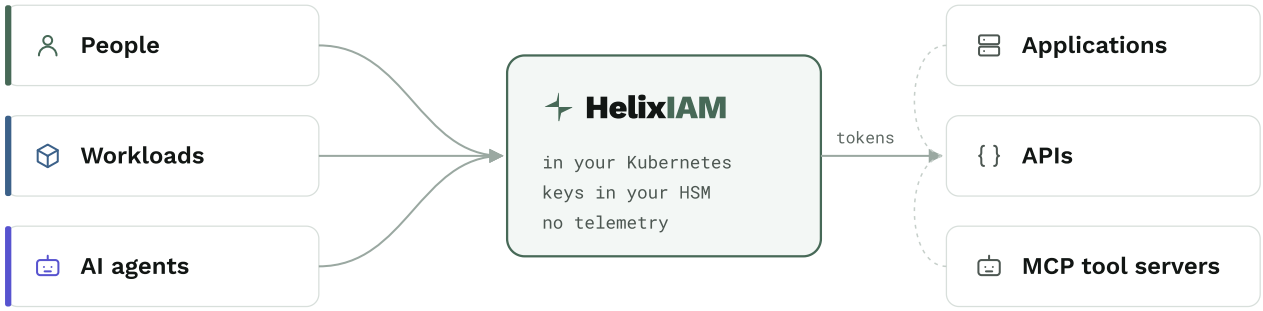
HelixIAM

WHITE PAPER · NOVEMBER 2026

# Identity infrastructure you control — for people, workloads and AI agents

HelixIAM reference architecture and platform overview

● Early access — now onboarding launch partners




**KubeDNA B.V.**
Enschede, the Netherlands

helixiam.com
contact@helixiam.com

EXECUTIVE SUMMARY

# One identity platform for people, workloads and AI agents — on infrastructure and keys you control.

Your systems now act for people, not just with them. A single request can pass through an employee, an AI agent acting for that employee, and a chain of workloads. Each hop needs an identity you can trace, limit and revoke.

HelixIAM is self-hosted identity and access management from KubeDNA B.V., a Dutch company. It runs in your own Kubernetes cluster with PostgreSQL, keeps signing keys in your HSM or in HashiCorp Vault or OpenBao, and sends no telemetry. You decide where data lives. KubeDNA offers deployment and operations support for HelixIAM on your own or a European cloud, with the scope set per contract.

### People

A standards-based OpenID Connect and OAuth 2 provider and a SAML 2.0 identity provider, with single sign-on and single logout, passkeys, MFA on by default, LDAP / Active Directory and SCIM 2.0.

### AI agents

Every agent gets its own identity with an owner and a lifecycle. Agents act on behalf of users through standard token exchange (RFC 8693), with authority that only ever narrows. Suspend an agent and it gets no new tokens.

### Workloads

Keyless workload identity for Kubernetes and CI: pods and pipelines exchange their platform token for a short-lived HelixIAM token, with no long-lived secrets.

### Your keys

Per-tenant signing keys with safe rotation. Keep them in your HSM over PKCS#11, or in Vault or OpenBao Transit — wrapped by your key-encryption key, or generated in your backend and never leaving it.

### European identity

DigiD, eHerkenning and eIDAS connectors included, for use with your own agreements. `Early access` Sign in with the EU Digital Identity Wallet, tested with the EU reference wallet.

### Built to operate

A Helm chart with high availability built in. PostgreSQL, with Redis optional; no message broker. Every release is cosign-signed with an attested SBOM.

PROPOSITION · HELIXIAM AI IDENTITY

### Know which agent did what, for whom — and stop it.

Agent and workload identity, often deployed alongside the workforce identity provider you already have. Sovereign by design: it runs on your infrastructure, under your keys.

PROPOSITION · HELIXIAM SOVEREIGN

### Enterprise IAM on infrastructure and keys you control.

The primary identity provider for employees, customers or citizens: SSO, MFA, federation, multi-tenant realms, customer-held keys and European eID connectors.

