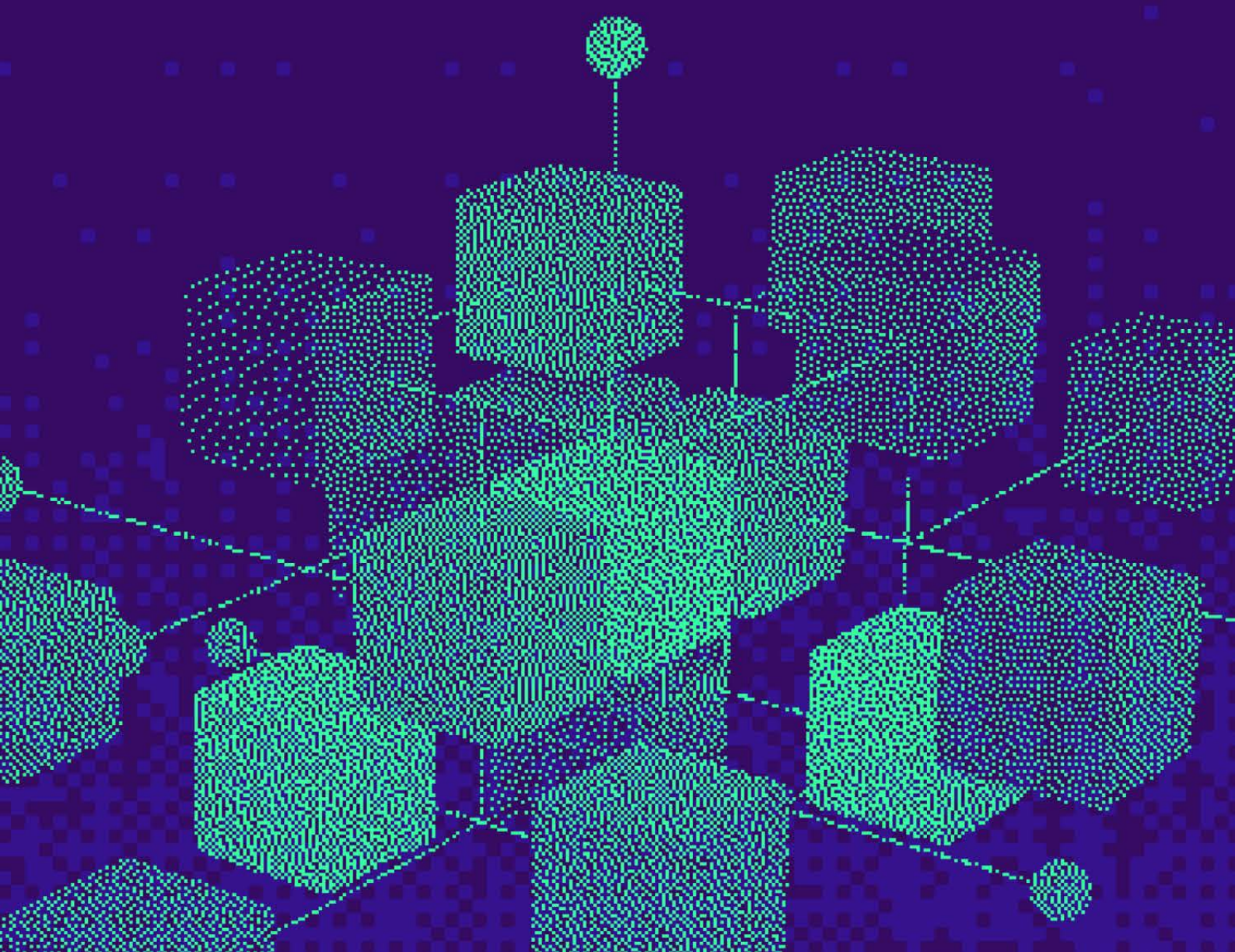
Weave Intelligence

Building the sovereign Internal Developer Platform (IDP)

Achieving strategic autonomy through an exit-by-design architecture

COMMISSIONED BY CYCLOID



Building the sovereign Internal Developer Platform (IDP)

Table of contents

About Cycloid

Cycloid's mission is to scale platform engineering initiatives. They provide the core building blocks for an enterprise Internal Developer Platform (IDP) that unifies the developer portal and platform orchestration, which can be completed with custom plugins. Cycloid gives platform engineering teams one control layer to stand up self-service golden paths in weeks - replacing the multi-year DIY builds typical of catalog-only tools.

Most vendors force a choice between a portal sitting on top of third-party automation or an orchestration engine without a developer-facing experience. Cycloid ships both, sharing one data model, one authorization engine, one API surface, and one audit trail.

Developers interact through a portal backed by reusable infrastructure templates (Stacks) and dynamic configuration forms (StackForms) that abstract provider complexity. The result: fewer tickets, lower cognitive load, and faster time from commit to production across hybrid, multi-cloud, and sovereign environments. Integrated FinOps and GreenOps modules - pre-deployment cost estimation, cloud cost management, carbon footprint tracking - give platform teams financial and environmental visibility without external tooling.

Architecturally, Cycloid is framework-agnostic and built for portability. It deploys as SaaS or fully self-hosted, treats IaC as the root of trust, and avoids proprietary lock-in to any single cloud provider - enabling the exit-by-design posture this whitepaper describes.

Cycloid serves regulated enterprises, public sector bodies, scale-ups, and works with global system integrators and managed service providers.

1

Executive summary

In 2026, digital sovereignty has transitioned from a theoretical policy discussion to a strict technical requirement for global enterprises. While the movement is spearheaded by European regulatory frameworks such as the NIS2 Directive, the Digital Operational Resilience Act (DORA), and the EU Data Act, the demand for strategic autonomy is now a recognized global phenomenon. Geopolitical shifts and legal conflicts - most notably the friction between the US Clarifying Lawful Overseas Use of Data Act (CLOUD Act) and regional privacy laws - have created a precarious environment for data and operational stability. Organizations are finding themselves trapped in a "legal sandwich," where conflicting jurisdictional mandates threaten their operational continuity.

This whitepaper, commissioned by **Cycloid** and authored by Weave Intelligence, presents a comprehensive reference architecture for a Sovereign Internal Developer Platform (IDP). Unlike traditional IDPs, often designed with the native, proprietary services of a single global hyperscaler in mind, a sovereign IDP decouples from reliance on a single hyperscaler. By leveraging open-source standards, localized service providers, and self-hosted management planes, organizations can achieve a state of "exit-strategy-by-design."

Key findings and architectural sovereignty principles

Compliance as a global driver

Regulatory frameworks in Europe and beyond now mandate explicit proof of provider diversification and the existence of viable, tested exit strategies to ensure operational resilience in critical sectors.

Self-hosted deployment for security control

SaaS-hosted IDP tooling cedes end-to-end lifecycle control to an external provider, making it impossible to enforce custom security policies. Self-hosted IDP building blocks ensure that the organization's security controls govern every layer of the platform - nothing enters or exits without explicit authorization.

The abstraction mandate	Sovereignty is achieved not only by where data physically resides, but by the portability of the workflows that process it. “Everything-as-code” is the primary technical enabler of this portability, shifting the root of trust from the vendor to the organization’s own Git repositories.
Decoupled orchestration	Using a framework-agnostic platform orchestrator, a backend that enables platform engineering teams to maintain a consistent, high-quality developer experience across a fragmented, multi-provider sovereign infrastructure and tooling landscape.
AI sovereignty	The integration of Artificial Intelligence (AI) into the software development lifecycle requires new safeguards. Sovereign IDPs must enable the optionality for localized models hosted on regional infrastructure to prevent the leakage of Intellectual Property (IP) into foreign training datasets.

