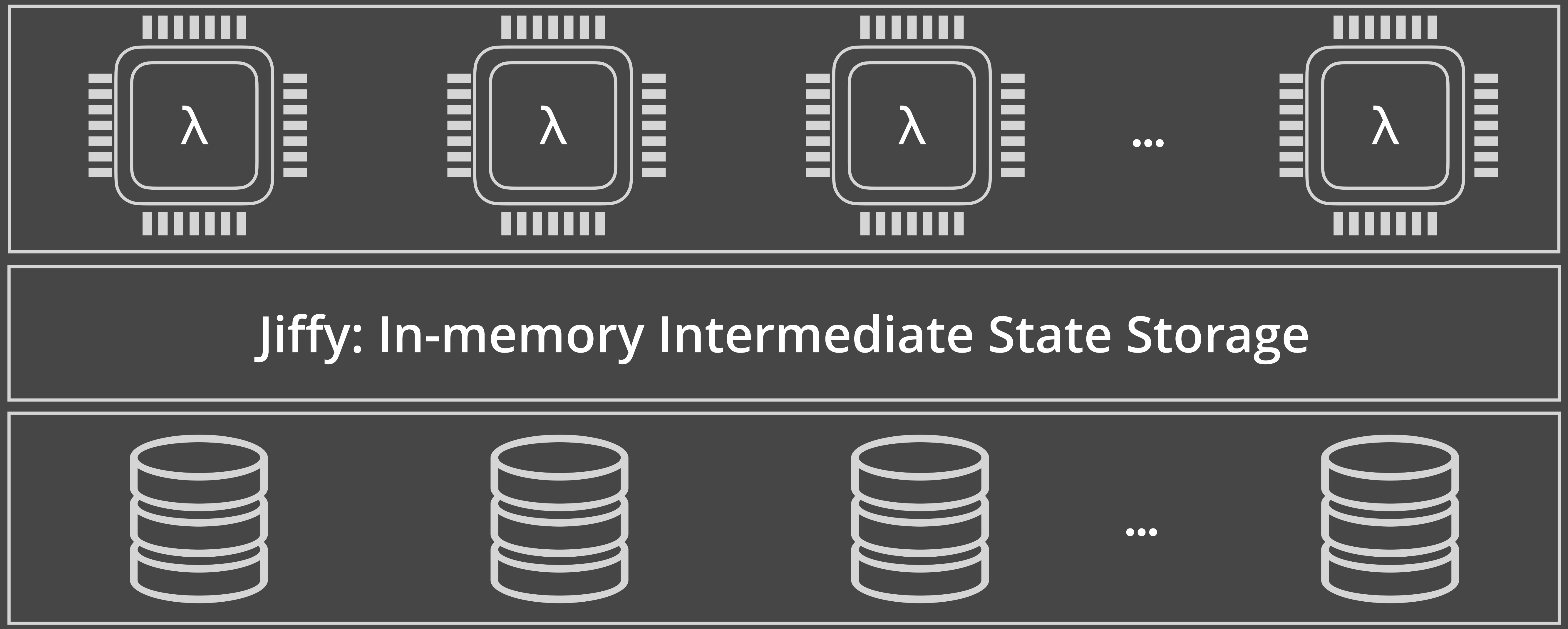
Anna [VLDB'19, IEEE TKDE'19]

Pocket [OSDI'18]

