

The background features a dark blue gradient with large, overlapping, semi-transparent shapes in shades of purple and magenta. Two thin, light blue lines intersect diagonally across the upper right portion of the image.

AWS re:Invent

DECEMBER 2 - 6, 2024 | LAS VEGAS, NV

DAT425 - NEW

Multi-Region strong consistency with Amazon DynamoDB global tables

Jeff Duffy

Principal Product Manager
Amazon Web Services

Somu Perianayagam

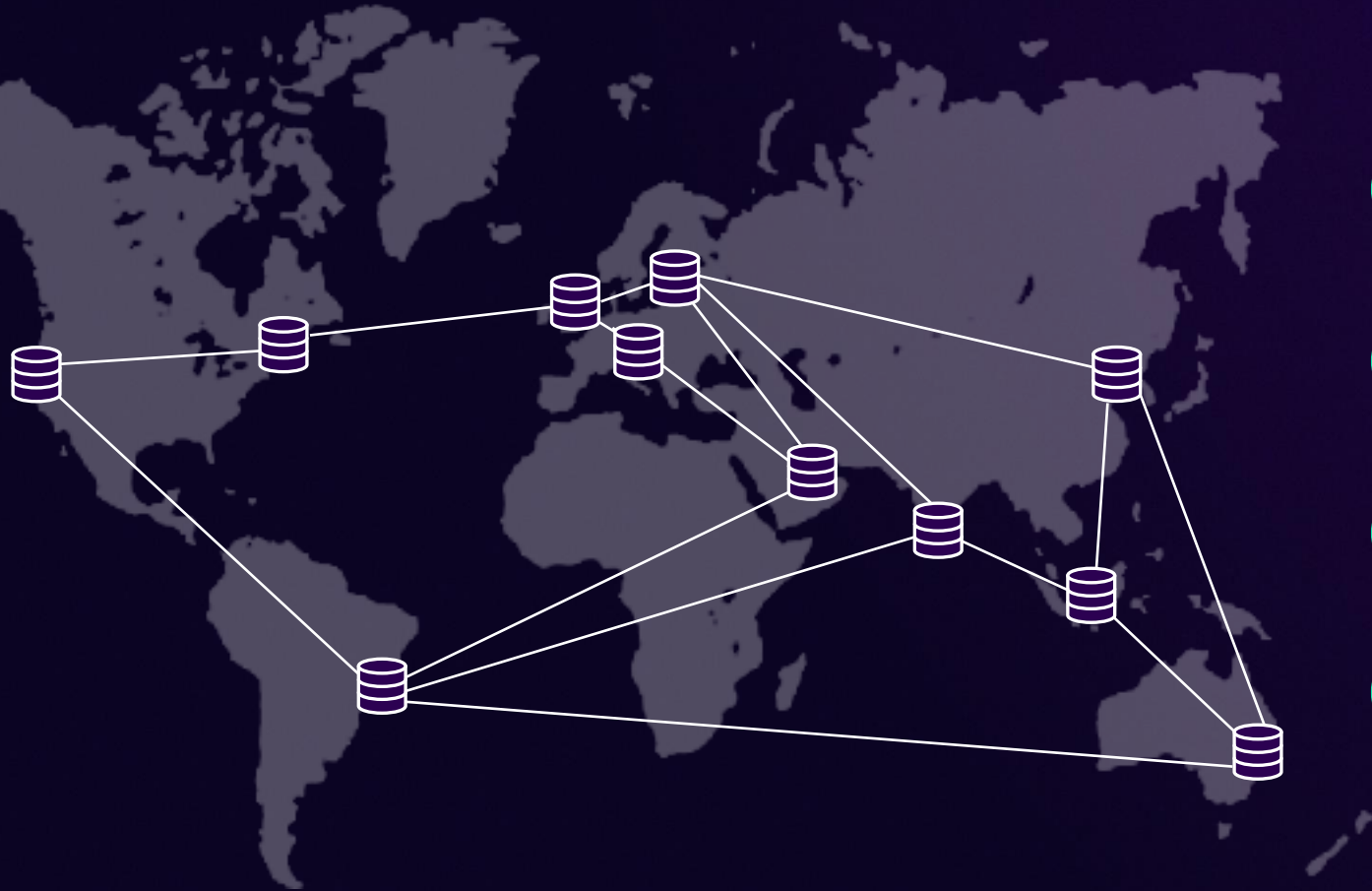
Senior Principal Engineer
Amazon Web Services



Agenda

- 01 Amazon DynamoDB consistency model
- 02 Multi-Region consistency use cases
- 03 Demo: Creating an MRSC table
- 04 How we built MRSC
- 05 Demo: How MRSC works

Amazon DynamoDB global tables



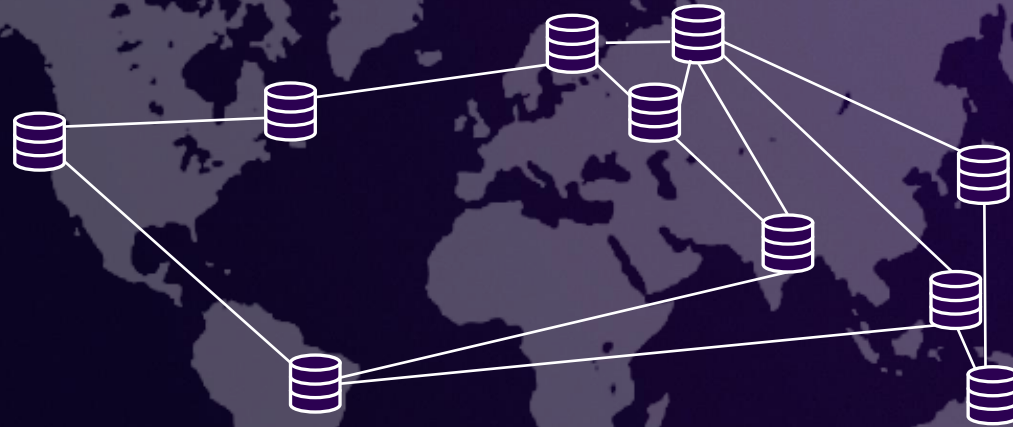
Multi-active, multi-Region

99.999% availability

Store data where it's used

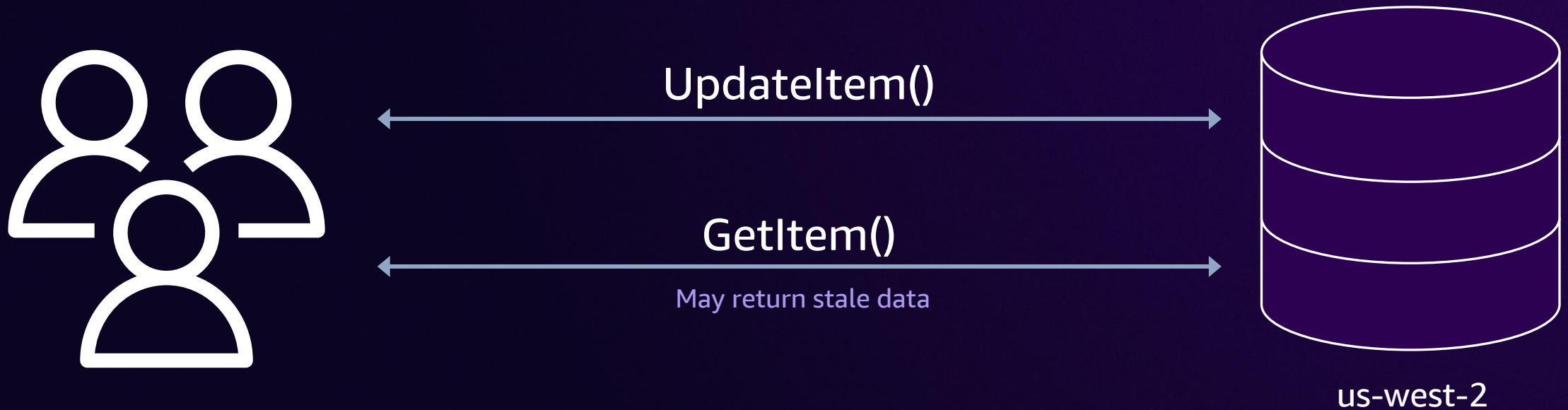
Easy to use

Multi-Region strong consistency (preview)



Build globally distributed applications with a
Recovery Point Objective (RPO) of zero

Eventual consistency – Single Region scope

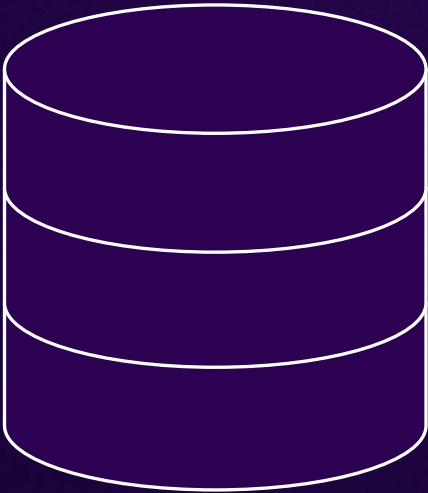
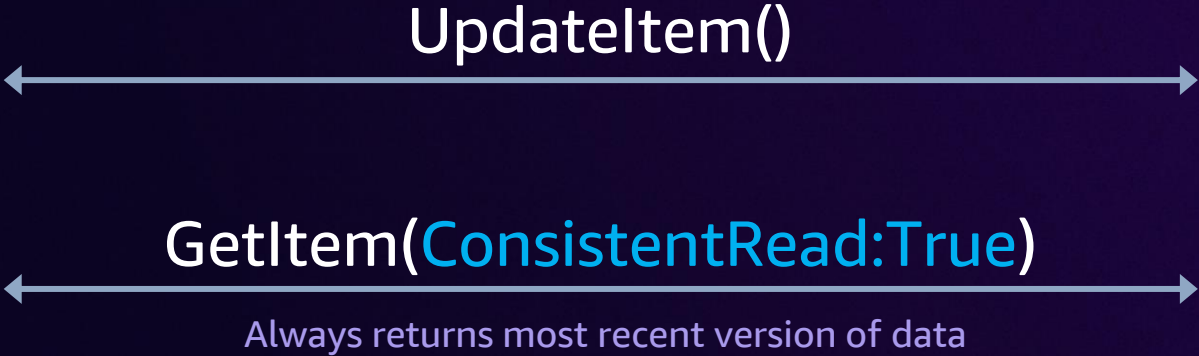


Eventually consistent read
(default)

Strong consistency – Single Region scope



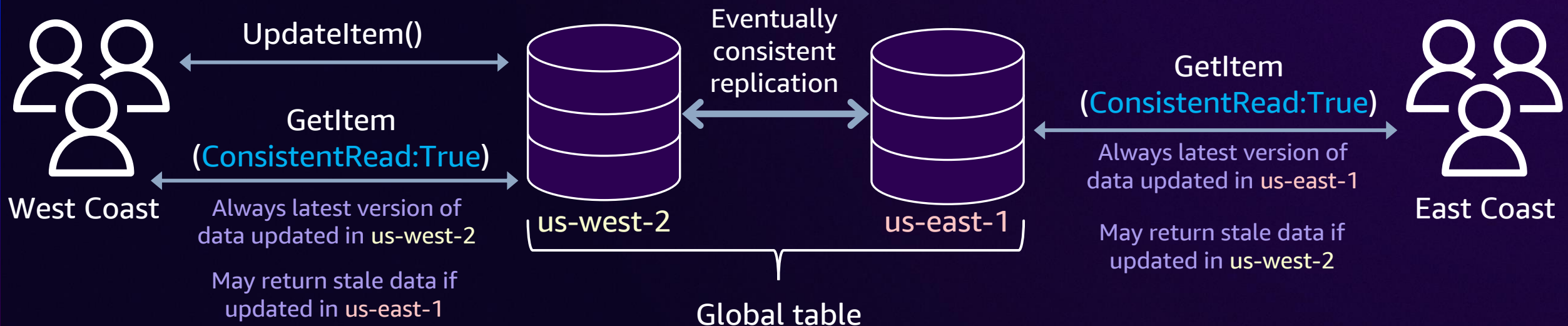
West Coast



us-west-2

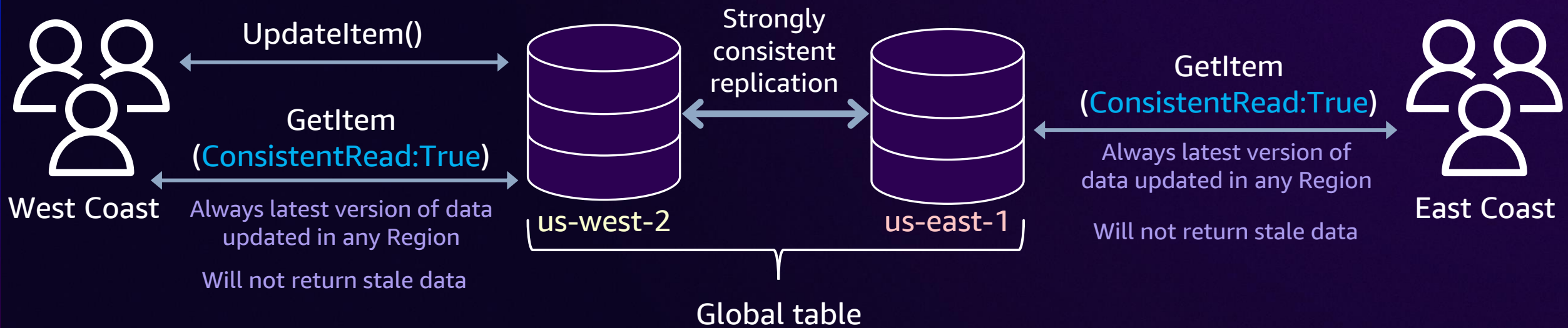
Strongly consistent read

Eventual consistency – Multi-Region scope



Multi-Region eventual consistency
(default)

Strong consistency – Multi-Region scope



Multi-Region strong consistency
(preview)

Multi-Region consistency comparison

Multi-Region eventual consistency

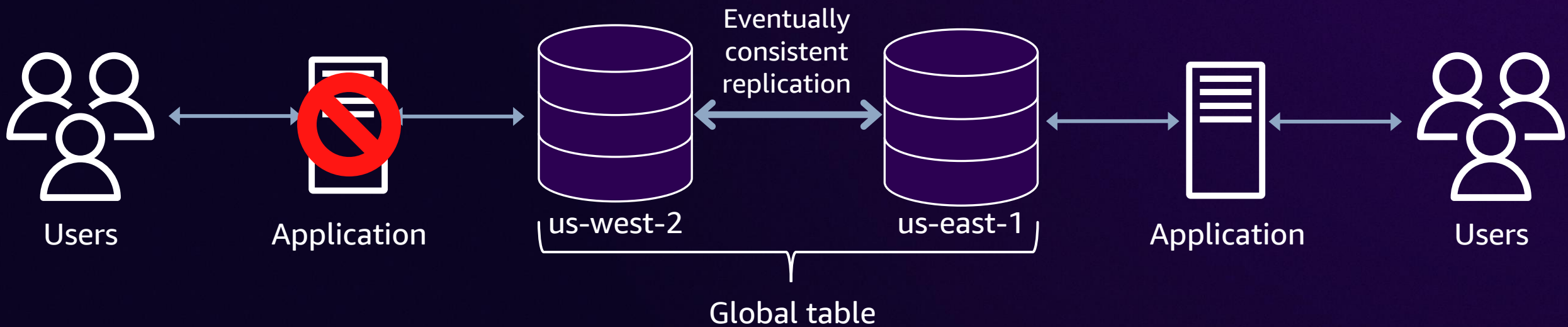
- Lower write and strongly consistent read latencies
- Strongly consistent reads can return stale data
- Conflicts are resolved with last writer wins
- Typical RPO is single-digit seconds

Multi-Region **strong** consistency

- Higher write and strongly consistent read latencies
- Strongly consistent reads never return stale data
- Conflicts return a `ReplicatedWriteConflictException`
- RPO is zero