To achieve true strategic autonomy, organizations must prioritize “sovereignty-by-design.” They must engineer their internal platforms to ensure the entire software delivery lifecycle remains fully operational even if foreign legal, commercial, or connectivity ties are severed.

2

Introduction: The global mandate for digital sovereignty

Digital sovereignty in the modern enterprise context encompasses an organization's ability to control its technical destiny and protect its intellectual property, maintaining operational continuity regardless of external geopolitical, legal, or commercial pressures. It is the architectural manifestation of self-determination.

For the past decade, enterprise IT strategy has been dominated by the "hyperscaler-first" mandate. The sheer velocity and scale of the managed service ecosystem provided by global cloud providers (predominantly US-based) offered undeniable speed-to-market advantages. However, the trade-off for this speed is deep technical lock-in through the adoption of proprietary APIs and vendor-specific operational tooling.

On 12 June 2026, the US government issued an export control directive that forced Anthropic, the maker of Claude, to disable Claude Mythos 5 and Fable 5 for every non-American customer worldwide, including EU governments mid-procurement, for cybersecurity reasons.

The lesson generalizes beyond more than one vendor or legal jurisdiction. Any frontier AI or LLM capability hosted in a single jurisdiction can be revoked overnight on national-security grounds that the affected users are never made aware of. This action highlights that sovereignty is not a procurement preference but is, in fact a continuity requirement.

The geopolitical and legal triggers

The landscape has shifted, driven by legal pressure. And while the tension is most acute in Europe, the Middle East and Asia, the regulatory ripples are global.

In Europe, the regulatory environment tightened dramatically to protect critical infrastructure and citizen data.

- The NIS2 Directive establishes a unified legal framework to ensure cybersecurity across 18 critical sectors in the European Union. It mandates that member states define national cybersecurity strategies and enforce strict incident reporting and risk management measures, particularly regarding supply chain security and cloud service providers.
- DORA is aimed specifically at the financial sector. The act forces institutions to map their Information and Communication Technology (ICT) dependencies. It requires financial entities to prove they are not overly dependent on a single critical ICT third-party provider (CTPP) and that they maintain highly functional, tested exit strategies to migrate away from a provider without operational disruption.

While Europe leads this regulatory charge, similar frameworks are establishing a global baseline for sovereignty:

- **India**: The Digital Personal Data Protection Act (DPDP) introduced stringent mandates for data protection, restricted cross-border transfers and processing of sensitive personal data, forcing multinationals to adopt localized infrastructure architectures.
- **Brazil**: The LGPD (General Data Protection Law) established advanced frameworks for restrictions on cross-border data transfers.
- **North America**: Recent legal rulings in Canada demonstrated that even European cloud providers operating local data centers can be caught in jurisdictional tugs-of-war over data stored on foreign soil, highlighting that the risk of extraterritorial legal overreach is bi-directional.

The geopolitical tensions in the Middle East, particularly following the strategic shifts that accelerated from 2019 onward, have prompted a broad reassessment of IT sovereignty. Middle Eastern states and other US allies are increasingly prioritizing IT sovereignty (e.g. by reducing dependence on foreign software, hardware, and cloud services) to retain autonomy and protect their own strategic interests.

The US government has similarly recognized the risks of over-reliance on commercial software in critical infrastructure contexts. While the US has long led global software and technology, a growing policy consensus favors retaining control and resilience in key areas, a shift toward domestic technology independence rather than global isolation.

China's approach to technology sovereignty is among the most comprehensive globally, having built a largely self-sufficient technology ecosystem over two decades to reduce reliance on foreign technology, protect its economy from sanctions, and project influence. This strategy, which is tied

to national security, economic development, and geopolitical objectives, is most visibly embodied in the “Made in China 2025” initiative.

The sovereignty imperative extends beyond cloud infrastructure. Organizations increasingly recognize that an IDP built on proprietary SaaS-based tooling (e.g., portals, orchestrators, CI/CD pipelines, and secrets managers governed by foreign legal jurisdictions) is not sovereign regardless of where the underlying compute resides, unless those tools are also available as fully on-premises enterprise deployments under the customer’s own jurisdictional control. The VCS that hosts the source code, the orchestrator that controls deployment logic, the identity provider that governs access, and the AI assistant that processes code: each of these IDP components must be subject to the same sovereignty evaluation criteria as the infrastructure layer they manage. A cloud infrastructure that achieves data residency compliance while the control plane operates under foreign legal jurisdiction provides only partial sovereignty.

3

The enterprise challenge: The legal sandwich and provider lock-in

The modern enterprise operates at a precarious intersection of geopolitical legal friction and commercial technology dependencies. This challenge manifests as a "legal sandwich," where conflicting jurisdictional mandates (most notably the friction between the US CLOUD Act and the EU GDPR) threaten operational continuity and data sovereignty. However, this legal vulnerability is profoundly compounded by a technical one: deep vendor lock-in driven by proprietary cloud APIs and the commercial co-option of open-source ecosystems. Together, these legal and structural forces create an unsustainable risk profile, requiring organizations to look past marketing definitions of open source and engineer true, self-hosted operational autonomy.

The mechanics of the legal sandwich

The US CLOUD Act grants United States law enforcement agencies the authority to compel US-based technology companies to hand over requested data, regardless of whether that data is stored on servers located within the US or in foreign countries.

The EU GDPR strictly prohibits the transfer of European citizens' personal data to third countries unless adequate safeguards are in place. It also protects against unwarranted access by foreign governments.

When a European enterprise uses a US-based cloud provider or a US-based internal developer portal (even if the provider stores data in a physical data center in Paris or Frankfurt), that provider is caught in the middle. If a US court issues a subpoena under the CLOUD Act, the provider is legally obligated by the US to hand over the data. Doing so, however, places them in direct violation of the GDPR, exposing them and their enterprise clients to severe penalties.

For the enterprise, this is an unacceptable, unmanageable risk. It undermines the premise that data residency equates to data sovereignty.

The technical reality of lock-in

Beyond the legal risks, the hyperscaler-first approach creates severe technical lock-in, making a rapid exit strategy nearly impossible to execute. To comply with frameworks like DORA, organizations cannot simply write a theoretical exit strategy document; they must have the operational capability to execute it. Achieving this requires a fundamental decoupling of software development and delivery from the underlying infrastructure providers.

Open source ecosystems vs. commercial capture

A common assumption holds that open source software is inherently sovereign and free from lock-in. The reality is more nuanced, and the trend is moving in an unfavorable direction.

Open source software (as defined by the Open Source Initiative) makes source code freely available for use, modification, and distribution under permissive licensing terms. While open source can serve as a foundation for technical sovereignty, several structural pressures are eroding that advantage.

TREND	EXAMPLE	SOVEREIGNTY RISK FACTOR
Open core model	Backstage, Elasticsearch, MongoDB, Redis, Terraform	Core is open, but enterprise features are proprietary — users migrate to open source and then face commercial lock-in for critical capabilities
License changes	Elastic (SSPL), MongoDB (AGPL → SSPL), HashiCorp (BSL)	Shifts from permissive licenses break trust and retroactively restrict usage
Cloud co-option	AWS OpenSearch, Google Cloud managed services	Hyperscalers repackage open source projects as managed services with proprietary layers, capturing users without contributing proportionately back
Acquisitions	Red Hat (IBM), GitHub (Microsoft), HashiCorp (IBM)	Community priorities shift post-acquisition

The practical conclusions for sovereign IDP design are these:

- Sovereignty requires community-driven, pure open source. While open-core models provide great entry points, their enterprise SaaS layers are proprietary products. True technical sovereignty relies on the uncompromised, freely modifiable core software.
- License changes erode trust. Organizations must audit licenses before adoption and monitor for changes. Recent licensing shifts by commercial entities underscore why enterprises must champion foundations like the Linux Foundation or CNCF, ensuring their core tooling remains perpetually open and community-governed.
- Cloud providers co-opt open source. Managed services built on open source projects frequently introduce proprietary layers that recreate the lock-in the original project was designed to avoid.

