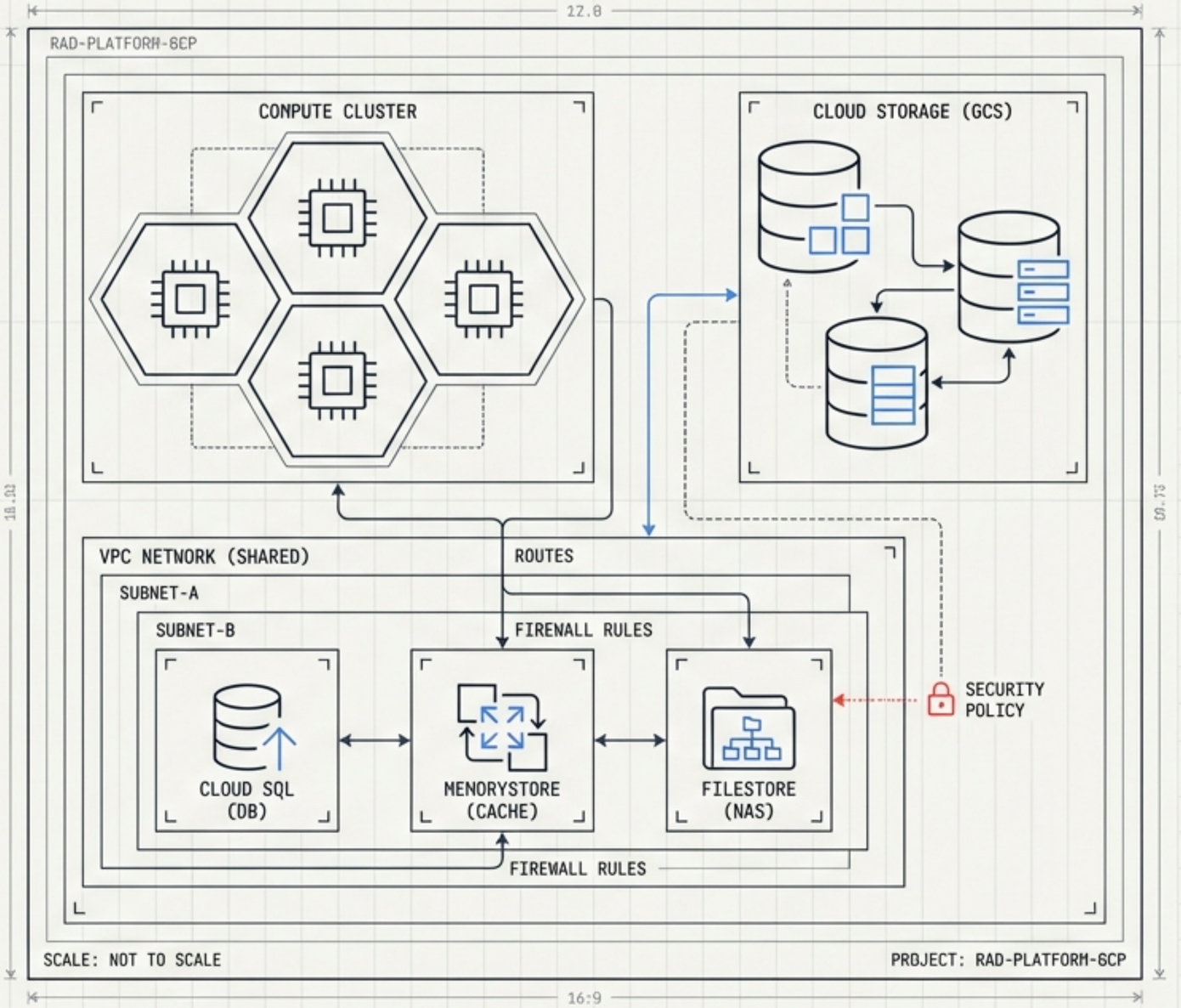
RAD Platform: GCS Services Module

The Foundational Infrastructure Layer

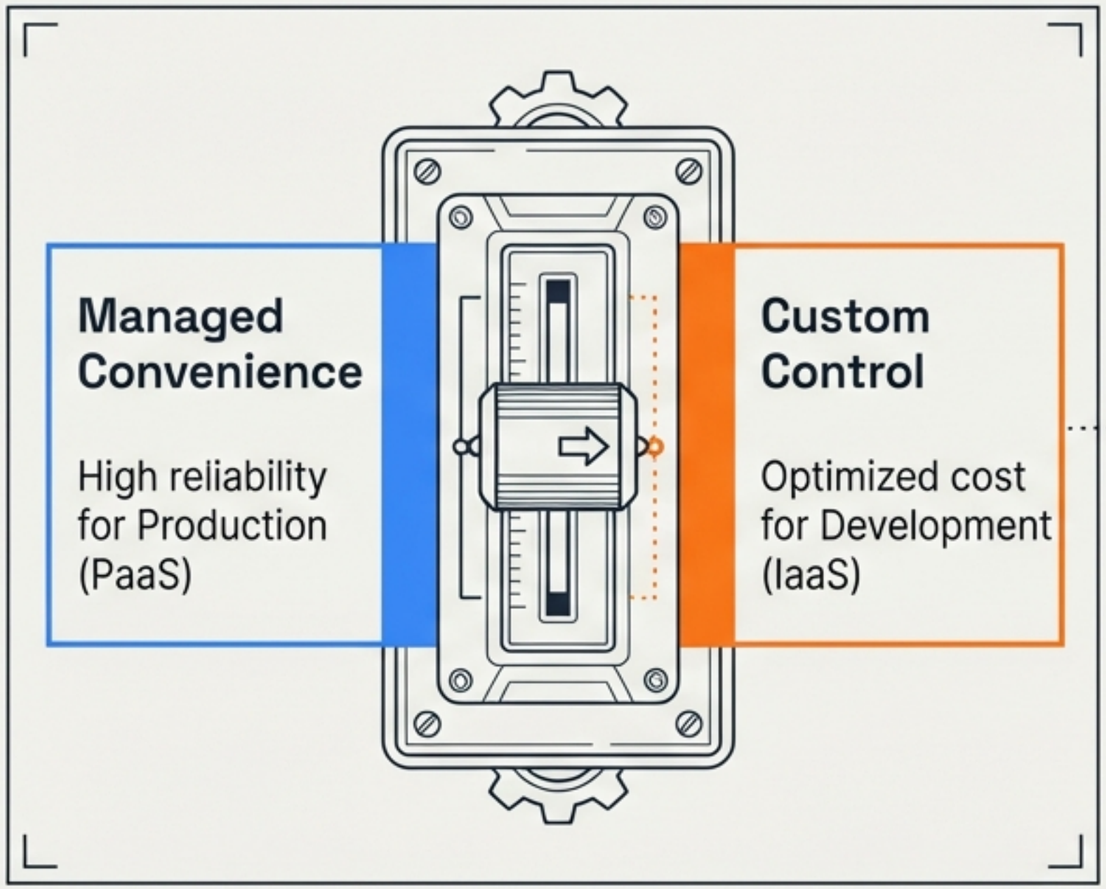
A Deep Dive Analysis of modules/GCP_Services.

MODULE CONTEXT: Shared environment hosting core networking, database, caching, and file storage services for application dependencies.



The Foundation of the RAD Platform

The Strategy



The Capabilities

Networking:
Secure private perimeter.

Databases:
`Cloud SQL (Postgres/MySQL)`

Caching:
`Redis (Memorystore or Custom VM)`

Storage:
`NFS (Filestore or Custom VM)`

Core Concept: A shared Foundation Layer providing core utilities.

Identity & Access Management Strategy

Identity Cards

The App Runner
Inter

ID Code: `cloudrun-sa`

Identity for Cloud Run application containers.

- Connects to Cloud SQL (`roles/cloudsql.client`)
- Manages GCS objects (`roles/storage.objectAdmin`)
- Accesses secrets (`roles/secretmanager.secretAccessor`)
- Routes traffic via Serverless VPC Access

The Builder
Inter

ID Code: `cloudbuild-sa`

CI/CD Pipeline Identity (High Privilege).

