DR in GCP

Memi Lavi
www.memilavi.com

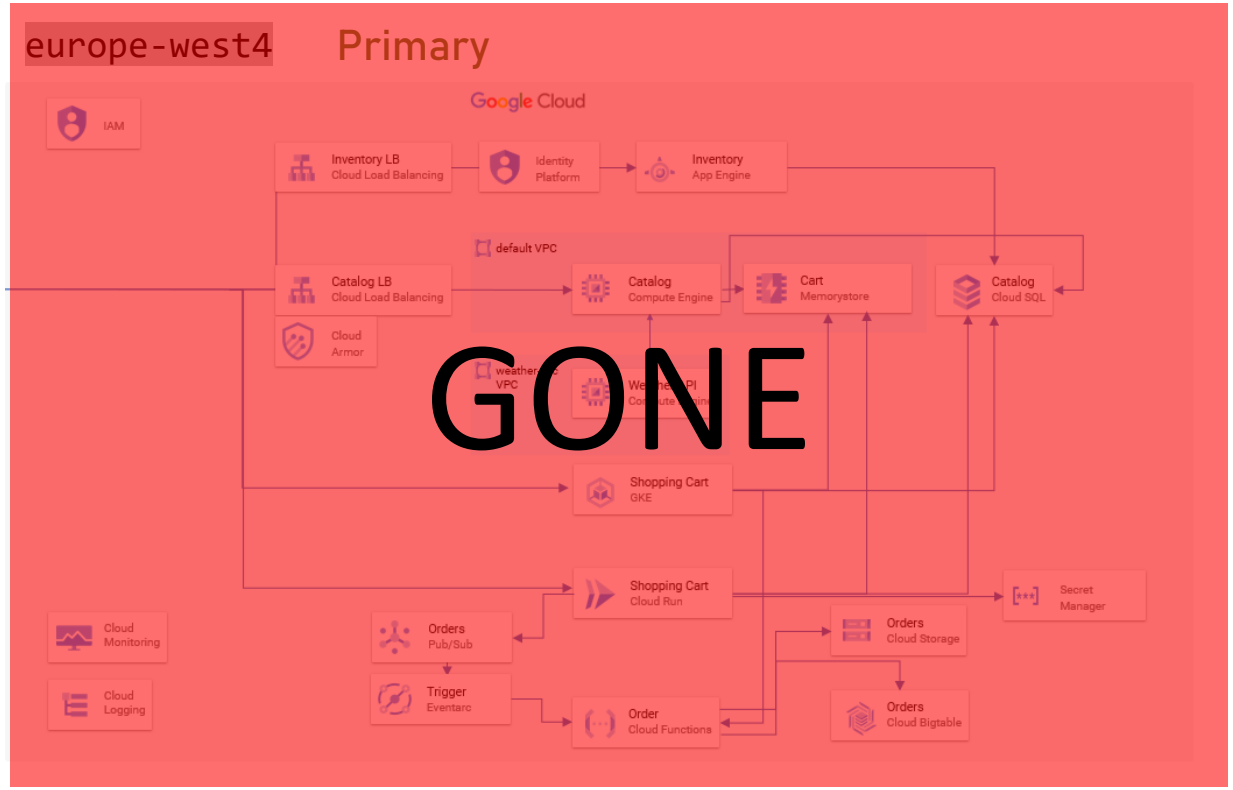
DR

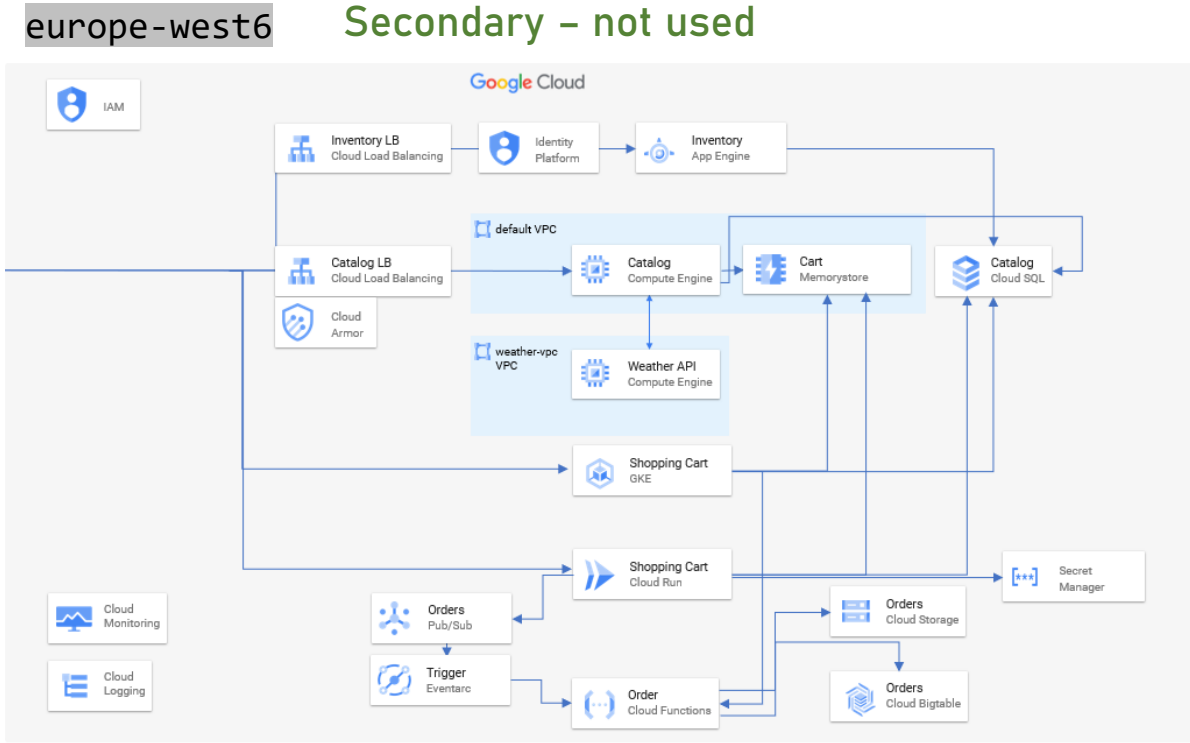
- Disaster Recovery
- A plan to recover from a complete shutdown of a Region
  - Usually as a result of a disaster (earthquake, flood, etc)
- Some apps require it, some don't
- Might have substantial cost aspects
- Remember: A complete shutdown of a Region is extremely rare