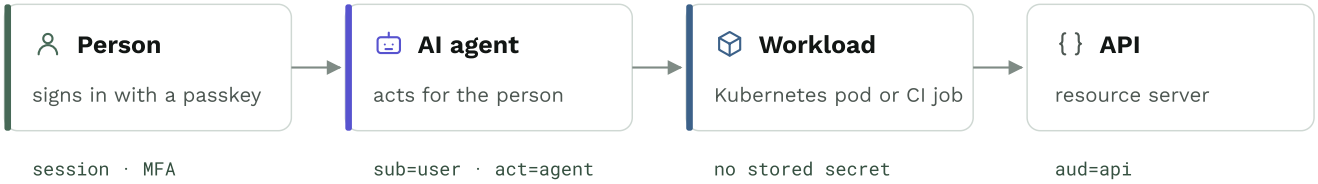
### Where HelixIAM stands

HelixIAM is released, documented and in early access: we are now onboarding launch partners. Organisations start with a technical demo on a running system, define their use case with us, and continue with a paid pilot on their own infrastructure — working directly with the engineers who build HelixIAM. HelixIAM is commercial software from KubeDNA B.V., available under a commercial licence.

01 THE IDENTITY CHALLENGE

# One request, many hops, three kinds of identity.

Identity used to mean people signing in to applications. Today a single business transaction can cross a person, an AI agent working on their behalf, and several workloads before it reaches the API that does the work.




Every hop carries its own token: a session for the person, a delegated token that names both user and agent, a short-lived workload token without a stored secret, and an audience-restricted token at the API.

### People

Employees, customers and citizens need single sign-on, MFA and federation across every application — web, SaaS and the systems you already run — and, in Europe, sign-in with national eIDs.

### AI agents

Assistants and MCP clients need an accountable owner, a clear principal they act for, and limits. Too often they run on a shared API key or on a person's own token.

### Workloads

Pods and CI pipelines need identity without long-lived secrets in configuration, pipelines and images — secrets that have to be stored, rotated and protected.

## The questions a security review asks

**Who did this?**
Every action should trace back to a named identity: a person, a workload or a specific agent — not a shared credential.

**For whom, and with what authority?**
An agent acting for an employee should never hold more than that employee, and usually much less.

**How do we stop one agent?**
Without breaking the others, and without rotating secrets across pipelines.

**Where do the signing keys live, and who can export them?**
Whoever controls the keys that sign your tokens controls who can act in your systems.

Answering these questions with separate products — one for workforce sign-in, another for machine credentials, another for agents — splits the audit trail, the policy and the keys. HelixIAM answers them with one platform: people, agents and workloads share one issuer, one audit stream and one key policy.

02 THE PLATFORM

# Two propositions on one identity core.

HelixIAM is one product. HelixIAM AI Identity and HelixIAM Sovereign are two entry points into it, for two different buying motivations. Both run the same server: one product, one licence.

| | HELIXIAM AI IDENTITY | HELIXIAM SOVEREIGN |
|---|---|---|
| **The need** | “We are deploying agents and can't answer who did this, for whom, with what authority.” | “Our IAM must run on infrastructure and keys we control, with European eIDs.” |
| **What it covers** | Agent identities with owner and lifecycle; on-behalf-of delegation that only narrows (RFC 8693); suspension; MCP protected-resource metadata (RFC 9728); keyless workload identity; DPoP and audience-restricted tokens. | OpenID Connect, OAuth 2 and SAML 2.0; SSO and single logout; MFA and passkeys; LDAP/AD; brokering; SCIM 2.0; multi-tenant realms; HSM and Vault keys; DigiD, eHerkenning and eIDAS connectors; audit events for your tool; GDPR tooling. |
| **Typical deployment** | Alongside your existing workforce IdP: people federate in through brokering, HelixIAM governs agents and workloads. | Primary identity provider for employees, customers or citizens; replaces or complements an existing IAM. |
| **Typical owner** | Platform engineering, AI platform teams, IAM architects. | CISO, head of IAM, enterprise architects, public-sector information security. |

The propositions reinforce each other. A team that starts with agents can later make the same installation its SSO provider for OIDC and SAML applications, without re-procurement. A Sovereign customer that adds AI agents already has the agent identity, delegation and suspension it needs — under the same keys and in the same audit stream.

## Three pillars

SOVEREIGN CORE

### Human identity & enterprise IAM

Single sign-on, MFA and federation for employees, customers and citizens, multi-tenant from day one.

- OIDC / OAuth 2 provider and SAML 2.0 IdP
- Passkeys, authenticator-app MFA, adaptive step-up
- LDAP / AD, brokering, SCIM 2.0
- Realms per tenant, organisations for B2B

AI IDENTITY CORE

### Non-human & AI-agent security

Every agent gets an owner, a lifecycle and delegated authority that only narrows.

