
AWS
re:Invent
NOV. 28 – DEC. 2, 2022 | LAS VEGAS, NV

ARC306-R

Multi-Region design patterns and best practices

Neeraj Kumar
Principal Solutions Architect
AWS

John Formento, Jr.
Principal Solutions Architect
AWS

Barry Sheward
Chief Cloud Architect
The Vanguard Group

Agenda

Multi-Region fundamentals

The Vanguard Group’s global multi-Region strategy

Multi-Region architecture patterns

Multi-Region fundamentals

Fundamental #1

Understand the requirements

“We need to create a multi-Region architecture.”

Maybe, but first...

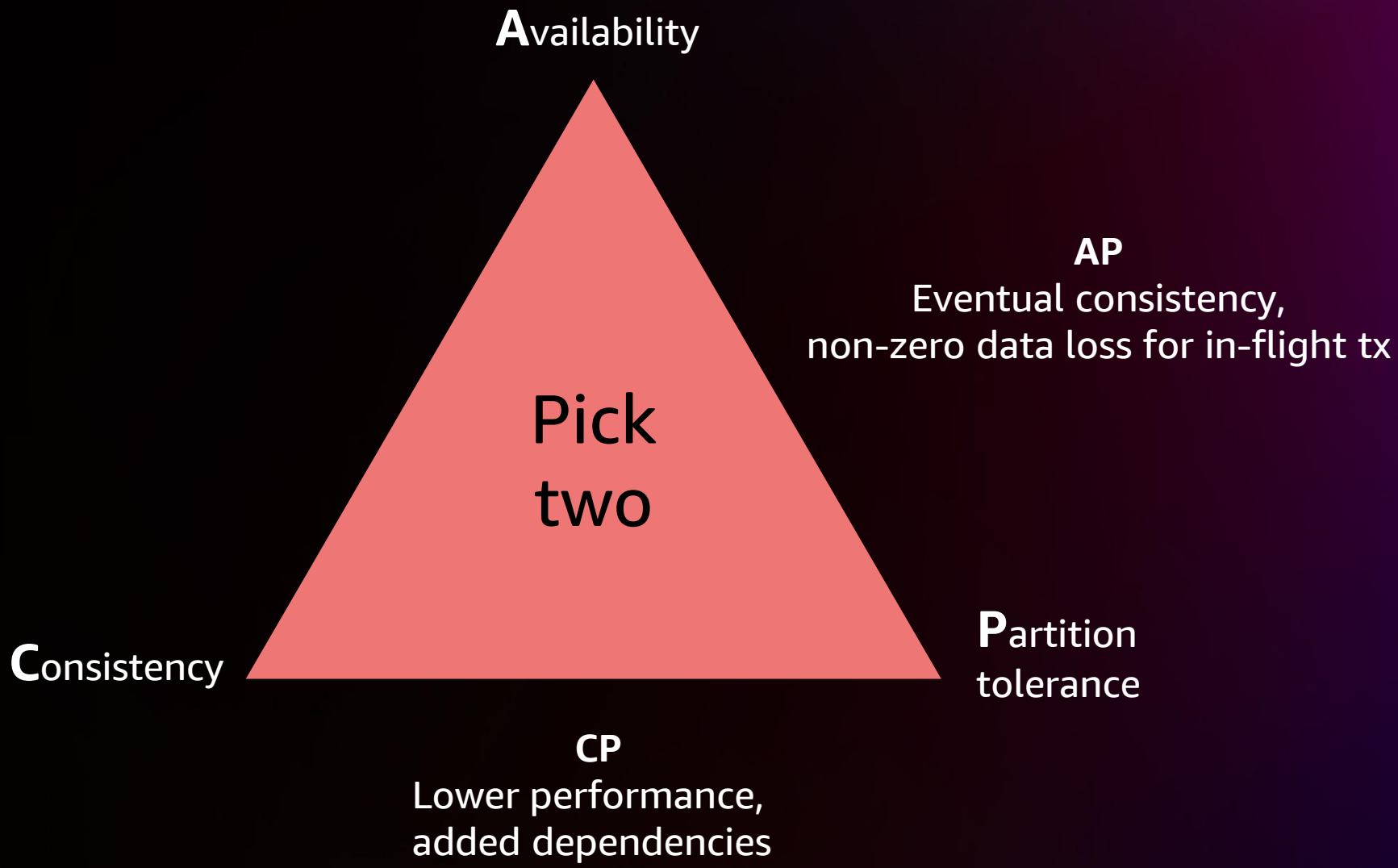
- What are the business requirements?
  - Do we need to meet an extreme resilience requirement?
  - Do we need to be able to prove the workload can run from another Region?
- Weigh the requirements against the cost and complexity

Tier	Max RTO	Max RPO	Criteria	Cost
Platinum	15 min	5 min	Mission-critical workloads	**$$$**
Gold	15 min–6 hrs	2 hrs	Important, but not mission-critical workloads	**$$**
Silver	**6 hrs–few days**	**24 hrs**	**Noncritical workloads**	**$**

