Accelerating Secure CI/CD Pipelines in Highly Regulated Industries

High-speed software delivery for financial services, healthcare, and critical infrastructure.

VARNISH SOFTWARE

A Varnish Software Strategy Guide

Executive Summary

Financial institutions, healthcare providers, and critical-infrastructure operators depend on large, intricate DevOps and CI/CD systems to build, test, and deliver software securely and predictably. These environments rely on a mix of dependencies that each form part of the supply chain. The registries that expose these artifacts come in the form of:

- **Source control platforms** (e.g. GitHub, GitLab)
- **Language dependencies** (e.g. Maven, NPM, Go, PyPI)
- **OS packages** (e.g. RPM, DEB)
- **Container registries** (e.g. Docker Hub, AWS ECR)
- **Orchestration environments** (e.g. Kubernetes)

Public and private registries are often hosted on SaaS platforms or accessed via self-hosted or cloud-based Universal Repository Managers (e.g. JFrog Artifactory, Sonatype Nexus). Across these systems, dependencies are fetched constantly by developers and automated build agents. In highly regulated environments, local builds can be restricted, with only centralized CI/CD systems permitted to compile code. While essential for auditability, this reliance creates bottlenecks and, critically, forces CI runners to maintain direct egress to the public internet for dependencies.

Varnish Orca addresses this challenge directly. Orca is a Virtual Registry Manager that accelerates registries for build and runtime artifacts. By consolidating registries into a single auditable point, Orca speeds up pipelines while proactively defending your supply chain. When deployed in front of artifact repositories, registries, and source servers, Varnish Orca can:

- **Accelerate CI/CD pipelines**: 2–10× faster builds and dependency fetches
- **Cut infrastructure & egress costs**: 50–80% backend offload
- **Boost resilience**: Continue serving content even during origin failures
- **Enable vendor neutrality**: Front JFrog, Nexus, GitHub, S3, and more
- **Improve developer velocity**: Reduce wait time, speed up releases
- **Enable network isolation**: CI runners can operate with zero public egress
- **Enhance security and control**: Enforcement, auditability, observability

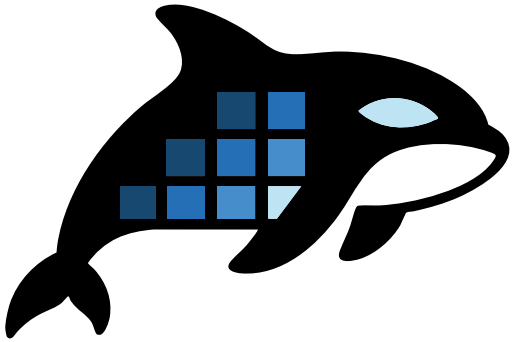
Orca scales performance and security simultaneously. Cache Docker images, NPM packages, Helm charts, Go modules, and more. Make CI/CD faster and more resilient without requiring developers to change tools or compromising strict compliance perimeters.

Who This Guide is For

Platform engineering, DevOps, technology leaders and security architects in financial services, healthcare, and critical-infrastructure industries such as energy, manufacturing, and telecom who are responsible for improving build and deployment performance while maintaining strict compliance, security, and audit controls.

Varnish Orca

(**O**bjects * **R**egistry * **C**ache * **A**rtifacts)



The Cost of Control

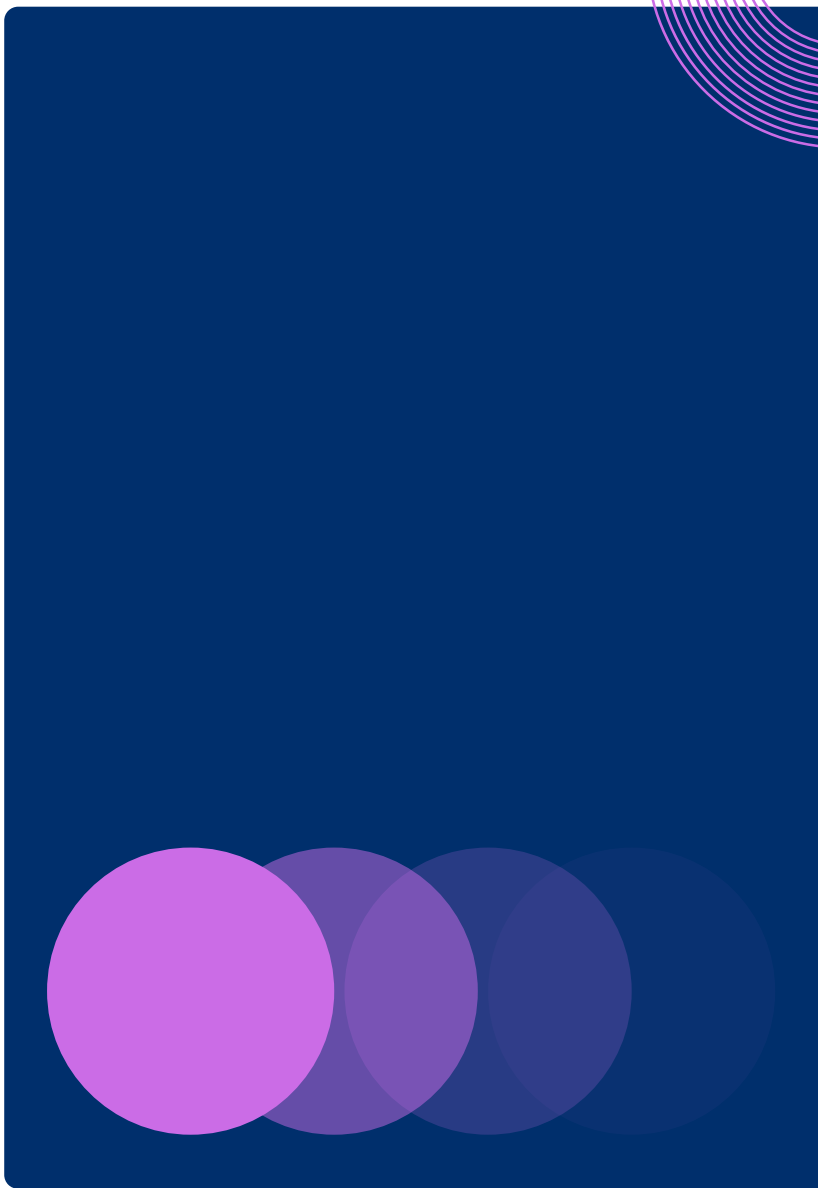
Even outside highly regulated sectors, large enterprises struggle to keep CI/CD fast. Thousands of concurrent jobs, large monorepos, and multi-gigabyte container images can overwhelm shared repositories and registries. In regulated environments, the challenge compounds. internal optimization must be fully traceable, and direct egress to public networks is a significant attack surface that traditional proxies cannot govern effectively.