How DR Works?

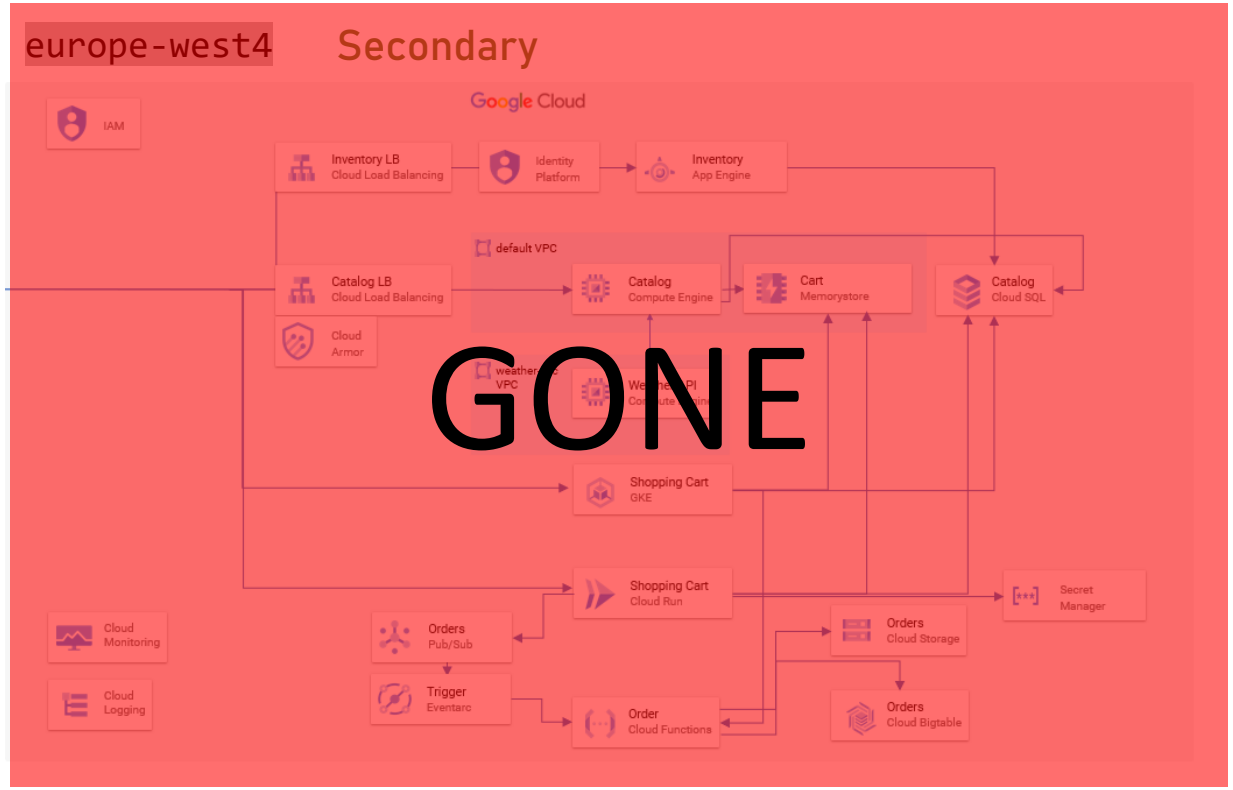
- In order to set up DR, we need to do the following:
  - Select a DR site
    - A secondary Region that will function as our primary in case of a disaster
    - Configure it to be ready for activation when necessary