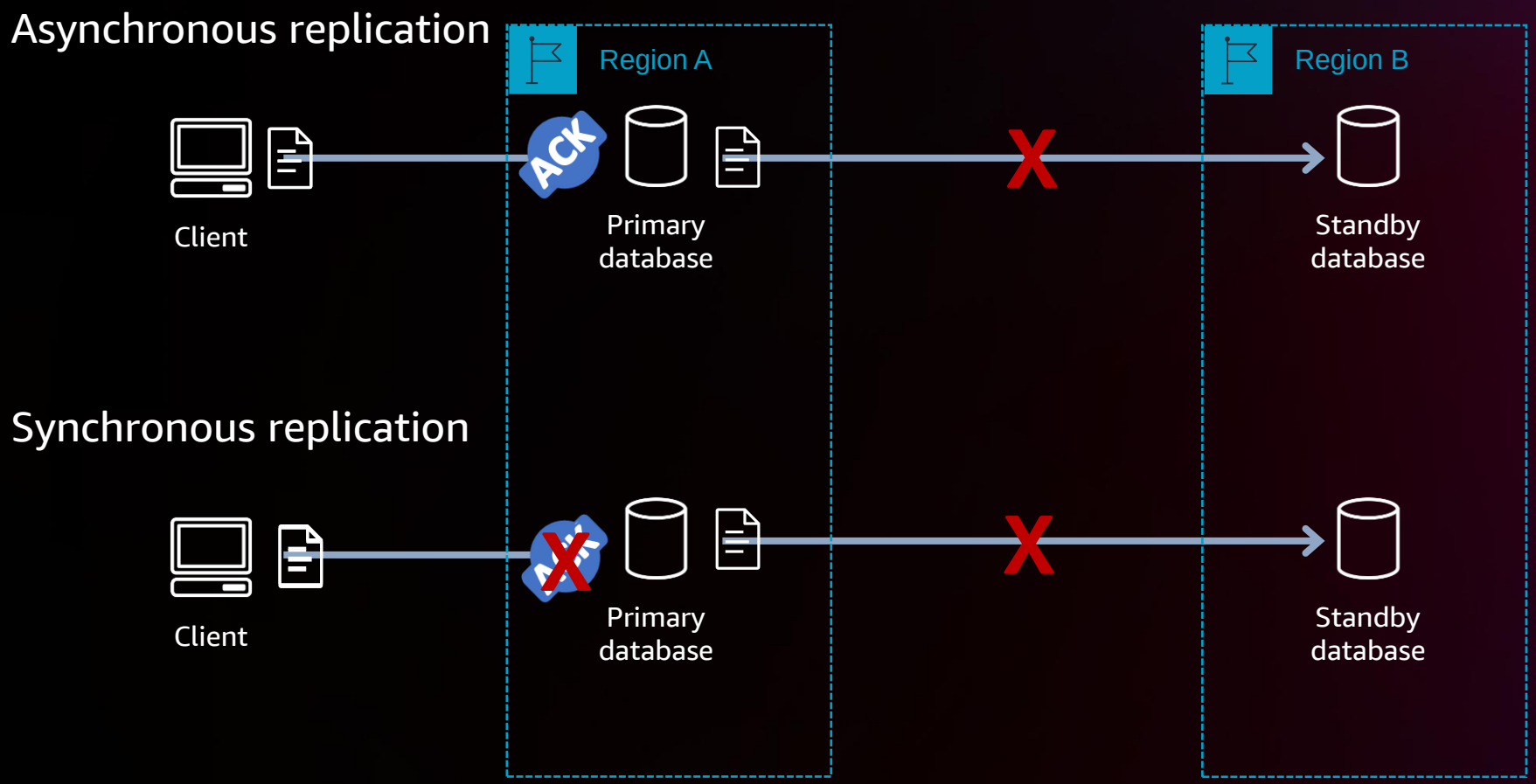
Fundamental #2

Understand the data

Data consistency requirements

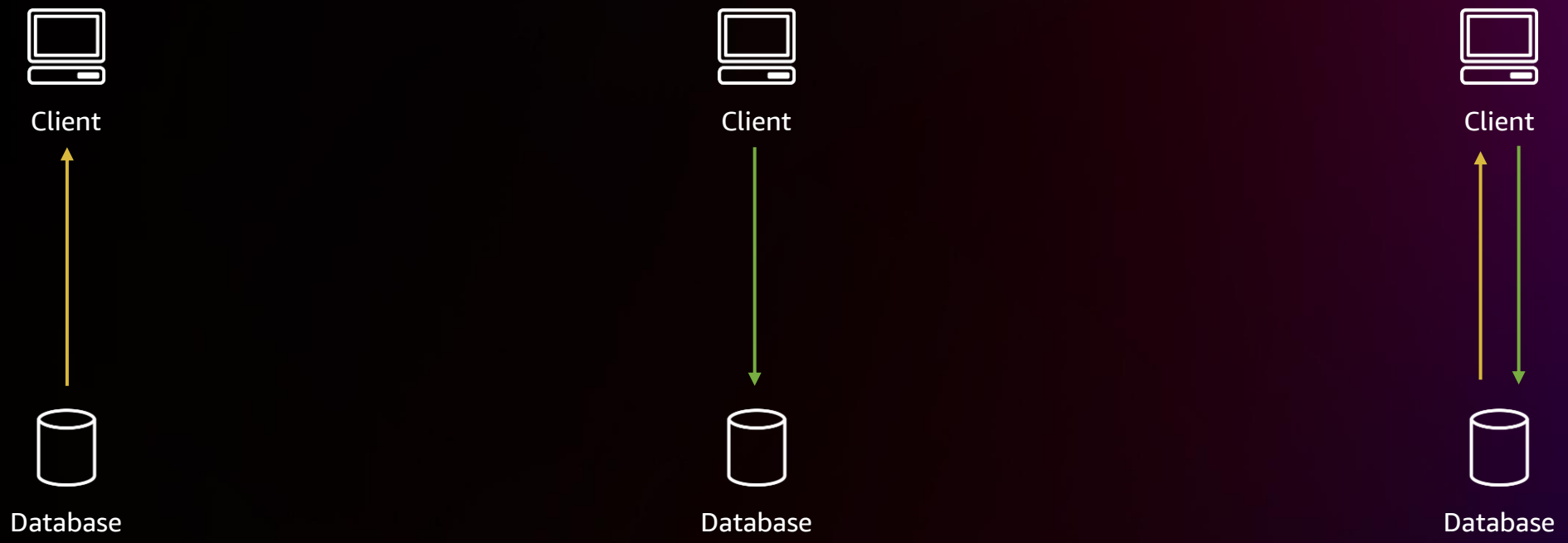


Data consistency requirements



Data access patterns

Understand the access characteristics



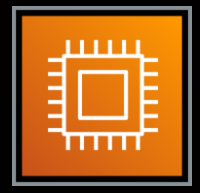
Read
Write

Fundamental #3

Understand dependencies

AWS services

- Determine what services are available in the desired Regions
- Identify what features will aid in your multi-Region journey


Compute


Storage


Database


Network and Content Delivery

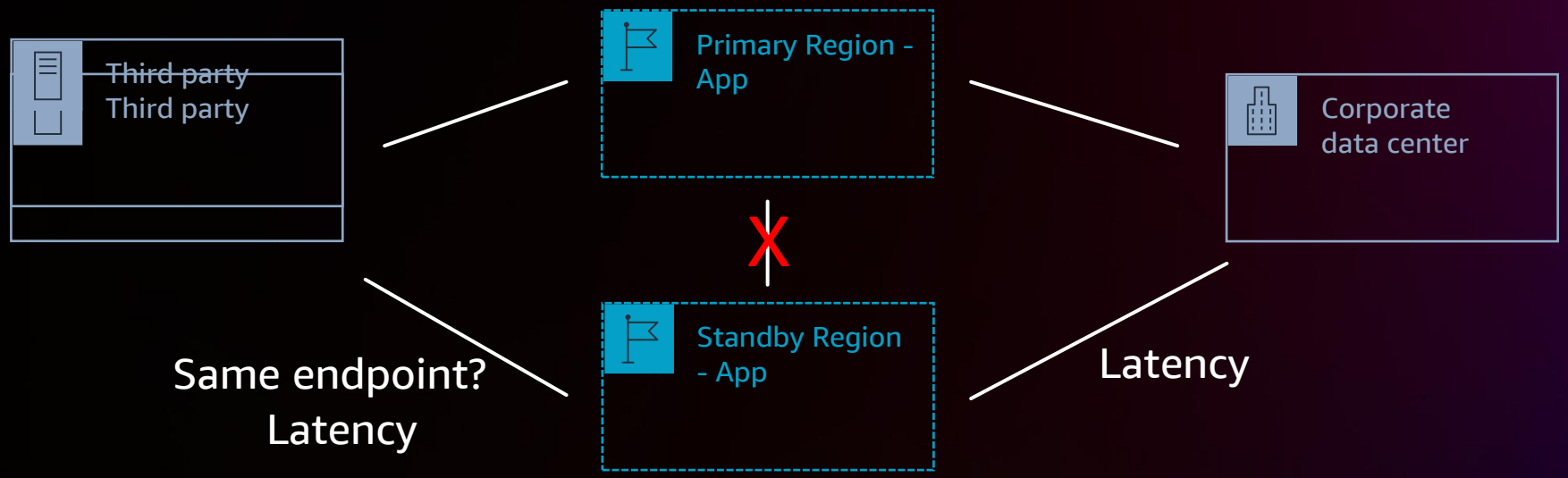

Analytics


Management and Governance


Security

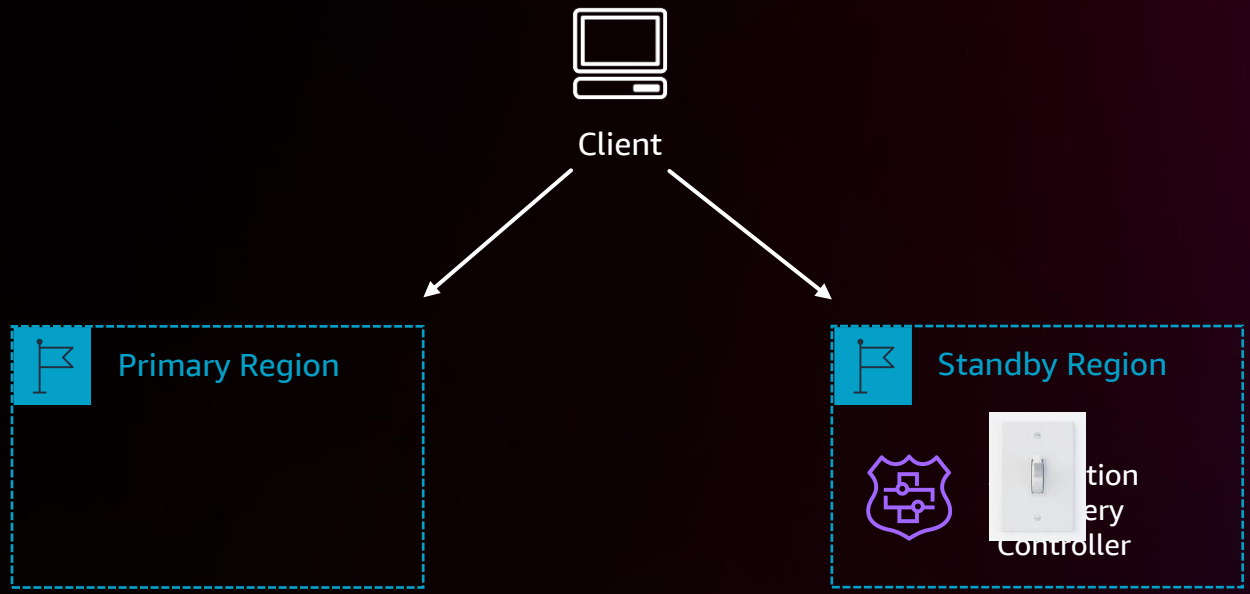
Internal and third-party dependencies

- Identify any on-premises dependencies
- Identify any third-party solutions in the workload



Failover mechanisms

- Never take a dependency on your primary Region for failover
- Scrutinize all dependencies in your failover mechanism

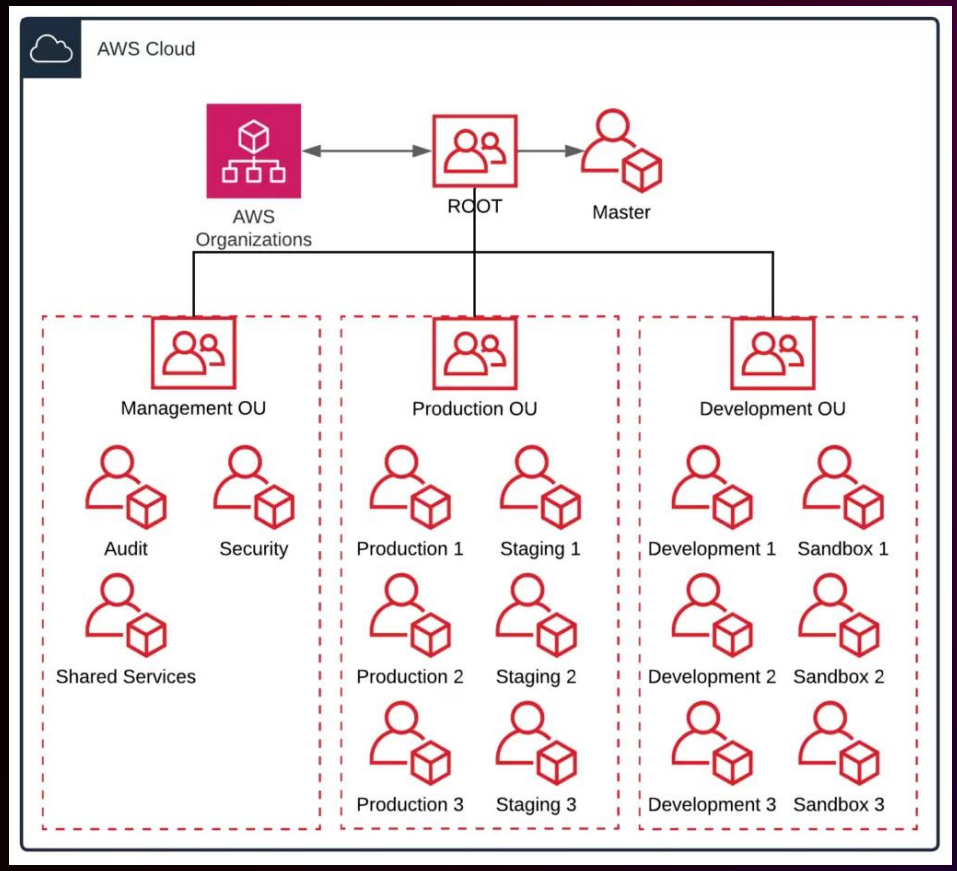


Fundamental #4

Operational readiness

AWS Account Management

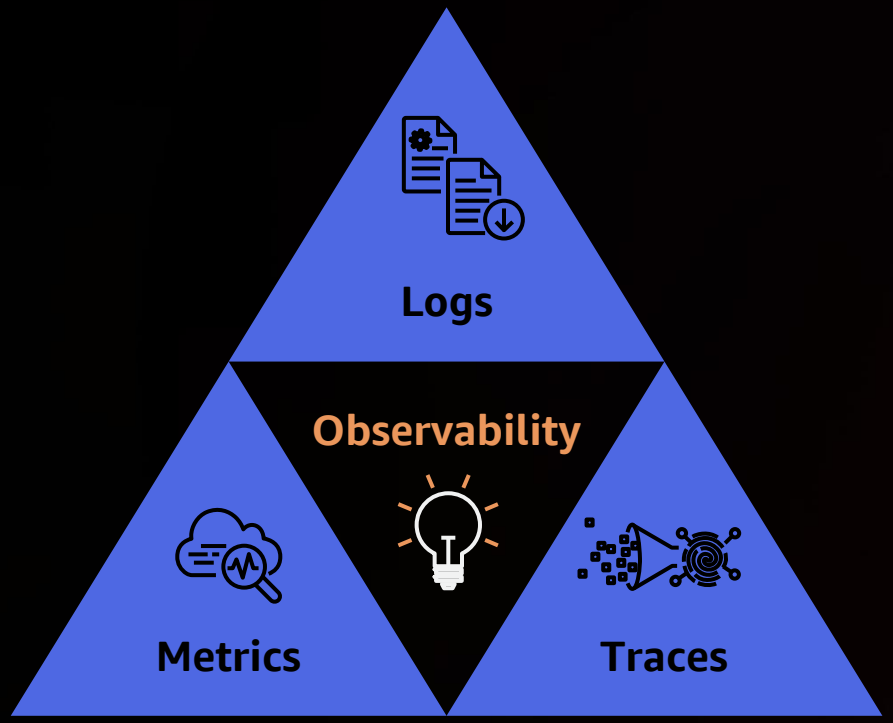
- AWS service quotas
- IAM permissions
- Service control policies



