Operate the legacy bundled deployment

Use this guide to keep an existing bundled Cal.diy or Hermes Agent deployment controlled while you prepare its migration to standalone connector hosts. The procedure freezes the compatibility baseline, preserves durable state, limits the ex

SECTION	Deploy and Operate
TYPE	Guide
PROTOCOL	AIP 1.0
SOURCE	docs/guides/legacy-bundled-deployment.md

The migration-only binary is `aipd-legacy-bundled`. Its command-line parser retains the former `aipd` command name and configuration surface for compatibility, but its Cargo package and produced executable use the legacy name. The package is marked `publish = false`; it is not a general release artifact or the recommended starting point for a new deployment.

This guide reflects source revision `97be86e9efedf07ecf1783b03800f683f107fb04`. It describes implementation behavior, not an AIP protocol requirement or evidence that a particular deployment has qualified for production use.

Confirm that this guide applies

Use the legacy binary only when all of these conditions are true:

- an existing deployment already runs a product connector inside `aipd`;
- that connector is Cal.diy or Hermes Agent;
- an immediate fleet migration would create more operational risk than a

bounded compatibility period;

- the operator can preserve the current artifact, configuration, state,

credentials, and provider-side operation records;

- the deployment has an owner and a dated migration exit.

The public connector boundary is exact:

Public connector	Legacy bundled path	Target path
Cal.diy	Temporary compatibility only	Standalone Cal.diy host
Hermes Agent	Temporary compatibility only	Standalone Hermes Agent host
Chatwoot	Not bundled	Standalone Chatwoot host
Dify	Not bundled	Standalone Dify host
CrewAI	Not bundled	Standalone CrewAI host
Twenty	Not bundled	Standalone Twenty host

Do not place Chatwoot, Dify, CrewAI, or Twenty configuration in the legacy process. Do not enable non-public development modules in a public deployment. For a new installation, start with the product-neutral `aipd` gateway and deploy the connector fleet.

Freeze the compatibility baseline

Before changing the process, capture the exact deployment you may need to restore. Record:

- the `aipd-legacy-bundled` artifact digest and source revision;
- the rendered command name, complete non-secret arguments, and environment

variable names;

- the service ID, trust domain, bind address, and public base URL;
- the selected bundled connector and its endpoint or account boundary;
- the canonical manifest and MCP tool catalog;
- runtime, replay, transaction, approval, callback, and connector state;
- provider-side operation IDs for active or unresolved mutations;
- credential references, scopes, issuers, and rotation timestamps without

copying secret values;

- ingress destinations, signature policy, and replay-store location;
- the last healthy and ready observations.

The source includes compatibility tests for the former CLI flags, normalized default readiness, default manifest, and default MCP tool names. Those tests protect a source baseline. They do not prove that your artifact, configuration, database, connector endpoint, or external product is healthy.

Build only the pinned package selected for the compatibility window:

```
SH
cargo build --locked --release -p aipd-legacy-bundled
```

Retain the resulting `target/release/aipd-legacy-bundled` digest with the deployment record. Do not substitute a newly built binary during an incident unless its source and configuration compatibility have been reviewed.

Choose a durable state boundary

The process selects runtime storage in this order:

1. PostgreSQL when `--postgres-url` or `AIPD_POSTGRES_URL` is present;
2. durable local runtime state when `--storage-dir` or `AIPD_STORAGE_DIR` is

present;

3. in-memory runtime state when neither is present.

Use PostgreSQL for a clustered logical deployment. Give all replicas in that logical deployment the same runtime database and give no unrelated service access to it. Use a local storage directory only for a deliberately single-process deployment whose filesystem, ownership, backup, and restore behavior are controlled.

Do not treat in-memory state as a recoverable deployment. A restart can lose action state, idempotency reservations, transaction progress, callback deliveries, approvals, and reconciliation evidence.

When a storage directory is configured, other local facilities can also use it, including MCP correlation state and the default Cal.diy webhook replay file. If PostgreSQL and a storage directory are both configured, core runtime state uses PostgreSQL while these file-backed facilities can still depend on the directory. Back up and restore each active store as one coordinated boundary.

