OPEN COMPONENT MODEL

Sovereign Cloud and Air-Gapped Delivery

Jakob Möller
Open Component Model

NeoNephos Foundation

THE LINUX FOUNDATION | Europe



Are you ready for sovereign operations?

"Sovereign readiness is an operating model, not a choice."

01
Configure, don’t rebuild
Regional differences belong in a supported configuration contract.

02
Operate independently
Deployment, validation, troubleshooting, and recovery must be operator-executable.

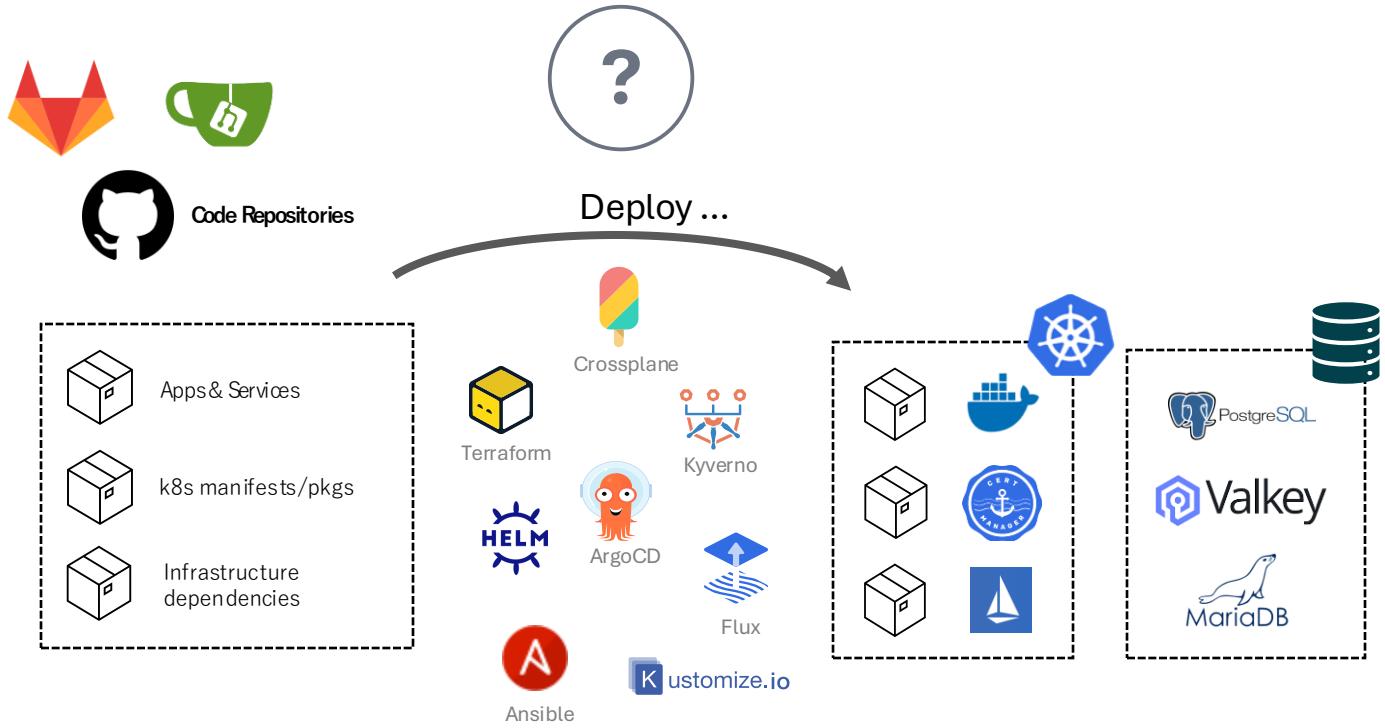
03
Stay inside the boundary
Artifacts, tooling, telemetry, licensing, and dependencies resolve in-region or turn off.

THE OPERATING TEST

Can an operator deploy, upgrade, validate, and recover the product using only shipped artifacts, automation, configuration, and release-specific documentation?