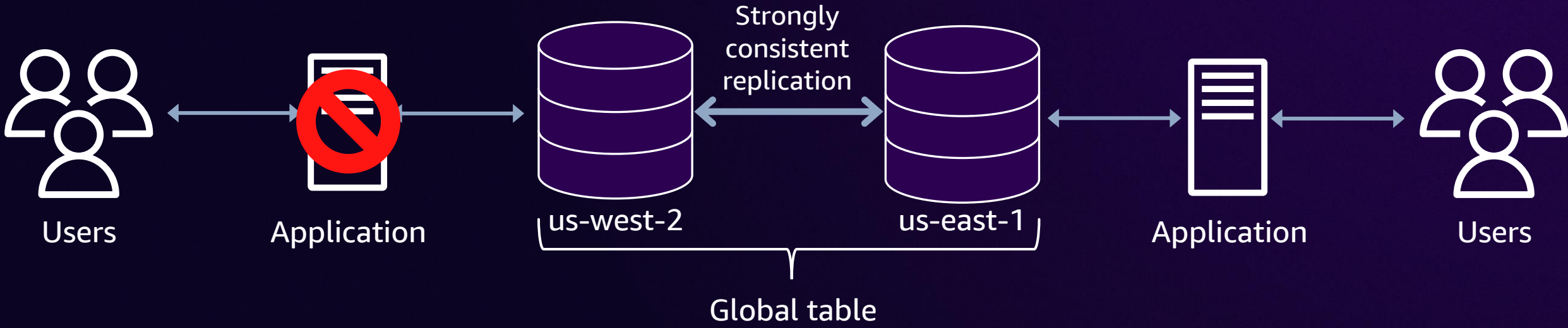
Eventually consistent reads are always eventually consistent

High availability – Eventual consistency



Recovery Point Objective (RPO) is typically single-digit seconds

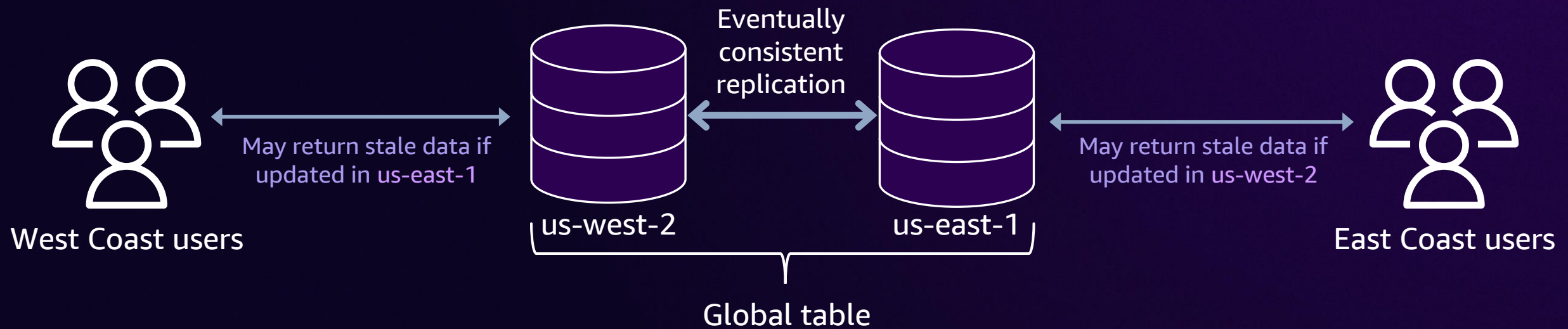
High availability – Strong consistency



Recovery Point Objective (RPO) is zero

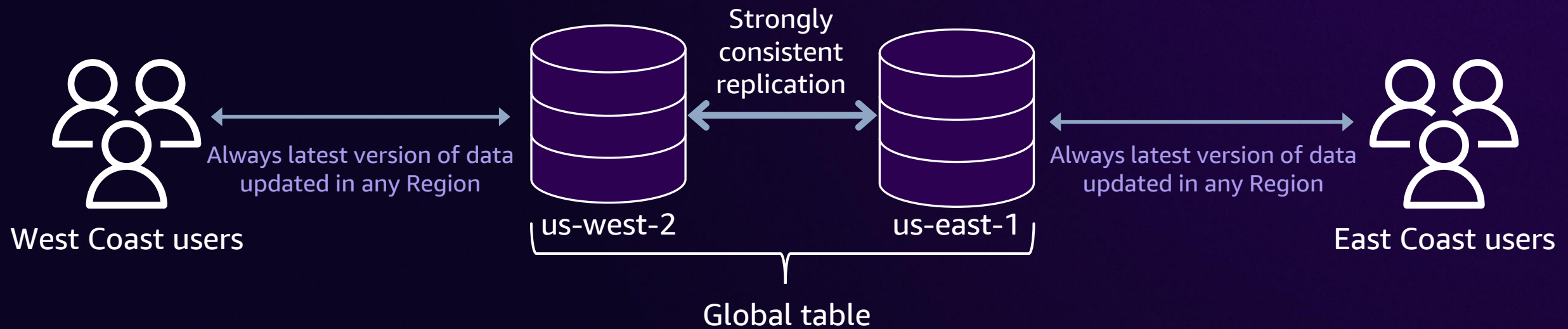


Data locality – Eventual consistency



Eventually consistent reads have lower latency, but data may be stale

Data locality – Strong consistency

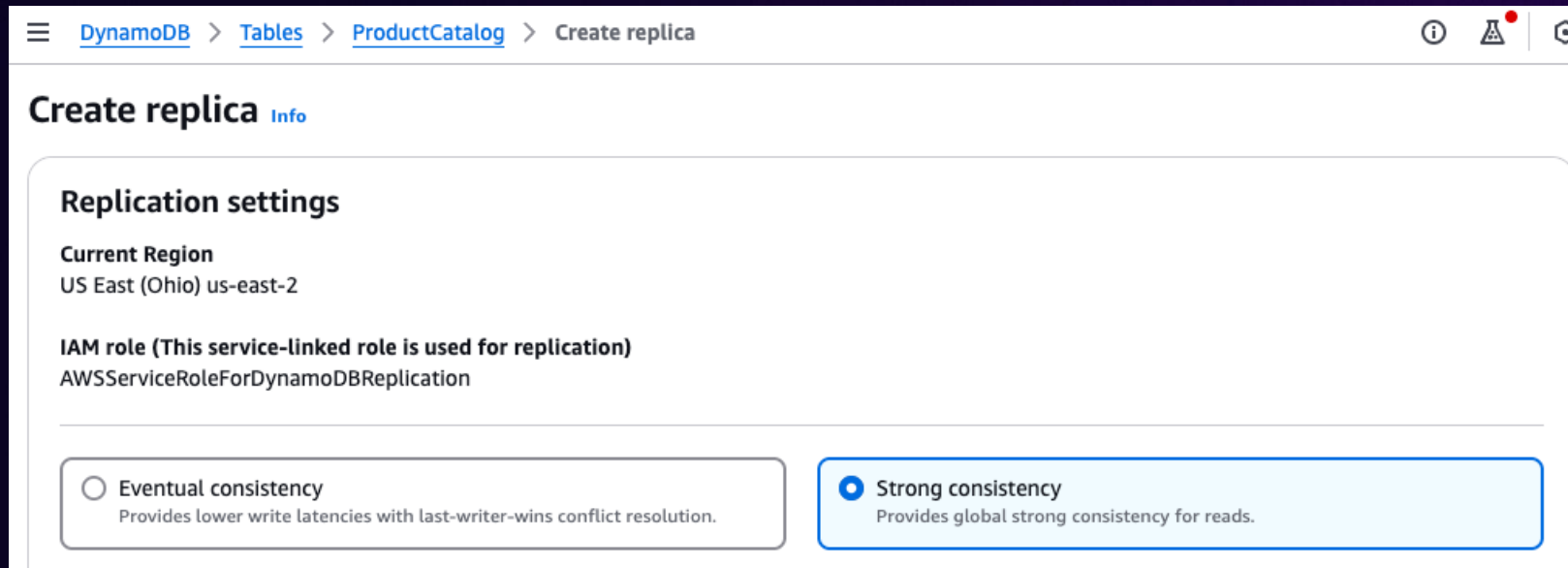


Strongly consistent reads have higher latency, but data is **always consistent**

Multi-Region strong consistency considerations

- This feature is in public preview – **not** ready for production
- Replication consistency applies to the entire global table
- You cannot switch replication consistency after creation
- Preview requires three-Region deployment

Demo: Creating an MRSC table



The screenshot shows the AWS Management Console interface for creating a replica of a DynamoDB table. The breadcrumb navigation at the top reads: [DynamoDB](#) > [Tables](#) > [ProductCatalog](#) > **Create replica**. On the right side of the header, there are icons for help, a warning (a triangle with a red dot), and a home button.

Create replica [Info](#)

Replication settings

Current Region
US East (Ohio) us-east-2

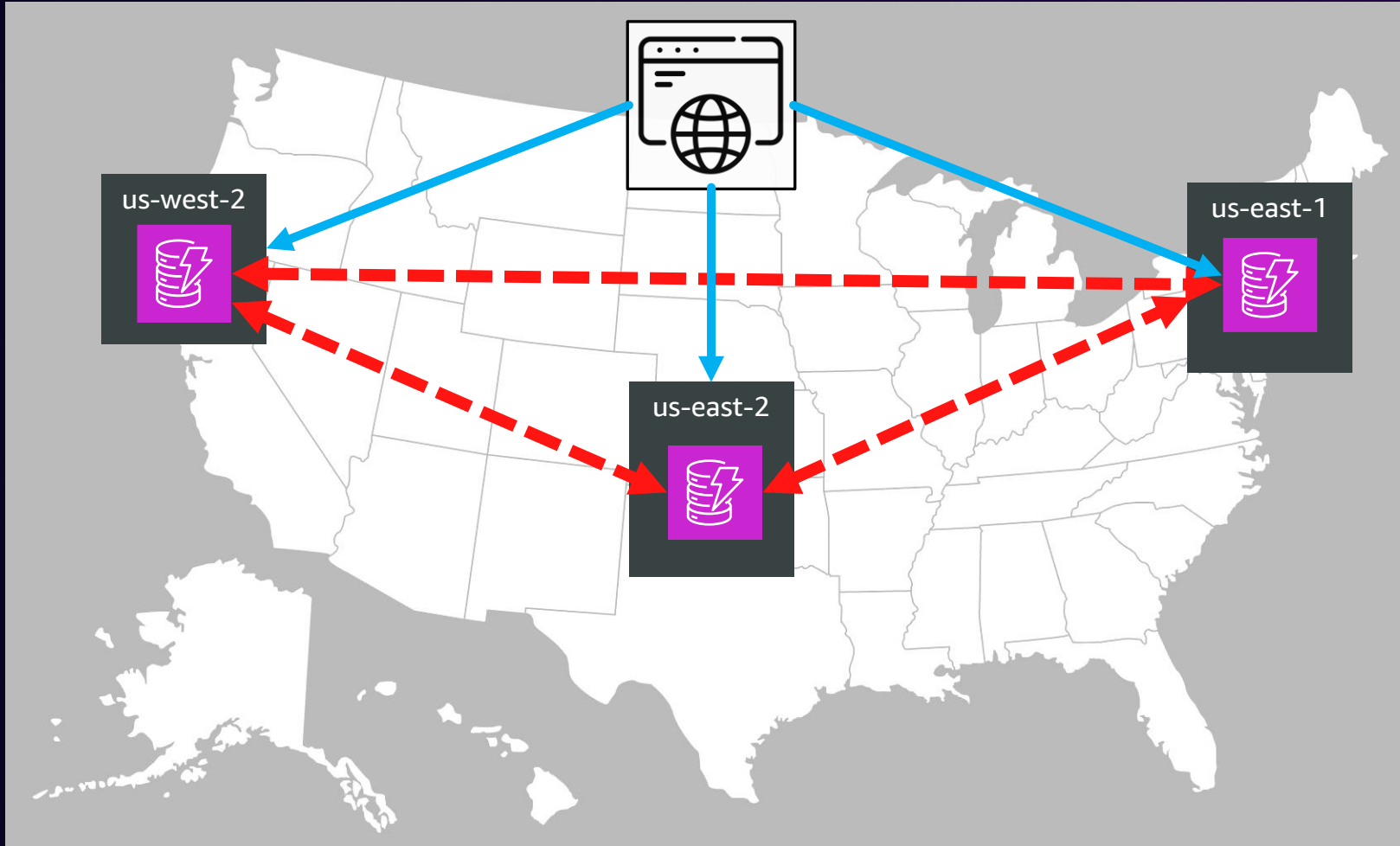
IAM role (This service-linked role is used for replication)
AWSServiceRoleForDynamoDBReplication

Eventual consistency
Provides lower write latencies with last-writer-wins conflict resolution.

Strong consistency
Provides global strong consistency for reads.

A close look at global tables

Demo



Demo

eventName	year	count
re:Invent	2024	0

Demo

```
//eventual consistency or  
Asynchronous
```

```
"TableName": "async-gt" "Replicas":  
[  
  {"RegionName": "us-east-2",  
   "ReplicaStatus": "ACTIVE"},  
  {"RegionName": "us-east-1",  
   "ReplicaStatus": "ACTIVE"},  
  {"RegionName": "us-west-2",  
   "ReplicaStatus": "ACTIVE"}  
],
```

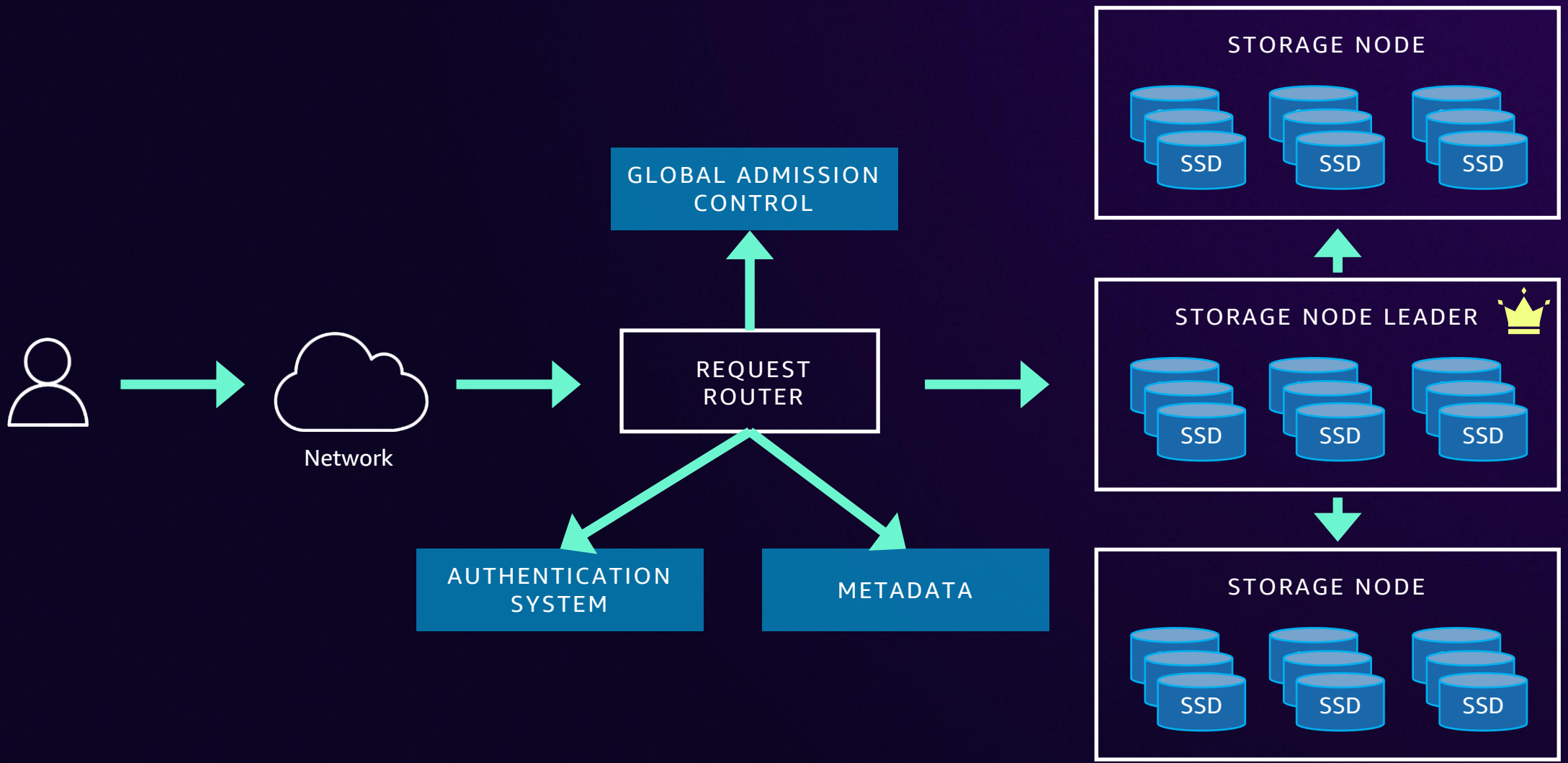
```
// multi-region Strong Consistency
```

```
"TableName": "mrsc-gt"  
"MultiRegionConsistency": "STRONG"  
"Replicas": [  
  {"RegionName": "us-east-2",  
   "ReplicaStatus": "ACTIVE"},  
  {"RegionName": "us-east-1",  
   "ReplicaStatus": "ACTIVE"},  
  {"RegionName": "us-west-2",  
   "ReplicaStatus": "ACTIVE"}  
],
```

