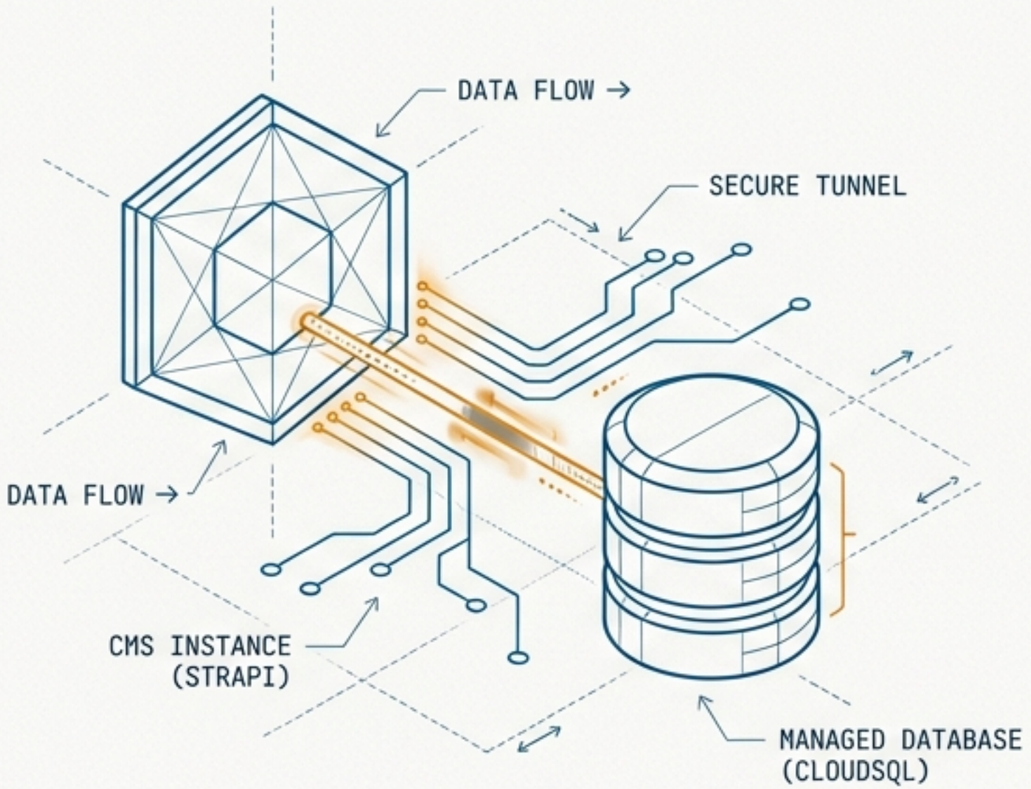


STRAPI ON GOOGLE CLOUD PLATFORM: RAD PLATFORM IMPLEMENTATION

Architectural Design, Security Posture, and Operational Strategy

strapi | INTEGRATION PARTNERS

VERSION 1.0 | CLASSIFICATION: TECHNICAL BLUEPRINT | DATE: 2024.10.27

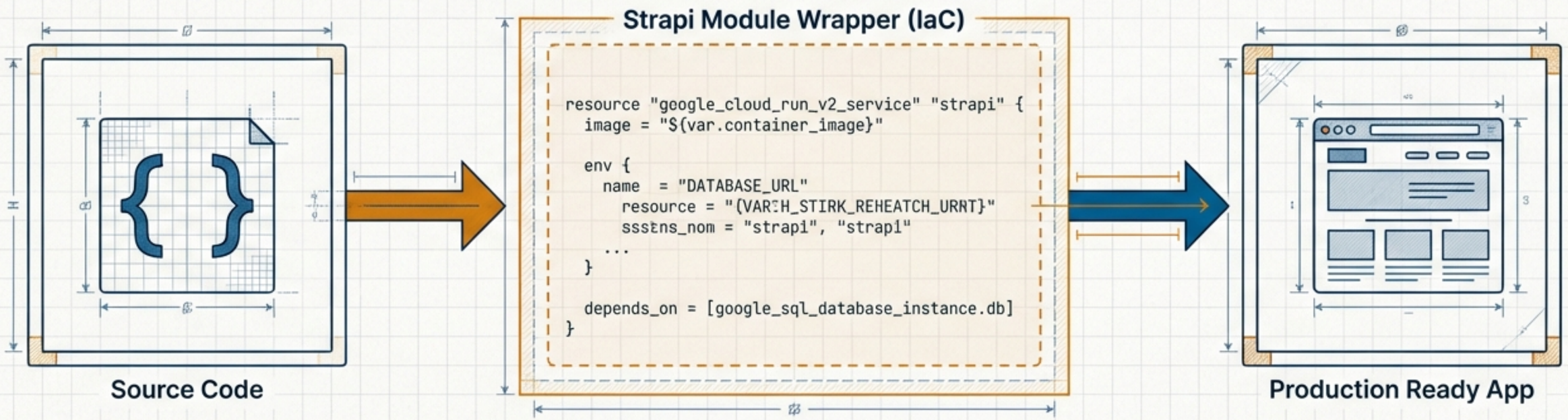
NotebookLM

Streamlining Headless CMS Deployment

The ‘modules/Strapi’ module functions as a specialized wrapper around the ‘modules/CloudRunApp’ foundation. It automates the transition from raw source code to a production-ready Headless CMS.