Before restart or upgrade:

1. stop accepting new mutation work;
2. enumerate active, waiting, and outcome-unknown actions;
3. retain provider operation identifiers and idempotency keys;
4. create a restorable database or filesystem backup;
5. record the last completed runtime and connector observations;
6. verify the restoration procedure outside the live state path.

Harden the external boundary

The default listener is `127.0.0.1:8080`. Keep it on loopback behind a trusted TLS and authentication boundary, or bind externally only with an explicit public HTTPS origin. A non-loopback listener without `--public-base-url` is rejected outside development mode. The public URL must be an HTTPS origin without credentials, path, query, or fragment.

Incoming native AIP envelopes require signatures by default. Configure the expected `did:key` signer-to-principal bindings and the trusted identity and approval-authority files for the identities the deployment accepts. If native HTTP bearer authentication is used, prefer `--native-bearer-token-file` over an inline token and bind the authenticated principal, tenant, and scopes explicitly.

`--allow-insecure-development` permits unauthenticated development traffic only on a loopback listener. It is not a production fallback. The implementation rejects insecure development mode on a non-loopback address.

Review each optional surface separately:

Surface	Required control
Native HTTP	Signed envelopes and trusted signers, or an explicit authenticated edge
MCP HTTP	OAuth introspection or a bounded local bearer policy, required scopes, and allowed browser origins
MCP stdio	A trusted spawning process and a private local channel
NATS	Authenticated server connection, bounded subjects, trust domain, queue ownership, and timeout
Callbacks and A2A push	Exact destination hosts, signing identity, encrypted push credentials, and HTTPS
Delegation	Authenticated peer identity, pinned DID, trust domain, and bounded route

Do not enable an unused transport. Plain HTTP and private-network callback exceptions expand the destination boundary and should remain off outside a reviewed local topology.

Keep secret material out of process arguments

Use owner-controlled regular files whenever the legacy interface provides a file option. Secret-file loading rejects symbolic links, empty files, oversized values, and group or other access on Unix. This applies to Cal.diy credentials, native HTTP bearer tokens, NATS passwords, MCP introspection secrets, and the connector-fleet signing seed.

Some retained compatibility options accept secret values only through an argument or environment variable. Hermes Agent uses `--hermes-api-key` or `AIPD_HERMES_API_KEY`; callback signing and A2A push encryption use hexadecimal values supplied through their arguments or environment variables. These are legacy limitations, not preferred secret-delivery interfaces.

For such a feature, inject the value through a restricted service manager or secret provider, prevent diagnostic and environment access by unrelated users, redact crash and support bundles, and rotate the credential after suspected exposure. Do not place secret-bearing commands in shell history, unit files, deployment records, or documentation examples. If the deployment cannot control this exposure, leave the feature disabled and prioritize migration.

Configure one bundled connector boundary

Enable only the bundled connector that the existing deployment needs. Supplying any part of a connector's configuration makes startup validate the complete boundary; a partial configuration fails instead of silently omitting it.

Cal.diy

Cal.diy requires a base URL, a stable deployment-local account ID, and exactly one authentication mode:

- a bearer-token file; or
- an OAuth client ID plus client-secret file.

Tenant-to-account routing requires both tenant-account mappings and credential-file mappings, plus a trusted identity directory that binds the same external accounts. Webhook replay storage requires at least one webhook secret mapping. Without an explicit replay file, the configured storage directory supplies the default file; without either, replay observation is in-memory.

Preserve the existing account and tenant boundary during compatibility operation. Do not use this page as the source for Cal.diy capability semantics or product qualification; use the connector's local documentation.

Hermes Agent

Hermes Agent requires one or more `ID=URL` endpoint mappings. The configured API key is shared across those endpoints. Keep one deployment boundary only when that shared credential and endpoint set have the same owner and trust policy.

Operator delegation is optional and disabled unless explicitly enabled. If it is enabled, review the model route, trusted instruction source, time and input bounds, event limit, delegation depth, claim lifetime, cancellation grace, capability prefixes, scope prefixes, and approval exemptions as one policy. Do not broaden an exemption merely to preserve old behavior.

Start under a process supervisor

Render the selected manifest before opening traffic. Clone the exact reviewed service invocation into a non-serving validation step, add `--print-manifest`, and retain its JSON output. The flag initializes the configured deployment, prints the resulting manifest, and exits. Running it without the selected connector settings prints the default manifest, not the deployment you intend to operate.