Demo

Multi-writer
Strong read-after-write consistency

DynamoDB primer



The log is the database



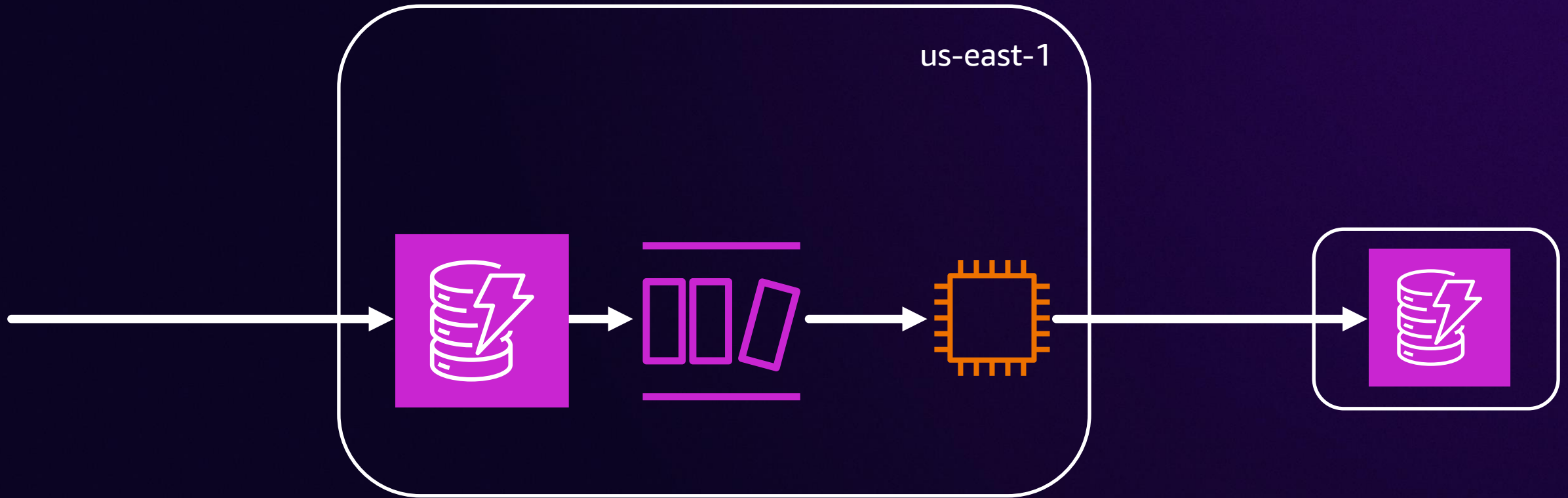
Backup restore



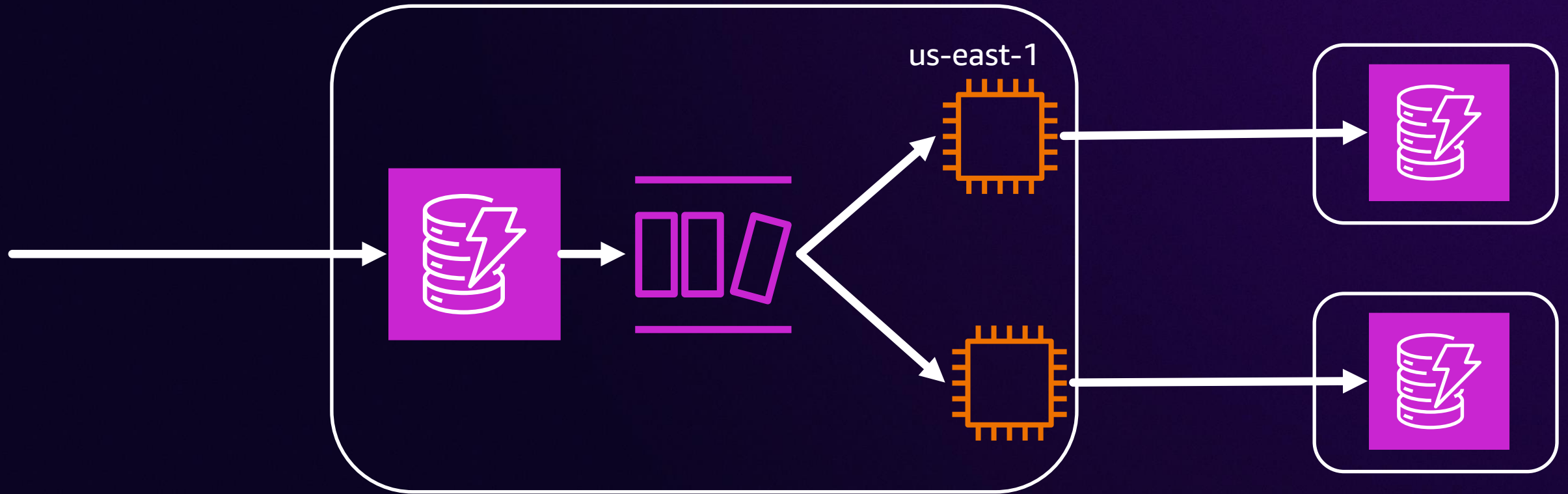
DynamoDB Streams



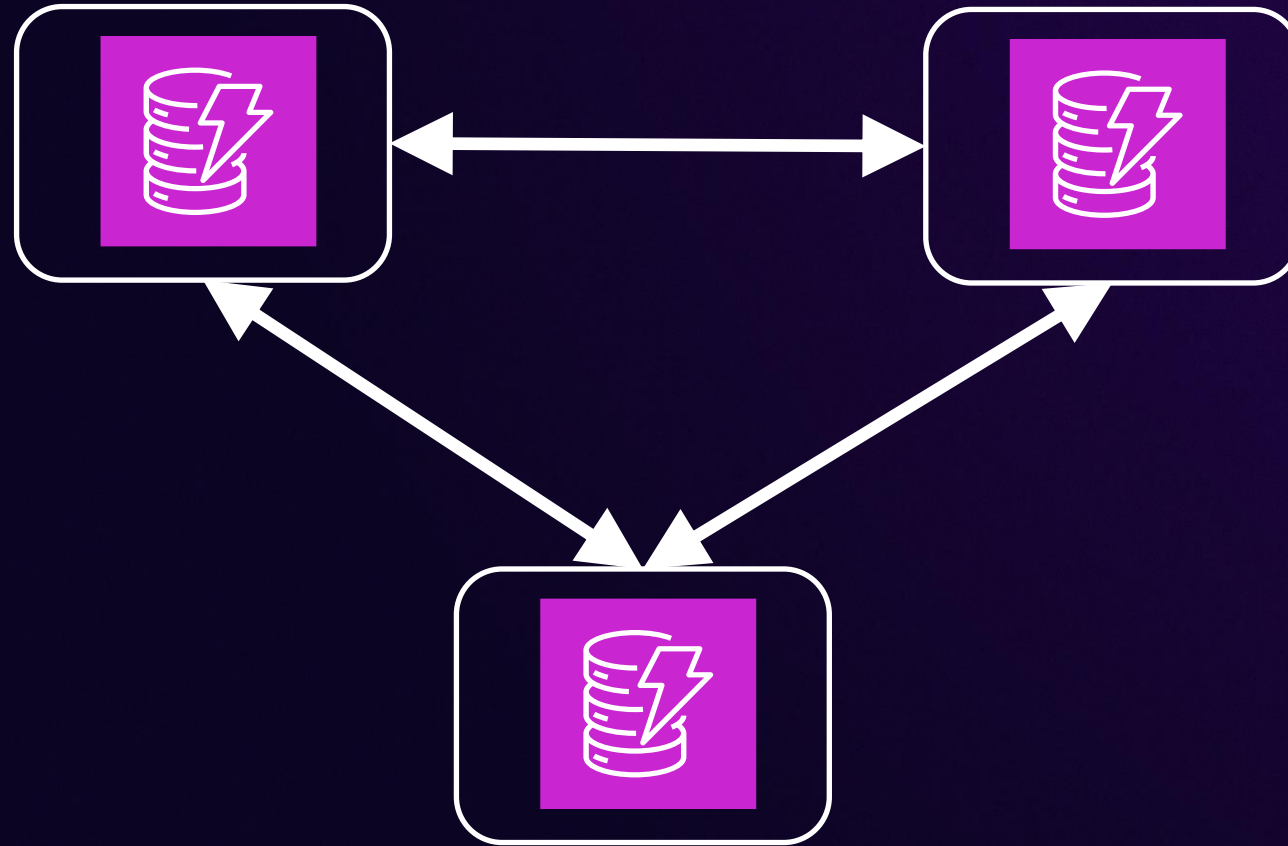
Asynchronous global tables – Replication engine



