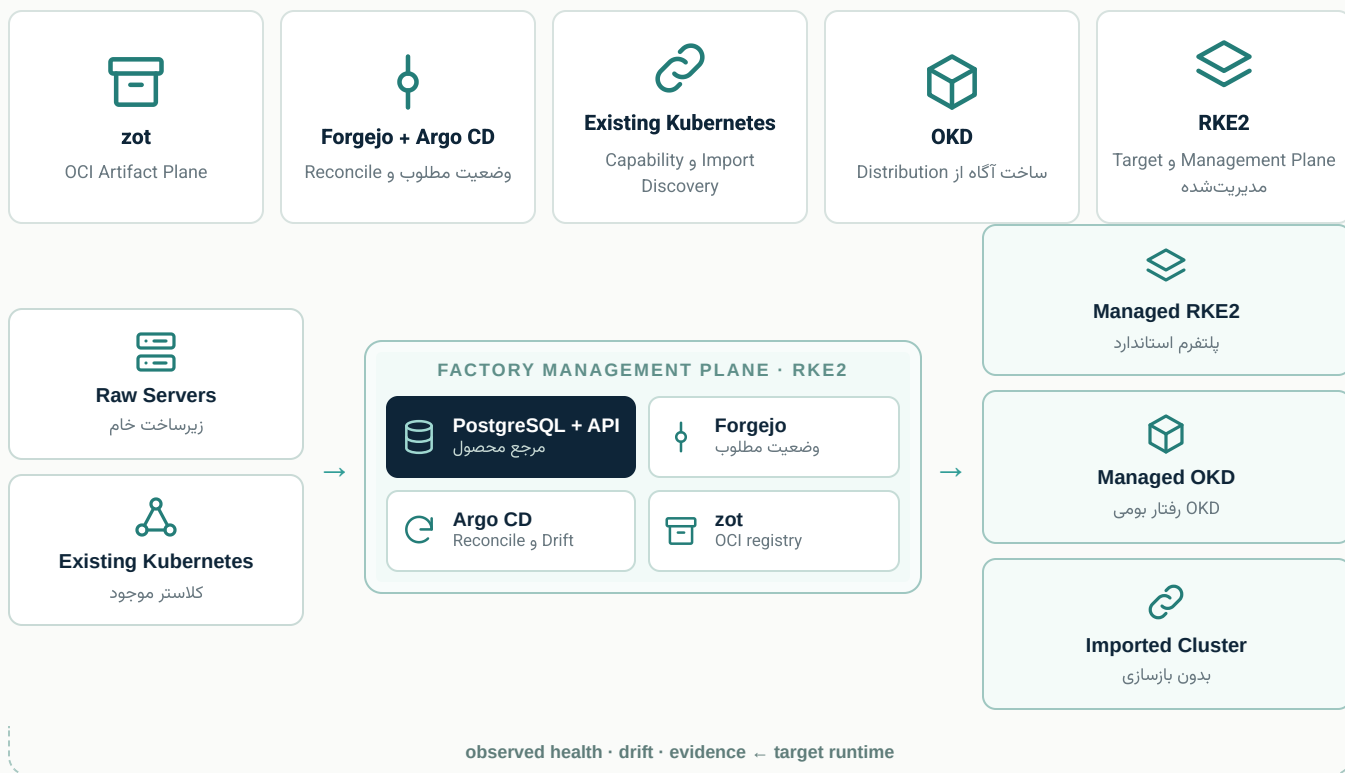


4SO Platform Factory

بروشور فنی

Build, Import and Operate Application Platforms

سرورهای خام یا Kubernetes موجود را به پلتفرم‌های استاندارد و قابل‌اداره تبدیل می‌کند؛
Product Control Plane زیر یک Catalog، Fleet، Release، Policy، Delivery و بازیابی قرار می‌گیرند.



Distribution بومی می‌ماند

RKE2 و OKD به یک implementation مصنوعی تقلیل پیدا نمی‌کنند و Stack تکراری نصب نمی‌شود.

Authority جدا از Runtime

PostgreSQL و Platform API مرجع محصول‌اند؛ Forgejo وضعیت مطلوب را نگه می‌دارد و Argo CD فقط Reconciler است.

Factory فقط Cluster نمی‌سازد؛ Lifecycle و Distribution، Capability، Artifact و Platform Contract تبدیل می‌کند.

