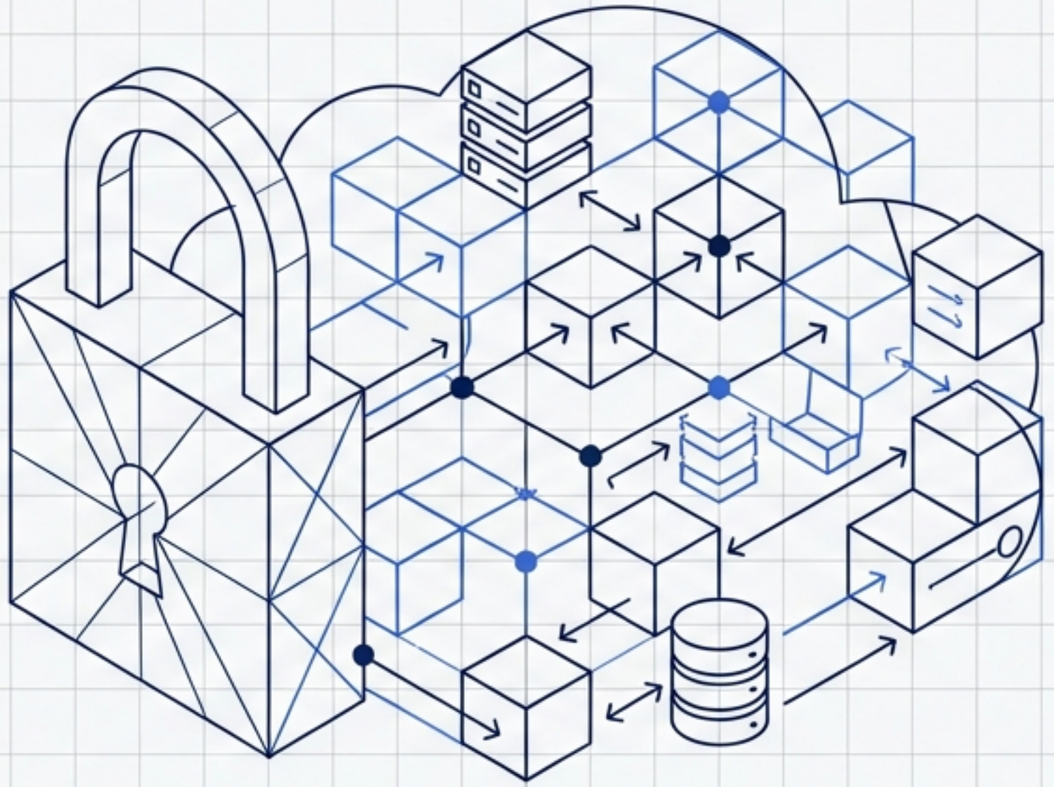
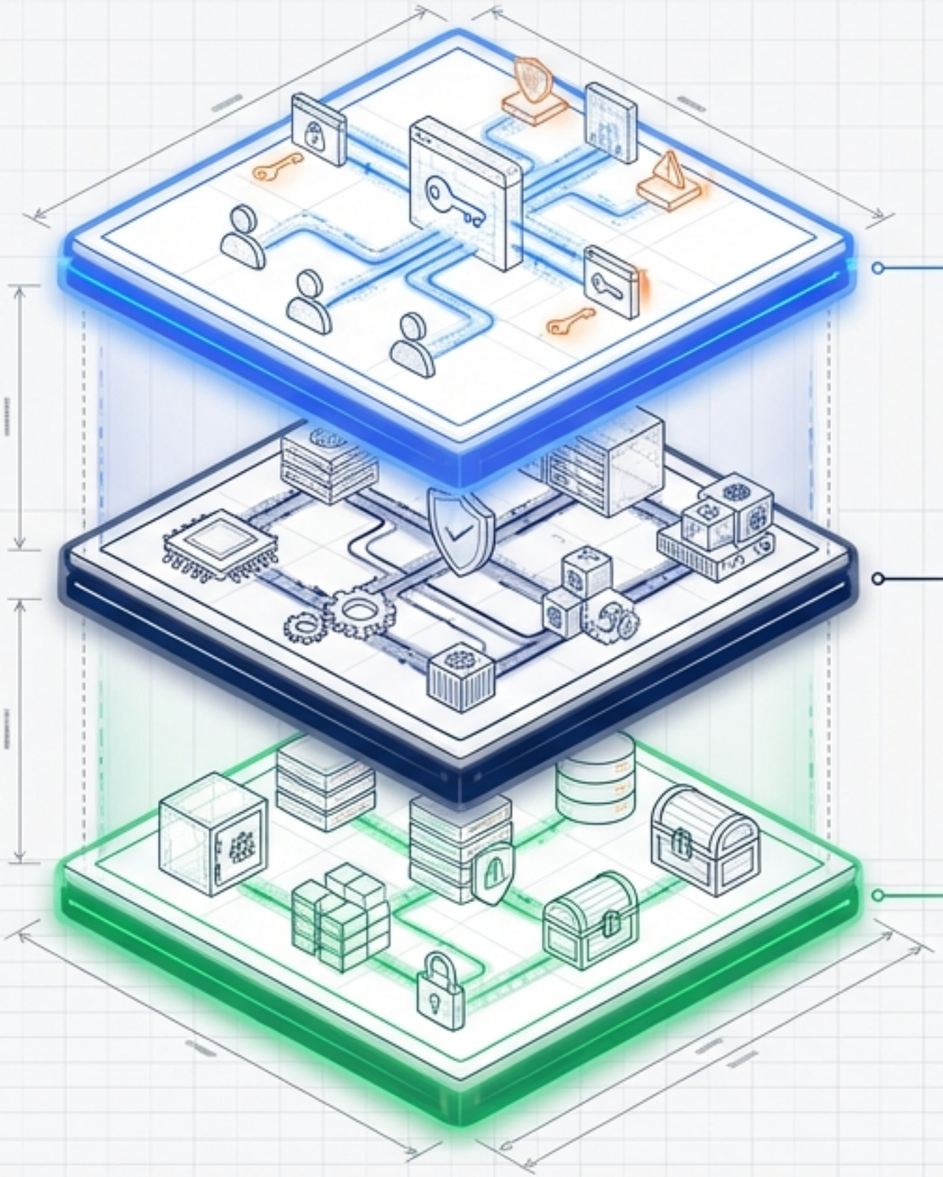