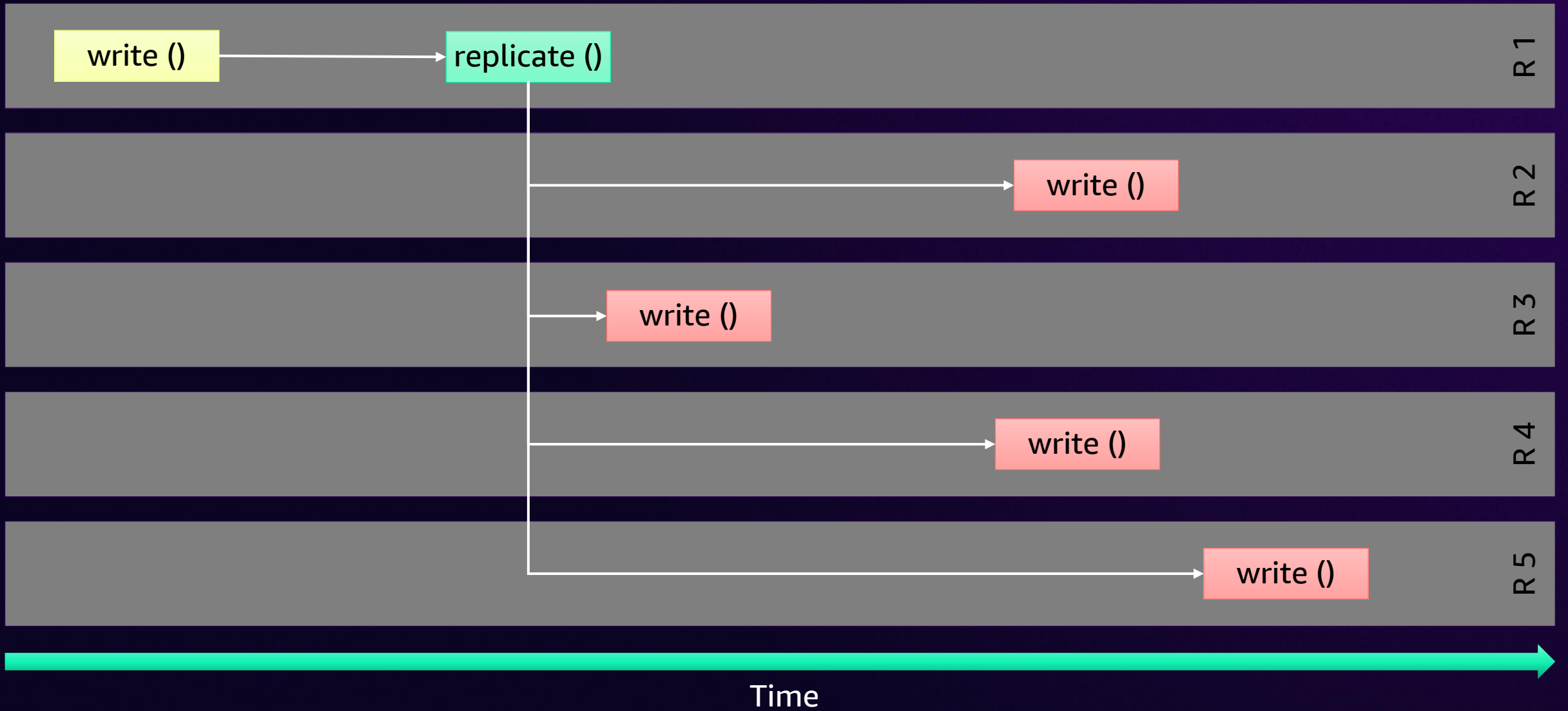
Asynchronous global tables – N Regions



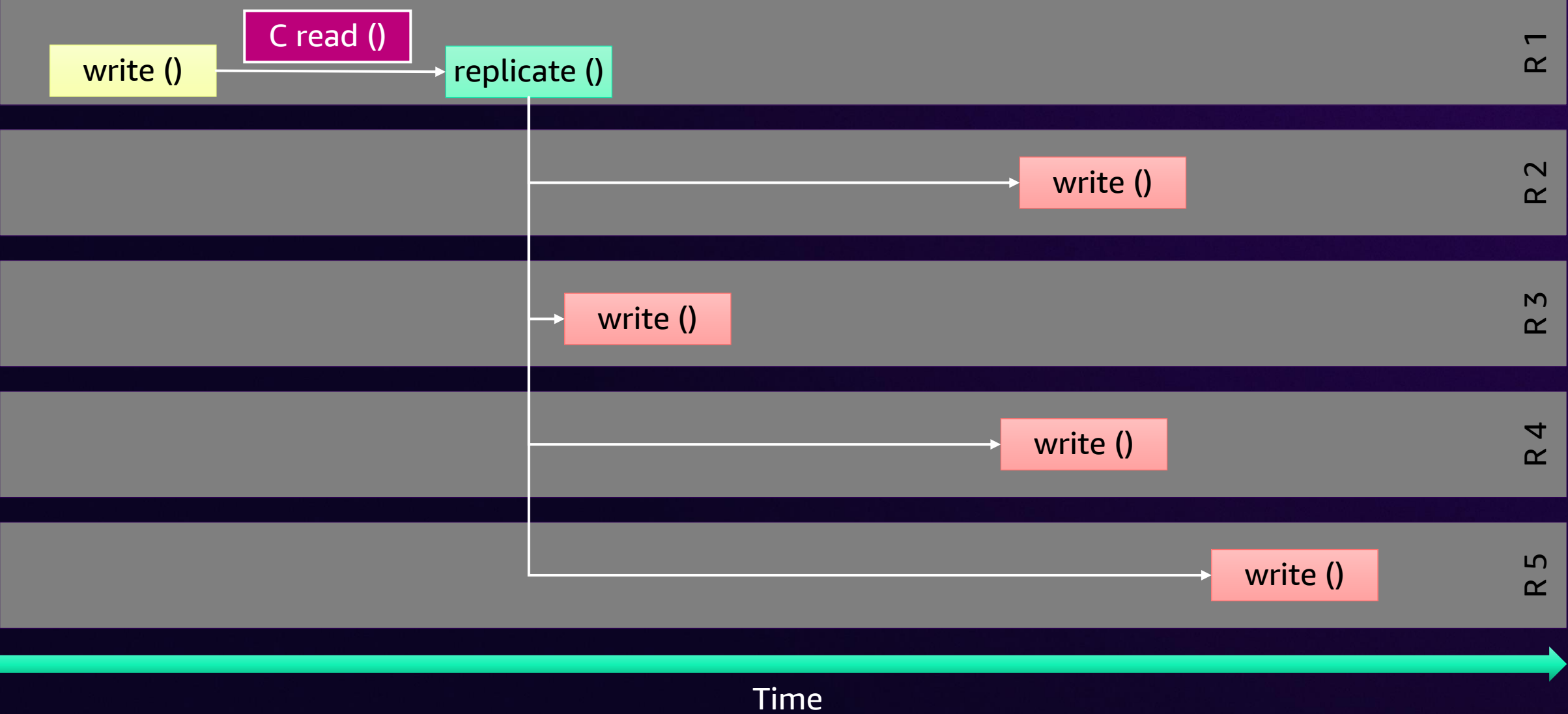
Asynchronous global tables – N Regions



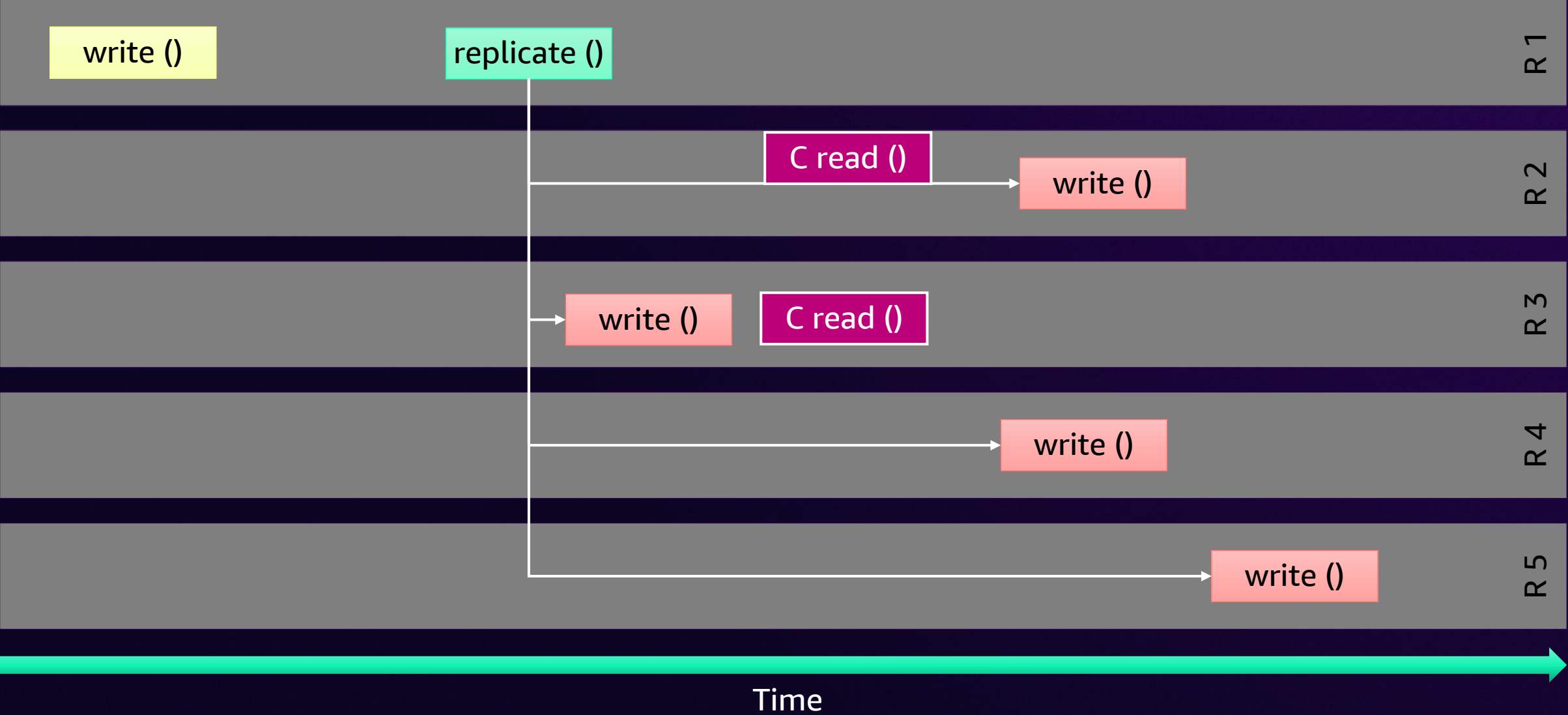
Asynchronous global tables



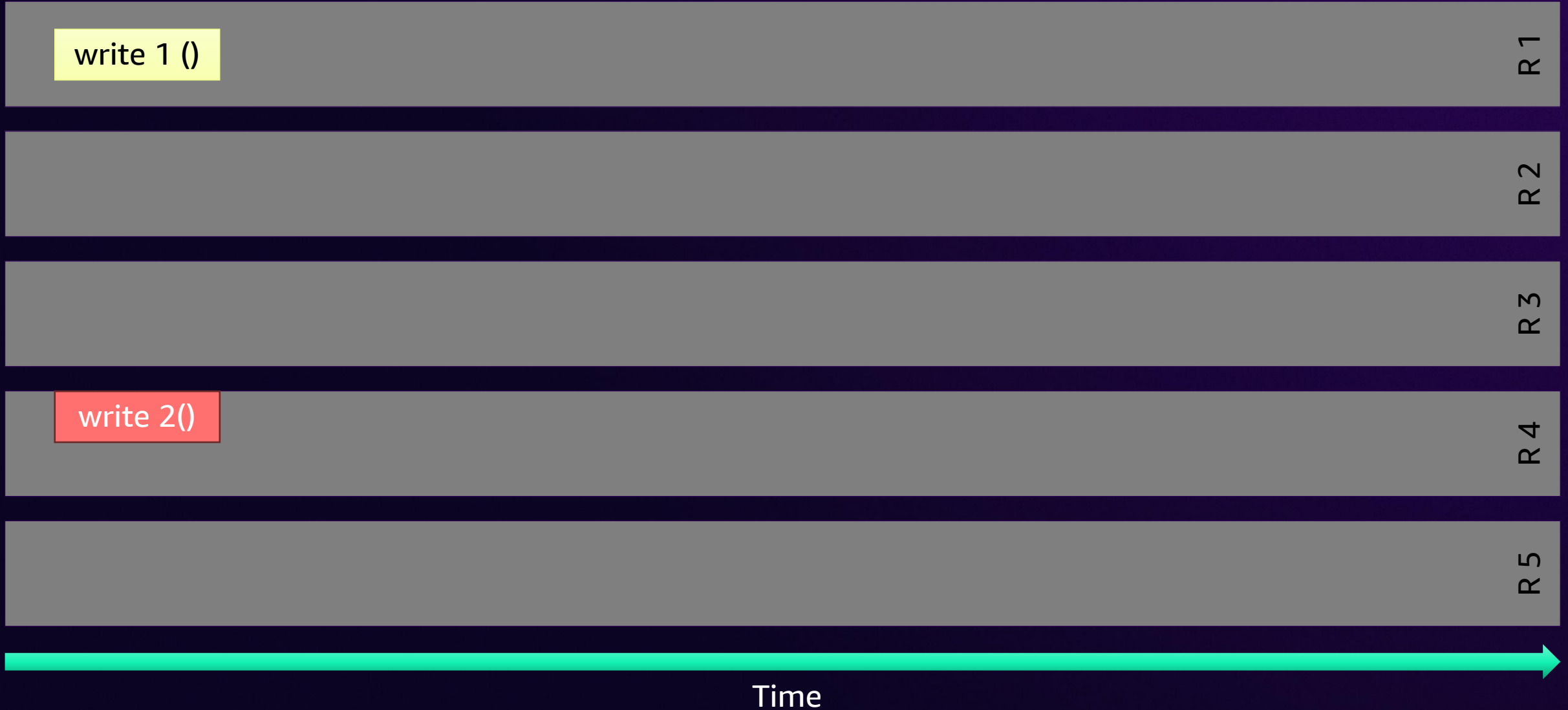
Asynchronous global tables – Strongly consistent read



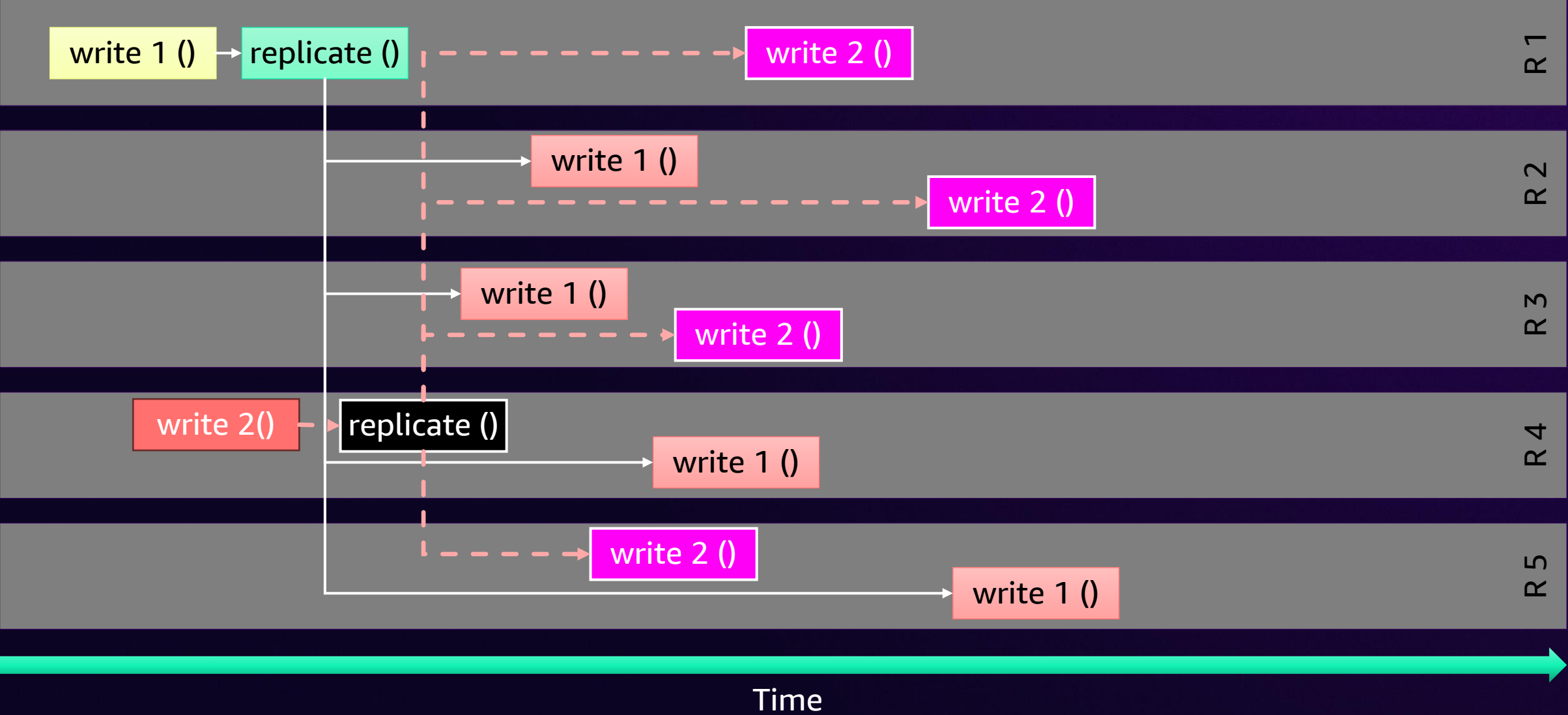
Asynchronous global tables – Strongly consistent read



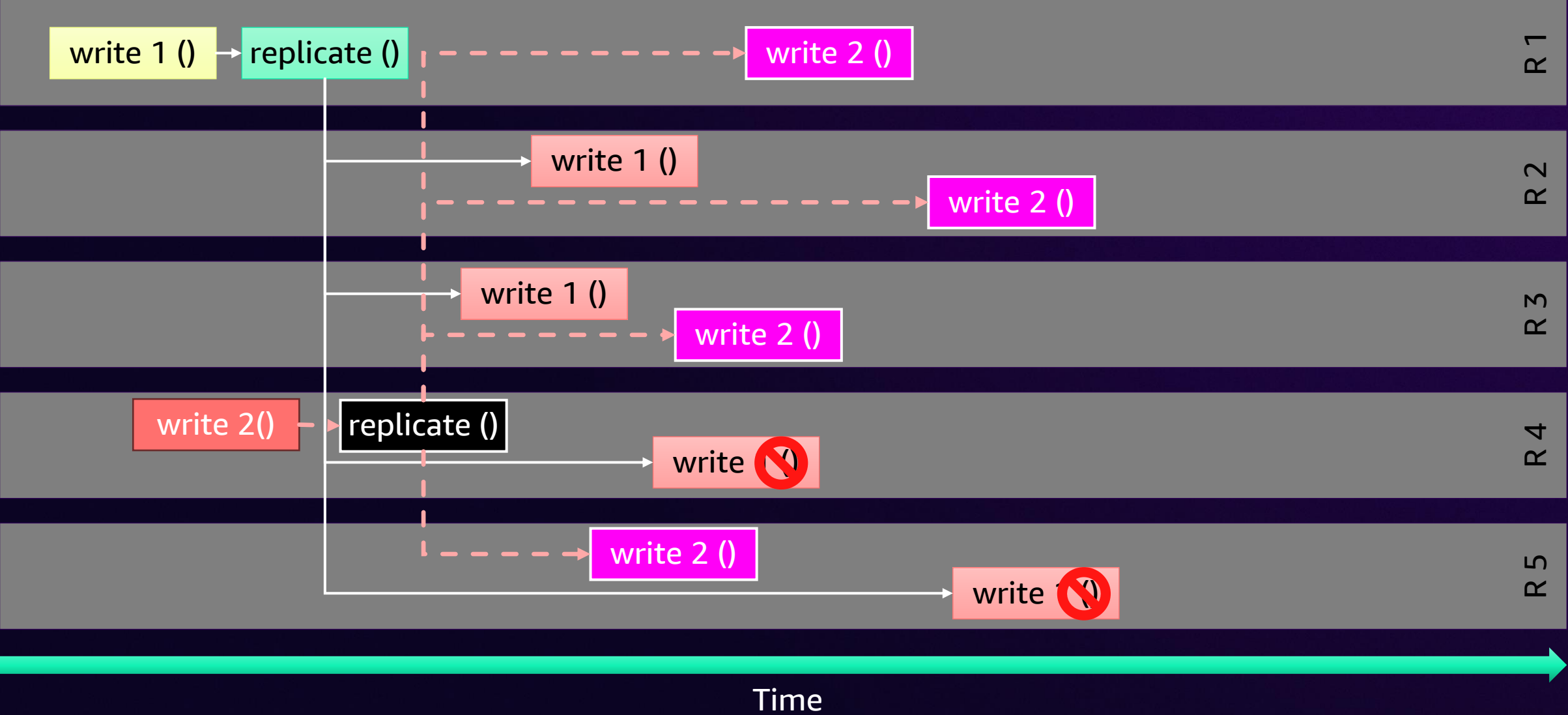
Asynchronous global tables (write-write conflict)



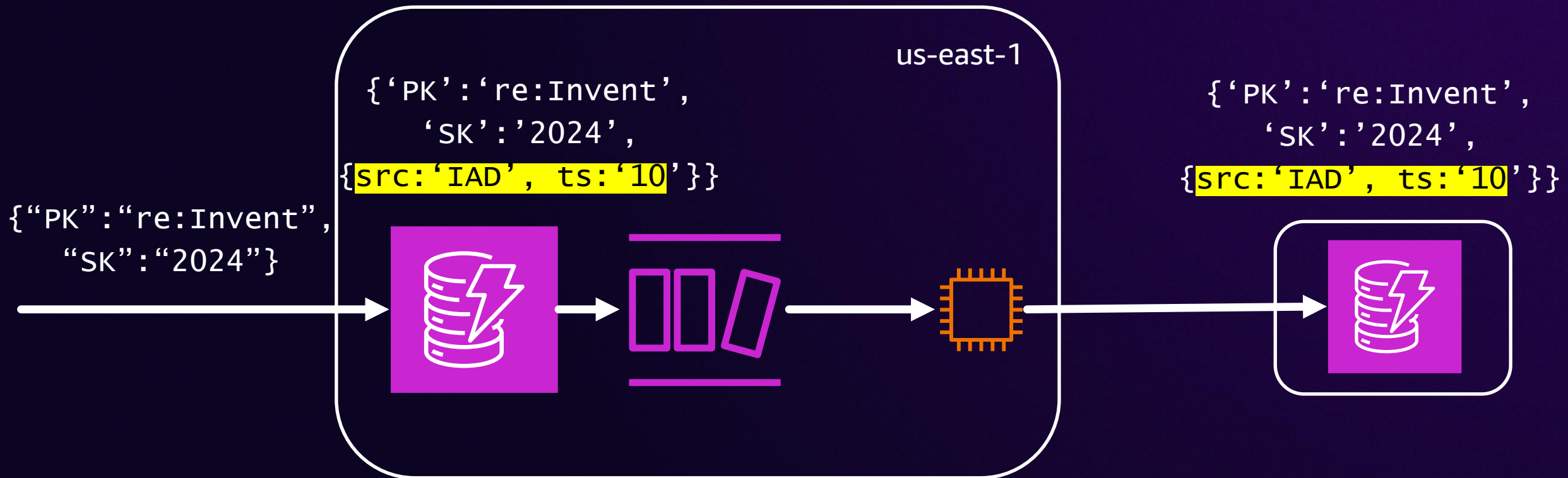
Asynchronous global tables (write-write conflict)



