AWS re:Invent

NOV. 29 – DEC. 3, 2021 | LAS VEGAS, NV

DAT303

Amazon DynamoDB: Driving innovation at any scale

Oscar Mullin (he/him/his)
Director of Engineering – Core Services, DBA & SRE
Mercado Libre

Pete Naylor (he/him/his)
Sr. Technical Product Manager, Amazon DynamoDB
AWS

© 2021, Amazon Web Services, Inc. or its affiliates. All rights reserved.

Session warmup

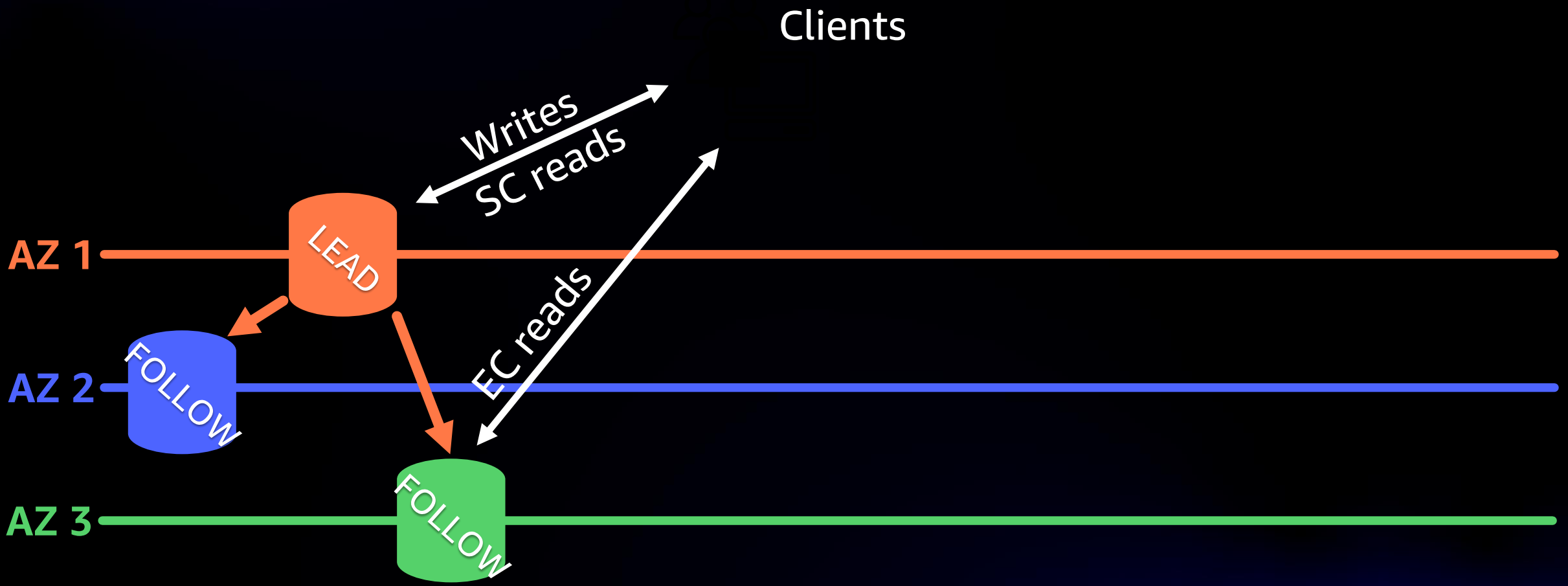
- How to think about Amazon DynamoDB
- Scaling elasticity and why it matters
- Availability and resilience in DynamoDB context
- Isolation and infrastructure automation – table, not cluster
- Adoption patterns at AWS customers with many teams

Mercado Libre (MELI) innovates faster with DynamoDB

aws

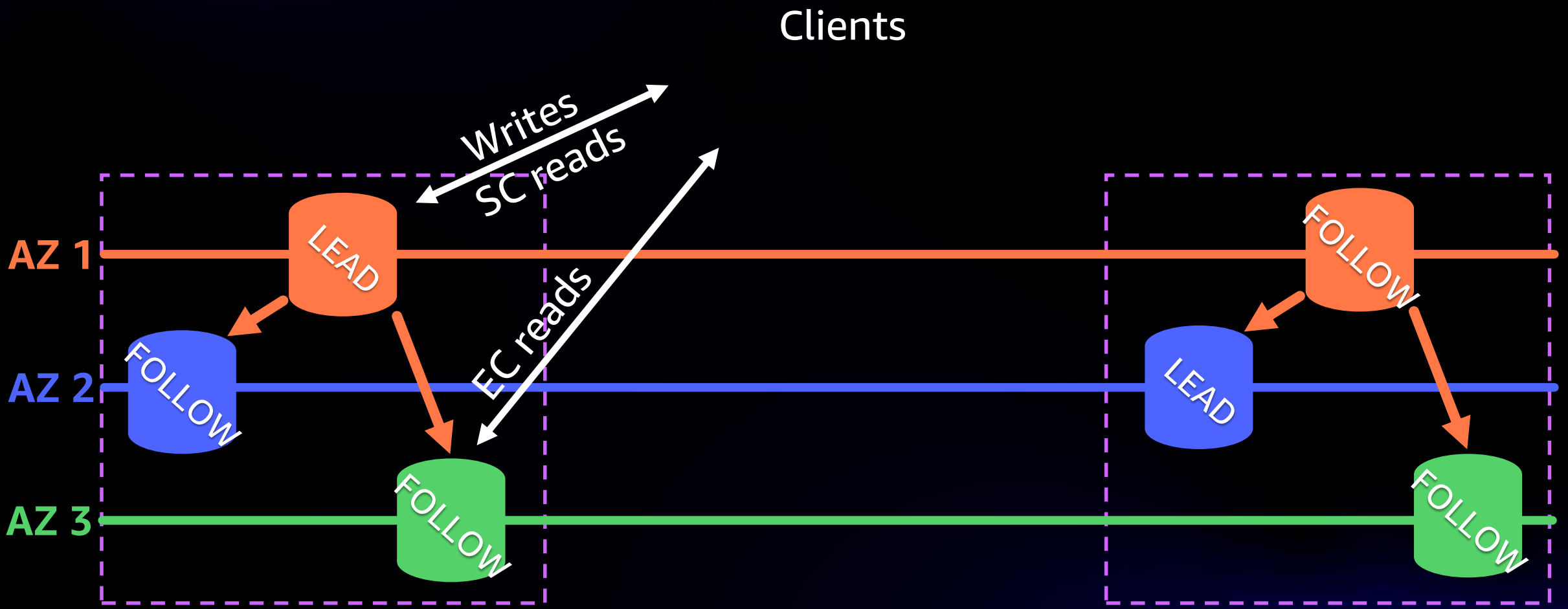
© 2021, Amazon Web Services, Inc. or its affiliates. All rights reserved.

Typical transactional database architecture



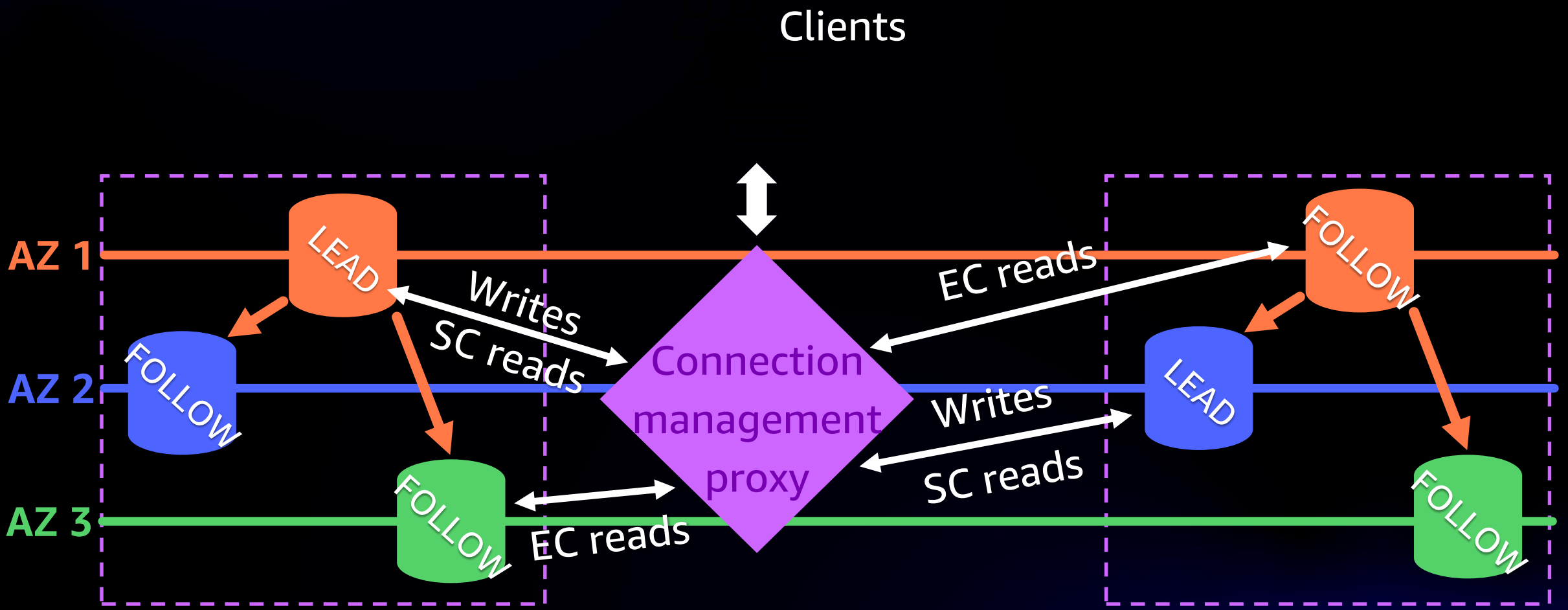
© 2021, Amazon Web Services, Inc. or its affiliates. All rights reserved.

Sharded transactional database architecture



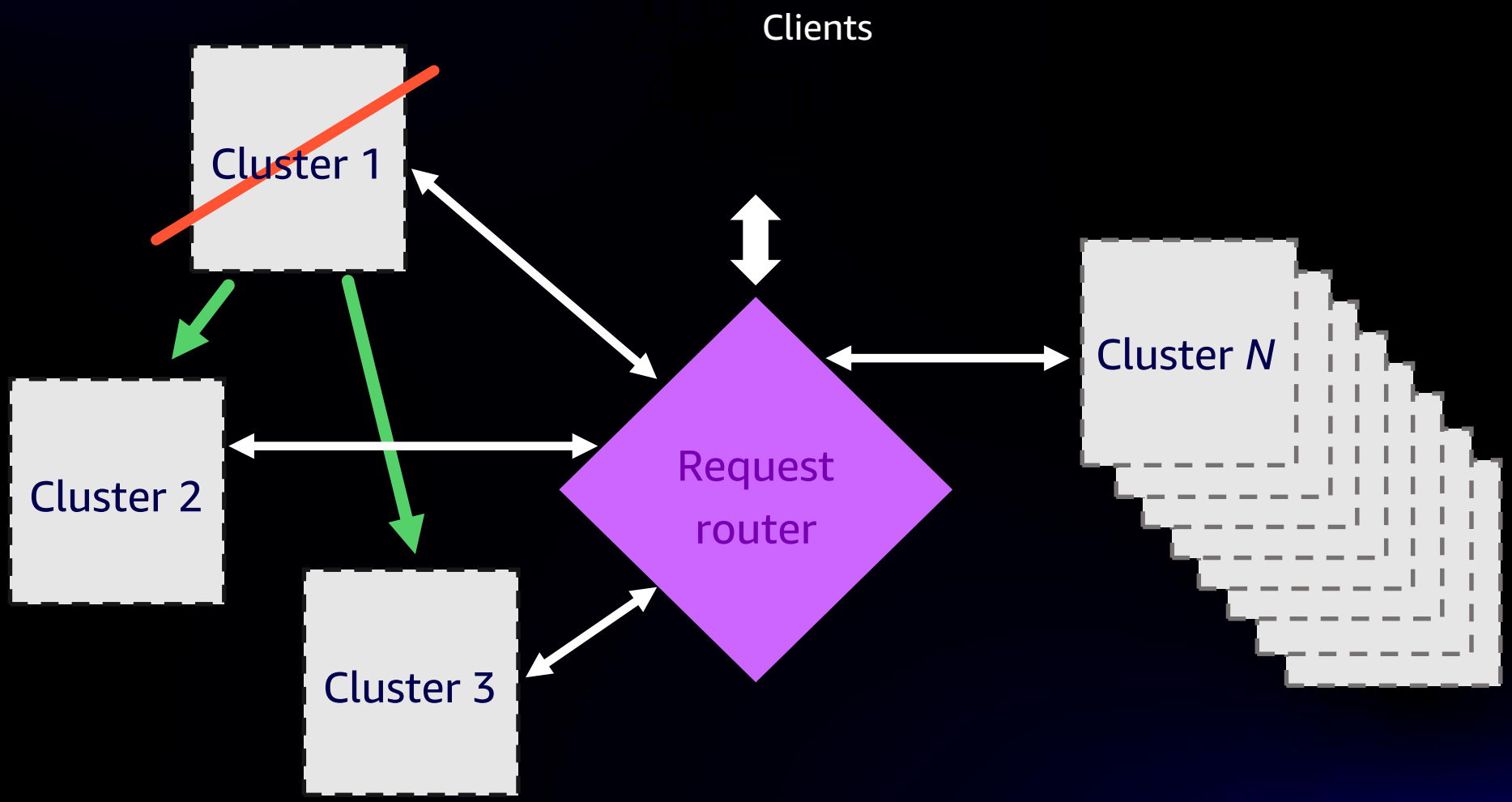
© 2021, Amazon Web Services, Inc. or its affiliates. All rights reserved.

Sharded database request distribution



© 2021, Amazon Web Services, Inc. or its affiliates. All rights reserved.

Managed shard-first approach with DynamoDB



© 2021, Amazon Web Services, Inc. or its affiliates. All rights reserved.

Enabling development teams with DynamoDB

- Organizations and teams structured around services
- Isolated data store per service
- Organizational standards are an important simplification
- Centralization of expertise with platform teams
- Consistent feature components to fit organizational culture

© 2021, Amazon Web Services, Inc. or its affiliates. All rights reserved.

Questions for you!

© 2021, Amazon Web Services, Inc. or its affiliates. All rights reserved.

Mercado Libre's journey

IN THIS SESSION, YOU WILL LEARN HOW WE ACCOMPLISHED THE FOLLOWING

- Organize a platform to support 24K microservices and more than 7K engineers
- Migrate 5K databases from Apache Cassandra to DynamoDB without requesting anyone in your product organization to change a single line of code or roadmap
- Migrate your relational databases to DynamoDB for better performance and scalability while keeping legacy databases working for backward compatibility
- Build an ecosystem of platform services relying on the power of DynamoDB Streams
- Use the flexibility of DynamoDB to build a platform multiservice strategy

aws

© 2021, Amazon Web Services, Inc. or its affiliates. All rights reserved.



Mercado Libre (MELI)

Since 1999

Special thanks to Pablo Pileci, Pablo Ferraris, Ignacio Alvarez, Federico Codino

aws

© 2021, Amazon Web Services, Inc. or its affiliates. All rights reserved.

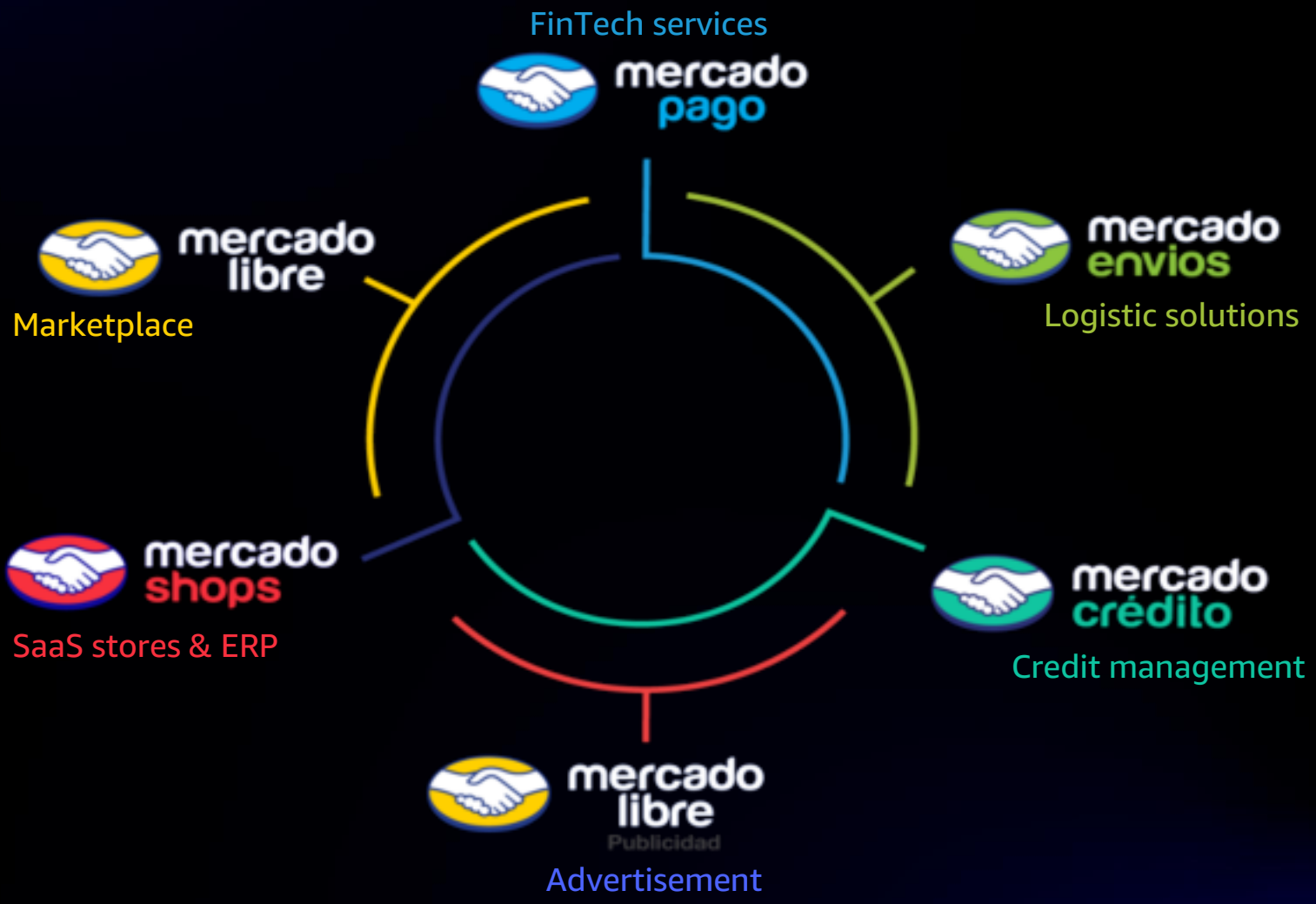
Largest ecommerce, FinTech & logistics company in LATAM



- 76M active users
- 18 countries
- 17,5B $ TPV Q2 2021
- 245M items sold Q2 2021
- 50% within 24 hours

© 2021, Amazon Web Services, Inc. or its affiliates. All rights reserved.

Powered by its business ecosystem



© 2021, Amazon Web Services, Inc. or its affiliates. All rights reserved.

Tech scale at MELI

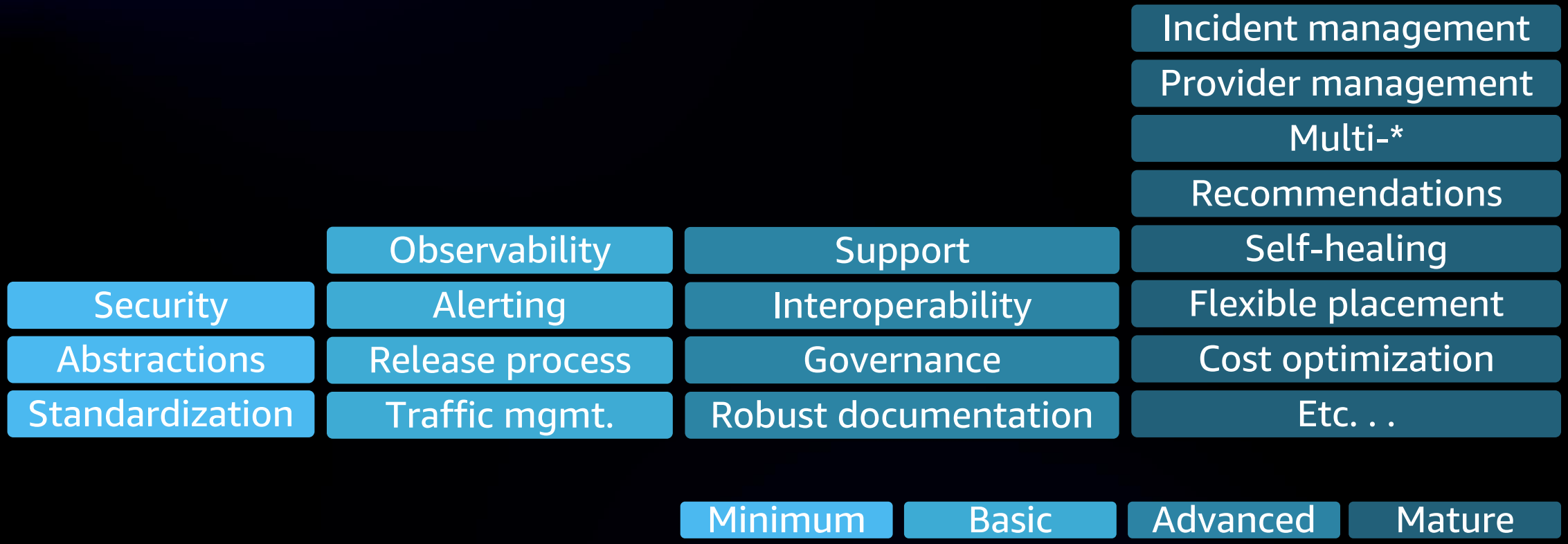
Developed & supported by

7K engineers

700M	RPM
100K	Instances
56K	Data services
24K	Microservices
17K	Databases
10K	Daily deploys
4	Clouds
1	On-premises DC

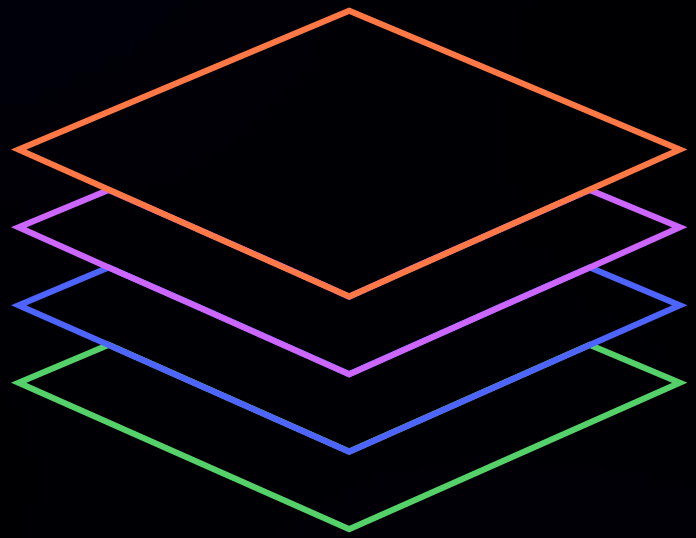
© 2021, Amazon Web Services, Inc. or its affiliates. All rights reserved.

To manage it we need . . .



. . . and more things; we call them “capabilities”

© 2021, Amazon Web Services, Inc. or its affiliates. All rights reserved.

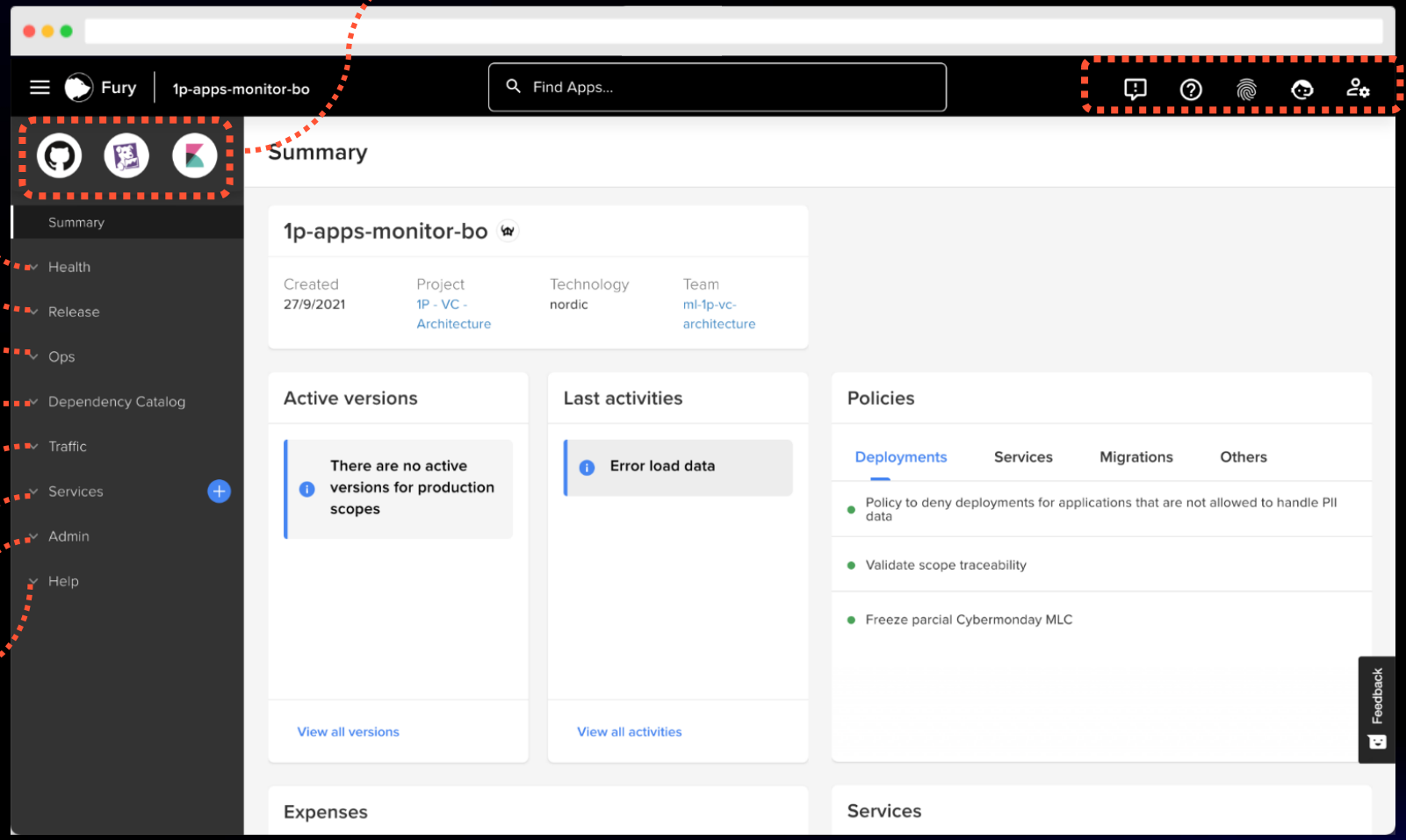


Standardization
Abstractions
Security
Etc.

PLATFORM

© 2021, Amazon Web Services, Inc. or its affiliates. All rights reserved.

Introducing Fury



© 2021, Amazon Web Services, Inc. or its affiliates. All rights reserved.

Introducing Fury

From 0 to production – Only 4 steps

Developer action



Select language & framework → **Create scope** → **Open pull request** → **Deploy version**

Fury autoresolves



Repo	Infra	Quality checks	Deploy strategies
Pipeline	Traffic mgmt.	Build & test	Deploy metrics
Monitoring	Alerting & logs	Create version	Deploy logs
Cost mgmt.	Optimizing	Security check	Centralized audit
Scaffolding	Autoscaling	Linting & metrics	Automatic rollback

aws

© 2021, Amazon Web Services, Inc. or its affiliates. All rights reserved.

Introducing Fury

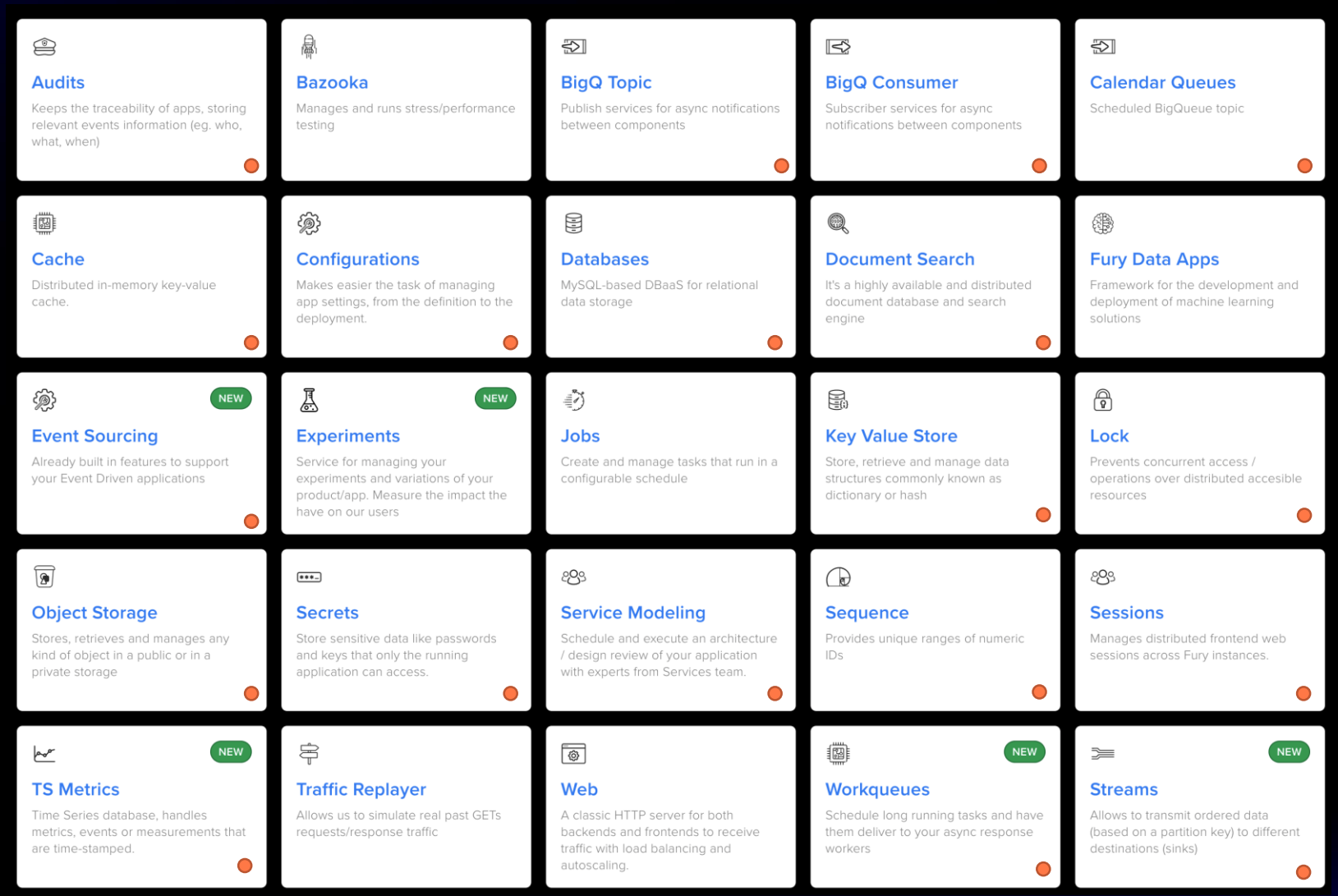
From 0 to production



- No tickets
- No wikis
- No permissions blockers
- No scripts
- No know-it-all person

© 2021, Amazon Web Services, Inc. or its affiliates. All rights reserved.

