Secure ML Chatbot Platform — Client Briefing

Date: 2025-10-04

Prepared by: Mohamed Gouda — Senior DevOps Engineer

1. Executive Summary

We engineered an end-to-end Google Cloud Platform landing zone and machine-learning chatbot workload, demonstrating our capability to deliver secure and automated AI platforms. The blueprint spans infrastructure, application, data, and operations with security baked in at every layer.

2. Business Outcomes Delivered

• Accelerated GenAI value: Vertex AI-powered conversations with safe rollout controls.

• Security by design: Zero Trust access patterns, layered defenses, automated scanning.

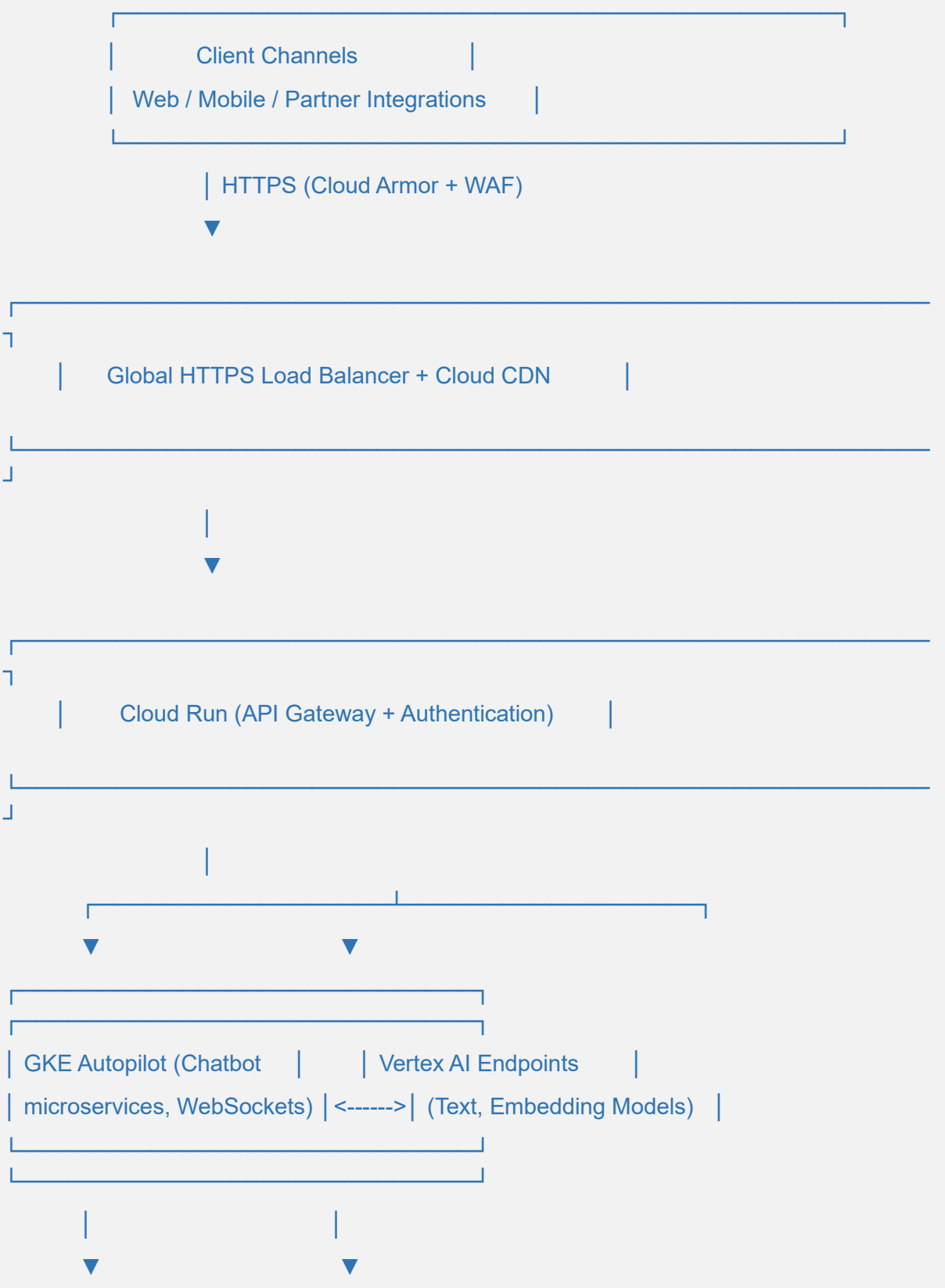
• Operational excellence: Terraform modules, GitHub Actions pipelines, policy-as-code guardrails.