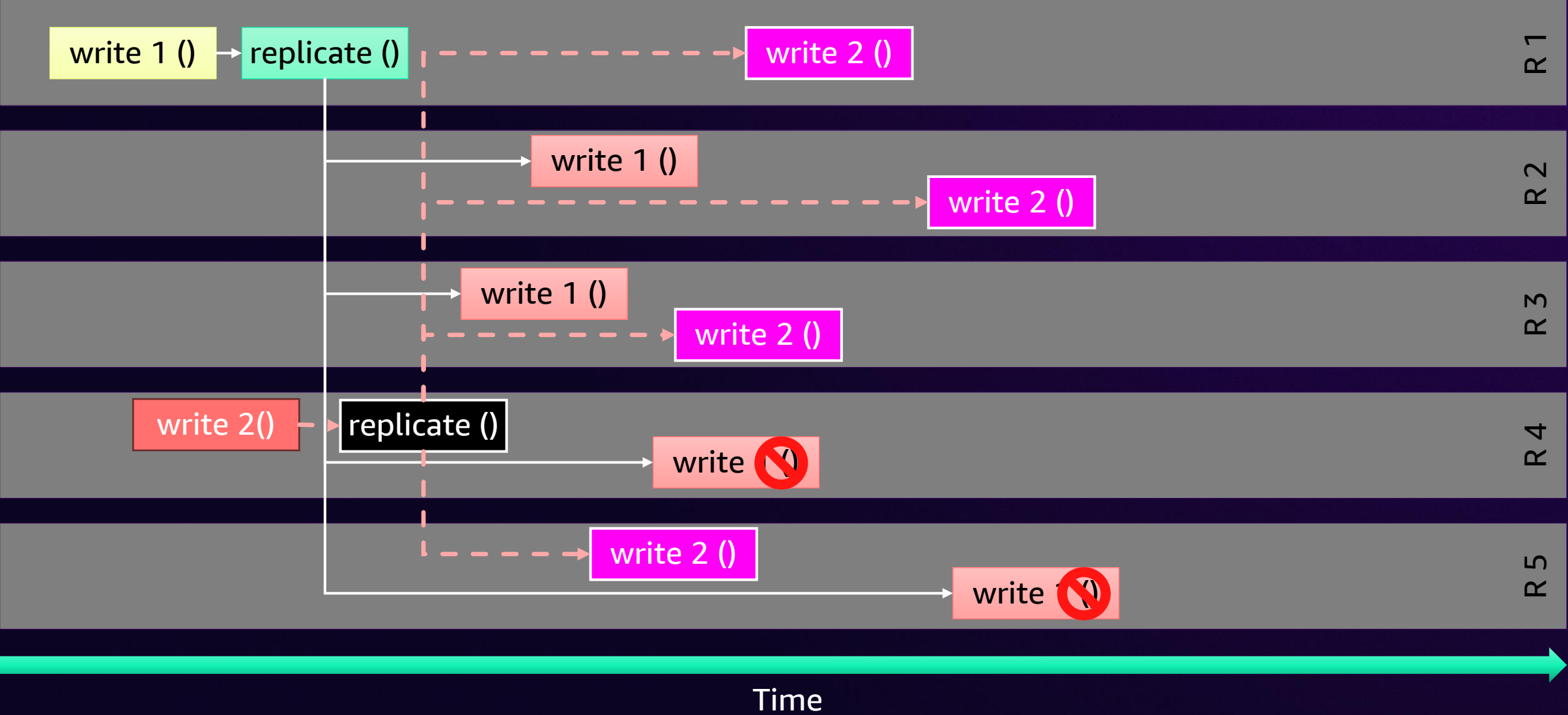
Asynchronous global tables (last writer wins)



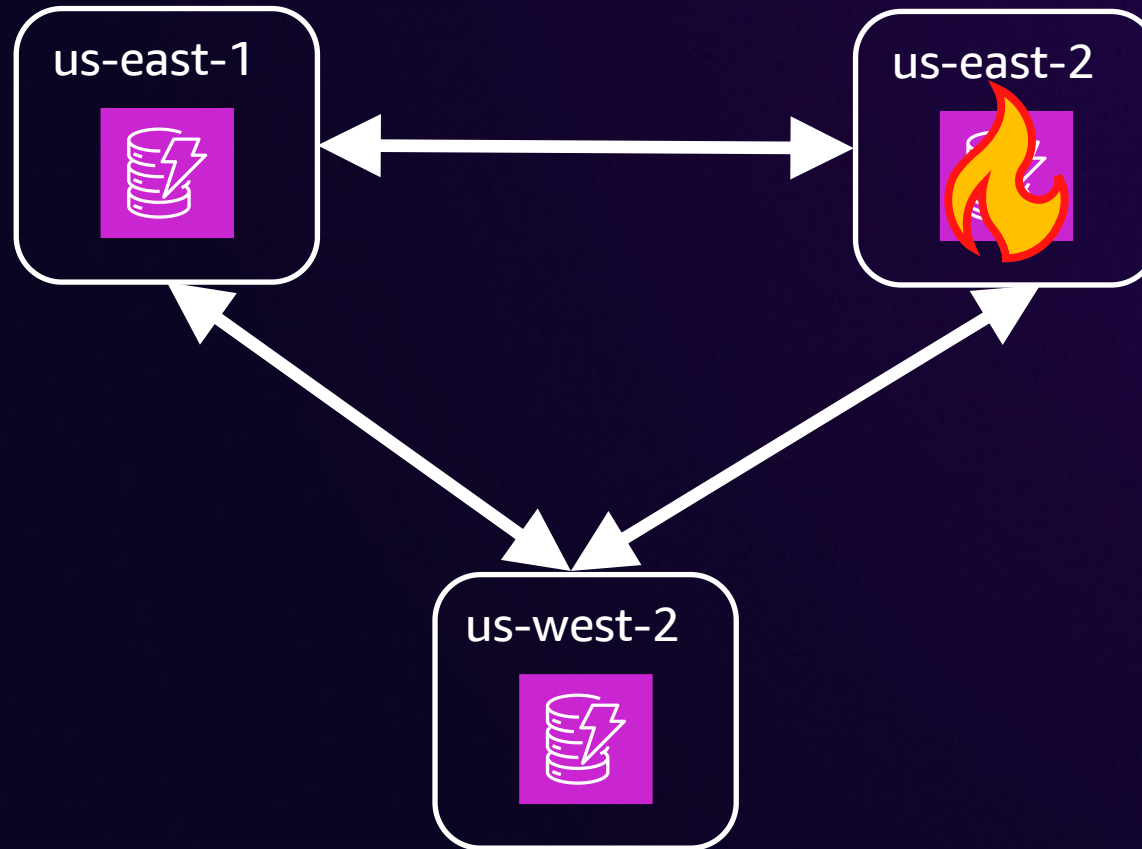
Asynchronous global tables – System metadata



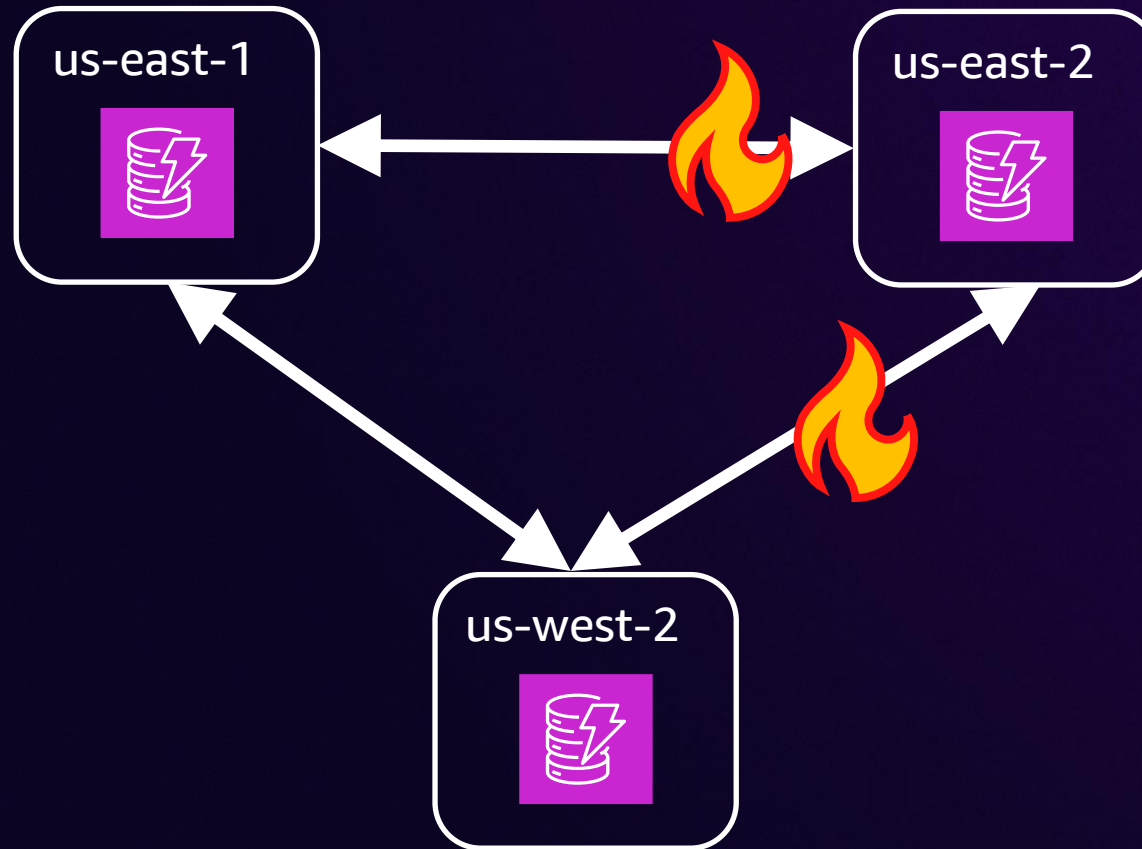
Asynchronous global tables (last writer wins)



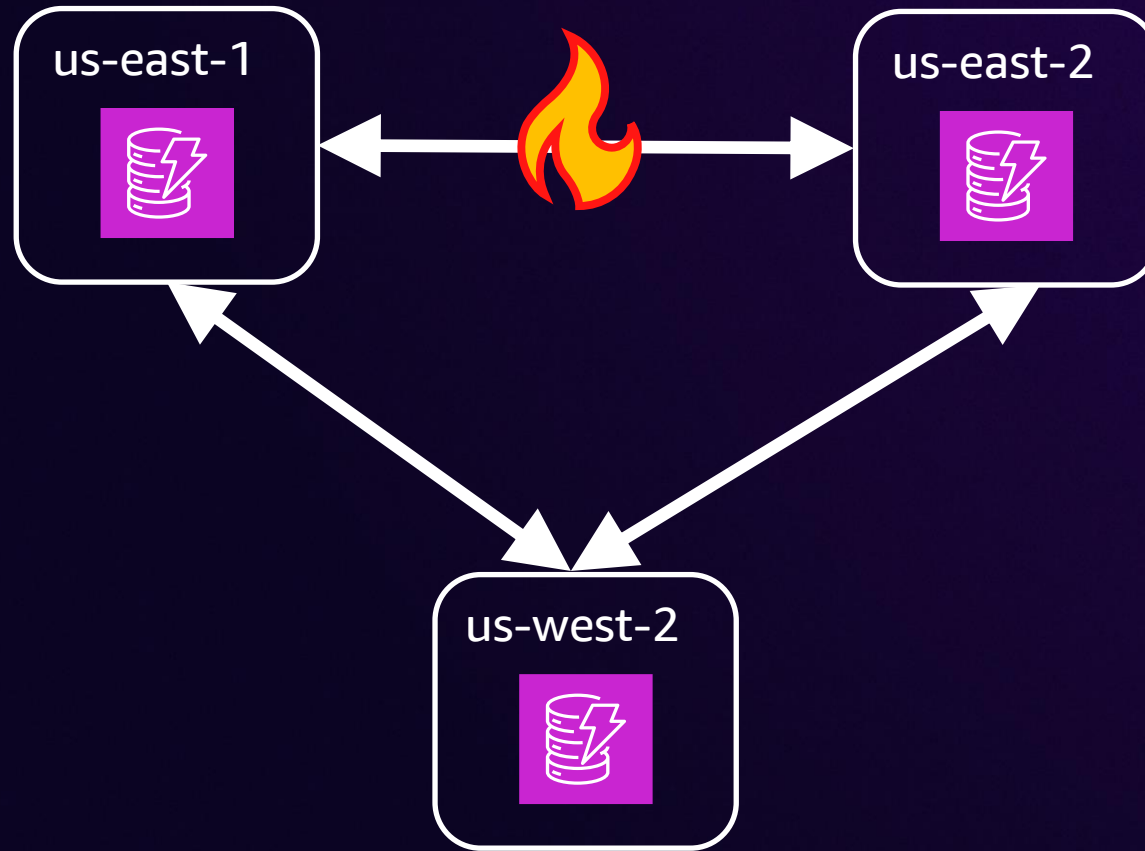
Asynchronous global tables – Region failure



Asynchronous global tables – Region isolation



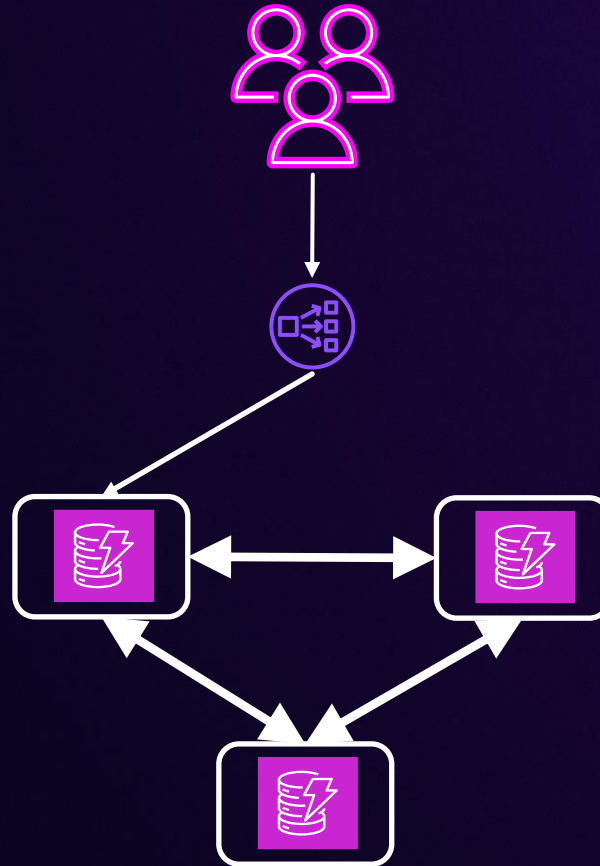
Asynchronous global tables – Network partition



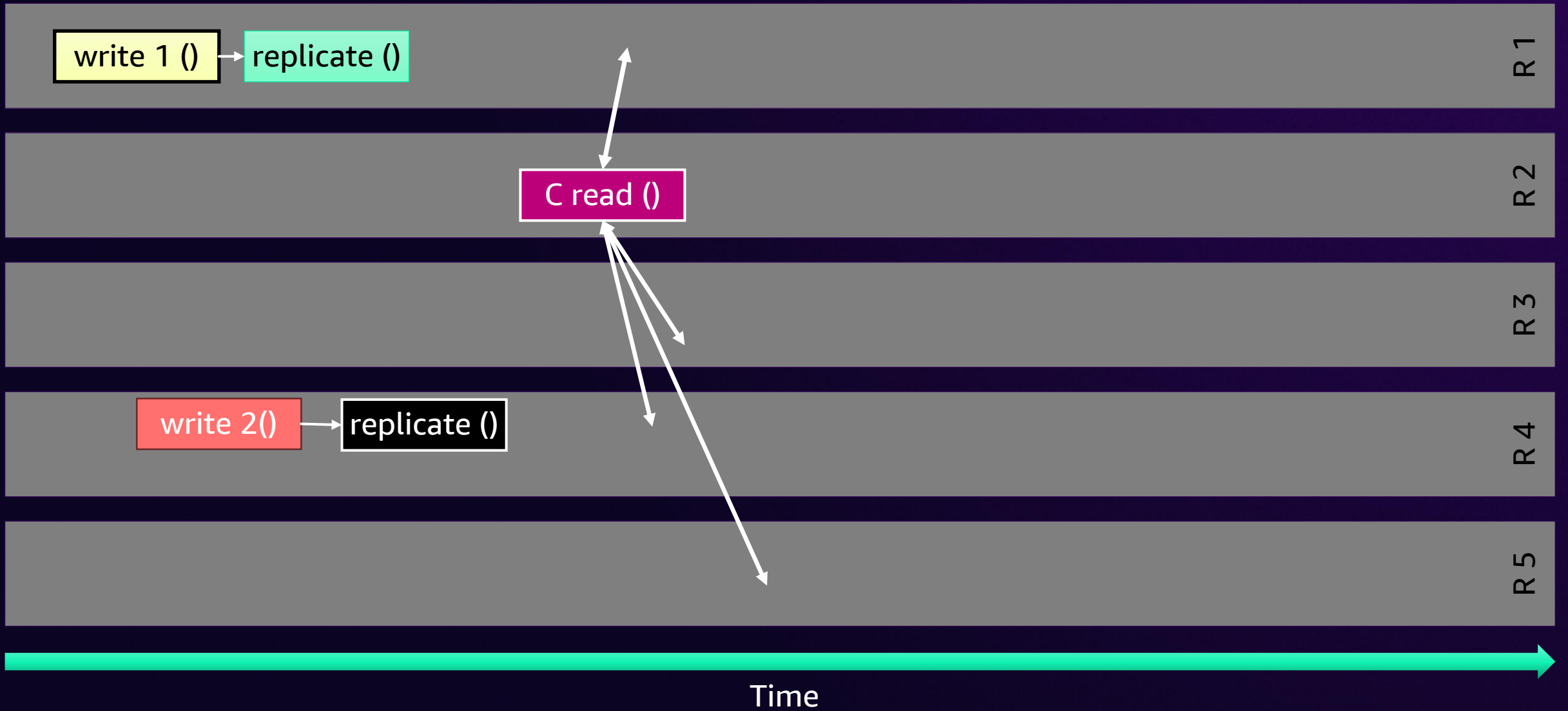
Read your writes from **any** Region



Single reader/writer Region



Quorum read



Multi-Region strong consistency global tables

Read your writes from **any** Region

Multi-Region strong consistency global tables

- Writes are highly available
- Regional failures
- SC read serializable with writes