THE PROBLEM

multi-source, multi-target landscapes



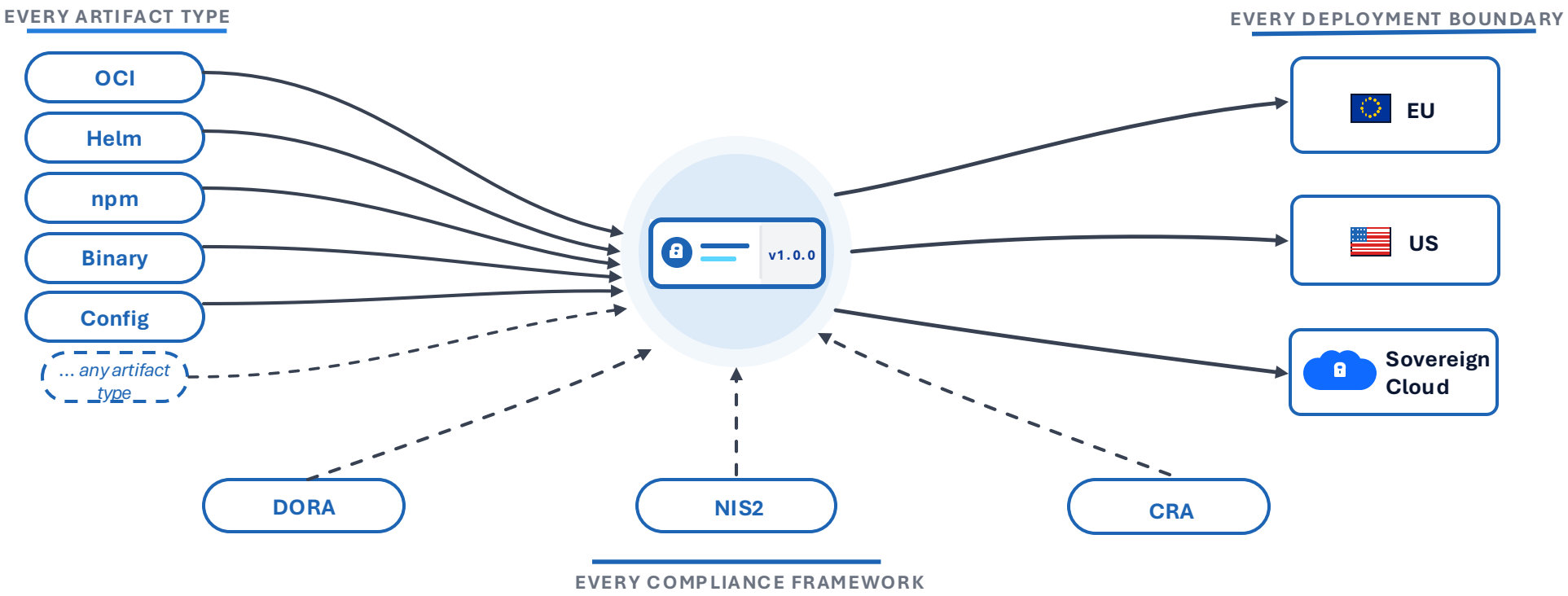
... each with its own config, format, and pipeline

To run your App, we need to deploy our service stack and all referenced components:

- **Infrastructure resources & dependencies**
  e.g. databases, entitlements, service instances…
- **Applications**
  e.g. Docker images, native packages…
- **Orchestration tooling**
  e.g. Day-2 Kubernetes operators, …
- **Foundational Dependencies**
  e.g. Bare Metal provisioning