Architecting Zero Trust on Google Cloud

A masterclass on access, security, and identity lifecycle management.

Based on the ACE Certification Section 4 Exploration Guide (20% Exam Weight) & the Tech Equity RAD Platform.

Moving Inward: The Defense-in-Depth Strategy



Human & Perimeter Access

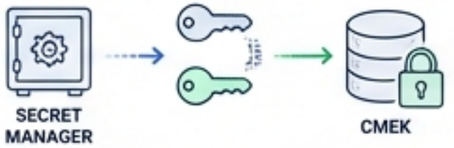
Establishing exactly who is allowed in via IAM and Identity-Aware Proxy.

Machine Identity & Compute

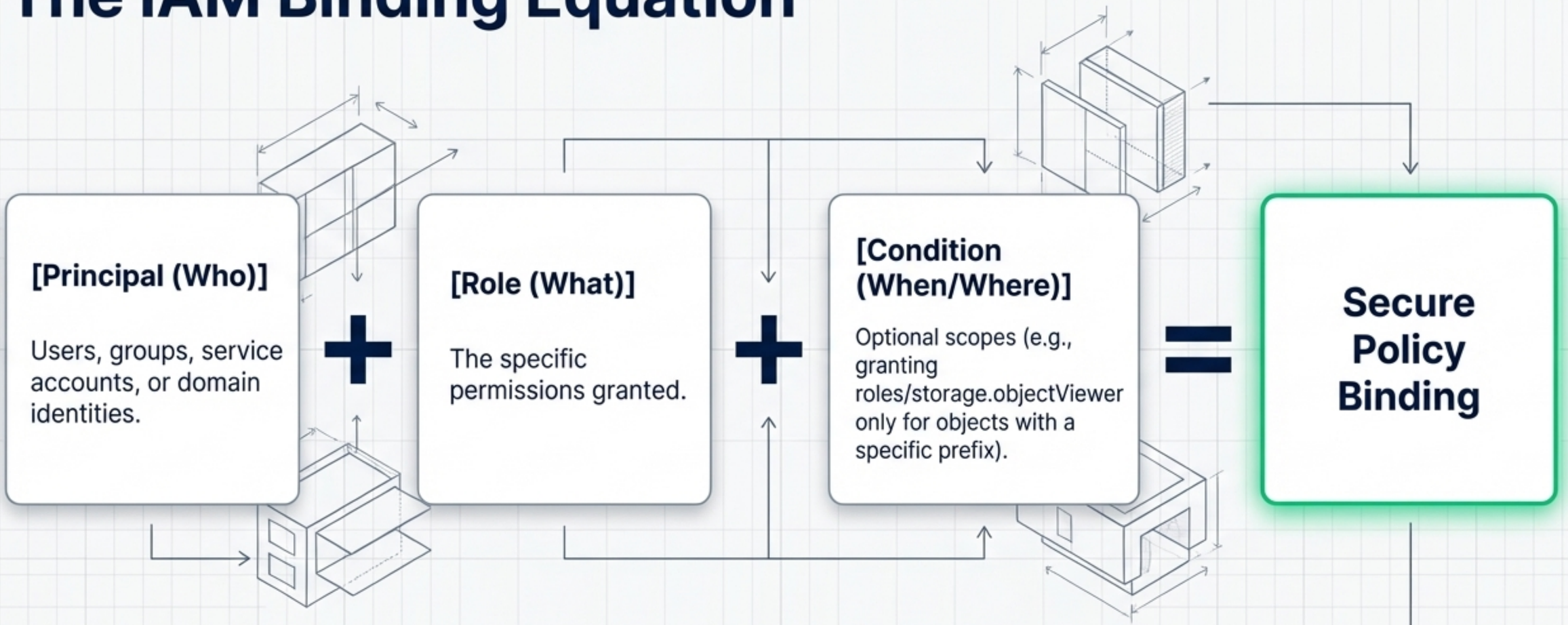
Securing how workloads communicate using Custom Service Accounts and Workload Identity.

Credentials & Data Security

Eliminating static keys with Secret Manager, temporary credentials, and CMEK.