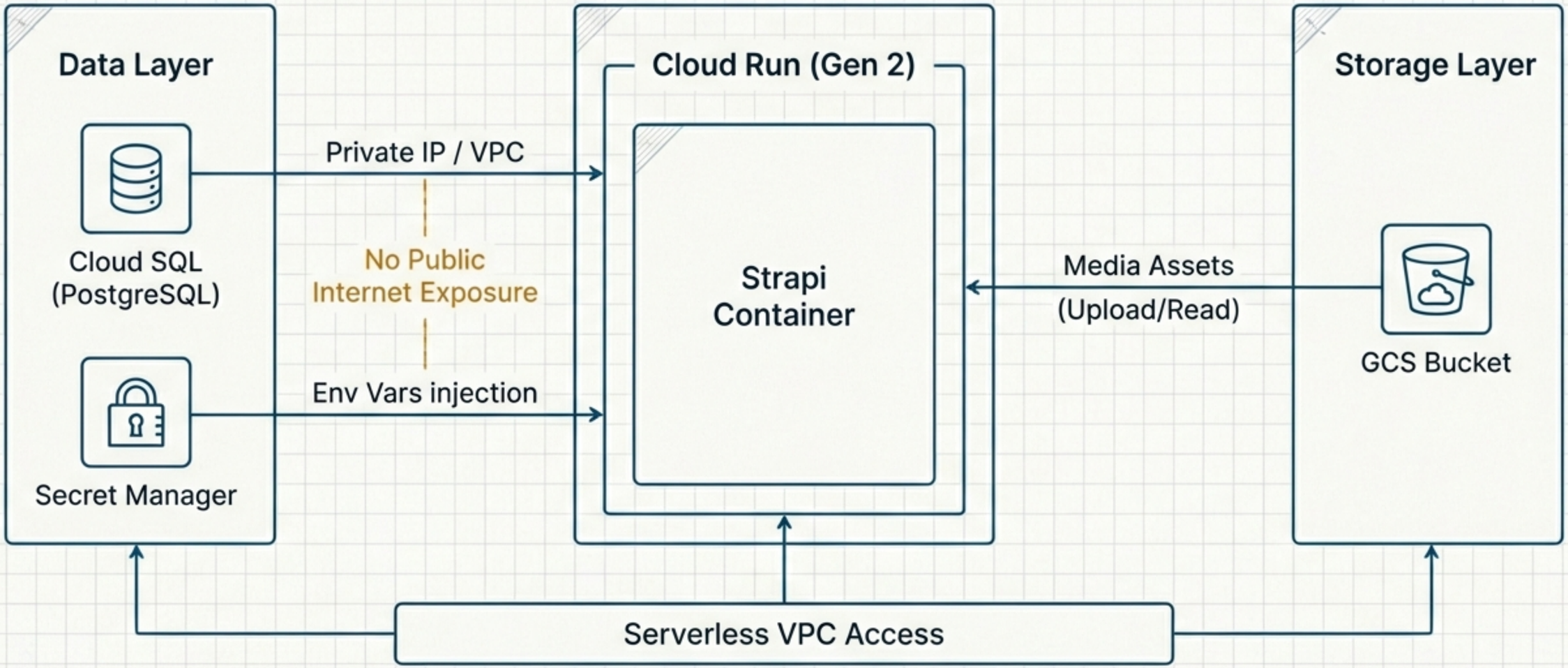
Key Specs

- **Compute:** Cloud Run Gen2
- **Database:** Postgres 15 (Cloud SQL)
- **Assets:** Google Cloud Storage (GCS)



NotebookLM

High-Level Architecture



NotebookLM

Application Hosting: Cloud Run

Configuration Card

RUNTIME:
Node:20-alpine

EXECUTION ENVIRONMENT:
Gen2 (Full Linux Capabilities)

PORT:
1337

RESOURCES:
1000m CPU / 1Gi Memory (Default)

MOUNT POINT:
/cloudsql (Unix Socket)

Native Cloud Run Autoscaling

NotebookLM

Data Persistence: SQL & Object Storage

Structured Data (Cloud SQL)

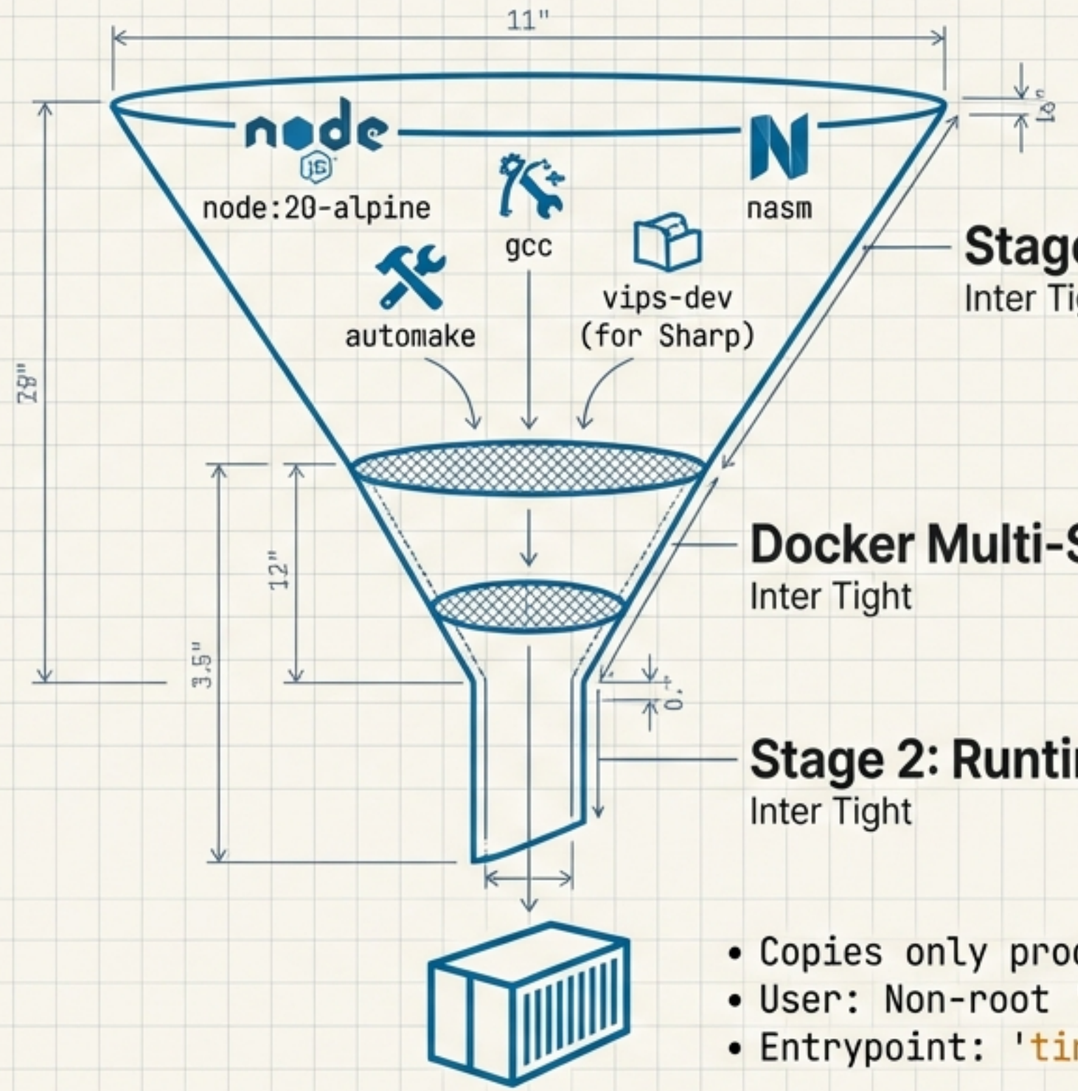
- Engine: PostgreSQL 15
- Network: Private IP via VPC
- Security: No public internet exposure
- Extensions: Configurable via Strapi defaults

