Veria Tech Spec v2 — Blitzy Edition (Private■Only)

Veria — Technical Specification v2 (Blitzy Edition, Private■Only)
Date: 2025-09-15
Owner: Daniel Connolly
Repo: PROACTIVA-US/Veria
Environment (dev): GCP veria-dev (project number 190356591245), region us-central1
---
1) Executive Summary
Veria is an AI■native compliance & distribution middleware for tokenized real■world assets.
This v2 spec supersedes tech-specV1.pdf and reflects the working **OIDC/WIF** CI/CD and a successful Cloud Run deployment of **ai-broker**.
Key facts
- OIDC/WIF auth throughout; **no JSON keys**.
- Deployments to Cloud Run are **by image digest** (not :latest).
- Organization policy requires **private■only** services (authenticated access).
---
2) Goals / Non■Goals
Goals
- Provide Blitzy a current, authoritative reference for dev operations.
- Keep OIDC/WIF & IAM unchanged; private■only access enforced.
- Enumerate required outputs from Blitzy (deploy report + rollback).
Non■Goals
- No production scaling/SLA in this doc.
- No changes to OIDC/WIF or branch protections.
---
3) Architecture Overview (dev)
- Runtime: Google Cloud Run (containers)
- Primary Service: **ai-broker** (Node 20, TypeScript, Express)
- Build Tooling: pnpm workspaces; Docker linux/amd64
- Registry: Artifact Registry (use existing path in workflow)
- CI/CD: GitHub Actions → `google-github-actions/auth@v2` (OIDC/WIF)
- Secrets: GitHub (WIF refs) + **Google Secret Manager** for runtime
- Observability: Cloud Logging (metrics: default Cloud Run for now)
---
4) Repos & Branching
- Repo: PROACTIVA-US/Veria
- Default branch: `main`
- Release tags: `v*.*.*` (allowed for deploy)
- Open PRs: none at time of publishing
---
5) Services
5.1 ai■broker
- Language: TypeScript (ESM); Framework: Express
- Endpoints: `GET /` (health), `POST /suggest` ({ prompt, maxTokens? })
- Status: compiles cleanly; tests passing; deployed to Cloud Run
- **Exposure:** **Private■only** by org policy
---
6) Environments & Access
- Project: **veria-dev**; Region: **us-central1**
- Cloud Run service name: **ai-broker**
- Runtime SA: default unless specified in workflow
- **Access:** **Private■only** — unauthenticated access is prohibited by org policy
---
7) CI/CD (GitHub OIDC/WIF)
- WIF Pool: `github-pool` (ACTIVE)
- WIF Provider: `github-provider` (ACTIVE)

- Provider Condition:
  `attribute.repository == "PROACTIVA-US/Veria"` AND
  `attribute.ref.startsWith("refs/heads/main") OR attribute.ref.startsWith("refs/tags/")`
- CI Service Account: `veria-automation@veria-dev.iam.gserviceaccount.com`
- Required GH Secrets: `GCP_PROJECT_ID`, `GCP_SA_EMAIL`, `WORKLOAD_IDENTITY_PROVIDER`
- Workflow: `.github/workflows/cd.yml` (push main + tags `v*.*.*`), build **linux/amd64**,
deploy by digest, **--no-allow-unauthenticated**
---
8) Build & Deploy
- Target: linux/amd64
- Container: service Dockerfiles
- Registry: Artifact Registry (path defined in workflow)
- Deploy: Cloud Run **by digest**; **--no-allow-unauthenticated**
- Min instances: 0 (scale-to-zero) unless specified otherwise
---
9) Config & Secrets
- Use env vars & Secret Manager; no plaintext secrets in repo/workflows.
---
10) Observability & Ops
- Logs: Cloud Logging (filter by `resource.labels.service_name="ai-broker"`)
- Rollback: shift traffic to previous revision (see Runbook)
---
11) Security
- No service account keys; **OIDC/WIF only**.
- Private■only access; grant `roles/run.invoker` only to required principals.
- Least privilege for CI SA (`run.admin`, `iam.serviceAccountUser`, `artifactregistry.writer`).
---
12) Acceptance Criteria (for Blitzy run)
- Build & push (linux/amd64) completes.
- Cloud Run deploy completes **by digest**; new revision created.
- Service healthy; **ID■token** smoke tests pass (private■only).
- Output report: URL, digest, revision, traffic split, rollback command.
---
13) Runbook (One■Screen)
1. Trigger CD: merge to `main` or tag `vX.Y.Z`
2. Describe service:
   ```bash
   gcloud config set project veria-dev
   gcloud run services describe ai-broker --region=us-central1
--format='yaml(status.url,status.traffic)'
   ```
3. **Private■only** Smoke test (ID token):
   ```bash
   SERVICE=ai-broker
   REGION=us-central1
   URL=$(gcloud run services describe "$SERVICE" --region="$REGION"
--format='value(status.url)')
   IDT=$(gcloud auth print-identity-token --audiences="$URL")
   curl -sSf -H "Authorization: Bearer $IDT" "$URL"/ || true
   curl -sSf -H "Authorization: Bearer $IDT" -H 'content-type: application/json' \
     -d '{"prompt":"hello"}' "$URL"/suggest || true
   ```
4. Rollback (replace REV):
   ```bash
   gcloud run services update-traffic ai-broker \
     --region=us-central1 \
     --to-revisions=REV=100
   ```

---
14) Blitzy Prompt (WHY / WHAT / HOW)
WHY
Autonomously build & operate Veria dev using our existing OIDC/WIF pipeline, deploying `ai-broker` to Cloud Run **by digest** with **private■only** access and producing a clear ops report + rollback plan.
WHAT
- Build, push, deploy `ai-broker` (private■only).
- Provide post■deploy report (URL, digest, revision, traffic).
- Confirm logs visible in Cloud Logging.
- Do **not** alter OIDC/WIF, IAM, or branch protections.
HOW
- Use `.github/workflows/cd.yml` (OIDC `auth@v2`, linux/amd64).
- Use existing Artifact Registry path; deploy by **digest** only.
- CI auth via `veria-automation@veria-dev.iam.gserviceaccount.com`.
- Request any runtime secrets/envs and mount via Secret Manager.

Appendix — Public Access
Not applicable. Organization policy prohibits unauthenticated access.