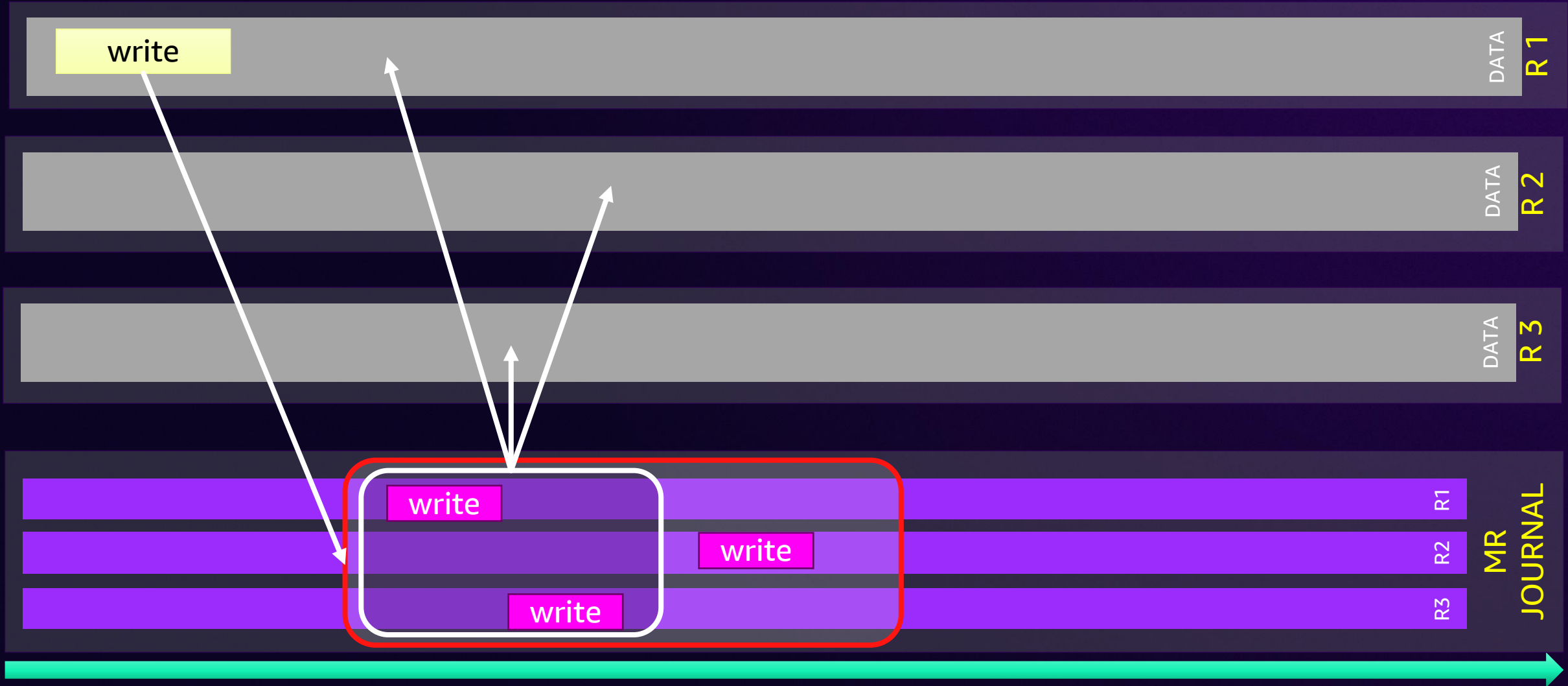
Multi-Region strong consistency global tables



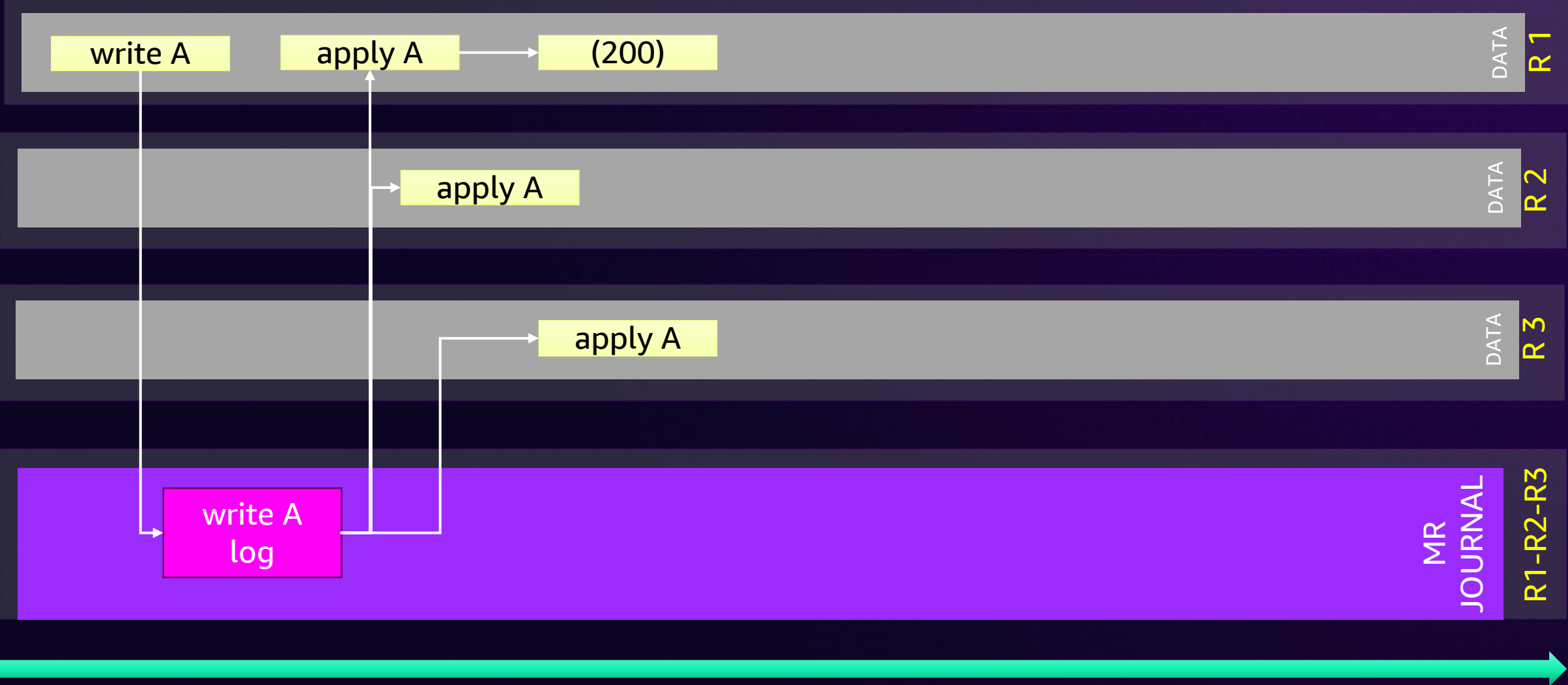
Multi-Region strong consistency global tables



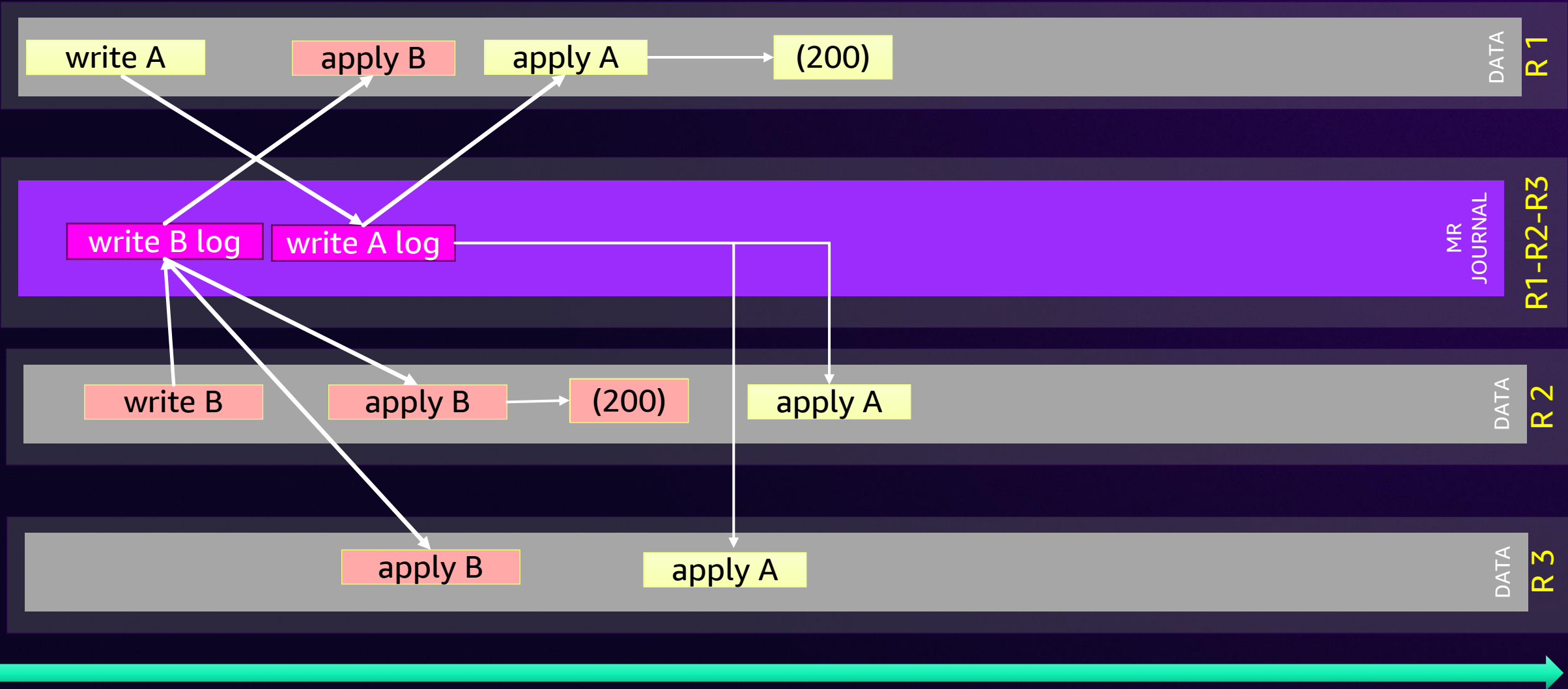
Multi-Region strong consistency global tables



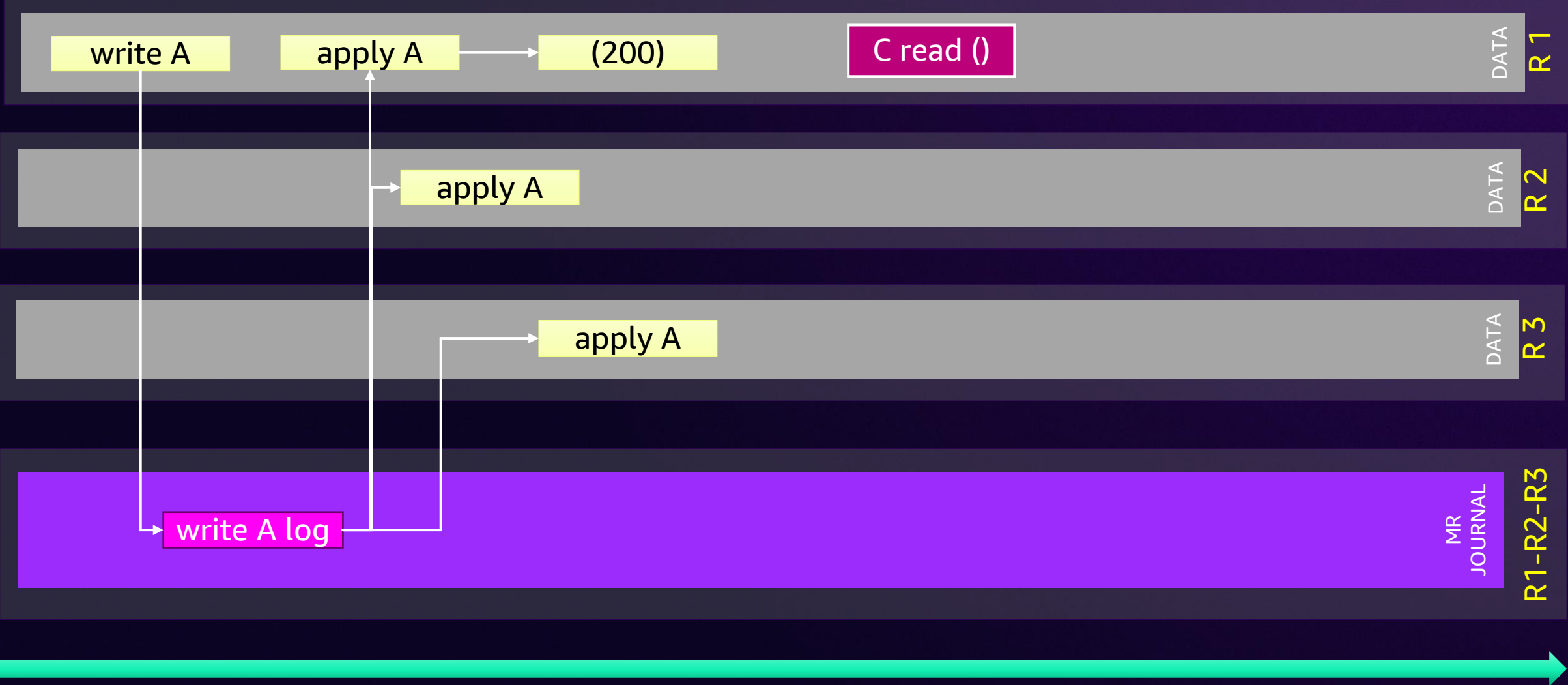
Multi-Region strong consistency global tables



Multi-Region strong consistency global tables



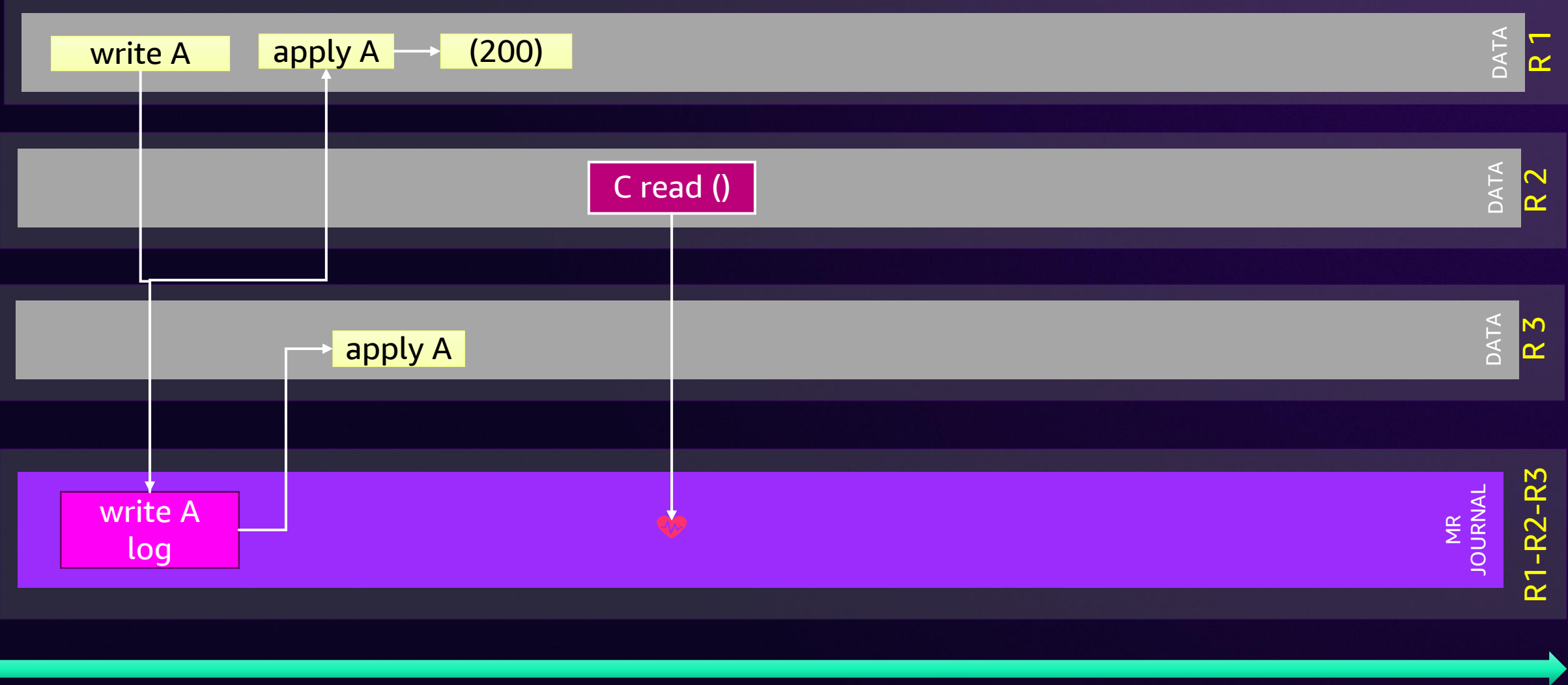
Multi-Region strong consistency global tables



