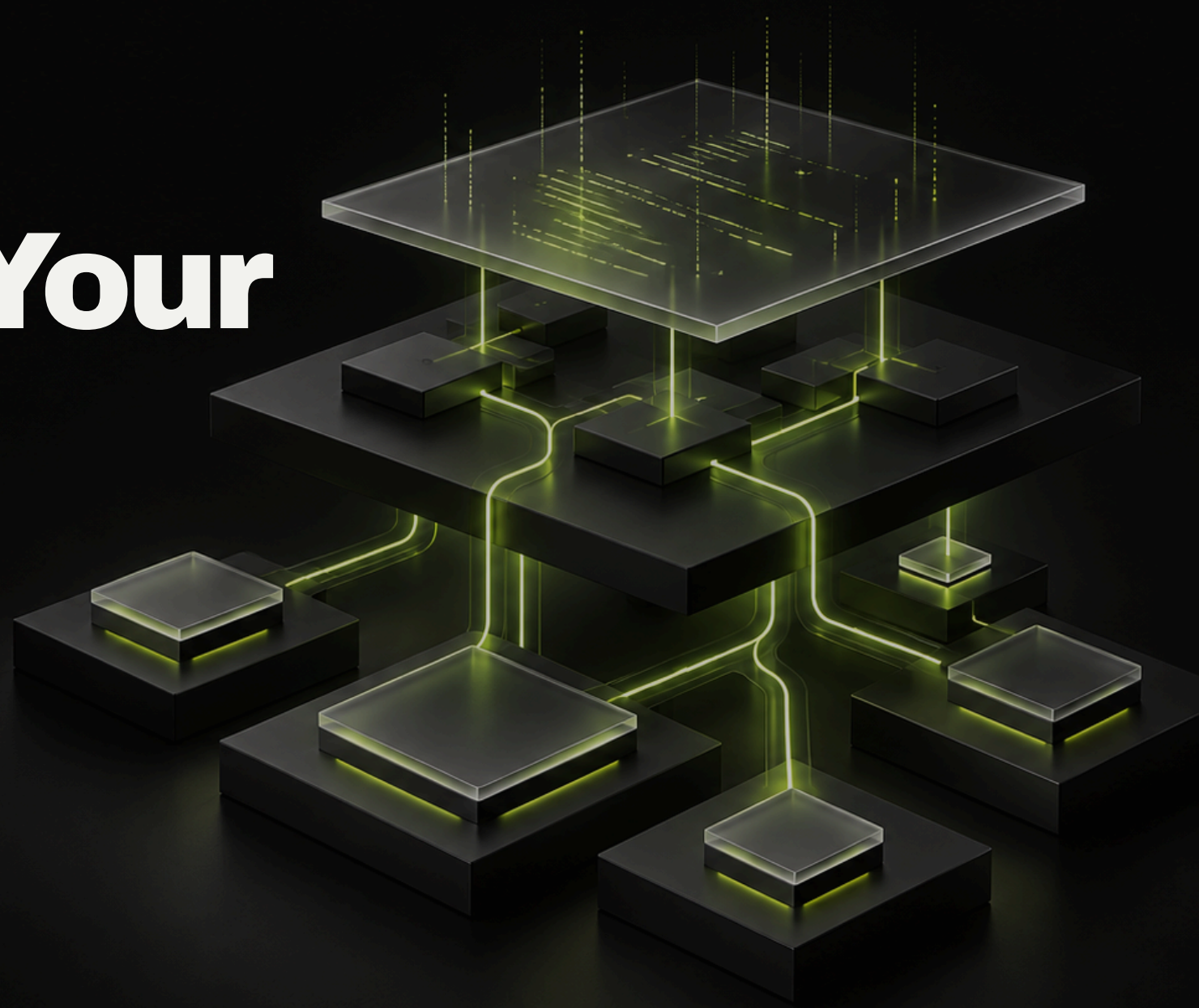


GOLF / GOLF FOR DEVELOPERS

Your code. Your next move.

A practical source-to-preview workflow.



From source to software.

Golf connects Git, collaboration, executable evidence, optional proposals and protected previews.

01

Source

Repositories, history, issues, PRs and human review.

02

Evidence

Approved exact-source tests, reports and proposals.

03

Delivery

Protected static and bounded Worker app previews.

Code. Check. Review. Preview.

Follow one exact change through the workspace.

01

01 / Source

Select the repository, branch and exact commit.

02

02 / Change

Open an issue or same-repository PR and request review.

03

03 / Check

Run an authorized profile and inspect the terminal report.