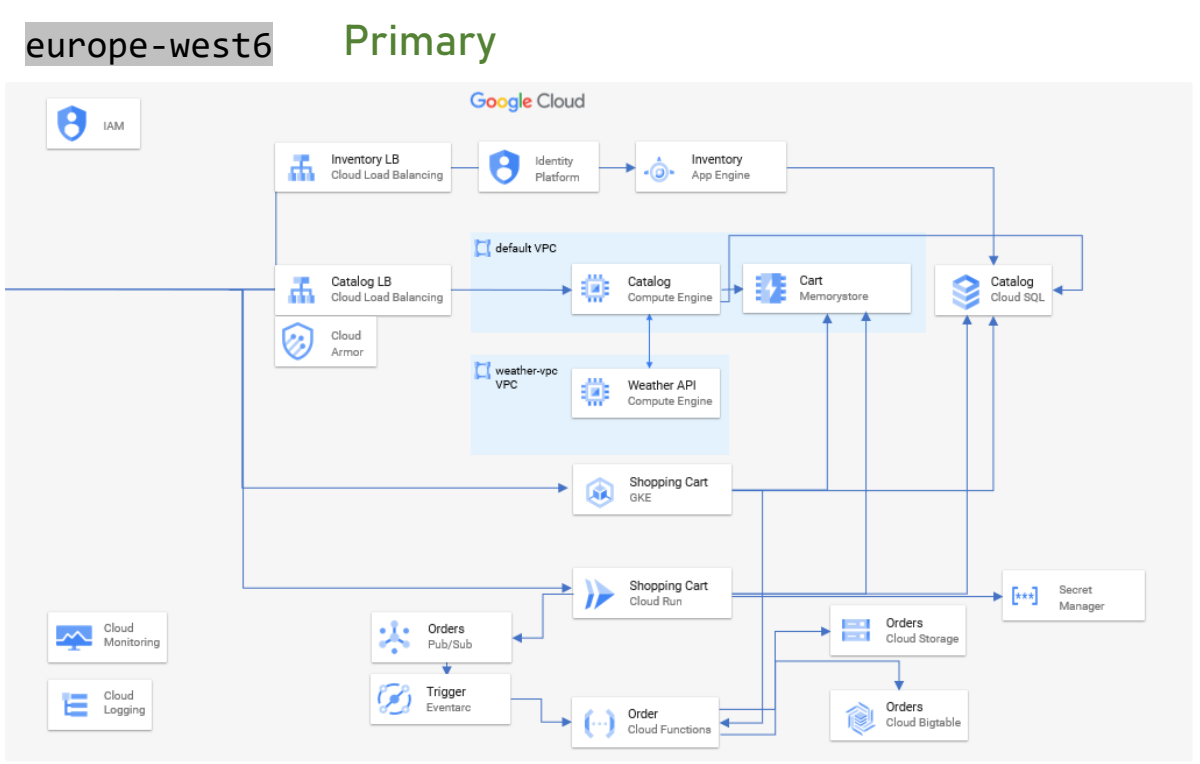
How DR Works?


europe-west4
Primary
Google Cloud
IAM
Inventory LB
Cloud Load Balancing
Identity Platform
Inventory
App Engine
default VPC
Catalog LB
Cloud Load Balancing
Catalog
Compute Engine
Cart
Memorystore
Catalog
Cloud SQL
Cloud Armor
GONE
Shopping Cart
GKE
Shopping Cart
Cloud Run
Secret Manager
Cloud Monitoring
Orders
Pub/Sub
Orders
Cloud Storage
Cloud Logging
Trigger
Eventarc
Order
Cloud Functions
Orders
Cloud Bigtable


europe-west6
Secondary – not used
Google Cloud
IAM
Inventory LB
Cloud Load Balancing
Identity Platform
Inventory
App Engine
default VPC
Catalog LB
Cloud Load Balancing
Catalog
Compute Engine
Cart
Memorystore
Catalog
Cloud SQL
Cloud Armor
weather-vpc
VPC
Weather API
Compute Engine
Shopping Cart
GKE
Shopping Cart
Cloud Run
Secret Manager
Cloud Monitoring
Orders
Pub/Sub
Orders
Cloud Storage
Cloud Logging
Trigger
Eventarc
Order
Cloud Functions
Orders
Cloud Bigtable

How DR Works?


europe-west4
Secondary
Google Cloud
IAM
Inventory LB
Cloud Load Balancing
Identity Platform
Inventory
App Engine
default VPC
Catalog LB
Cloud Load Balancing
Catalog
Compute Engine
Cart
Memorystore
Catalog
Cloud SQL
Cloud Armor
GONE
Shopping Cart
GKE
Shopping Cart
Cloud Run
Secret Manager
Cloud Monitoring
Orders
Pub/Sub
Orders
Cloud Storage
Cloud Logging
Trigger
Eventarc
Order
Cloud Functions
Orders
Cloud Bigtable


europe-west6
Primary
Google Cloud
IAM
Inventory LB
Cloud Load Balancing
Identity Platform
Inventory
App Engine
default VPC
Catalog LB
Cloud Load Balancing
Catalog
Compute Engine
Cart
Memorystore
Catalog
Cloud SQL
Cloud Armor
weather-vpc
VPC
Weather API
Compute Engine
Shopping Cart
GKE
Shopping Cart
Cloud Run
Secret Manager
Cloud Monitoring
Orders
Pub/Sub
Orders
Cloud Storage
Cloud Logging
Trigger
Eventarc
Order
Cloud Functions
Orders
Cloud Bigtable

DR Concepts

- Hot / Cold

Hot

- Failover to secondary site happens automatically with no downtime
- No data loss
- Requires duplicate infrastructure
- The most expensive method