Unstructured Assets (Cloud Storage)

- Resource: [project-id]-strapi-uploads
- Class: Standard Storage
- Security: Uniform Bucket Level Access
- Provider: @strapi-community/strapi-provider-upload-google-cloud-storage

NotebookLM

Container Build Optimization



Stage 1: Build (The Heavy Lift)
Inter Tight

Docker Multi-Stage Build
Inter Tight

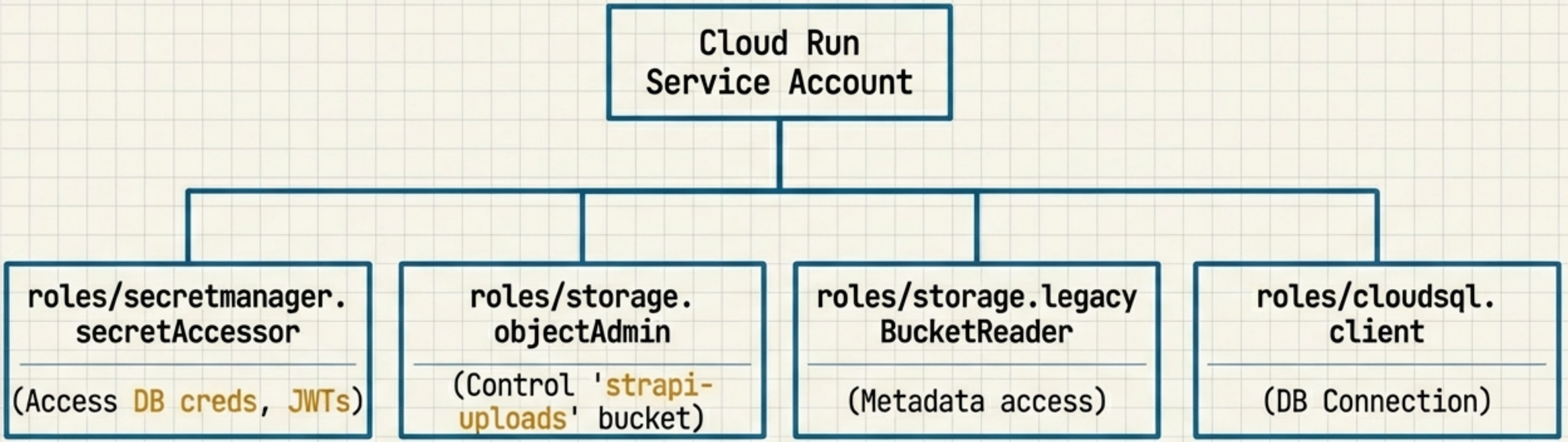
Stage 2: Runtime (The Result)
Inter Tight