04

04 / Preview

Publish an allowed protected preview and inspect the result.

Docker: the portable Golf stack.

Run a complete developer core on your machine or owned Docker infrastructure.

01

What runs

Golf React/Go, Forgejo and independent PostgreSQL databases/roles. Private configuration and persistent volumes survive ordinary restarts.

02

How to start

Initialize private environment files, start Compose, check /readyz, then enroll a mapped Golf account. Optional Docker workers, agents and object storage require their own setup.

03

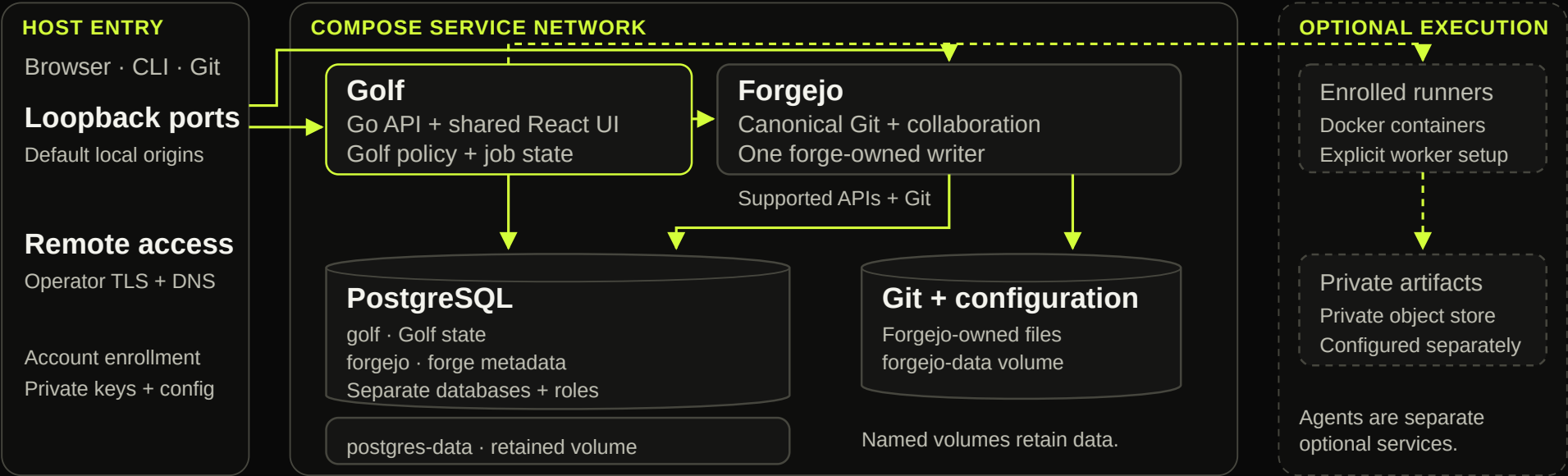
What you own

Image/version choices, HTTPS beyond loopback, secrets, data volumes and coordinated recovery. No Cloudflare or AI account is needed for core work.

Docker: what runs, what persists.

The portable core uses Forgejo and PostgreSQL. Execution workers are allocated separately.

g golf. DOCKER / PORTABLE SERVICE OWNERSHIP



Golf uses forge APIs; each service owns its data. Persistence and recovery need operator verification.

[Editable architecture diagram](#) · [Open full size](#)

Kubernetes: deploy Golf deliberately.

Golf's Helm chart packages the portable core and makes its operating boundaries explicit.

01

What the chart supplies

Golf replicas, one Forgejo writer, optional PostgreSQL, resources/probes, retained claims, non-root isolation and default-deny policies.

02

How it is hardened

Upgrade identity/endpoint fences, API hostname spread and multi-replica disruption budget, graceful drain, private database CA mounts and explicit DNS/egress peers.