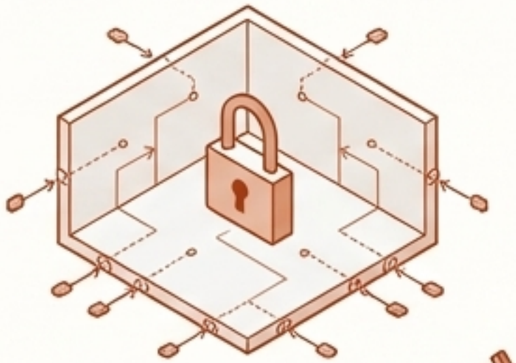


The IAM Binding Equation



Console Action: Navigate to IAM & Admin > IAM. The project's overall IAM policy is simply the sum of all these individual bindings.

The IAM Role Typology Matrix

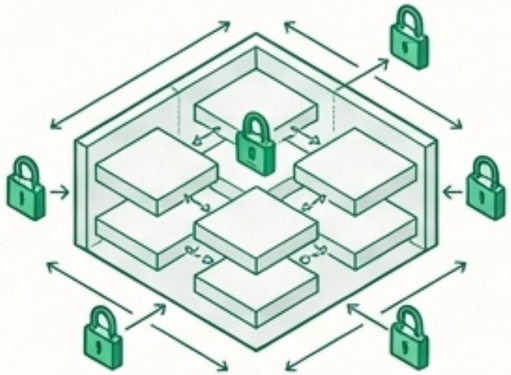


Basic Roles (Primitive)

Broad access. `viewer`, `editor`.

Anti-pattern in production (e.g., `editor` grants write access to most GCP services).

Use only in small dev environments.



Predefined Roles

Curated, service-scoped sets (e.g., `roles/cloudsql.client`, `roles/run.invoker`).

Google-maintained and automatically updated.

The recommended choice for humans and services.

Custom Roles

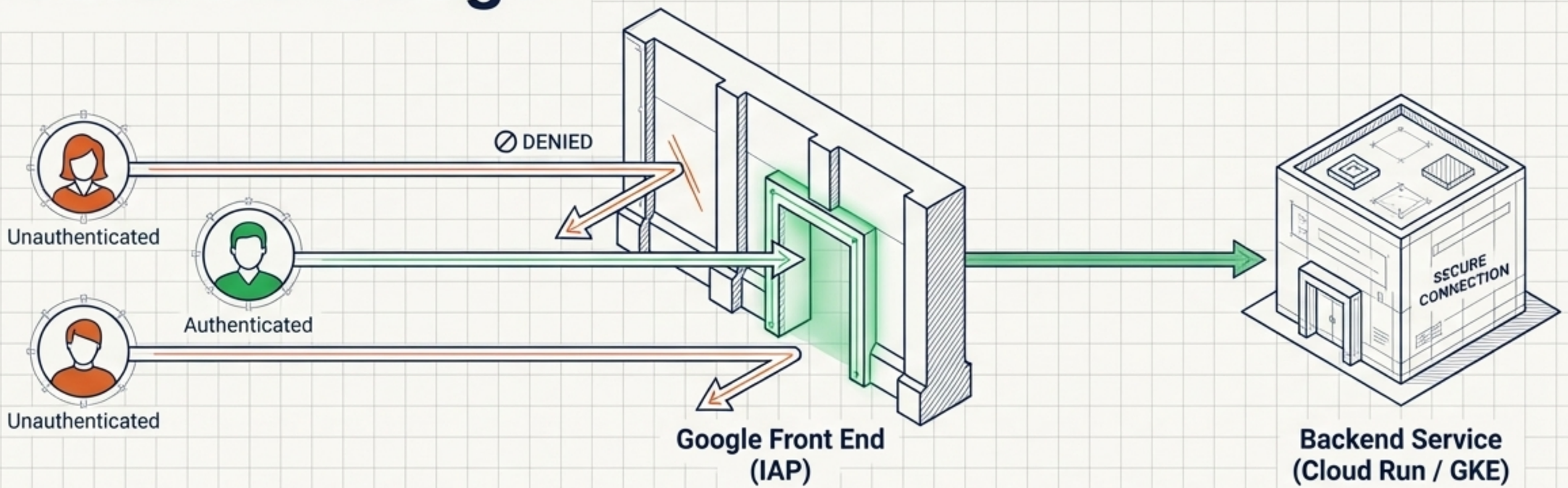
User-defined permission sets.

High operational overhead.

Does not auto-update when Google adds new API methods.

Use only when predefined roles are too permissive.