- Deploys apps (`roles/run.admin`)
- Manages artifacts
- Acts as other SAs (`roles/iam.serviceAccountUser`)
- Manages encryption keys (`roles/cloudkms.admin`)

The Utility Worker
Inter

ID Code: `nfsserver-sa`

Identity for custom NFS GCE instance.

- Writes logs (`roles/logging.logWriter`)
- Manages own instance (`roles/compute.instanceAdmin.v1`) for self-healing

System Agents & Secret Operations

The Automators (System Agents)

Service Networking Agent

Grants `roles/servicenetworking.serviceAgent`

Allows Private Service Connect peering.

VPC

Cloud Run Agent

Ensures serverless containers attach to VPC

Permission: `roles/compute.networkUser`

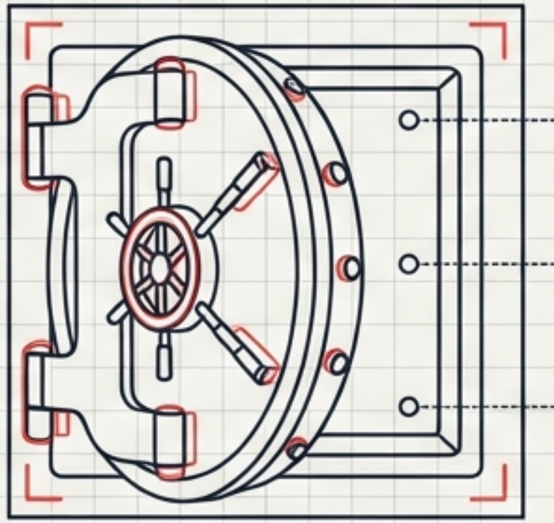
The Vault (Google Secret Manager)

App
(cloudrun-sa)

Explicit Access Only

Build
(cloudbuild-sa)