03

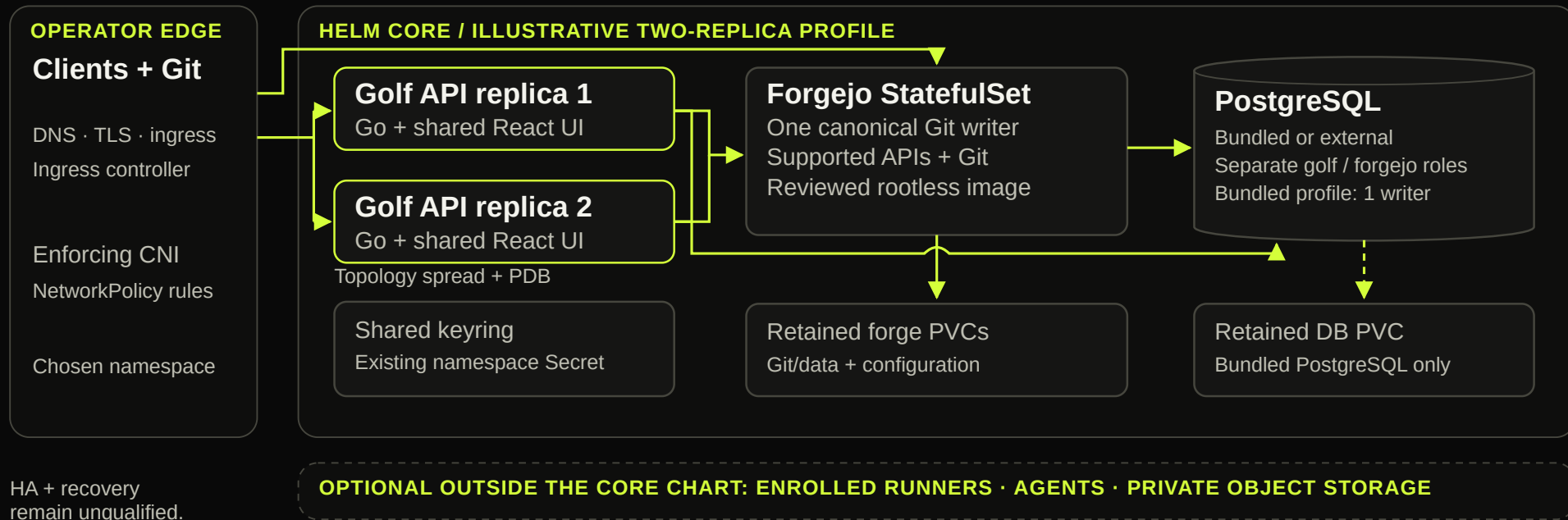
What you supply

Built image, existing credential/certificate Secrets, enforcing CNI, storage, ingress/TLS/DNS and a tested recovery procedure. Runners and agent services remain separate.

Kubernetes: placement and ownership.

API replicas share one forge writer and retained storage. More replicas alone do not establish HA.

g golf. KUBERNETES / REPLICAS AND RETAINED DATA



Editable architecture diagram · [Open full size ↗](#)

Cloudflare: a native hosted control plane.

Artifacts, D1 and Workers replace origin-backed forge infrastructure on this track.

01

What runs

Shared UI/Connect services in Workers; Artifacts canonical Git; D1 collaboration/policy; SQLite Durable Objects coordinating reads and execution.

02

How the workflow grows

Queues, Workflows and Containers produce evidence. Private R2, managed registry and protected Worker apps retain outputs. Optional AI, browser snapshots, email and operations add focused consumers.