Deploy to a Region at a time

BLUE/GREEN OR CANARY DEPLOYMENTS

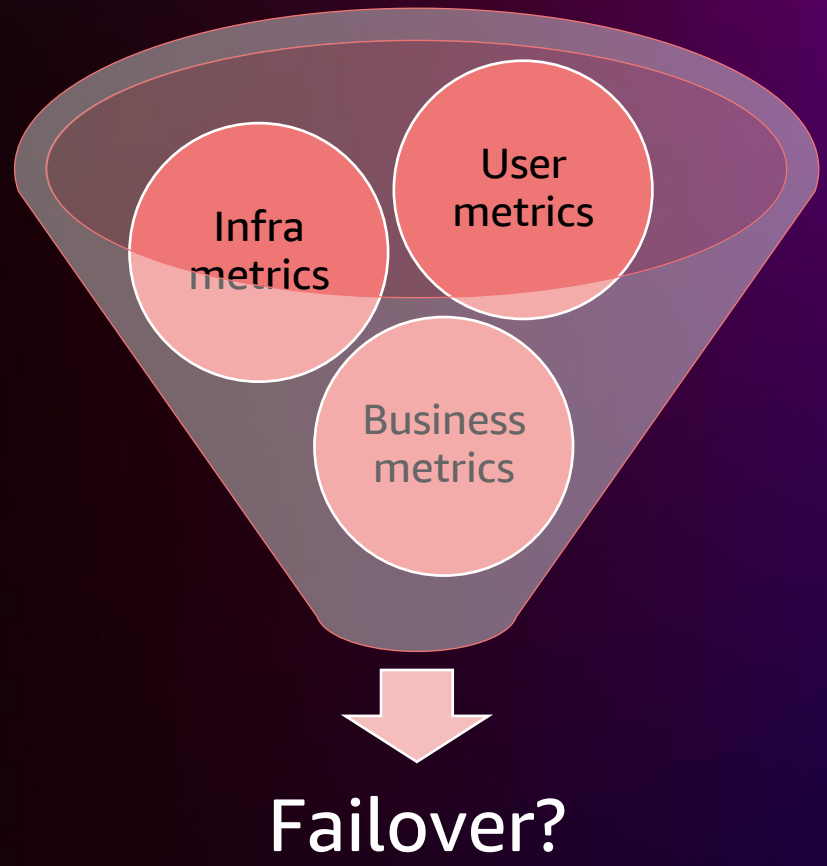
Monitoring: What changes in multi-Region?



- Replicate only high-level metrics to the other Region
  - User experience
  - User count
  - Transaction count
  - Replication status
- Monitor the health of the application in Region A from tooling in Region B
- Use telemetry from Region B for determining when to failover from application in Region A

Processes and people

- Untested DR strategy = no DR strategy
- Scope of failover
  - Individual microservices?
  - Business capability?
- Not only does the failover mechanism need to be tested regularly
  - Process for when to failover needs to be defined



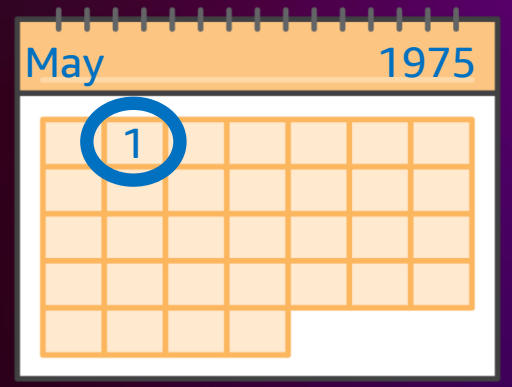
Cost and complexity

- Engineering effort
  - Is the application(s) ready for multi-Region?
- Operational overhead
  - People and process
  - Testing
  - Enhanced deployment processes
- Cost
  - Isolation of application stacks between Regions can result in 2x cost
  - Increased costs from the operational efforts and engineer efforts are on top of infrastructure costs

The Vanguard Group's global multi-Region strategy

Vanguard: Background

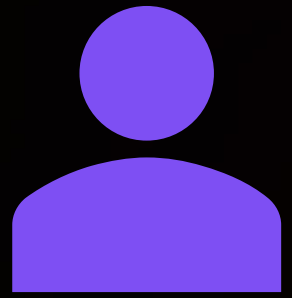
One of the world's largest investment companies, offering a large selection of low-cost mutual funds, ETFs, advice, and related services



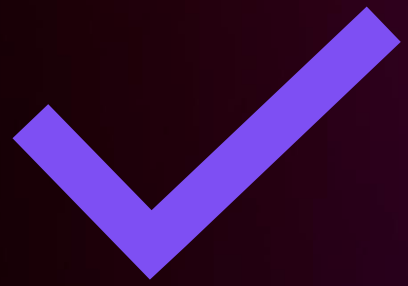
Began operations on May 1, 1975, in Valley Forge, PA

Headquartered in Malvern, PA, with operations in Arizona; London, UK; and Sydney, Australia

Understanding the requirements



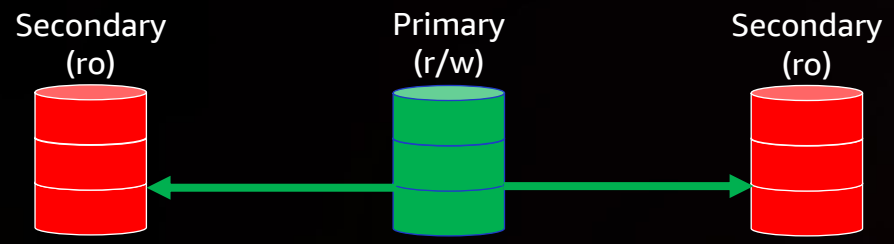
User experience



High availability

Understanding the data

Centralized – *Hub and spoke*



Some data doesn't change between *start of day* and *end of day*.
Data change is workday specific.

Pass the book/trade – *Follow-the-sun*

Works with both

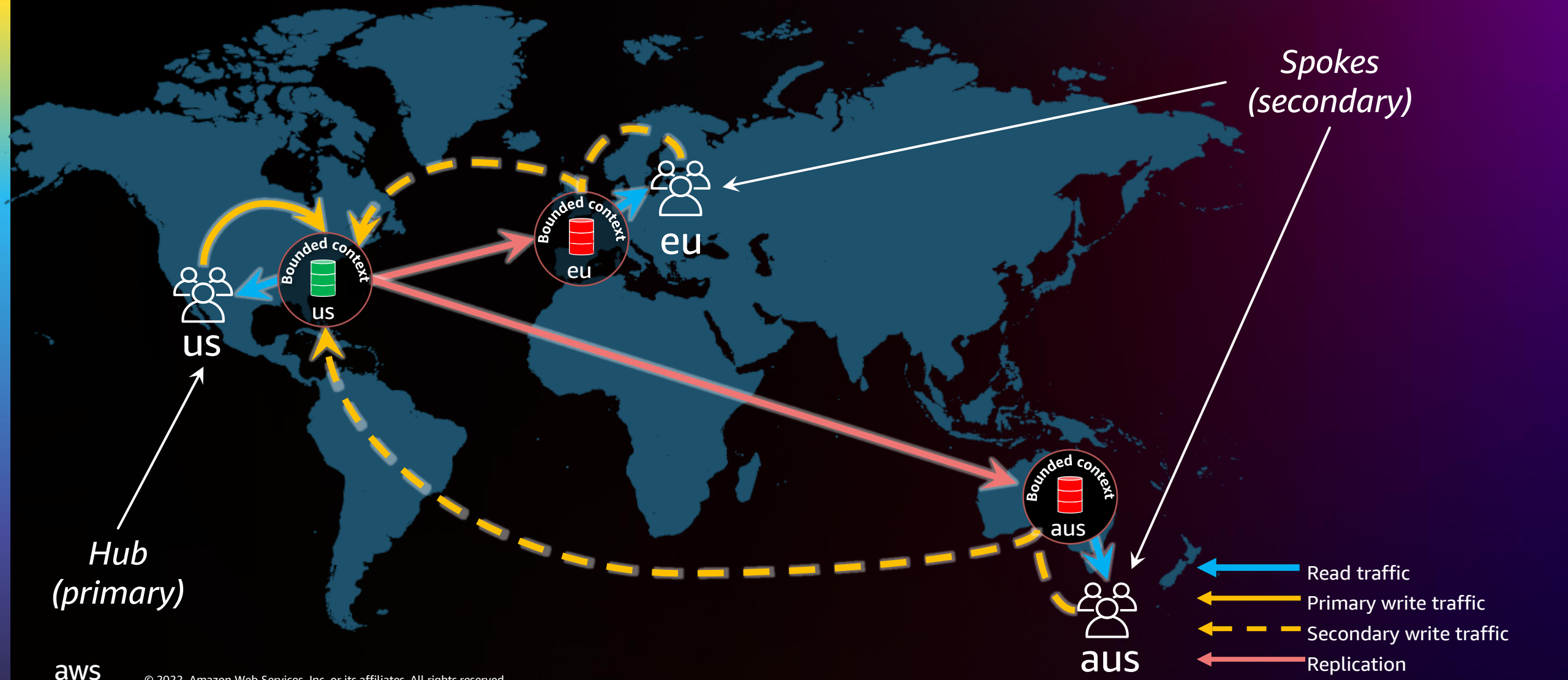
Amazon DynamoDB

Multi-primary data stores (e.g., DynamoDB)