- Agent identities with owner review
- Token exchange (RFC 8693), sub + act
- Suspend: no new tokens
- Ready for MCP (RFC 9728)

AI IDENTITY + SOVEREIGN

### Workload identity & cloud-native security

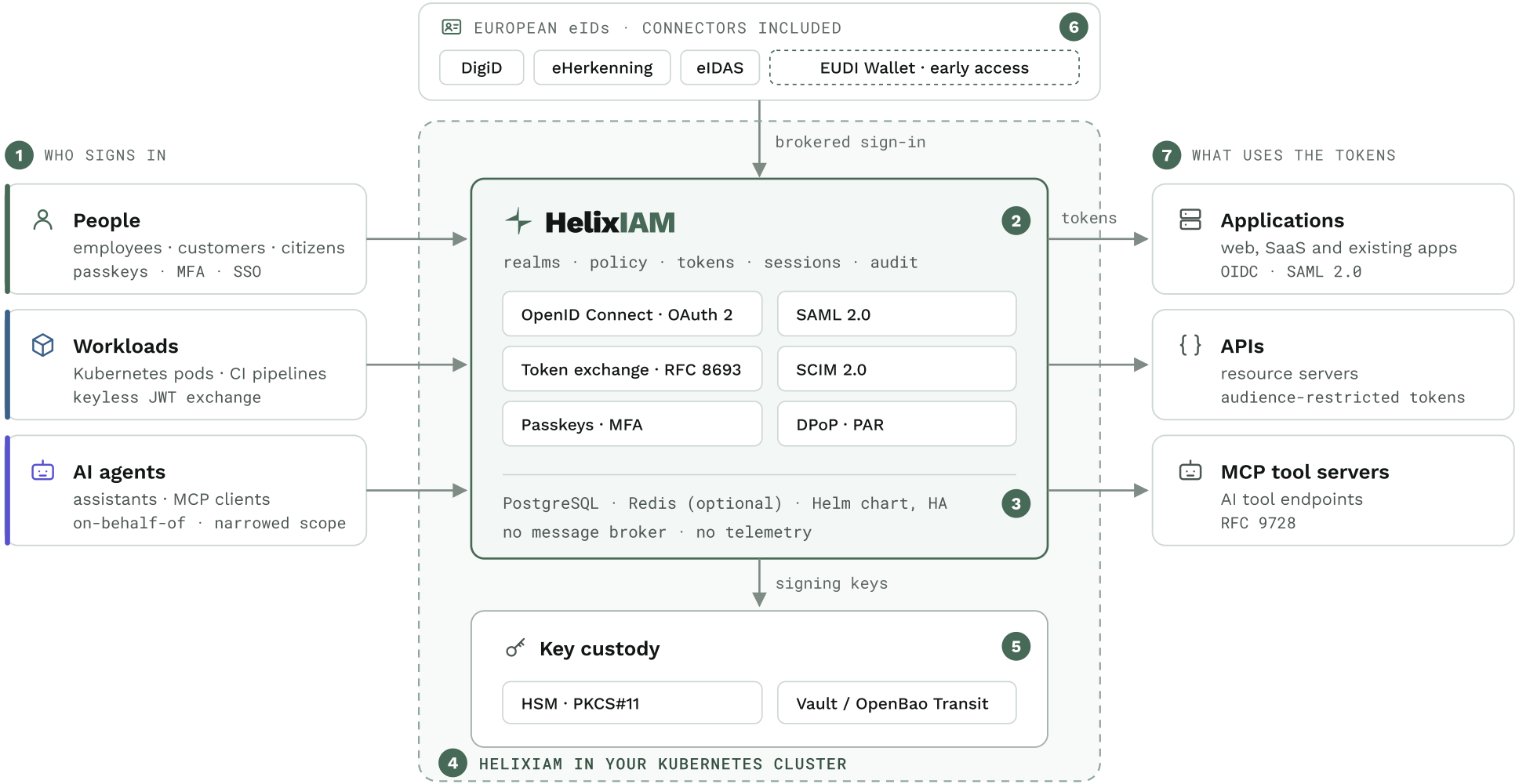
Keyless identity for Kubernetes and CI, sender-constrained tokens, Kubernetes-native operations.