- Copies only production artifacts
- User: Non-root 'node'
- Entrypoint: 'tini' (PID 1 signal handling)

NotebookLM

Identity & Access Management (IAM)

Principle of Least Privilege via Service Accounts

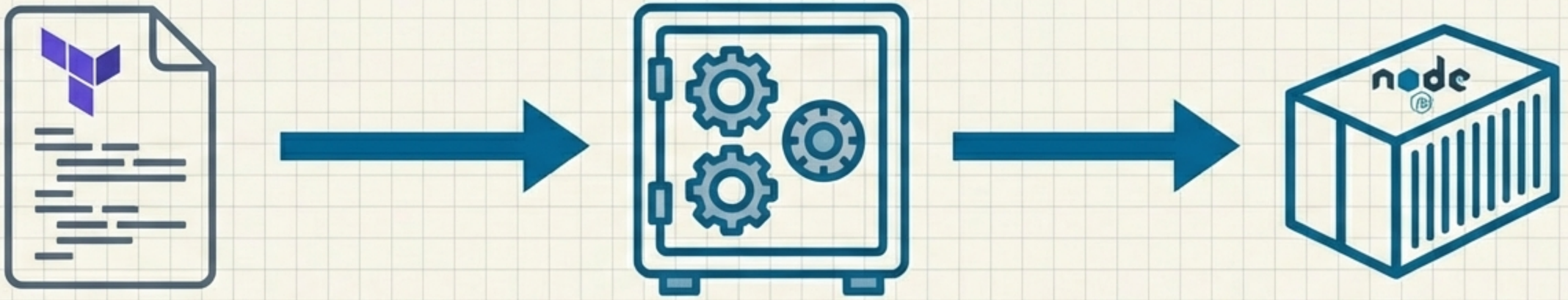


Optional: Cloud Build SA

Requires roles/run.developer

NotebookLM

Secret Management Strategy



Terraform Generation

Secret Manager Storage

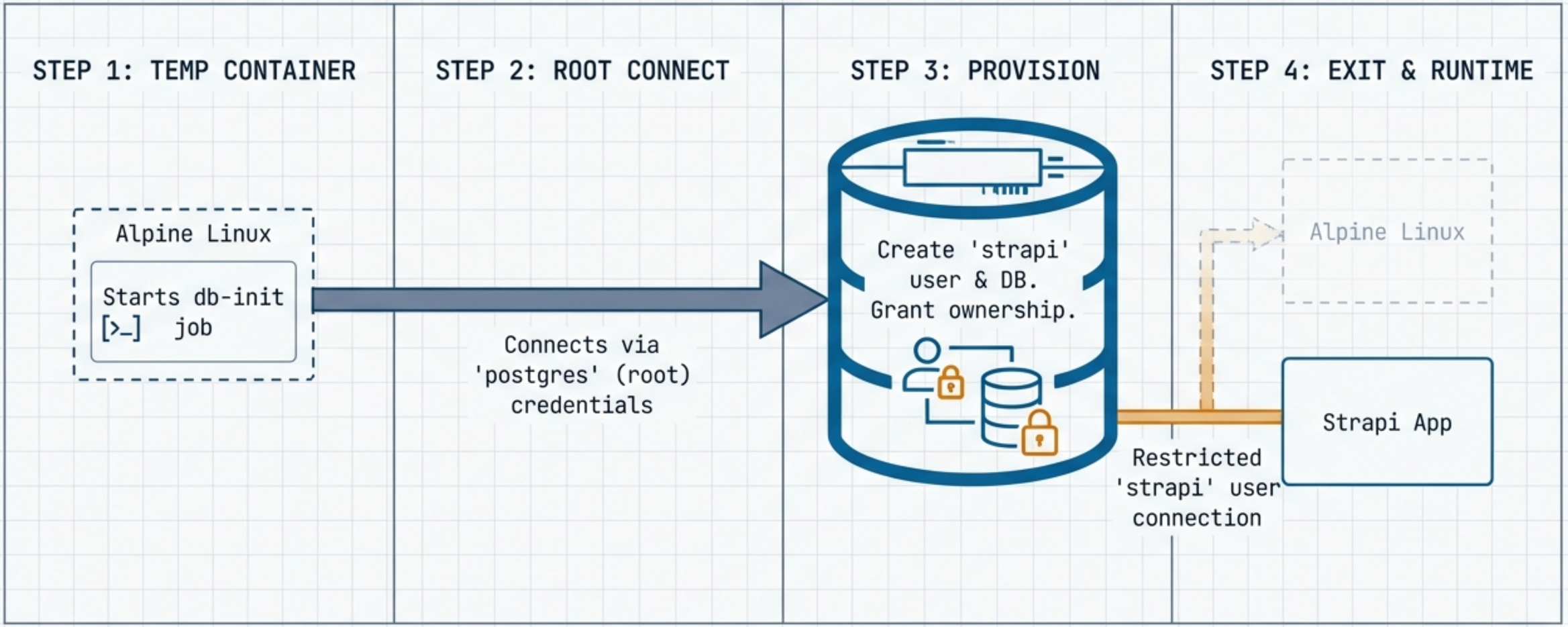
Environment Variables

```
database_password
jwt_secret
admin_jwt_secret
api_token_salt
transfer_token_salt
app_keys
```