Explicit Access Only

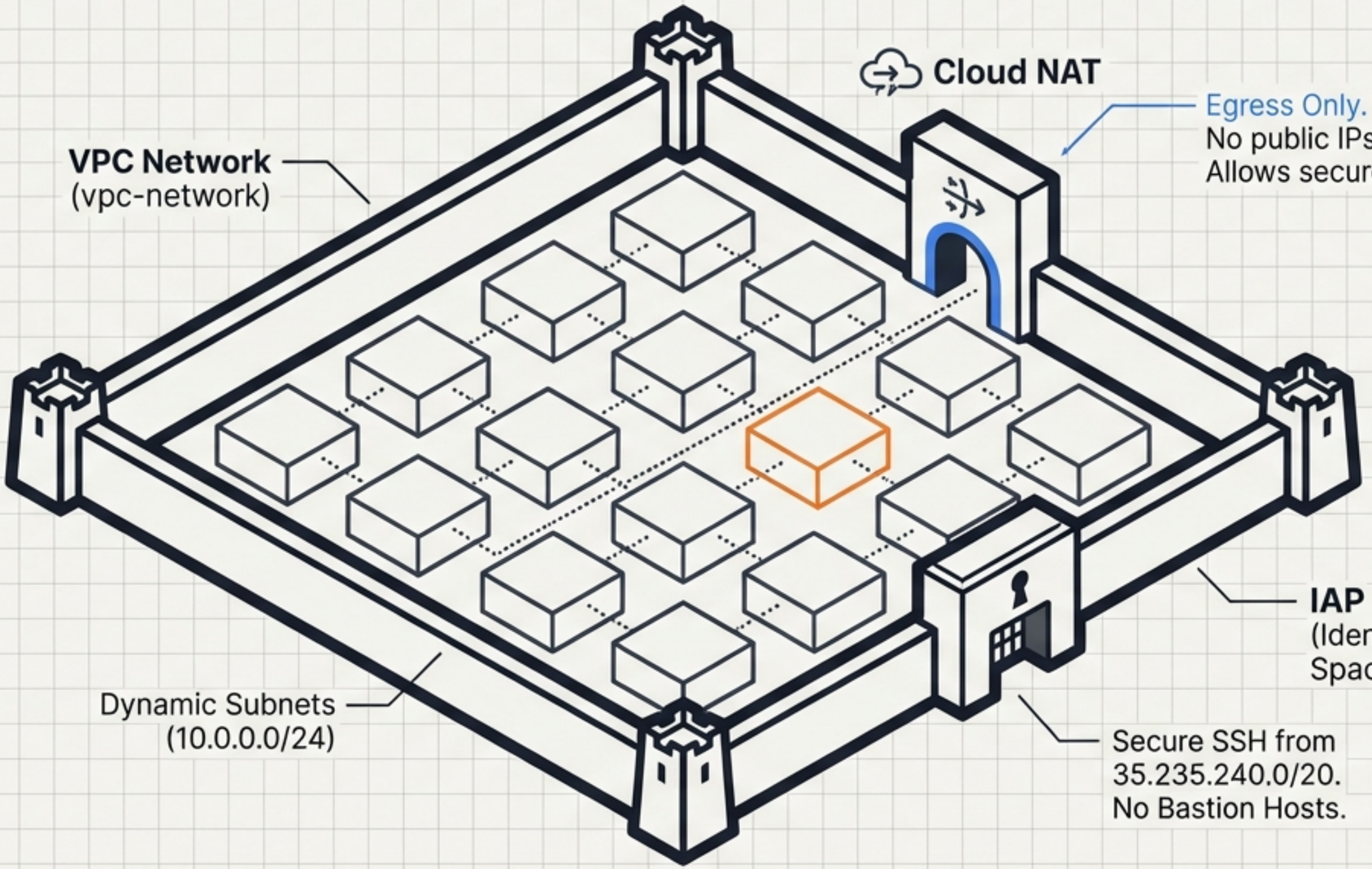


Cloud SQL Root Password
(*-root-password)

Redis Host
(redis-host-*)

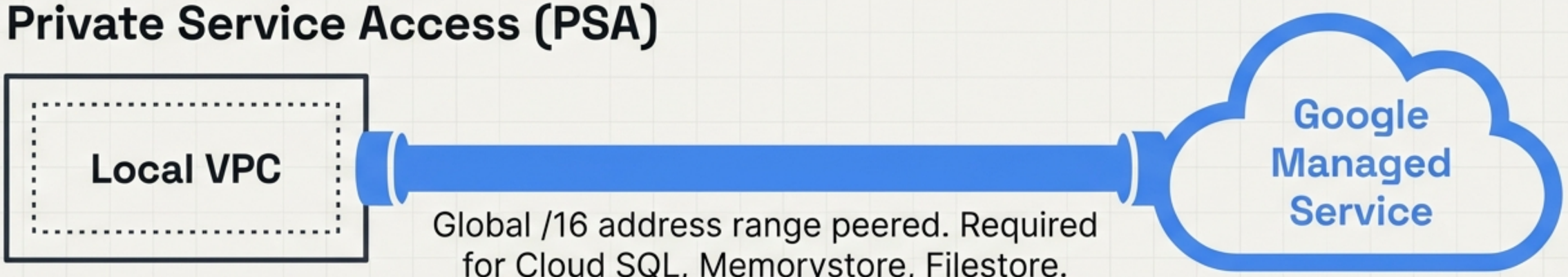
Redis Port
(redis-port-*)

Network Architecture: The Private Perimeter



Connectivity & Firewall Logic

Private Service Access (PSA)