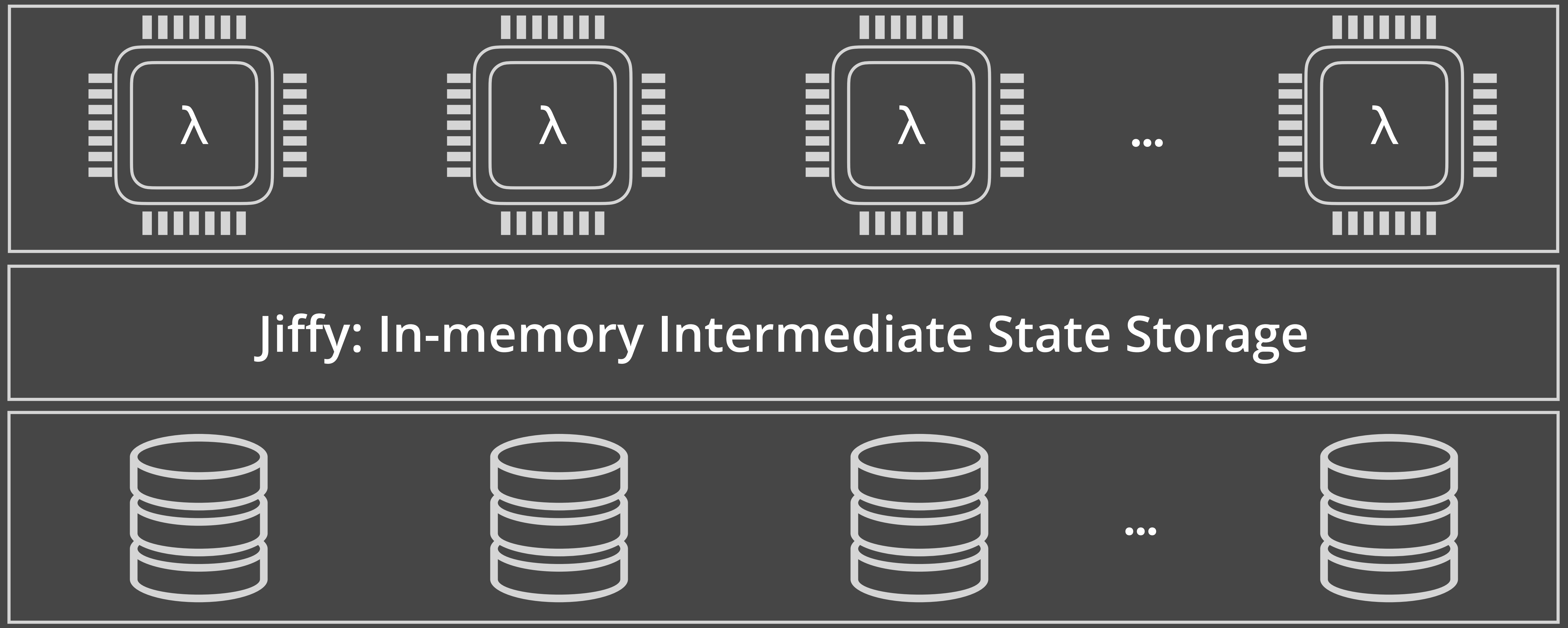
JIFFY: INTERMEDIATE STATE STORAGE & MANAGEMENT