Fury Core Services

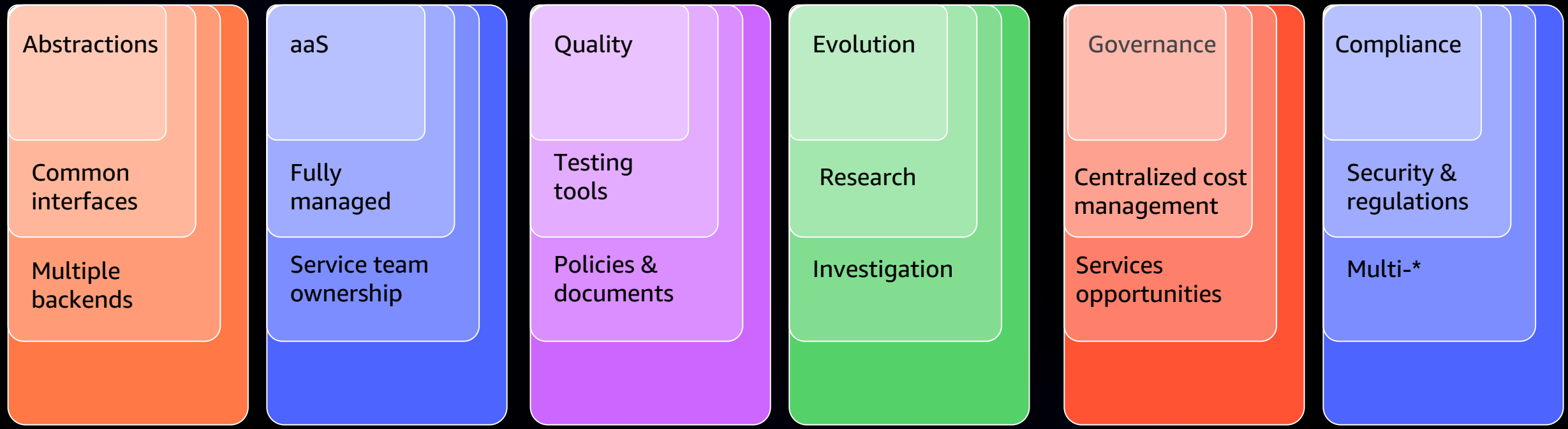


Actual Fury screenshot

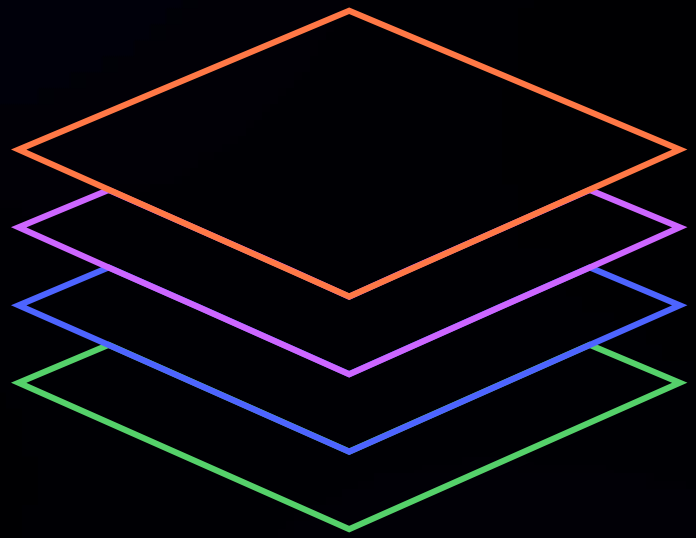
- Managed by the Core Services team

© 2021, Amazon Web Services, Inc. or its affiliates. All rights reserved.

Introducing Fury Core Services



© 2021, Amazon Web Services, Inc. or its affiliates. All rights reserved.



Common interfaces

Quality tools

Abstractions

Etc.

SERVICES
CAPABILITIES

© 2021, Amazon Web Services, Inc. or its affiliates. All rights reserved.

Introducing Fury Services

From 0 to production – Only 3 steps

Developer action



Select service type → **Select criticality & metadata** → **Create service**

Fury autoresolves



- Placement
- HA specifics
- Security
- Integrations
- Auth
- Snippets
- Alerting
- Monitoring
- Autoscaling
- Opportunities

aws

© 2021, Amazon Web Services, Inc. or its affiliates. All rights reserved.

Why not provide direct access to cloud/self-managed services?



Biased small-scale decisions

"I like"
"I have experience with"



Hard to do organization-wide changes

"I know efficiency & tech debt is important, but I have this other business priority also"



Complex setup & maintenance

"Oops didn't know I had to configure that! I spend most of my time coding the app that uses this"



Improvable troubleshooting

"I don't know what's going on; who can give us support on this outdated version of [insert service]?"



Slows down internal rotations

"Oh so you use Kafka and not Amazon Kinesis here"



Duplicated efforts

"We had that problem before, but we solved it with 1 topic, 1MySQL, 1 cache, and 2 buckets… funny you only used 1 stream and 1 MongoDB"

aws

© 2021, Amazon Web Services, Inc. or its affiliates. All rights reserved.

Services rationale



DX

Rapid growth
Traffic, storage, and engineering growth is exponential – change teams without a huge technical cognitive load

Global changes
Ability to implement company-wide changes without moving the whole organization – minimize vendor lock-in

Focus core decisions
Unify core technical decisions and evolution of core components – avoid implementation drifting

Different businesses
Require different technologies & patterns at different moments

Dynamic regulations
Ability to adhere to new regulations in a simple way

© 2021, Amazon Web Services, Inc. or its affiliates. All rights reserved.

Off Fury



Sandbox

Experimentation

Expert teams

Specific domains that need services not yet available @ Fury

© 2021, Amazon Web Services, Inc. or its affiliates. All rights reserved.

Mercado Libre's journey

IN THIS SESSION, YOU WILL LEARN HOW WE ACCOMPLISHED THE FOLLOWING

- [x] Organize a platform to support 24K microservices and more than 7K engineers
- [ ] Migrate 5K databases from Apache Cassandra to DynamoDB without requesting anyone in your product organization to change a single line of code or roadmap
- [ ] Migrate your relational databases to DynamoDB for better performance and scalability while keeping legacy databases working for backward compatibility
- [ ] Build an ecosystem of platform services relying on the power of DynamoDB Streams
- [ ] Use the flexibility of DynamoDB to build a platform multiservice strategy

aws

© 2021, Amazon Web Services, Inc. or its affiliates. All rights reserved.

Story I
Global changes

© 2021, Amazon Web Services, Inc. or its affiliates. All rights reserved.



KVS
(Key-Value Store)

© 2021, Amazon Web Services, Inc. or its affiliates. All rights reserved.

What is KVS?

Product characteristics

- Stores key values (not only)
- Most used DB service in MELI
- Used for the most-critical applications & data
  - Users
  - Orders
  - Payments
  - Items
  - Shipments
- Multi-flavor
  - Database
  - Cache
  - Vault
  - Q-KVS
- It has at least 6 out-of-the box custom integrations

Technical characteristics

- Abstracted database (cloud & backend agnostic)
- API fronting-based
- Proprietary SDK
- Heavy use of CDC features

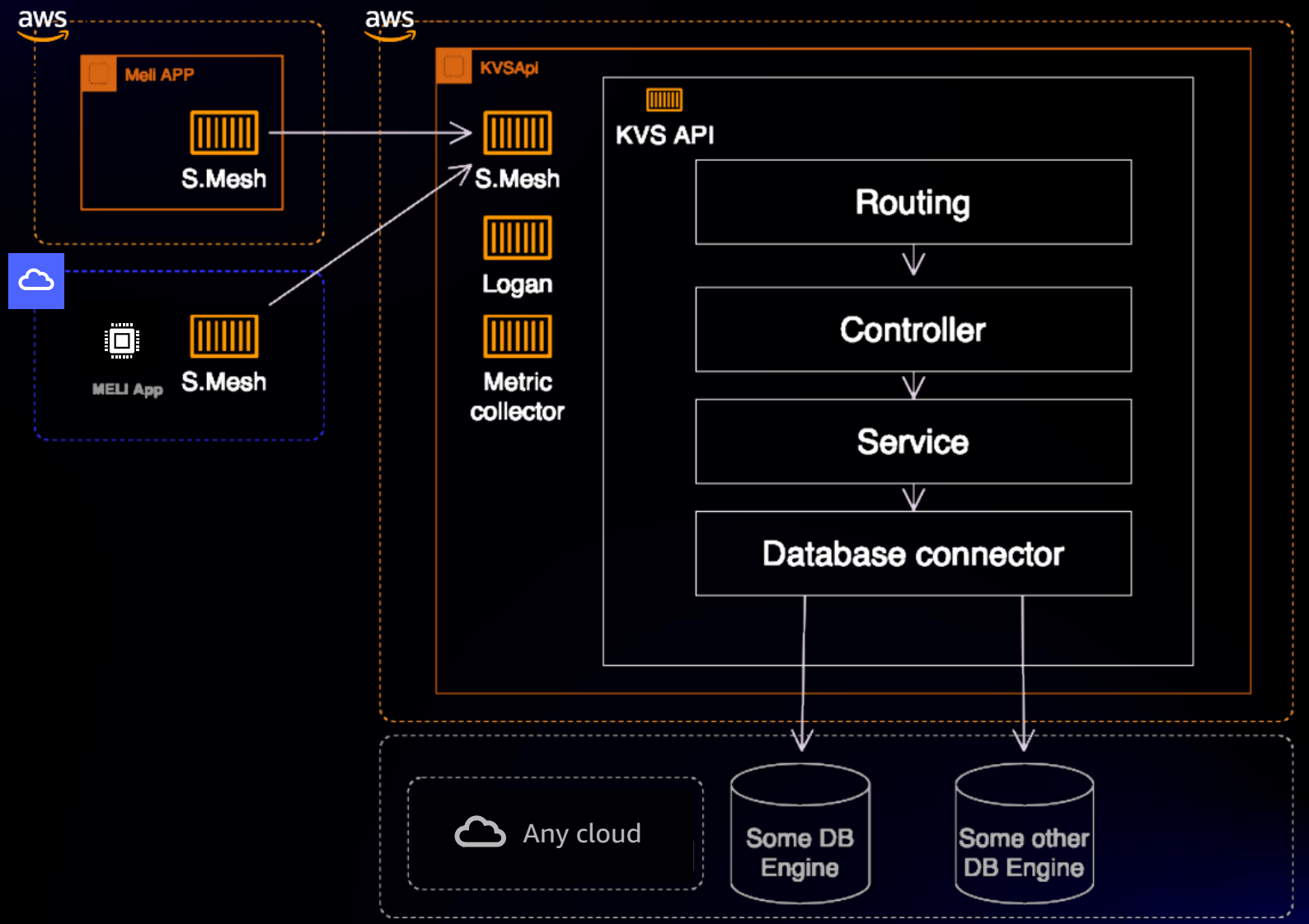
Key metrics

- 300M RPM (read) & 35M RPM (write)
- P99 latency on reads: 6 ms
- 4,000+ developers using it
- 2,400 applications using it
- 6,900 KVS currently active
- Uptime (September) → 99.998%
- Team size: 6 engineers

aws

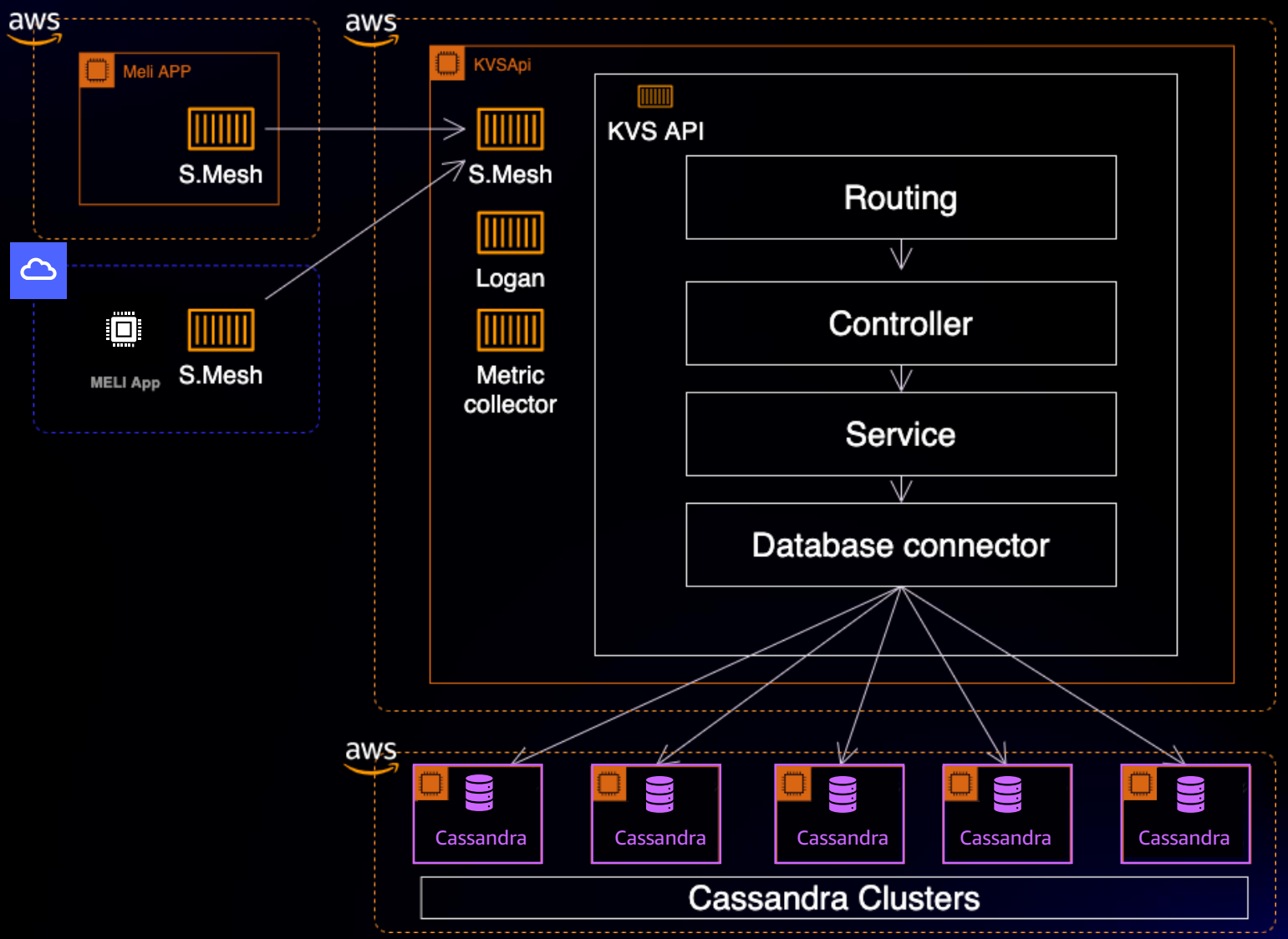
© 2021, Amazon Web Services, Inc. or its affiliates. All rights reserved.

How does it work?



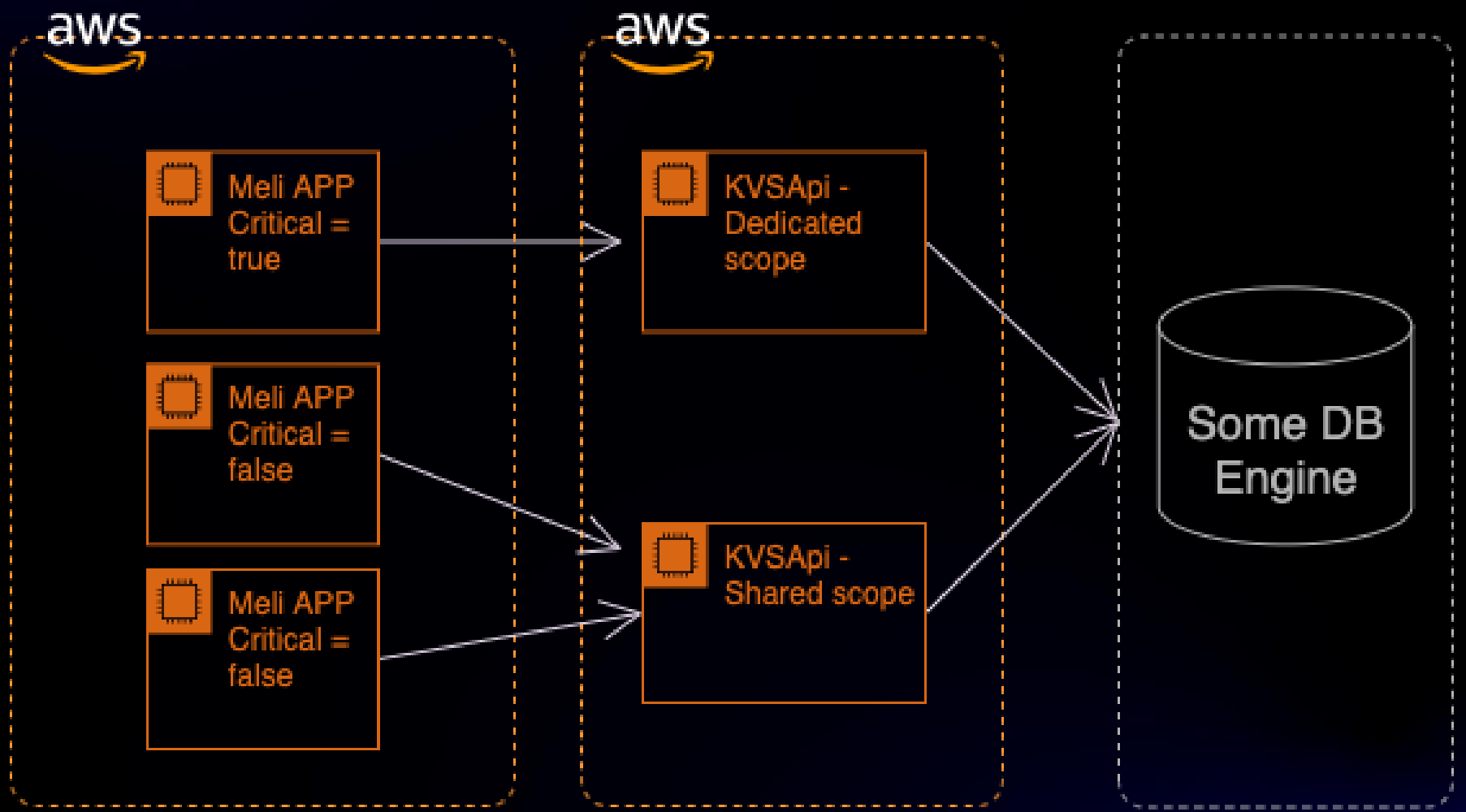
© 2021, Amazon Web Services, Inc. or its affiliates. All rights reserved.

Original KVS: Self-managed Cassandra-based



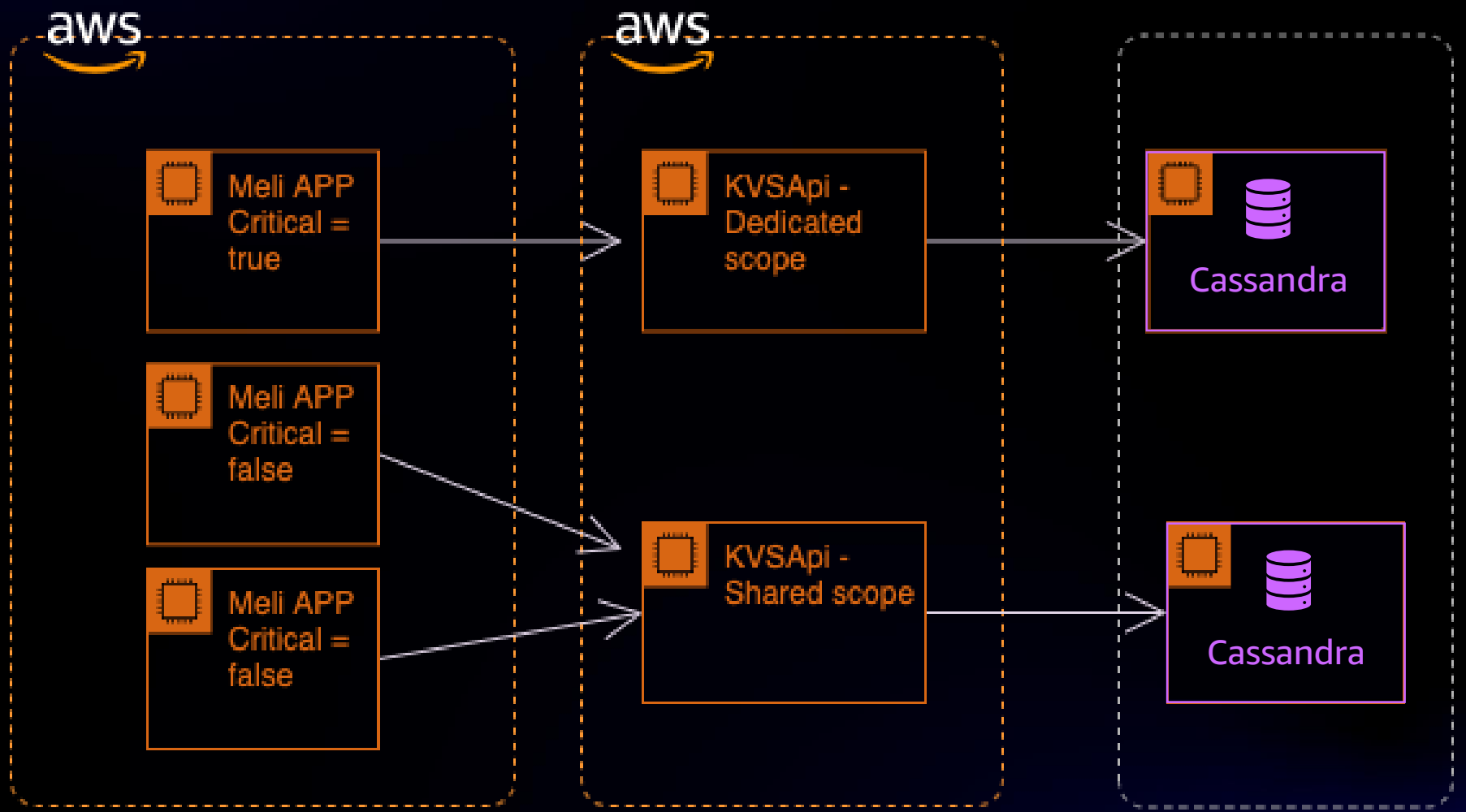
© 2021, Amazon Web Services, Inc. or its affiliates. All rights reserved.

App criticality isolation



© 2021, Amazon Web Services, Inc. or its affiliates. All rights reserved.

App criticality isolation with Cassandra



© 2021, Amazon Web Services, Inc. or its affiliates. All rights reserved.

Our previous data persistence layer

- Orders
- Items
- Users
- Payments
- Recommendations
- Shipping
- Questions
- Search
- Login
- Sessions
- Activities
- Loyalty
- Auth
- Fulfillment
- Pictures
- Others

30+ Cassandra clusters

Complexity



Isolation



aws

© 2021, Amazon Web Services, Inc. or its affiliates. All rights reserved.

Pain points



Ouch

Maintenance & upgrades
Complex, impacts performance, timeouts, nightly work

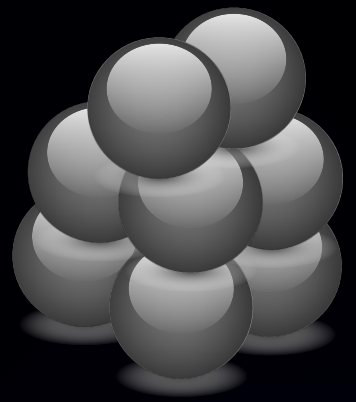
Scale & special events
Complex to scale up (quorum) & resource under utilization

Isolation
Grow in # of clusters – more complexity

Features & evolution
CDC wasn't great (TTLs) – daily backups, hotkey mgmt.

© 2021, Amazon Web Services, Inc. or its affiliates. All rights reserved.

Critical incidents (engine-related)

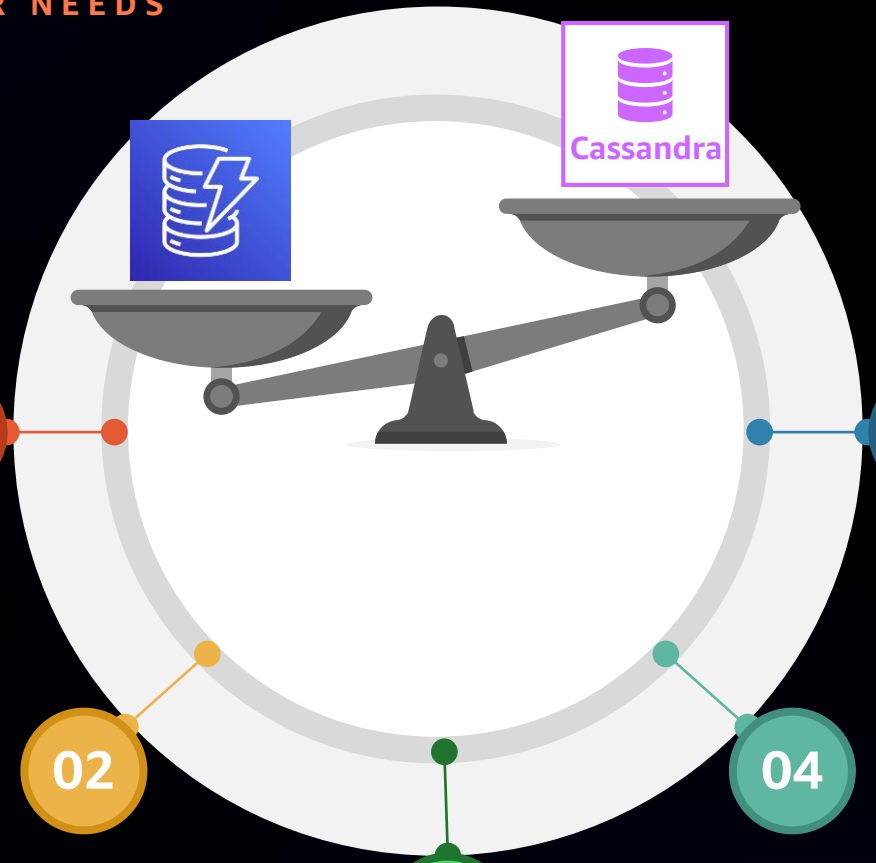


10

© 2021, Amazon Web Services, Inc. or its affiliates. All rights reserved.

Migration key points

WEIGHING FEATURES AGAINST OUR NEEDS



Complexity
01
Account scaling vs. cluster scaling

Scaling
02
Automatic with simple downscaling vs. manual with resource underutilization

CDC
03
DynamoDB Streams vs. API level and no TTL support

Multi-Region
04
Global tables one click vs. setup effort

Backups
05
PITR vs. daily backups

aws

© 2021, Amazon Web Services, Inc. or its affiliates. All rights reserved.

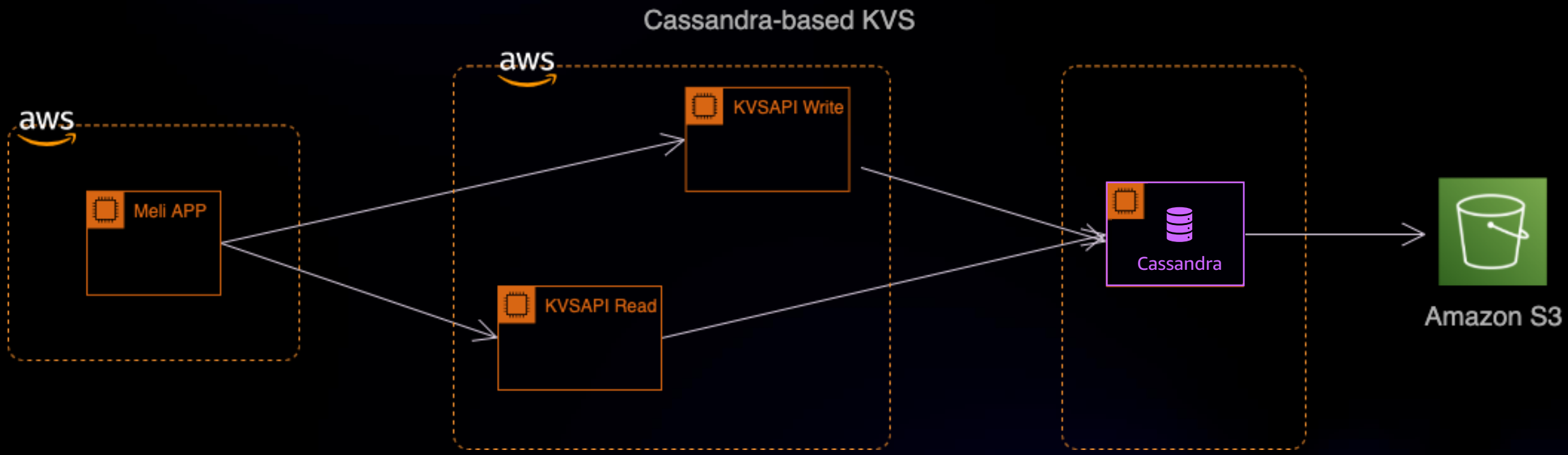


KVS
Migration process step-by-step

© 2021, Amazon Web Services, Inc. or its affiliates. All rights reserved.