For organizations evaluating IDP tooling, the test is not whether a tool has an open source license, but whether the organization can self-host, operate, and migrate away from it without dependency on a single commercial entity. This is the criterion that defines sovereign-ready tooling.

4

Strategic framework: Diversification for strategic autonomy

Real sovereignty requires a paradigm shift from geographical residency to technical autonomy. A technology stack that is deemed sovereign solely because its physical servers reside in a specific country is an illusion of sovereignty if the control plane, management APIs, and underlying hypervisor are governed by foreign legal jurisdictions.

To achieve strategic autonomy, enterprises must adopt a framework based on two core pillars: provider diversity and architectural decoupling.

Pillar 1: Provider diversity and the multi-cloud imperative

Strategic autonomy mandates moving beyond a single-hyperscaler monoculture. Organizations must design for a multi-cloud or hybrid-cloud environment that incorporates sovereign, localized providers alongside global hyperscalers.

In the European context, this involves integrating providers certified under stringent sovereign frameworks, such as the French SecNumCloud qualification. SecNumCloud guarantees that the cloud provider is European-owned, immune to extra-European legislation (like the CLOUD Act), and provides the highest levels of security isolation.

A sovereign infrastructure strategy within the EU typically segments workloads based on risk and regulatory requirements:

- → **European sovereign clouds**: Highly sensitive data, regulated financial transactions, and critical infrastructure workloads are mandated to run on providers such as OVHcloud, Cloud Temple, NumSpot, or Bleu.
- → **Global infrastructure**: Non-sensitive workloads and highly elastic web applications, along with non-regulated environments, may continue to run on global providers (AWS, Azure, GCP) to leverage specific managed services or global edge networks.

Pillar 2: Architectural decoupling via the IDP

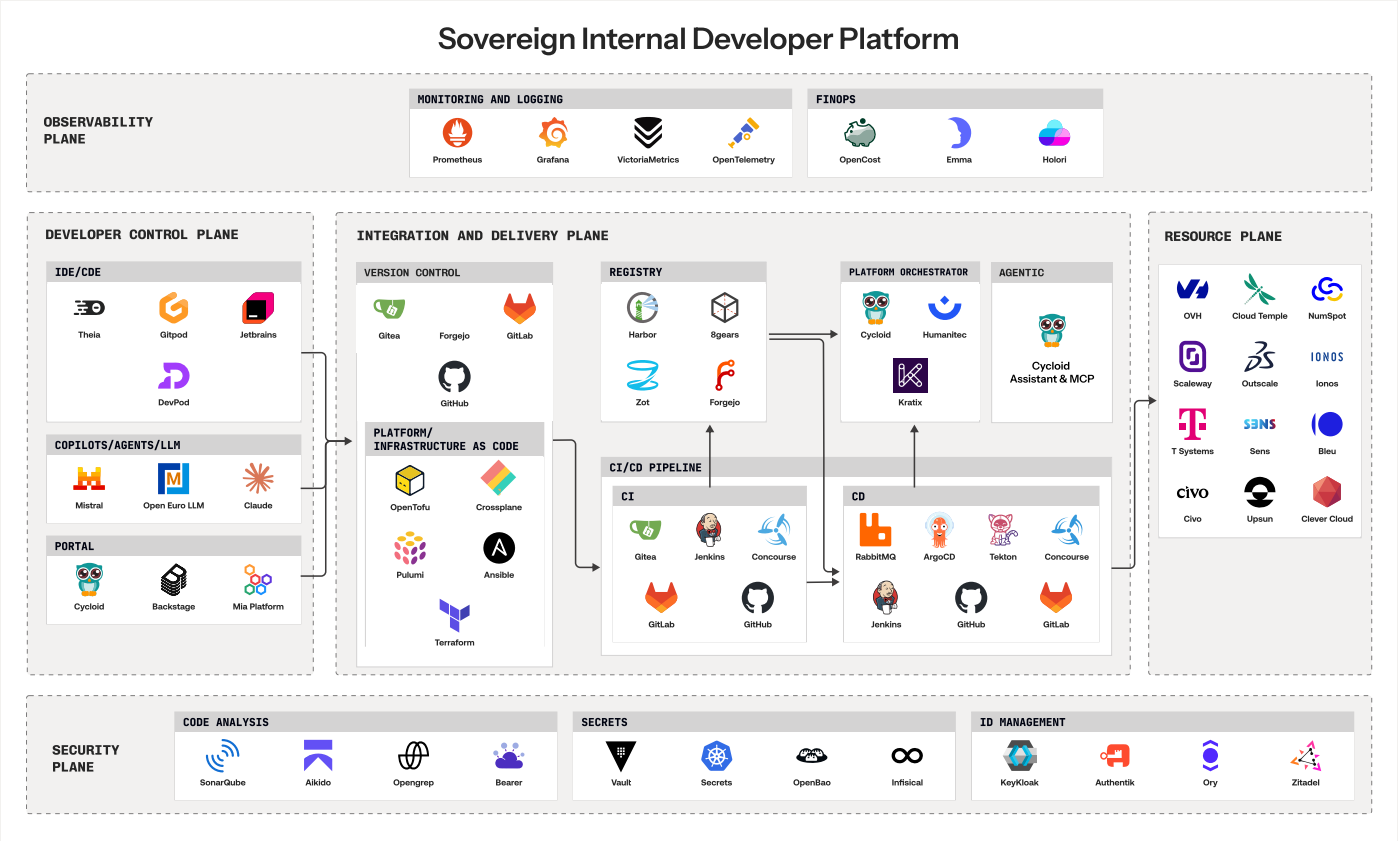
The presence of multiple infrastructure providers introduces immense operational complexity. If a developer must learn a different deployment process for an EU-based provider like OVHcloud than the one they use on AWS, development velocity will collapse.

This is the strategic role of the sovereign IDP with a portal and platform orchestrator like **Cycloid**s' as its centerpiece. The portal can serve as a central access point, and the orchestrator acts as an abstraction layer between the developer and the fragmented infrastructure landscape. By centralizing the portal and the orchestration, the IDP ensures that the golden paths used by developers to build, test, and deploy software remain consistent, well-supported and optimize day 2 operations, which are as important as the design, build, and deployment phase. This standard applies regardless of whether the target destination is a local SecNumCloud provider, a global hyperscaler, or an on-premise air-gapped environment.

5

Reference architecture for the sovereign IDP

This section presents a enterprise-grade reference architecture for a sovereign IDP. It is designed to showcase best practices for infrastructure deployment that prioritize scalability, security, reliability, and absolute strategic autonomy across distributed systems.



The architecture is built upon the fundamental principle of utilizing self-hosted or open-source-first components. This design localizes the root of trust within the organization's legal and physical control, rather than outsourcing it to a foreign SaaS provider.