JIFFY: INTERMEDIATE STATE STORAGE & MANAGEMENT



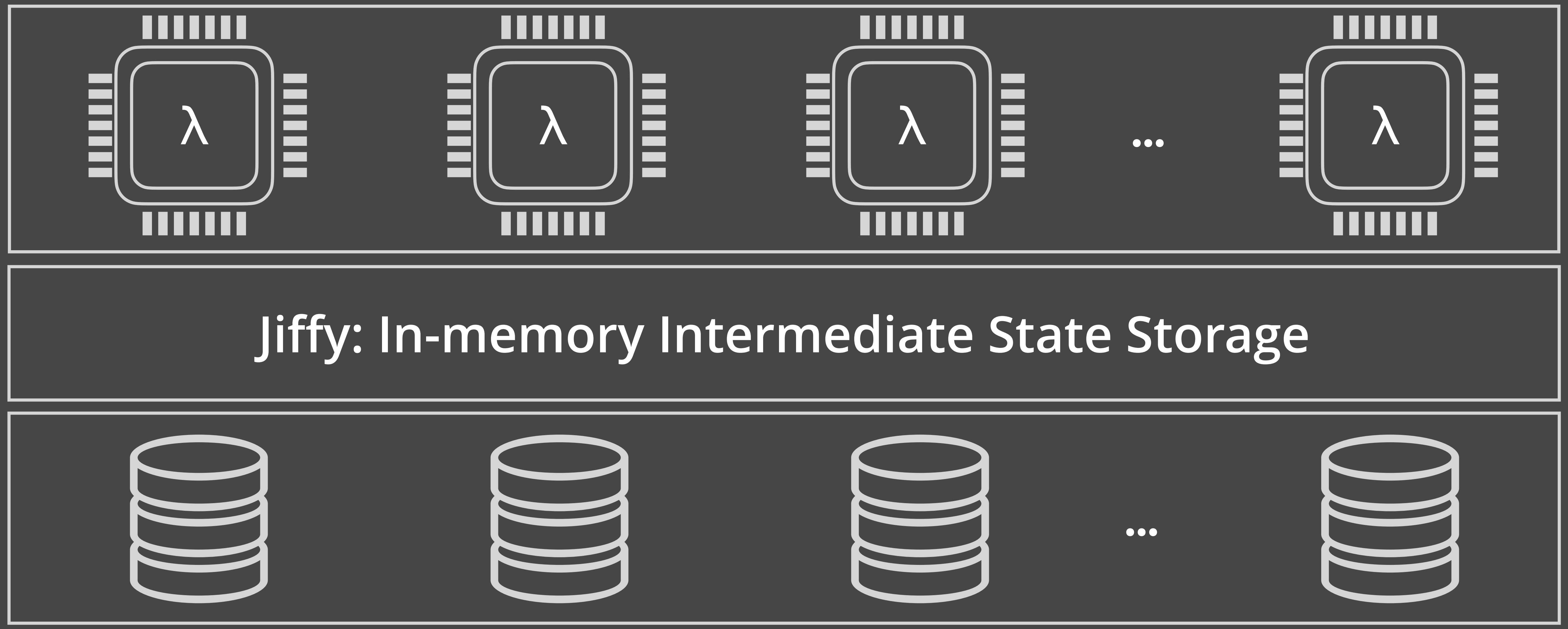
JIFFY: INTERMEDIATE STATE STORAGE & MANAGEMENT