Step 1 – Replicate data

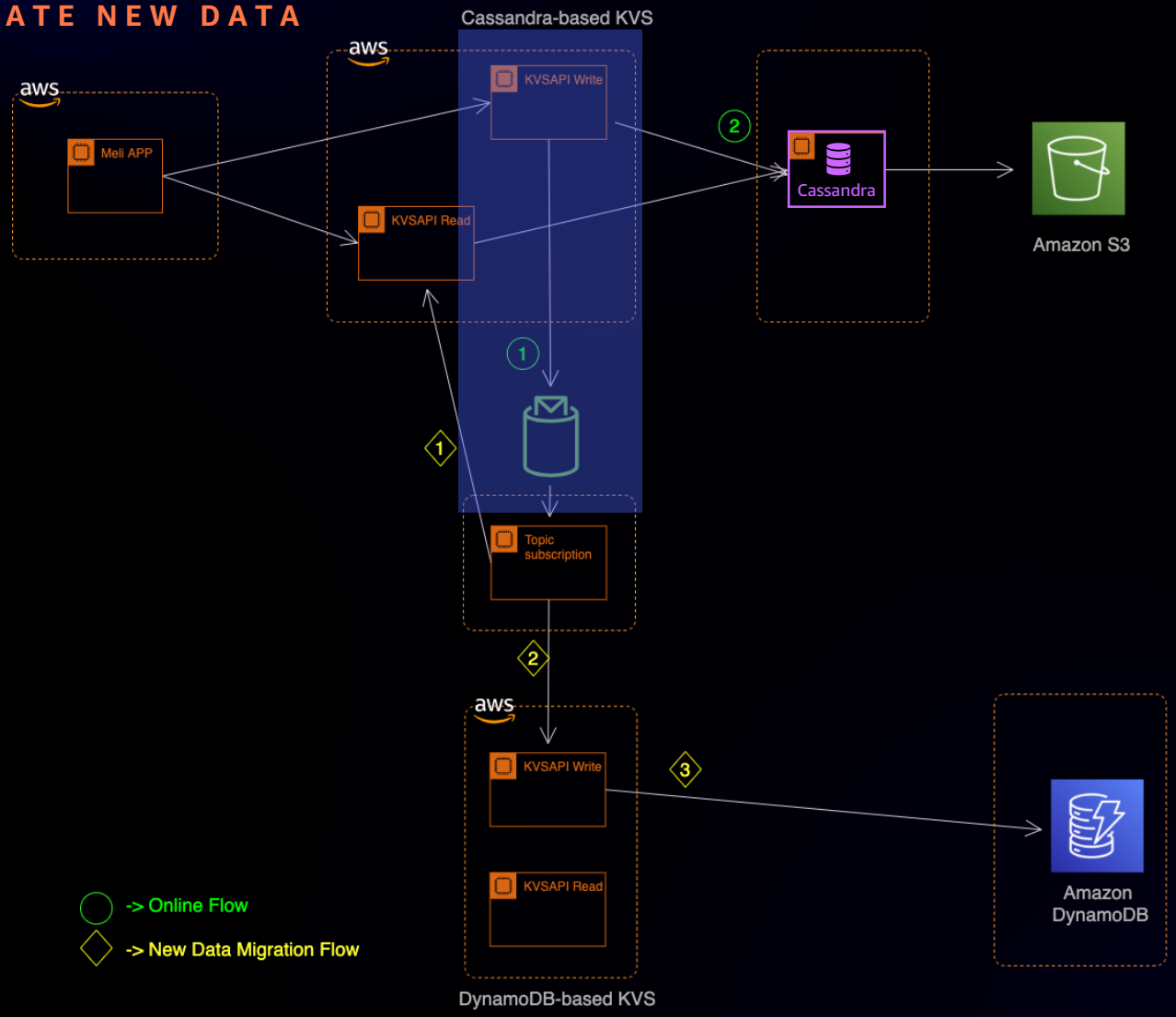
BUSINESS AS USUAL



© 2021, Amazon Web Services, Inc. or its affiliates. All rights reserved.

Step 1 – Replicate data

REPLICATE NEW DATA



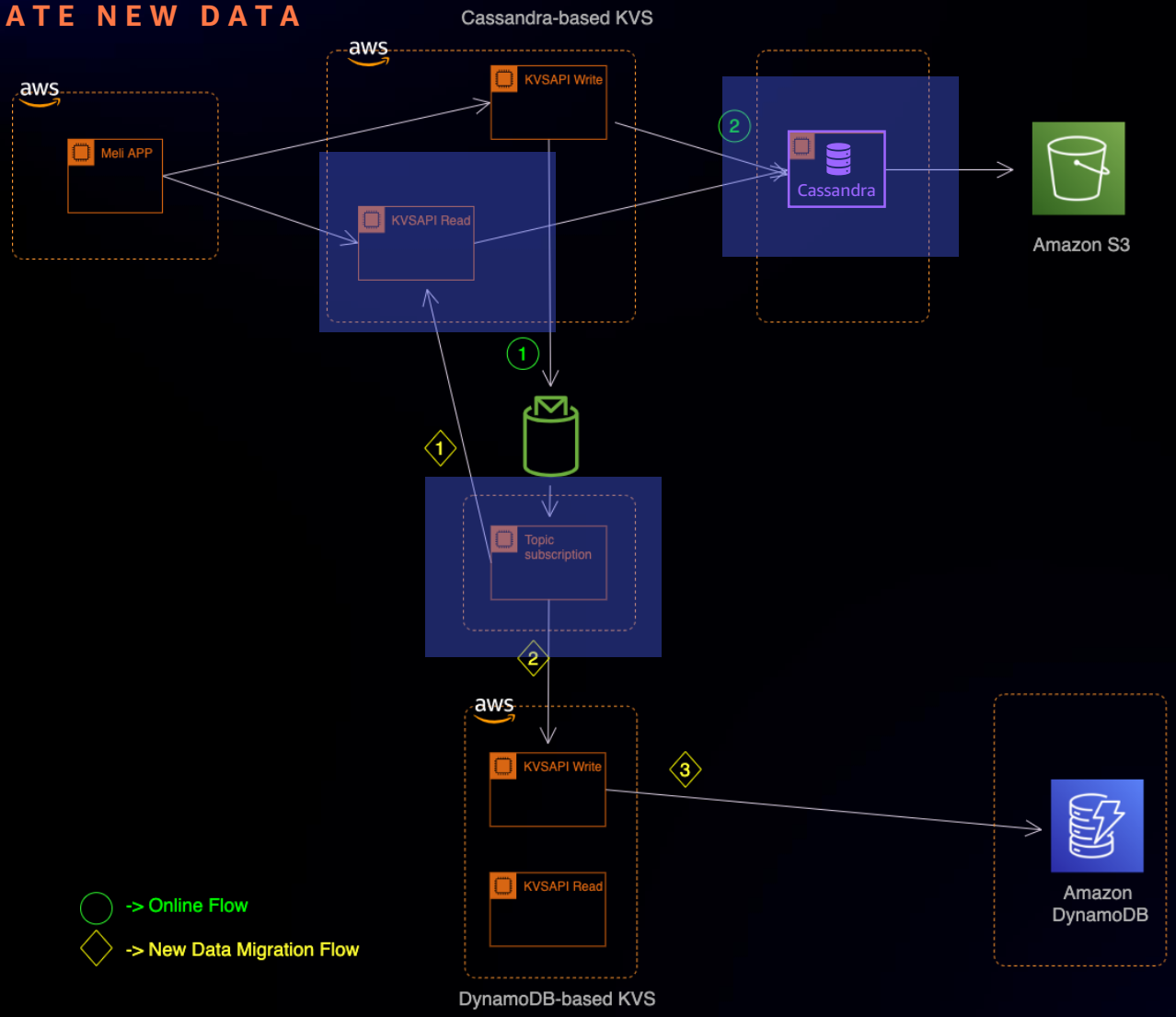
Key points

- Application kept writing and reading from Cassandra
- New synchronous step was added (write to pub/sub topic P99 5 ms)
- Was done at different rates during the day to avoid stressing the original cluster
- Always reads the current value before writing

© 2021, Amazon Web Services, Inc. or its affiliates. All rights reserved.

Step 1 – Replicate data

REPLICATE NEW DATA



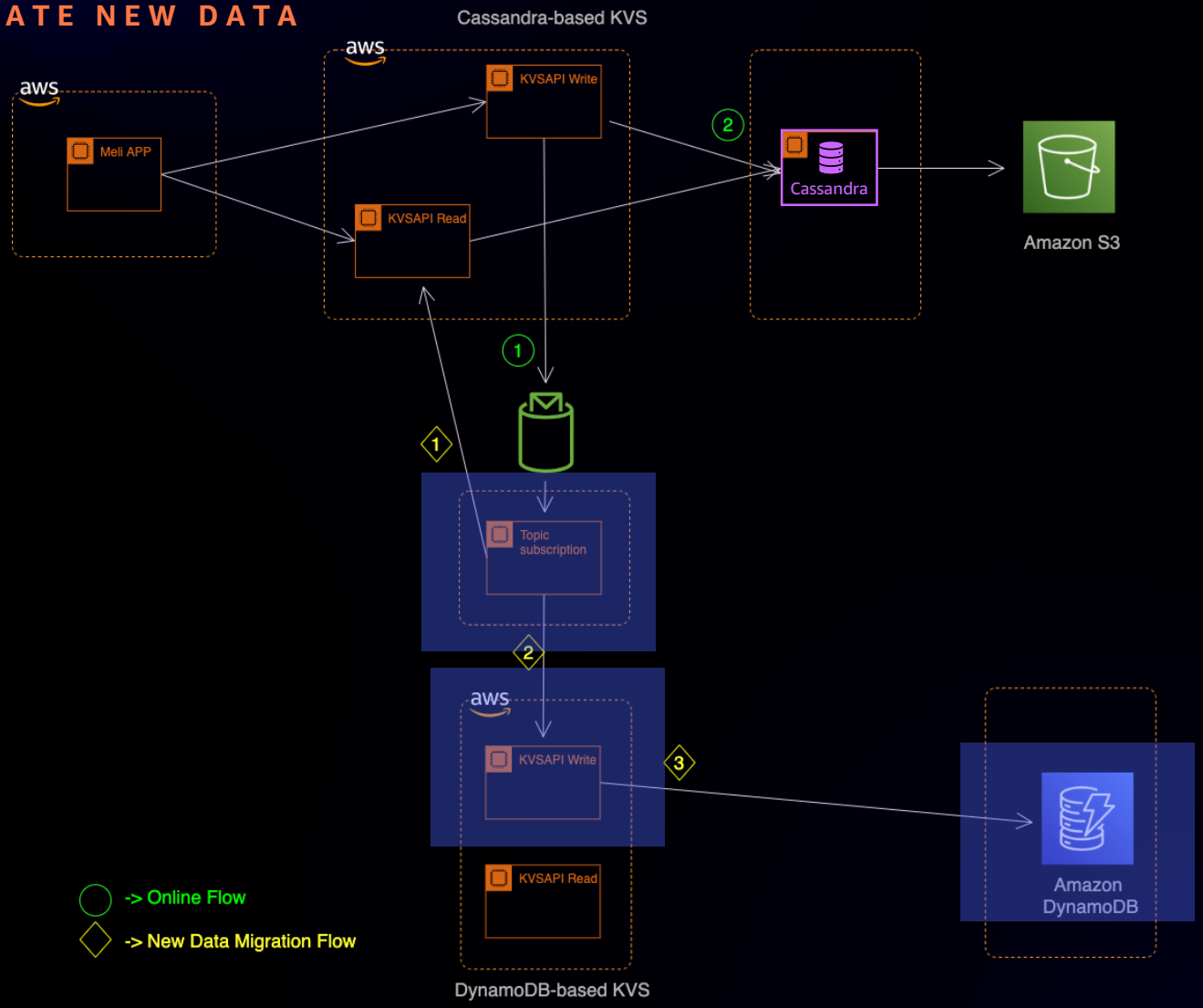
Key points

- Application kept writing and reading from Cassandra
- New synchronous step was added (write to pub/sub topic P99 5 ms)
- Was done at different rates during the day to avoid stressing the original cluster
- Always reads the current value before writing

© 2021, Amazon Web Services, Inc. or its affiliates. All rights reserved.

Step 1 – Replicate data

REPLICATE NEW DATA



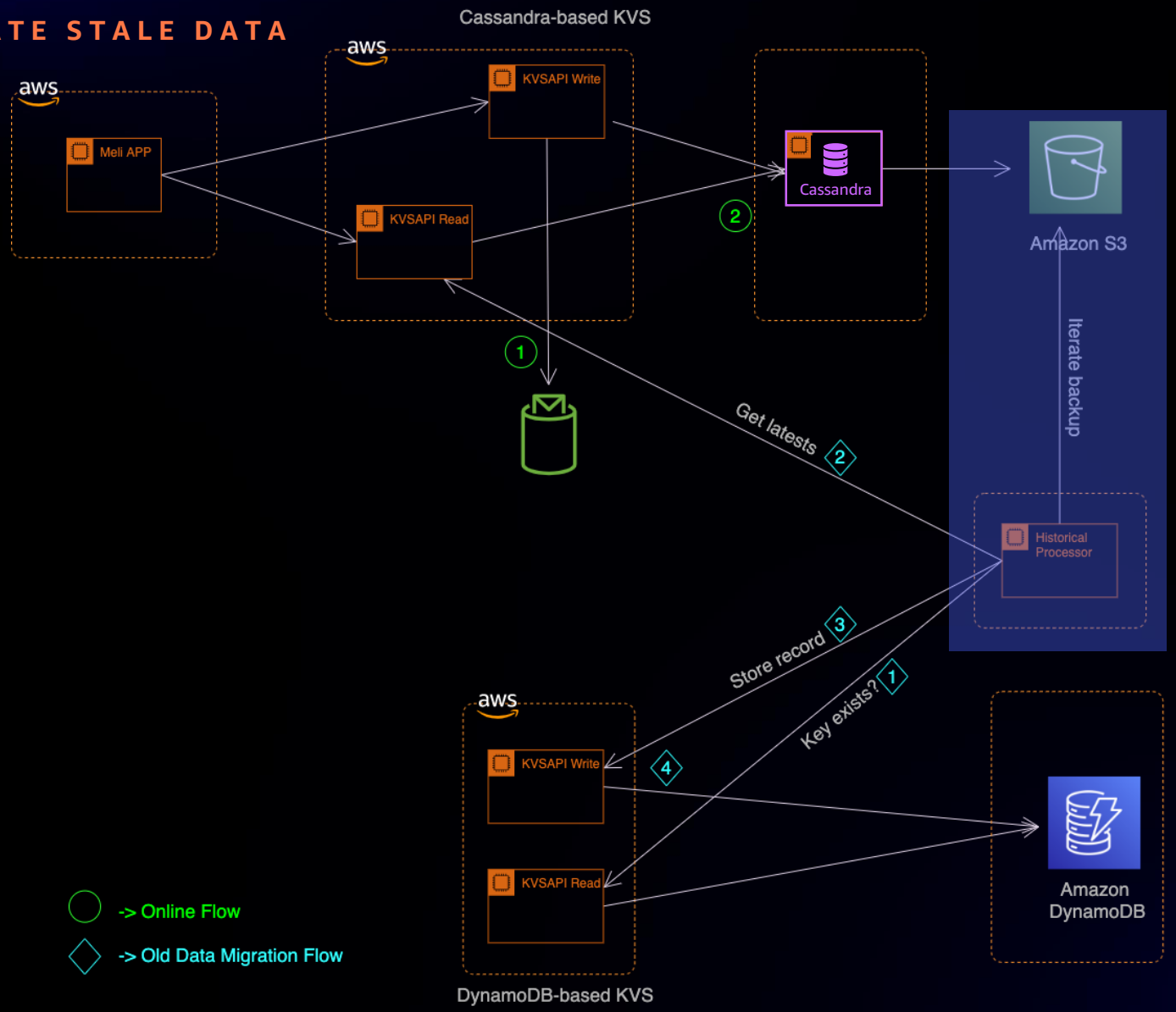
Key points

- Application kept writing and reading from Cassandra
- New synchronous step was added (write to pub/sub topic P99 5 ms)
- Was done at different rates during the day to avoid stressing the original cluster
- Always reads the current value before writing

© 2021, Amazon Web Services, Inc. or its affiliates. All rights reserved.

Step 1 – Replicate data

REPLICATE STALE DATA



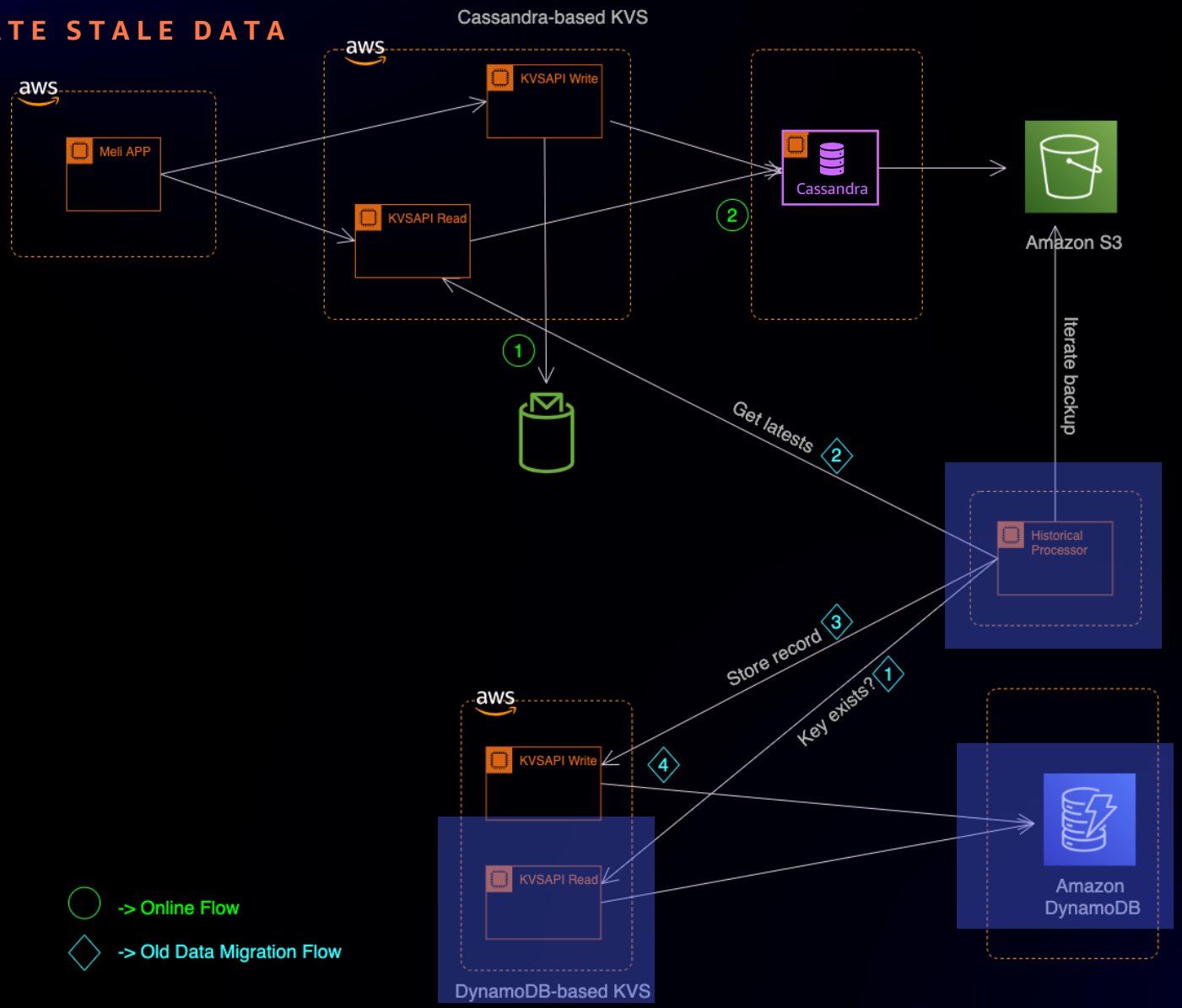
Key points

- Define a “golden” backup copy
- Backup wrote only if current key was missing (not already added by new data flow)
- Always reads the current value before writing

© 2021, Amazon Web Services, Inc. or its affiliates. All rights reserved.

Step 1 – Replicate data

REPLICATE STALE DATA



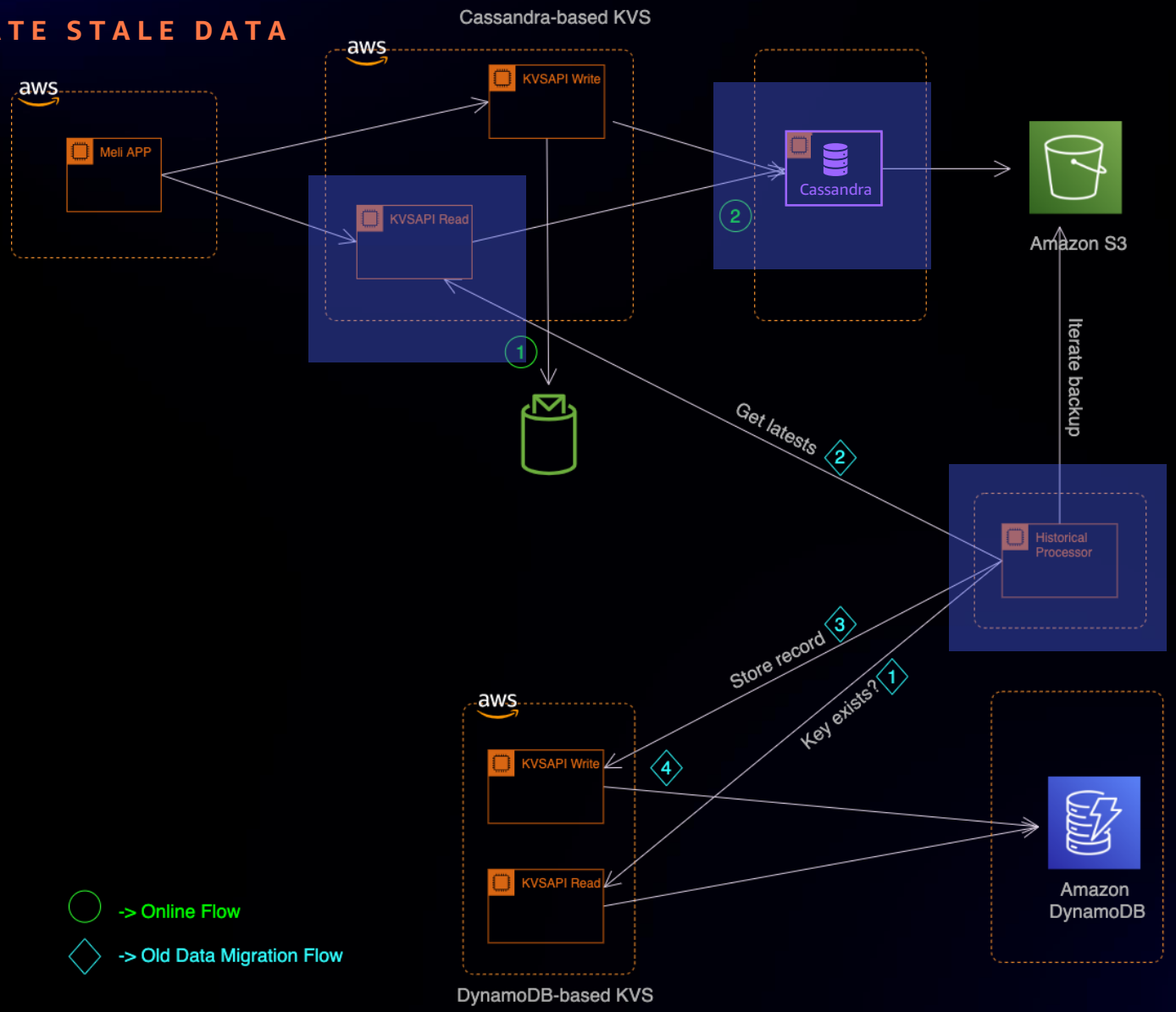
Key points

- Define a “golden” backup copy
- Backup wrote only if current key was missing (not already added by new data flow)
- Always reads the current value before writing

© 2021, Amazon Web Services, Inc. or its affiliates. All rights reserved.

Step 1 – Replicate data

REPLICATE STALE DATA



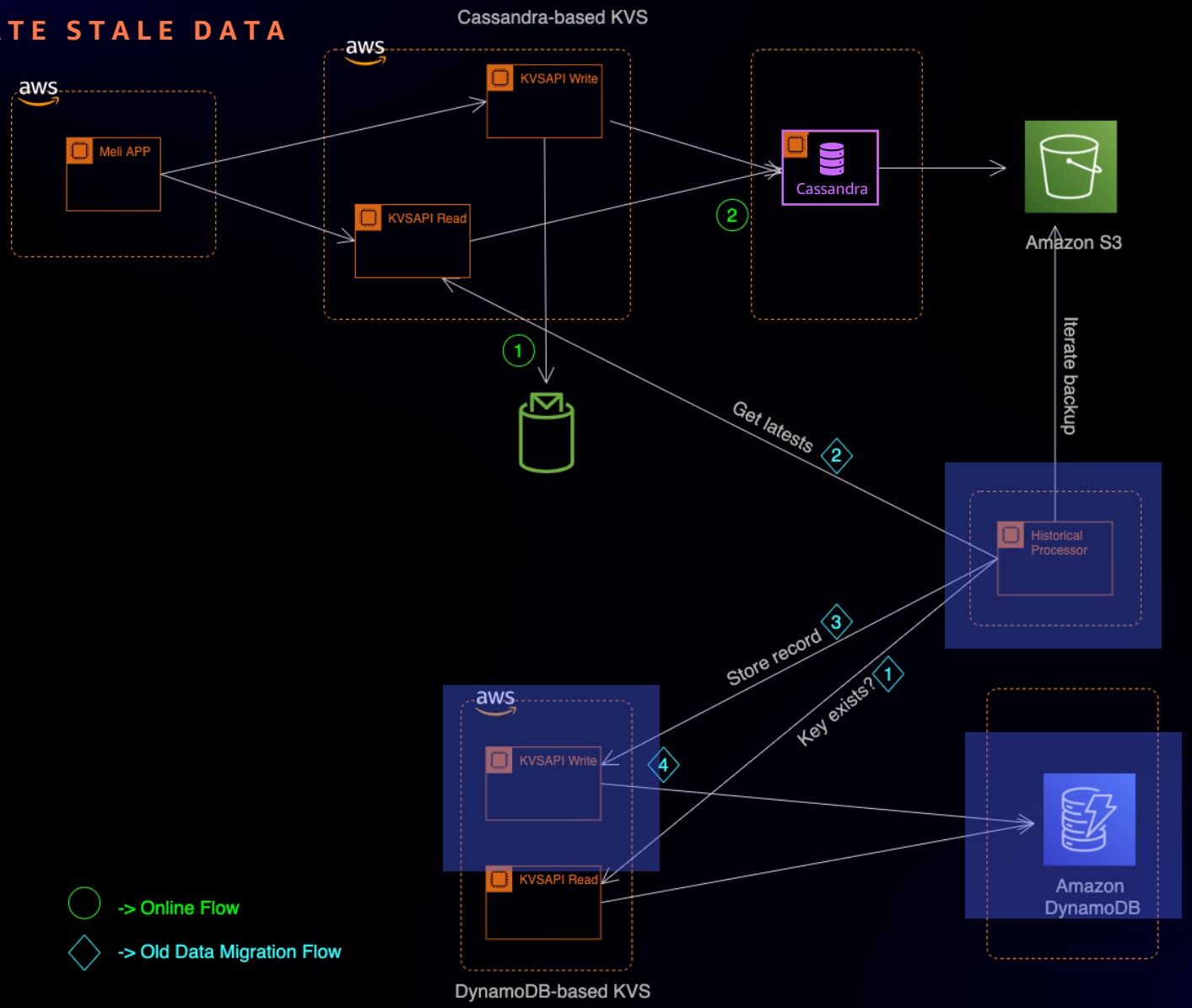
Key points

- Define a “golden” backup copy
- Backup wrote only if current key was missing (not already added by new data flow)
- Always reads the current value before writing

© 2021, Amazon Web Services, Inc. or its affiliates. All rights reserved.