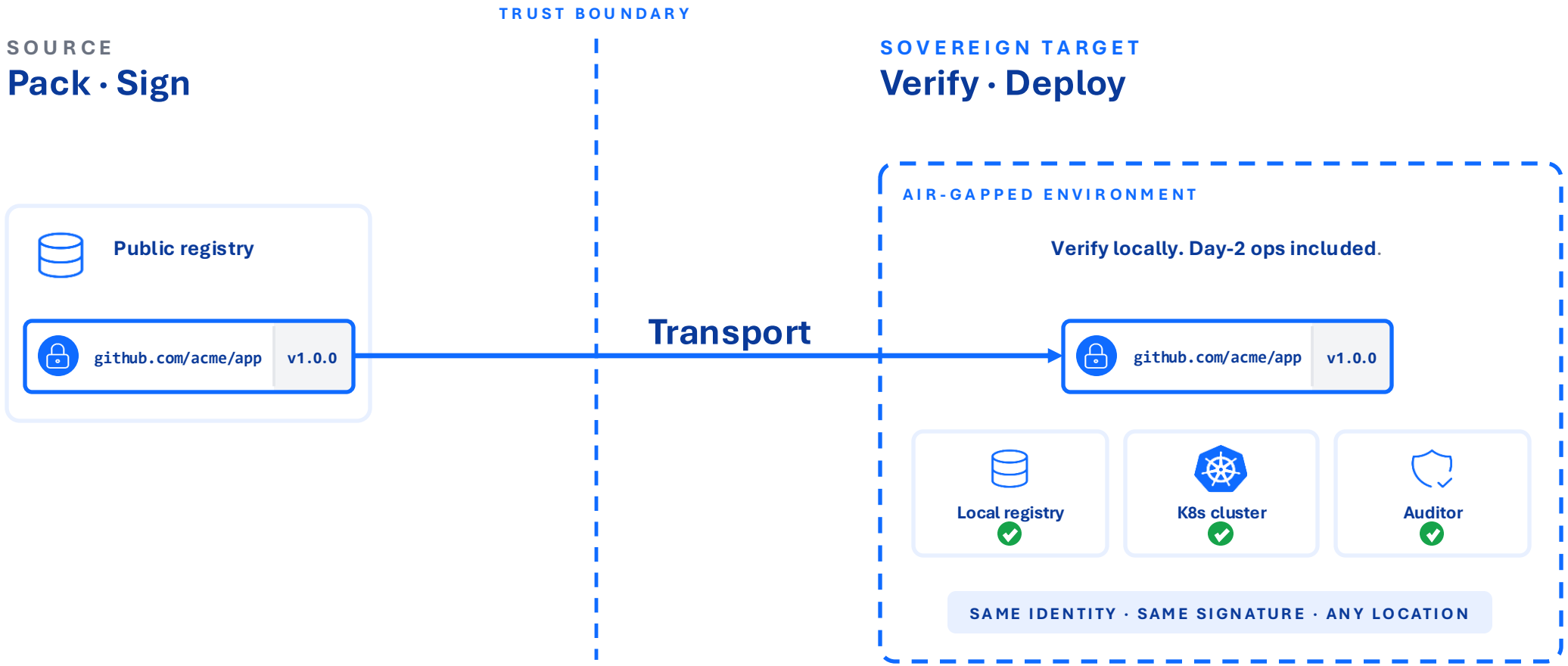
THE ANSWER

Meet OCM. One identity, every boundary.



SOVEREIGN-READY - AIR-GAP

Trust travels with the component.



BUILT IN THE OPEN

OCM and Open Source.

OCM IN NUMBERS

LFX Insights · past 365 days

86 active contributors +76% vs. prior year	**16** contributing orgs +129% vs. prior year	**3,751** pull requests 2,841 merged
698 issues closed +58% vs. prior year	**< 48 h** to first human review 2 days avg. to merge	**94** health score / 100 LFX: Excellent

WORKS WITH

Kubernetes · Flux · Argo CD · kro · Crossplane · Sigstore

Deployers, graph engines and signing documented on ocm.software

WHAT OUR TSC & LEADERS DO UPSTREAM

OCM Technical Steering Committee
Members from SAP, Kubermatic and BWI; Kubermatic chairs TSC and SIG Runtime

NeoNephos Foundation
Seat on the Technical Advisory Council · SAP, BWI Premier; Kubermatic General

kro · kubernetes-sigs
Maintainer and reviewer seats in the graph engine behind this demo

External Secrets Operator · CNCF Sandbox
Maintainer of the Kubernetes secrets operator

Kubernetes
Members of the kubernetes and kubernetes-sigs GitHub orgs

Open Defense Cloud · BWI
Builds its bundles on OCM; contributed ocm-kit to the OCM community

Sources: OCM docs/steering/OWNERS.md · kubernetes/org · kro OWNERS_ALIASES · ESO MAINTAINERS.md · neonephos.org TAC and members