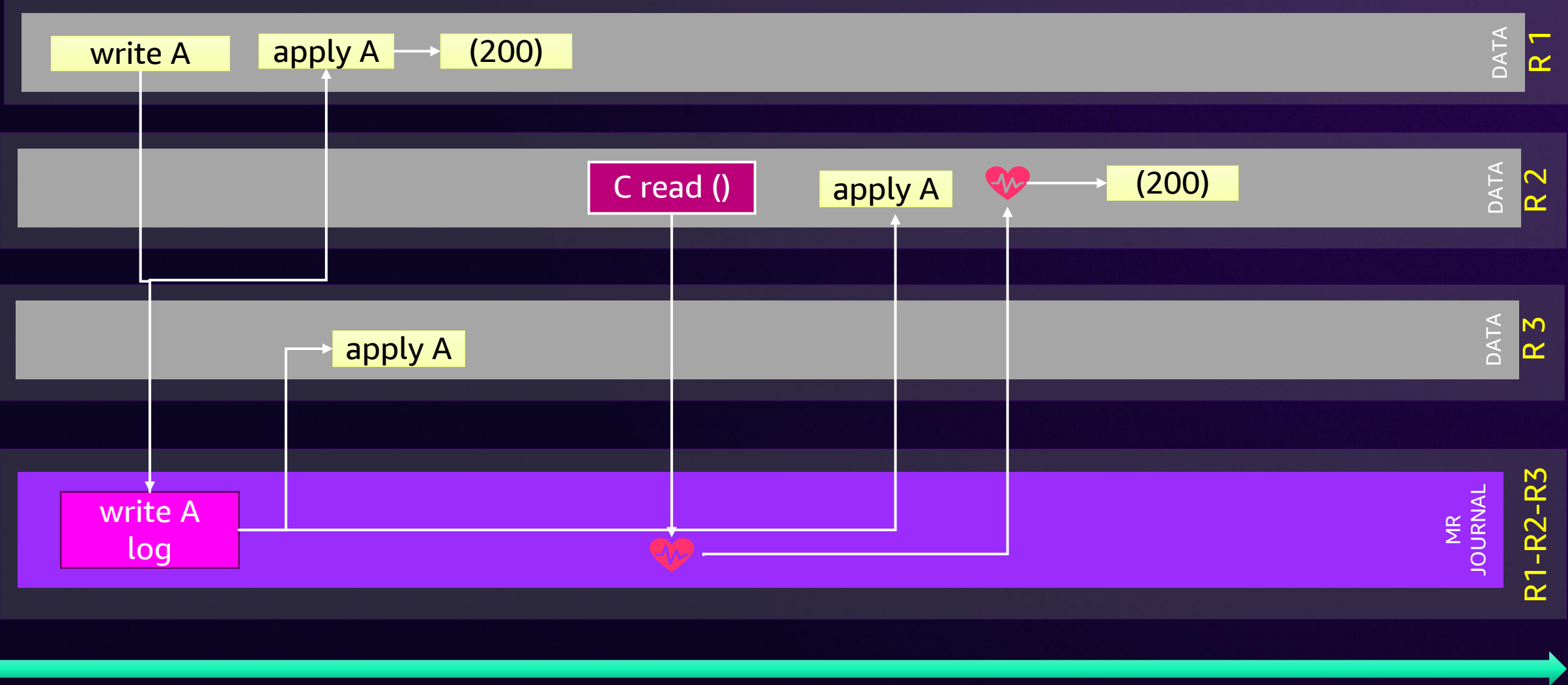
Multi-Region strong consistency global tables



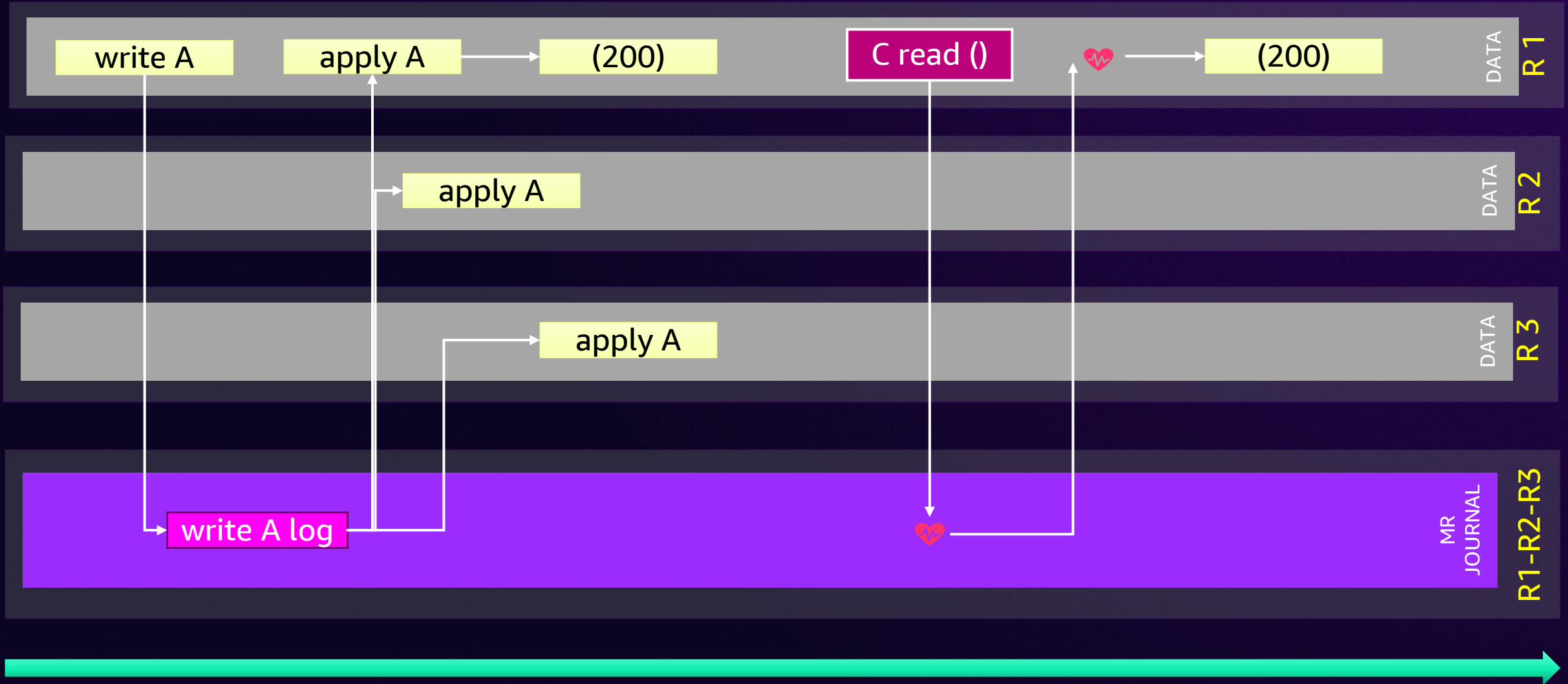
Multi-Region strong consistency global tables



Multi-Region strong consistency global tables

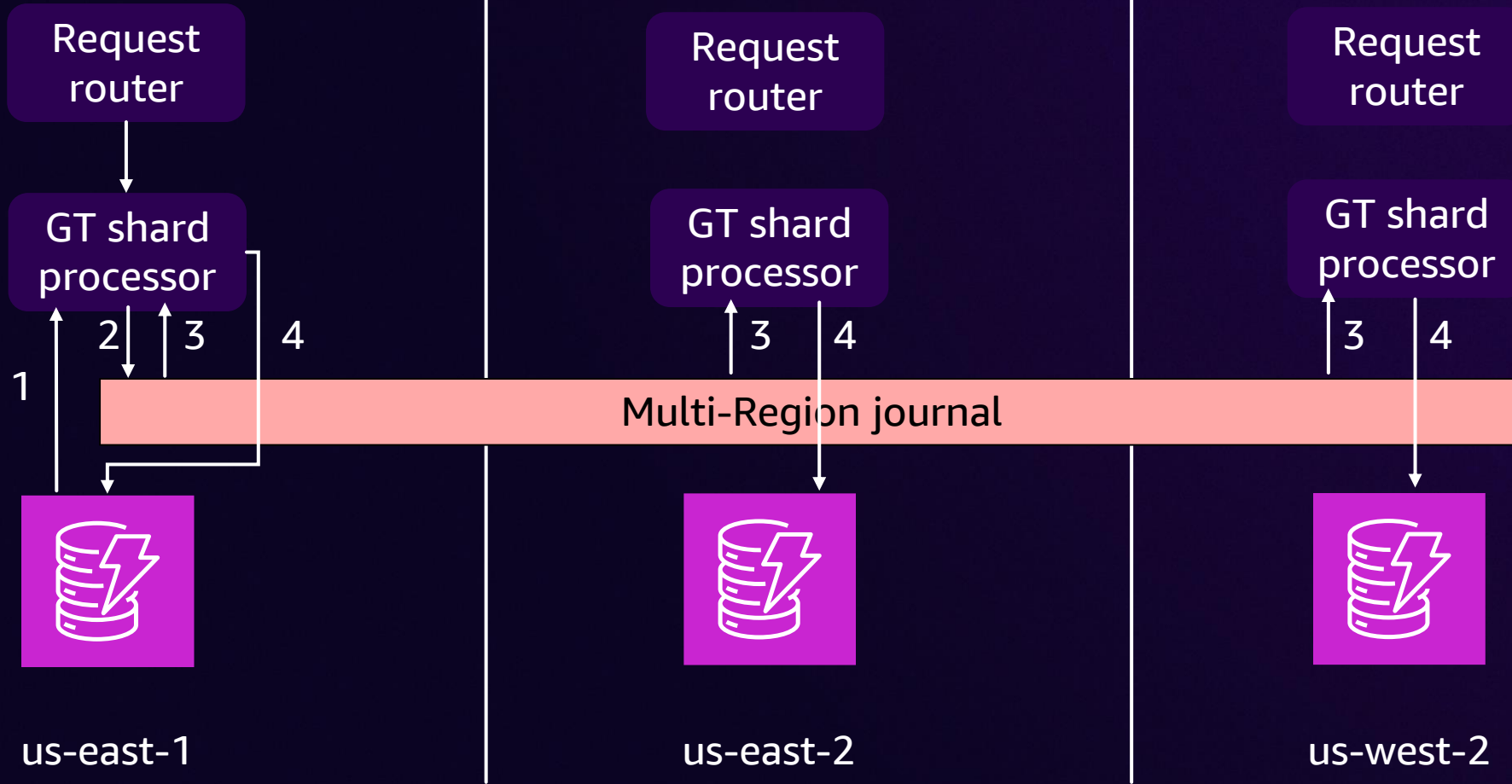


Multi-Region strong consistency global tables



Multi-Region writes

Write request



Idempotent writes

Write Request

+

Current item

=

Conditional insert or delete

Idempotent writes

Increment count



"re:Invent", 2024, 5,
{"ts":1733195894}

```
put {"re:Invent", 2024, 6, 1733195898}  
if ts == 1733195894
```


Idempotent writes

200



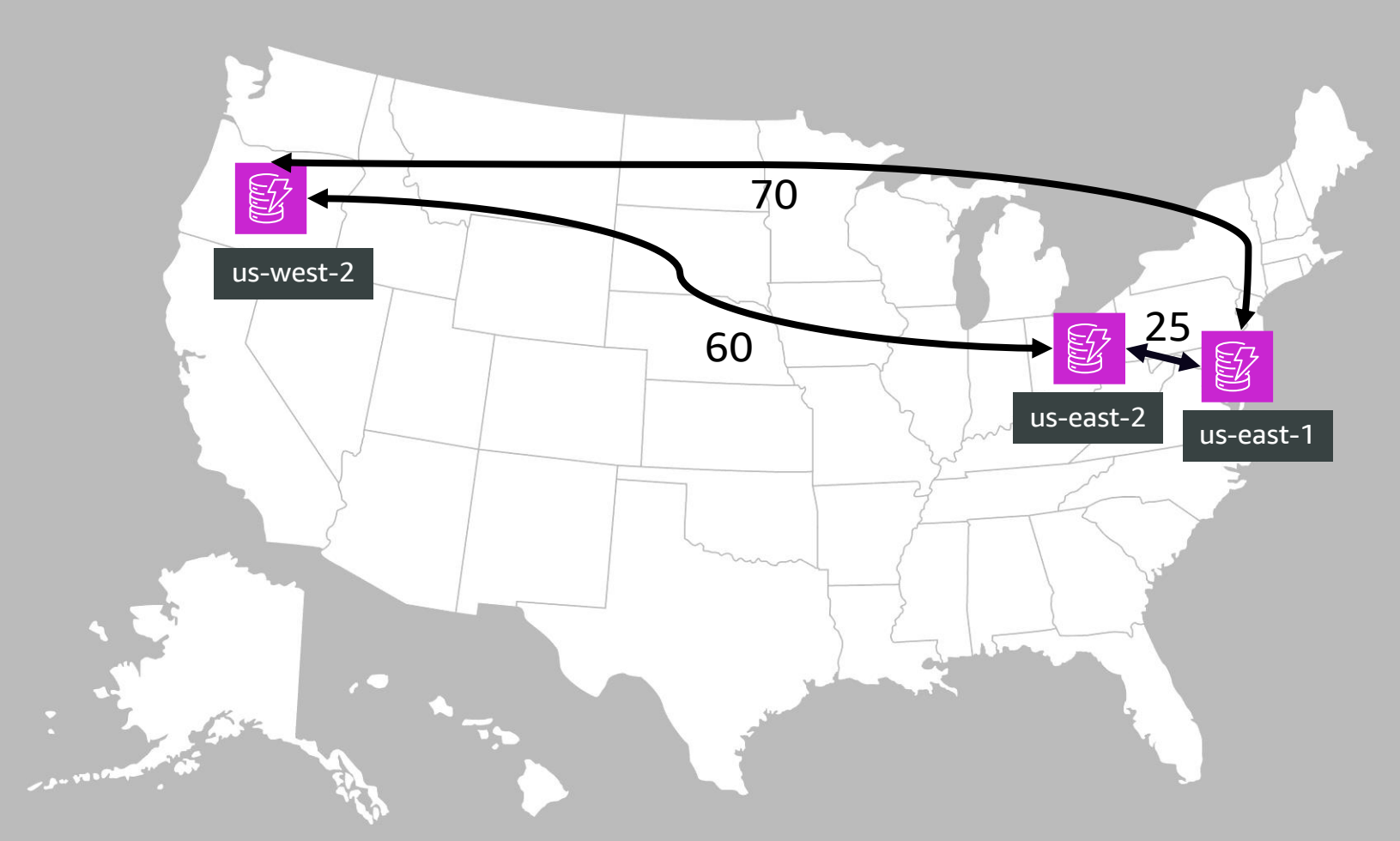
```
put {"re:Invent", 2024, 6, 1733195898}  
if ts == 1733195894
```

```
put {"re:Invent", 2024, 6, 1733195903}  
if ts == 1733195894
```