- Kubernetes and CI JWT exchange
- DPoP, resource indicators, PAR
- Helm chart, HA built in
- Prometheus metrics and alert rules

03 REFERENCE ARCHITECTURE

# HelixIAM in your own infrastructure.

HelixIAM runs in your Kubernetes cluster with PostgreSQL. Signing keys stay in your HSM or in Vault / OpenBao. European eID connectors are included; you connect DigiD or eHerkenning under your own agreement. Your applications, APIs and agents can run in the same infrastructure. Numbers refer to the walkthrough on the following pages.



# The components, one by one.

Each number matches a marker on the reference architecture diagram.

### 1 People, workloads and AI agents

Three kinds of caller authenticate to the same issuer, each in the way that fits it.

- **People** — employees, customers and citizens — sign in with passkeys (WebAuthn/FIDO2), passwords plus authenticator-app MFA, or an optional magic link, and get single sign-on across OIDC and SAML applications. Push approval with number matching via the HelixIAM Authenticator app is available as `Early access`.
- **Workloads** — Kubernetes pods and CI pipelines — present the token their platform already gives them and exchange it for a short-lived HelixIAM token. No long-lived secrets.
- **AI agents** — assistants and MCP clients — are registered principals with an owner. When they act for a user, they use standard token exchange to obtain a token that names both, with narrowed authority.

### 2 The HelixIAM server and admin console

HelixIAM ships as two container images that are deployed and scaled independently:

- the **server**: the protocol endpoints (OpenID Connect, OAuth 2, SAML 2.0, SCIM 2.0, token exchange), the sign-in pages, the admin REST API and the end-user account console;
- the **admin console**: the operator interface for realms, applications, users, roles, sessions, keys, audit settings and the agent inventory. It can be kept internal-only.

The server is stateless: sessions, authorization codes, tokens, SSO sessions and flow state live in shared storage, so any replica can answer any request. Each tenant is a **realm** with its own users, keys, sign-in flows and branding. Everything in the console is available through a documented admin REST API.

### 3 PostgreSQL and optional Redis

HelixIAM runs on PostgreSQL, with Redis optional; there is no message broker. You can run with PostgreSQL only, or add Redis for sessions and tokens at higher throughput. Multi-step flows — such as push approval, QR sign-in and device enrolment — and security rate limits are shared across replicas either way.

### 4 Your Kubernetes cluster, deployed with Helm

A Helm chart with high availability built in: two server replicas by default, a PodDisruptionBudget, topology spread and NetworkPolicy. The console is a separate Deployment with its own replica count, Service, Ingress host and NetworkPolicy, so scaling the server never scales the console.

- Containers run as non-root, with a read-only root filesystem and all capabilities dropped.
- The admin REST API and the master realm can be restricted to an internal admin host; on the public sign-in host they answer 404. Tenant administrators can sign in to their own realm on a tenant console.
- Prometheus metrics, a ServiceMonitor and alert rules come with the chart.

### 5 Key custody: your HSM, Vault or OpenBao

Each realm has its own signing keys with safe rotation, and you can bring your own CA-signed keys. A realm can keep its keys in a backend you run: an HSM over PKCS#11, or HashiCorp Vault or OpenBao Transit. Three custody profiles let you choose, per realm, how far the keys stay inside your backend — see section 05.

### 6 European eID connectors

DigiD, eHerkenning and eIDAS connectors are included. You connect them under your own Logius or broker agreement, and each connection has a minimum level of assurance that is both requested and enforced. `Early access` Sign-in with the EU Digital Identity Wallet over OpenID4VP, tested with the EU reference wallet.

### 7 Applications, APIs and MCP tool servers

- **Applications** — web, SaaS and existing apps — use OpenID Connect or SAML 2.0, with single sign-on and single logout.
- **APIs** receive audience-restricted tokens (resource indicators) that can be sender-constrained with DPoP, so a token minted for one API cannot simply be replayed against another.
- **MCP tool servers** publish OAuth protected-resource metadata (RFC 9728) and accept tokens bound to the one server they were minted for.