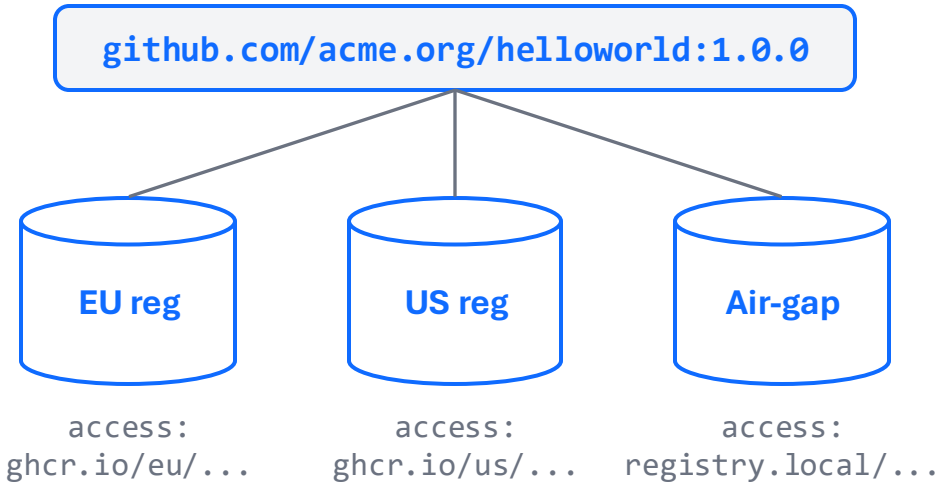
Open governance, upstream stewardship, and a stack that plugs into what you run.

OPEN COMPONENT MODEL · OCM.SOFTWARE

THE HINGE

Identity that travels with the release.

- **Component identity:** name and version of the component. Globally unique. Location-agnostic.
- **Digest:** every resource inside the component carries a content hash. Computed once.
- **Access:** where the resource currently lives. Rewritten on transfer. Digest stays.



Move the artifact. The digest stays. Only the access changes.

CONSTRUCTOR

What you write.





Software-Bill-of-Delivery (SBoD)

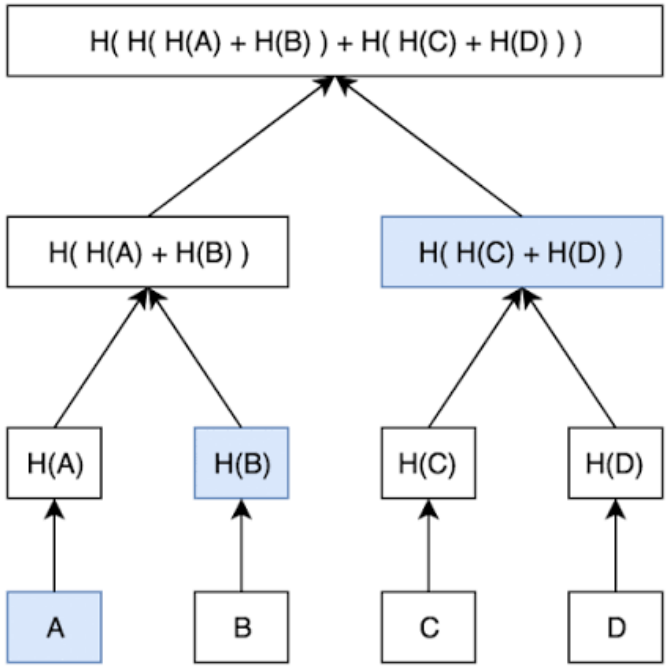
https://github.com/open-component-model/ocm-spec

```yaml
components:
- name: github.com/acme.org/helloworld
  version: 1.0.0
  provider:
    name: acme.org
  resources:
    - name: mylocalfile
      type: blob
      input:                    # Embed by value
        type: File/v1
        path: ./my-local-resource.txt

    - name: image
      type: ociImage
      version: 1.0.0
      access:                   # Reference external artifact
        type: OCIImage/v1
        imageReference: ghcr.io/stefanprodan/podinfo:6.9.1
```

DESCRIPTOR

Merkle proof over your entire infrastructure.