The five architectural planes

Modern platform engineering architectures are designed as interacting planes rather than rigid layers. Layers imply a strict, top-to-bottom monolithic dependency. Planes are composed of discrete, swappable capabilities that interact via standardized Application Programming Interfaces (APIs). This composability is the bedrock of the exit-by-design strategy.

The sovereign IDP reference architecture consists of five distinct planes:

01 — The **Developer Control Plane** acts as the primary interface and interaction layer for software engineers. Examples include Visual Studio Code, Mistral Code, Claude Code Self-hosted, Backstage or **Cycloid** Self-Hosted.

02 — The **Integration and Delivery Plane** is the mechanical engine handling continuous integration, continuous delivery, and infrastructure orchestration. Examples include Gitea, Terraform, Ansible, Harbor Self-Hosted and ArgoCD to name a few.

03 — The **Resource Plane** includes physical and virtual infrastructure across sovereign and global providers. Examples include OVH, Scaleway, IONOS, Outscale or Clevercloud.

04 — The **Security Plane** provides localized identity, secret management, and policy enforcement across all other planes.

05 — The **Observability Plane** is the overarching monitoring and observability layer that provides deep visibility into platform health, financials, and the carbon footprint.

The platform orchestrator acts as the central nervous system, connecting the developer intent established in the control plane with the execution mechanics of the Integration and Delivery Plane, ultimately provisioning resources on the Resource Plane.

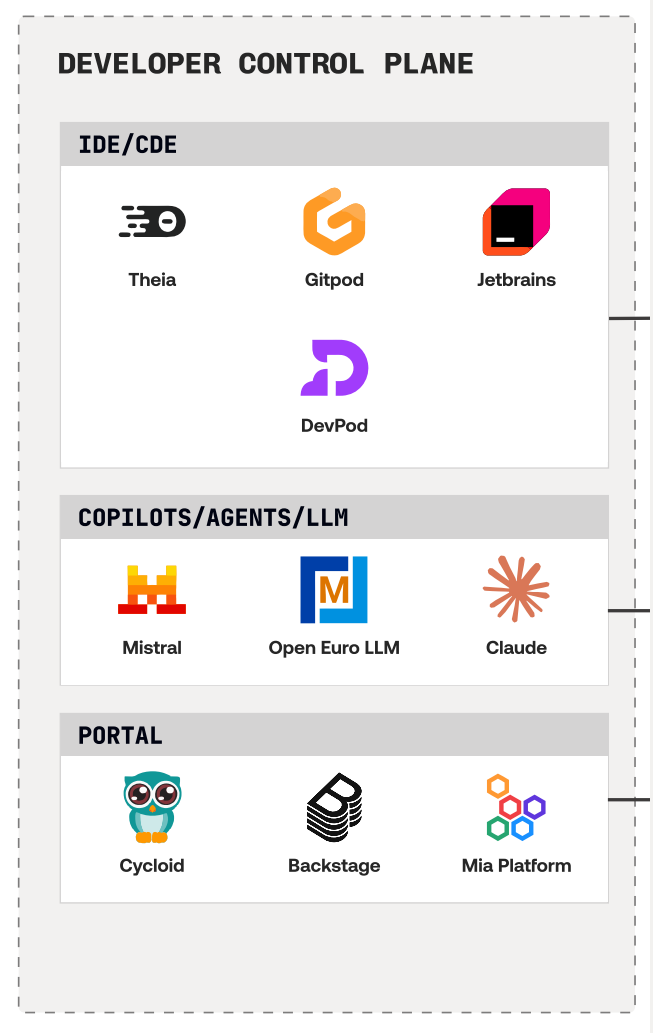
6

Architectural planes in detail

The following sections provide a deep dive into the specific tooling, mechanisms, and sovereign design principles embedded within each architectural plane.

6.1: The Developer Control Plane

The Developer Control Plane is the entry point for developers interacting with the platform. Its primary directive is to abstract the complexity of the sovereign infrastructure below it, providing a frictionless, self-service experience that minimizes cognitive load.



The portal

The portal serves as the single pane of glass for the entire engineering organization. In a sovereign context, relying on a SaaS-based portal hosted outside the organization's jurisdiction introduces unacceptable risk.

This architecture uses a self-hosted **Cycloid** portal instance as an example. By self-hosting, the organization guarantees that its service catalog, resource discovery mechanisms, plugins, and internal documentation remain accessible even in an air-gapped scenario or during a major external network partition. Developers and end-users use the portal to request new tools or access them, request new environments through forms, and monitor deployments. They can access logs without ever needing to log directly into a cloud provider account and get full visibility of ownership and metrics. They can validate requests through approvals, get a view on

asset inventory, and use plugins designed by platform teams across all ecosystems (open source or proprietary) or control cloud cost management and carbon footprint. AI assistants can help end-users navigate onboarding, locate information, and trigger day-2 operations where organizational governance permits.

Integrated Development Environments (IDEs) and Cloud Development Environments (CDEs)

Development environments must be standardized to prevent the “works on my machine” anti-pattern. The architecture supports standardized local IDEs (such as Visual Studio Code) or self-hosted CDEs. All developers operate with the same baseline tooling and security configurations, reducing onboarding friction and improving code quality.

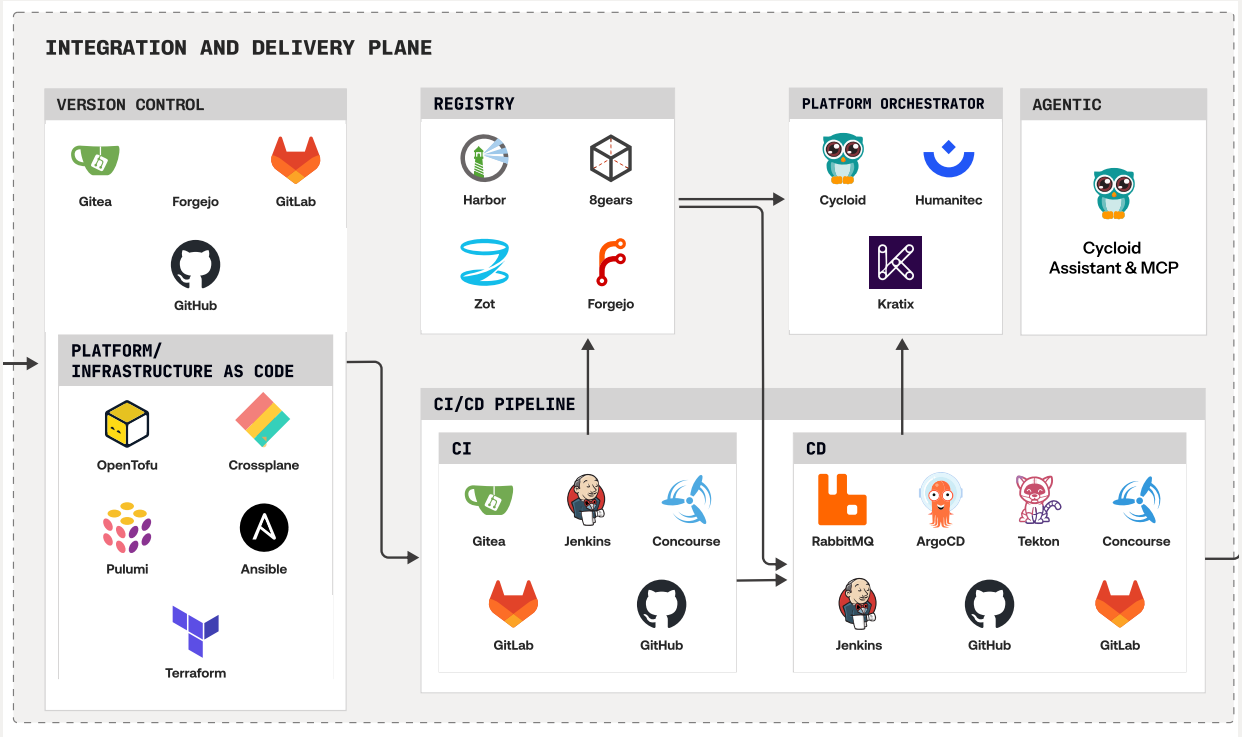
Sovereign AI, Copilots, and Large Language Models (LLMs)

The rapid adoption of AI coding assistants introduces a critical new vector for IP leakage. Traditional, proprietary Copilots constantly transmit code snippets and context back to vendor-controlled servers for processing and potential model training, violating sovereign data boundaries.