03

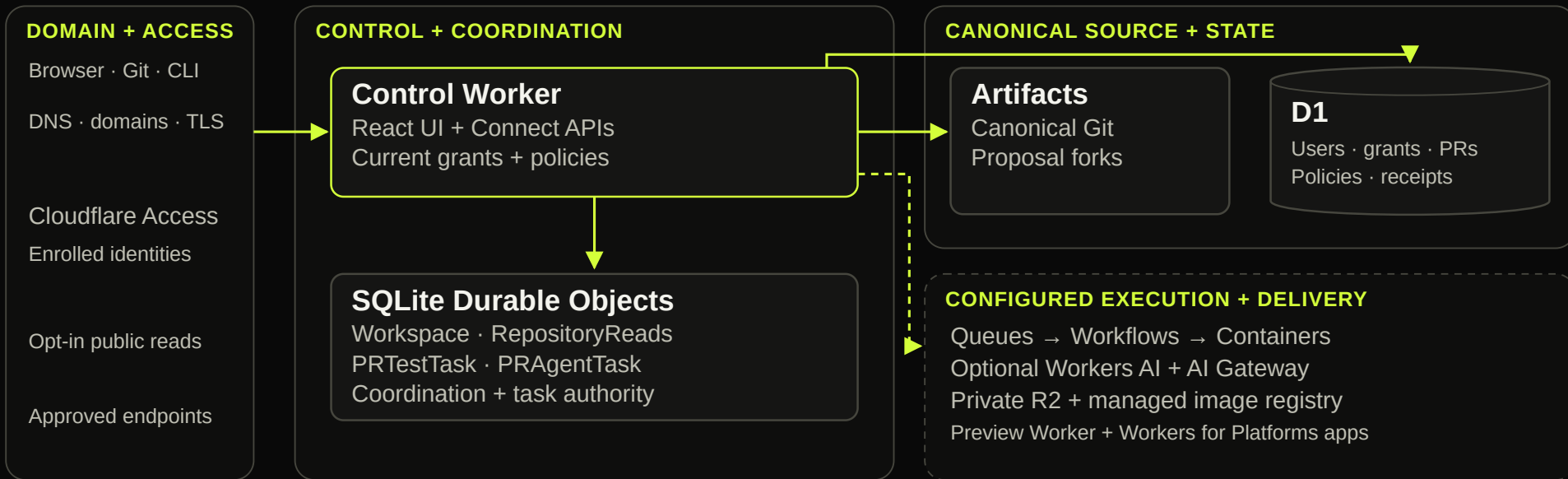
How to deploy

Alchemy or the native cf recipe provisions owned resources under the selected account, Access identity, domains, images and permissions. Verify actual schema, grants, readback and cleanup.

Cloudflare: where each part lives.

Workers own the application, Artifacts owns Git, and D1 owns collaboration state.

CLOUDFLARE / NATIVE OWNERSHIP AND PLACEMENT



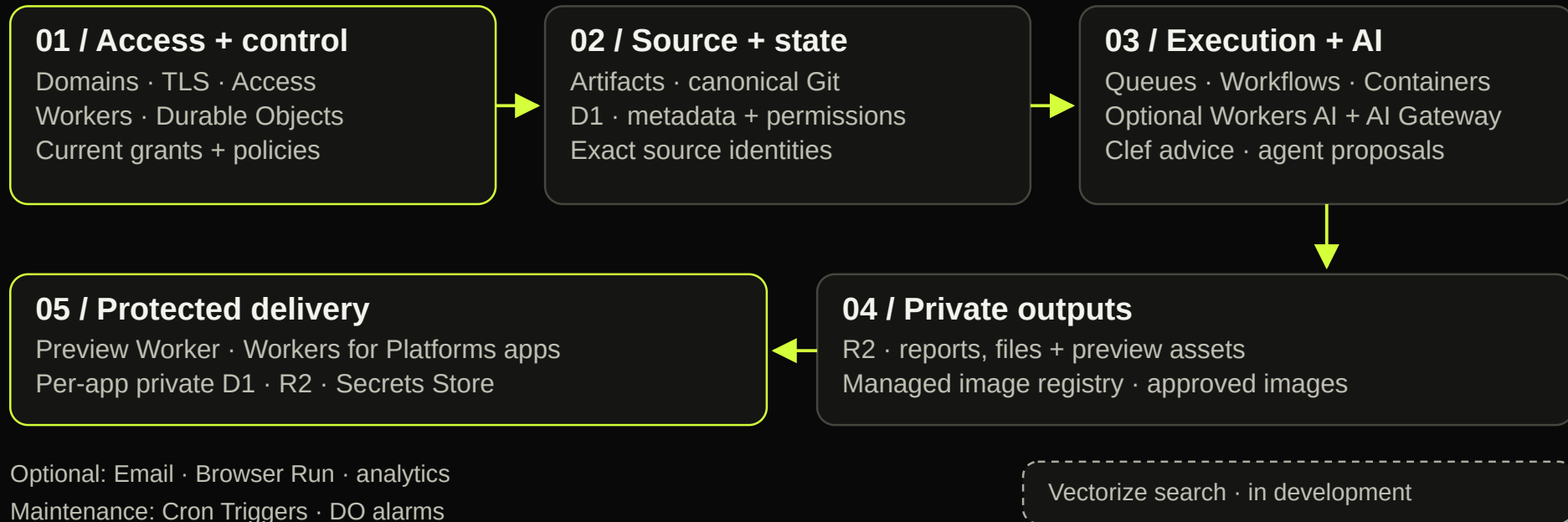
Configured services, current permissions and execution evidence determine available workflows.

[Editable architecture diagram](#) · [Open full size](#)

A platform built from managed primitives.

Each service has a specific role. Golf retains current authorization.

golf. CLOUDFLARE / NATIVE PLATFORM



[Editable architecture diagram](#) · [Open full size](#)

Canonical Git. Explicit authority.

The control plane is native to Cloudflare.

01

Workers

Shared React UI, generated Connect APIs, Git brokerage and trusted service entrypoints.

02

Artifacts

Canonical Git objects and refs, source/history and independent proposal forks.