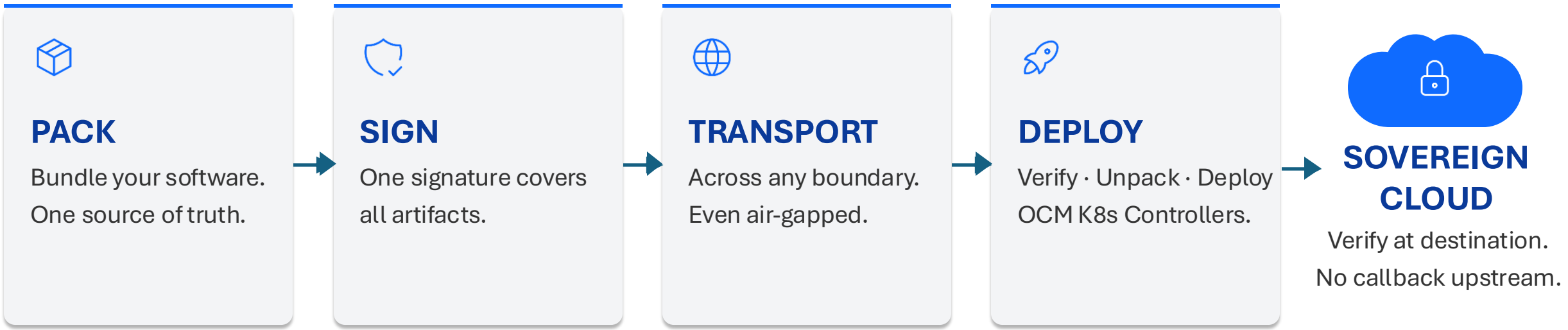

```yaml
component:                                  # (fields trimmed)
  name: github.com/acme.org/helloworld
  version: 1.0.0
  resources:
    - name: image
      type: ociImage
      access:                               # excluded - rewritten on transfer
        type: OCIImage/v1
        imageReference: ghcr.io/stefanprodan/podinfo@sha256:8fa5691d768ef456...

      digest:                               # content identity - input to descriptor hash
        hashAlgorithm: SHA-256
        value: 262578cde928d5c9eba3bce0...

signatures:                                 # signature: one hash over the canonicalized descriptor
  - name: acme-release-key
    digest:                                 # of the descriptor
      hashAlgorithm: SHA-256
      value: a4b1c2d3e4f5...
    signature:
      algorithm: RSASSA-PSS
      value: <hex-encoded signature>
```

OCM IN ONE PICTURE

Pack · Sign · Transport · Deploy



DEMO · SOVEREIGN SCENARIO

One product. Three components.

PRODUCT `acme.org/sovereign/product`

product-rgd
kro ResourceGraphDefinition, shipped as a resource

componentReferences
notes · postgres, pinned by digest

NOTES `acme.org/sovereign/notes`

Container image
built from Go source, embedded by value

Helm chart
embedded by value

OpenAPI spec
API contract travels with the release

ORD document
Open Resource Discovery metadata

POSTGRES `acme.org/sovereign/postgres`

postgres:18 image
official image, by reference

Helm chart
StatefulSet, embedded by value

SOVEREIGN TARGET

Kind cluster with its own registry. No route back to the build environment.

Kubernetes 1.34
Kind cluster

registry:5000
air-gapped OCI registry

OCM controllers
resolve · verify · deliver

kro 0.9.4
ResourceGraphDefinitions

Flux
source + helm controller

The same scenario gates every OCM build in CI: first run and upgrade.

DEMO 1 · FIRST RUN

Pack and sign on the workstation.