IAP: Context-Aware Access at the Edge

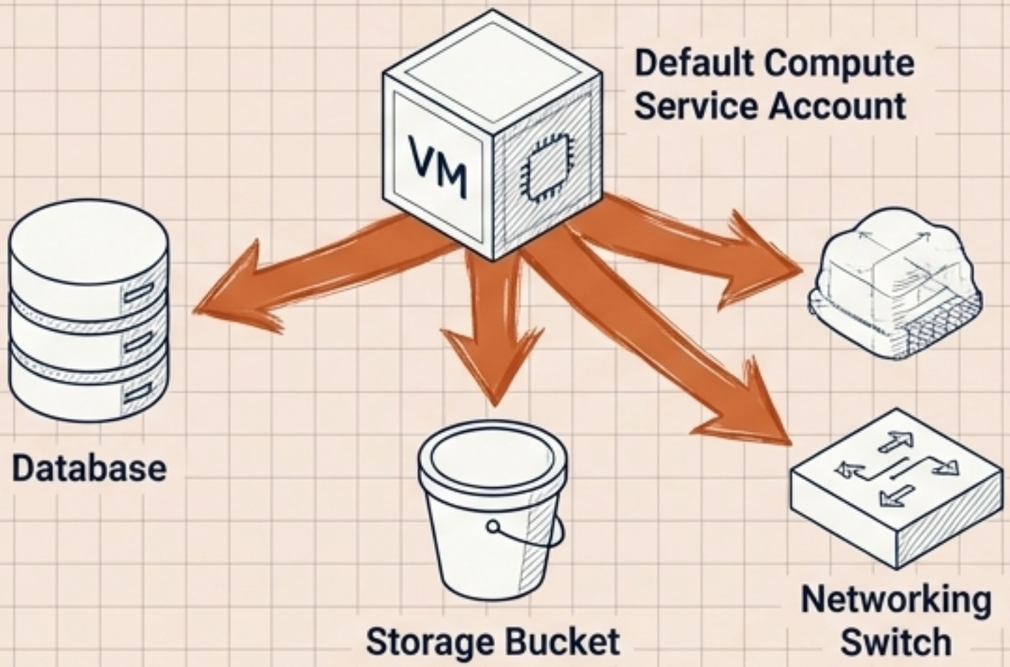


IAP evaluates policies at the network edge before traffic ever reaches your backend application, eliminating the need for traditional VPNs.

Protects against unauthenticated access even if the underlying application code contains vulnerabilities.

```
Configured via iap_authorized_users and
iap_authorized_groups. Can be extended with
BeyondCorp Access Context Manager levels (device
compliance, source IPs).
```

The Compute Identity Imperative



Automatically granted broad Editor permissions.

Never assign `roles/owner` or `roles/editor` to service accounts in production.

Minimum required predefined roles.

The RAD platform strictly provisions dedicated custom service accounts (e.g., `roles/cloudsql.client` or `roles/artifactregistry.reader`) to enforce the principle of least privilege.

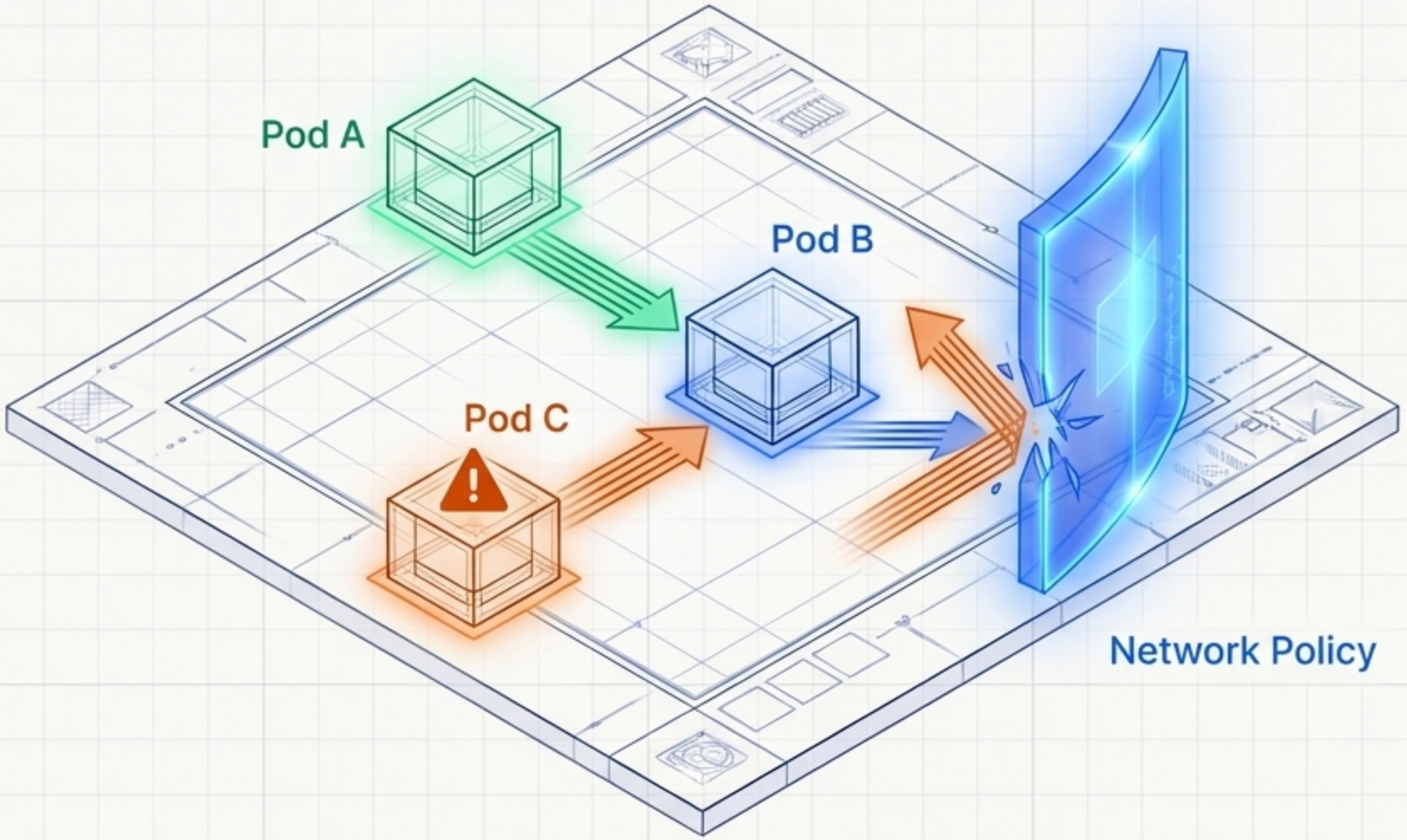
Workload Identity: Credential-Less Authentication

Securely maps a KSA to an underlying GSA.