CAPABILITY SNAPSHOT

Capability Snapshot

از زیرساخت خام تا پلتفرم کاربردی

Factory فقط یک نصب‌کننده نیست؛ چرخه‌ی کامل Platform Engineering را از ایجاد Target تا تحویل Workload و Day-2 Operations پوشش می‌دهد.



GitOps

Forgejo تاریخچه را نگه می‌دارد و Argo CD همگرا می‌کند



Fleet Operations

Maintenance و Health، Drift برای مجموعه‌ی Targetها



Catalog & Blueprint

Template و Capability نسخه‌دار برای تیم‌ها



Create / Import

ساخت Target مدیریت‌شده یا اتصال Kubernetes موجود



Node & Upgrade

Drain، Replace و Upgrade آگاه از Distribution



Backup & Restore

Policy، Backup Run، Restore Drill و



Workload Delivery

Application Release و Environment Binding



Registry / Disconnected

zot و تحویل مستقل از اینترنت

اصول عملیاتی



Management Plane مستقل روی RKE2 اجرا می‌شود



قابلیت بومی Target دوباره نصب نمی‌شود



Build plane و Target lifecycle دو owner جدا دارند



موفقیت UI جای شواهد Runtime را نمی‌گیرد

مرز مالکیت محصول



Product مرجع – PostgreSQL + Platform API Authority



Forgejo – وضعیت مطلوب، نه Runtime truth



Argo CD – Reconciler، نه Source of Truth



zot – هویت و Provenance هر Artifact

Core contract: Observed Target → Reconciliation → Desired State → Product Authority – هیچ‌کدام جای

دیگری را نمی‌گیرد.

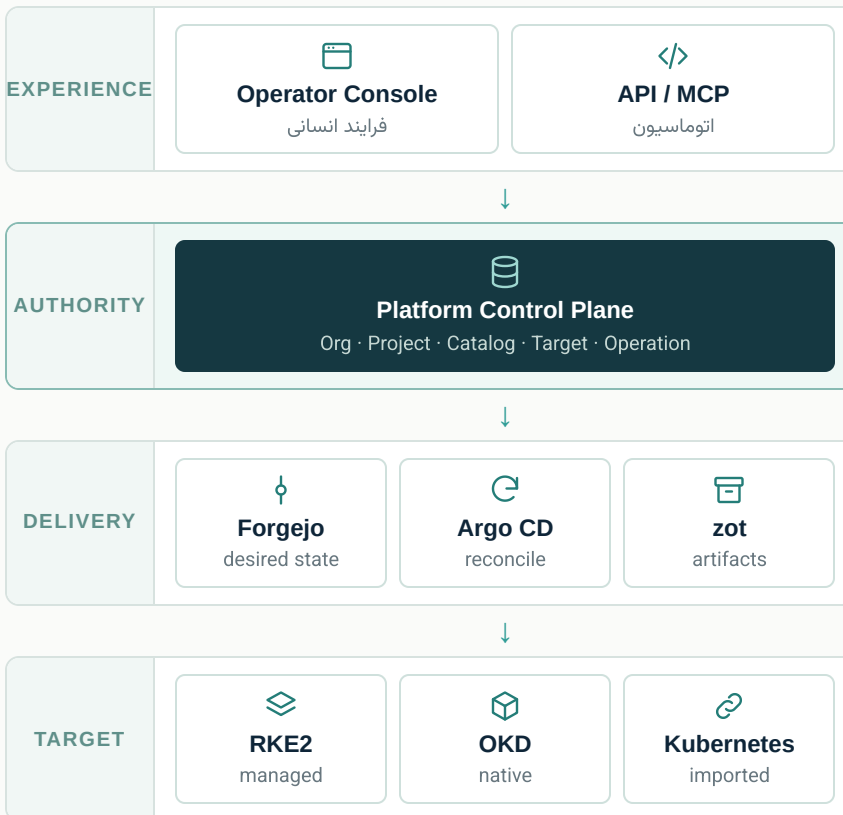


ARCHITECTURE OVERVIEW

Architecture Overview

مرجع محصول، وضعیت مطلوب و محیط اجرا جدا می‌مانند

مرجع محصول تصمیم و چرخه‌ی عمر را نگه می‌دارد، تاریخچه‌ی وضعیت مطلوب ثبت می‌شود، Reconciler آن را اعمال می‌کند و Target نتیجه را نشان می‌دهد.



01 Management Plane ≠ Target

RKE2 مدیریتی خودش Target عمومی نیست.

02 zot مرجع Artifact است

Registry داخلی OKD جای zot را نمی‌گیرد.