1 build:notes

docker buildx → OCI layout → embedded by value

`input: file · oci.layout.v1+tar+gzip`

2 build:postgres

Official image referenced, digest pinned at pack time

`access: ociArtifact · postgres:18`

3 build:product

References both services and ships the kro graph

`componentReferences: notes, postgres`

4 sign

RSA 4096 key, RSASSA-PSS over the product descriptor

`ocm sign cv ctf//acme.org/sovereign/product`

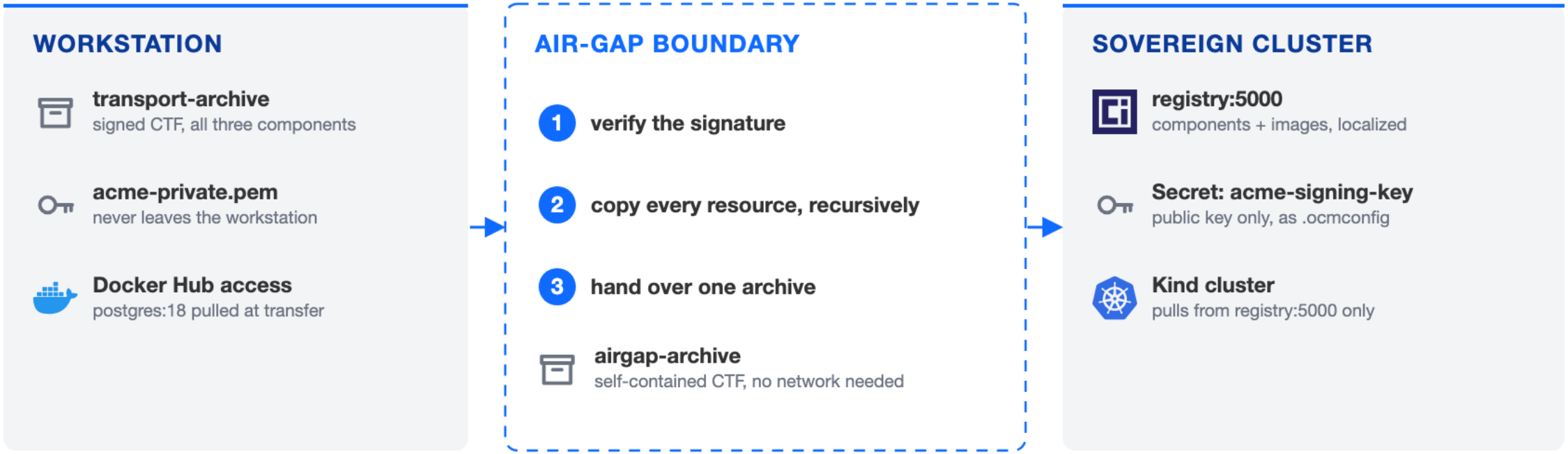
components/product/component-constructor.yaml

```yaml
components:
  - name: acme.org/sovereign/product
    version: ${VERSION}
    componentReferences:
      - name: notes
        componentName: acme.org/sovereign/notes
        version: ${VERSION}
      - name: postgres
        componentName: acme.org/sovereign/postgres
        version: ${VERSION}
    resources:
      - name: product-rgd
        type: blob
        input:            # embedded by value
          type: file
          path: ./deploy/rgd.yaml
          mediaType: application/vnd.cncf.kro
            .resourcegraphdefinition.v1+yaml
```

By value or by reference. One signature over the whole product.

DEMO 1 · FIRST RUN

Carry it across the air gap.




```
task transfer:airgap · task cluster:import
$ ocm verify cv    tmp/transport-archive//acme.org/sovereign/product:1.0.0
$ ocm transfer cv  --copy-resources --recursive  tmp/transport-archive//…product:1.0.0  tmp/airgap-archive
$ ocm transfer cv  --copy-resources --recursive  tmp/airgap-archive//…product:1.0.0  http://registry:5000
```

Only the public key enters the cluster. Access is rewritten; digests stay.

DEMO 1 · FIRST RUN

Bootstrap once. Then it runs itself.

PHASE 1 · BOOTSTRAP.YAML

Repository
sovereign-repo → registry:5000

↓

Component
resolves 1.0.0, verifies signature

↓

Resource
extracts product-rgd

↓

NamespacedDeployer
applies it as a bounded service account

ResourceGraphDefinition installed ✓

PHASE 2 · ONE SOVEREIGNPRODUCT

```yaml
apiVersion: kro.run/v1alpha1
kind: SovereignProduct
spec:
  version: "1.0.0"
  notes:
    replicas: 2
```

→

kro

- registers the SovereignProduct CRD
- gates every step with readyWhen
- wires OCM status into Flux via CEL

↓

SELF-MANAGED	APP CONFIG	POSTGRES	NOTES
Namespace	ConfigMap	Resource (chart)	Resource (chart)
Repository	Secret	Resource (image)	Resource (image)
Component		OCIRepository	OCIRepository
Resource (RGD)		HelmRelease	HelmRelease
NamespacedDeployer			

One custom resource in. Fifteen resources out, in dependency order.

OPEN COMPONENT MODEL · OCM.SOFTWARE

DEMO 2 · UPGRADE

Change one field.

```yaml
sample-product-1.0.0.yaml
kind: SovereignProduct
spec:
  version: "1.0.0"
  notes:
    replicas: 2
```

task upgrade

```yaml
sample-product-1.1.0.yaml
kind: SovereignProduct
spec:
  version: "1.1.0"
  notes:
    replicas: 2
```

GO notes 1.1.0

Incremental migrations evolve the 1.0.0 schema. Every note gains a title; existing data stays.

1 PACK & SIGN

notes, postgres and product rebuilt at 1.1.0 and signed

2 AIR GAP

verify, copy to a fresh air-gap CTF, import to registry:5000

3 VERIFY

Component resolves 1.1.0 and checks the RSA signature in-cluster

4 RE-GRAPH

kro applies the RGD shipped in 1.1.0: the pipeline updates itself

5 ROLL

Resources resolve new digests; Flux rolls notes with the migration

```
$ task upgrade      # build:product · transfer:airgap · cluster:import at 1.1.0, then apply the new spec
$ kubectl wait --for=condition=Ready -n sovereign-product sovereignproduct/sovereign-product
```

One field changes. Signature, digests and the delivery graph follow.