03

D1

Accounts, repository grants, collaboration, policy, operation receipts and job records.

04

Durable Objects

Workspace and RepositoryReads coordination; PRTestTask and PRAgentTask execution authority.

Durable admission. Bounded compute.

Run a selected exact-source profile and retain what actually happened.

01

Queues + dead-letter queue

Buffer evidence admission; optional DLQ retains exhausted messages for deliberate recovery.

02

Workflows

Retain staged orchestration and outcomes; never grant repository authority.

03

Containers

Approved images execute checks and proposals with bounded source and network access.

04

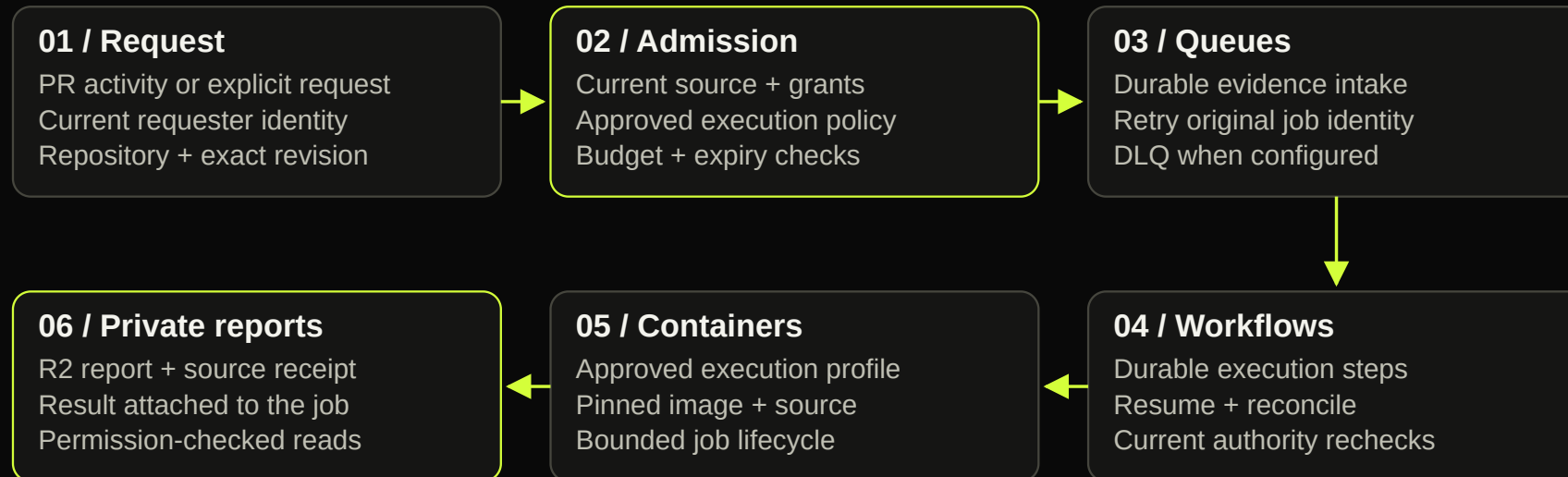
Sandbox SDK file helpers

File staging inside raw Containers. A separate general Sandbox runtime is not implied.

Follow one exact-source check.

From authorized admission to retained private evidence.

g **golf.** EXACT-SOURCE EXECUTION



Job retries keep the original identity; readiness requires completed execution.

[Editable architecture diagram](#) · [Open full size](#)

Keep the source. Keep the proof.

Use different storage for different responsibilities.

01

Artifacts

Git history and proposal forks, with current reader/writer checks.

02

Private R2

Reports, agent context, preview bundles, release assets and discussion attachments.

03

Managed image registry

Approved execution images and CI publication on registry.cloudflare.com; Golf retains digest receipts.

04

Secrets Store

Broker authority, signing material and encryption keys; never browser-held provider secrets.

Source, state and outputs.

Give each kind of data the right home.

golf. SOURCE / STATE / PRIVATE OUTPUTS

Artifacts · Git

Canonical objects + refs
Independent proposal forks
Exact commit + tree identities

D1 · authority + state

Current users · grants · pull requests
Policies · approvals · job journals
Original receipts + version fences

Private R2 · outputs

Reports + agent context bundles
Preview assets + permitted uploads
Authorized reads + integrity receipts

Managed image registry

Approved container images
Pinned image identities + digests
Separate image publication authority

Golf checks current repository access before each private read.

[Editable architecture diagram](#) · [Open full size](#) ↗

Advice and proposals have different jobs.