Allows pods to natively authenticate to Google Cloud APIs.

The Security Win: Eliminates the need to export service account JSON keys. Exported keys do not auto-rotate, are easily exfiltrated, and are difficult to audit.

Defense in Depth: Internal Access Controls

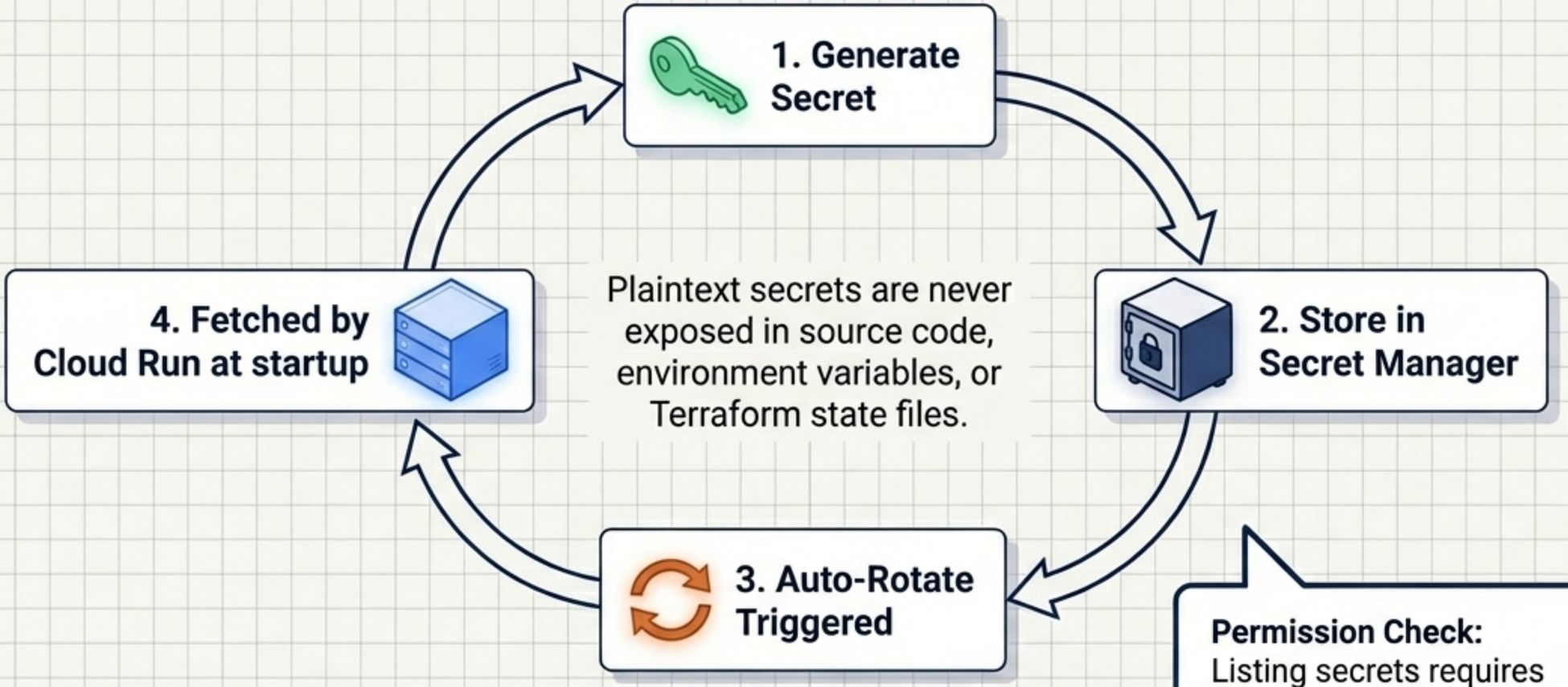


Securing the perimeter is not enough. Strict internal access controls (Network Policies) restrict pod-to-pod communication.

If a single pod is compromised, lateral movement within the cluster is mathematically denied by default.

```
kubectl describe networkpolicies
-n <namespace>
```