ADOPTER PROOF

Open Source by design, used from small-scale to enterprise.

Part of NeoNephos Foundation















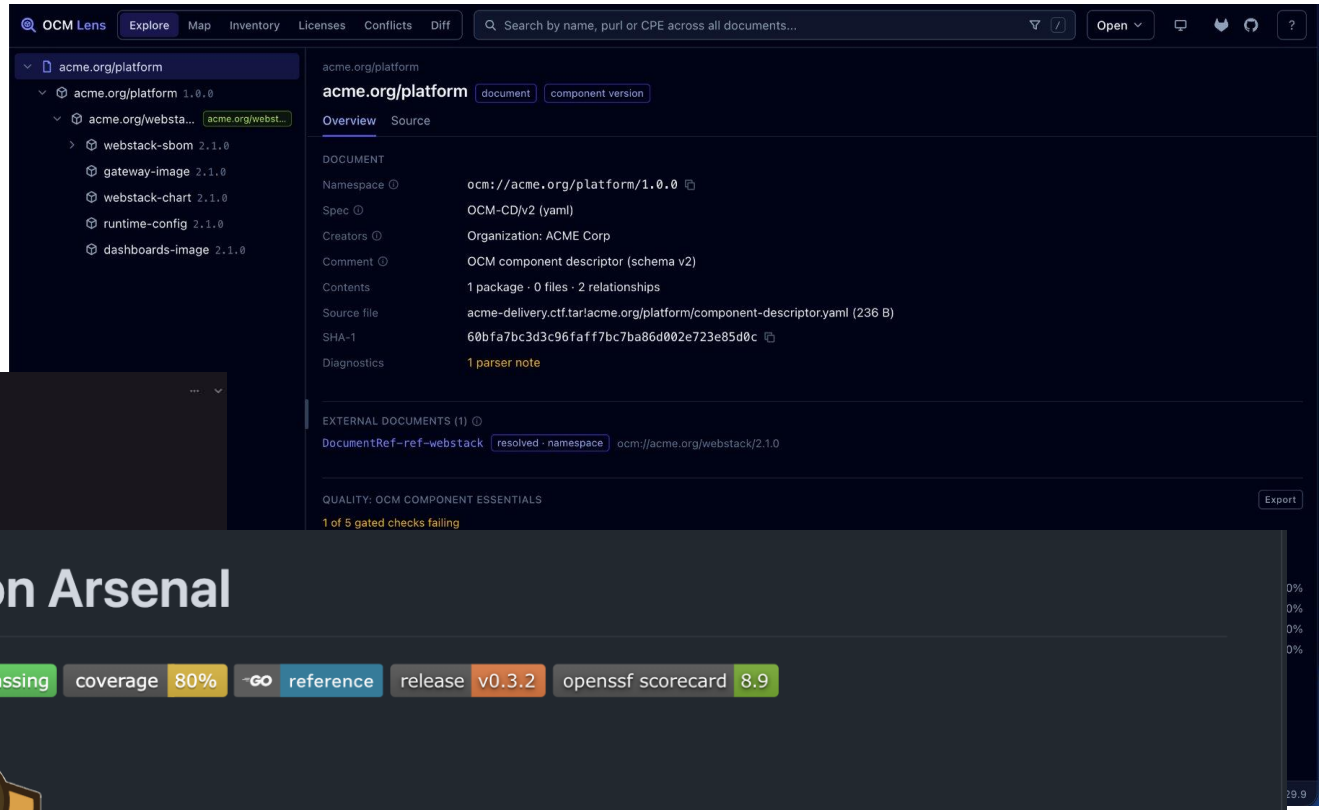


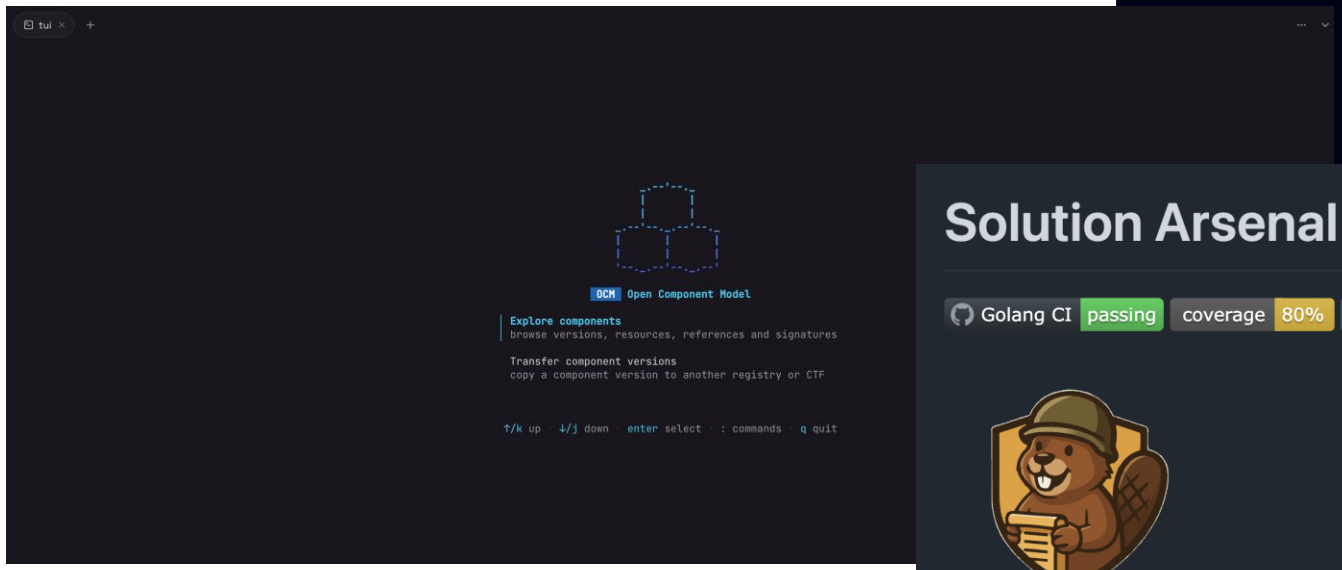






ADOPTER PROOF





Solution Arsenal

Golang CI passing | coverage 80% | reference | release v0.3.2 | openssf scorecard 8.9

Solution Arsenal (SolAr) is an application catalog based on Open Component Model packages (ocm.software) and fleet rollout management for these solutions onto Kubernetes Clusters. It features a catalog of solutions, which are application bundles provided as OCM packages from an OCI compliant registry. Additionally Kubernetes clusters can be registered with SolAr to turn them into deployment targets for the solutions from the catalog. The deployment itself then uses OCM Controllers with fluxCD as a deployer (https://ocm.software/docs/concepts/kubernetes-controllers/).

OCM TECHNOLOGY LANDSCAPE

Supported technologies today.

ARTIFACTS & PACKAGES

- OCI artifacts
- Container images
- Helm charts
- YAML · JSON
- Files · dirs
- Local blobs

Inputs: File · Dir · UTF8 · Helm