Step 1 – Replicate data

REPLICATE STALE DATA

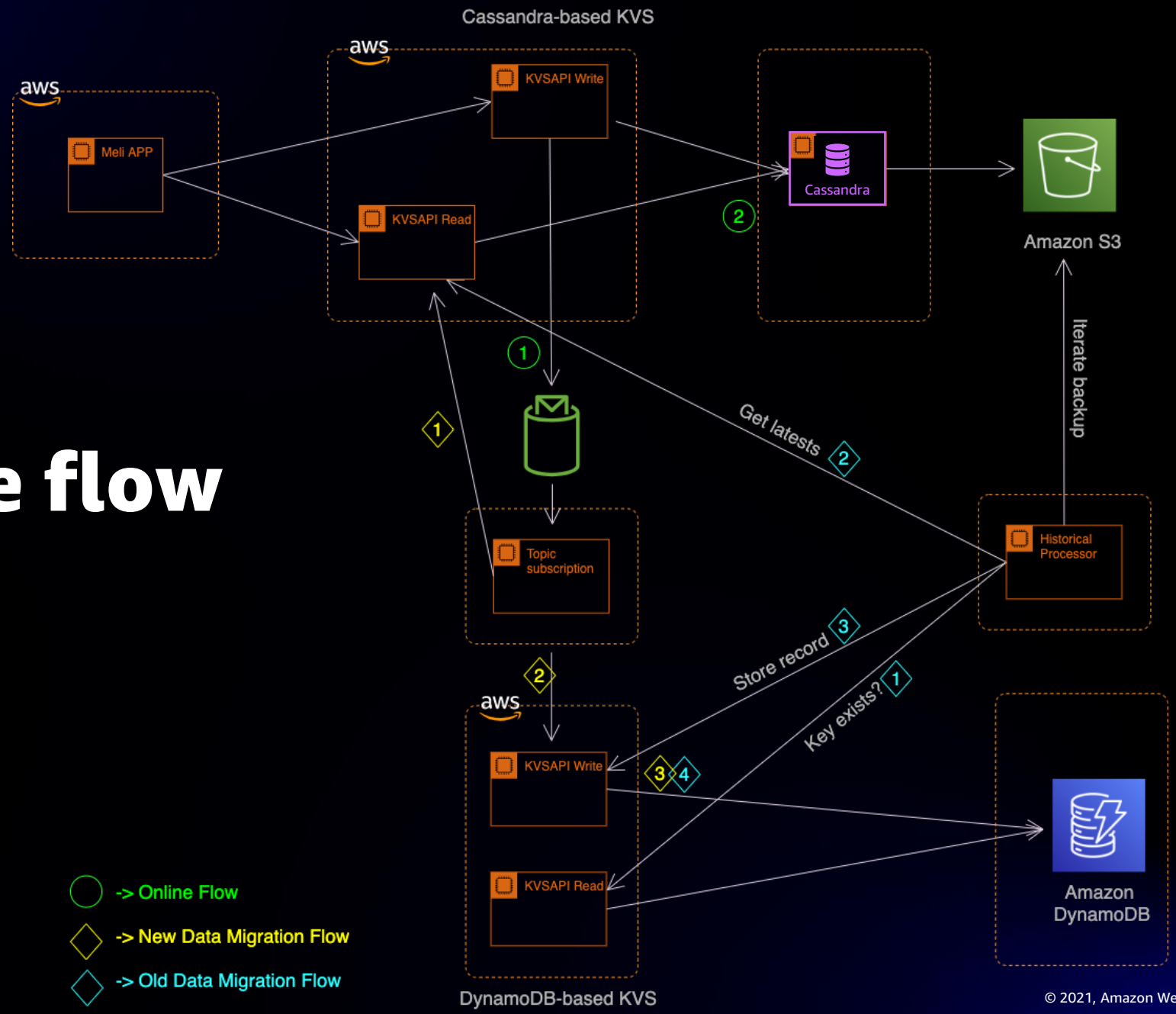


Key points

- Define a “golden” backup copy
- Backup wrote only if current key was missing (not already added by new data flow)
- Always reads the current value before writing

© 2021, Amazon Web Services, Inc. or its affiliates. All rights reserved.

Complete flow



© 2021, Amazon Web Services, Inc. or its affiliates. All rights reserved.

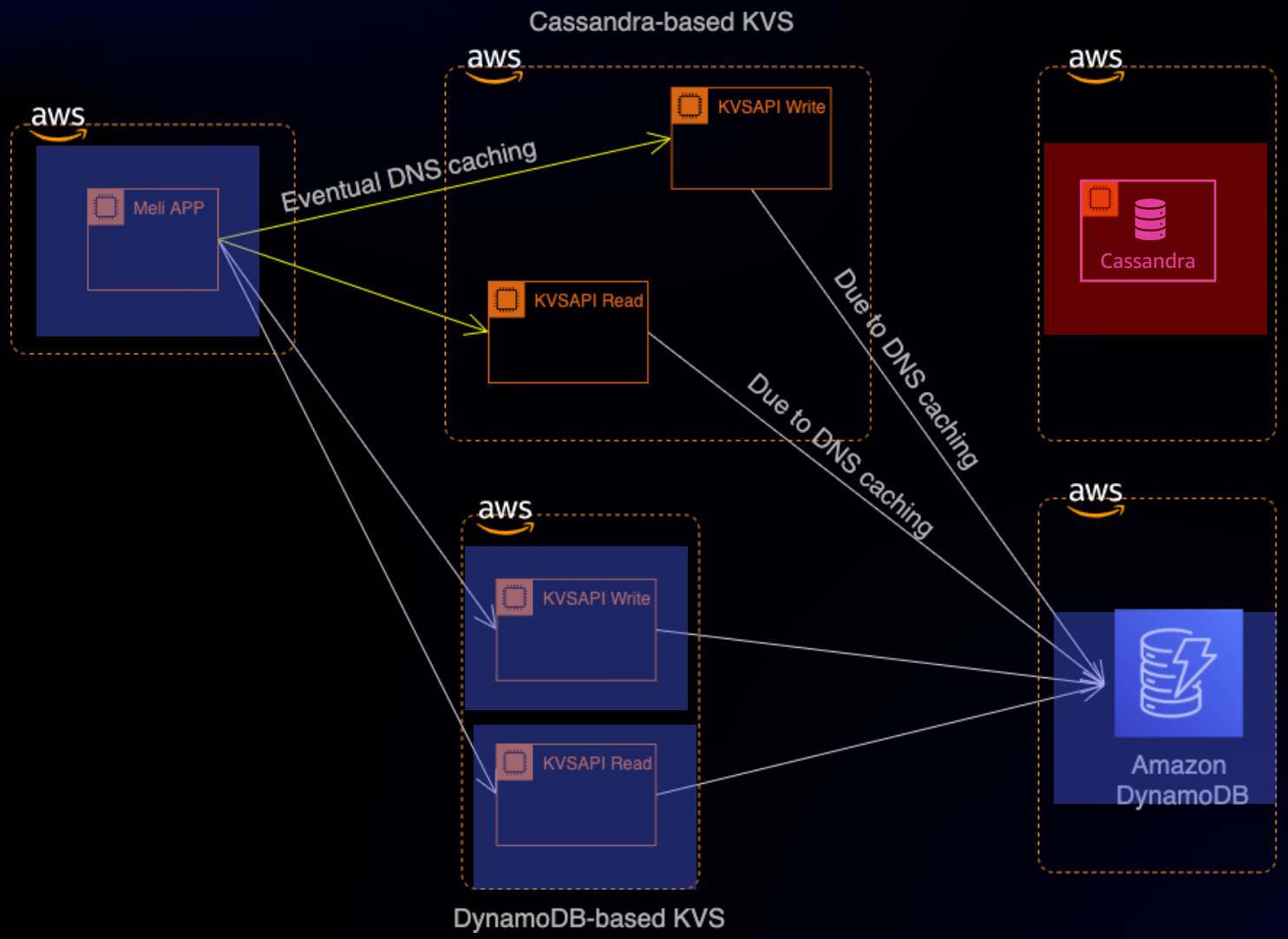
Step 2 – Communication & coordination

Step 2

- When backup processing is completed
- When pub/sub subscription is up to date with backlog
- Coordinate planned write downtime with the team owner of the database (KVS)
- Mark KVS as fully migrated in the metadata

© 2021, Amazon Web Services, Inc. or its affiliates. All rights reserved.

Step 3 – Reroute traffic

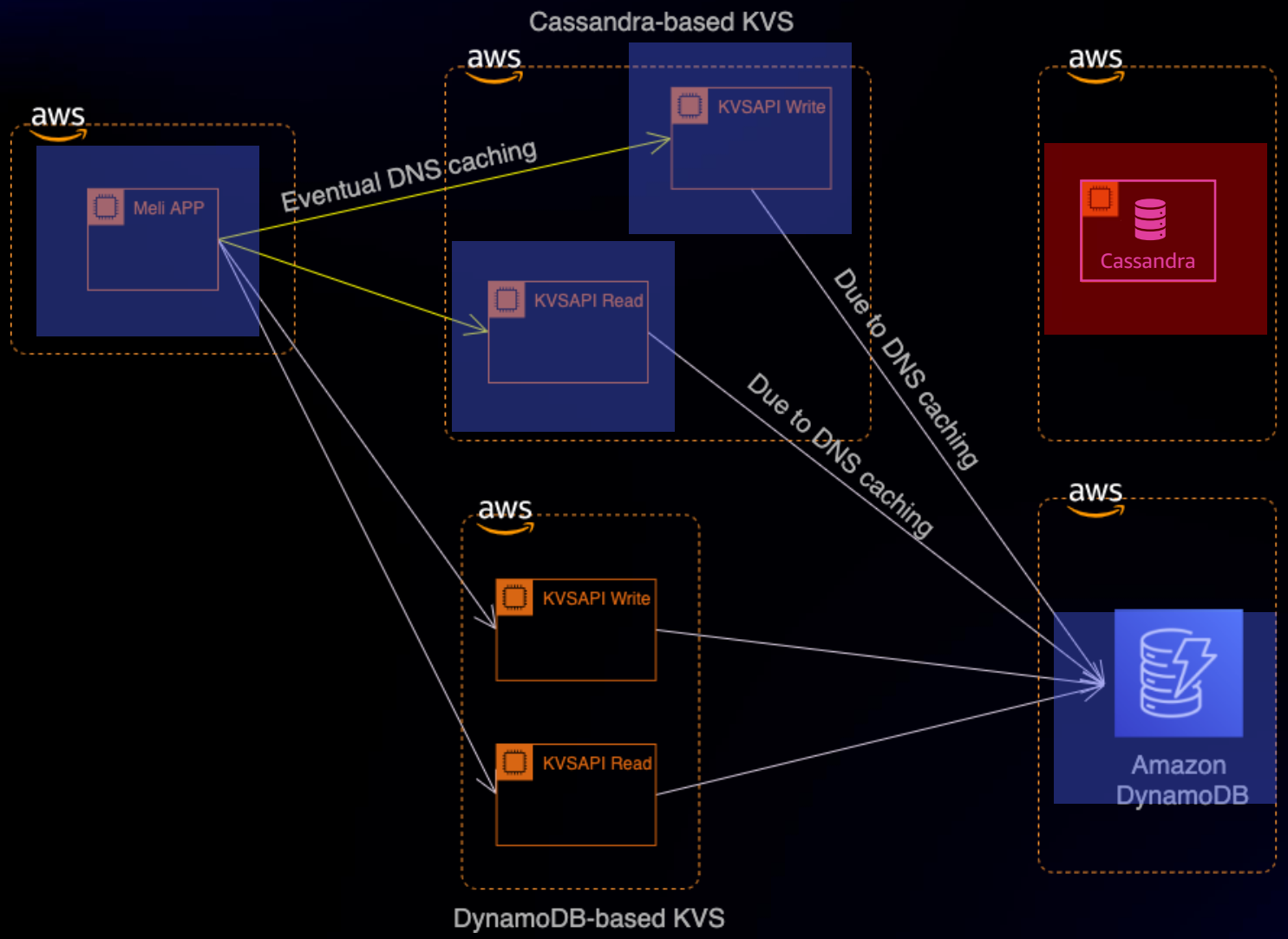


Key points

- Transparently change DNS routing for application
- Now application reads and writes from DynamoDB
- Old KVS API infrastructure points to DynamoDB to avoid errors while DNS refreshes
- Be sure; rollback is not simple

© 2021, Amazon Web Services, Inc. or its affiliates. All rights reserved.

Step 3 – Reroute traffic

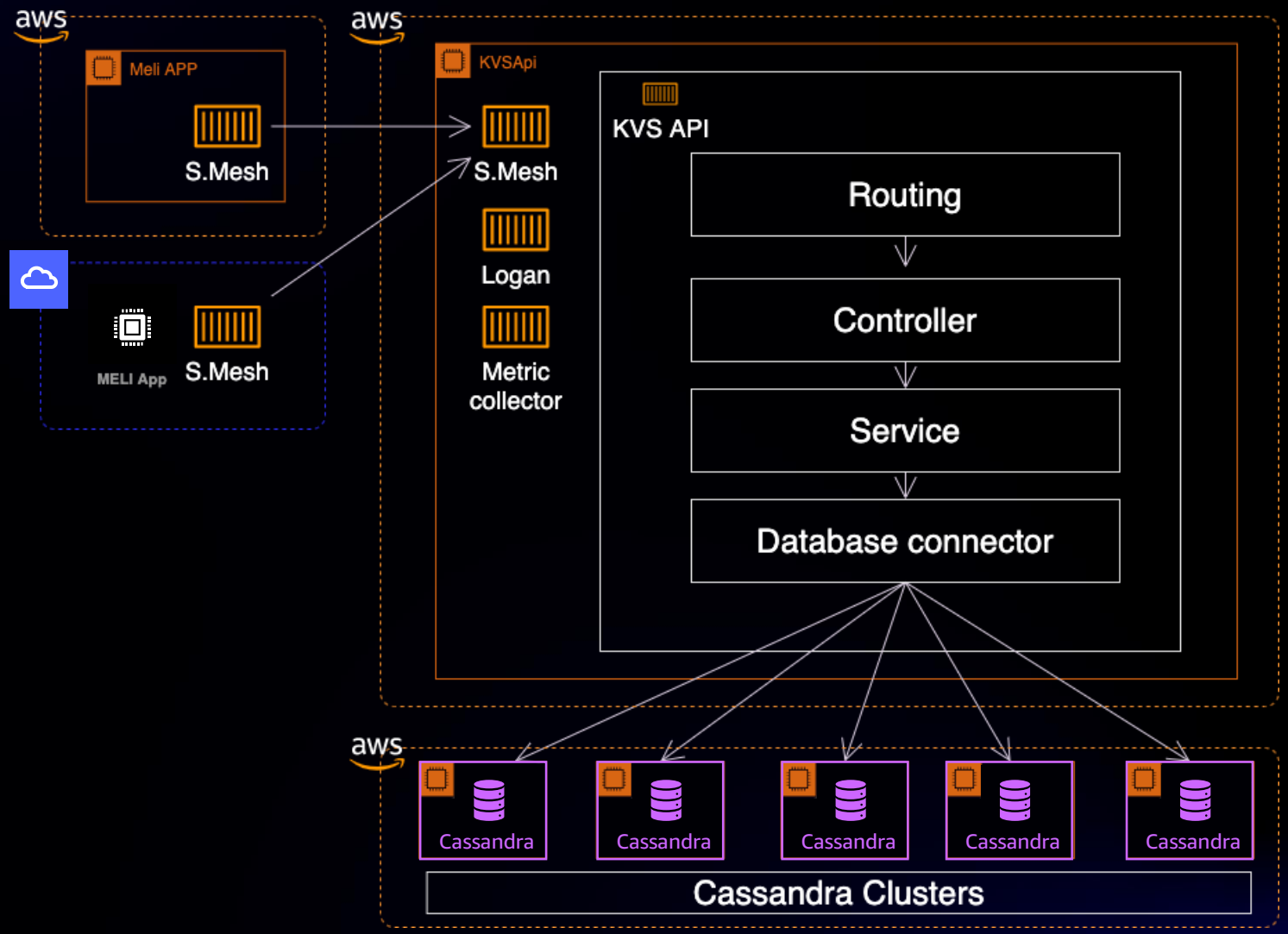


Key points

- Transparently change DNS routing for application
- Now application reads and writes from DynamoDB
- Old KVS API infrastructure points to DynamoDB to avoid errors while DNS refreshes
- Be sure; rollback is not simple

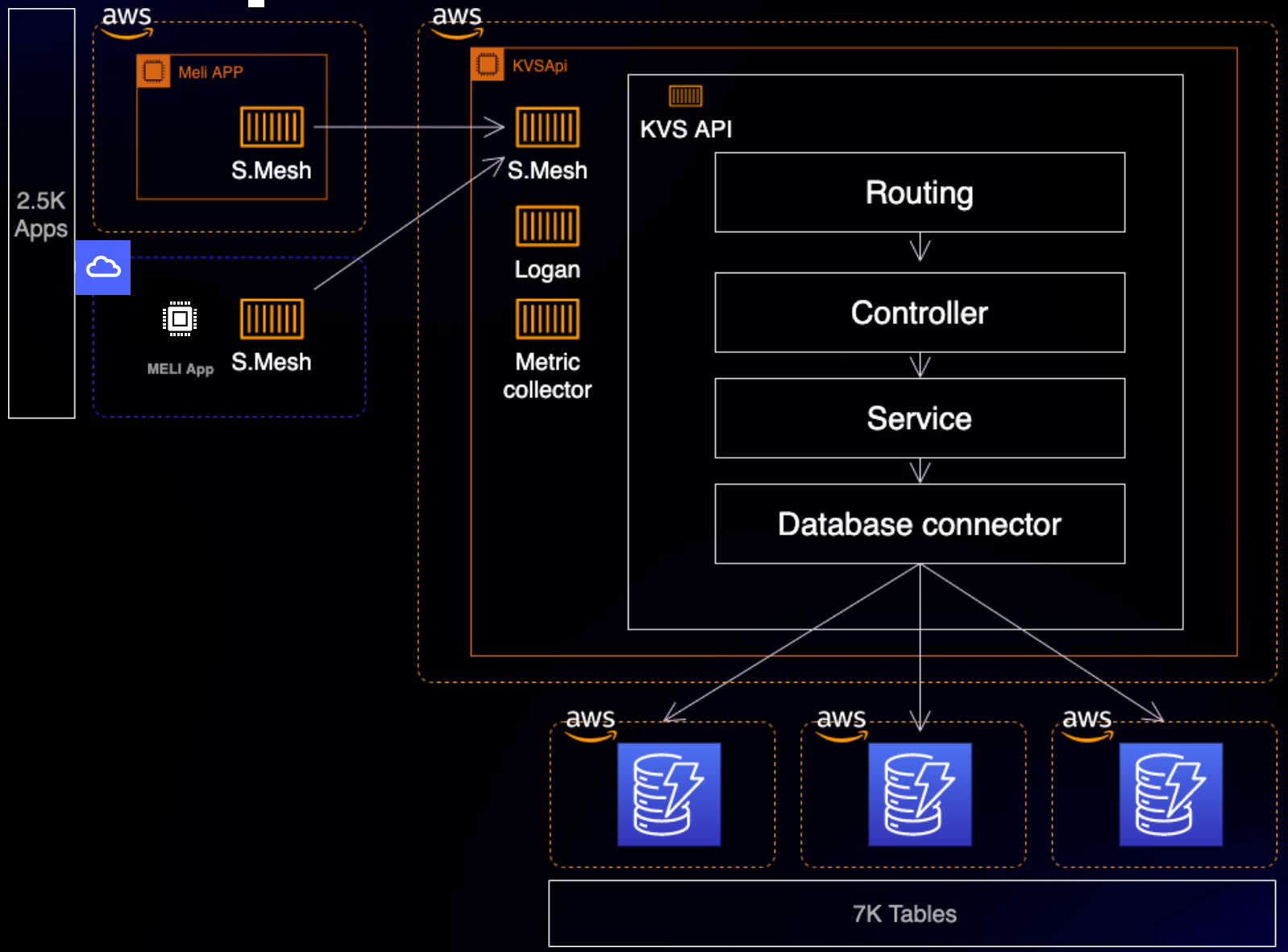
© 2021, Amazon Web Services, Inc. or its affiliates. All rights reserved.

Original KVS: Self-managed Cassandra-based



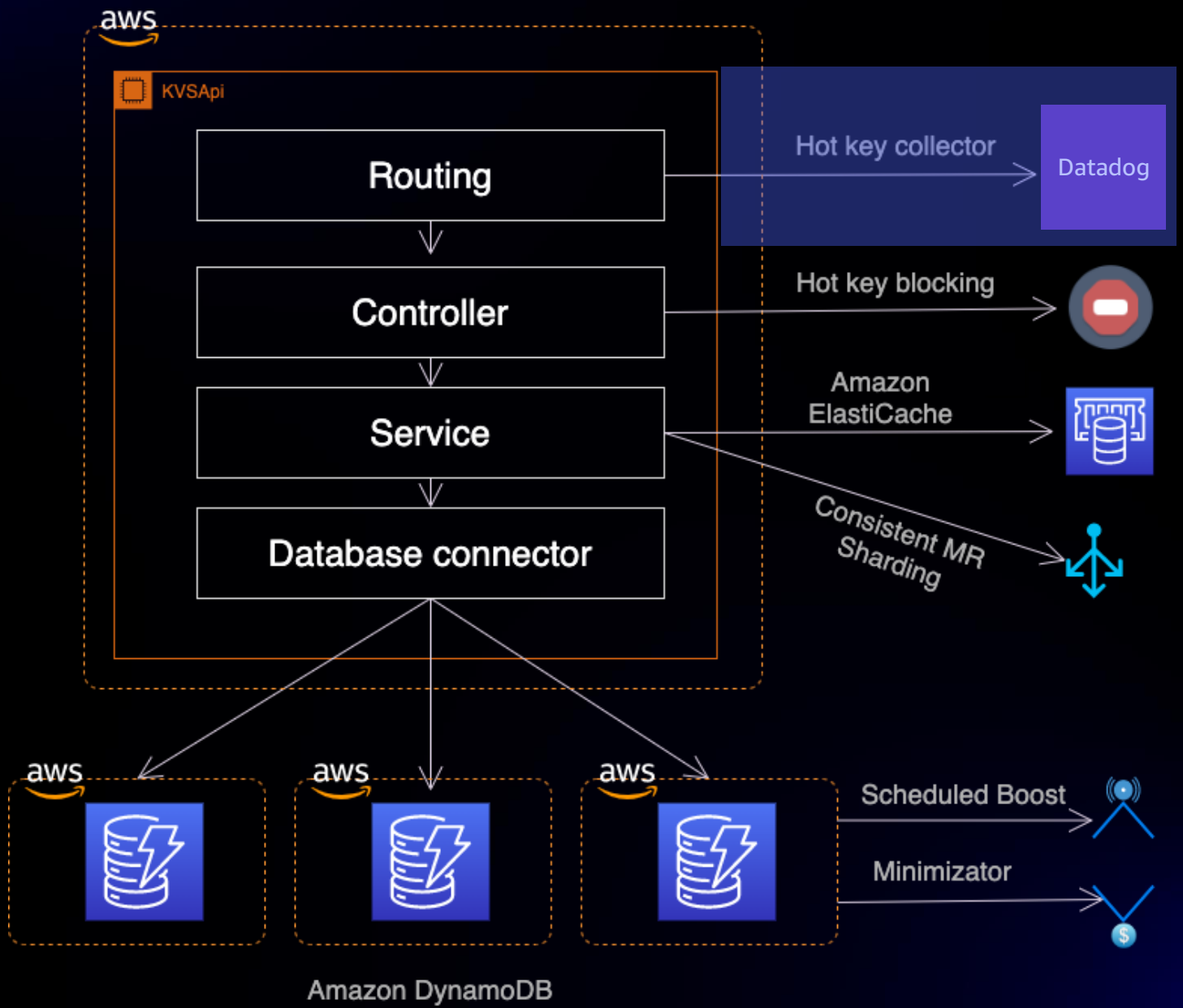
© 2021, Amazon Web Services, Inc. or its affiliates. All rights reserved.

Current implementation



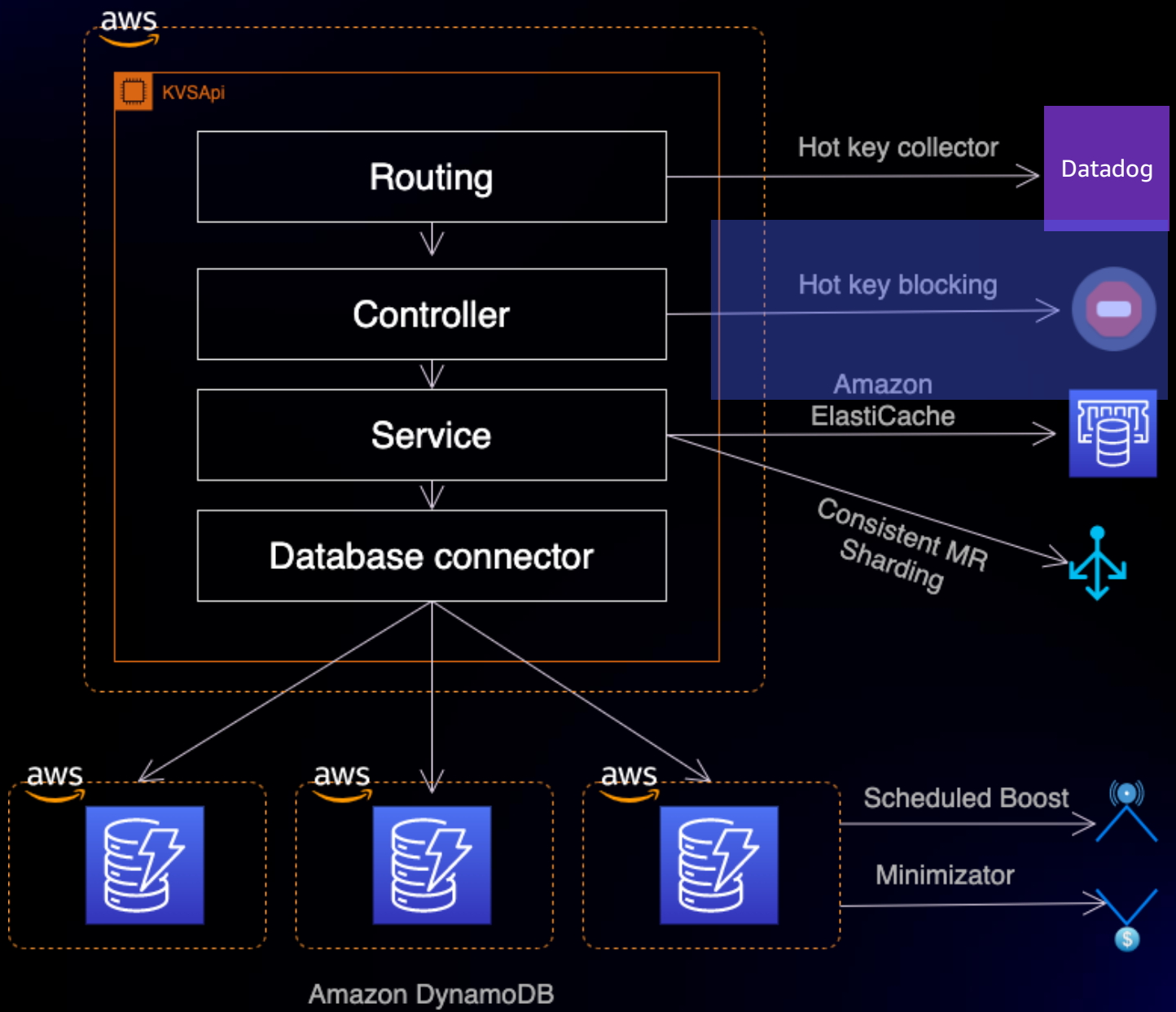
© 2021, Amazon Web Services, Inc. or its affiliates. All rights reserved.

KVS API – Add-ons



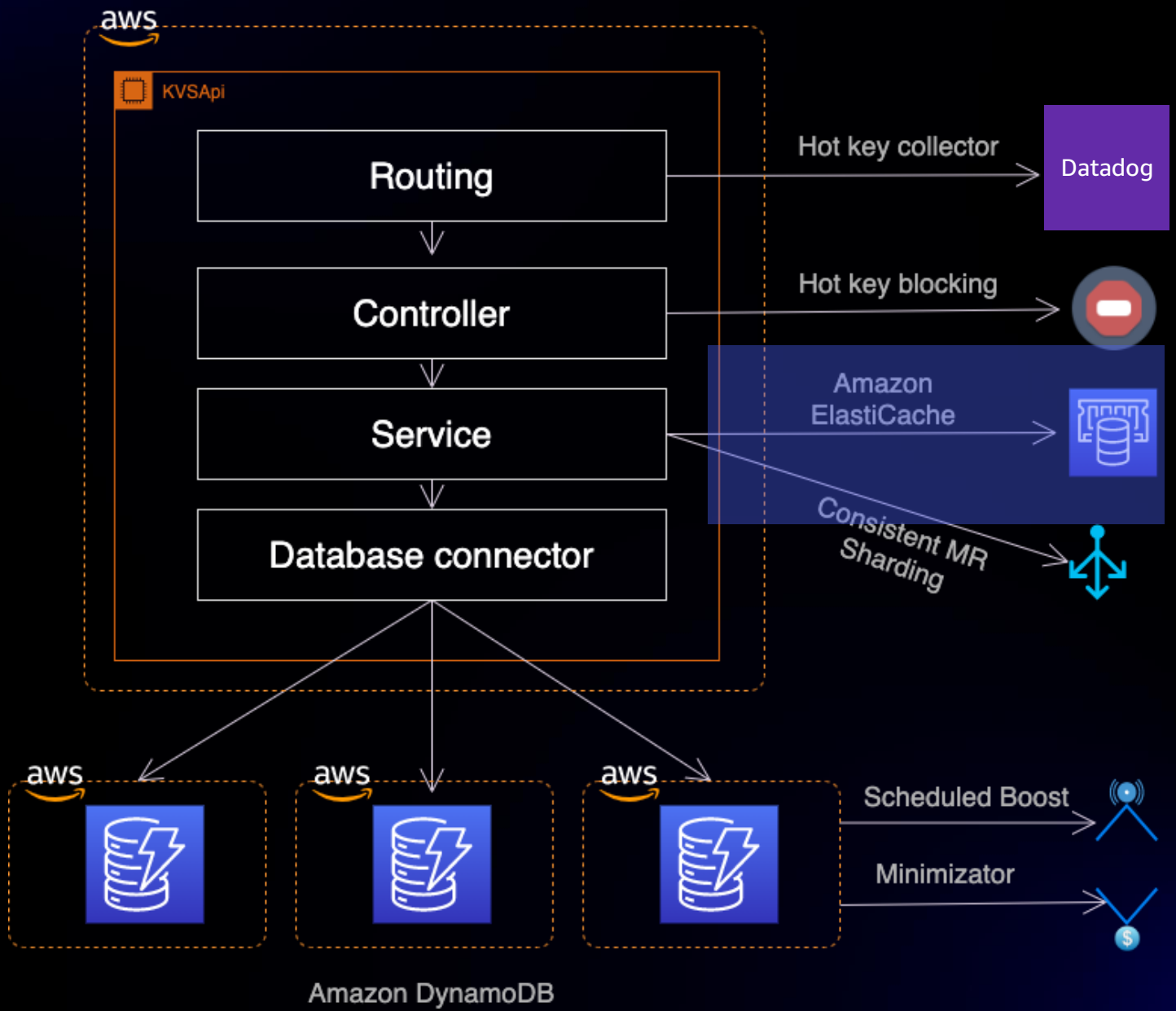
© 2021, Amazon Web Services, Inc. or its affiliates. All rights reserved.