To maintain IP control, this architecture mandates the use of sovereign AI. This requires utilizing open-weight models, such as Mistral Code or locally deployed instances of other open-weight models, hosted on localized Graphics Processing Unit (GPU) infrastructure provided by European vendors like Scaleway. By routing all developer AI assistance queries to self-hosted or localized models, the enterprise ensures that its proprietary algorithms, business logic, and infrastructure definitions remain under its control and are never ingested by foreign, centralized AI training pipelines.

6.2 The Integration and Delivery Plane

This plane is the mechanical core of the IDP. It translates the intent submitted by the developer via the portal into executable code. It then tests and delivers it to the designated infrastructure. To ensure an exit strategy, every component here must be open-source or fully controllable by the enterprise.



Sovereign Version Control Systems (VCS)

The source code repository is the ultimate source of truth in any modern engineering organization. Relying on centralized, SaaS-based platforms (like GitHub or GitLab SaaS) means the enterprise doesn't truly own its source of truth. If the SaaS provider changes their terms of service, experiences an outage, or blocks access due to geopolitical sanctions, the enterprise's ability to operate ceases immediately.

The sovereign architecture utilizes self-hosted, open-source VCS solutions such as Forgejo or Gitea. These lightweight, highly performant solutions provide full Git functionality. They also provide pull request workflows and issue tracking, ensuring the organization maintains absolute cryptographic control over its codebase. As an alternative, you can use self-hosted GitHub Enterprise or GitLab.

CI/CD pipelines

The CI pipeline is responsible for building, testing, and packaging applications. The architecture uses localized CI runners (configured via tools such as GitLab CI runners or Jenkins) connected to the self-hosted VCS to ensure the build process runs entirely within the sovereign boundary.

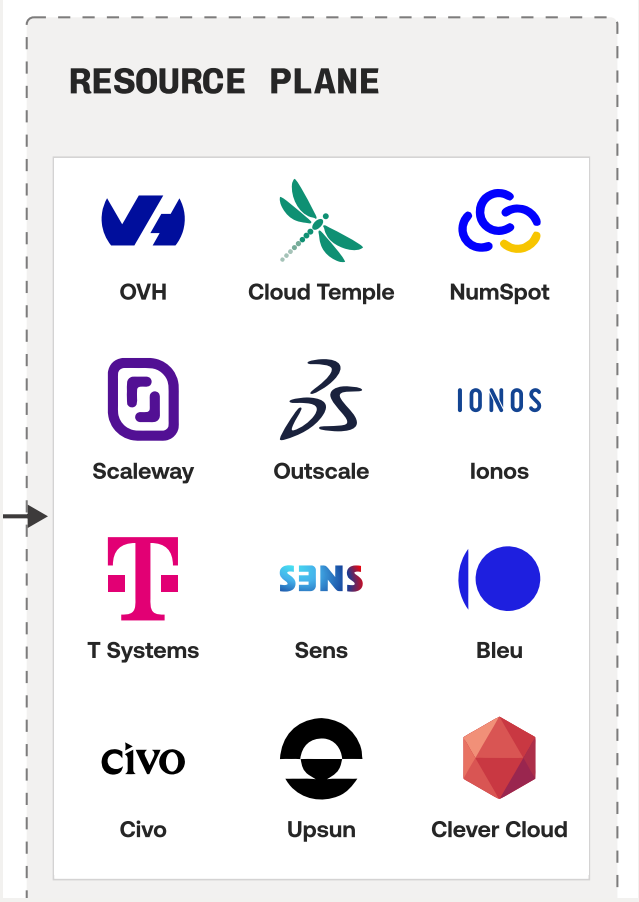
The Platform Orchestrator

A platform orchestrator like **Cycloid**, Humanitec, or Kratix is the crucial decoupling mechanism in this architecture. While CI/CD pipelines handle the application code, the orchestrator handles the infrastructure and the deployment logic.

The orchestrator acts as the translation layer between the developer's request for a service and the specific Infrastructure as Code (IaC) modules required to build that service on the target cloud. By centralizing the orchestration logic, the organization avoids building brittle, hard-coded deployment scripts. An orchestrator like **Cycloid** reads the organizational policies. It pulls the correct standardized infrastructure modules and executes the deployment across the varied Resource Plane.

6.3 The Resource Plane

The Resource Plane represents the actual compute, network, database, and storage capacity where the applications reside. This architecture intentionally fragments the Resource Plane to mitigate single-provider risk and comply with DORA requirements.



Certified sovereign cloud providers (EU-wide)

These providers meet strict EU sovereignty criteria, including data residency, legal immunity from non-EU laws, and EU-based ownership.

Tier 1: Fully sovereign (EU-owned, EU-hosted, no US influence)

PROVIDER	COUNTRY	COMPLIANCE CERTIFICATIONS	KEY FEATURES	TARGET USE CASES
Outscale (by Dassault Systèmes)	France	SecNumCloud, GDPR, SOC 2 Type II	Sovereign IaaS/PaaS, compatible with AWS APIs (for migration)	Enterprise, industrial sectors
OVHcloud	France	GDPR, SecNumCloud, ISO 27001	One of the largest EU-owned cloud providers, self-owned data centers, anti-DDoS protection	SMEs, startups, global enterprises
Scaleway	France	GDPR, ISO 27001, SOC 2	Sovereign bare metal, Kubernetes, AI/ML tools	Developers, AI startups
T-Systems (Deutsche Telekom)	Germany	GDPR, ISO 27001, C5 (German Cloud)	Hybrid cloud, sovereign Kubernetes (Open Telekom Cloud)	Automotive, finance, public sector
PlusServer	Germany	GDPR, ISO 27001, BSI C5	100% German-owned, carbon-neutral data centers	Mid-market, regulated industries
Stackit (by Schwarz Group)	Germany	GDPR, ISO 27001, C5	Sovereign Kubernetes, serverless, edge computing	Retail, logistics, IoT
Aruba Cloud	Italy	GDPR, ISO 27001, AGID	Sovereign IaaS/PaaS, Italian government-approved	Public administration, healthcare
CloudFerro	Poland	GDPR, ISO 27001, Polish Government Cloud	Sovereign cloud for Polish public sector	Government, defense, critical infrastructure

Tier 2: Sovereign-ready (EU-hosted, strong compliance)

These providers host data in the EU and meet most sovereignty requirements, but may have some foreign ownership or dependencies.