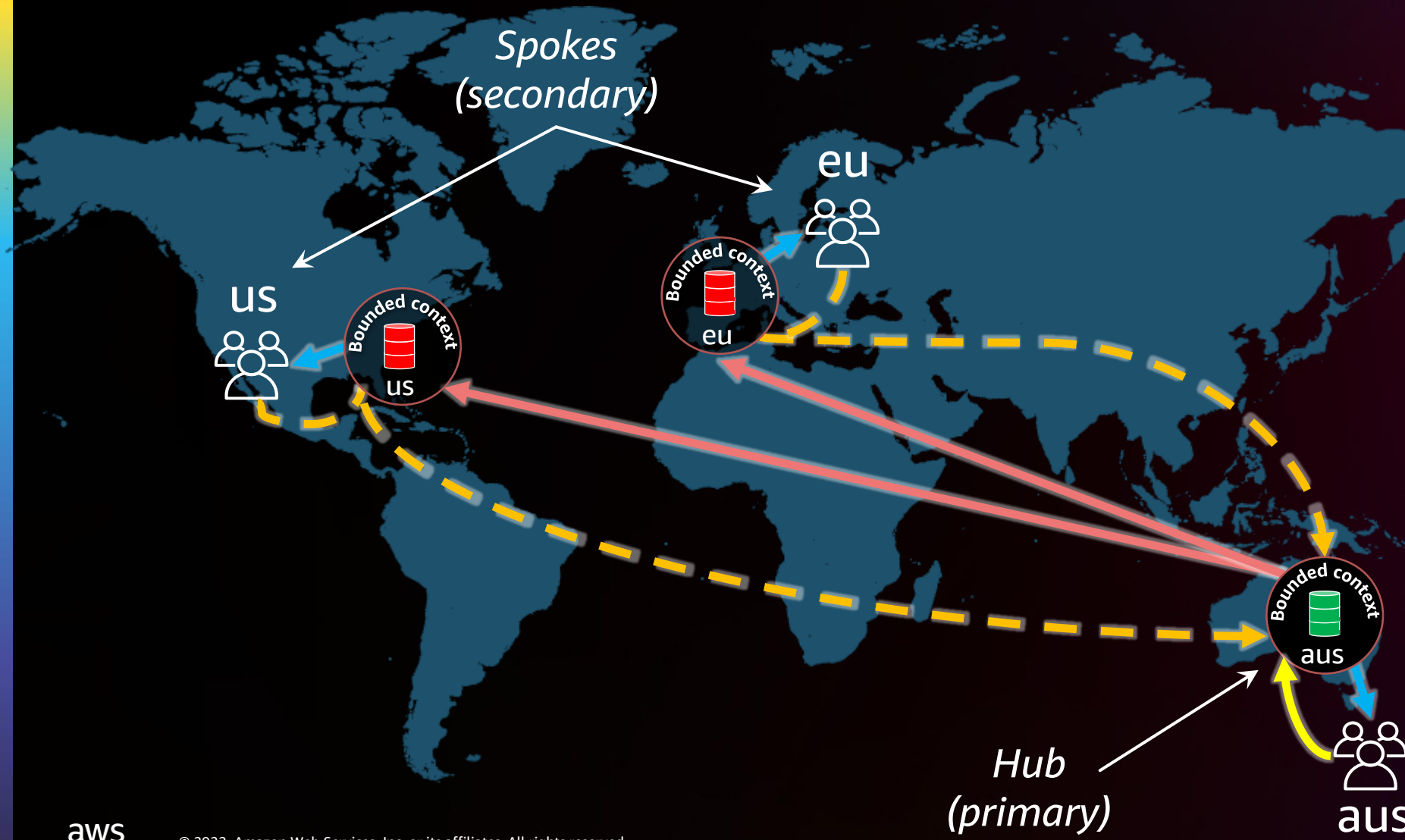
Amazon Aurora

Primary/secondary data stores (e.g., Aurora Global Database)

Data access patterns – Hub and spoke



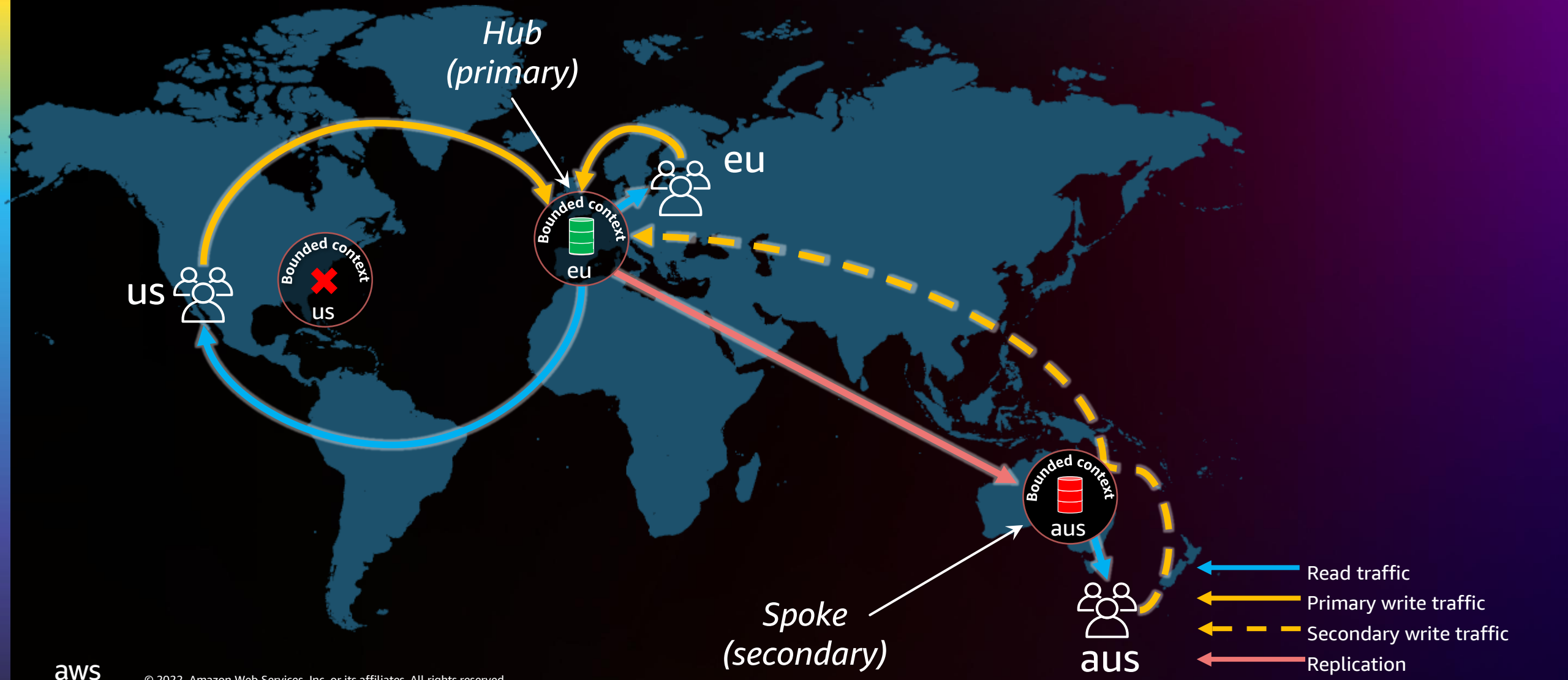
Data access patterns – Switchable hubs



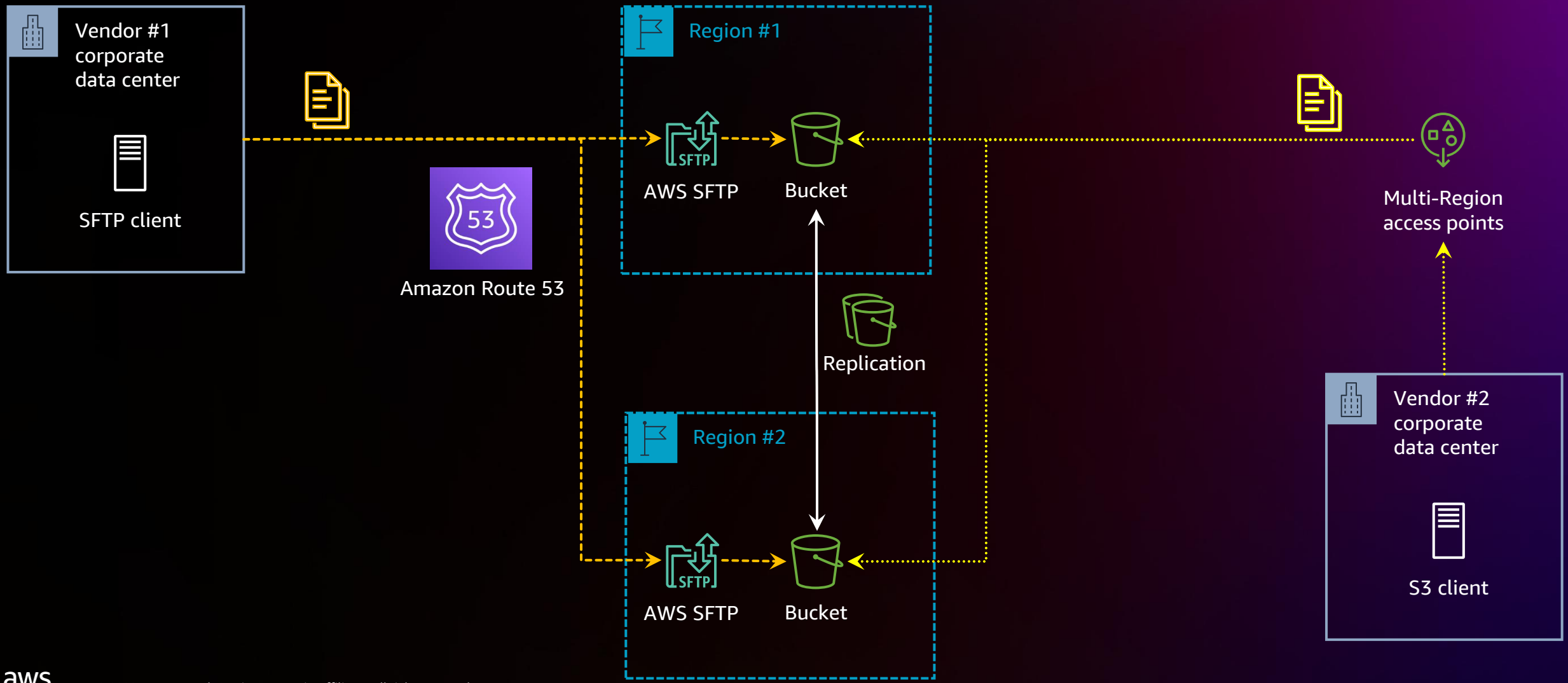
Using Aurora Global Database planned failover to move the hub (primary) during the day, we get *follow-the-sun*