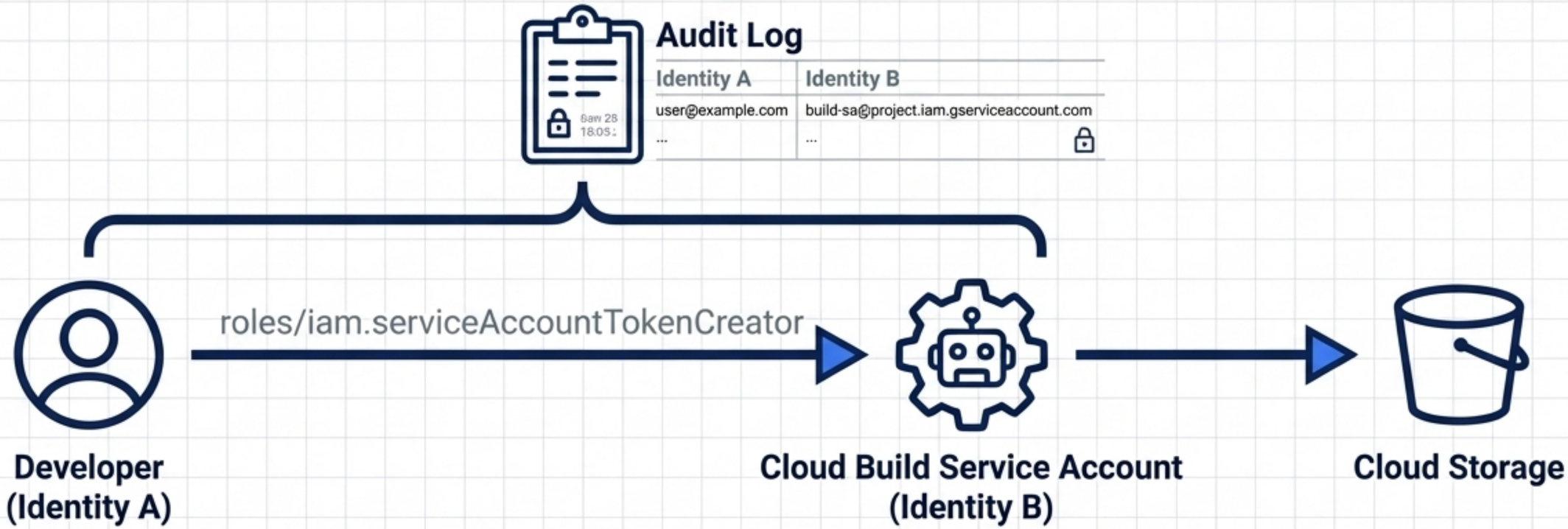
The Secret Auto-Rotation Lifecycle



When a database password rotates (e.g., enable_auto_password_rotation), the secret version updates automatically. Cloud Run fetches the new version upon startup—no redeployment required.

Permission Check:
Listing secrets requires roles/secretmanager.viewer.
Actually reading the plaintext values requires a separate, elevated role: roles/secretmanager.secretAccessor.

Temporary Power: Service Account Impersonation



Allows a user or service account to make API calls as a target service account without downloading a key.

The Audit Advantage: Every API call is logged with both the original caller and the impersonated identity, ensuring perfect traceability.

```
gcloud storage ls --impersonate-service-
account=deployer@project.iam...
```

The Short-Lived Credential Matrix

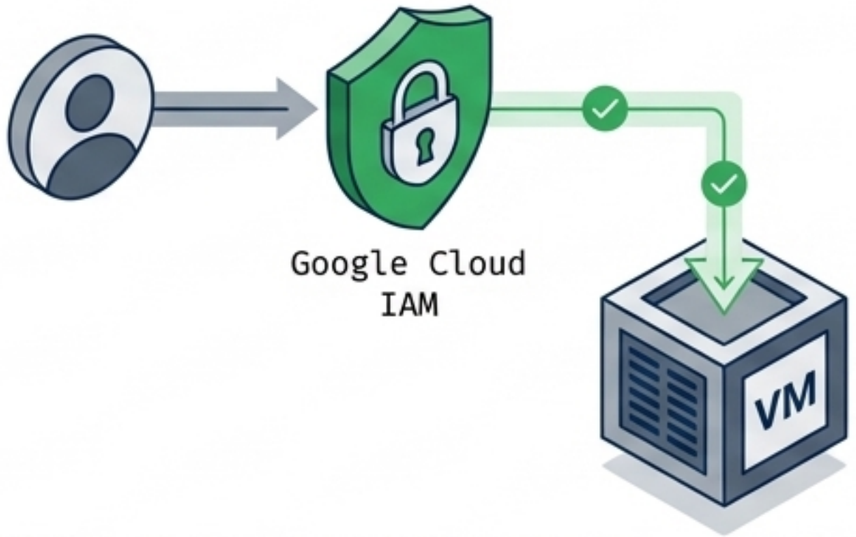
Header Column	Details Column
Access Tokens	OAuth 2.0 bearer tokens. Valid 1–12 hours. Generated via Generated via `gcloud auth print-access-token` or `generateAccessToken` API.
ID Tokens	JWT tokens. Used for authenticating to Cloud Run, Cloud Functions, Generated via `gcloud auth print-identity-token` or `generateIdToken` API.
Service Account Keys	Long-lived JSON private keys. Never expire. Google recommendation: Avoid wherever possible. If created, active monitoring for key age is mandatory.

Compute Security: OS Login vs. SSH Keys

Legacy SSH

Project-wide public SSH keys propagated manually to metadata. Hard to audit, lifecycle management is manual.