KVS API – Add-ons



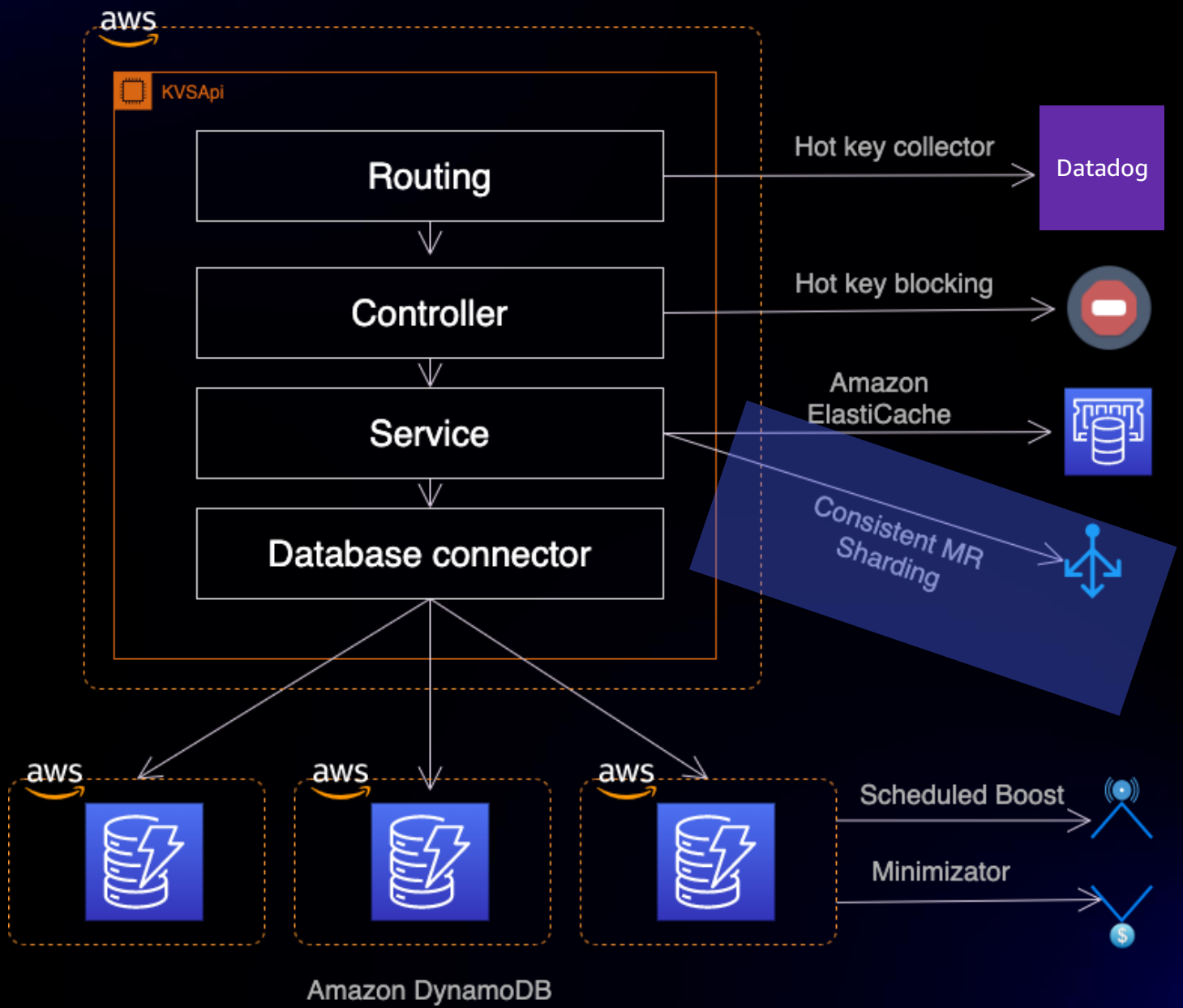
© 2021, Amazon Web Services, Inc. or its affiliates. All rights reserved.

KVS API – Add-ons



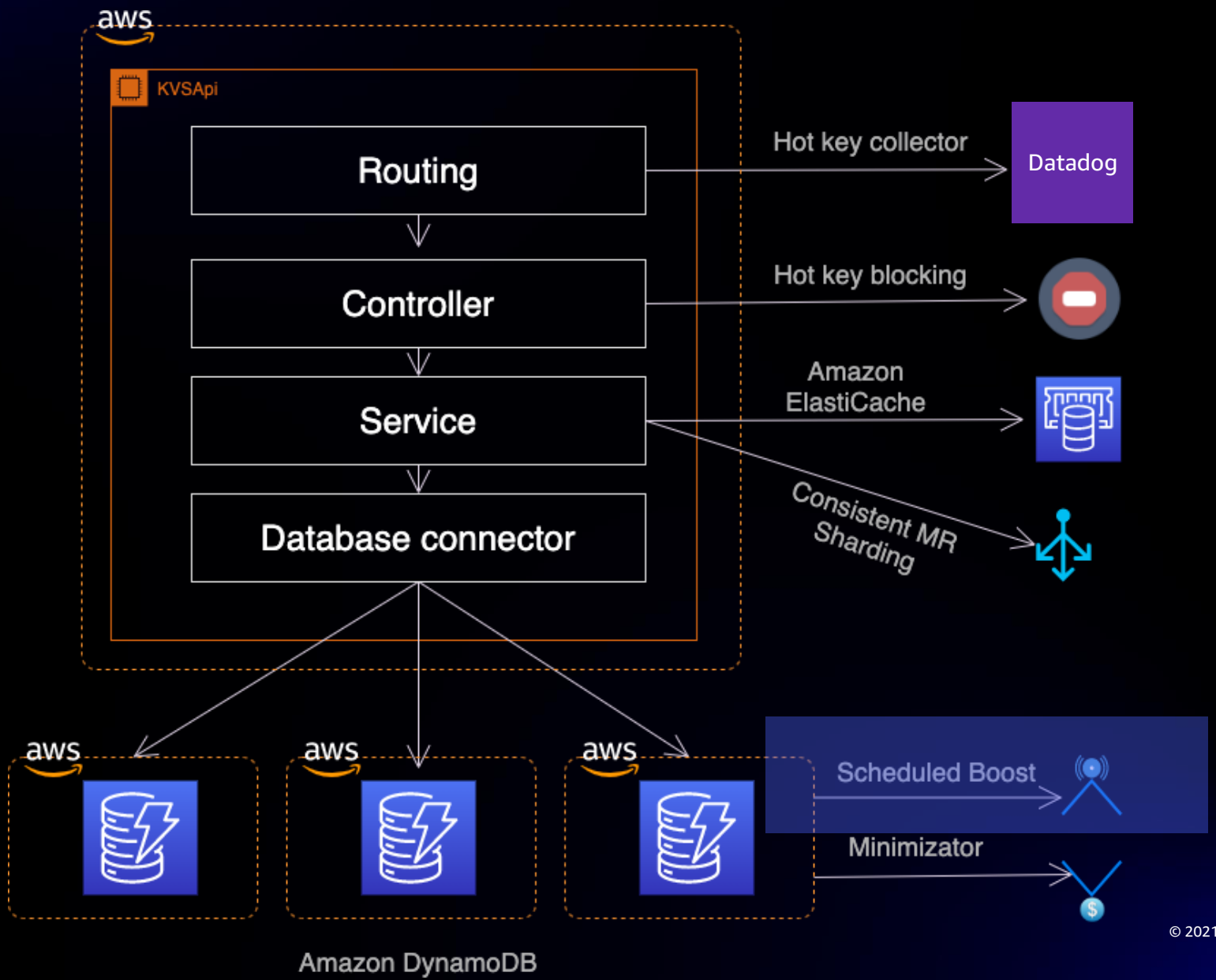
© 2021, Amazon Web Services, Inc. or its affiliates. All rights reserved.

KVS API – Add-ons



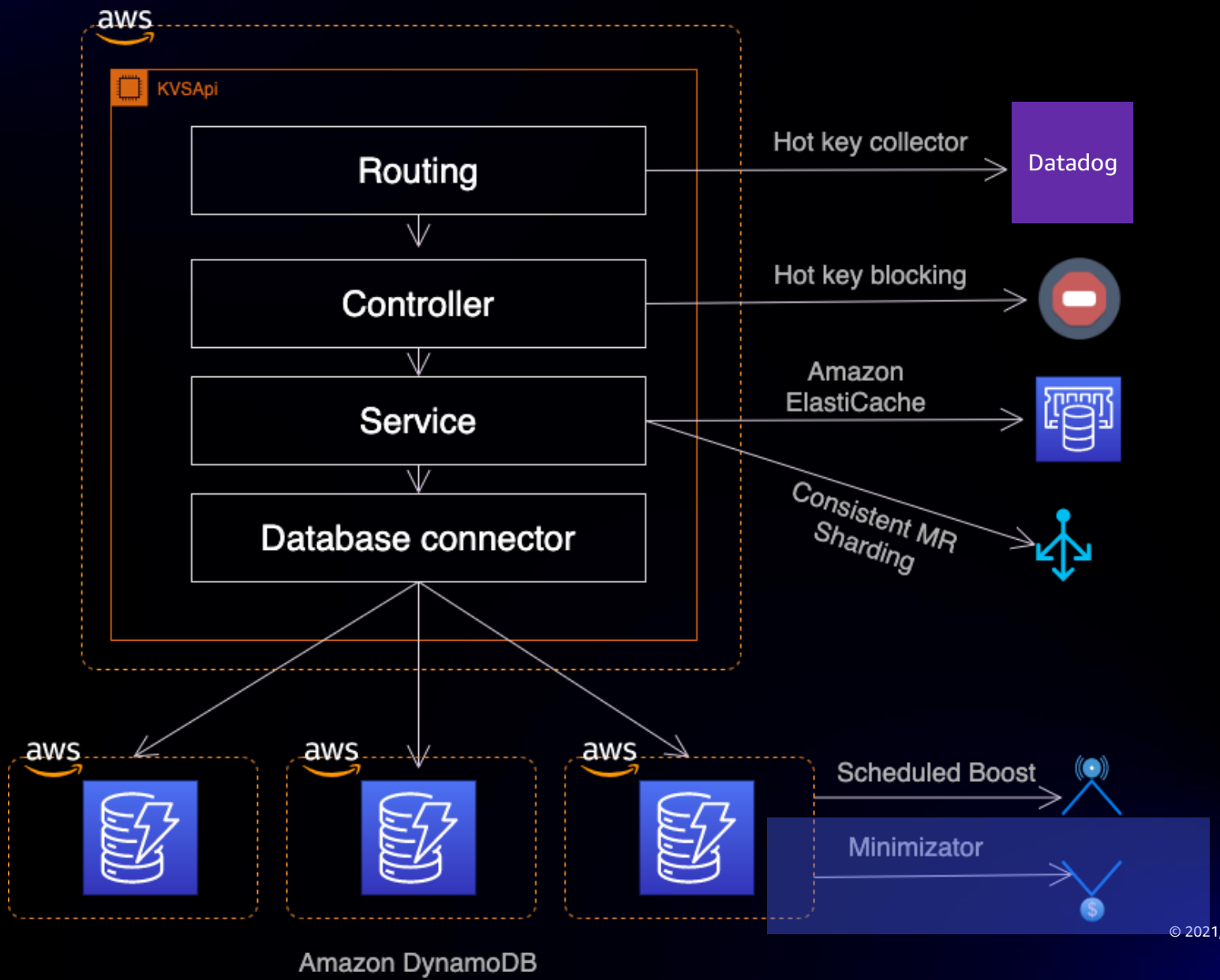
© 2021, Amazon Web Services, Inc. or its affiliates. All rights reserved.

KVS API – Add-ons



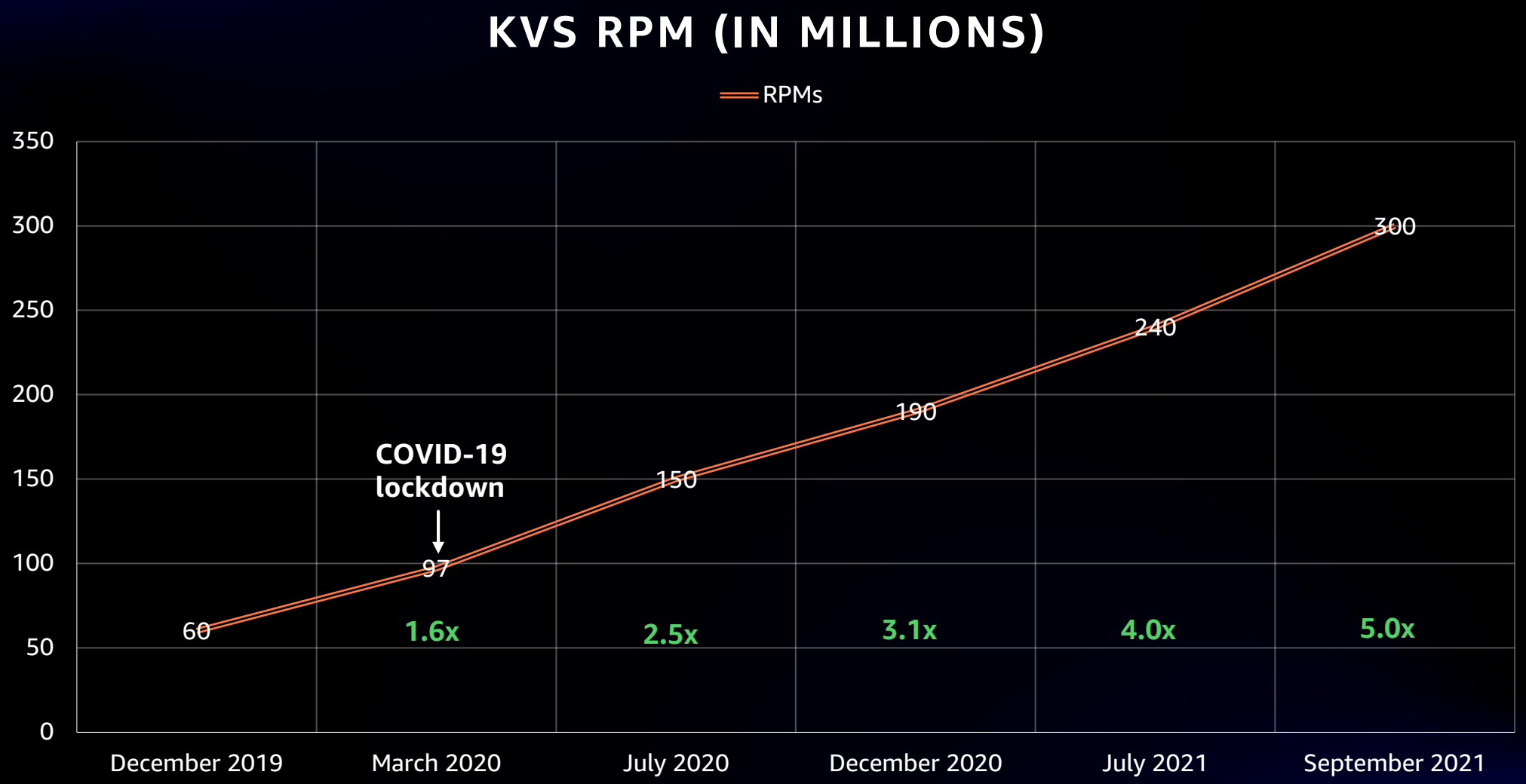
© 2021, Amazon Web Services, Inc. or its affiliates. All rights reserved.

KVS API – Add-ons



© 2021, Amazon Web Services, Inc. or its affiliates. All rights reserved.

Traffic evolution

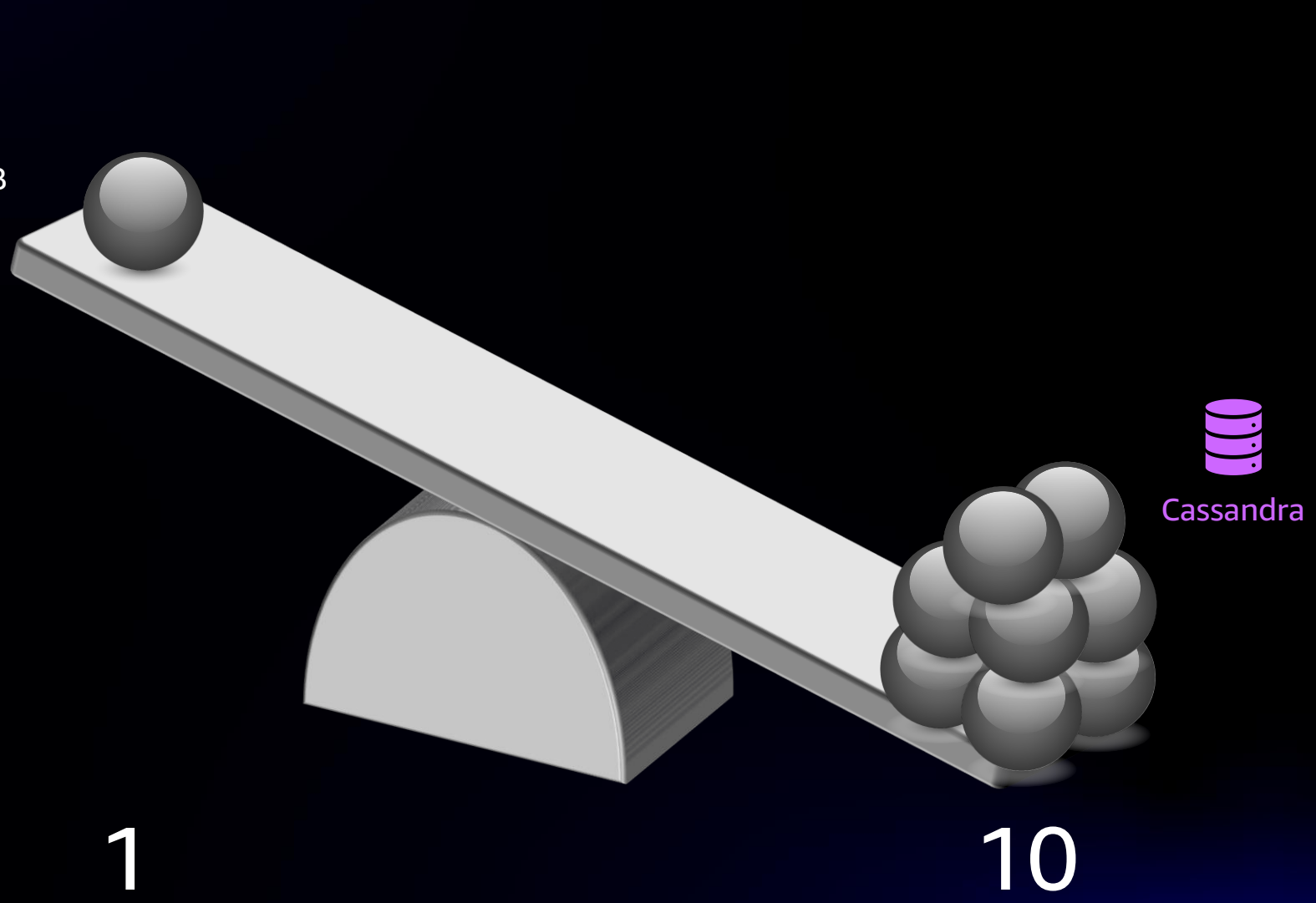




DynamoDB was key!

© 2021, Amazon Web Services, Inc. or its affiliates. All rights reserved.

KVS critical incidents (engine only)



© 2021, Amazon Web Services, Inc. or its affiliates. All rights reserved.

Disclaimer

Mercado Libre has nothing against Cassandra; it is great software
(**except for repairs**)

We only told our journey with KVS and the experiences we have had,
which may be very different in another context

© 2021, Amazon Web Services, Inc. or its affiliates. All rights reserved.

Mercado Libre's journey

IN THIS SESSION, YOU WILL LEARN HOW WE ACCOMPLISHED THE FOLLOWING

- [x] Organize a platform to support 24K microservices and more than 7K engineers
- [x] Migrate 5K databases from Apache Cassandra to DynamoDB without requesting anyone in your product organization to change a single line of code or roadmap
- [ ] Migrate your relational databases to DynamoDB for better performance and scalability while keeping legacy databases working for backward compatibility
- [ ] Build an ecosystem of platform services relying on the power of DynamoDB Streams
- [ ] Use the flexibility of DynamoDB to build a platform multiservice strategy

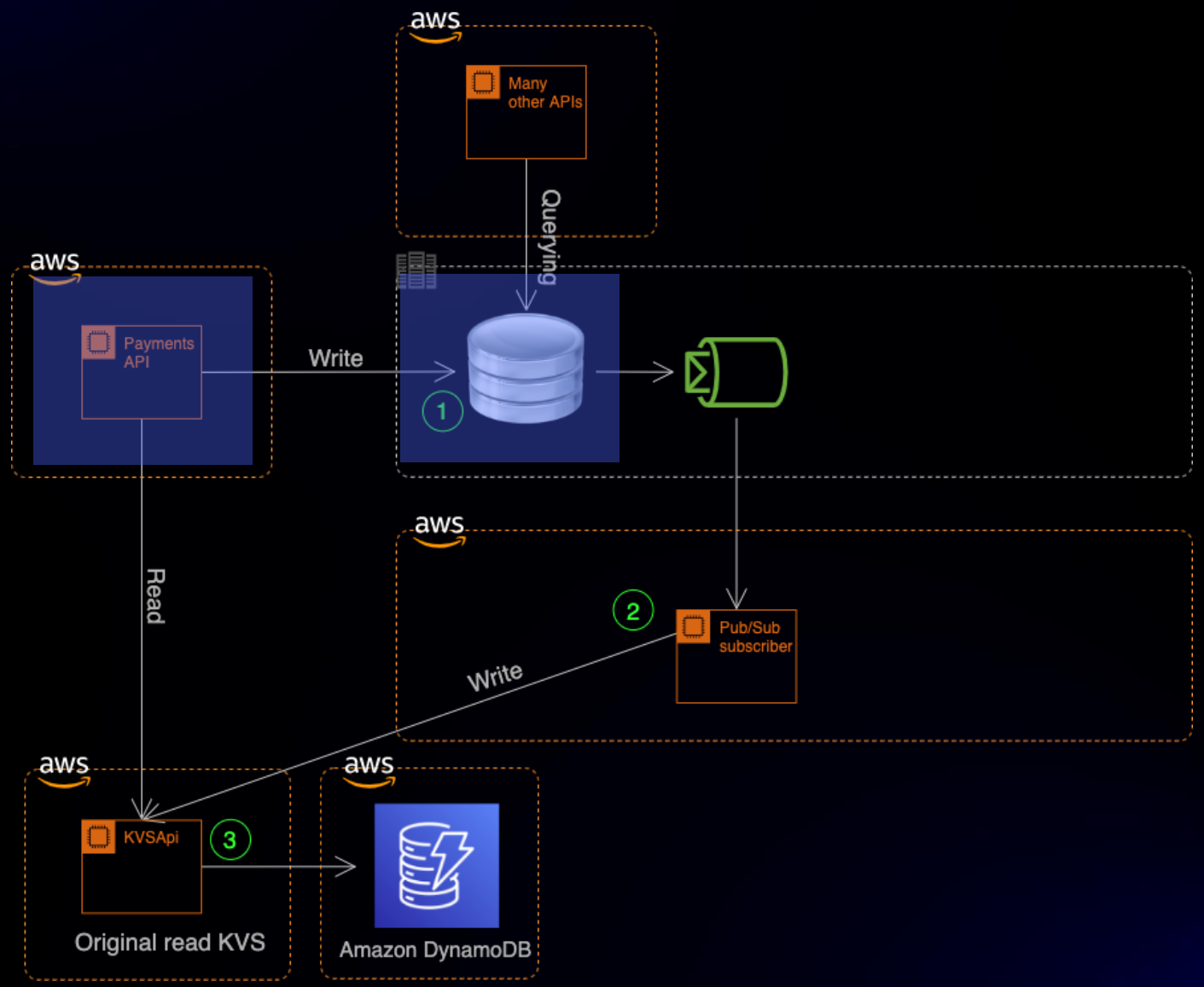
aws

© 2021, Amazon Web Services, Inc. or its affiliates. All rights reserved.

Story II
Great results . . . why not leverage that for our legacy relational databases?

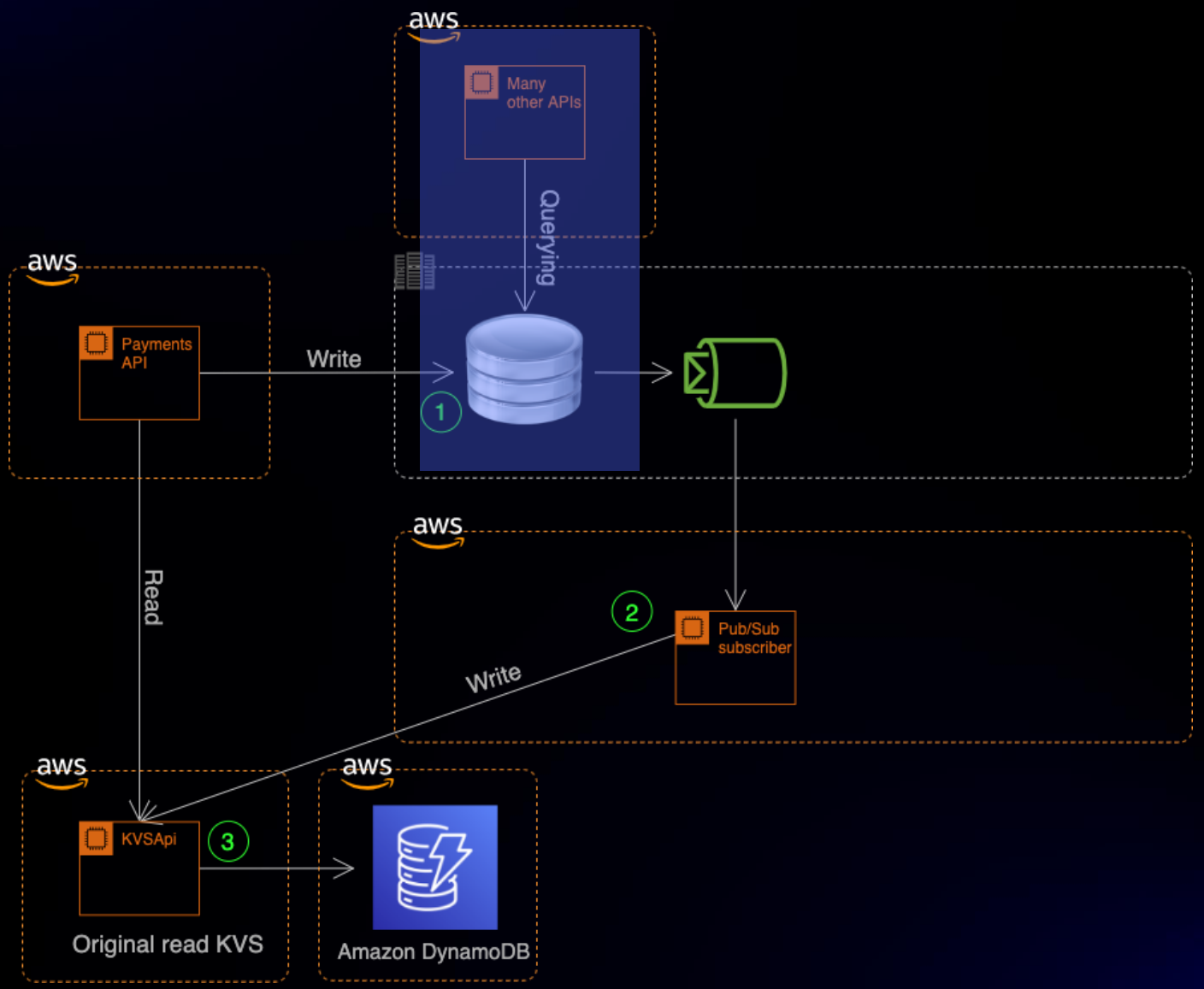
© 2021, Amazon Web Services, Inc. or its affiliates. All rights reserved.

Original payments setup



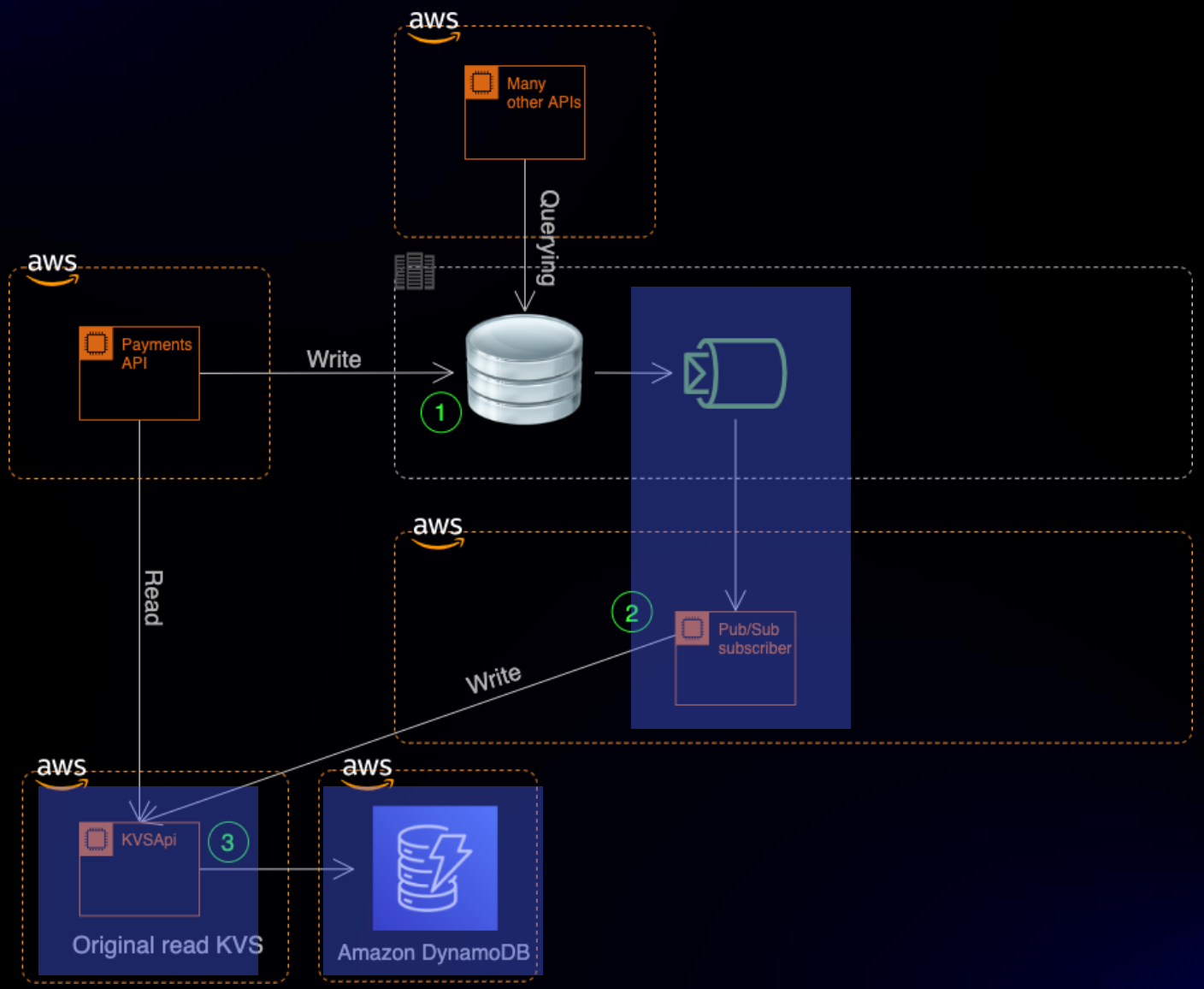
© 2021, Amazon Web Services, Inc. or its affiliates. All rights reserved.

