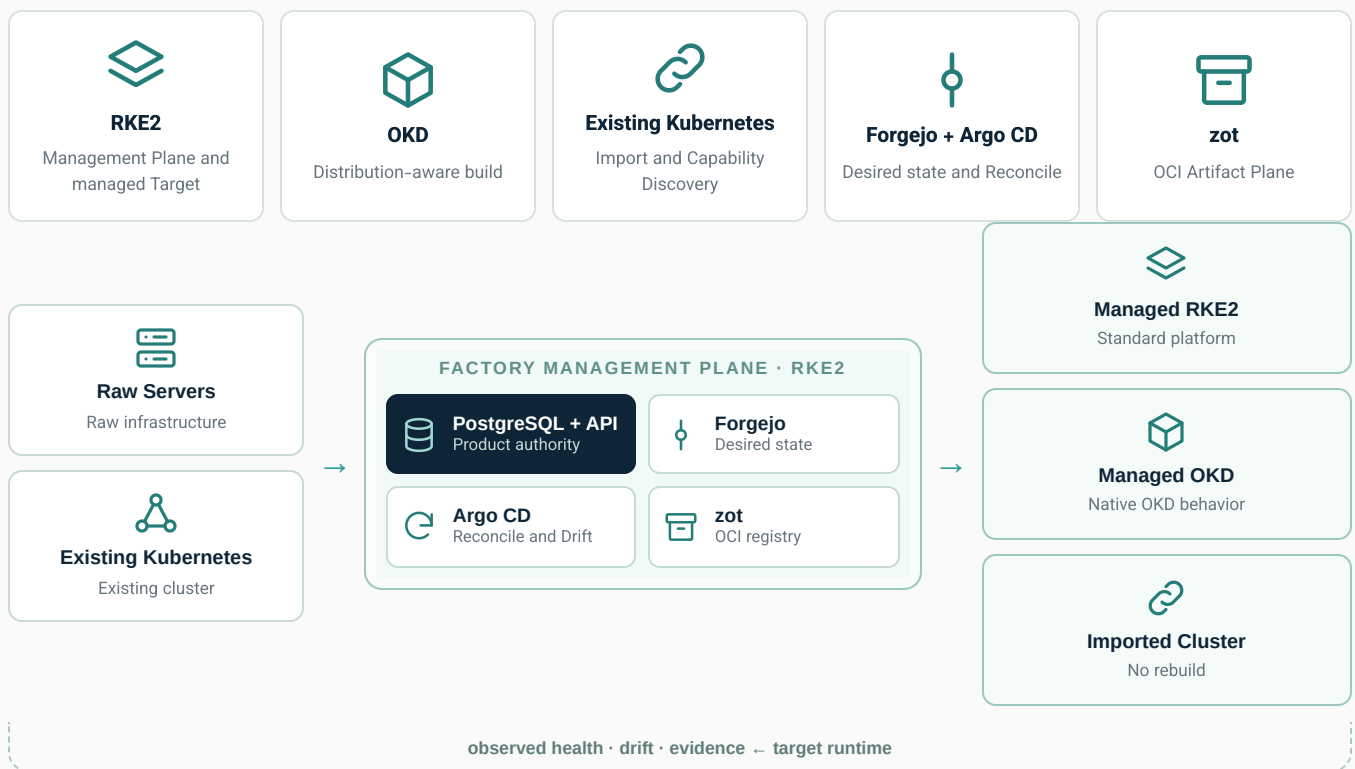


4SO Platform Factory

Technical Brochure

Build, Import and Operate Application Platforms

Turns raw servers or existing Kubernetes into standardized, operable platforms; Catalog, Fleet, Release, Policy, Delivery and recovery sit under one Product Control Plane.



Authority separate from Runtime

PostgreSQL and the Platform API are the product authority; Forgejo holds desired state and Argo CD is only the Reconciler.

Distributions stay native

RKE2 and OKD are not reduced to one artificial implementation, and no duplicate Stack is installed.

The Factory does more than build Clusters; it turns Distribution, Capability, Artifact and Lifecycle into one Platform Contract.

CAPABILITY SNAPSHOT

Capability Snapshot

From raw infrastructure to application platform

The Factory is not just an installer; it covers the full Platform Engineering lifecycle, from Target creation to Workload delivery and Day-2 Operations.