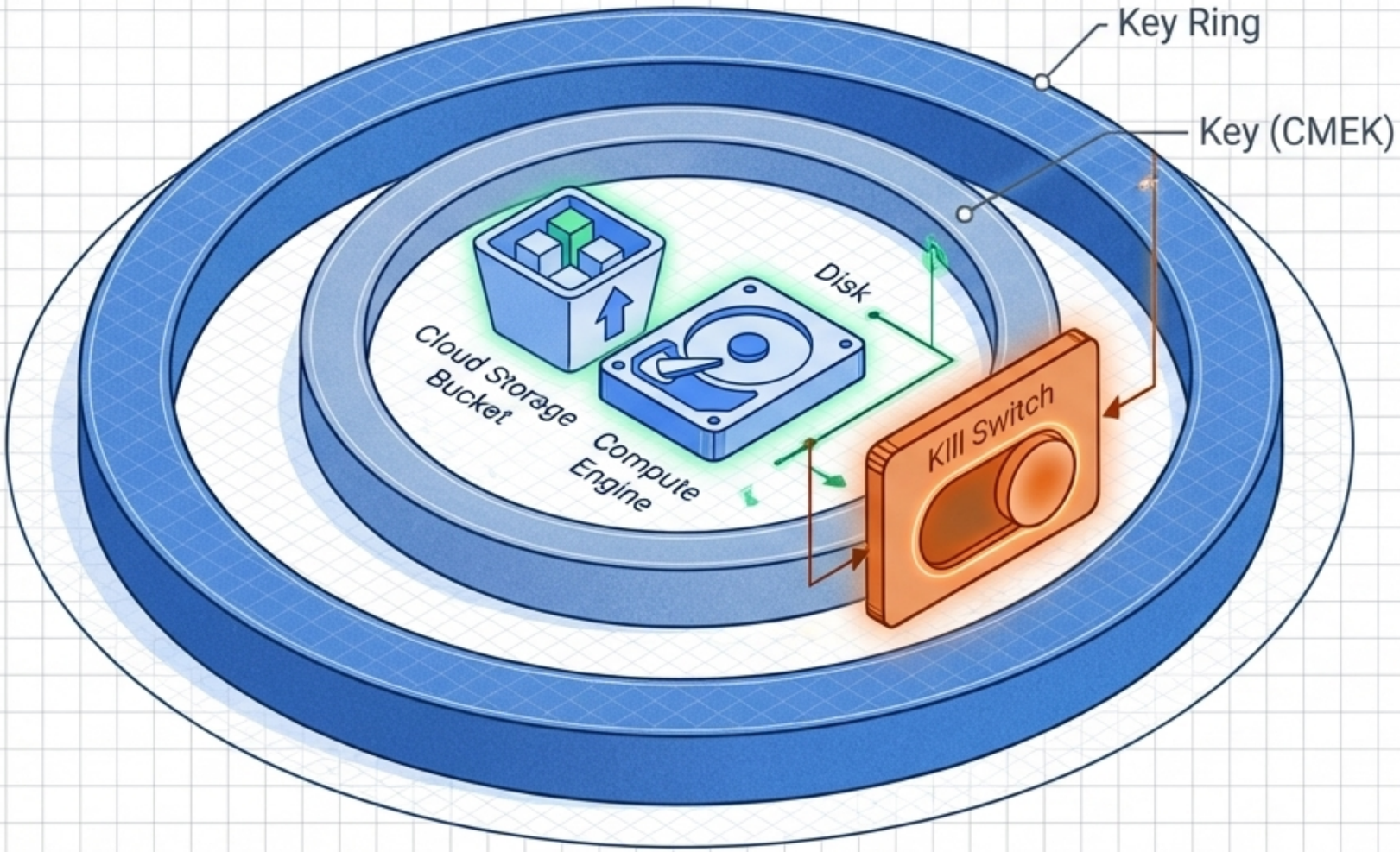
OS Login



The Google-recommended method. Ties SSH access directly to IAM (`roles/compute.osLogin` or `osAdminLogin`).

Access is revoked instantly when IAM bindings are removed, and every login produces an audit log entry in Cloud Logging.