Original payments setup



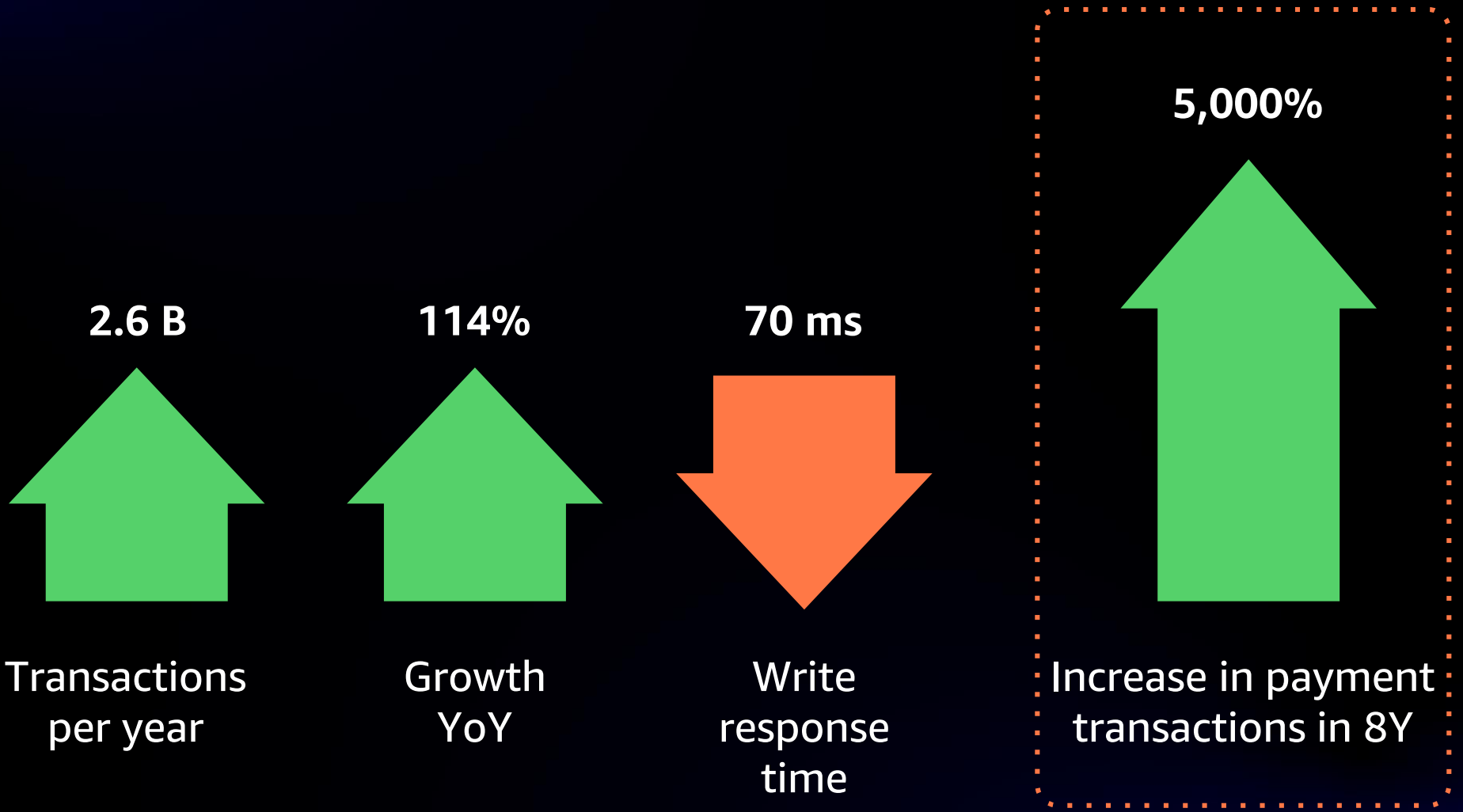
© 2021, Amazon Web Services, Inc. or its affiliates. All rights reserved.

Original payments setup



© 2021, Amazon Web Services, Inc. or its affiliates. All rights reserved.

Some key metrics



© 2021, Amazon Web Services, Inc. or its affiliates. All rights reserved.

Pain points



Ouch

Model JSON as normalized DB
More than 15 tables involved in writing a payment

High availability issues
Maintenance of highly complex RDBMs is difficult and recovery usually slow

Complexity for developers
Right configuration of connection pools and alike are complex

Scalability
Unable to scale out our data

© 2021, Amazon Web Services, Inc. or its affiliates. All rights reserved.

Step 1 – Rethink the model

Step 1

- Rethink new model
- Evaluate all access patterns
- Resolve all “out-of-the-box” functionality offered by RDBMs (such as auto-increment)
- Prepare rollback and fallback process
- Redefine domain model to use new persistence model
- Define RTO and levels of availability

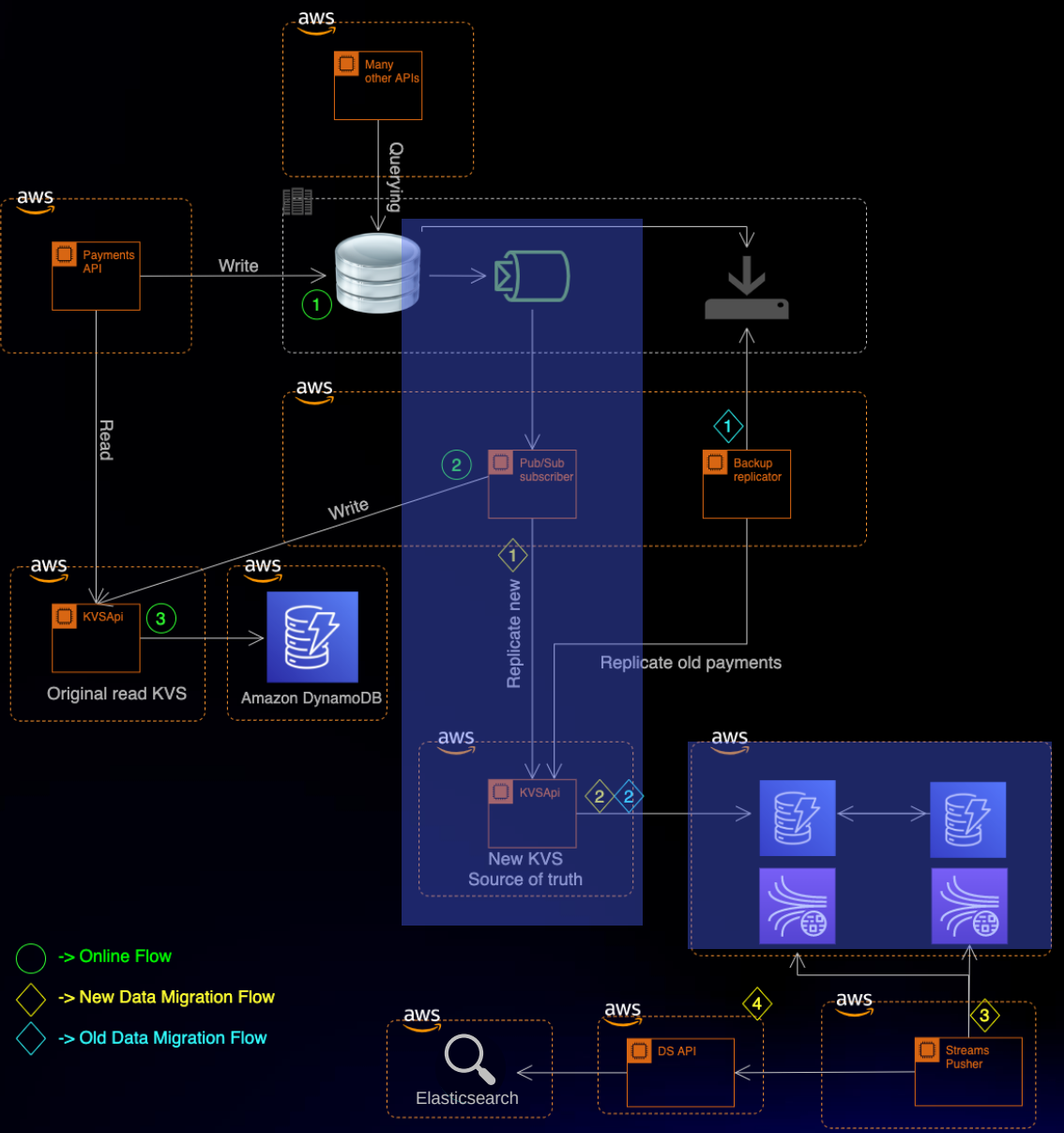
© 2021, Amazon Web Services, Inc. or its affiliates. All rights reserved.

Step 2 – Replicate data

REPLICATE NEW DATA

Key points

- Use DynamoDB GT for maximum resiliency
- Replicate the new data asynchronously
- Add new replication method for new querying engine



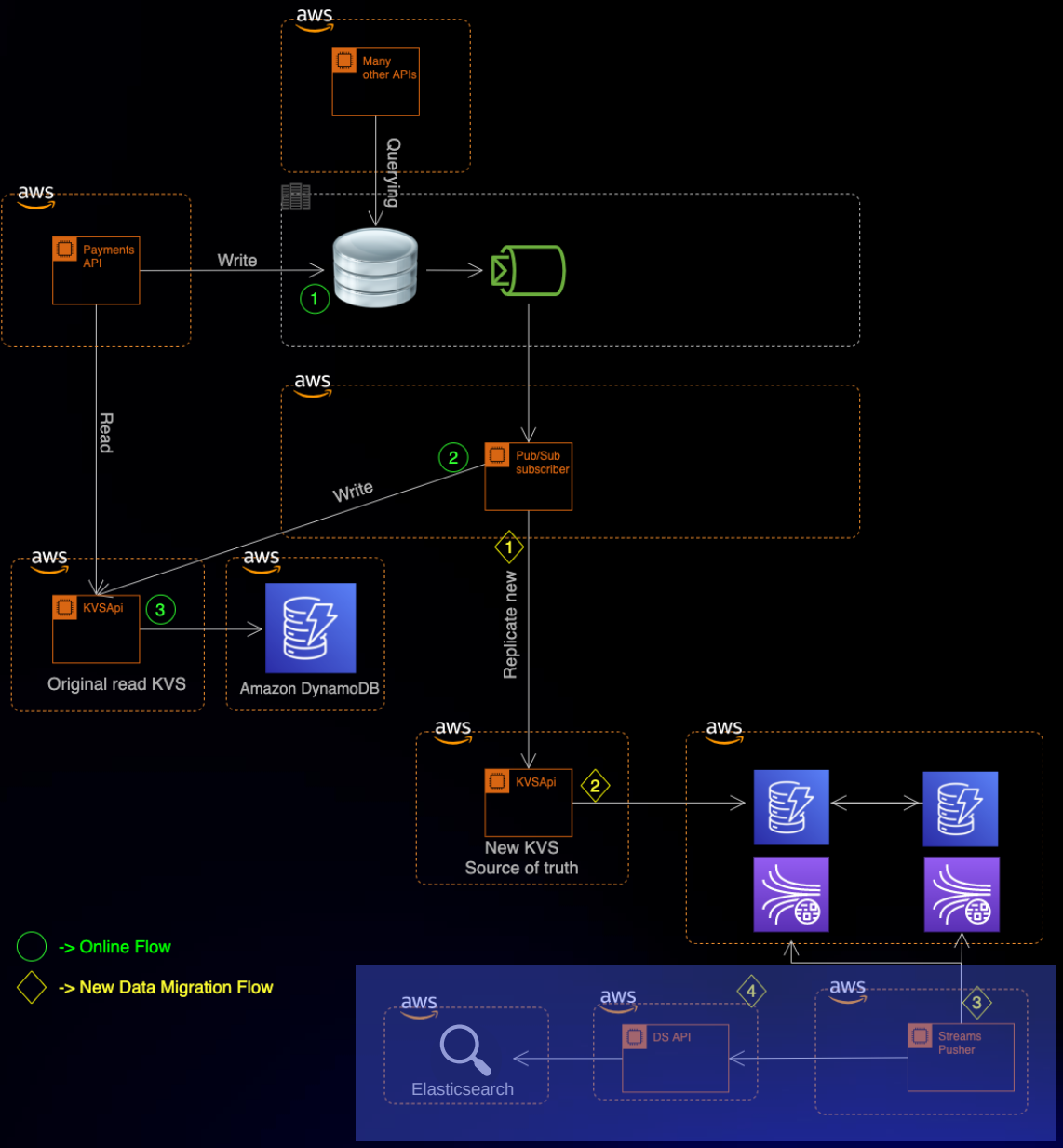
© 2021, Amazon Web Services, Inc. or its affiliates. All rights reserved.

Step 2 – Replicate data

REPLICATE NEW DATA

Key points

- Rely on Stream service for syncs



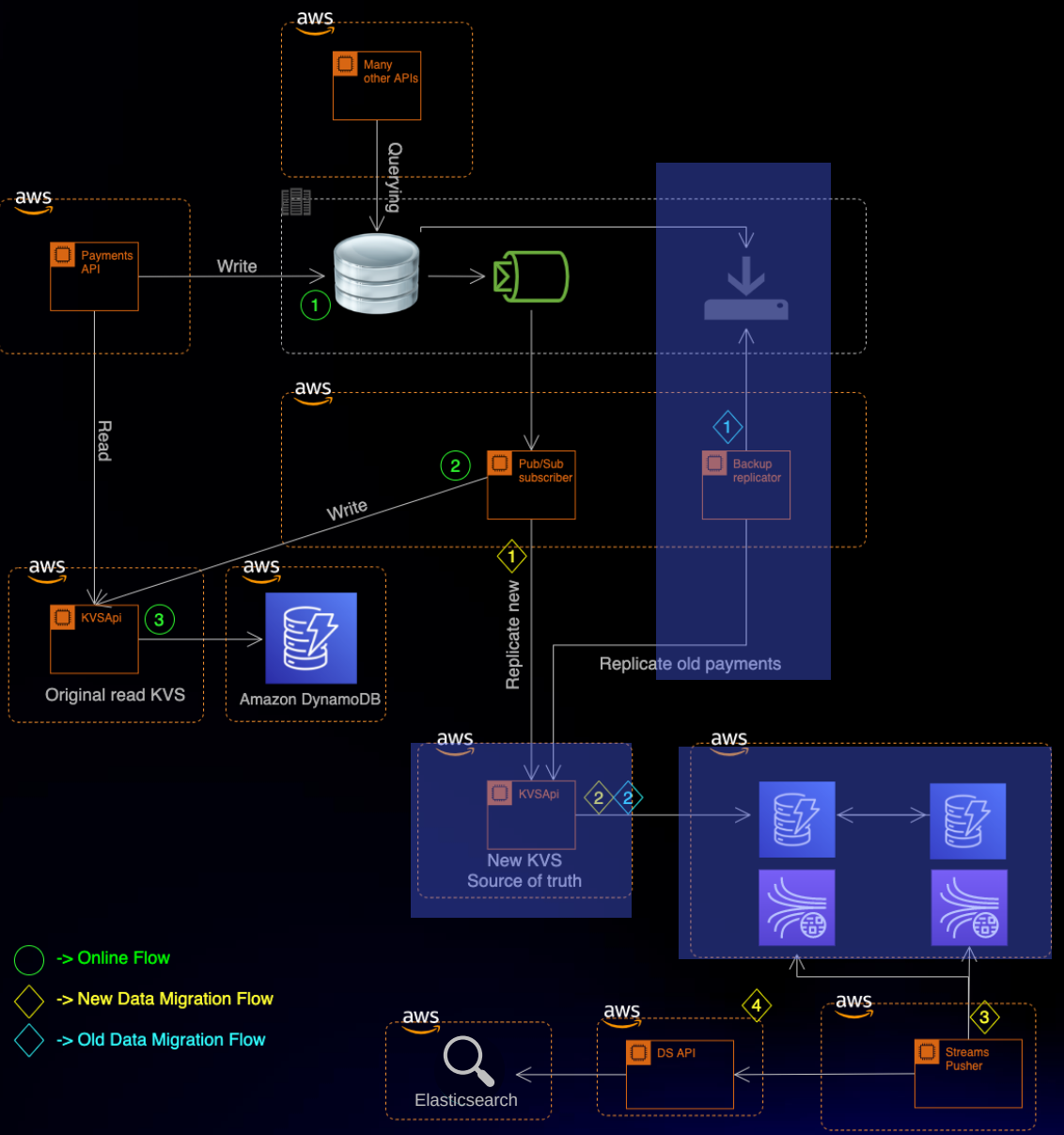
© 2021, Amazon Web Services, Inc. or its affiliates. All rights reserved.

Step 2 – Replicate data

REPLICATE STALE DATA

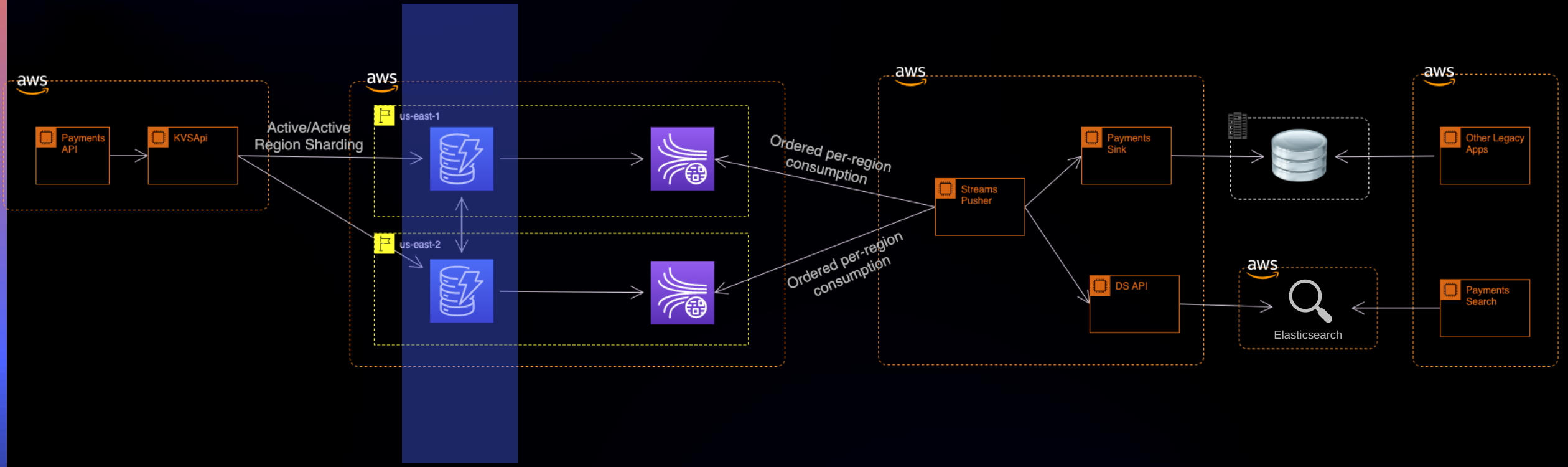
Key points

- Replicate the old data by iterating through the backup
- Create a single source of truth



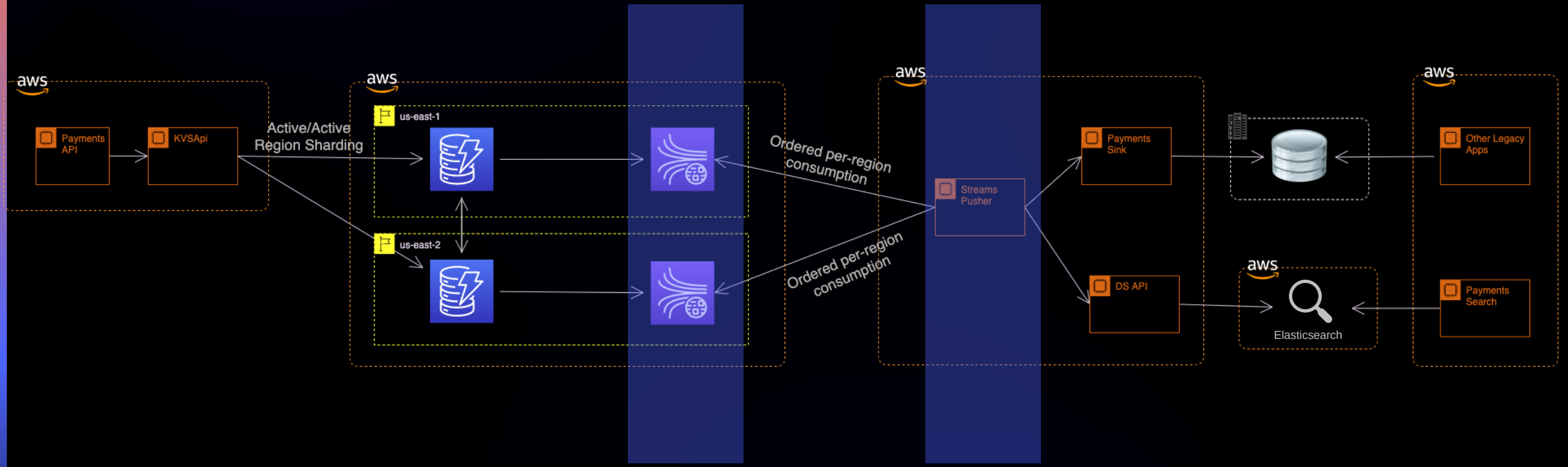
© 2021, Amazon Web Services, Inc. or its affiliates. All rights reserved.

Step 3 – Reroute traffic & replicate for legacy



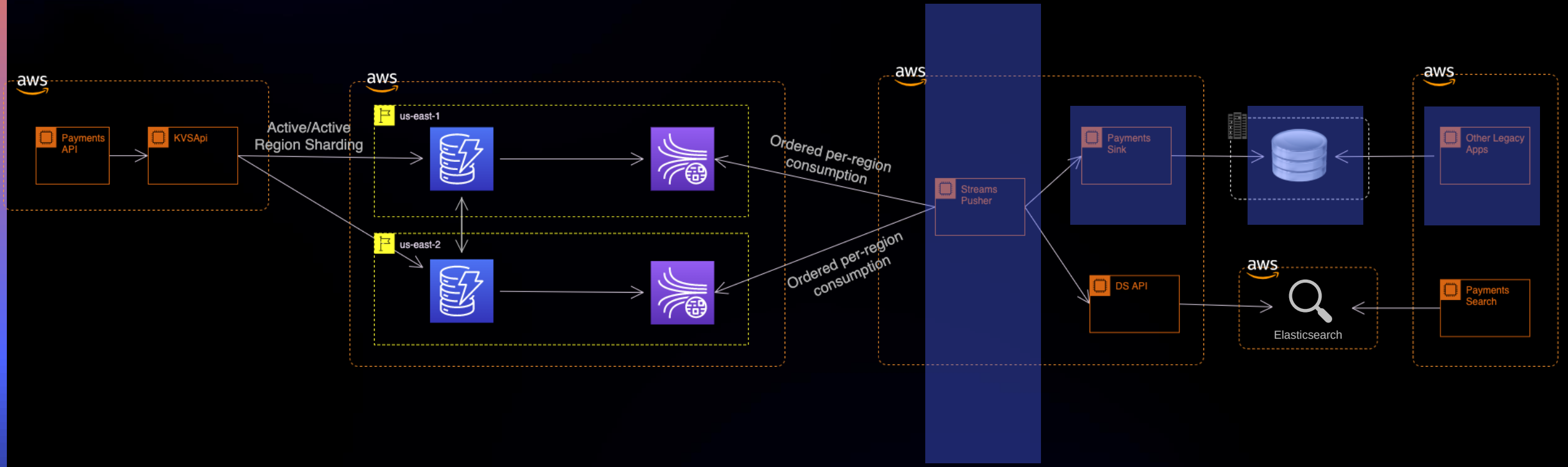
© 2021, Amazon Web Services, Inc. or its affiliates. All rights reserved.

Step 3 – Reroute traffic & replicate for legacy



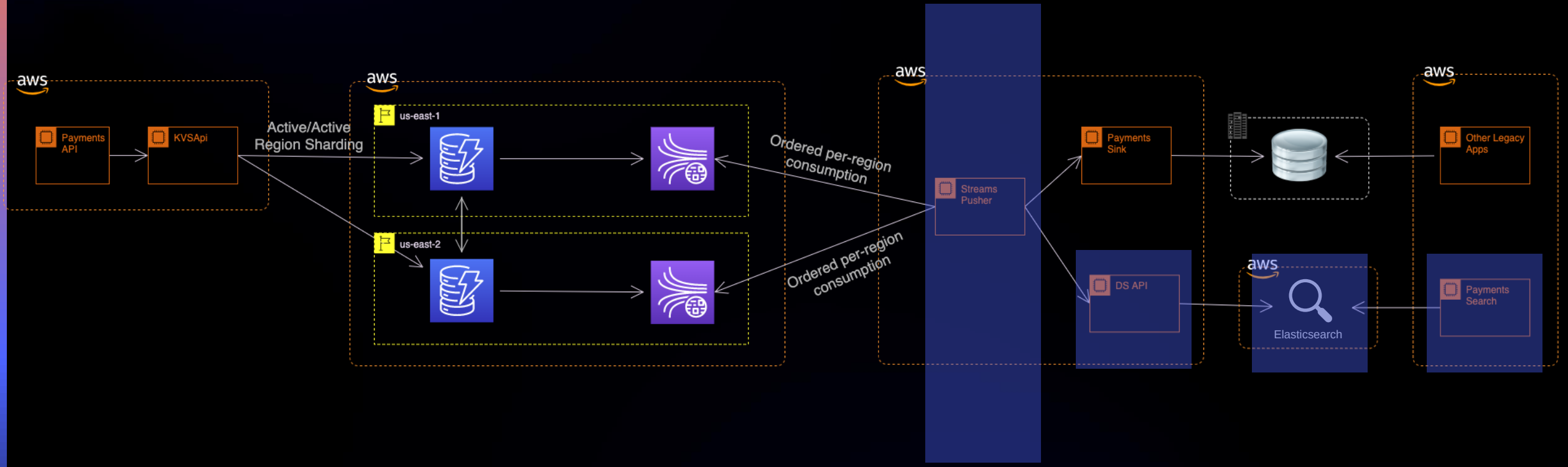
© 2021, Amazon Web Services, Inc. or its affiliates. All rights reserved.

Step 3 – Reroute traffic & replicate for legacy



© 2021, Amazon Web Services, Inc. or its affiliates. All rights reserved.

Step 3 – Reroute traffic & replicate for legacy



© 2021, Amazon Web Services, Inc. or its affiliates. All rights reserved.

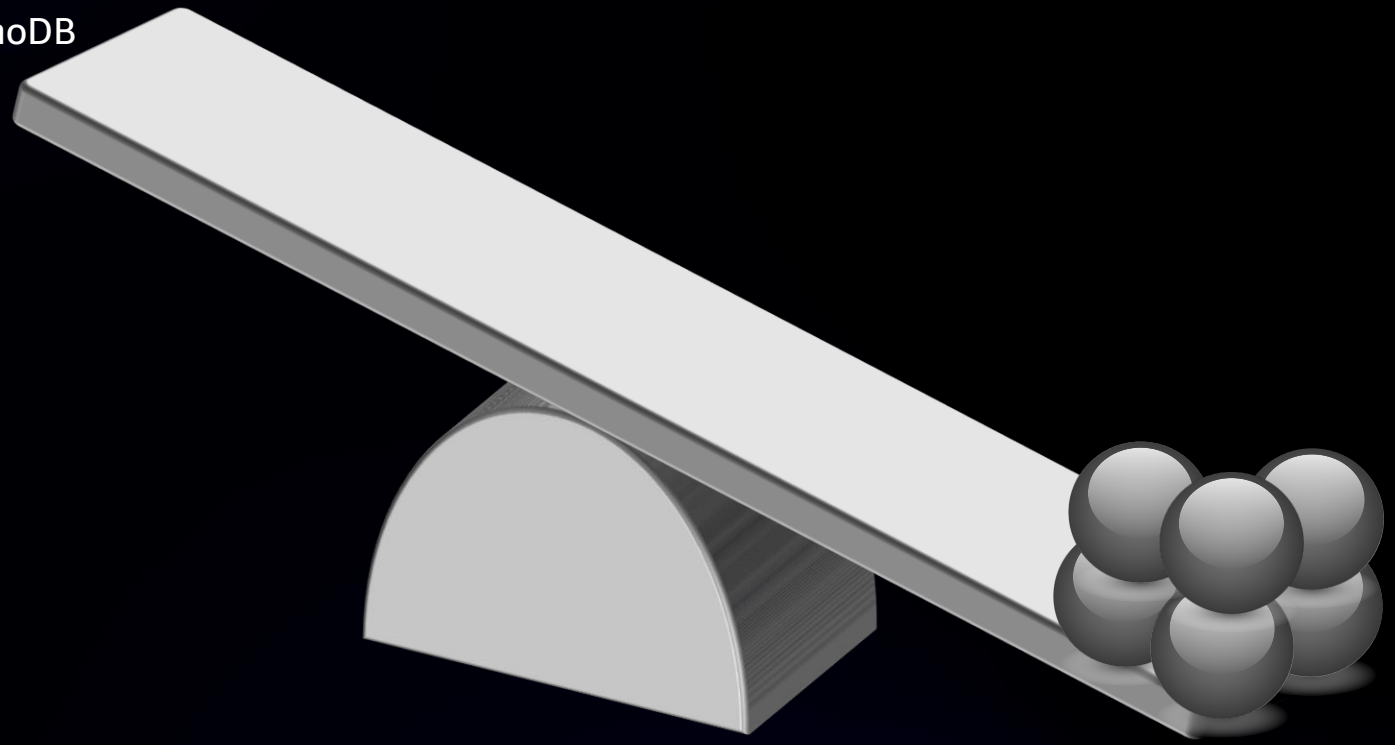
Critical incidents (engine only) – After migration



KVS



Amazon DynamoDB



0

7

© 2021, Amazon Web Services, Inc. or its affiliates. All rights reserved.