## Deployment notes

| TOPIC | WHAT TO PLAN |
|---|---|
| **Ingress layout** | A public sign-in host for people, apps, agents and workloads; an internal admin host for operators; optionally tenant consoles for per-tenant administrators. |
| **Replicas and state** | Two server replicas for high availability, more for larger tenant counts. Decide on PostgreSQL topology and backups, and whether to add Redis. |
| **Keys** | A custody profile per realm, and the HSM, Vault or OpenBao set-up to connect. |
| **Observability** | Prometheus scraping (authenticated by default) and alert routing; audit events as JSON lines on standard output; your log collector routes them to your SIEM. |
| **Configuration as code** | Realm export and import; the documented admin REST API; a Terraform provider for applications, roles and wallet connections. |
| **Images and evaluation** | linux/amd64 images, cosign-signed with an attested SBOM, from a private registry. Evaluate in minutes with Docker Compose (with an evaluation licence); run production on Helm. |

04 AI-AGENT IDENTITY

# AI-agent identity, sovereign by design.

Every agent gets an identity, an owner and limits. You can always answer which agent did what, for whom and with what authority — and stop it.

### Ownership

Each agent is registered with an accountable owner, a purpose, its own roles and an expiry date. Its tokens carry a non-human marker and the agent's id, owner and purpose.

### Lifecycle

Active, suspended, expired, revoked. An owner review flags agents whose owner has left; deprovisioning a user suspends the agents they own. Every change is audited.

### Suspension

Suspend an agent from the console or the API and it gets no new tokens; resource servers that introspect see the revocation at once. Keep agent token lifetimes short.

## Delegation with RFC 8693: authority only ever narrows

When an agent acts for a user, it calls the standard token endpoint with OAuth token exchange (RFC 8693), presenting the user's token as the subject and its own as the actor. In the issued token the user stays the subject, the agent is named as the actor (nested for chains of agents), and the roles are the **intersection** of what the user and the agent may do. Any OAuth-aware resource server can read it. For AI tool servers, HelixIAM is ready for MCP: OAuth protected-resource metadata (RFC 9728), with tokens bound to the one server they were minted for.

token exchange · RFC 8693 — 200 OK

REQUEST · AGENT ACTING FOR ALICE

```
POST /realms/acme/oauth2/token
grant_type=urn:ietf:params:oauth:
           grant-type:token-exchange
subject_token=eyJ…   # alice (the user)
actor_token=eyJ…     # agent: invoice-assistant
scope=invoices:read
```

ISSUED ACCESS TOKEN · DECODED

```
{
  "iss": "https://id.acme.example/realms/acme",
  "sub": "alice",                                          (1)
  "act": { "sub": "agent:invoice-assistant" },             (2)
  "realm_access": { "roles": ["invoice-reader"] },         (3)
  "scope": "invoices:read",
  "aud": "https://api.acme.example/invoices",              (4)
  "exp": 1798732800
}
```

1. **sub** — the user stays the subject, so downstream systems still see who the work is for.
2. **act** — the agent is named in every token it uses; each call traces back to a specific agent and its owner.
3. **roles** — the intersection of alice's and the agent's roles. If alice may also approve invoices but the agent may only read them, the token can only read.
4. **aud** — restricted to one API. Suspend the agent and it gets no new tokens.

Illustrative values; the realm, user, agent and API are examples.

### A note on cloud-only alternatives

Agent identity is also offered by large vendors as cloud services operated by US-headquartered companies — for example Microsoft Entra Agent ID, and Okta and Auth0 for AI Agents. HelixIAM is a different choice: agent identities, ownership, delegation and suspension run on your own infrastructure, tokens are signed with keys in your own HSM or Vault, and the software comes from a Dutch company, KubeDNA B.V. It also runs next to the identity provider you have: your people federate in through brokering, while HelixIAM governs agents and workloads.

05 SOVEREIGNTY AND KEY CUSTODY

# Your infrastructure. Your keys.

HelixIAM is self-hosted on your own infrastructure: you decide where data lives, and there is no telemetry — the software sends nothing to KubeDNA. The keys that sign your tokens can stay in a backend you run.

## Three custody profiles, chosen per realm

| PROFILE | WHERE PRIVATE KEYS LIVE | WHERE SIGNING HAPPENS |
|---|---|---|
| **Standard** `default` | Generated in the server, stored encrypted in the database. | In the server. |
| **Protected** | Stored in the database only wrapped by the realm's key-encryption key (KEK), which lives in your HSM or Vault. | In the server, after the pod unwraps the key into memory. |
| **High assurance** | Generated inside your HSM or Vault. They never leave it. | Every signature is a call to your backend. |

### Backends

An HSM over PKCS#11; HashiCorp Vault or OpenBao Transit. Custody covers realm token-signing keys, SAML keys, EU wallet verifier keys, and credential-issuer and eID keys.