Create / Import
Build a managed Target or enroll existing Kubernetes



Catalog & Blueprint
Versioned Templates and Capabilities for teams



Fleet Operations
Health, Drift and Maintenance across Targets



GitOps
Forgejo keeps history; Argo CD converges it



Registry / Disconnected
zot and Internet-independent delivery



Workload Delivery
Application Release and Environment Binding



Backup & Restore
Policy, Backup Run, Restore and Drill



Node & Upgrade
Distribution-aware Drain, Replace and Upgrade

Product ownership boundary

PostgreSQL + Platform API – the Product Authority	
Forgejo – desired state, not Runtime truth	
Argo CD – Reconciler, not Source of Truth	
zot – identity and Provenance of every Artifact	

Operating principles

The Management Plane runs self-contained on RKE2	
Native Target capabilities are not reinstalled	
Build plane and Target lifecycle have separate owners	
UI success never replaces Runtime evidence	



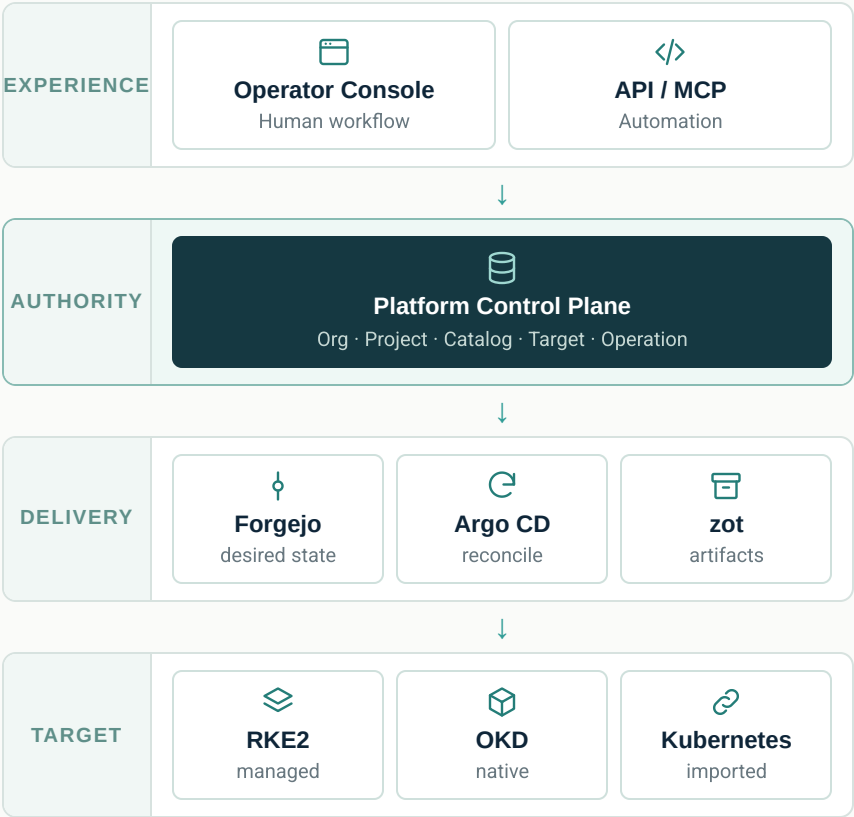
Core contract: Product Authority → Desired State → Reconciliation → Observed Target – none of them replaces another.

ARCHITECTURE OVERVIEW

Architecture Overview

Product authority, desired state and runtime stay separate

The product authority holds decisions and lifecycle, desired-state history is recorded, the Reconciler applies it and the Target shows the result.



- 01 Management Plane ≠ Target**
The management RKE2 is not itself a general-purpose Target.
- 02 zot is the Artifact authority**
The OKD internal Registry does not replace zot.
- 03 Provider ≠ Distribution**
Physical infrastructure and Distribution identity are separate concepts.
- 04 Dapr is only a Trait**
Optional and Target-side; it never takes over product authority.

Domain	Mechanism	Behavior
Managed OKD	openshift-install / oc + Redfish boot media	Bare-metal provisioning goes through OKD's own contract.
Disconnected OKD	oc-mirror v2 → zot	Acquisition, Inventory and Mirror are sealed and reviewable before install.
Runtime trait	optional Dapr sidecar	Invocation, Pub/Sub and Binding; without taking over State, Workflow or Secrets.