DynamoDB doesn't need planned failover

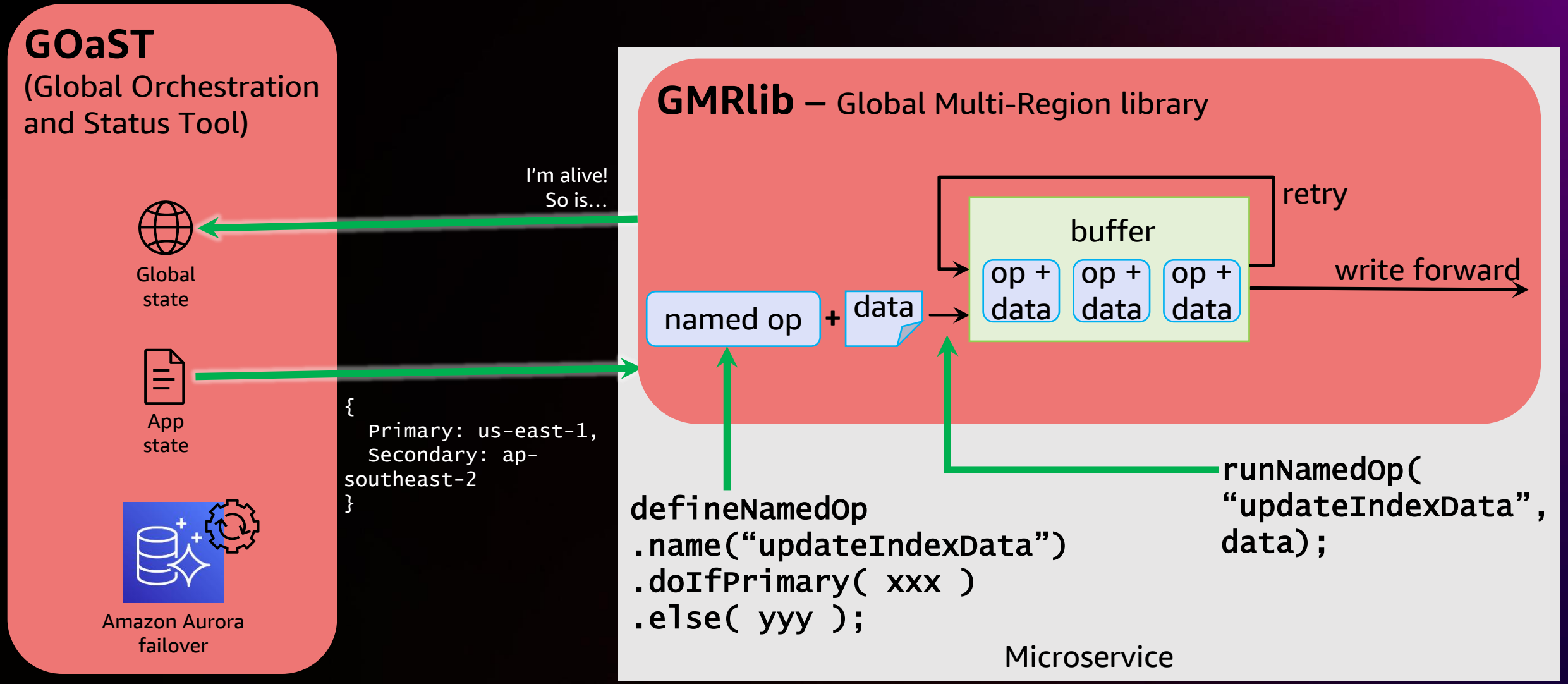
Data access patterns – Failures



Understanding dependencies – Third party



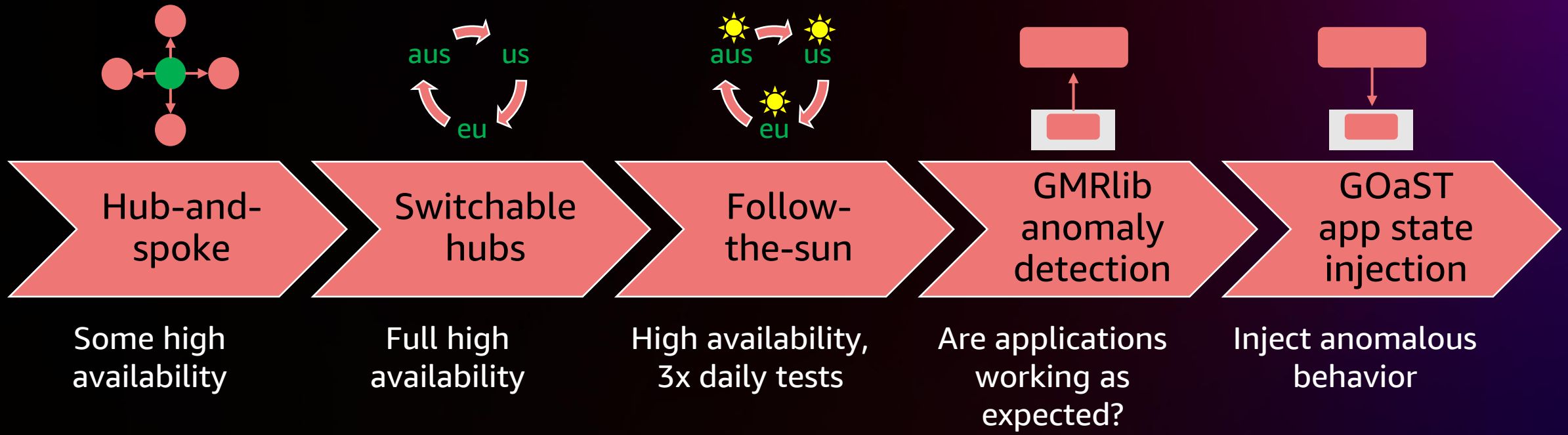
Understanding dependencies – Failover



Operational readiness

Remember “an untested DR strategy…is no DR strategy”?

With follow-the-sun, we made failover part of the standard operating procedure



Outcomes

Resiliency

User experience

Single points of failure

Costs **$2x** but for one workload, a single incident avoided could cover the lifetime cost!

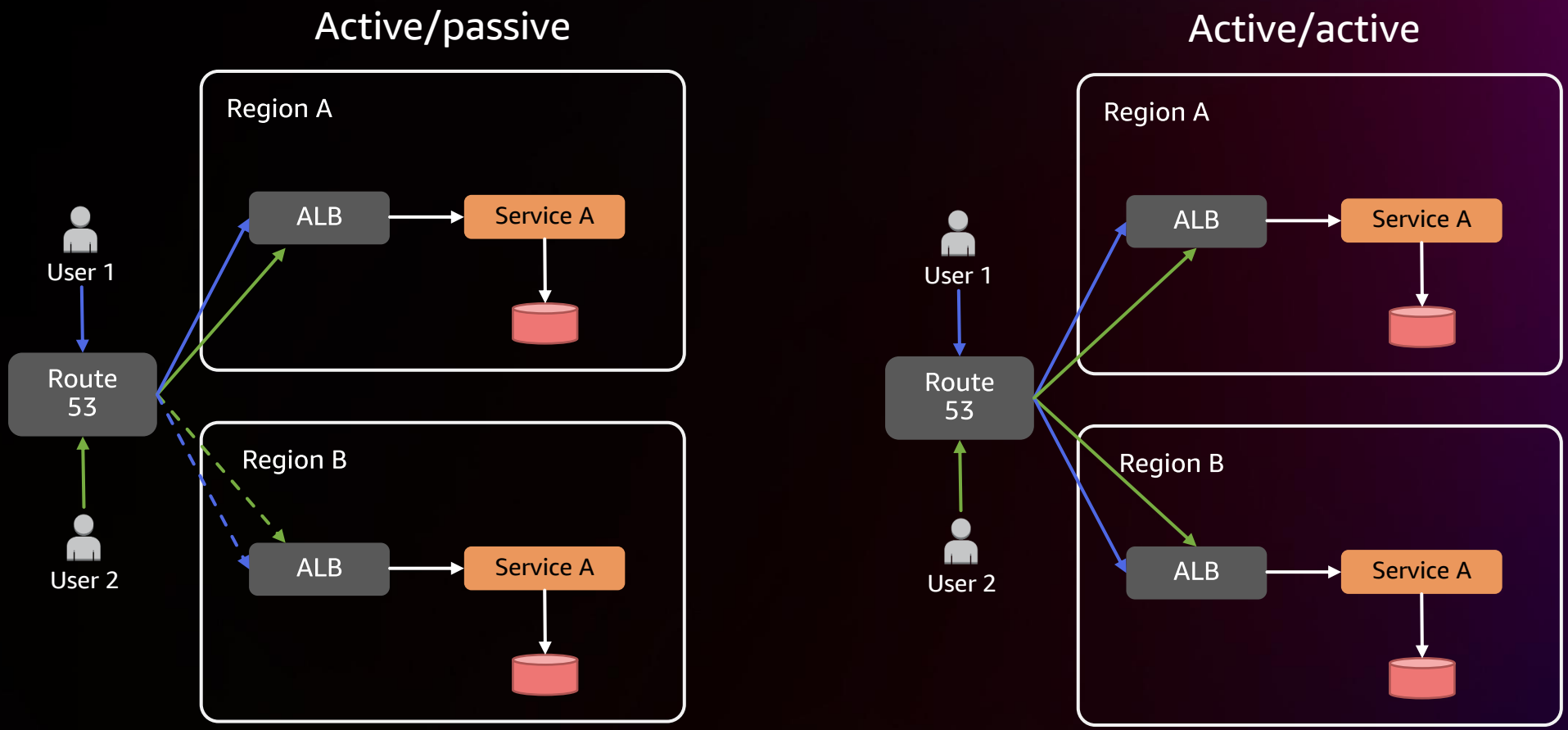
Lessons learned

In summary: global multi-Region is difficult!