Mercado Libre's journey

IN THIS SESSION, YOU WILL LEARN HOW WE ACCOMPLISHED THE FOLLOWING

- [x] Organize a platform to support 24K microservices and more than 7K engineers
- [x] Migrate 5K databases from Apache Cassandra to DynamoDB without requesting anyone in your product organization to change a single line of code or roadmap
- [x] Migrate your relational databases to DynamoDB for better performance and scalability while keeping legacy databases working for backward compatibility
- [ ] Build an ecosystem of platform services relying on the power of DynamoDB Streams
- [ ] Use the flexibility of DynamoDB to build a platform multiservice strategy

aws

© 2021, Amazon Web Services, Inc. or its affiliates. All rights reserved.

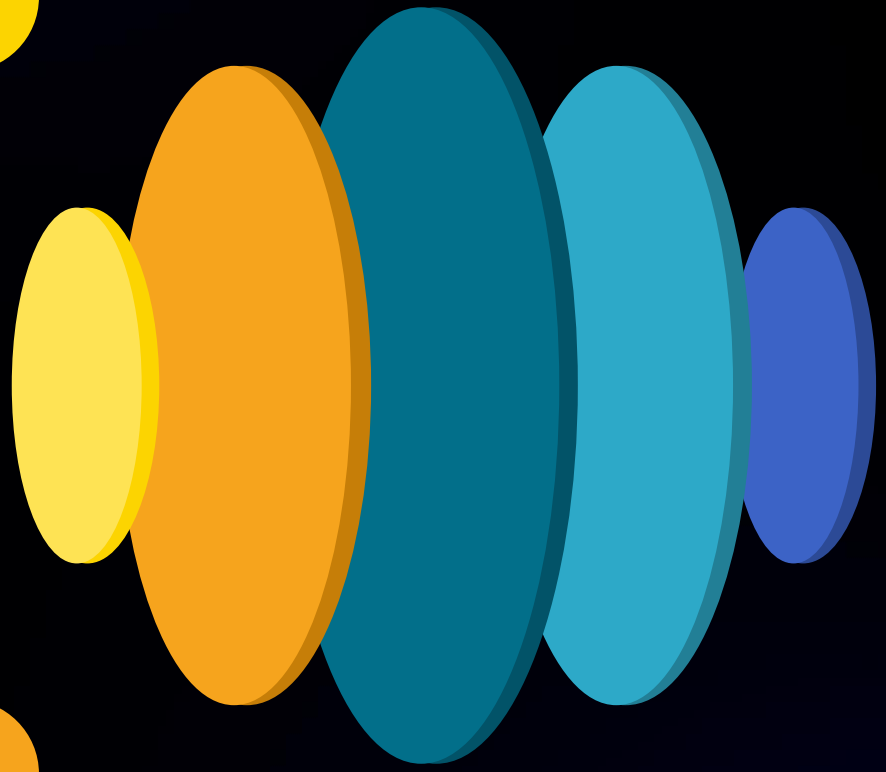
Story III
An ecosystem of services

© 2021, Amazon Web Services, Inc. or its affiliates. All rights reserved.

Apps need more than just fast KV database

Complex queries

Auditing



Exports

Aggregations & reporting

Different types of queueing

© 2021, Amazon Web Services, Inc. or its affiliates. All rights reserved.

As with the services

Every developer on its own

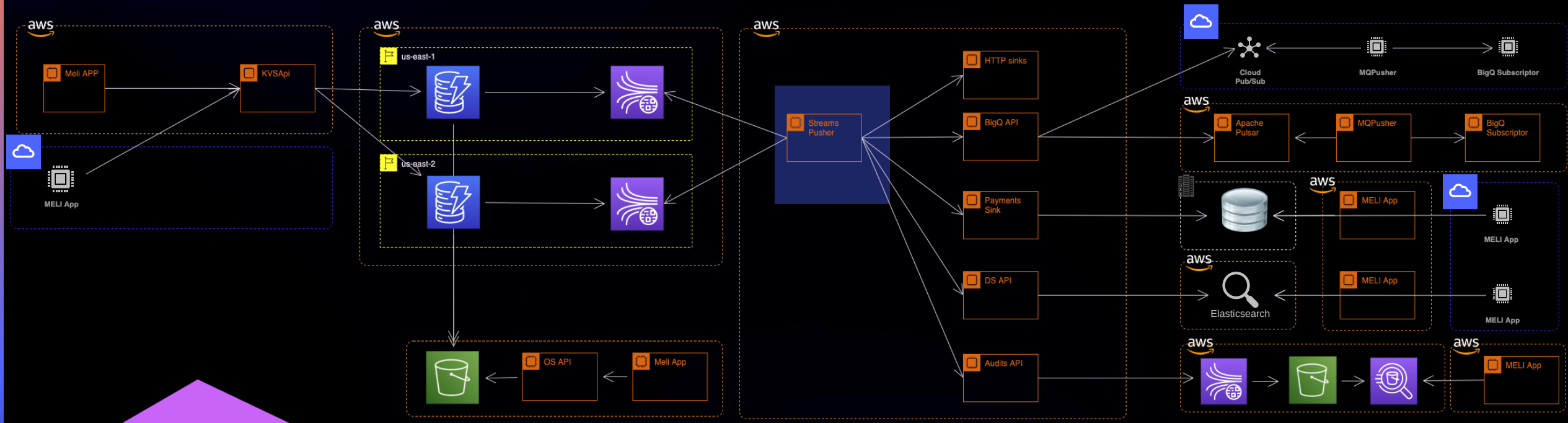


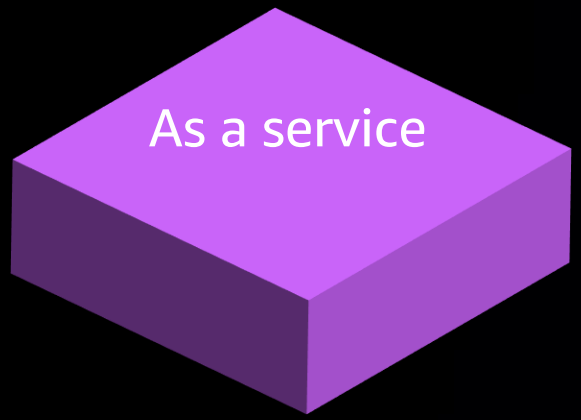
Intermediate interface

As a service

© 2021, Amazon Web Services, Inc. or its affiliates. All rights reserved.

Powered by DynamoDB Streams

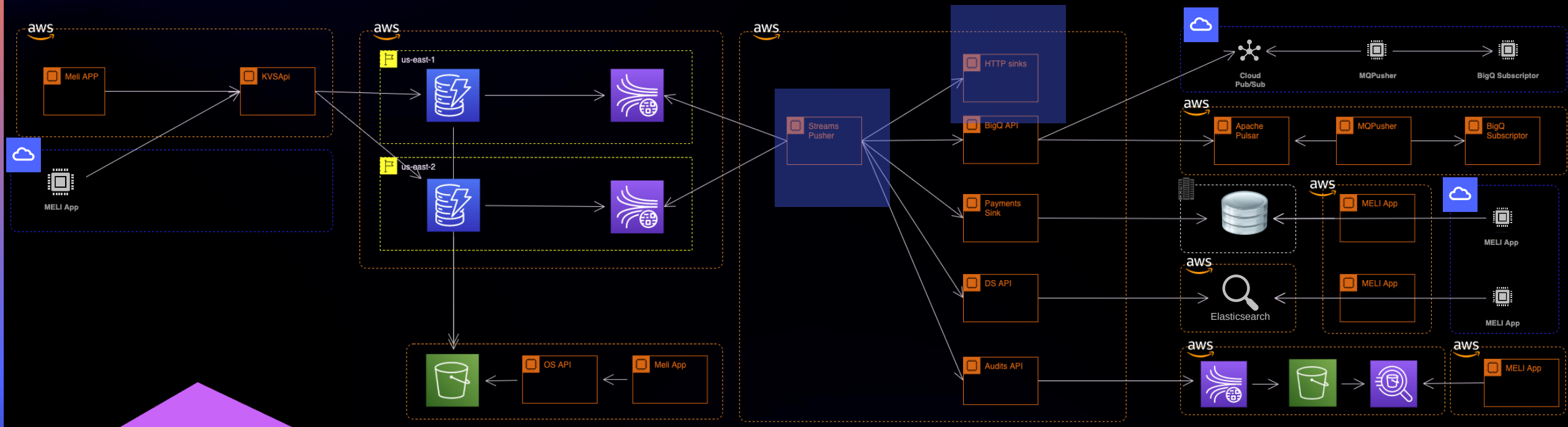




aws

© 2021, Amazon Web Services, Inc. or its affiliates. All rights reserved.

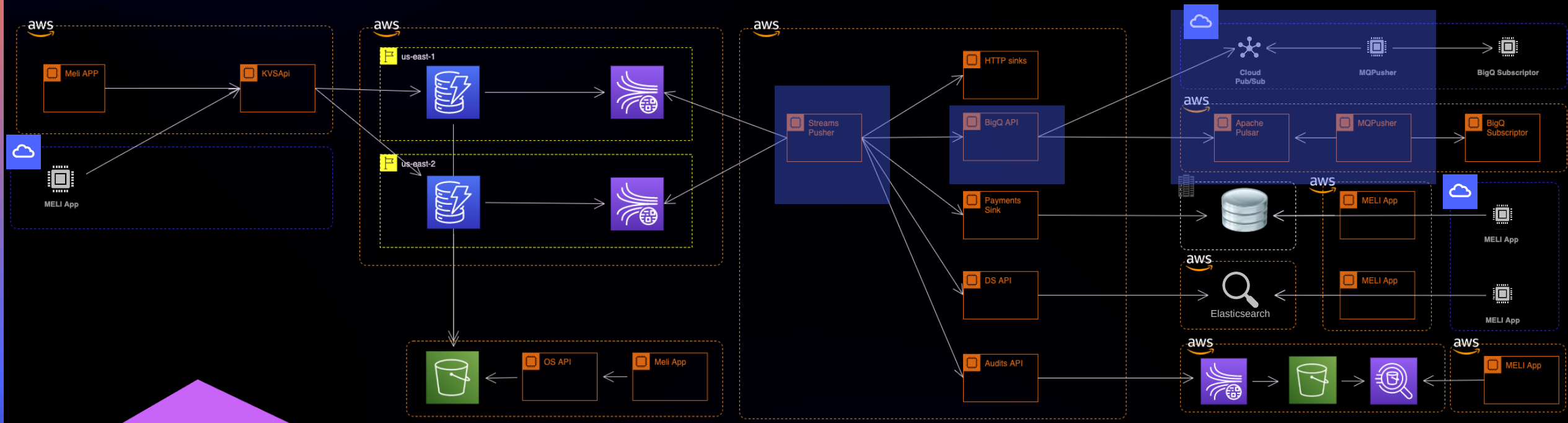
Powered by DynamoDB Streams

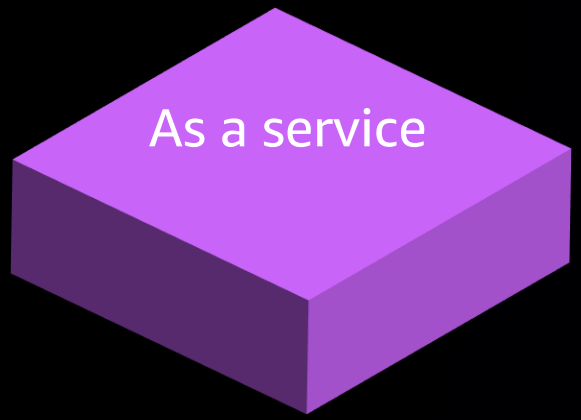


As a service

© 2021, Amazon Web Services, Inc. or its affiliates. All rights reserved.

Powered by DynamoDB Streams

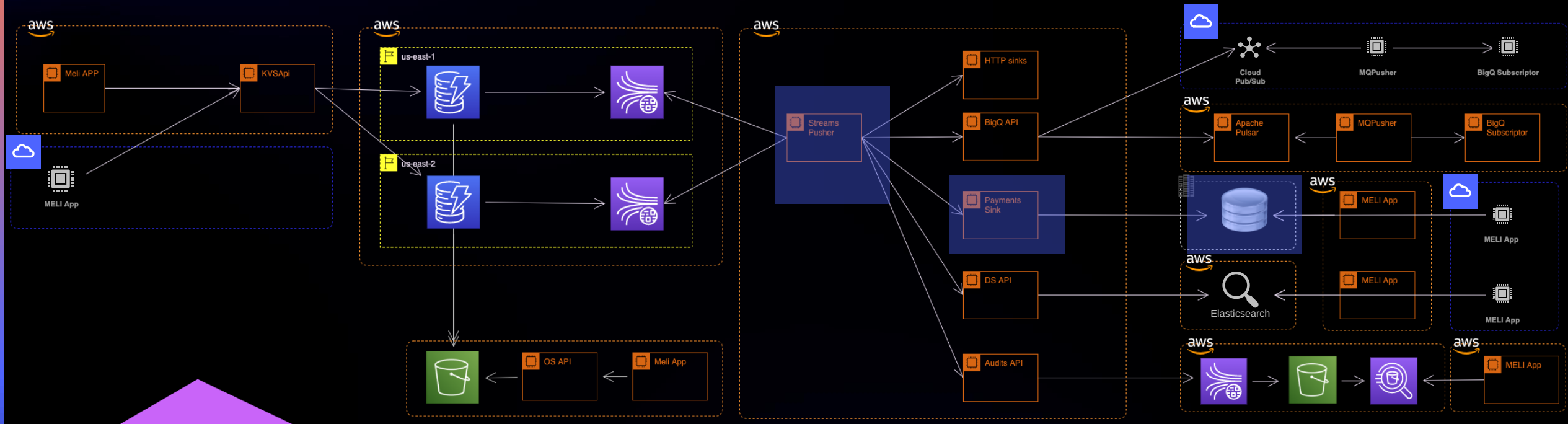


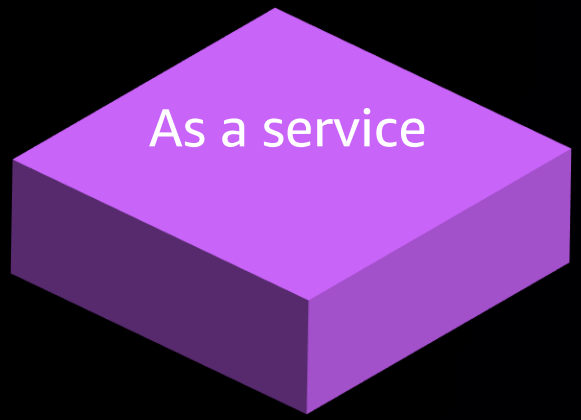


aws

© 2021, Amazon Web Services, Inc. or its affiliates. All rights reserved.

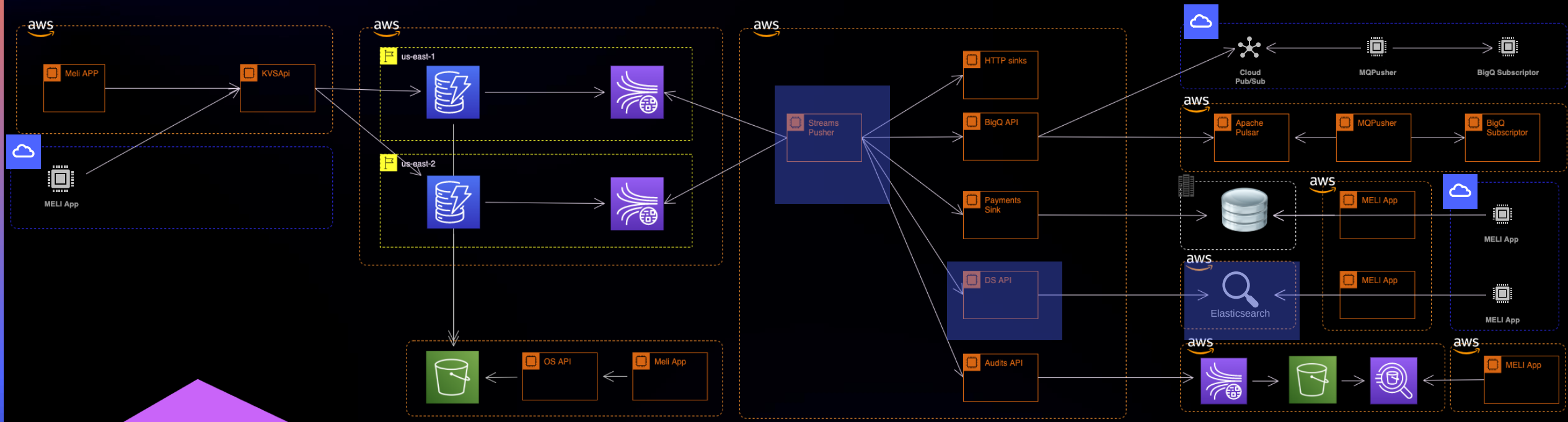
Powered by DynamoDB Streams

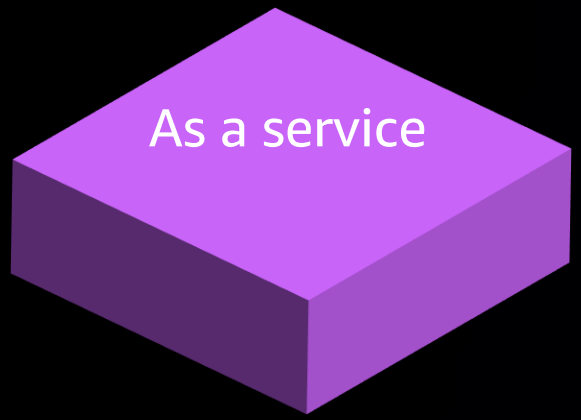




© 2021, Amazon Web Services, Inc. or its affiliates. All rights reserved.

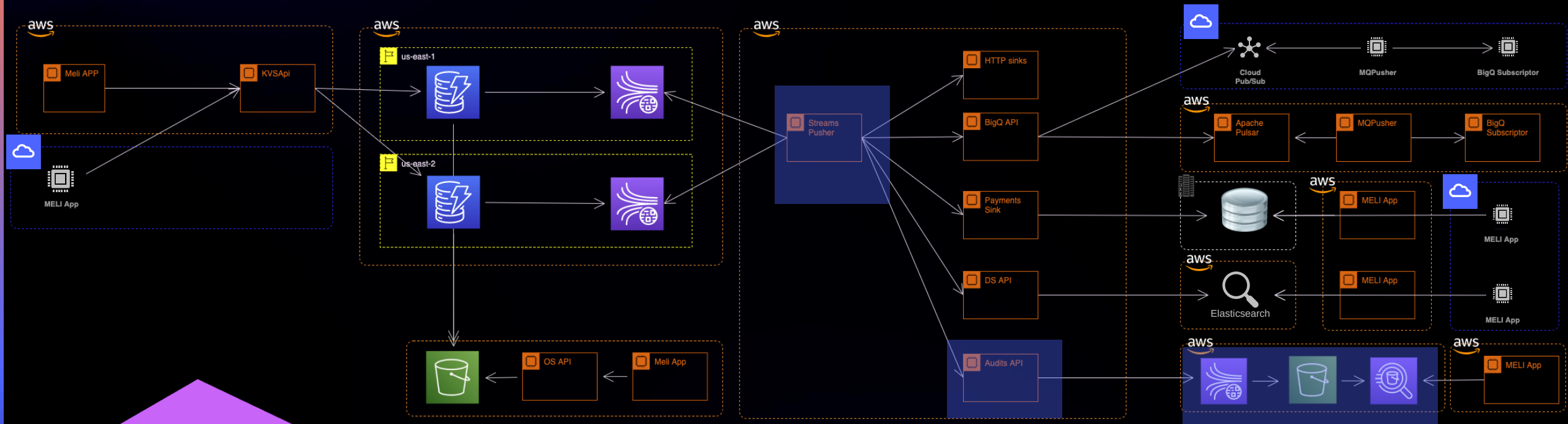
Powered by DynamoDB Streams

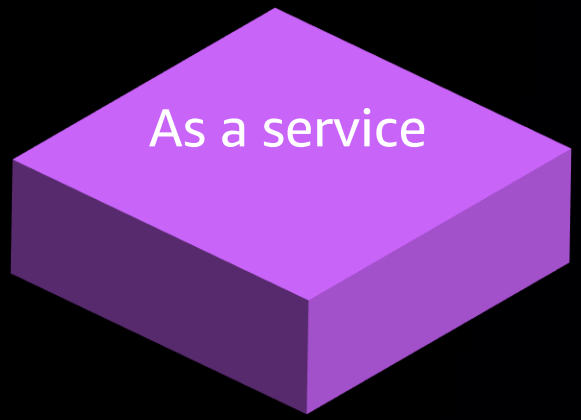




© 2021, Amazon Web Services, Inc. or its affiliates. All rights reserved.

Powered by DynamoDB Streams

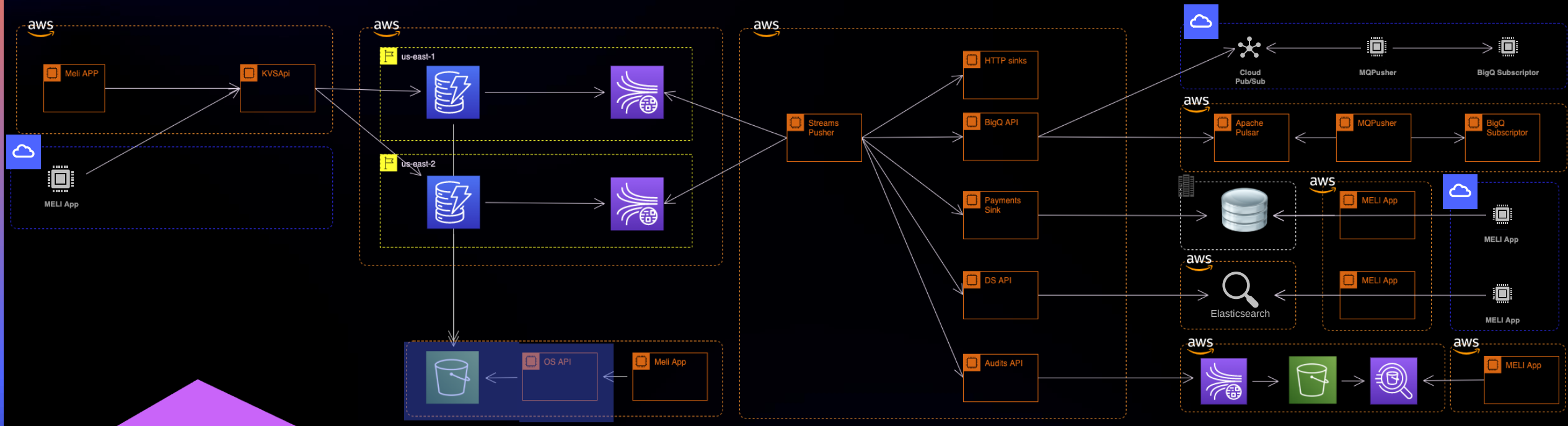


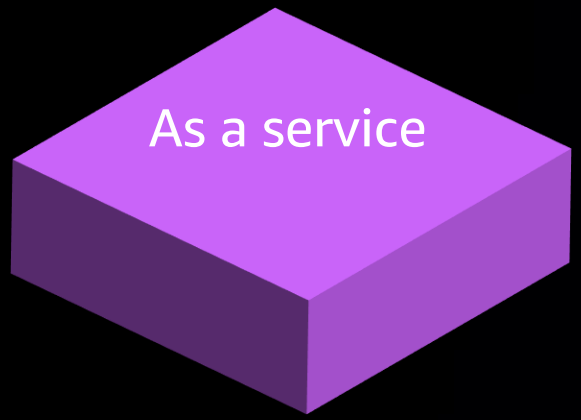


aws

© 2021, Amazon Web Services, Inc. or its affiliates. All rights reserved.

Powered by DynamoDB Streams





© 2021, Amazon Web Services, Inc. or its affiliates. All rights reserved.

Mercado Libre's journey

IN THIS SESSION, YOU WILL LEARN HOW WE ACCOMPLISHED THE FOLLOWING

- ✓ Organize a platform to support 24K microservices and more than 7K engineers
- ✓ Migrate 5K databases from Apache Cassandra to DynamoDB without requesting anyone in your product organization to change a single line of code or roadmap
- ✓ Migrate your relational databases to DynamoDB for better performance and scalability while keeping legacy databases working for backward compatibility
- ✓ Build an ecosystem of platform services relying on the power of DynamoDB Streams
- ○ Use the flexibility of DynamoDB to build a platform multiservice strategy

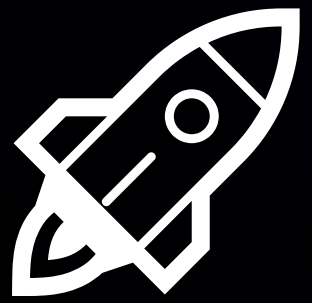
aws

© 2021, Amazon Web Services, Inc. or its affiliates. All rights reserved.

Story IV
Multiple KVS flavors

© 2021, Amazon Web Services, Inc. or its affiliates. All rights reserved.

Maximize KVS value proposition



Transparent KVS segmentation
Ability to segment data based on certain criteria almost transparently for our developers

KVS Vault
Extra secure KVS for matching PII data requirements

Q-KVS
Secondary indexes and sort key to allow more complex searches

aws

© 2021, Amazon Web Services, Inc. or its affiliates. All rights reserved.

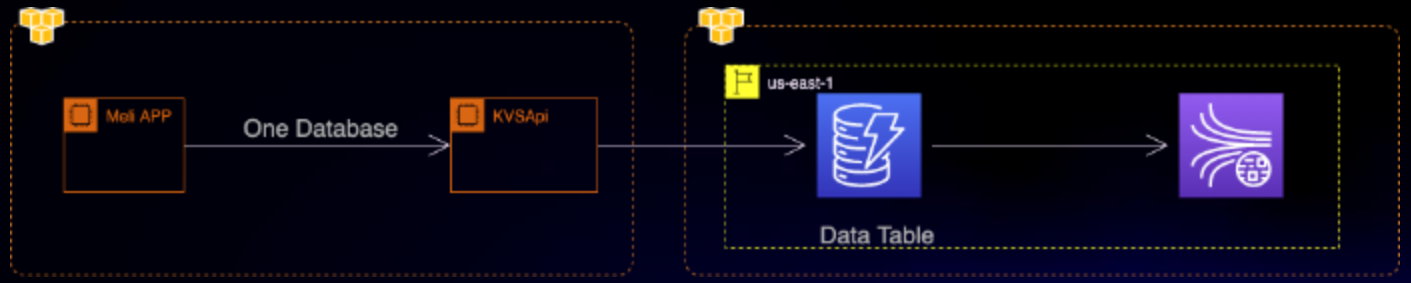
Data segmentation

© 2021, Amazon Web Services, Inc. or its affiliates. All rights reserved.

Semi-transparent data segmentation



Classic KVS table	
K	V
#	
#	
#	
#	
#	
#	
#	



© 2021, Amazon Web Services, Inc. or its affiliates. All rights reserved.