TARGET / DISTRIBUTION MODEL

Targets & Platform Profiles

One Factory, several Platform Target types

The Management Plane stays fixed; Distribution differences are resolved at the Target boundary so each environment receives only the Stack it needs.

Target	Entry path	Ownership	Use
Managed RKE2	Factory build	Lifecycle under the Product Control Plane	Standard, self-contained Private Cloud
Existing Kubernetes	Import + discovery	Existing runtime + product lifecycle boundary	Bring existing clusters under management
Managed OKD	Distribution-aware build	OKD owns its native capabilities	OpenShift-compatible platform
Existing OKD	Import + discovery	Native OKD + product workflows	Manage without duplicating internal stacks





RKE2 Platform
Managed · Catalog add-ons · Fleet



OKD Platform
OVN, OLM and SCC aware; no duplicate Stack



Imported Kubernetes
Identity discovery and lifecycle admission



Application Platform
Release, Binding and deployment evidence



Disconnected
Acquire, Mirror and Install without Internet



Fleet
Nodes, Upgrade, Diagnostics and recovery

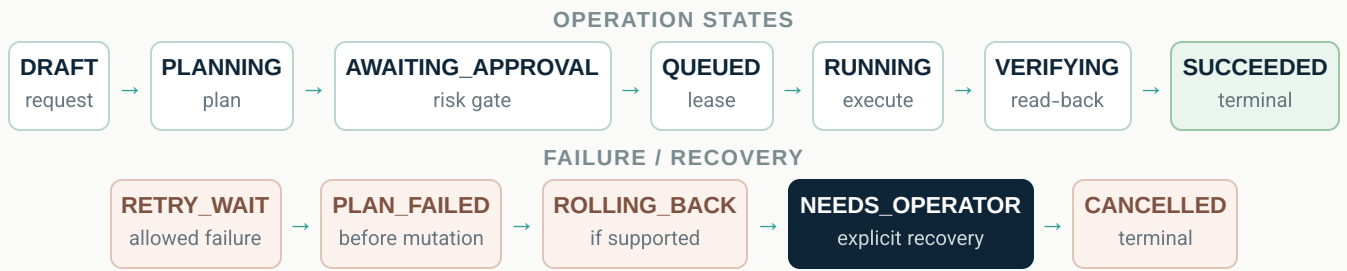
 **Topology-aware:** existing Distribution capabilities are not duplicated; the resolver removes the duplicate stack or makes it optional.

ACTION & WORKFLOW CONTRACT

Action & Workflow Contract

Every Operation has an explicit State Machine from Plan to Verification

Operation class, risk, sealed payload, approval, retry policy and observed verification are part of the product model.



Action	Admission	Execution / Lock	Success	Failure / Recovery
Create Platform	Profile, Provider input and a plannable Blueprint	sealed payload + exact-target fence	convergence with the Profile and observed health	PLAN_FAILED before execution; after a side effect: retry / rollback / operator
Import Cluster	access, identity and capability inventory	no rebuild; classify and register	registered Target with observed capabilities	an incomplete Import is not Ready; conflict → operator
Deploy	immutable Release + EnvironmentBinding	approval-gated with target RBAC	Deployment evidence matching the Binding	unknown outcome: read-only recovery, no redispach
Node Replace	node identity, failure domain and safety admission	lease / fence against concurrent execution	new member Ready, old member drained	RETRY_WAIT or NEEDS_OPERATOR
Upgrade	compatibility, backup and baseline health	staged and Distribution-aware	version and health after each stage	rollback only if the workflow supports it
Disconnected	source lock, inventory and digest	acquire → verify → mirror	Artifacts in zot match the inventory	mismatch → fail-closed