### Algorithms and rotation

RS256 and PS256 per realm, with ES256 and EdDSA options; per-realm rotation; bring your own CA-signed keys.

### Fails closed

A request that needs an unavailable backend is never signed with a fallback key: it gets a `503` with `Retry-After`. Each pod applies a time bound per call, load shedding and a circuit breaker per backend. Discovery, JWKS and every realm that does not depend on that backend keep working.

### Backend secrets

PINs, tokens and secret ids are read from mounted files, and never stored, logged or shown in the console.

## Tenant isolation

Multi-tenant by design: each tenant gets its own realm, users, keys and branding, with fine-grained admin permissions and per-tenant administrators. A key reference resolves only in its own realm. That suits shared-service organisations and providers: one installation, a realm per entity, brand or municipality, and organisations within a realm for B2B membership, branding and claims.

### Key custody beyond signing keys

- **A dedicated key store per tenant:** bind a dedicated HSM partition or Vault credential to a tenant's realm, so that tenant isolation also holds inside the key store.
- **Online KEK rotation:** rotate a Protected realm's key-encryption key online, while tokens keep being signed.
- **Wider custody:** credential-issuer and eID keys in your HSM or Vault, and optional signed SAML metadata.

## What sovereignty means in practice

- **Location and call-home:** HelixIAM runs where you run it — your data centre, your private cloud or a European cloud — and contacts only the external services you configure yourself.
- **Keys:** realm signing keys can be generated in, and never leave, a backend under your control.
- **Vendor:** commercial software from KubeDNA B.V., a company registered in the Netherlands, which can also deploy and operate HelixIAM with you.

06 ENTERPRISE IAM, WORKLOAD IDENTITY AND EUROPEAN EID

# The enterprise IAM you expect, with standards to match.

Under both propositions sits a complete identity provider. It speaks the standards your applications already use, so you can introduce it application by application.

### Sign-in

- Passkeys (WebAuthn/FIDO2)
- Authenticator-app MFA, on by default, with recovery codes
- Optional magic-link sign-in
- Adaptive step-up authentication
- Argon2id password hashing, lockout and breached-password checks
- Push approval with number matching `Early access`

### Federation & SSO

- OpenID Connect and OAuth 2 provider
- SAML 2.0 identity provider
- Single sign-on and single logout across OIDC and SAML apps
- Connect your LDAP or Active Directory
- Broker sign-in from OIDC, SAML and social providers
- SCIM 2.0, inbound and outbound

### Tokens, APIs & admin

- DPoP, PAR, resource indicators
- High-security profiles: mTLS-bound tokens, PS256
- Device-code sign-in for TVs, CLIs and IoT
- Dynamic client registration with admin-controlled policy
- Documented admin REST API; Terraform for applications, roles and wallet connections

## Workload identity: no secrets in pipelines and pods

Kubernetes service-account tokens and CI OIDC tokens are exchanged for short-lived HelixIAM tokens. HelixIAM checks that the issuer is trusted and that subject and audience match; otherwise the request is refused. The resulting token is short-lived and bound to the roles of one application.

Nothing is left to store or rotate in the pipeline. Because workloads use the same issuer as people and agents, they share one audit stream and one key policy.

workload identity · JWT exchange — 200 OK

REQUEST · CI JOB PRESENTS ITS PLATFORM TOKEN

```
POST /realms/acme/workload-identity/token
grant_type=urn:ietf:params:oauth:
           grant-type:token-exchange
subject_token=eyJ… # signed by the CI platform
```

ISSUED TOKEN · DECODED

```
{
  "sub": "service-account-deploy-pipeline",
  "realm_access": { "roles": ["deployer"] },
  "exp": 1798732800,
  "iss": "https://id.acme.example/realms/acme"
}
```