Cold

- Failover to secondary site takes some time
- Might be manual
- Some data might be lost
- Less expensive

DR Concepts

- Hot or Cold – how to decide?
- Depends on the system's requirements
- A global ecommerce website, serving million of customers – probably Hot
- An HR app for the organization – definitely Cold (if at all…)

DR Concepts

- RPO / RTO

RPO

- Recovery Point Objective
- How much data we allow ourselves to lose in case of a disaster
- Usually measured in minutes
- In other words – what's the frequency of data sync to the secondary region
- Example: We have an RPO of 5 minutes

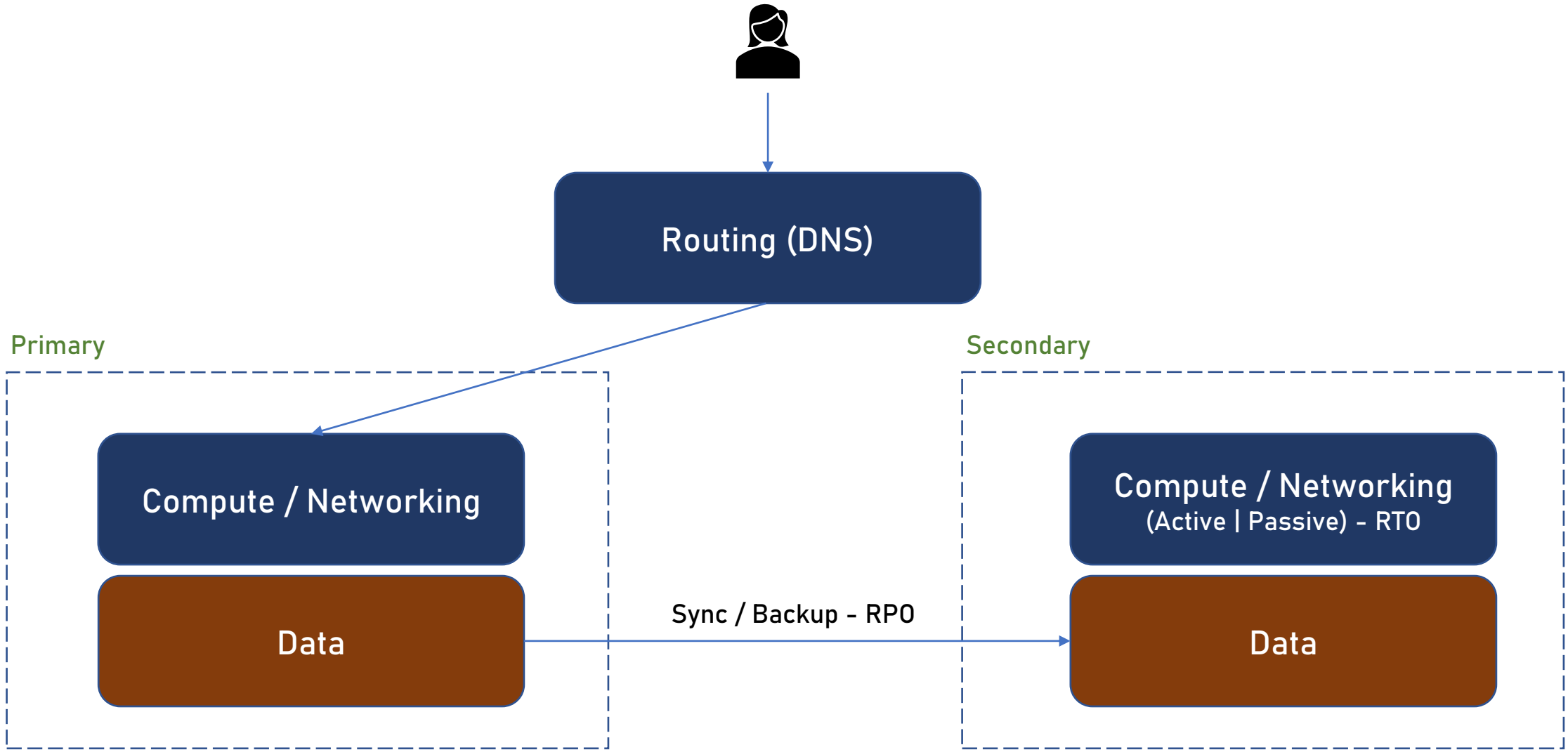
RTO

- Recovery Time Objective
- How much downtime we can tolerate in case of a disaster
- Usually measured in minutes
- In other words – how long it should take before the system is up again
- Not necessarily with the most up to date data, depends on the RPO