Optional inference serves a specific workflow.

01

Clef advice

Metadata-only PR reasoning through Workers AI. Advisory evidence; no approval or merge authority.

02

GLM 5.3 Flash proposals

Maintained harness in an independent Artifacts fork. Approved file changes, tests and private reports.

03

AI Gateway

Authenticated inference; logs and cache disabled. Configured 60/min and \$5/day limits.

Two AI paths. Explicit authority.

Clef advice and separately tested proposals serve different jobs.

g **golf.** AI / ADVICE AND REVIEWABLE PROPOSALS

Clef · metadata-only advice

Worker → optional Workers AI + AI Gateway · no repository code context

01 / Agent workspace

Maintained harness · Container
Approved execution profile
Independent Artifacts fork



02 / Bounded inference

Workers AI + AI Gateway
Exact fork source context
Operator-enabled AI budget



03 / Verify proposal

Approved tests on the fork
Private R2 patch + reports
Exact-source proposal receipt



04 / Human review

Current repository access + source revision · inspect the patch and evidence
Proposal generation leaves canonical Git unchanged; agents receive no repository values.

[Editable architecture diagram](#) · [Open full size](#)

Review a running change.

Temporary, source-bound previews keep production separate.

01

Preview Worker + R2

Broker the exact static bundle with current reader checks and expiry.

02

Workers for Platforms

Dispatch approved modules with private per-preview D1/R2 and signing material.

03

Outbound Worker

Enforce the managed app egress policy instead of handing apps unrestricted Internet access.

04

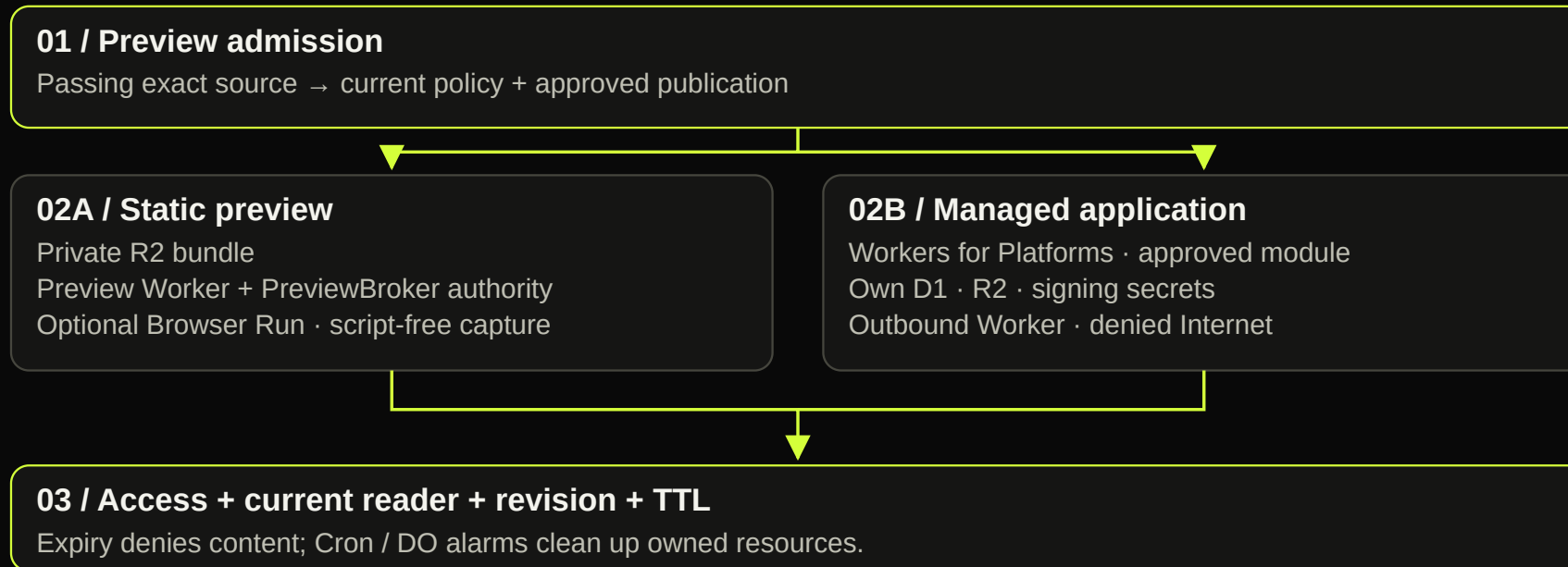
Browser Run

Optional script-free static HTML snapshots. Full interactive application QA remains open.