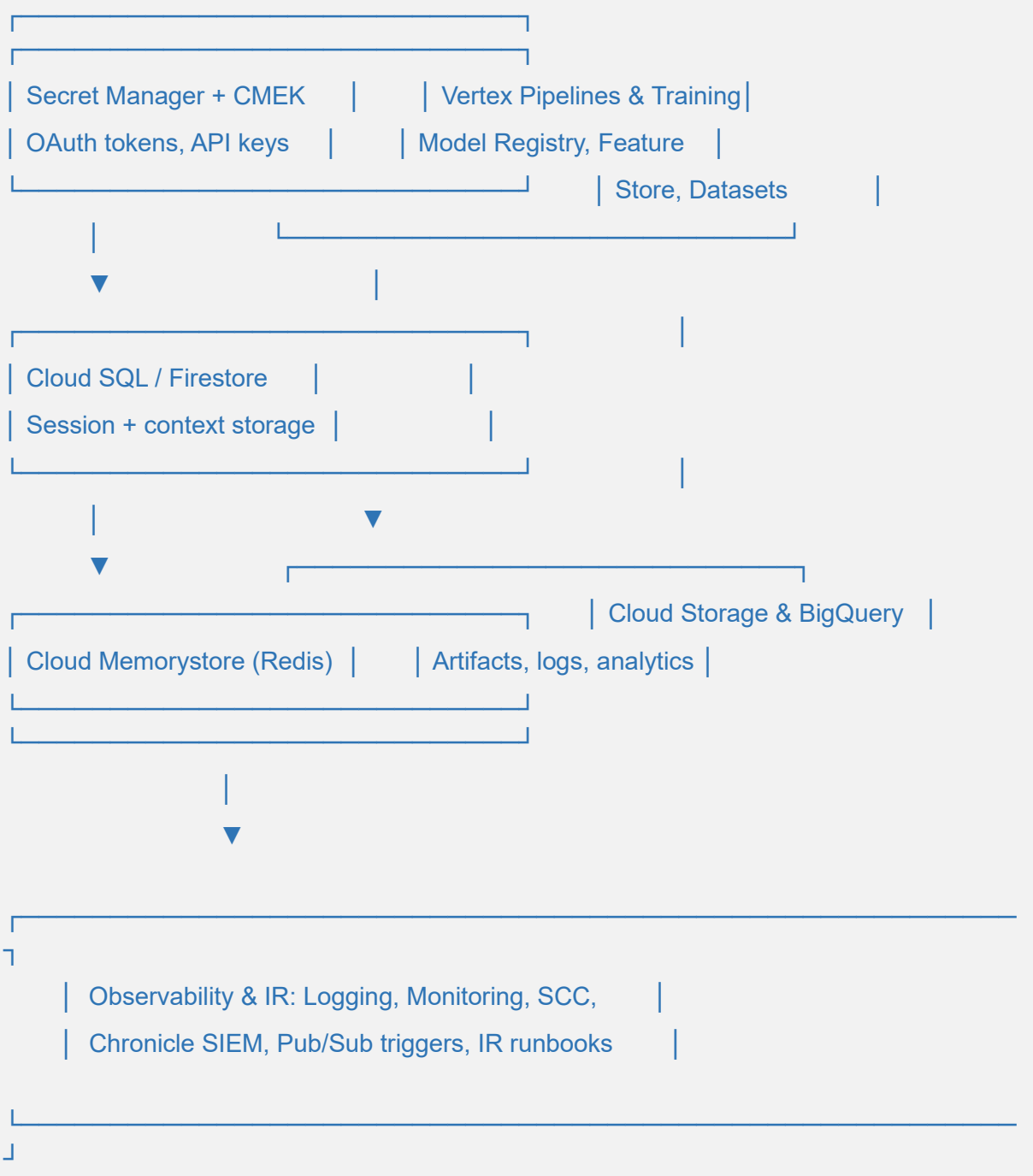
• Developer productivity: Devcontainer mirroring CI tooling and scripted security checks.