Semi-transparent data segmentation



BR table	
K	V
#	
#	
#	

CO table	
K	V
#	

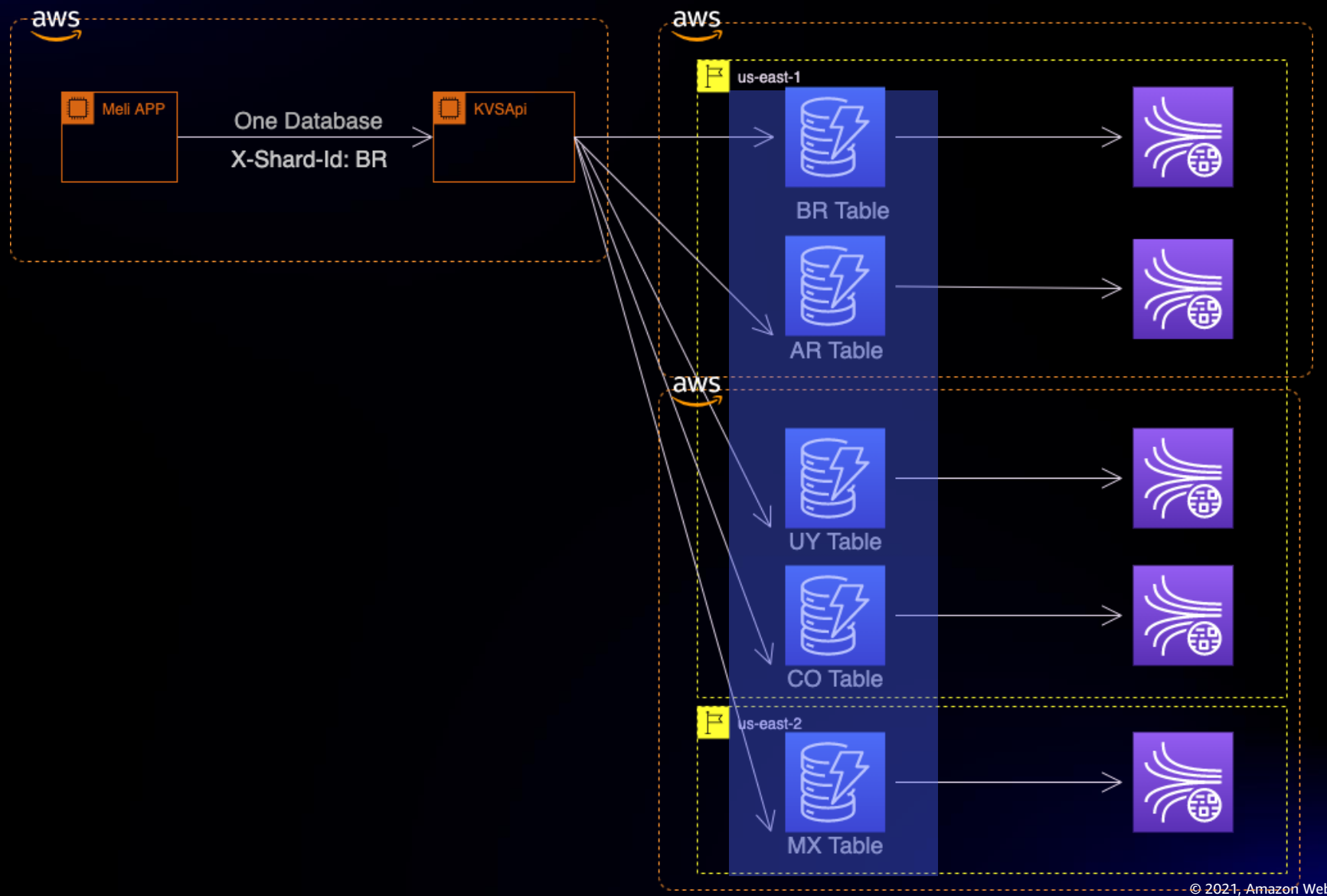
UY table	
K	V
#	

MX table	
K	V
#	
#	
#	

AR table	
K	V
#	
#	
#	

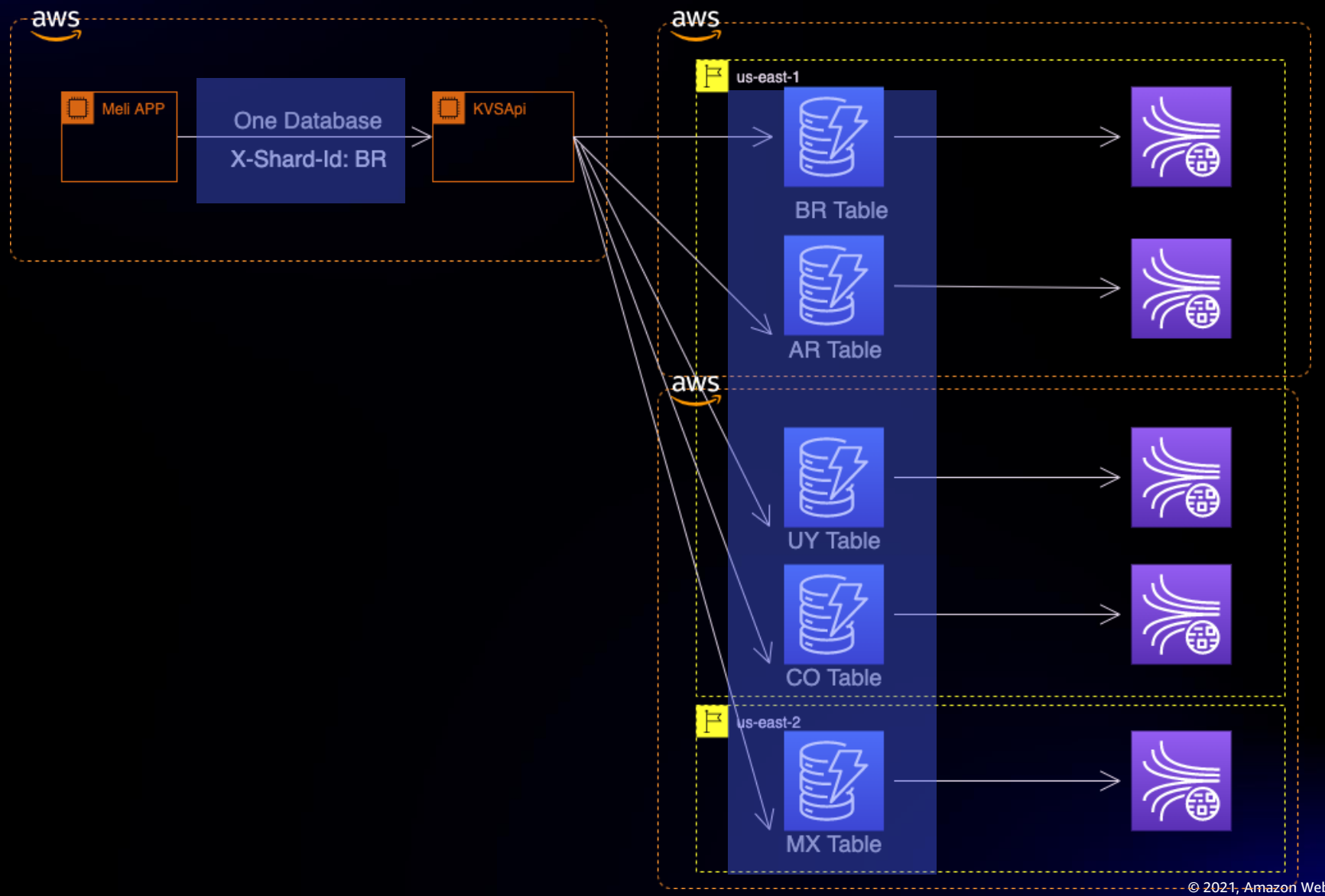
© 2021, Amazon Web Services, Inc. or its affiliates. All rights reserved.

Semi-transparent data segmentation – New



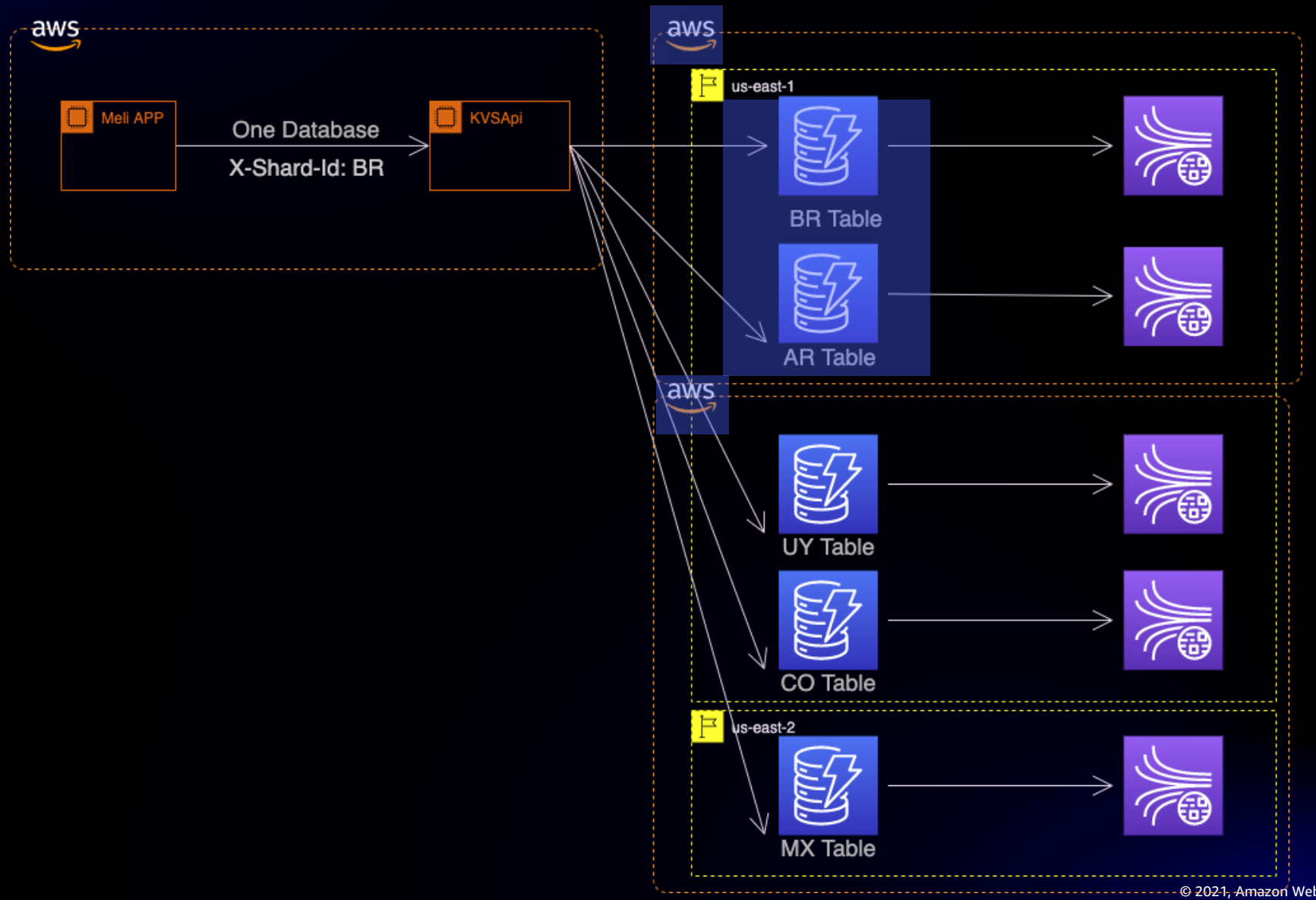
© 2021, Amazon Web Services, Inc. or its affiliates. All rights reserved.

Semi-transparent data segmentation – New



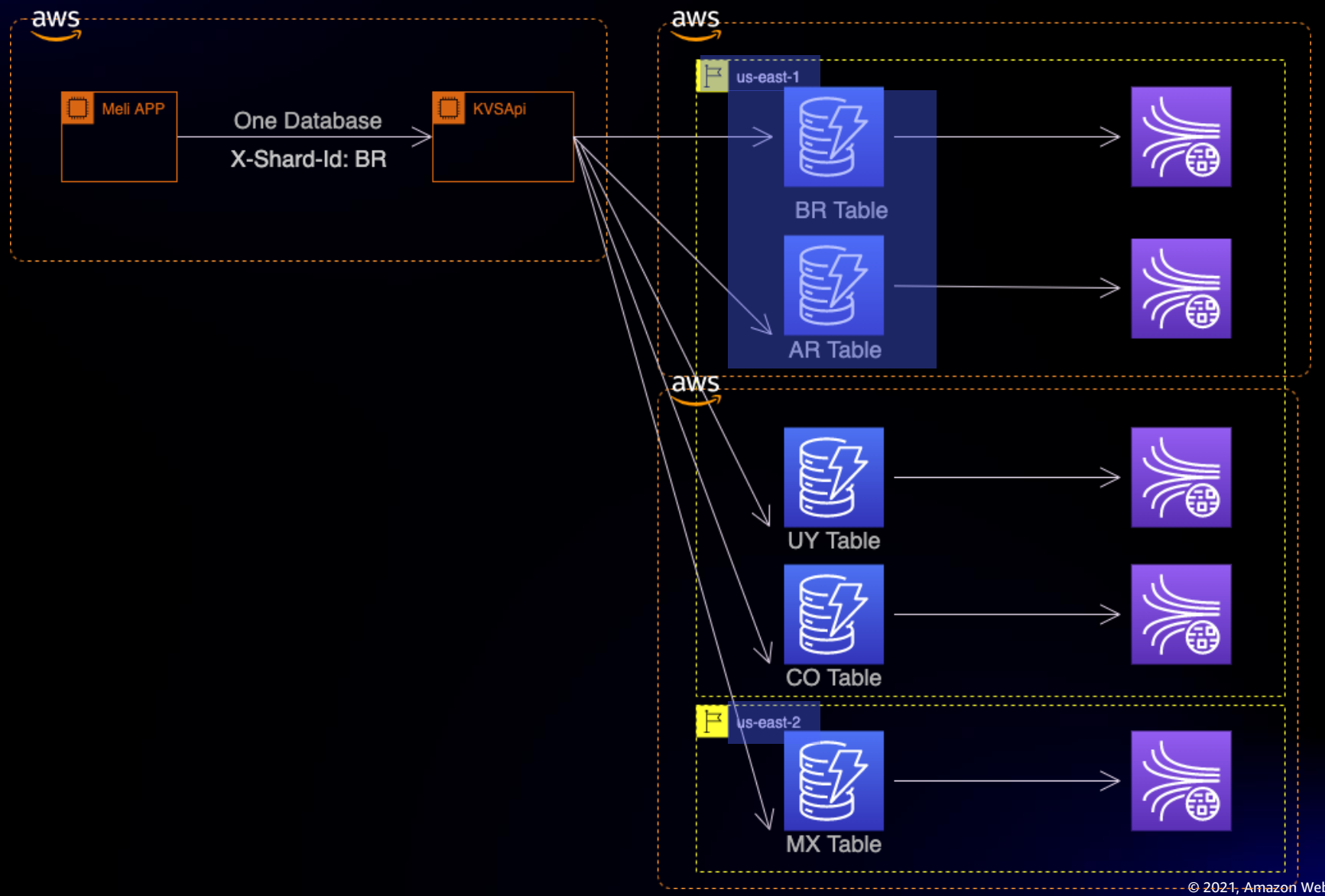
© 2021, Amazon Web Services, Inc. or its affiliates. All rights reserved.

Semi-transparent data segmentation – New



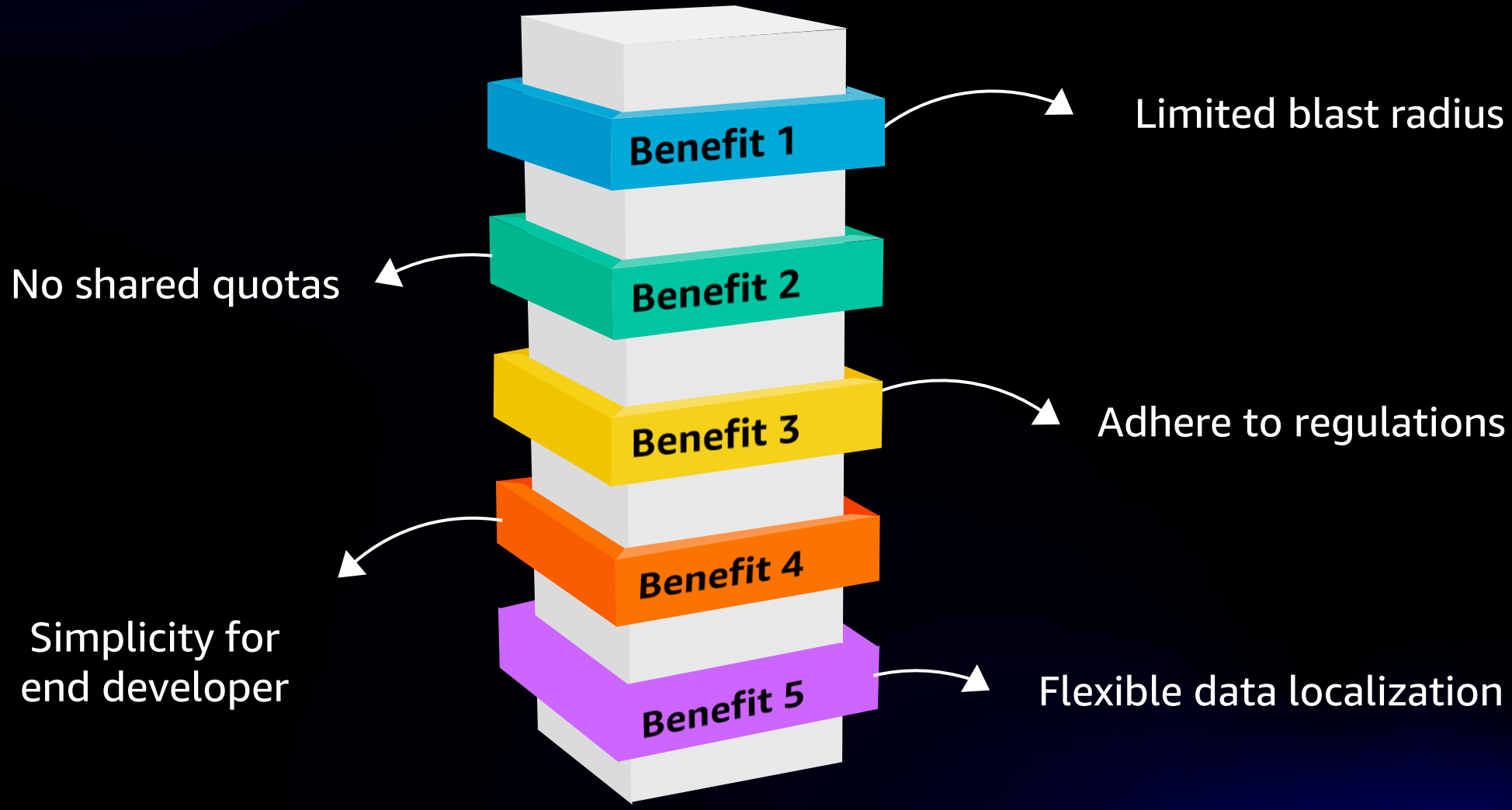
© 2021, Amazon Web Services, Inc. or its affiliates. All rights reserved.

Semi-transparent data segmentation – New



© 2021, Amazon Web Services, Inc. or its affiliates. All rights reserved.

Data segmentation – Benefits



© 2021, Amazon Web Services, Inc. or its affiliates. All rights reserved.

Vault

© 2021, Amazon Web Services, Inc. or its affiliates. All rights reserved.

KVS Vault
Containing PII data



Key points

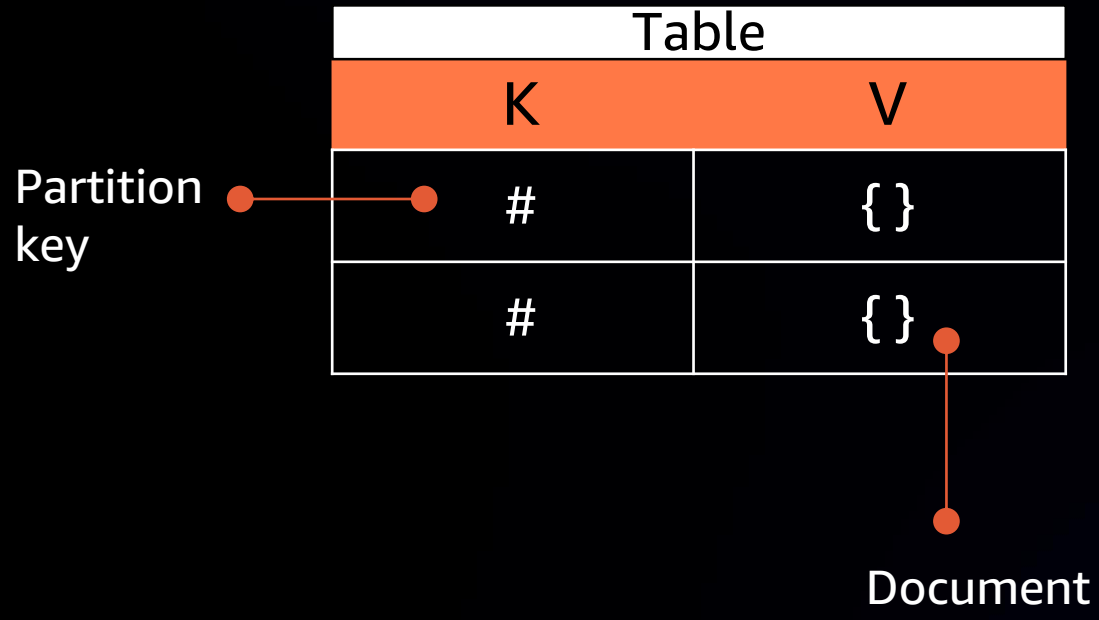
- Isolated AWS accounts
- Encryption at API level
- Restricted integrations
- Audit all reads and writes using our audit service
- Other “secret” security measures

© 2021, Amazon Web Services, Inc. or its affiliates. All rights reserved.

Q-KVS

© 2021, Amazon Web Services, Inc. or its affiliates. All rights reserved.

Traditional KVS



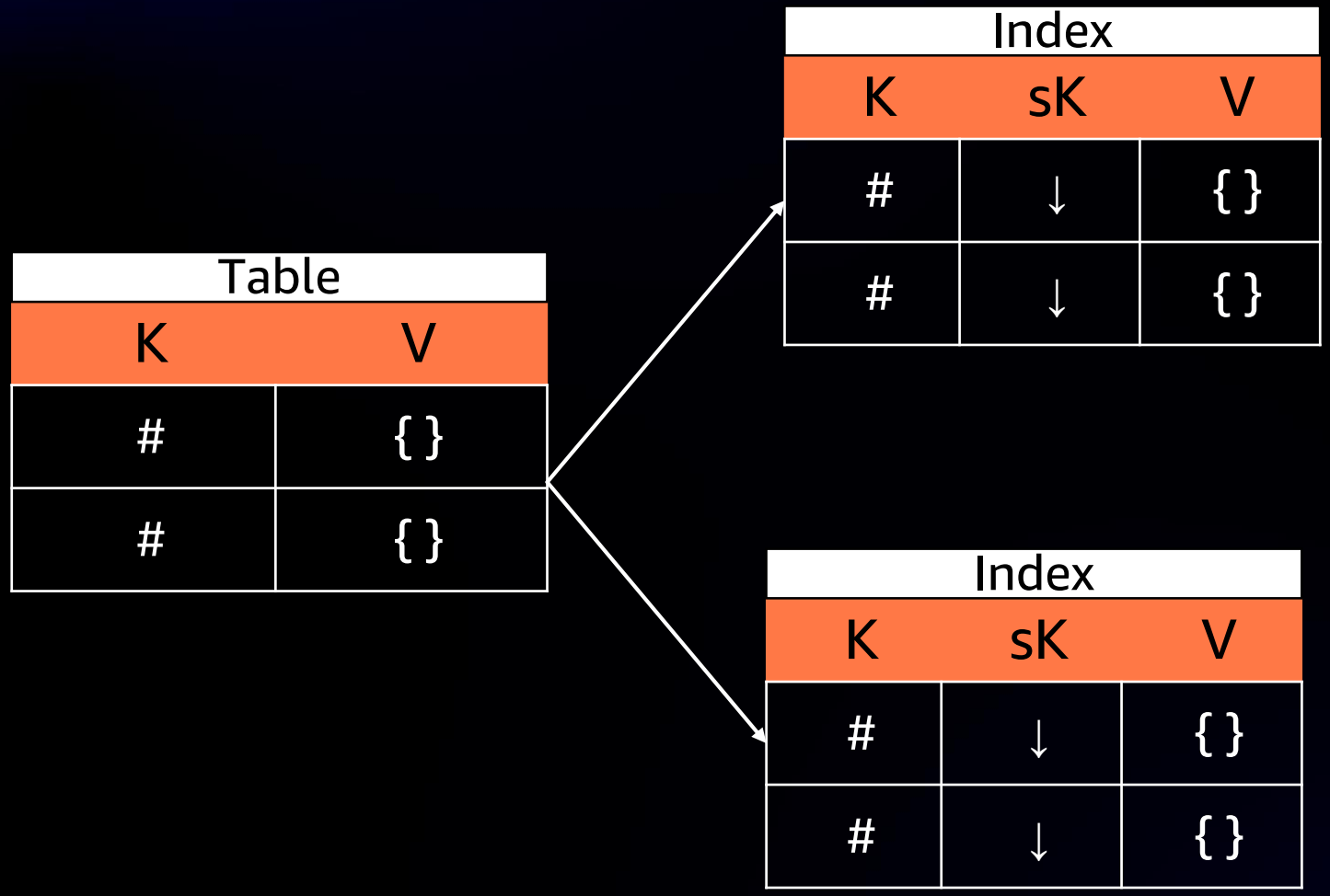
Key points

- Only allow document access by key
- Queries are carried out by enabling Document Search integration

aws

© 2021, Amazon Web Services, Inc. or its affiliates. All rights reserved.

Q-KVS – Introducing indexes

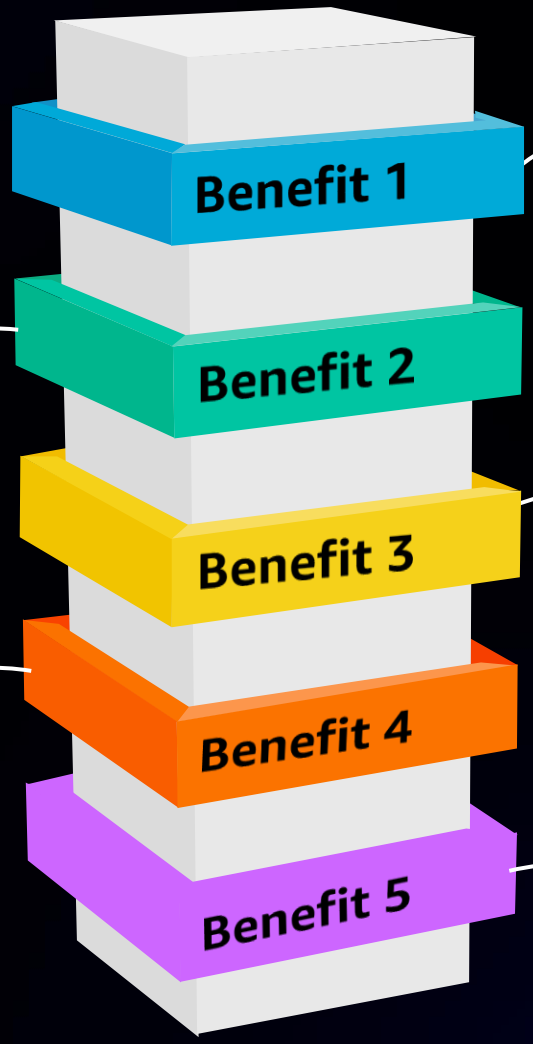


Key points

- Appends indexes, selecting a document field as key and optionally another as sort key
- Allows simple queries by key and sort keys with some operators

© 2021, Amazon Web Services, Inc. or its affiliates. All rights reserved.

Q-KVS – Benefits



Less infrastructure

Fewer points of failure

Single SDK

Familiar backend

Solved integrations

© 2021, Amazon Web Services, Inc. or its affiliates. All rights reserved.

Mercado Libre's journey

IN THIS SESSION, YOU WILL LEARN HOW WE ACCOMPLISHED THE FOLLOWING

- Organize a platform to support 24K microservices and more than 7K engineers
- Migrate 5K databases from Apache Cassandra to DynamoDB without requesting anyone in your product organization to change a single line of code or roadmap
- Migrate your relational databases to DynamoDB for better performance and scalability while keeping legacy databases working for backward compatibility
- Build an ecosystem of platform services relying on the power of DynamoDB Streams
- Use the flexibility of DynamoDB to build a platform multiservice strategy

aws

© 2021, Amazon Web Services, Inc. or its affiliates. All rights reserved.

Key takeaways

© 2021, Amazon Web Services, Inc. or its affiliates. All rights reserved.



1

Service layer

In MELI, adding an abstraction of service layer was key for flexibility and velocity

© 2021, Amazon Web Services, Inc. or its affiliates. All rights reserved.



2

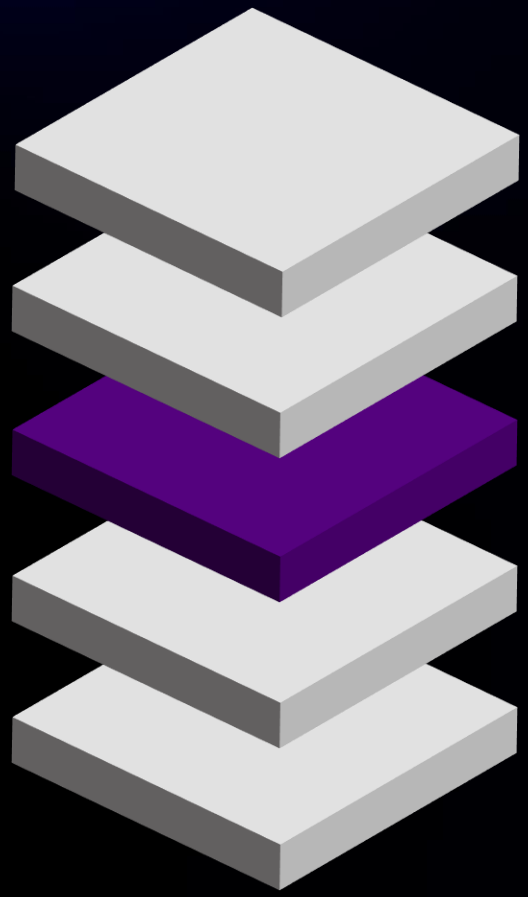
Migration to DynamoDB

Migration to DynamoDB was key in keeping up with traffic increase with high availability

Process was not trivial but added significant value

Fast evolution of DynamoDB made it possible

© 2021, Amazon Web Services, Inc. or its affiliates. All rights reserved.



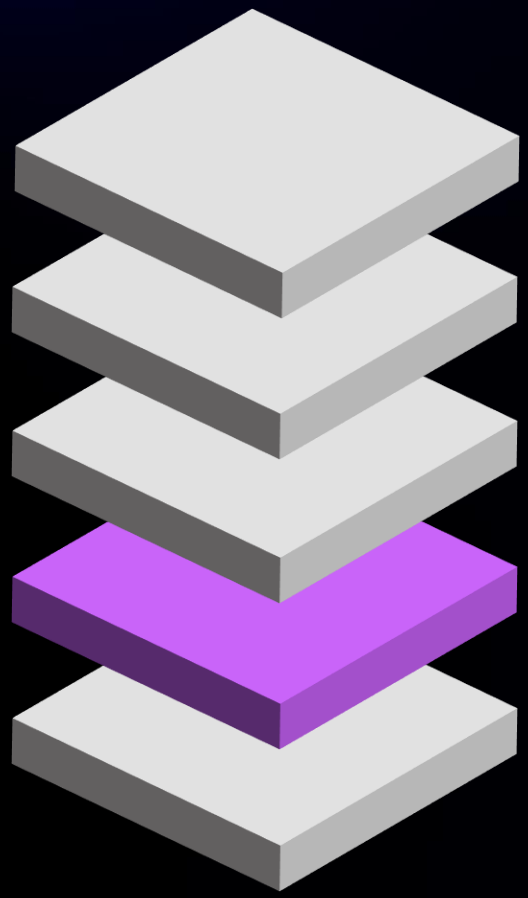
3

DynamoDB Streams & global tables

DynamoDB Streams is a great enabler for service-to-service integrations

Global tables provided the level of availability we needed to move critical databases

© 2021, Amazon Web Services, Inc. or its affiliates. All rights reserved.



4

Migration of relational DBs to DynamoDB

With the right preconditions, you can move many of your traditional relational DBs to NoSQL models and use DynamoDB

© 2021, Amazon Web Services, Inc. or its affiliates. All rights reserved.



5

Multi-flavor service on top of DynamoDB

DynamoDB flexibility allow for multiple abstracted services to be built on top of it

© 2021, Amazon Web Services, Inc. or its affiliates. All rights reserved.

Takeaways



Service Layer

DynamoDB Streams & global tables

Multi-flavor service on top of DynamoDB

Migration to DynamoDB

Migration of relational DBs to DynamoDB

© 2021, Amazon Web Services, Inc. or its affiliates. All rights reserved.

Thank you!

Oscar Mullin
oscar.mullin@mercadolibre.com

Pete Naylor
pnnaylor@amazon.com

© 2021, Amazon Web Services, Inc. or its affiliates. All rights reserved.