The preview has its own boundary.

Protect readers and isolate the application's data and network access.

g **golf.** PROTECTED PREVIEWS / TWO DELIVERY PATHS



[Editable architecture diagram](#) · [Open full size](#)

Identity is the start of permission.

Access, Golf authority and deployment policy work together.

01

Access / Zero Trust

Separate human OTP and machine service admission; Golf requires an enrolled account and current grants.

02

DNS, custom domains + TLS

Owned-zone control/preview hostnames and email sender records. Protection covers approved endpoints.

03

Secrets + environment policy

Encrypt repository values; approve exact-commit PR workflow release. Restrict preview branch, lifetime and concurrency.

04

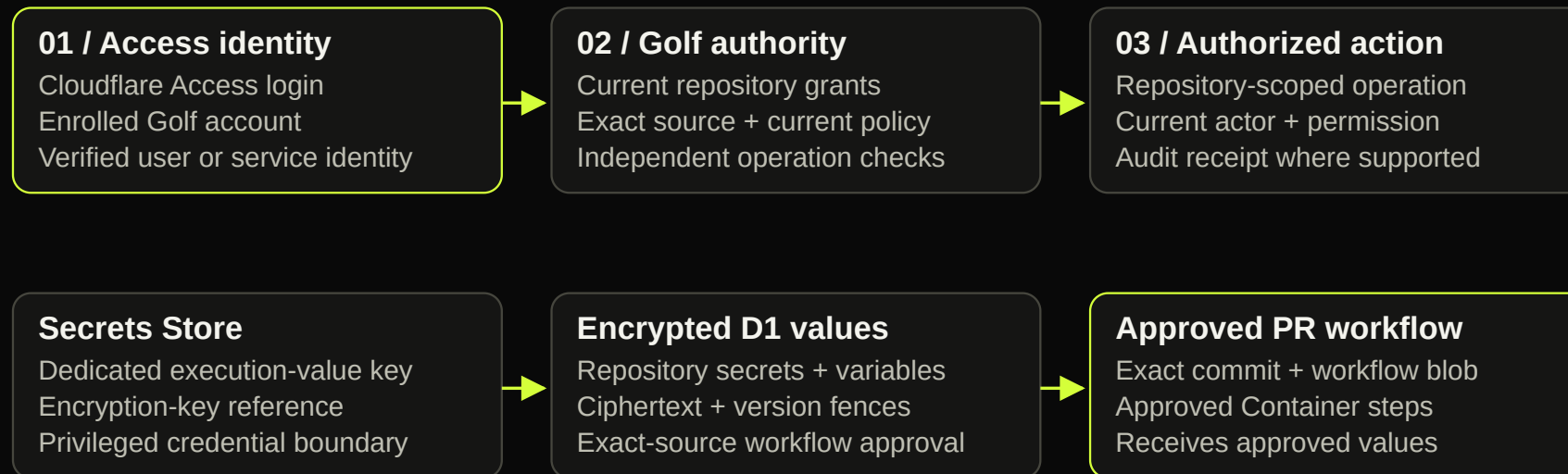
Workers rate limiting

Approximate anonymous read/clone throttling by network and location. Errors fail open; not a global quota.

From identity to permission.

Release secrets only to the approved exact-commit workflow.

golf. IDENTITY / AUTHORITY / WORKFLOW SECRETS



Only approved workflows receive repository values; agent jobs receive none.

[Editable architecture diagram](#) · [Open full size](#)

Keep people connected to the work.

Real messages retain consent, current access and delivery observations.

01

Email Service

Own-address subscriptions and consent; sender subdomain and DNS onboarding. Retained provider acceptance is recorded.

02

Signed webhooks

Encrypted subscriptions, approved origins and stable event identity. A retained Slack test delivery was accepted.