NotebookLM

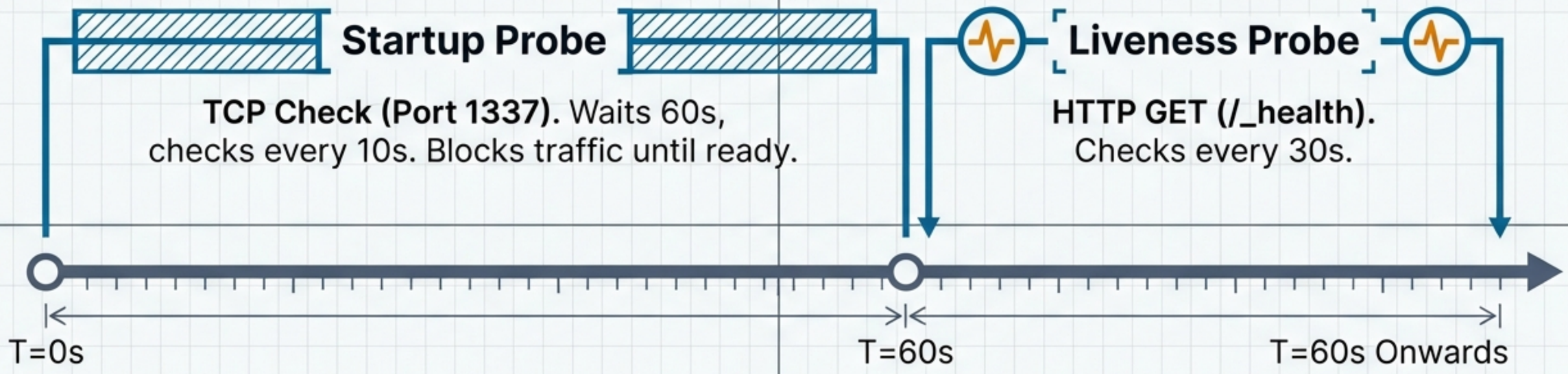
Secure Database Bootstrapping

Solving the Day 0 Provisioning Problem



NotebookLM

Operational Reliability

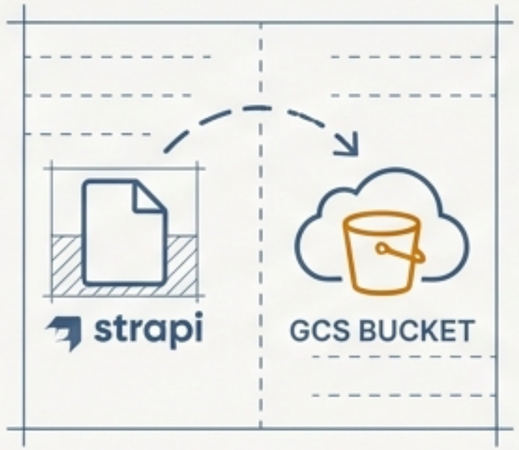


Self-Healing: Unhealthy instances are automatically terminated and replaced.

NotebookLM

Integration Features

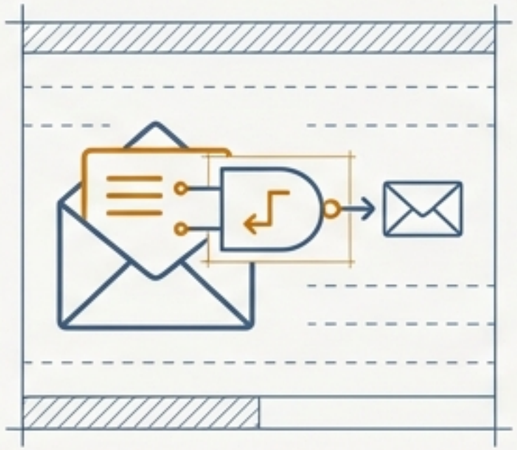
Integrated Media Uploads



`plugins.js` pre-configured for GCS provider.