• Observability & response: Security Command Center, Chronicle SIEM, automated runbooks.

3. Architecture Overview

3.1 High-Level Architecture





The architecture emphasizes a service-perimeter model: privileged APIs stay private behind VPC Service Controls while public entry points are hardened with Cloud Armor and OAuth-based identity-aware proxies.

3.2 Environment Topology

• Dev and Prod projects inherit organization policy constraints (CMEK required, restricted services, audit log sinks).

• Terraform maintains isolated remote state per environment in CMEK-protected GCS buckets; plan/apply gated via approvals.

4. Infrastructure Automation

Terraform modules compose networking, IAM, security, logging, GKE, Vertex, Cloud Build, and Cloud Run workloads. Policy-as-code (OPA + Config Validator) enforces guardrails before apply.

Sample Terraform Module

```
module "gke" {
  source  = "../../modules/gke"
  project = var.project_id
  name    = "chatbot-gke"

  network    = module.network.vpc_name
  subnetwork = module.network.app_subnet

  enable_private_endpoint     = true
  enable_binary_authorization = true

  workload_identity = {
    enabled = true
    issuer  = "https://github.com/${var.github_org}/${var.repo}"
  }

  master_authorized_ranges = [
    {
      cidr_block   = var.corp_cidr
      display_name = "corp"
    }
  ]
}
```

5. Application & MLOps Layer

The FastAPI chatbot service authenticates with Google Identity tokens, routes prompts to Vertex AI, and logs context for analytics. Vertex Pipelines orchestrate data prep, training, and model promotion to online endpoints.

Sample FastAPI Endpoint

```
from fastapi import FastAPI, Depends
from google.cloud import aiplatform
from app.auth import verify_token

app = FastAPI()
vertex_client = aiplatform.gapic.PredictionServiceClient()
endpoint = vertex_client.endpoint_path(
    project="${PROJECT_ID}", location="us-central1", endpoint="chat-endpoint"
)

@app.post("/chat")
async def chat(request: ChatRequest, user=Depends(verify_token)):
    instances = [{"prompt": request.prompt, "context": request.context}]
    response = vertex_client.predict(endpoint=endpoint, instances=instances)
    return {"reply": response.predictions[0]["content"]}
```

6. Security & Compliance Posture

• Network & perimeter: TLS 1.2+, mutual TLS east-west, Cloud Armor WAF/bot protection, BeyondCorp Enterprise for workforce.

• IAM & secrets: Workload Identity Federation, IAM Conditions, CMEK-backed Secret Manager, Binary Authorization for containers.

• Continuous assurance: Trivy, Checkov, Conftest, Bandit, kube-bench in CI; Security Command Center findings flowing into Chronicle for threat hunting.

• Incident readiness: Immutable log sinks, automated workload isolation, forensics snapshots, and documented runbooks.

7. CI/CD Workflow (GitHub Actions)

Stage 1 — prepare: checkout, tooling install (terraform, opa, trivy), dependency cache.

Stage 2 — lint/test: Python lint (ruff), security (bandit), unit tests, coverage reporting.

Stage 3 — IaC validation: terraform fmt/validate, plan (non-destructive), Checkov, Conftest.

Stage 4 — container build: Cloud Build via OIDC, push to Artifact Registry, Trivy scan.

Stage 5 — deployment: manual approval gates, terraform apply (dev/prod), Vertex deployment, post-deploy smoke tests.

8. Engagement Roadmap

1. Discovery & tailoring workshop: align regulatory, data, integration requirements; finalize success metrics.

2. Pilot delivery: provision dev environment, deploy chatbot MVP, integrate Vertex AI, configure monitoring dashboards.

3. Harden & extend: onboard identity providers, enforce custom policies, integrate SIEM/ITSM tooling.

4. Production rollout: execute prod Terraform plan, enable operational runbooks, hand over documentation and training.

We can deliver as a dedicated pod or embed with your teams for accelerated knowledge transfer.