## European identity connectors included

| SCHEME | WHAT HELIXIAM PROVIDES |
|---|---|
| **DigiD** | DigiD connector included; you connect with your own Logius agreement. A minimum level of assurance per connection, both requested and enforced. |
| **eHerkenning** | eHerkenning connector included, for use with your own broker agreement. |
| **eIDAS** | eIDAS cross-border sign-in connector included. |

| SCHEME | WHAT HELIXIAM PROVIDES |
|---|---|
| **EU Digital Identity Wallet** Early access | Sign in with the EU Digital Identity Wallet over OpenID4VP (SD-JWT VC and mdoc; same- and cross-device), tested with the EU reference wallet. Verify age or attributes from the wallet without a full sign-in. Trust anchors from the official EU trusted lists, refreshed automatically. |
| **Credential issuance** Early access | Issue verifiable credentials to your users' EU wallets (OpenID4VCI). |

### Moving from your current IAM

Your applications speak OpenID Connect and SAML, and so does HelixIAM. Import clients, roles and identity providers from a Keycloak realm export, then validate and cut over application by application, with both identity providers running.

---

07 SECURITY ENGINEERING AND TRUST

## Security you can check for yourself.

An identity platform is part of your security perimeter. HelixIAM is built so that your own security team can verify what it runs, where it runs and what it signs.

#### { } Standards first

Standard protocols rather than proprietary ones: OpenID Connect, OAuth 2, SAML 2.0, SCIM 2.0, WebAuthn/FIDO2, RFC 8693, RFC 9449 (DPoP), RFC 9126 (PAR), RFC 8707, RFC 9728 and, in early access, OpenID4VP and OpenID4VCI. Your applications and resource servers rely on documented behaviour, not on a vendor SDK.

#### Security testing in CI

Static analysis, dependency and image scanning on every change: CodeQL for static analysis, Trivy for container and configuration scanning, Dependabot for dependencies; CI actions are pinned to commit SHAs. HelixIAM is built with security reviews and automated security testing in CI; an independent penetration test is in preparation.

#### Signed releases with SBOM

Every release is cosign-signed (keyless, Sigstore) with an attested SPDX SBOM. Images, the Helm chart and the compose file are delivered to licensed customers from a private registry, so you can verify exactly what you deploy.

#### Hardened by default

MFA on by default; Argon2id password hashing, lockout and breached-password checks; non-root containers with read-only root filesystems; NetworkPolicies in the chart; an admin surface that can be kept off the public host.

### Evidence for audit and privacy

- Audit events for sign-ins and admin changes, as JSON lines for your collector. Your audit tool stores them and sets the retention.
- Built-in GDPR data export and erasure per user, and factual input for your DPIA.

During an evaluation we work through your security questionnaire with you, walk your team through the architecture and the key-custody design, and support your own testing of your own installation.

08 DEPLOYMENT, SUPPORT AND HOW TO START

# Built and backed by KubeDNA.

HelixIAM comes from KubeDNA B.V., a Dutch company that builds Kubernetes infrastructure software. You get the identity software, and help with running it, from one accountable vendor.

### Deployment and operations support

KubeDNA offers deployment and operations support for HelixIAM on your own or a European cloud, with the scope set per contract. Run it yourself with the Helm chart, or let KubeDNA deploy and operate it with you.

### One vendor, one product

One accountable vendor for the server, the admin console, the Helm chart and operations help. Support terms and service levels are defined per contract.

## How to start

1

### Technical demo

A session on a running system: agents and delegation, workload identity, SSO and key custody — mapped onto your own deployment.

2

### Your use case

An architecture consultation on your requirements, infrastructure, key custody and eID needs, and success criteria agreed up front.

3

### Paid pilot

HelixIAM on your own infrastructure, on your own use case, working directly with the engineers who build it — ending in a plan for production.

**Early access — now onboarding launch partners**

HelixIAM is released and documented. Launch partners deploy HelixIAM on their own infrastructure, work directly with the engineering team and help shape the roadmap.

## Talk to us

Book a technical demo or request an architecture consultation.

E-mail **contact@helixiam.com**
Web **helixiam.com**
Docs **docs.helixiam.com**

**KubeDNA B.V.**
Tegelerweg 110
7535 CJ Enschede
the Netherlands
KvK 95511059

HelixIAM is commercial software from KubeDNA B.V., available under a commercial licence. This white paper describes HelixIAM as of November 2026. Features marked “Early access” are available to launch partners for evaluation. Token examples use illustrative values. Microsoft Entra, Okta, Auth0, HashiCorp Vault, Kubernetes and other names are trademarks of their respective owners and are used only to describe interoperability or the market. © 2026 KubeDNA B.V.