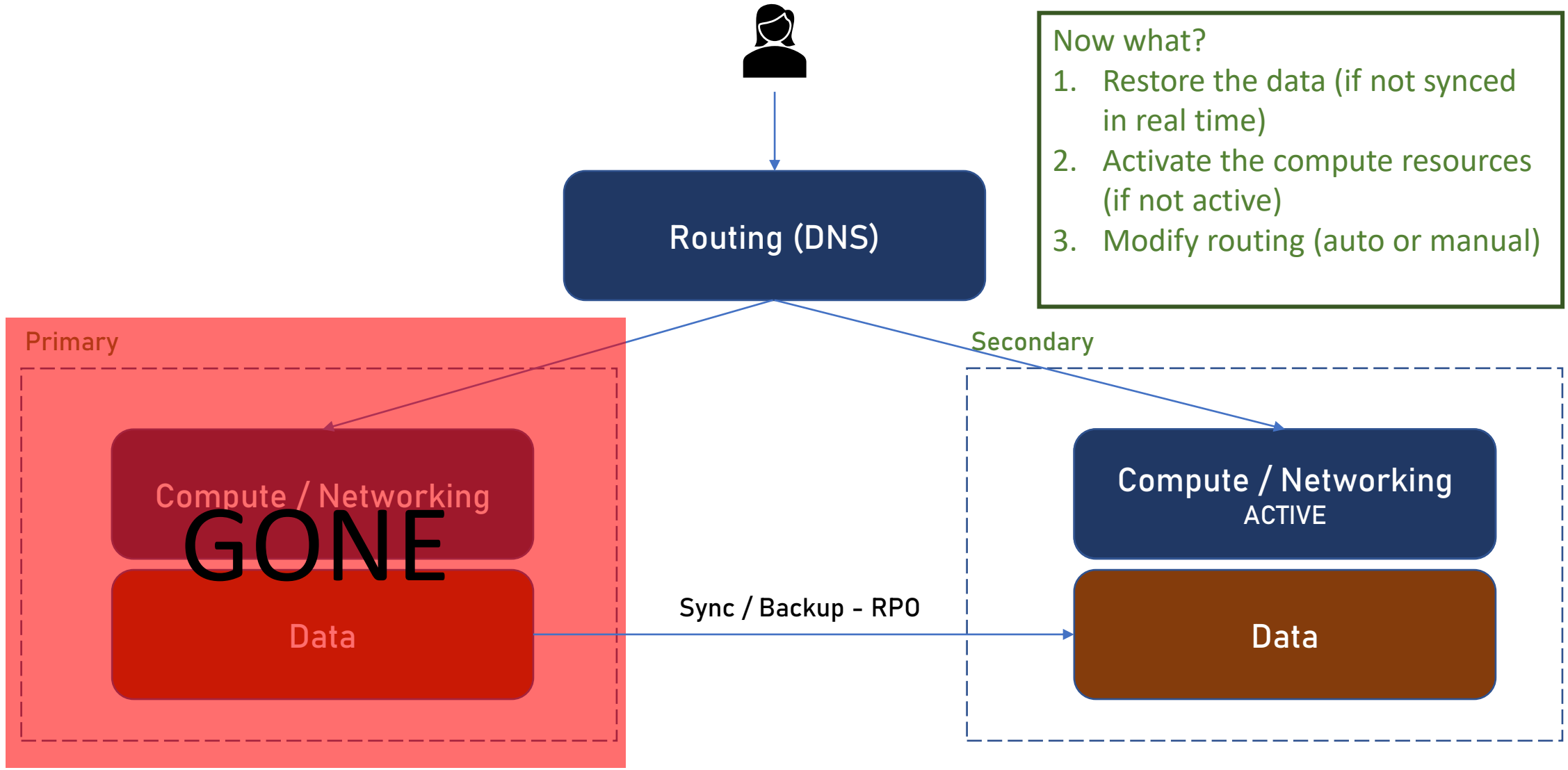
DR Concepts

- RPO and RTO– how to decide?
- Depends on the system's requirements
- A massive reporting system will probably go for low RPO, but can compromise on the RTO
- A global chat will focus on RTO

Basics of DR Implementation



Basics of DR Implementation



DR of Data in GCP

- Main question when designing the DR of data is:

What is the RPO?

(Or – how much data loss do we tolerate?)

DR of Data in GCP

- If RPO = 0 (no data loss in case of disaster):
  - We need database that always syncs with the secondary region
  - Supported databases:


Cloud SQL (with cross-region replicas)


BigTable (with replication enabled)


AlloyDB (with cross-region replicas)


Cloud Storage (with dual-region or multi-region bucket)


Spanner (with multi-region configuration)

DR of Data in GCP

- If RPO > 0 (some data can be lost):
  - Ensure DB's backup frequency is compliant with the RPO
  - The backup storage should be in a different region

DR of Data in GCP

- Example – Cloud SQL MySQL:

Automated backups

Automated backups are taken daily, within a 4-hour backup window. The backup starts during the backup window. When possible, schedule backups when your instance has the least activity.

During the backup window, automated backups occur every day your instance is running. One additional automated backup is taken after your instance is stopped to safeguard all changes prior to the instance stopping. Up to seven most recent backups are retained, by default. You can configure how many automated backups to retain, from 1 to 365. Backup and transaction log retention values can be changed from the default setting. Learn more.

Default backup locations

If you do not specify a storage location, your backups are stored in the multiregion that is geographically closest to the location of your Cloud SQL instance. For example, if your Cloud SQL instance is in `us-central1`, your backups are stored in the `us` multi-region by default. However, a default location like `australia-southeast1` is outside of a multi-region. The closest multi-region is `asia`.

DR of Data in GCP

- Note that:
  - Backup-based solutions are usually much cheaper
  - No additional active instance of the database is needed
  - Only storage for the backup

DR of Compute in GCP

- Main question when designing the DR of compute is:

What is the RTO?