Data Sovereignty via Cloud KMS

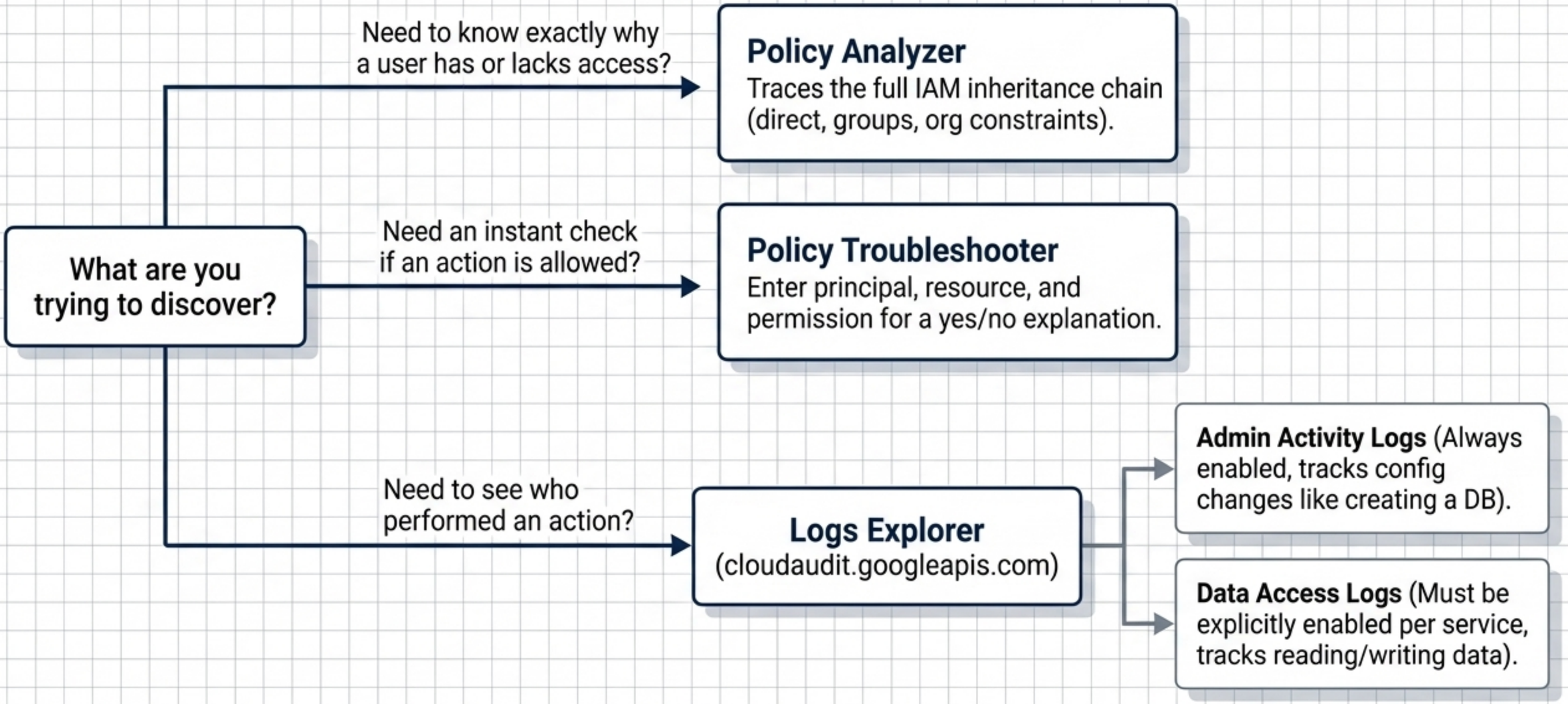


Default: Google manages the encryption keys automatically.

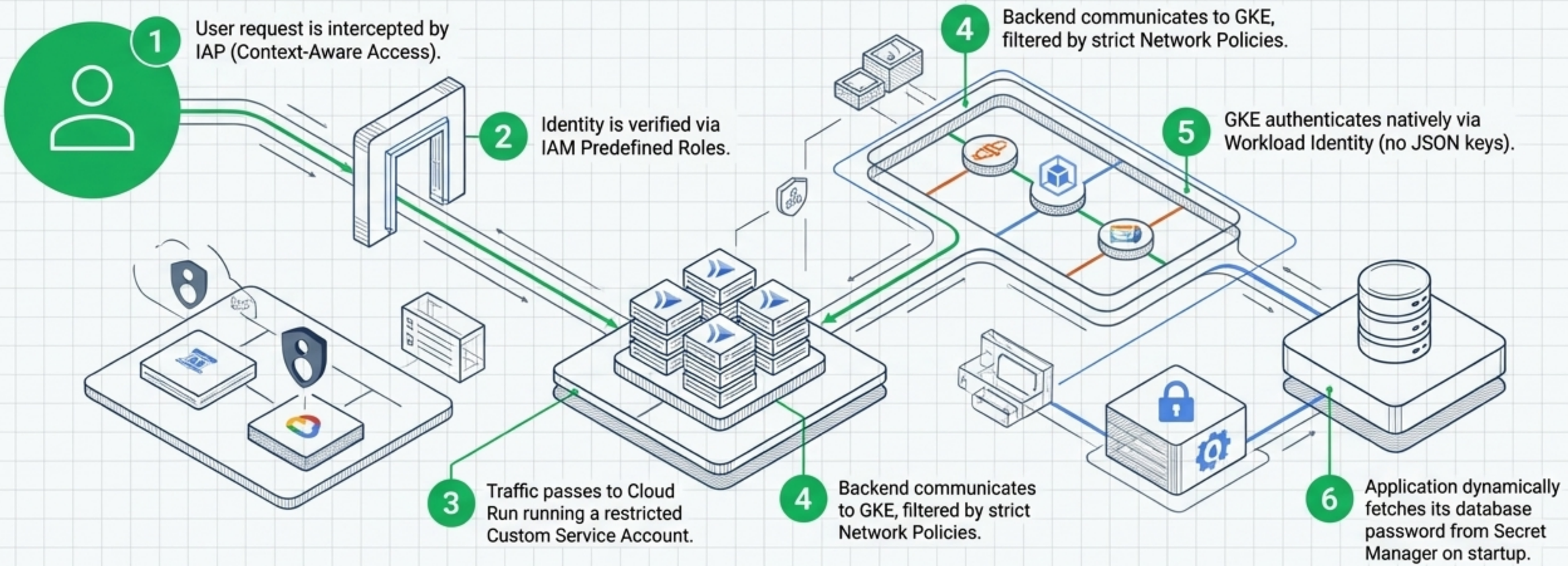
CMEK (Customer-Managed Encryption Keys): Gives you complete control over the key lifecycle.

By rotating, disabling, or destroying a key in Cloud KMS, you effectively revoke Google Cloud's ability to decrypt the protected data, even for its own underlying maintenance operations.

Diagnostic Decision Tree



Synthesis: The Least Privilege Ecosystem



Google Cloud security is not a checklist of isolated features. It is a deeply integrated web of defense-in-depth mechanisms working in unison to eliminate static risk.