PROVIDER	COUNTRY	COMPLIANCE CERTIFICATIONS	KEY FEATURES	TARGET USE CASES
IBM Cloud (EU)	Germany/ Netherlands	GDPR, ISO 27001, SOC 2	Hybrid cloud, AI/ML, EU data residency	Enterprise, regulated industries
SAP BTP	Germany	GDPR, ISO 27001, SOC 2	Sovereign PaaS, integrated with SAP apps	ERP, finance, supply chain
Atos OneCloud	France	GDPR, SecNumCloud, ISO 27001	Hybrid cloud, cybersecurity services	Defense, aerospace, energy
Orange Business Services	France	GDPR, ISO 27001, SecNumCloud	Sovereign SD-WAN, edge computing	Telecom, IoT, global enterprises
Leaseweb	Netherlands	GDPR, ISO 27001, SOC 2	Bare metal, dedicated servers, CDN	Gaming, media, high-performance computing
Hetzner Cloud	Germany	GDPR, ISO 27001	Cost-effective, developer-friendly	Startups, SMEs, open-source projects
UpCloud	Finland	GDPR, ISO 27001, SOC 2	High-performance cloud, MaxIOPS storage	Gaming, e-commerce, databases

European sovereign cloud initiatives and alliances

These consortia, frameworks, and government-backed projects aim to standardize sovereignty requirements and promote EU cloud independence.

Government and industry alliances

INITIATIVE	LEADERS	GOAL	KEY MEMBERS	STATUS (2026)
Gaia-X	Germany, France	Federated, sovereign cloud ecosystem for Europe	SAP, Deutsche Telekom, OVHcloud, Atos, BMW, Siemens	Live
EU Cloud Rulebook	European Commission	Standardized sovereignty requirements for cloud providers	All Tier 1 providers	Advanced draft (2025–2026), mandatory for public sector procurement expected
Sovereign Cloud Stack (SCS)	Open source community	Open-source toolkit for sovereign clouds	SAP, PlusServer, Stackit	Active (used by German government)
European Data Spaces	EU Commission	Sector-specific sovereign data sharing	Siemens, Airbus, Philips, Volkswagen	In force (2025), operational rollout 2028–2029
Open Telekom Cloud (OTC)	Deutsche Telekom	Sovereign public cloud for Germany/EU	T-Systems, SAP, Siemens	Live (GDPR and C5 certified)

Global infrastructure

The architecture acknowledges that not all workloads require strict sovereignty. For non-sensitive, highly elastic edge workloads, the IDP can still orchestrate deployments to global hyperscalers (AWS, Azure, GCP).

The key architectural safeguard is that control of these global resources remains within the sovereign Integration and Delivery Plane. The enterprise utilizes global infrastructure as a pure commodity, managing it entirely through abstract IaC, ensuring that if a global provider must be abandoned, the workload can be redeployed to a sovereign provider with minimal refactoring.

6.4 The Security Plane

Security cannot be bolted on as an afterthought; it must be a foundational plane that cuts across all other architectural components. In a sovereign context, the concept of localizing the root of trust" is paramount.

Identity and Access Management (IAM)

Relying on a cloud provider's native IAM (e.g., AWS IAM) ties the organization's identity perimeter to that specific vendor. The sovereign architecture utilizes an independent, self-hosted identity broker like Keycloak. Keycloak integrates with the enterprise's existing Active Directory, providing centralized authentication and authorization across the entire platform, regardless of the underlying cloud provider.

Secrets Management

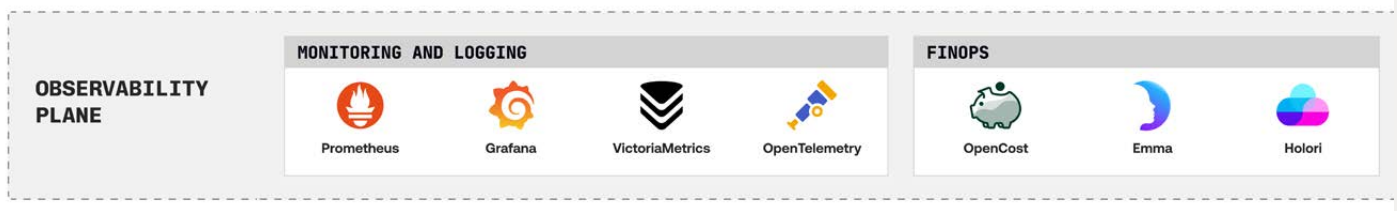
Similarly, utilizing provider-native Key Management Services (KMS) creates severe lock-in. The architecture mandates the use of a centralized, self-hosted secrets manager like Akeyless, Kubernetes Secrets store, External Secrets Operator, Doppler, SOPS, Sealed Secrets, Chamber, PAss, or Gopass. They serves as the single source of truth for all API keys, database passwords, and cryptographic certificates. Applications fetch secrets dynamically from the secrets manager, ensuring that sensitive credentials are never hard-coded in repositories or tied to a specific cloud provider's proprietary vault.

Policy as Code

To enforce compliance and security standards automatically, the architecture utilizes Open Policy Agent (OPA). OPA allows security teams to define policies as code (e.g., "No unencrypted storage buckets may be created" or "Workloads tagged as 'sensitive' can only be deployed to SecNumCloud providers"). These policies are enforced during the CI/CD pipeline and by the **Cycloid** orchestrator prior to any infrastructure provisioning.

6.5 The Observability Plane

Comprehensive visibility is required to manage a fragmented, multi-cloud environment. The Observability Plane cuts across all components, providing real-time data on system health and costs, along with environmental impact.



Metrics, logging, and tracing

The architecture utilizes the industry-standard, open-source monitoring stack: Prometheus for metric collection and alerting, and Grafana for data visualization. By self-hosting this stack, the enterprise ensures that its operational telemetry (which often contains sensitive metadata about system architecture and traffic patterns) is not exposed to third-party SaaS monitoring tools.

FinOps and GreenOps

A unique capability of a mature IDP is tracking both financial and environmental costs.

The architecture also integrates optional GreenOps tracking. By monitoring the carbon footprint of compute resources, organizations can make data-driven decisions to route specific workloads to cloud regions powered by renewable energy, aligning their technology strategy with corporate sustainability and Environmental, Social, and Governance (ESG) mandates.

7

Portability via IaC and GitOps

The defining characteristic of a sovereign IDP is the methodological enforcement of how tools are used. Rather than a legal document filed away for compliance purposes, the exit strategy is a highly functional technical capability.

In this reference architecture, that capability is realized through the strict, unyielding enforcement of everything-as-code and the principles of GitOps.

The abstraction layer: IaC and configuration management

Infrastructure as Code (IaC) is the mechanism that abstracts physical hardware into declarative configuration files. However, the choice of IaC tooling is critical to maintaining sovereignty.

Terraform was the industry standard. However, changes to proprietary licensing models highlight the risk of relying on tooling governed by a single commercial entity. To ensure true open-source freedom and long-term viability, this reference architecture explicitly mandates the use of OpenTofu.

OpenTofu, governed by the Linux Foundation, offers a drop-in replacement for Terraform, guaranteeing the core engine defining the organization's infrastructure remains open and free from vendor capture.

By defining all infrastructure, from OVHcloud load balancers to AWS S3 buckets, in declarative OpenTofu configuration files, the enterprise treats infrastructure as disposable and reproducible.

GitOps and the exit strategy by design

GitOps is the operational framework that enforces the. In a GitOps model, the self-hosted VCS (e.g., Forgejo or Gitea) is the sole source of truth for both application code and infrastructure configuration.

The architecture utilizes ArgoCD as the GitOps controller. ArgoCD continuously monitors the VCS repository. If the desired state declared in the Git repository differs from the actual state running in the cluster, ArgoCD automatically reconciles the difference.