Run the actual process through a supervisor that sets a restrictive service user, read-only artifact, controlled working directory, bounded file descriptors, restart policy, and access to only the required secrets and state. Pass the reviewed non-secret settings explicitly. The following shape omits all connector and credential values by design:

```
SH
exec target/release/aipd-legacy-bundled \
  --bind 127.0.0.1:8080 \
  --public-base-url https://aip.example \
  --service-id service:aipd:legacy \
  --trust-domain example \
  --storage-dir /var/lib/aipd
```

Replace local storage with `AIPD_POSTGRES_URL` for a clustered deployment. Configure either Cal.diy or Hermes Agent through the supervisor's protected configuration, then compare the emitted manifest with the frozen baseline.

The process handles interrupt and termination signals through graceful HTTP shutdown. It gives background runtime and NATS tasks a bounded period to stop. This does not prove that an external provider mutation reached a terminal state. Reconcile active work before stopping the old process.

Verify before accepting traffic

Check the surfaces in this order:

1. `GET /health` confirms that the HTTP process answers; it is a liveness

check, not a routing or connector proof.

2. `GET /ready` must return HTTP 200 and `status: ready`. Read the gateway,

supervisor, and module details; required modules, the runtime worker, and a configured NATS listener participate in readiness.

3. `GET /aip/v1/manifest` must contain only the reviewed public capabilities

for the selected configuration.

4. `GET /metrics` must be reachable through the observability boundary and

show the daemon ready without a repeatedly failing worker.

5. Authenticated capability discovery must return only the caller's intended

tenant-visible contract.

6. An approved non-mutating connector action must reach the expected provider

account or endpoint and return a typed result.

Do not begin with a create, update, delete, message-send, payment, or other mutation. A healthy process with an unexpected manifest, tenant boundary, or provider target is not ready for traffic.

Use the fleet bridge only for migration

The legacy binary can install the same product-neutral registry catalog and remote handler used by `aipd`. This permits an incremental migration: one connector can remain bundled while another tenant binding routes to an admitted standalone host.

Fleet mode is atomic at startup. It requires the connector-registry URL, an owner-only signing-seed file, and either an explicit connector-host allowlist or trust in admitted registry endpoints. A partial set of fleet options is rejected.

Keep one mutation authority for each tenant and capability. Do not expose a bundled implementation and an enabled fleet binding for the same provider boundary at the same time. Follow Migrate bundled connectors to the fleet for the drain, disabled-binding verification, cutover, and rollback sequence.

Recover without duplicating mutations

If readiness fails, keep traffic closed and classify the failure before restarting:

Observation	First action
Configuration error on startup	Correct the rejected field; do not weaken the security check
Runtime store unavailable	Restore the same durable store before accepting work
Required module not ready	Verify connector endpoint, credential, account boundary, and module detail
Runtime worker stopped	Keep the process out of service and inspect durable work before restart
NATS required but listener down	Restore the authenticated NATS path or remove the reviewed dependency
Manifest drift	Stop rollout and restore the exact artifact and non-secret configuration
Provider outcome unknown	Reconcile by provider operation ID and idempotency key before retry

Do not erase local state or replay files to make readiness pass. Do not retry an ambiguous mutation under a new action ID. Preserve the failed process's state, logs, metrics snapshot, manifest, and provider evidence until recovery or rollback is complete.

End the compatibility period

The legacy deployment is ready to retire when:

- every public connector runs in an admitted standalone host;
- the product-neutral `aipd` owns gateway, catalog, routing, and central event

ingress responsibilities;

- provider credentials are absent from the gateway process;
- each tenant capability has exactly one enabled execution path;
- active bundled actions are terminal or reconciled;
- webhook and callback destinations point to the reviewed target boundary;
- rollback artifacts and retention obligations are satisfied;
- the legacy artifact, configuration, secrets, and state are revoked or

archived according to policy.

After cutover, remove the compatibility binary from service definitions and artifact promotion. Keep it only as a retained rollback artifact for the approved window. Use Connector host lifecycle and Connector credentials and rotation for ongoing fleet operation.