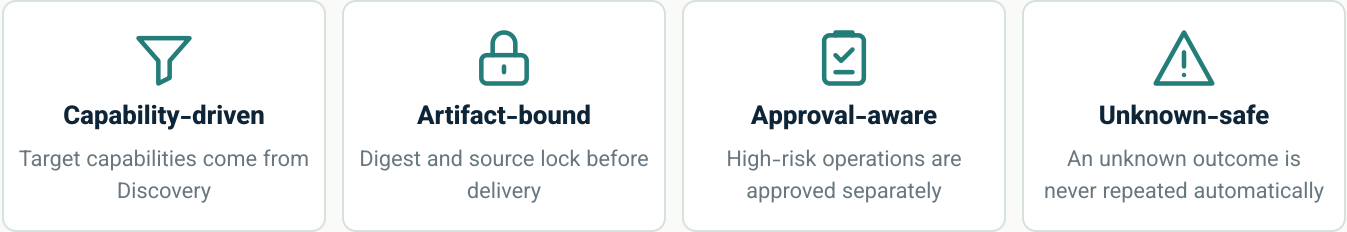
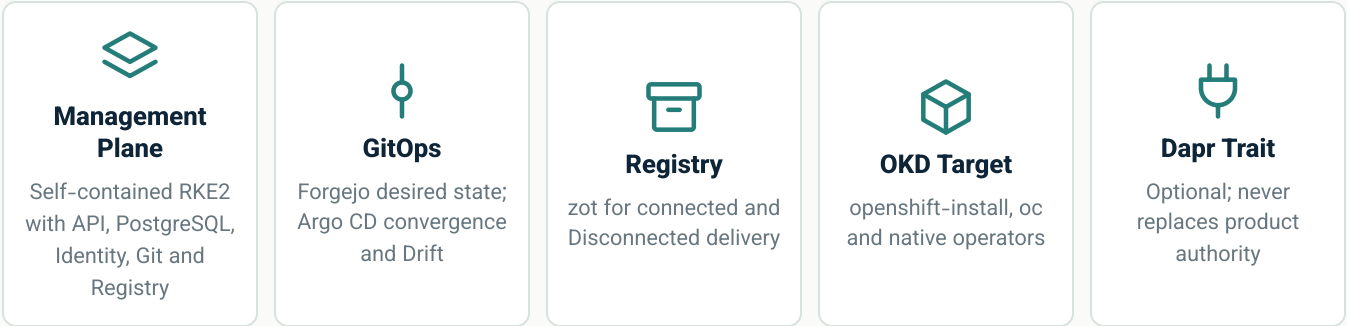
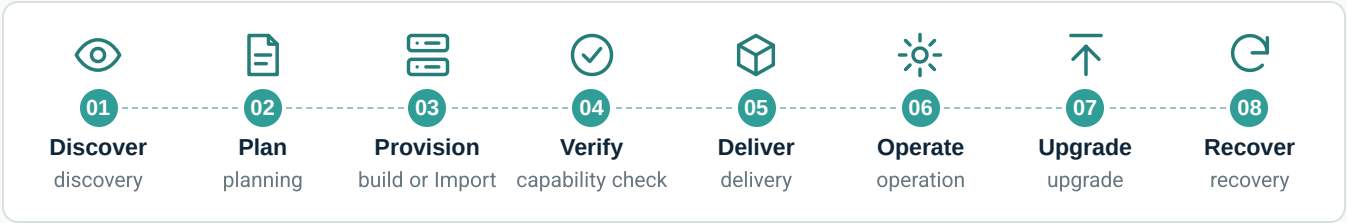
 **Retry is bounded:** only allowed failure classes are retried; rollback is not a general promise and must be supported by that same workflow.

LIFECYCLE OPERATIONS

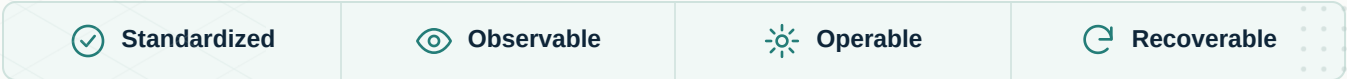
Lifecycle & Components

Platform lifecycle from Discover to Recover

A Target is not only built; discovery, planning, capability verification, delivery, operation, upgrade and recovery are all part of its lifecycle.