All deployments, configuration changes, infrastructure provisioning, and policy updates are executed via Git pull requests. There are no manual interventions. This traceability is the foundation of the exit strategy. If the primary data center experiences a catastrophic failure or a sudden legal injunction forces the abandonment of a provider, the organization possesses the entire state of its business defined as code in its Git repositories. Disaster recovery becomes a matter of pointing ArgoCD at a new cluster on a different sovereign provider and allowing it to reconstruct the entire environment from scratch.

Air-gapped and on-premise capabilities

For highly regulated industries such as defense contractors and certain financial institutions, connecting to any public internet service (even a sovereign one) is not an option.

The requirement for an IDP to function in a fully disconnected, air-gapped environment is the ultimate stress test of its sovereignty. Because this reference architecture relies entirely on self-hosted control planes, self-hosted VCS, local container registries, and open-source orchestration (**Cycloid** and ArgoCD), the entire platform can be deployed within an isolated on-premise data center. The development organization maintains high-velocity, automated software delivery without sending a single packet of data across the public internet.

8

Sample use cases

The following use cases illustrate how the reference architecture is applied across representative enterprise scenarios. Each demonstrates how the sovereign IDP decouples development workflows from specific infrastructure providers, enabling compliance by design rather than by policy document.

Use case 1: Multi-cloud and hybrid infrastructure provisioning

Scenario: A European financial institution needs to deploy applications across AWS (Frankfurt), Azure (Germany), and on-premises OpenStack while complying with GDPR and NIS2.

STEP	ACTION	TOOLS	SOVEREIGNTY BENEFIT
1. Define IaC	Use OpenTofu to define cloud-agnostic infrastructure	OpenTofu, Pulumi, Crossplane	Avoids cloud-specific lock-in
2. Store IaC in a sovereign Git repo	Host code in a self-hosted Gitea instance	Gitea, Forgejo	No GitHub lock-in; data stays in EU
3. Use a sovereign CI/CD pipeline	Deploy Tekton or Argo Workflows for cloud-agnostic CI/CD	Tekton, Argo, Workflows, Concourse	No GitHub Actions/Azure Pipelines lock-in
4. Provision infrastructure via Cycloid	Use **Cycloid**'s catalog to deploy IaC modules across AWS, Azure, OpenStack	**Cycloid**, Crossplane	Single pane of glass for multi-cloud
5. Enforce compliance with Policy as Code	Use OPA or Kyverno to validate IaC against GDPR and NIS2	OPA, Kyverno	Automated compliance checks
6. Audit and monitor	Use Prometheus, Grafana, and Loki for observability	Prometheus, Grafana, Loki	No Datadog/New Relic lock-in

Key benefits:

- → **Avoids vendor lock-in (works with any cloud or on-premise environment)**
- → **Compliance by design (GDPR and NIS2 enforced via Policy as Code)**
- → **Full organizational control over data and infrastructure**

Use case 2: Sovereign Kubernetes (K8s) management

Scenario: A German automotive manufacturer wants to manage Kubernetes clusters across on-premises (OpenShift) and sovereign clouds (OVHcloud, T-Systems) while complying with C5 and GDPR.

STEP	ACTION	TOOLS	SOVEREIGNTYBENEFIT
1. Deploy self-managed Kubernetes	Use k3s, OpenShift, or Rancher for on-premises Kubernetes	k3s, OpenShift, Rancher	No EKS/AKS/GKE lock-in
2. Use Cycloid for cluster provisioning	Define Kubernetes clusters as catalog items in **Cycloid**	**Cycloid**, Terraform	Single interface for all clusters
3. Enforce security policies	Use Kyverno or OPA Gatekeeper to enforce pod security policies	Kyverno, OPA Gatekeeper	CIS benchmarks, NIS2 compliance
4. Manage secrets with SOPS or Akeyless	Store Kubernetes secrets in SOPS-encrypted files or Akeyless	SOPS, Akeyless	No proprietary vault l ock-in
5. Deploy apps with ArgoCD	Use ArgoCD for GitOps deployments	ArgoCD	No proprietary CI/CD lock-in
6. Monitor with Prometheus and Grafana	Self-host the observability stack	Prometheus, Grafana, Loki	No Datadog/New Relic lock-in

Key benefits:

- → **Full control over Kubernetes clusters (no managed service lock-in)**
- → **Compliance automated (C5 and GDPR enforced via Policy as Code)**
- → **Portable across clouds (works with any Kubernetes distribution)**

Use case 3: Sovereign CI/CD pipelines

Scenario: A French healthcare provider needs CI/CD pipelines that comply with GDPR, with no data leaving the EU.

STEP	ACTION	TOOLS	SOVEREIGNTY BENEFIT
1. Self-host GitLab or Gitea	Replace GitHub/GitLab.com with a self-hosted instance	GitLab (self-hosted), Gitea	No US jurisdiction; data stays in EU
2. Use Tekton for CI/CD	Replace GitHub Actions/Azure Pipelines with Tekton	Tekton	No vendor lock-in
3. Store pipeline secrets in Akeyless	Use Akeyless for dynamic secrets	Akeyless	No proprietary vault lock-in
4. Define pipelines in Cycloid	Use **Cycloid**'s catalog to standardize pipelines across teams	**Cycloid**, Tekton	Single pane of glass for CI/CD
5. Enforce compliance with OPA	Use OPA to validate pipelines against GDPR	OPA	Automated compliance checks
6. Deploy to sovereign clouds	Deploy apps to OVHcloud, Outscale, or on-premises	OVHcloud, Outscale, OpenStack	No AWS/Azure/GCP lock-in

Key benefits:

→ **No data leaves the EU (complies with GDPR)**

→ **No vendor lock-in (works with any cloud or on-premises environment)**

→ **Auditability (all pipeline runs are logged and compliant)**

Use case 4: Sovereign Identity and Access Management (IAM)

Scenario: A Swedish government agency needs centralized IAM for developers, operations staff, and contractors while complying with GDPR and Swedish data protection laws.

STEP	ACTION	TOOLS	SOVEREIGNTY BENEFIT
1. Deploy Keycloak	Self-host Keycloak for SSO, OIDC, and LDAP	Keycloak	No Okta/Azure AD lock-in
2. Integrate with Cycloid	Use Keycloak as **Cycloid**'s identity provider	**Cycloid**, Keycloak	Single sign-on for all tools
3. Enforce RBAC	Define roles and permissions in **Cycloid**'s catalog	**Cycloid**, OPA	Fine-grained access control
4. Audit access with Loki	Log all authentication events in Loki	Loki, Grafana	Immutable audit trails
5. Use short-lived credentials	Issue temporary credentials via Teleport or Akeyless	Teleport, Akeyless	Zero-trust access
6. Comply with GDPR	Use OPA to enforce data access policies	OPA	Automated GDPR compliance

Key benefits:

→ **Full control over identity and access**

→ **No US jurisdiction (Keycloak is EU-hosted)**

→ **Auditability (all access is logged and compliant)**

Use case 5: Sovereign observability and monitoring

Scenario: A Dutch financial services company needs monitoring and logging that complies with GDPR and DORA, with no data leaving the EU.