03 Provider ≠ Distribution

زیرساخت فیزیکی و هویت Distribution دو مفهوم جدا هستند.

04 Dapr فقط یک Trait است

اختیاری و سمت Target؛ authority محصول را تصاحب نمی‌کند.

حوزه	Mechanism	رفتار
Managed OKD	openshift-install / oc + Redfish boot media	Provisioning روی bare metal از قرارداد خود OKD عبور می‌کند.
Disconnected OKD	oc-mirror v2 → zot	Acquisition، Inventory و Mirror پیش از نصب بسته و قابل بررسی می‌شوند.
Runtime trait	optional Dapr sidecar	Binding و Invocation، Pub/Sub یا Secret بدون تصاحب Workflow، State یا

Targets & Platform Profiles

یک Factory، چند نوع Platform Target

Management Plane ثابت می‌ماند؛ تفاوت Distribution در مرز Target حل می‌شود تا هر محیط فقط Stack لازم خودش را دریافت کند.

کاربرد	Ownership	روش ورود	Target
self- Private Cloud استاندارد و contained	چرخه‌ی عمر زیر Product Control Plane	ساخت توسط Factory	Managed RKE2
آوردن کلاسترهای موجود زیر مدیریت	محیط اجرای موجود + مرز چرخه‌ی عمر محصول	Discovery و Import	Existing Kubernetes
پلتفرم OpenShift-compatible	OKD مالک قابلیت‌های بومی خودش	ساخت آگاه از Distribution	Managed OKD
مدیریت بدون تکرار پشته‌های داخلی	OKD بومی + فرایندهای محصول	Discovery و Import	Existing OKD

RAW → RKE2

RAW → OKD

IMPORT → REGISTER

CONNECTED → DISCONNECTED

RELEASE → DEPLOY



Imported Kubernetes

Identity discovery و پذیرش چرخه‌ی عمر



OKD Platform

آگاه از OLM، OVN، SCC؛ بدون Stack تکراری



RKE2 Platform

مدیریت‌شده · Fleet · Catalog add-ons



Fleet

نود، Upgrade، Diagnostics و بازیابی



Disconnected

Mirror، Acquire و Install بدون اینترنت



Application Platform

Release، Binding و شواهد استقرار

Topology-aware: قابلیت موجود Distribution تکرار نمی‌شود؛ resolver پشته‌ی تکراری را حذف یا اختیاری می‌کند.

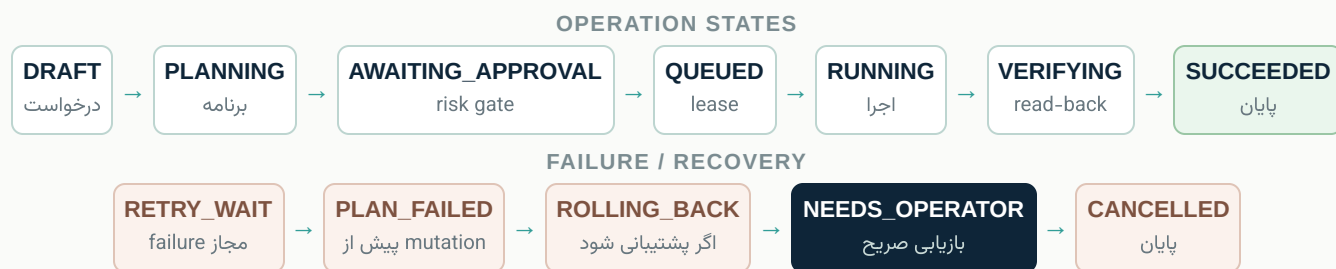


ACTION & WORKFLOW CONTRACT

Action & Workflow Contract

Operation از Plan تا Verification یک State Machine صریح دارد

Operation class، risk، payload، approval، retry policy، مهر و موم شده، verification مشاهده شده بخشی از مدل محصول اند.