- Understand the requirements
- Understand business processes
- Understand the data
- Understand technology limitations
- Understand dependencies
- Set user expectations
- Prepare and test

Expanded multi-Region architecture patterns

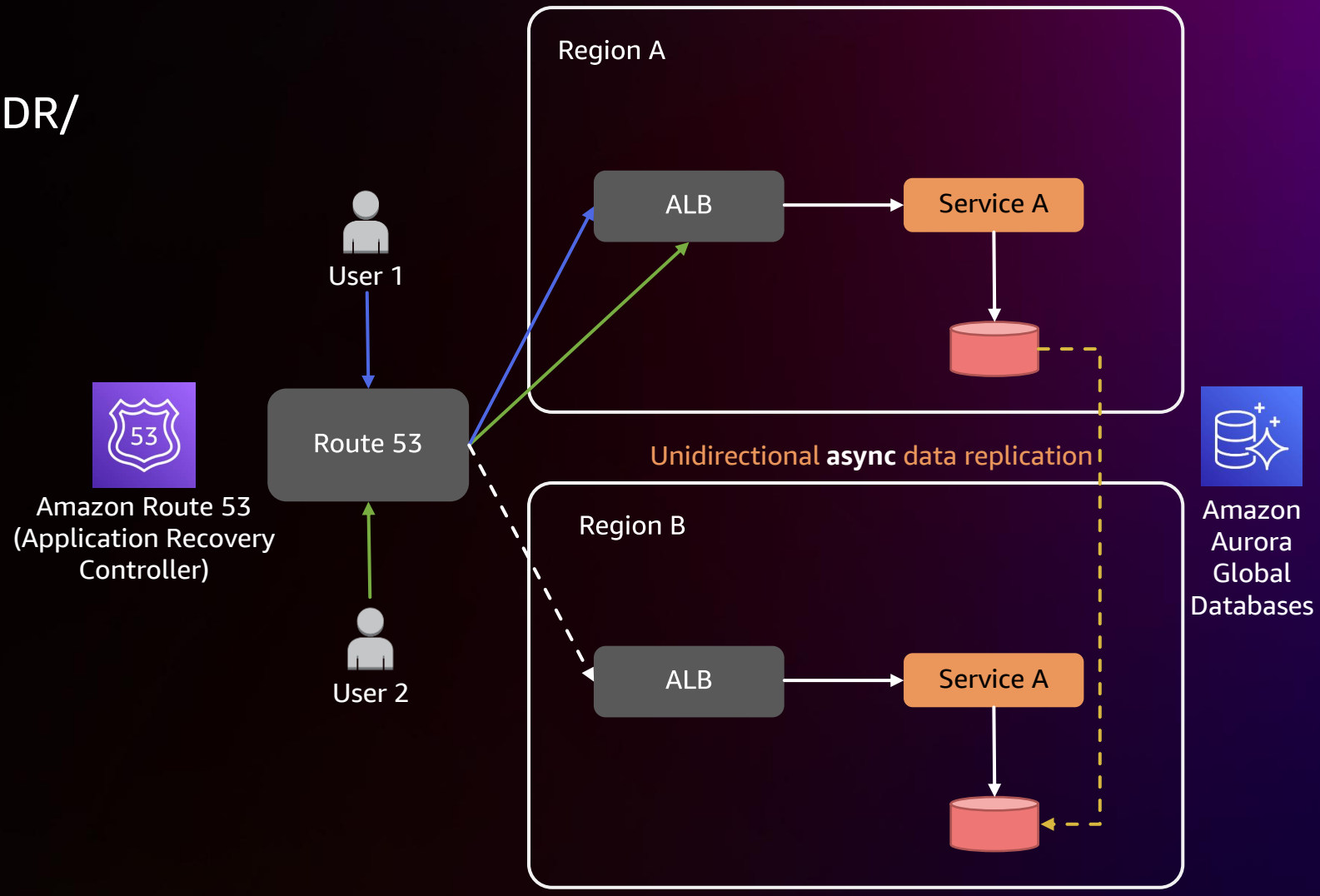
Two macro patterns



Pattern #1: Active/passive

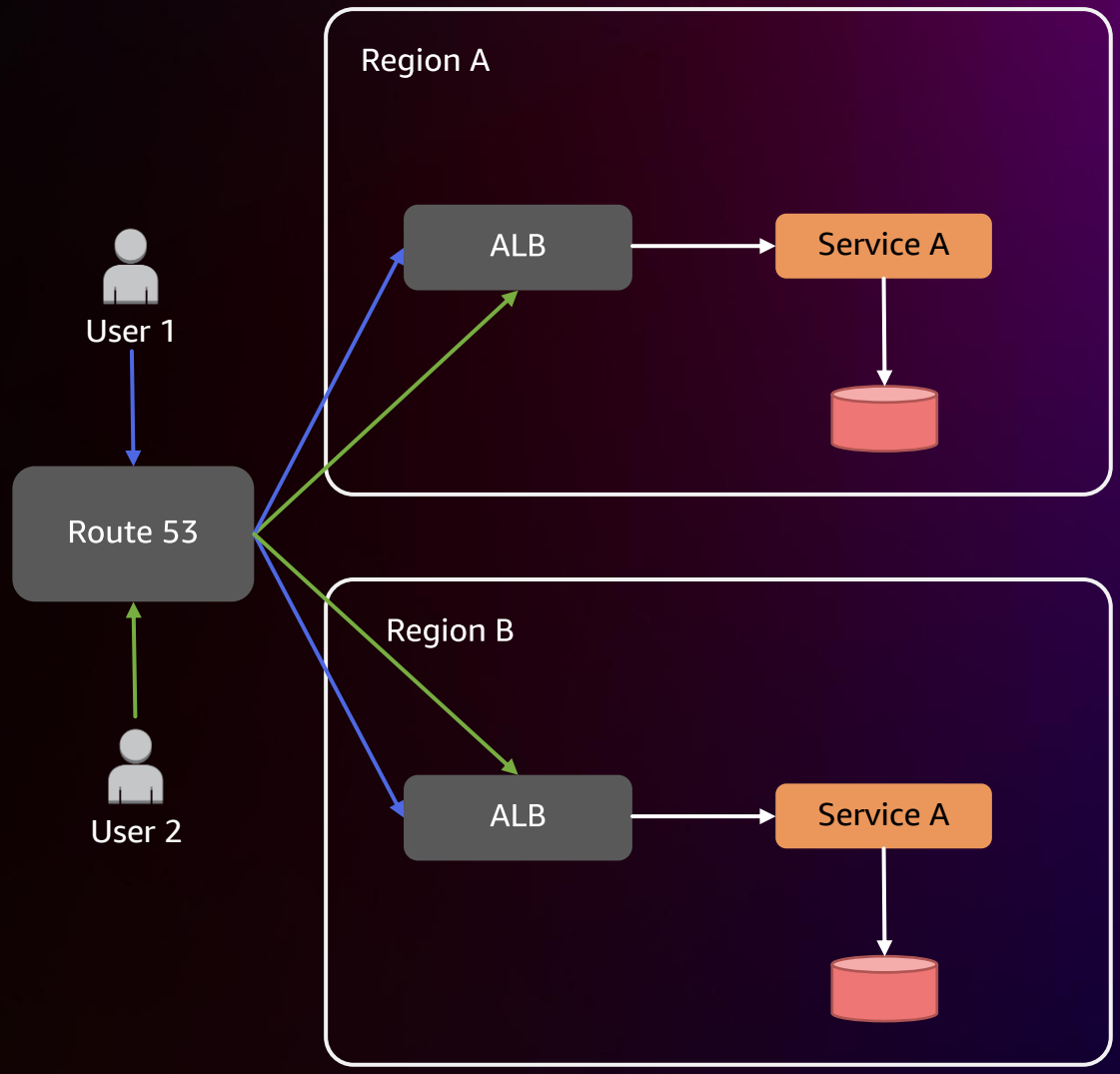
PILOT LIGHT, WARM STANDBY, ACTIVE/HOT STANDBY

- Use cases
  - Regional failover/failback for DR/ operational continuity
  - Regulatory requirements
- Key design considerations
  - Observability (health checks)
  - CI/CD, code and configuration drifts
  - Routing strategy
  - Failover/failback procedures