Automatically maps bucket name and public access settings from env vars.

Email Support (Conditional)



Logic:
`IF 'SMTP_HOST' is defined THEN configure 'nodemailer'.`

Enables zero-code integration with SendGrid/Mailgun.

NotebookLM

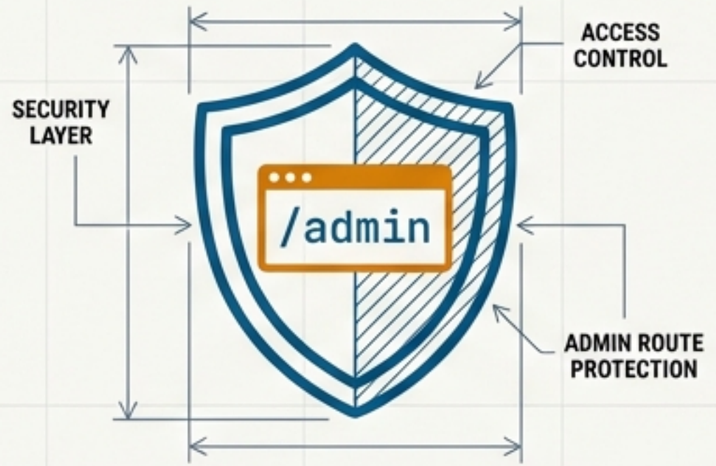
Enhancements: Performance & Scale

Implementation: Cloud CDN

Places Cloud CDN in **front of GCS bucket**. Drastically reduces latency for global media delivery. Offloads static asset traffic from the Strapi application server.