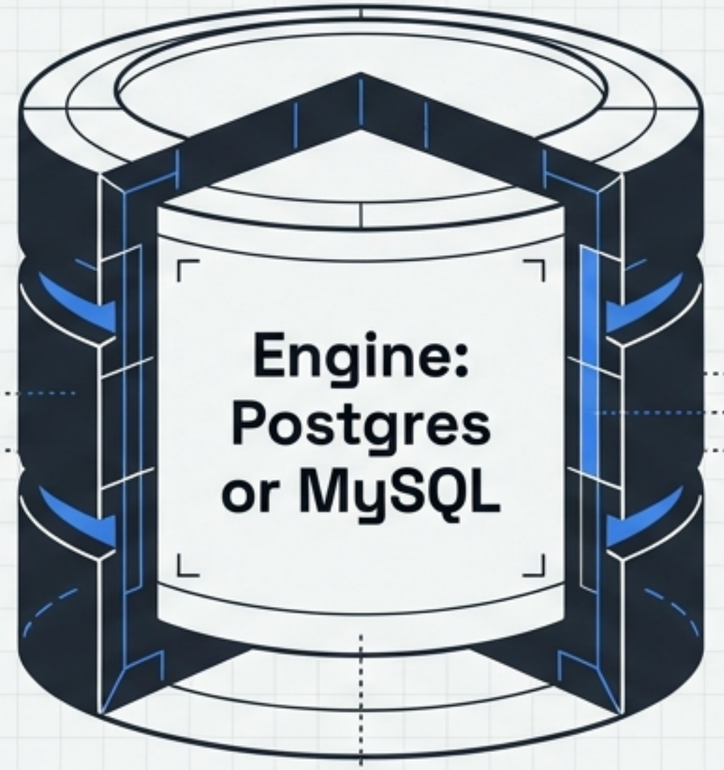
Firewall Rules (The Gatekeepers)

1. Internal Traffic	→ Allow TCP/UDP/ICMP within VPC ranges.
2. Load Balancers	→ Allow Ingress from 35.191.0.0/16 & 130.211.0.0/22.
3. Service Specific	→ NFS: Port 2049 (tag: nfsserver) → Redis: Port 6379 (tag: redisserver)

Database Strategy: Cloud SQL

Dev Mode
Zonal Availability

Prod Mode
Regional Availability (HA)



Security
Private IP Only. No Public Internet Exposure.

Ops & Reliability

Storage
SSD with disk_autoresize = true
Inter, `JetBrains Mono`

Backups
Automated daily (04:00 start), 7-day retention

Recovery
Point-in-Time Recovery (PITR) for Postgres

Caching Strategy: Redis Two Ways

Option A: Managed (The Standard)

Cloud Memorystore (redis.tf)

```
Trigger: var.create_redis = true
```

- Tiers: Basic or Standard HA
- Managed maintenance windows
- Best for Production

Option B: Custom (The Cost-Saver)

Redis on VM (nfs.tf)

Architecture: Co-located on NFS server to minimize compute costs.

- Configuration:
  Binds to 0.0.0.0:6379 (Firewall Protected)
- Persistence: RDB snapshots only (AOF disabled)

File Storage: NFS Two Ways

Option A: Managed Filestore

Filestore (filestore.tf)

```
Trigger: var.create_filestore_nfs = true
```

- Standard GCP Filestore (Basic HDD/SSD)
- Standard Managed Offering

Option B: Custom NFS Server

Custom VM (nfs.tf)

Architecture: Managed Instance Group (MIG) Size 1

OS: Ubuntu 22.04 LTS

Reliability Features

- Self-Healing: Auto-recreates VM if TCP 2049 Health Check fails
- Backups: Daily snapshots of Zonal Persistent Disk (7-day retention)

The Architectural Toolbox

Secure Networking

Private-only subnetting, Cloud NAT, IAP access.

DB Reliability

Cloud SQL with auto-backups and auto-resizing.

Cost Ops

Two-in-one Custom VM option for Redis & NFS (e2-small capable).

Self-Healing

Managed Instance Group (MIG) auto-healing for custom servers.

Secret Ops

Credentials auto-generated in Secret Manager.

Deployment Ready

Pre-configured Service Accounts for App and Build.

Roadmap: Security Hardening

From Secure to Fortified

Enhancement 1: SSL Enforcement

Current State: Allows unencrypted fallback.

Goal: Set `ssl_mode = ENCRYPTED_ONLY` (Postgres) or `require_ssl` (MySQL).

Enhancement 2: Deletion Protection

Goal: Prevent accidental Terraform destruction of Production databases.