Pattern #2: Active/active

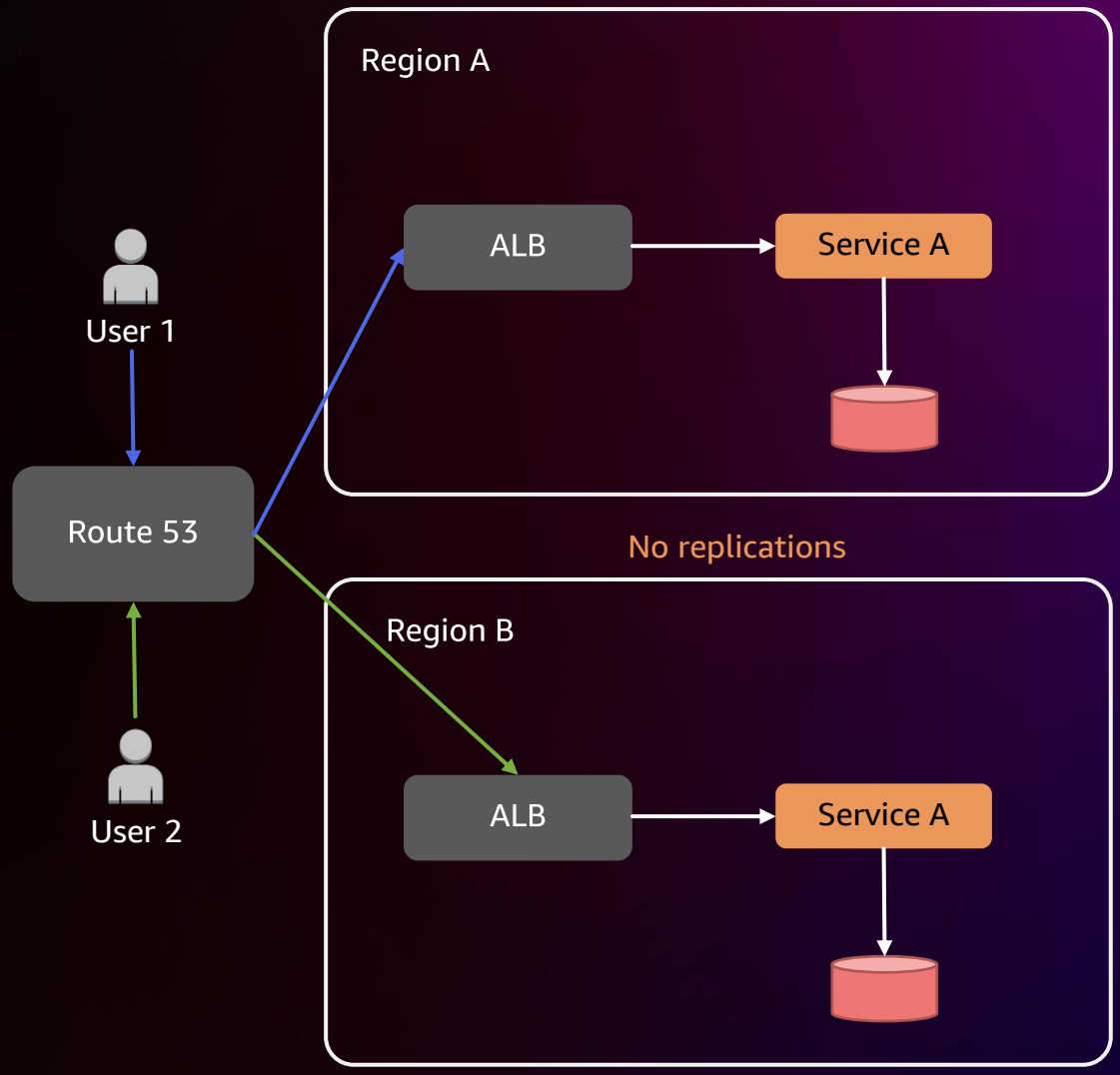
- Use cases
  - Extreme high availability
  - Performance – latency sensitivity
  - Low RTO/RPO



Pattern #2.1: Active/active

REGIONAL SHARDING

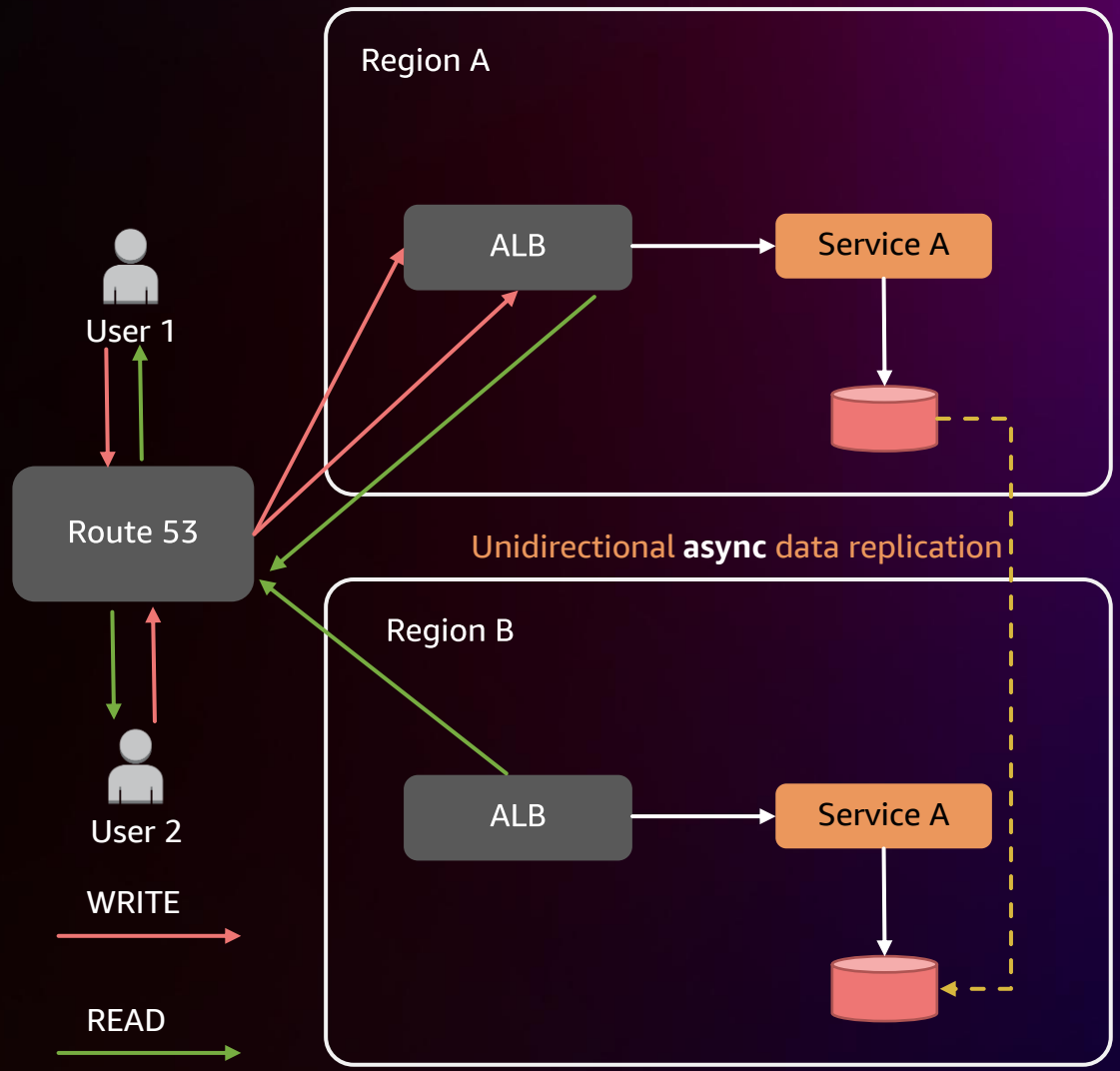
- Use cases
  - Global expansion
  - Data locality, regulatory
  - Performance – latency-sensitive
- Key design considerations
  - Observability (health checks)
  - CI/CD, code and configuration drifts
  - Routing strategy



Pattern #2.2: Active/active

SINGLE WRITER; MULTIPLE READERS

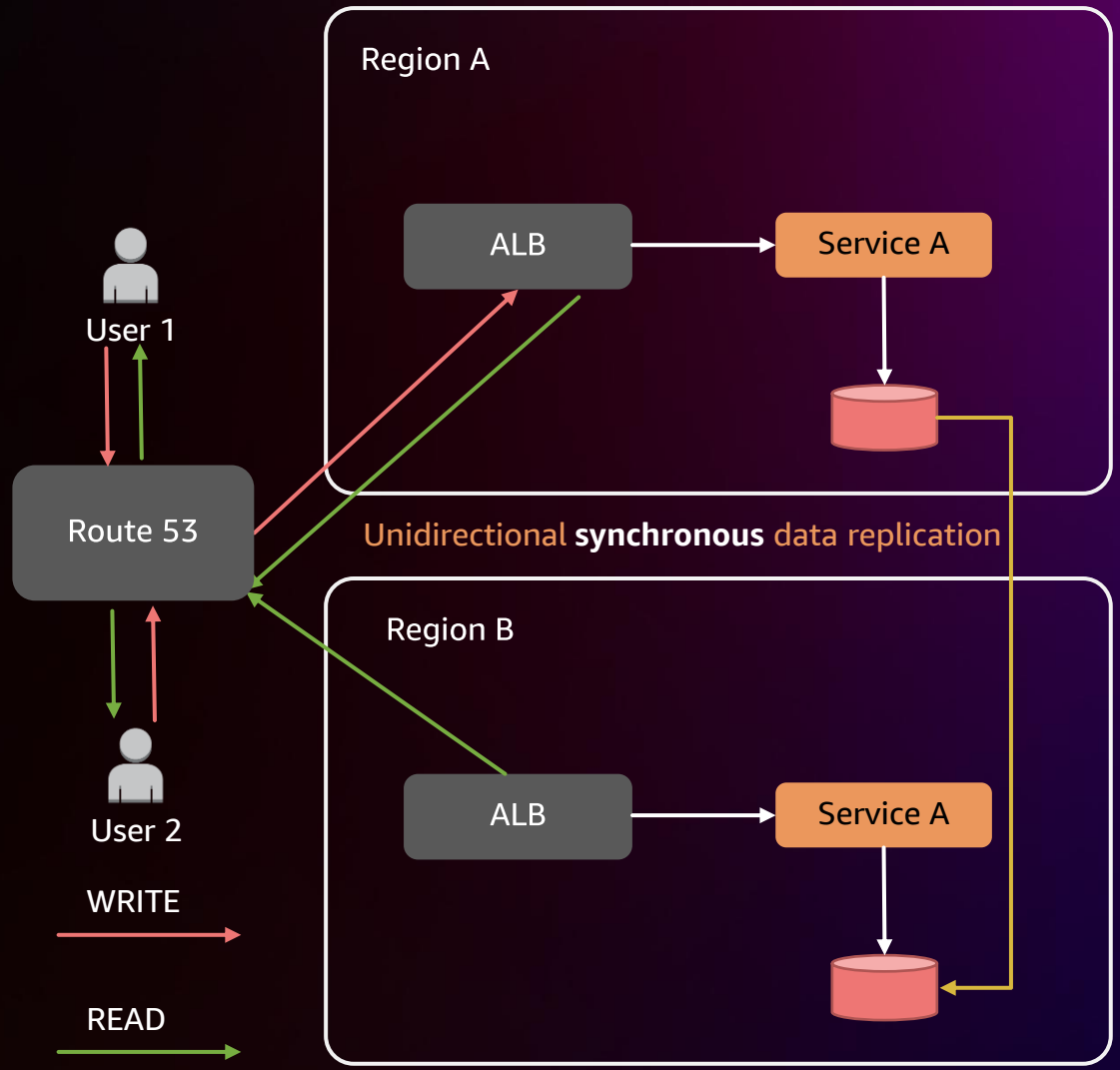
- Use cases
  - High availability
  - Latency sensitivity
  - Eventual consistency – async replication
  - Read-heavy workloads
- Key design considerations
  - Observability (health checks)
  - CI/CD, code and configuration drifts
  - Routing strategy
  - Failover/failback procedures (writer)
  - Read after write consistency