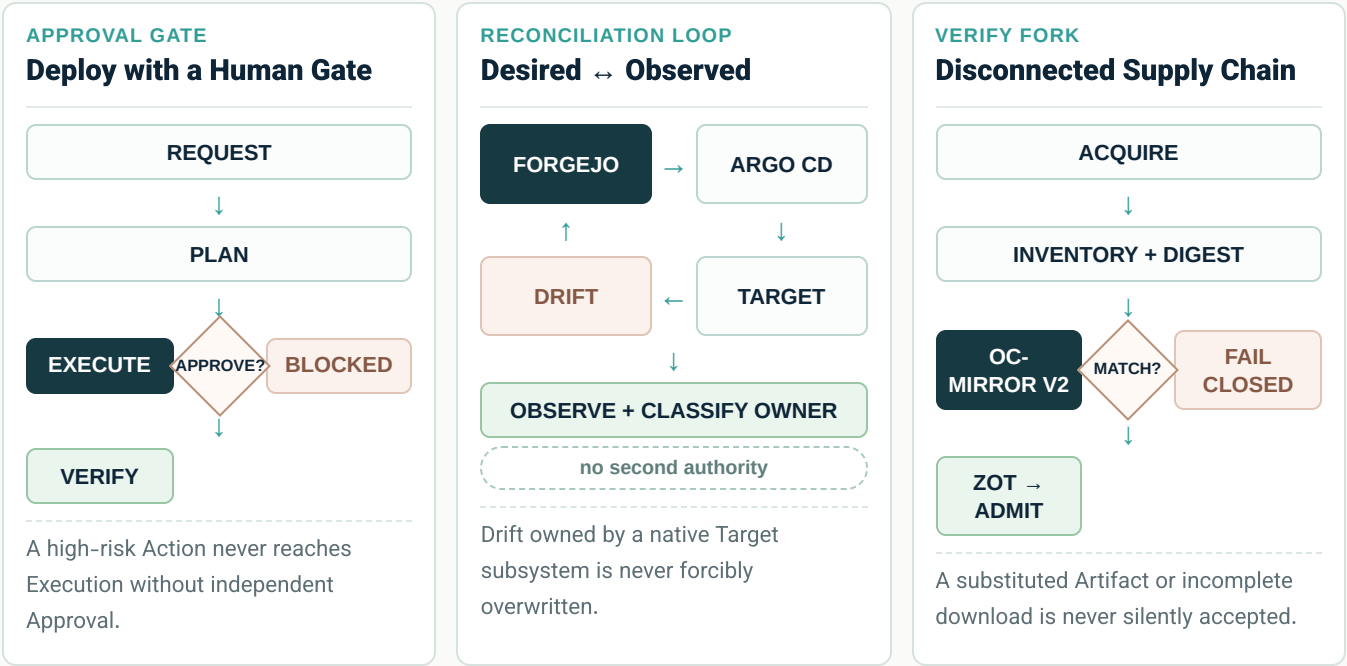
Day-2 First: node lifecycle, Upgrade, Backup, Restore and Diagnostics are part of the product itself, not an add-on.



DEEP TECH WORKFLOWS

Failure & Recovery Semantics

Three non-linear paths, each with a clear failure boundary



Failure Domain	Detection	Product Behavior	Recovery
Management service	health and Operation evidence	Target Workloads continue where possible	restore control services, then Reconcile
Target drift	Argo CD and Runtime observation	Drift is flagged; Authority is not overwritten	Reconcile desired state
Provisioning	Provider and Installer evidence	no false Target registration	repair and resume from the owner step
Artifact mismatch	digest and inventory matching	delivery is blocked	Acquire and Seal again
Unknown outcome	lease lost after a side effect	no automatic redispach	read-only recovery first

Restore Drill: a Backup without verification, or a Restore without Workload-level read-back, is not accepted as recovery evidence.

ACTION & OPERATIONS CATALOG

Action & Operations Catalog

Intent-level operations, not scattered commands

Platform •

- create
- import
- register
- status
- diagnose



Fleet •

- add-node
- drain
- remove
- replace
- repair



Catalog •

- revision
- review
- publish
- deprecate
- revoke



Delivery •

- release
- environment-bind
- deploy
- reconcile
- rollback-when-supported



Data Protection •

- backup-policy
- backup-run
- restore-drill
- restore-run



Disconnected / OKD •

- acquire
- seal
- oc-mirror-v2
- mirror-verify
- upgrade-admit



Identity and Trust

Org/Project membership and Service Accounts in the product authority



API tokens with a one-way digest and scoped permissions



Target credentials held only as references



Observability and Evidence

Observed state shown beside desired state



Delivery Insights from deployment evidence



Bounded Diagnostics with no Secret disclosure



Action safety: there is no general target shell or API proxy; every Action has a defined owner, scope, risk, payload and verification contract.

Move beyond Cluster Management; make Platform Engineering a product.

4SO Platform Factory builds platforms that remain standardized, observable and operable after delivery.