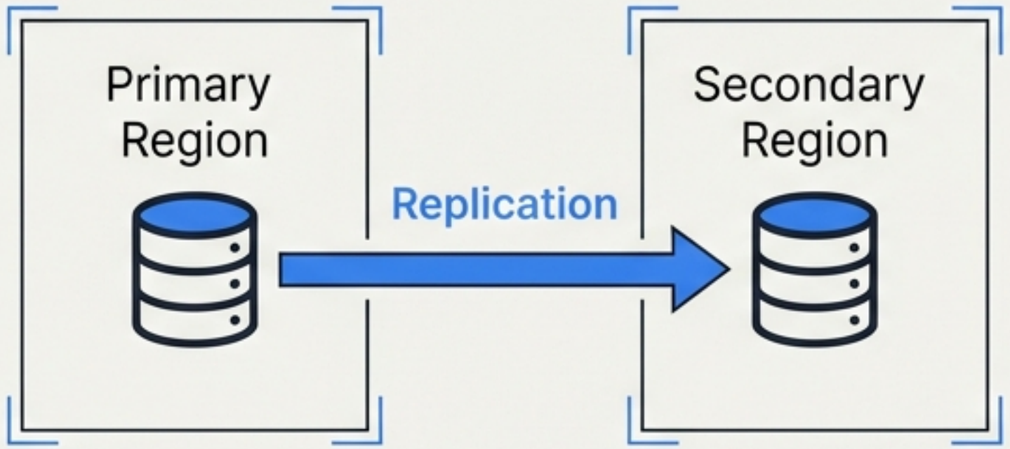
Enhancement 3: CMEK Support

Goal: Customer-Managed Encryption Keys for Cloud SQL and GCS buckets (High Compliance).

Roadmap: Reliability & Availability

Preparing for Enterprise Scale

Multi-Region SQL



Deploy Read Replicas in secondary regions for Disaster Recovery (Currently defaults to local).

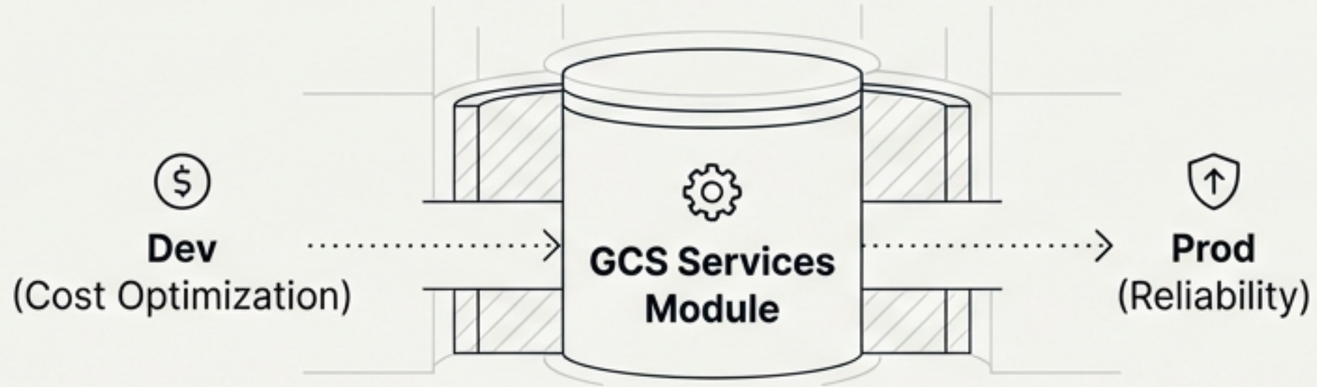
Enterprise Tier Upgrades

- Filestore: Upgrade to Enterprise (Regional availability).
- Memorystore: Upgrade to Standard Tier (Cross-zone replication & auto-failover).

Building for Scale

The GCS Services Module provides a production-ready, flexible foundation that adapts to the lifecycle of the application—optimizing for cost in dev and reliability in prod.

A secure, private, and self-healing infrastructure layer.



Next Steps

Review proposed Security and Reliability enhancements for the upcoming sprint.