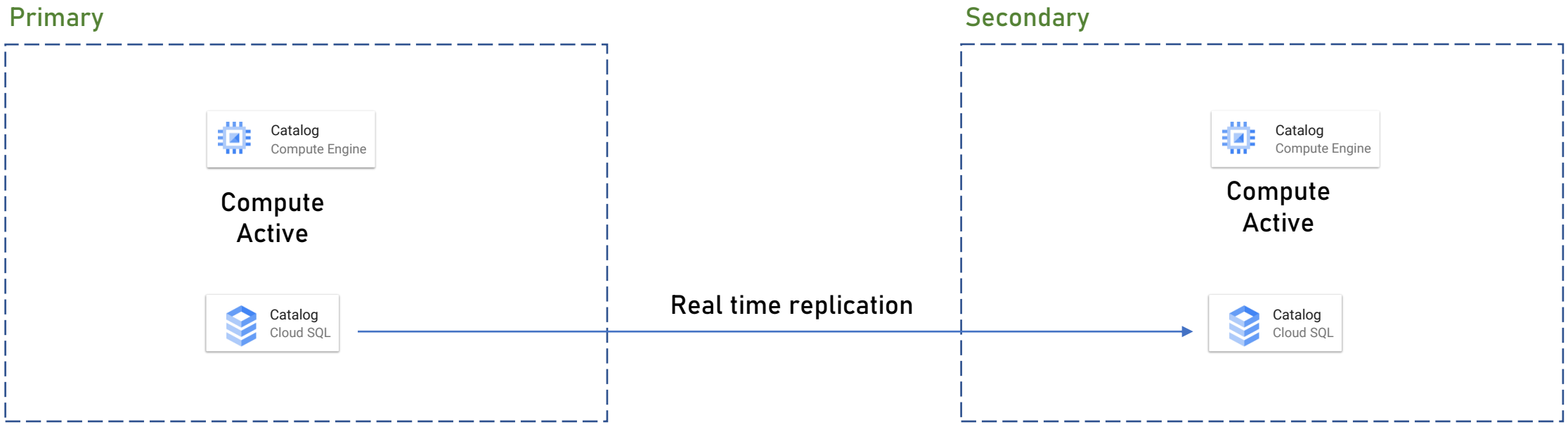
(Or – how much downtime can we tolerate?)

DR of Compute in GCP

- If RTO = 0 (no downtime in case of disaster):
  - Compute in secondary region should always be up and running

DR of Compute in GCP

RTO = 0

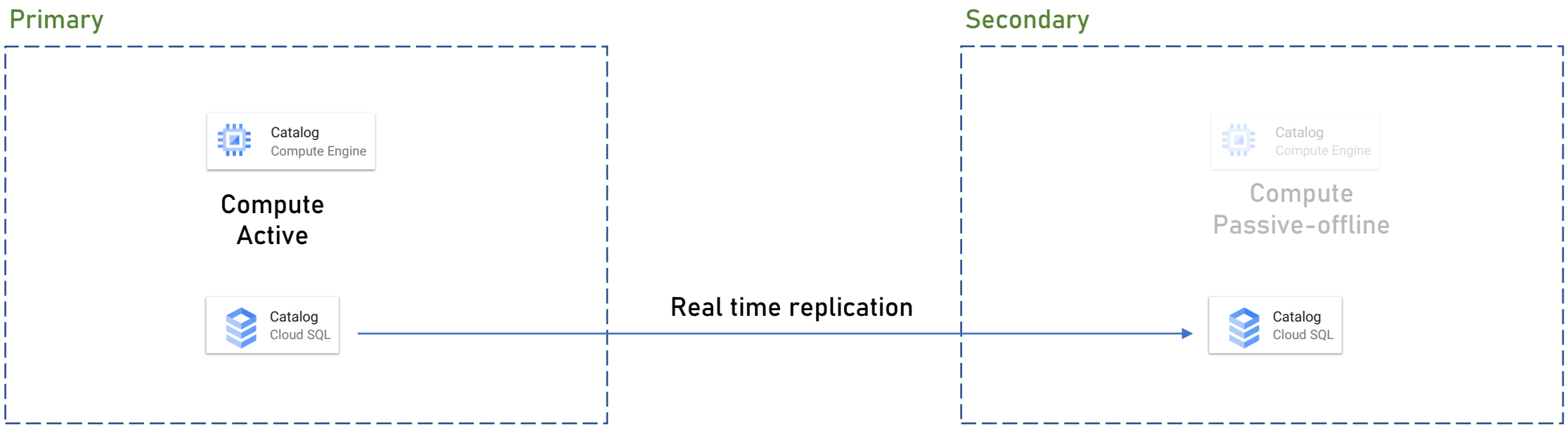


DR of Compute in GCP

- If RTO > 0 (some downtime is tolerated):
  - Either:
    - Have non-active (passive) compute on standby in secondary region
    - Create the compute when disaster occurs in secondary region