Application: Scale intermediate storage resources independent of other resources

Cloud Provider: Multiplex intermediate storage for high utilization

JIFFY: INTERMEDIATE STATE STORAGE & MANAGEMENT

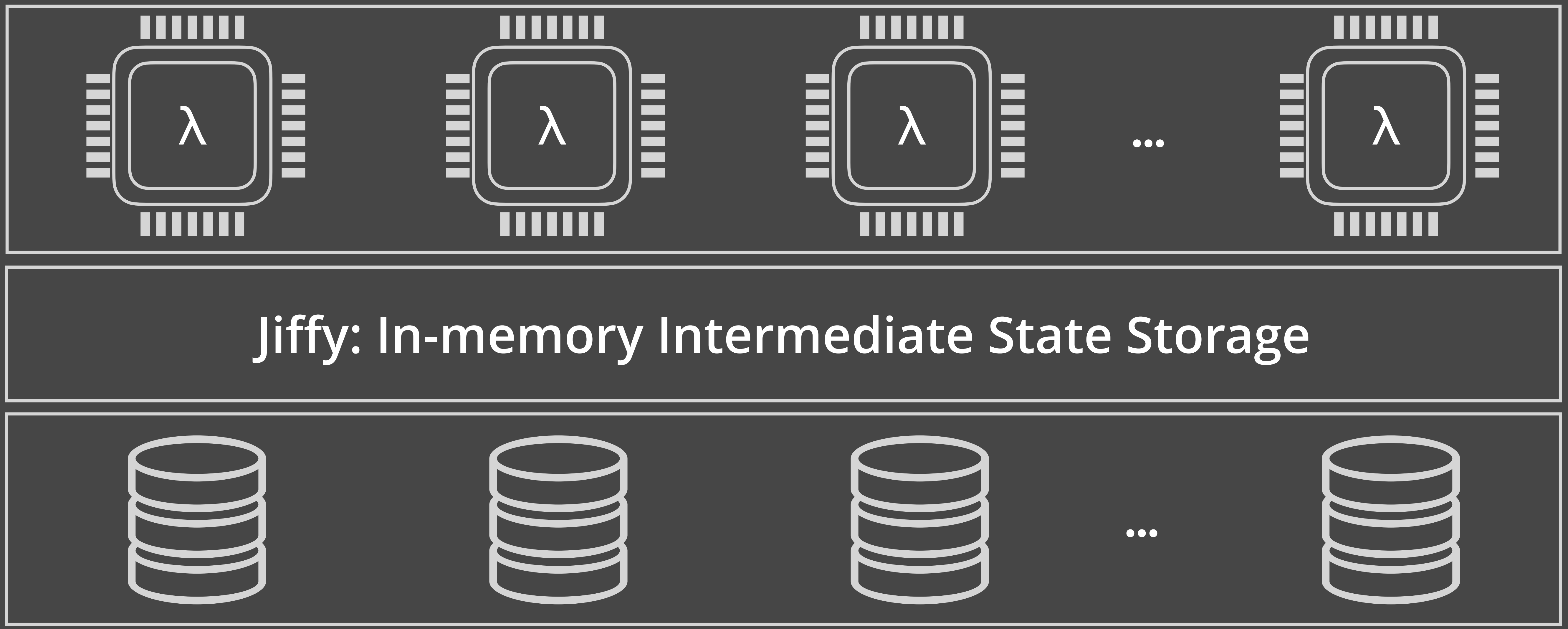


Application: Scale intermediate storage resources independent of other resources

Cloud Provider: Multiplex intermediate storage for high utilization

Challenges:

JIFFY: INTERMEDIATE STATE STORAGE & MANAGEMENT



Application: Scale intermediate storage resources independent of other resources

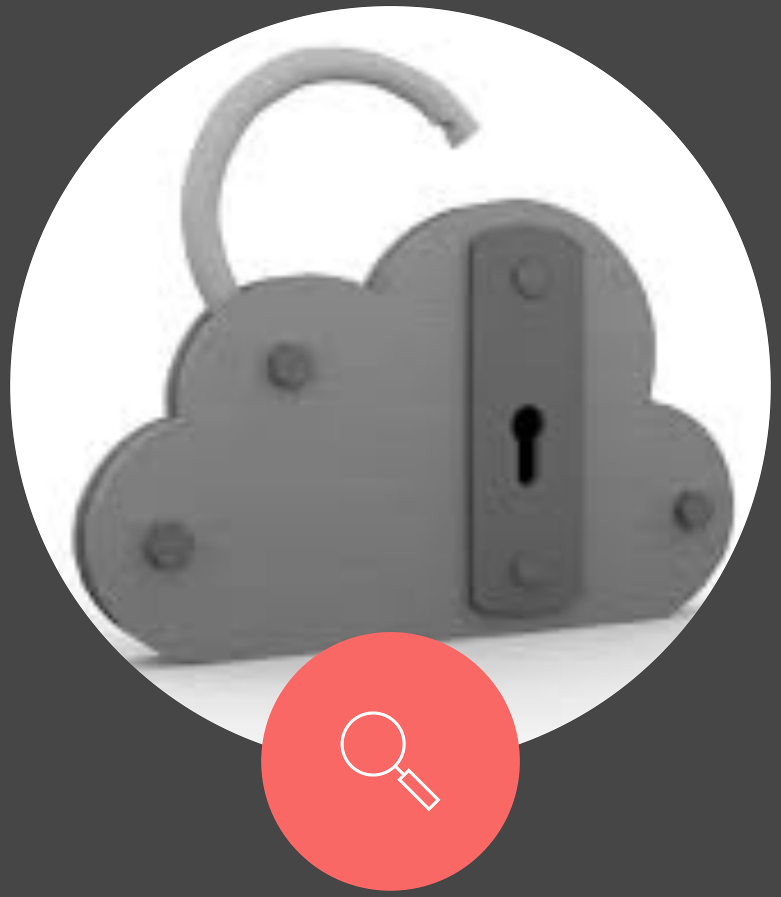
Cloud Provider: Multiplex intermediate storage for high utilization

Challenges:

- What is the right interface?
- How can we share intermediate storage across applications with isolation?
- How should we manage lifetimes of application storage?
- How to facilitate efficient communication across tasks?