03

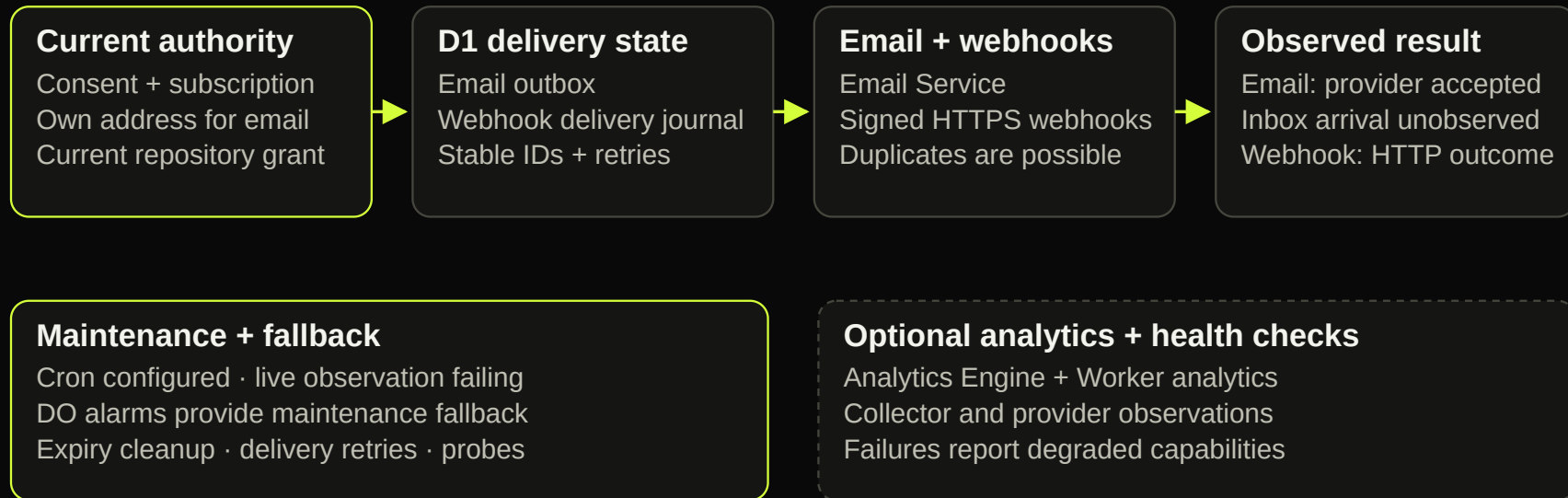
D1 outboxes

Maintenance and bounded retries carry notification delivery. These consumers do not use the evidence queue.

Keep the operating loop visible.

Separate accepted notifications from maintenance and collection health.

g **golf.** NOTIFICATIONS / MAINTENANCE / OBSERVATIONS



Email uses D1 outbox maintenance; evidence Queues are a separate execution path.

[Editable architecture diagram](#) · [Open full size](#)

Know what is running. Know what failed.

Infrastructure definitions and actual outcomes are separate evidence.

01

Analytics Engine

Bound optional evidence metrics; hosted ingestion/readback is not yet verified.

02

Workers analytics + observability

Separate read-only analytics token and enabled provider observations; analytics collection has recorded failures.

03

Cron + Durable Object alarms

Cron schedule is configured but delivery is failing on this account.
Workspace alarms have completed maintenance.

04

Alchemy + native cf CLI

Owned resource provisioning, additive schema migration, token renewal, readback and cleanup.

A bounded merge you can inspect.

Native standard merge has passed hosted Git checks.

01

Supported first slice

Same-repository merge commit when the base is already an ancestor of head; verified ref/commit readback.

02

Explicit refusals

Squash, diverged branches, forks, drafts and locked targets are refused.

03

Still to qualify

Complete trusted protected publication, broad merge semantics and general production promotion.

Search beyond exact matches.

Semantic code and issue search is being added.

01

Workers AI embeddings

Create semantic representations from approved source content.

02

Vectorize

Index and query those representations to find related code and issues.

03

Current reader access

Filter results by live repository authority and propagate deletion/revocation.

A developer preview with real evidence.

Dated engineering proofs cover selected Git, test, proposal and protected-preview paths. Production readiness still needs testing.

01

Demonstrated

Hosted Git, collaboration, approved tests, Clef advice, scoped proposals, protected previews, email acceptance and secret-to-step delivery.

02

Incomplete or degraded

Cron delivery and analytics collection; Analytics Engine ingest; retained screenshots; broader role/load/cleanup matrices.

03

Next acceptance

Full parity, protected merge, production CD, native/Forgejo recovery, tenant onboarding and scale. Vectorize search is in development.

YOUR NEXT MOVE

Try Golf for yourself.

Start with a browser simulation, then install Golf to evaluate real workflows.

01

Browser simulations

gitgolf.app/demo: explore local and Cloudflare interfaces with sample data.
No backend or live execution.

02

Choose your deployment

Docker Compose or Golf's Helm chart for portable hosting. Use the native Alchemy recipe in your own Cloudflare account.

03

Evaluate real work

Enroll a user and grant repository access on your installation. Check an exact commit and inspect its report.
docs.gitgolf.app

gitgolf.app/try ↗