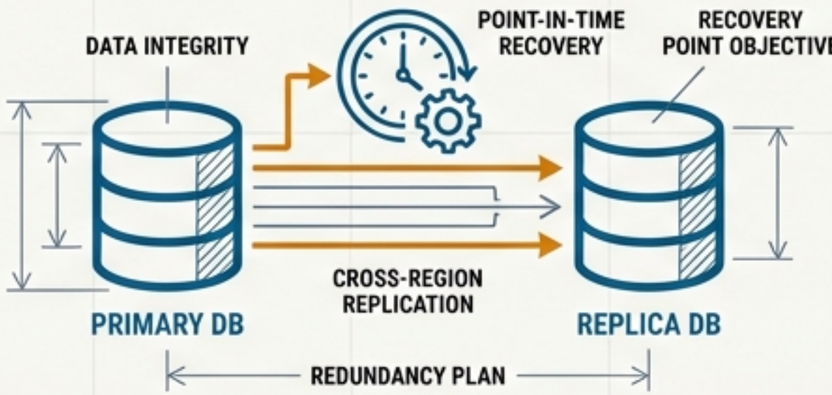
NotebookLM

Enhancements: Security & Maintenance



Security Hardening

- **WAF (Cloud Armor):** Protects API and admin routes.
- **IAP (Identity-Aware Proxy):** Enforces Google Workspace auth for Admin panel access.



Disaster Recovery

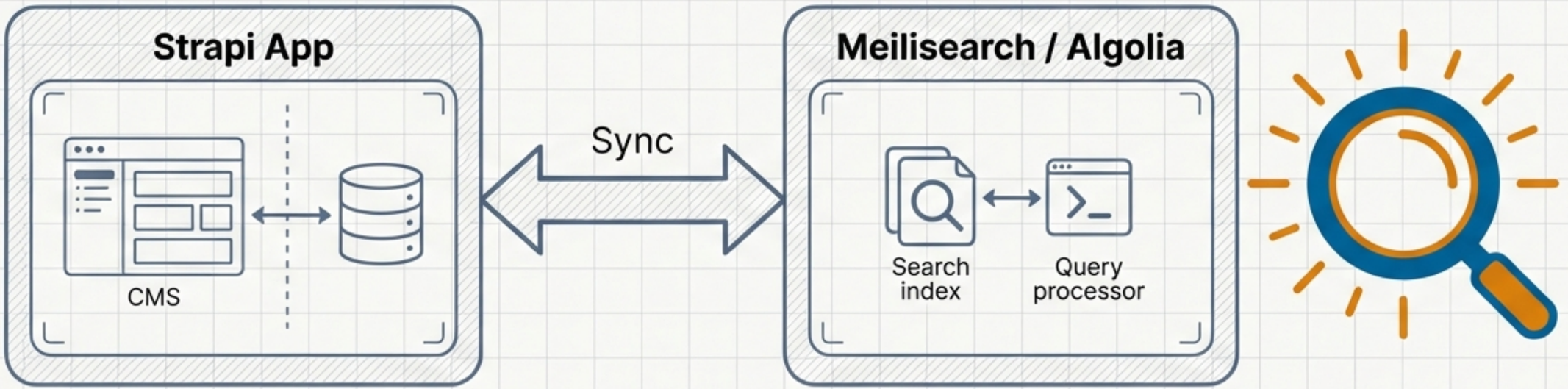
- **Database Backups:** Point-in-Time Recovery (PITR).
- **Replication:** Cross-region redundancy.

PROJECT: STRAPI ENHANCEMENTS
SHEET: SECURITY_MAINTENANCE
REV: A.01

NotebookLM

Enhancements: Functional Capabilities

Search & Discovery



Provisioning a dedicated search service enables high-speed full-text search capabilities within the Strapi content API, bypassing database query limitations.

NotebookLM

Architecture Validation

- ✓ **Secure:** Least privilege IAM, Secret Manager, Non-root runtime.
- ✓ **Scalable:** Cloud Run Autoscaling, Cloud SQL, Separated Storage.
- ✓ **Automated:** db-init bootstrapping, Terraform deployment.
- ✓ **Extensible:** Multi-stage Docker build, Plugin-ready configuration.

A robust foundation for Headless CMS workloads.

NotebookLM