STEP	ACTION	TOOLS	SOVEREIGNTY BENEFIT
1. Deploy Prometheus and Grafana	Self-host Prometheus and Grafana for metrics and dashboards	Prometheus, Grafana	No Datadog/New Relic lock-in
2. Use Loki for logs	Self-host Loki for log aggregation	Loki	No Splunk/ELK lock-in
3. Integrate with Cycloid	Use **Cycloid**'s catalog to standardize monitoring stacks	**Cycloid**, Prometheus, Grafana	Single pane of glass for observability
4. Enforce retention policies	Use OPA to enforce log retention policies	OPA	Automated compliance
5. Store logs in sovereign storage	Use MinIO or Ceph for long-term log storage	MinIO, Ceph	No AWS S3/Azure Blob lock-in
6. Alert on anomalies	Use Alertmanager for compliance alerts	Alertmanager	Proactive issue detection

Key benefits:

→ **No data leaves the EU (complies with GDPR and DORA)**

→ **No vendor lock-in (works with any storage or cloud)**

→ **Full control over monitoring data**

Use case 6: Sovereign database management

Scenario: A Polish e-commerce company needs to manage PostgreSQL and MongoDB databases while complying with GDPR and Polish data protection laws.

STEP	ACTION	TOOLS	SOVEREIGNTY BENEFIT
1. Self-host PostgreSQL/ MongoDB	Deploy PostgreSQL and MongoDB on sovereign clouds (OVHcloud, CloudFerro)	PostgreSQL, MongoDB	No AWS RDS/Azure Cosmos DB lock-in
2. Use Cycloid for database provisioning	Define database templates in **Cycloid**'s catalog	**Cycloid**, Terraform	Single interface for all databases
3. Manage credentials with Akeyless	Use Akeyless for dynamic database credentials	Akeyless	No proprietary vault lock-in
4. Enforce backup policies	Use OPA to enforce backup retention policies	OPA	Automated compliance
5. Store backups in MinIO	Use MinIO for sovereign object storage	MinIO	No AWS S3 lock-in
6. Monitor database performance	Use Prometheus and Grafana for database metrics	Prometheus, Grafana	No Datadog lock-in

Key benefits:

→ **Full control over database data**

→ **No vendor lock-in (works with any cloud or on-premises environment)**

→ **Compliance by design (GDPR and Polish data protection laws)**

9

Conclusion: Future-proofing the platform

Building a sovereign IDP is more than an IT infrastructure project. It's a corporate risk mitigation strategy. It also serves as an architectural defense against future rigidity and vendor lock-in, while addressing the growing complexity of international legal conflicts.

By adopting a sovereignty-by-design approach, enterprises guarantee that their platform, and by extension, their business operations, will survive. Even if vendor relationships deteriorate or legal and geopolitical conditions shift, mandating immediate infrastructural isolation. The era of trusting a single global vendor to manage the entirety of an enterprise's technical destiny has ended, replaced by the mandate for provable, technical autonomy.

Recommendations for strategic action:

01 — **Audit present plans immediately**

Identify which components of your management stack (VCS, secret management, CI/CD, orchestration) are currently SaaS-only and lack a viable self-hosted or localized alternative. These are single points of failure in an exit strategy.

02 — **Establish golden paths on sovereign infrastructure**

Begin the transition by migrating non-critical or newly developed workloads to European sovereign providers (SecNumCloud), utilizing framework-agnostic orchestrators. Doing so validates the portability of your IaC modules before attempting a critical migration.

03 — **Implement strict IP tracking for AI**

Halt the uncontrolled usage of proprietary Copilots. Standardize the organization on localized, open-weight models (like Mistral) to maintain visibility and control over artifact generation and model lineage.

The future of the enterprise technology stack is modular. It's highly portable and fiercely autonomous. By leveraging the open-source community along with standardized IaC and framework-agnostic orchestration, organizations can escape the constraints of the legal sandwich and build a platform that truly serves their own strategic interests.

10

Appendix: The sovereign validation matrix

To help platform engineering leaders evaluate the true sovereignty of their architecture, Weave Intelligence has developed the following validation matrix. A truly sovereign IDP must answer “yes” to all criteria across all three domains.

Domain 1: Infrastructure and data

- → Does the primary infrastructure provider hold relevant regional sovereignty certifications (e.g., SecNumCloud, BSI C5)?
- → Is the data physically stored exclusively within the designated legal jurisdiction?
- → Is the infrastructure provider corporately immune to extraterritorial data requests (e.g., the US CLOUD Act)?

Domain 2: Tooling and control plane

- → Is the VCS self-hosted or controlled by the enterprise within its jurisdiction?
- → Is the platform orchestrator framework-agnostic, capable of deploying to both sovereign and global providers without requiring vendor-specific plugins?
- → Are all cryptographic keys, secrets, and IAM policies managed by a self-hosted solution (e.g., local Keycloak/Vault) rather than a cloud provider’s native managed service?
- → Is the IaC tooling open-source and free from restrictive vendor licensing (e.g., OpenTofu vs. proprietary Terraform)?

Domain 3: AI and intellectual property

- → Are AI coding assistants and LLMs hosted on local or sovereign infrastructure?
- → Does the enterprise possess an explicit guarantee that its code and telemetry data are not being ingested into external, foreign-owned AI training models?

Authors

Florian Lipp

SENIOR ANALYST @ WEAVE INTELLIGENCE

Dr. Florian Lipp is an analyst at Weave Intelligence, where he researches how platform engineering teams design, operate, and evolve their platforms. His current work spans topics including cloud economics and observability, helping teams move beyond dashboards and cost reports toward feedback systems that actually inform decisions. He co-organizes PlatformCon, curates its content program, and serves as editor in chief at platformengineering.org, two of the community's most important gathering points.

Sam Barlien

RESEARCHER @ WEAVE INTELLIGENCE

Sam Barlien is a researcher at Weave Intelligence, the research arm of platformengineering.org, the world's largest platform engineering community. With more than 10 years tracking technology communities and ecosystems, he brings first-hand perspective to his research on platform engineering and industry trends. He contributes to Weave Intelligence reports and conducts the weekly research interview series. He co-hosts PlatformCon, the world's largest platform engineering conference, and contributes to Platform Weekly, the Ambassador program and the platformengineering.org blog.

11

References

1. European Commission, The NIS2 Directive: High common level of cybersecurity across the Union, 2024.
2. European Parliament, Digital Operational Resilience Act (DORA), 2022.
3. Indian Ministry of Electronics and Information Technology, The Digital Personal Data Protection Act (DPDP), 2023.
4. Canadian data order risks blowing a hole in EU sovereignty," The Register, 27 November 2025, accessed June 2026, https://www.theregister.com/2025/11/27/canada_court_ovh/
5. Agence Nationale de la Sécurité des Systèmes d'Information (ANSSI), SecNumCloud 3.2 Reference Framework, 2025.
6. OpenTofu Project, The state of open-source infrastructure as code, 2025.
7. Weave Intelligence, Reference architecture of an Internal Developer Platform on AWS, 2025, https://weaveintelligence.io/research/ref-arch-aws

About Weave Intelligence

Weave Intelligence

Weave Intelligence is a leading analyst firm specializing in platform engineering.

By uniting a team of senior analysts with industry experts and enterprise leaders, we deliver the rigorous research that defines the field.

We enable organizations to leverage the #1 trend in IT as the modern framework for operational excellence and innovation.

Weave Intelligence GmbH
Wöhlertstraße 12-13
10115 Berlin

Disclaimer

This whitepaper was commissioned by **Cycloid**. Weave Intelligence retained full editorial independence throughout the research and writing process.

Weave Intelligence does not endorse any specific vendor, product, or service. The information contained in this report has been obtained from sources believed to be reliable. Weave Intelligence disclaims all warranties as to the accuracy, completeness, or adequacy of such information. This publication is provided on an “as-is” basis without warranty of any kind, either express or implied. Weave Intelligence shall have no liability for errors, omissions, or inadequacies in the information contained herein or for interpretations thereof. The reader assumes sole responsibility for the selection of these materials to achieve its intended results.