Failure / Recovery	Success	Execution / Lock	Admission	Action
PLAN_FAILED پیش از اجرا؛ پس از side effect: retry / rollback / operator	همگرایی با Profile و health مشاهده شده	payload مهر و موم شده + exact-target fence	Profile، ورودی Provider و Blueprint قابل برنامه ریزی	Create Platform
Import ناقص Ready نیست؛ conflict ← operator	Target ثبت شده با قابلیت های مشاهده شده	بدون بازسازی؛ register و classify	دسترسی، identity و capability inventory	Import Cluster
نتیجه نامعلوم: recovery redispatch بدون	شواهد Deployment Binding مطابق	approval-gated target RBAC با	Release تغییرناپذیر + EnvironmentBinding	Deploy
RETRY_WAIT یا NEEDS_OPERATOR	عضو جدید Ready، عضو قدیمی drain شده	lease / fence ضد اجرای همزمان	هویت نود، failure domain و safety admission	Node Replace
rollback فقط اگر workflow پشتیبانی کند	health و version پس از هر مرحله	مرحله ای و آگاه از Distribution	compatibility، backup و health اولیه	Upgrade
mismatch ← fail-closed	Artifact در zot برابر	acquire → verify → mirror	source lock، inventory digest و	Disconnected

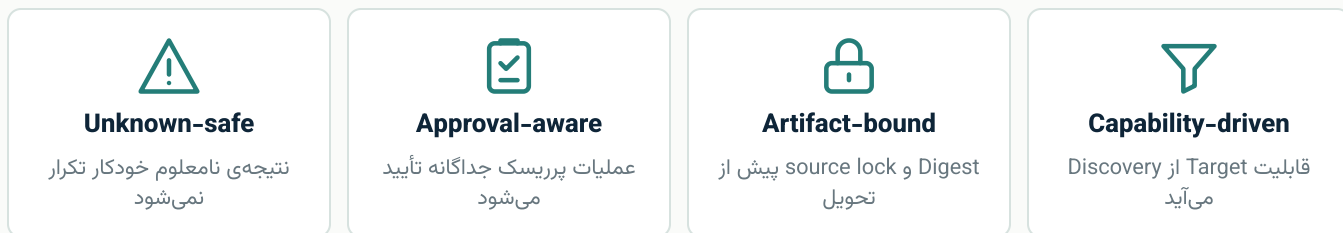
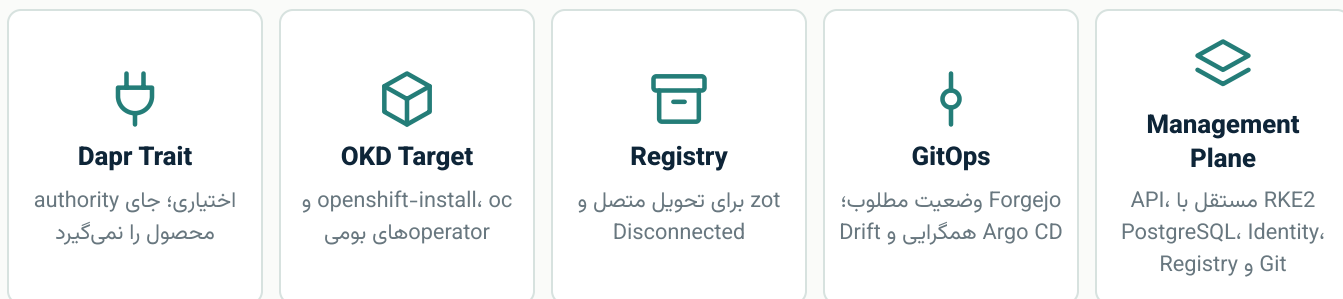
Retry محدود است: فقط failure class های مجاز دوباره تلاش می شوند؛ rollback یک ادعای عمومی نیست و باید توسط همان workflow پشتیبانی شود.



Lifecycle & Components

چرخه‌ی عمر پلتفرم از Discover تا Recover

Target فقط ساخته نمی‌شود؛ کشف، برنامه‌ریزی، بررسی قابلیت‌ها، تحویل، بهره‌برداری، ارتقا و بازیابی همه بخشی از چرخه‌ی عمر آن هستند.



Day-2 First: چرخه‌ی عمر نود، Upgrade، Backup، Restore و Diagnostics بخشی از خود محصول‌اند، نه پیوست آن.