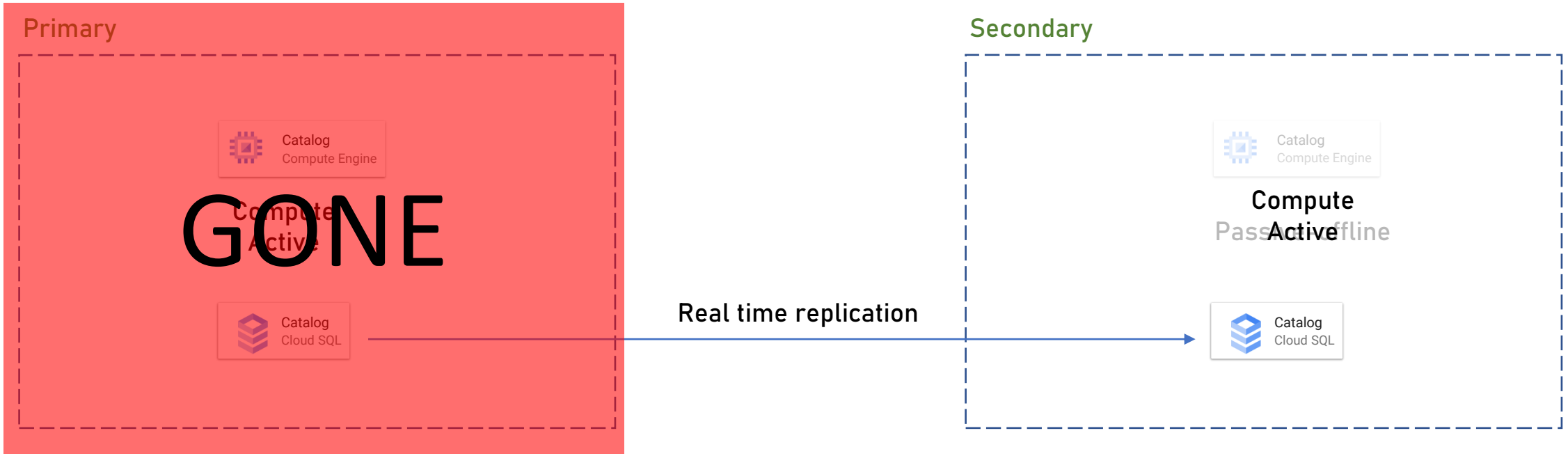
DR of Compute in GCP

RTO > 0



DR of Compute in GCP

RTO > 0



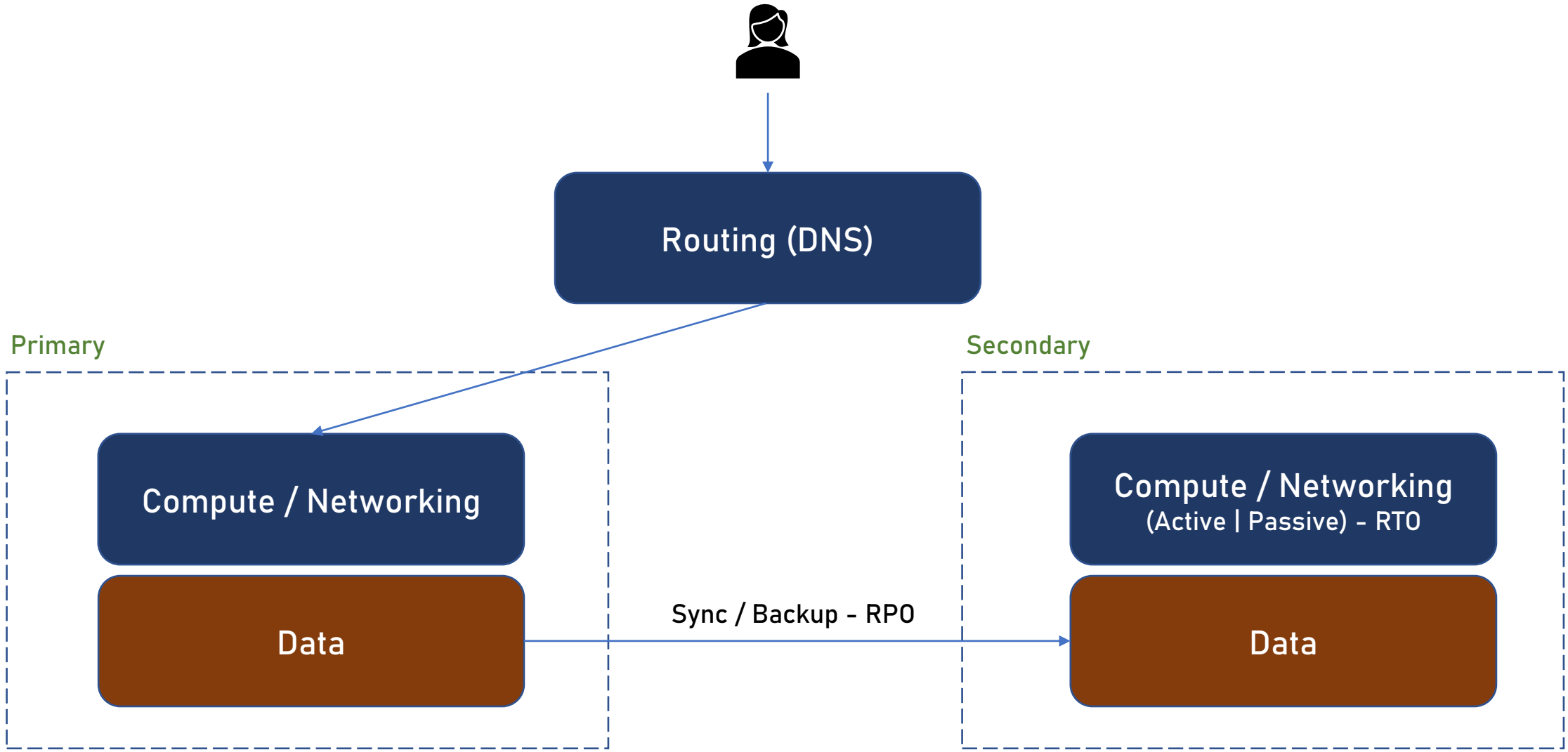
DR of Compute in GCP

- Note that:
  - The second example is much cheaper, no secondary active compute is needed when primary is active

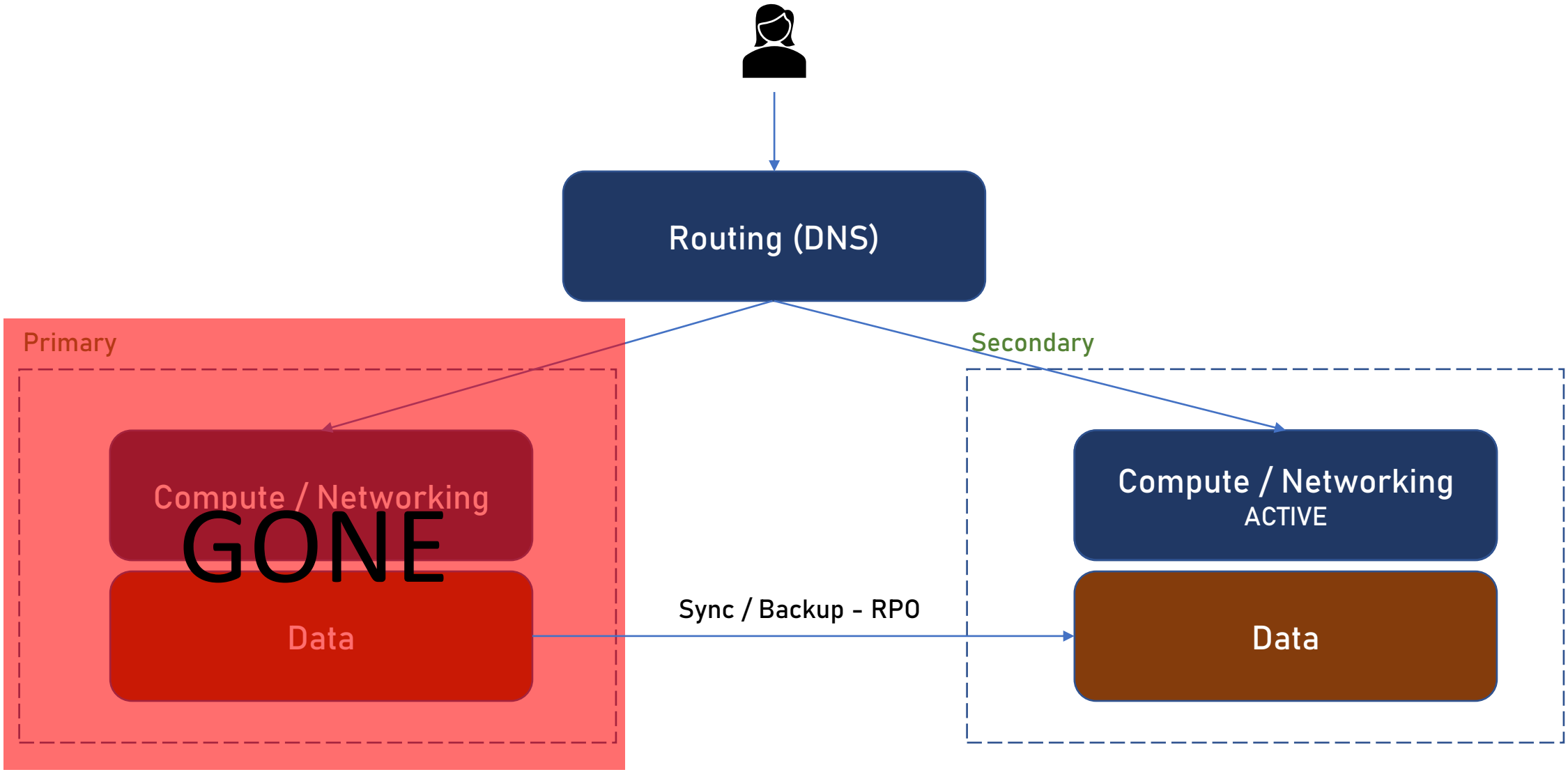
Routing in DR

- During DR users should be routed to the secondary region

Basics of DR Implementation



Basics of DR Implementation



Routing in DR

- Major three methods:
  - Inform the users about the new address of the app (in the secondary region)
  - Manually change DNS record to point to the secondary region
  - Use automatic routing

Routing in DR

- Global External Load Balancer is the best way to handle automatic routing in GCP
- Can have backends in multiple regions
- Automatic health checks
- Easy to manage