Pattern #2.3: Active/active

SINGLE WRITER; MULTIPLE READERS (STRONG CONSISTENCY)

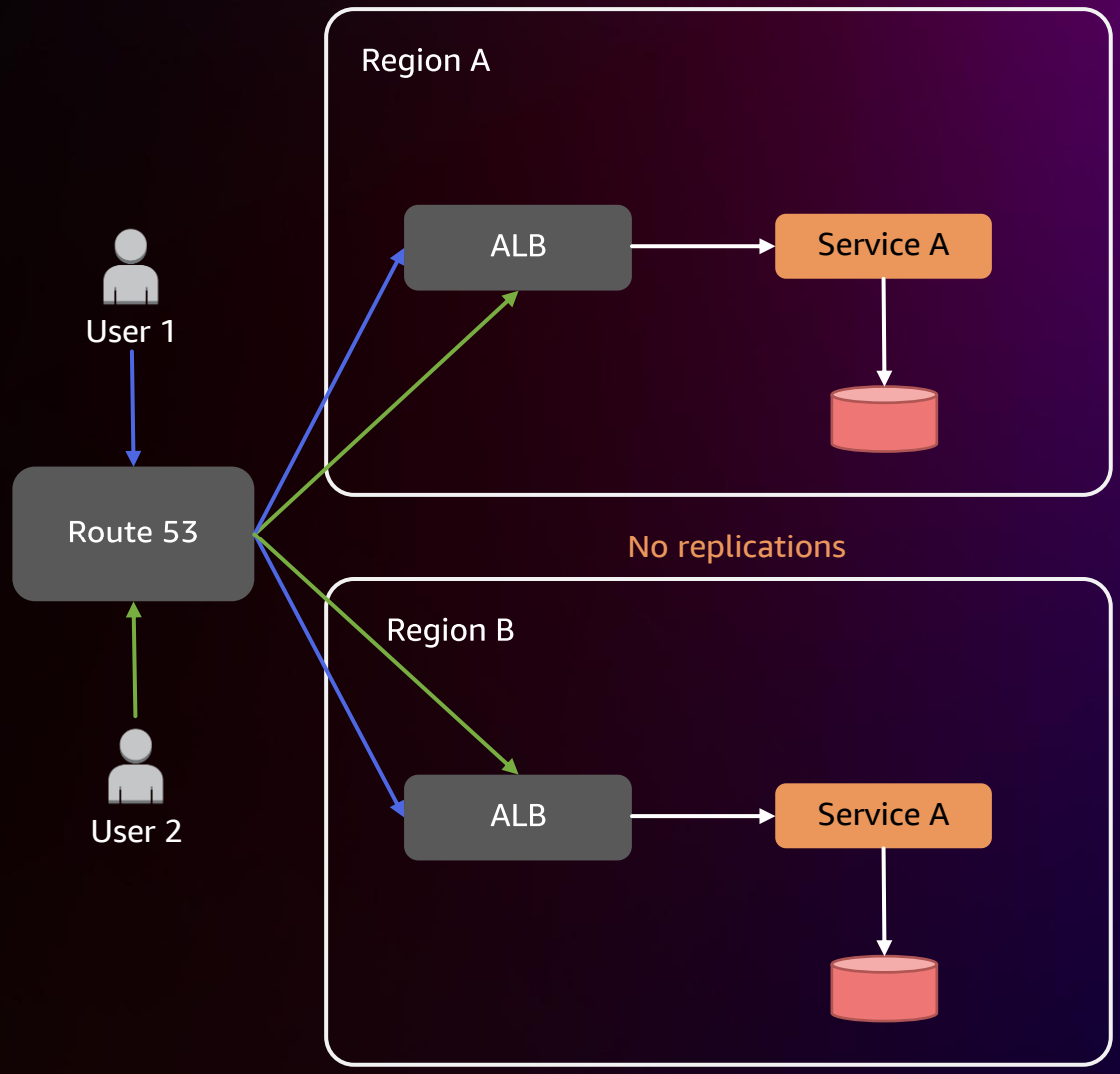
- Use cases
  - Read-heavy workloads
  - Strong read after write consistency
  - Regional failover/failback for DR/operational continuity
  - RPO near 0; regulatory requirements
- Key design considerations
  - Observability (health checks)
  - CI/CD, code and configuration drifts
  - Routing strategy
  - Failover/failback procedures (writer)
  - Increased latency – performance



Pattern #2.4: Active/active

DUAL WRITES

- Use cases
  - Read-heavy workloads
  - Strong read after write consistency
  - Regional failover/failback for DR/operational continuity
  - RPO near 0; regulatory requirements
- Key design considerations
  - Observability (health checks)
  - CI/CD, code and configuration drifts
  - Routing strategy
  - Idempotency
  - Managing data atomicity at the client side (highly complex)

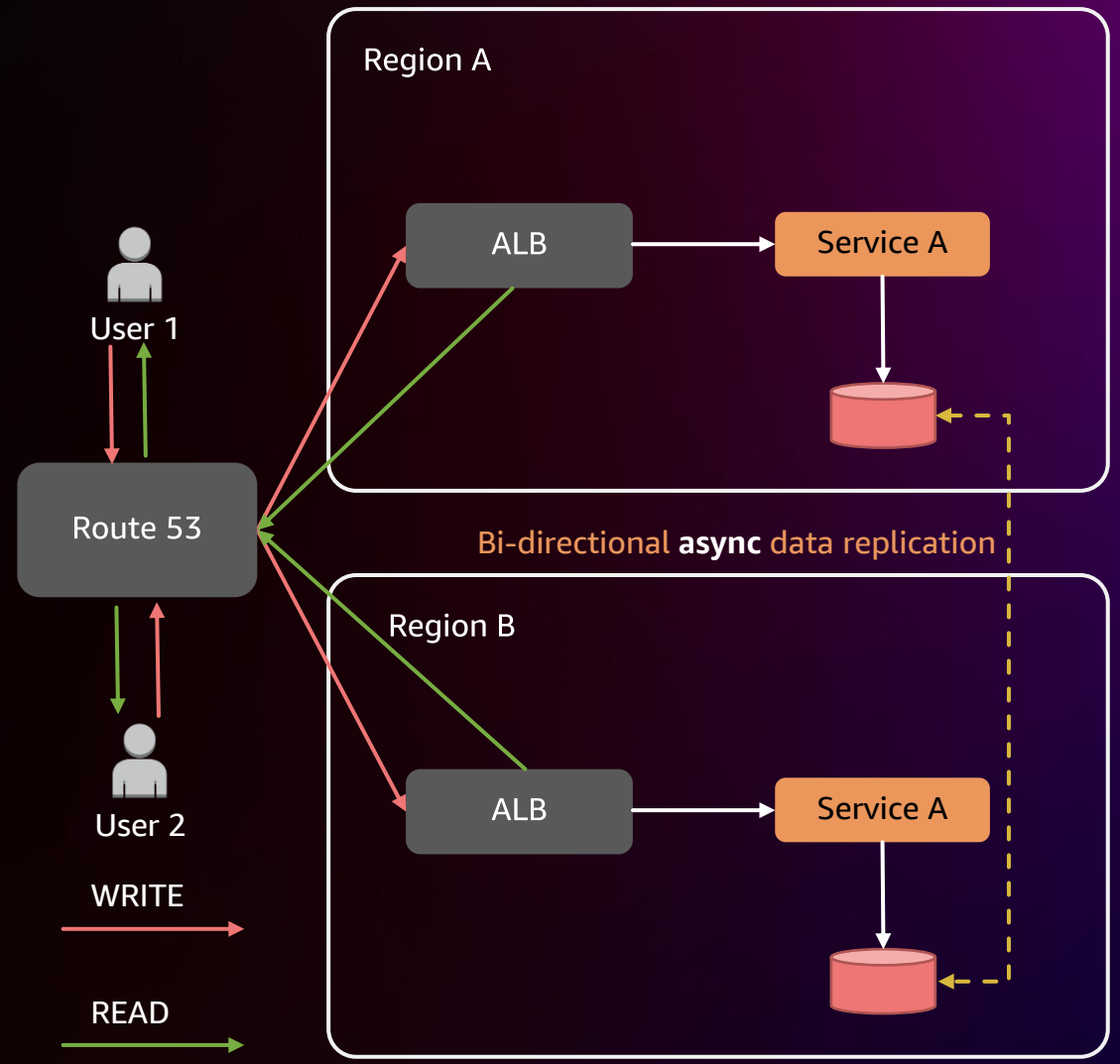


Pattern #2.5: Active/active

MULTIPLE WRITERS; MULTIPLE READERS

- Use cases
  - High availability
  - Latency sensitivity
  - Eventual consistency – async replication
  - Read/write-heavy workloads – scalability
- Key design considerations
  - Observability (health checks)
  - CI/CD, code and configuration drifts
  - Routing strategy
  - Conflict resolution
  - Read after write consistency

Amazon DynamoDB



Routing patterns

Amazon Route 53
(Application Recovery Controller)

Amazon CloudFront
(Lambda@Edge)

AWS Global Accelerator

Key takeaways

- Do not start with multi-Region as a starting point for resilience
- Multi-Region is not just a technology decision – it’s a business decision
- Build a business case first
- Don’t forget about dependencies – internal or external
- Use Regions for what they are designed for – independent failure domains
- Build deep observability – do you know when to press the big red button?
- Multi-Region can be complex – choose the right patterns for right reasons

Thank you!

John Formento, Jr.
https://www.linkedin.com/in/johnformentojr/

Neeraj Kumar
https://www.linkedin.com/in/neerajkumar78/

Barry P. Sheward
https://www.linkedin.com/in/barrysheward/

Please complete the session survey in the **mobile app**