SERVERLESS: BEYOND STATE MANAGEMENT

SERVERLESS: BEYOND STATE MANAGEMENT

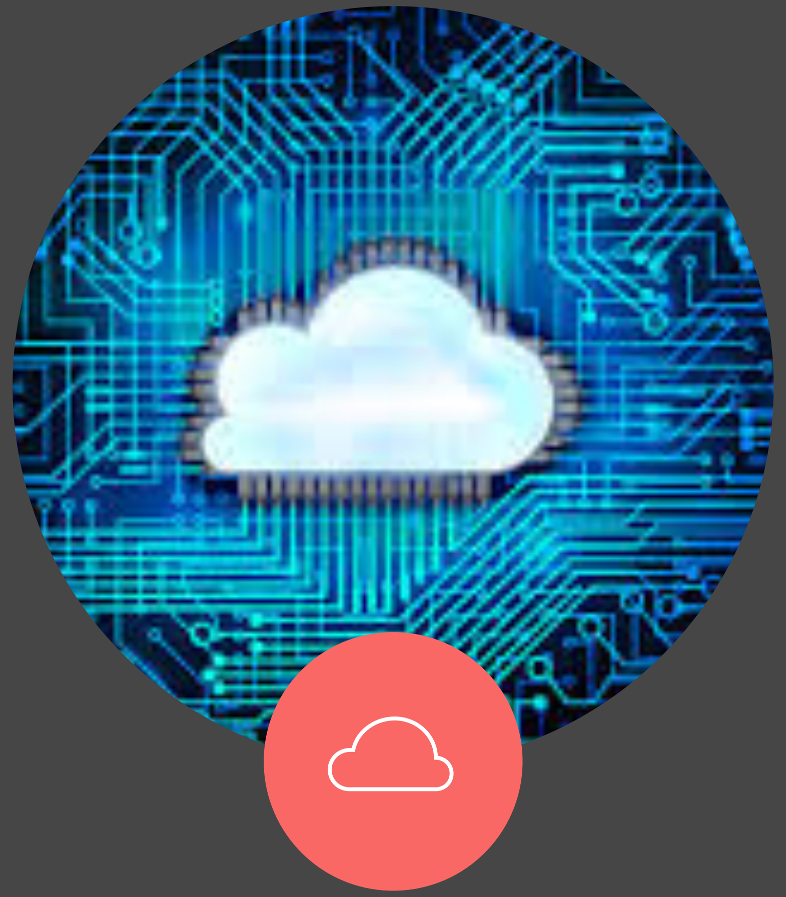


Security

Side-channels, Information leakage via network communications

SLA Guarantees

Performance guarantees, Performance isolation



Heterogenous Hardware

FPGAs, GPUs, TPUs, etc

SERVERLESS IS THE FUTURE

SERVERLESS IS THE FUTURE

- Serverless cloud hides management complexity, facilitates consumption-based billing and automatic scaling, benefiting:
  - Developers & Users (simpler programming, no management, lower costs)
  - Cloud providers (high resource utilization)

SERVERLESS IS THE FUTURE

- Serverless cloud hides management complexity, facilitates consumption-based billing and automatic scaling, benefiting:
  - Developers & Users (simpler programming, no management, lower costs)
  - Cloud providers (high resource utilization)
- Today's Serverless Cloud is stateless and event-driven

SERVERLESS IS THE FUTURE

- Serverless cloud hides management complexity, facilitates consumption-based billing and automatic scaling, benefiting:
  - Developers & Users (simpler programming, no management, lower costs)
  - Cloud providers (high resource utilization)
- Today's Serverless Cloud is stateless and event-driven
- Future Serverless Cloud will address state management
  - Also: Security, SLA guarantees, Heterogenous hardware.

SERVERLESS IS THE FUTURE

- Serverless cloud hides management complexity, facilitates consumption-based billing and automatic scaling, benefiting:
  - Developers & Users (simpler programming, no management, lower costs)
  - Cloud providers (high resource utilization)
- Today's Serverless Cloud is stateless and event-driven
- Future Serverless Cloud will address state management
  - Also: Security, SLA guarantees, Heterogenous hardware.


Thank You..

Thank You...

Questions?