Recoverable

Operable

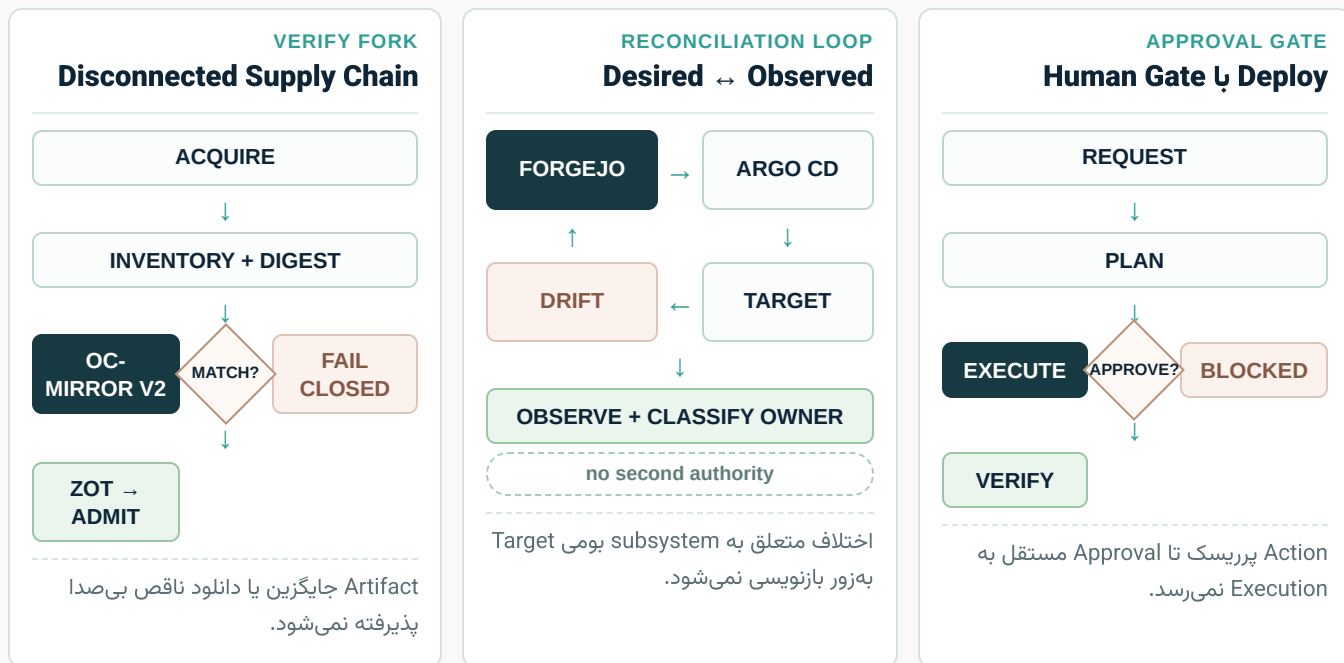
Observable

Standardized



Failure & Recovery Semantics

سه مسیر غیرخطی، هر کدام با مرز شکست روشن



Recovery	Product Behavior	Detection	Failure Domain
بازیابی سرویس های کنترل، سپس Reconcile	Workload های Target در حد امکان ادامه می دهند	Operation health و شواهد	Management service
Reconcile وضعیت مطلوب	Drift علامت می خورد؛ Authority بازنویسی نمی شود	مشاهده ی Argo CD و Runtime	Target drift
تعمیر و ادامه از گام owner	ثبت کاذب Target رخ نمی دهد	شواهد Provider و Installer	Provisioning
Seal و Acquire دوباره	تحويل مسدود می شود	تطبیق digest و inventory	Artifact mismatch
اول recovery فقط خواندنی	بدون redispatch خودکار	از دست رفتن lease پس از side effect	Unknown outcome

Restore Drill: Backup بدون verification و Restore بدون read-back در سطح Workload به عنوان شواهد بازیابی پذیرفته نمی شود.



Action & Operations Catalog

عملیات در سطح Intent، نه دستورهای پراکنده



Fleet •

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



Platform •

- status
- register
- import
- create
- diagnose



Delivery •

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



Catalog •

- publish
- review
- revision
- revoke
- deprecate



Disconnected / OKD •

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



Data Protection •

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

Evidence و Observability |



وضعیت مشاهده‌شده کنار وضعیت مطلوب



Delivery Insights از شواهد استقرار



Diagnostics محدود و بدون افشای Secret

Trust و Identity |



عضویت Org/Project و Service Account در مرجع محصول



API token با digest یک‌طرفه و مجوز محدود



Credential هدف فقط به‌صورت reference

Action safety: هیچ target shell یا API proxy عمومی وجود ندارد؛ هر Action owner، scope، risk، payload و قرارداد verification مشخص دارد.



از Cluster Management عبور کنید؛ Platform Engineering را محصول کنید.

4SO Platform Factory پلتفرم‌هایی می‌سازد که بعد از تحویل هم استاندارد، قابل‌مشاهده و قابل‌اداره می‌مانند.