Access: OCIImage · OCIImageLayer · LocalBlob

SOURCE CODE

- Any Git server
- GitHub
- GitLab
- Bitbucket
- Gitea
- Azure DevOps

Git/v1 via HTTPS, SSH, git:// or local path
GitHub/v1 kept for published components

HTTP & OBJECT STORAGE

- HTTP · HTTPS
- Amazon S3
- MinIO
- Ceph RGW
- Cloudflare R2

Wget/v1 and S3/v2 as input or access
Digest verified on every fetch

TRANSFER TARGETS

- OCI registries
- CTF archives
- Artifactory
- Nexus
- Maven
- npm

OCI, Artifactory and Nexus uploaders
CTF archives carry releases into air gaps

DEPLOY & EXTEND

- Kubernetes
- OCM controllers
- Flux
- Argo CD
- kro
- Plugins

Repository · Component · Resource · Deployer
Replication · Discovery · plugin access types

SIGNING & TRUST

- Sigstore
- cosign
- Fulcio
- Rekor
- OIDC identity
- RSA-PSS keys

Keyless via OIDC or key-based signing
One signature covers every digest

Explore OCM yourself

https://ocm.software

jakob.moeller@sap.com

Pilot - Pack one product as an OCM component

Standardize - Make OCM the default component delivery

Steward - Bring yourself into the OCM community





NeoNephos Foundation