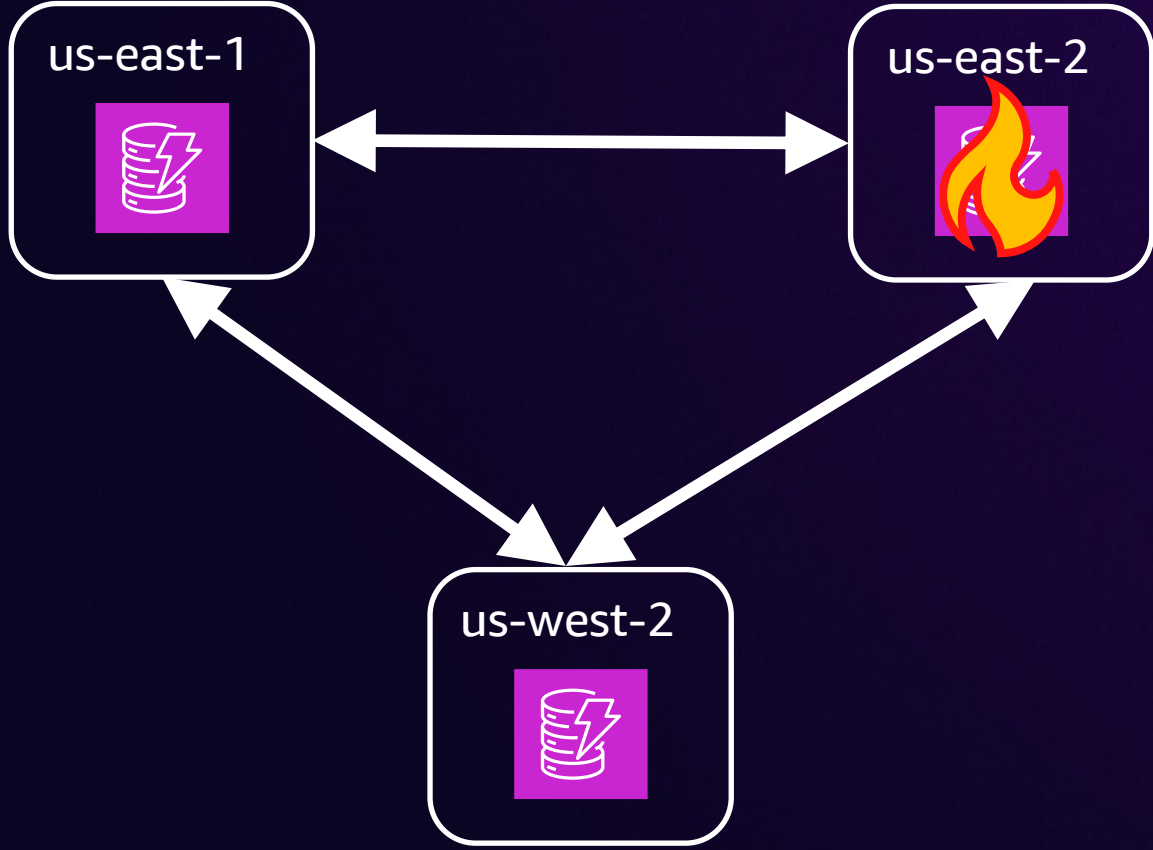


ReplicatedWriteConflictException

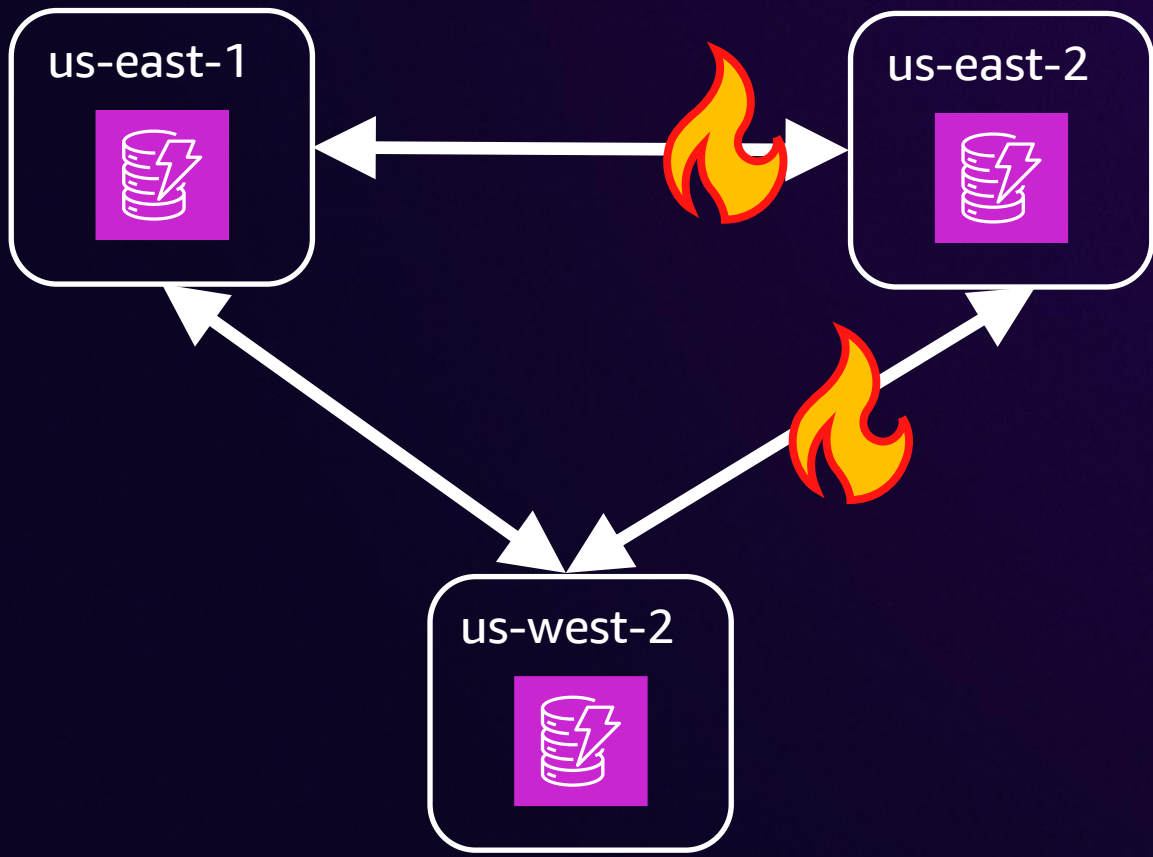
Latencies in milliseconds



Multi-Region strongly consistent – Region failure

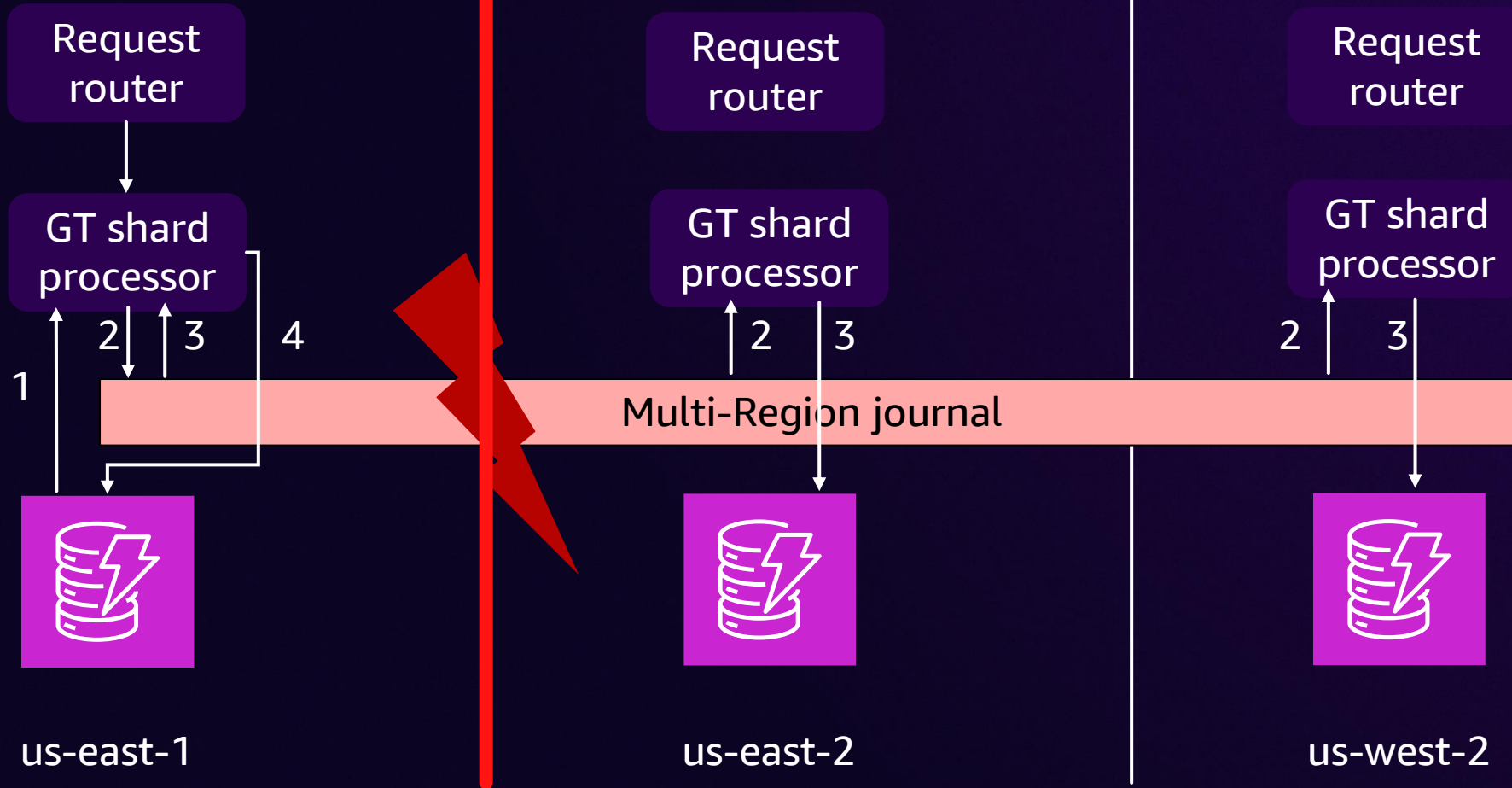


Multi-Region strongly consistent – Region isolation

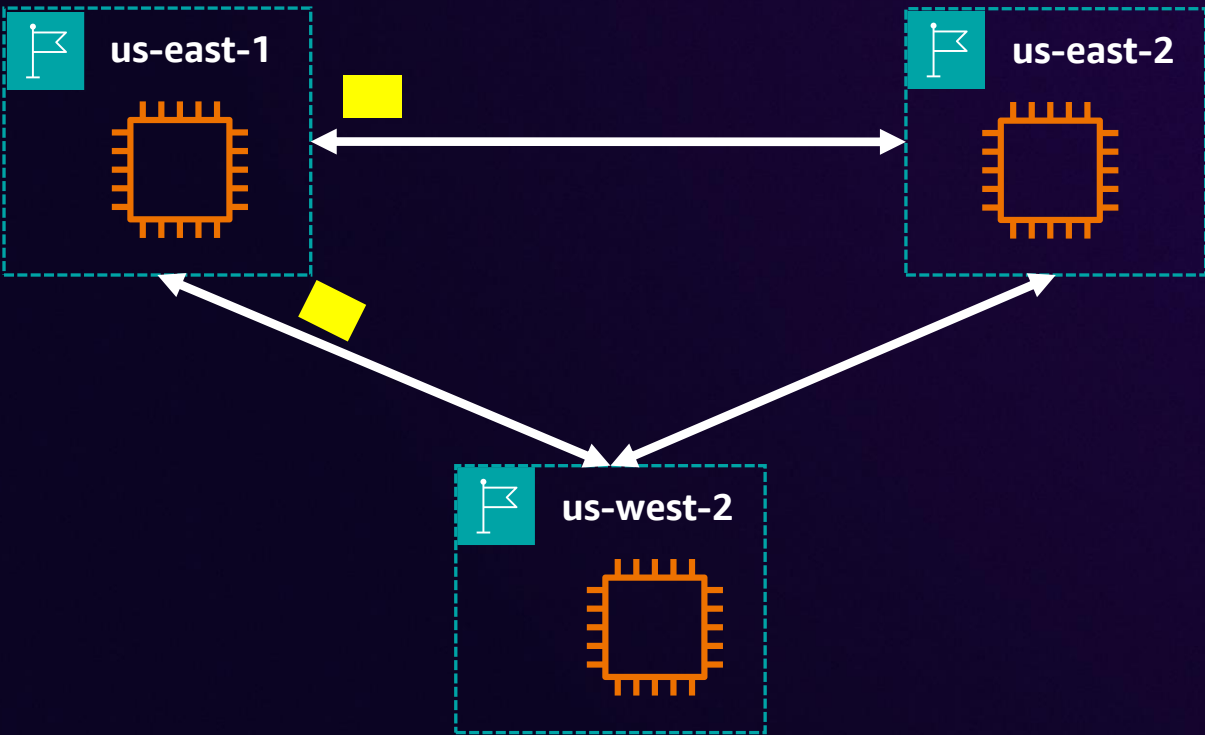


Multi-Region strongly consistent – Network partition

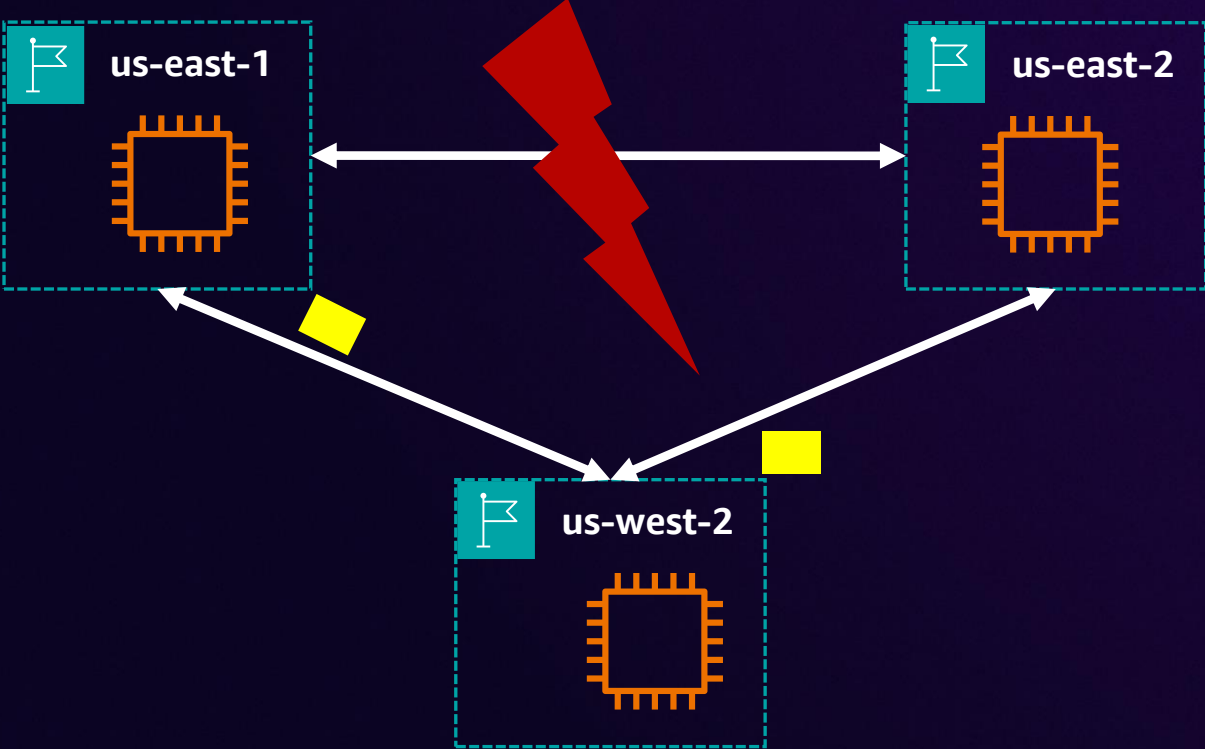
Write request



Multi-Region journal



Multi-Region journal – Resiliency



Eventual vs Multi-Region Strong Consistency

Scenario	Asynchronous Global Tables	MRSC Global Tables
Regional Failure	Highly available in remaining healthy regions	Highly available in remaining healthy regions*
Network Isolation	Isolated region remains available for reads/writes	Isolated region unavailable for strongly consistent reads/writes
Network Partition	Writes won't propagate between partitioned regions	All regions available for strongly consistent reads/writes*

* May experience higher latency

Model checking

P-model checker and P-observe



Chaos and scale testing



Anti-entropy



AWS Fault Injection Service

aws:dynamodb:global-table-pause-replication

Comparison of global tables

Asynchronous global tables

- Asynchronous replication of writes
- Eventual state is deterministic
- No universal ordering of writes
- Regional read-after-write consistency
- Low latency

MRSC global tables

- Synchronous replication of writes
- Eventual state is deterministic
- Universal write ordering
- Global read-after-write consistency
- Latencies depend on the Region pairs

Thank you!

Jeff Duffy

Somu Perianayagam



Please complete the session survey in the mobile app