Core DevOps Bottlenecks

Enterprise DevOps run at scale across many repos, specialized toolchains, and distributed teams competing for limited infrastructure on internal platforms:

- **Burstiness and scale**. CI peaks and large monolithic builds saturate shared artifact and registry servers.
- **Large, repetitive payloads**. The same Git packfiles, base images, and dependency graphs are fetched thousands of times.
- **Repository limits**. On-prem systems hit CPU/memory/licensing ceilings; SaaS repos add egress costs, throttling and rate limits.
- **Network latency**. Distributed teams span geographies; cross-region fetches add seconds or minutes to every build.
- **Chatty protocols**. Package managers make hundreds of small requests; each one traverses proxies and firewalls.
- **Shared hotspots**. Central repository managers or registry nodes become bottlenecks for global teams.
- **Vendor lock-in**. Consolidated platforms centralize value but raise cost and inertia.

Together, these factors increase build times, infrastructure costs, and operational risk. In regulated enterprises, these challenges meet the additional weight of governance and control frameworks, making technical optimization even harder.



> **“In regulated enterprises, these challenges meet the additional weight of governance and control frameworks, making technical optimization even harder.”**

Regulated Complexity

Regulatory frameworks extend the core challenges with layers of control and accountability:

- **Network inspection**. Artifact and source requests frequently traverse security appliances optimized for end-user traffic, not automated builds.
- **Third-party risk controls**. DORA and NIS2 encourage isolation of internal software supply chains from public networks.
- **Change control**. FFIEC, OCC, SOX, and PCI DSS require traceable changes and approvals before modifying infrastructure.
- **Provenance and integrity**. NIST SSDF and SLSA standards emphasize signed, reproducible builds, adding extra verification stages.
- **Data protection**. HIPAA, GDPR, and PCI DSS restrict replication outside specific zones.

Everyday Friction

Slow Clones and Fetches

Repository and registry access slowed by repeated authentication and network inspection.

Repository Bottlenecks

Central artifact and container registries overloaded by thousands of redundant requests.

Network Latency

Cross-region CI jobs and remote runners suffer multi-second delays on each pull.

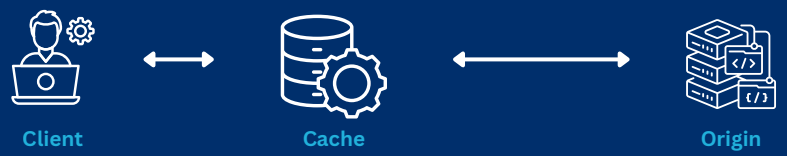
Redundant Traffic

Identical dependencies fetched repeatedly instead of reused.

Can It Be Cached?

Caching is one of the simplest and most effective ways to reduce latency and infrastructure load. By storing frequently accessed content closer to where it's consumed, delivery moves from remote repositories to local cache tiers inside secure data centers or VPCs. In DevOps, many high-latency transfers involve immutable, versioned data. Caching these artifacts stores the pre-computed result, meaning the expensive task of re-generating or re-fetching the artifact is avoided every time a build runs. This makes immutable data a perfect candidate for compliant caching.

Data Type	Cacheability	Notes
OCI layers and blobs **(Container images, Docker, ...)**	Immutable by digest; cache safely and indefinitely.	Manifests and image metadata are validated at pull time; underlying layers never change.
Git packfiles	Immutable objects; cache once fetched.	While obtained via POST, they represent fixed repository snapshots that can be safely reused.
Package artifacts **(Maven, npm, PyPI, NuGet, Helm, ...)**	Immutable, versioned bundles; cache aggressively.	Only manifests and metadata need periodic refresh.
Repository metadata	Semi-static; safe short-TTL cache.	Metadata changes can be revalidated automatically by the origin.
Helm charts	Typically OCI-based; same caching behavior as other OCI artifacts.	Helm v3+ uses OCI registry semantics for safe, immutable caching.



Caching stores frequently accessed content closer to where it's consumed.

Immutable artifacts like binaries, images, and bundles can be cached safely and for long durations, provided authentication, logging, and expiry are enforced. Metadata and manifests are usually lightweight and can either be revalidated automatically or cached for short intervals.

Varnish Orca in the CI/CD Stack

Varnish Orca is a Virtual Registry Manager that consolidates and accelerates registries for build and runtime artifacts. It is a self-service layer developers use automatically whenever they fetch code or containers, creating a seamless path for rapid retrieval while disconnecting runners from public egress. By caching Docker images, NPM packages, Helm charts, Go modules, and many more, Orca speeds up CI/CD pipelines, reduces developer friction, and lowers operational costs.

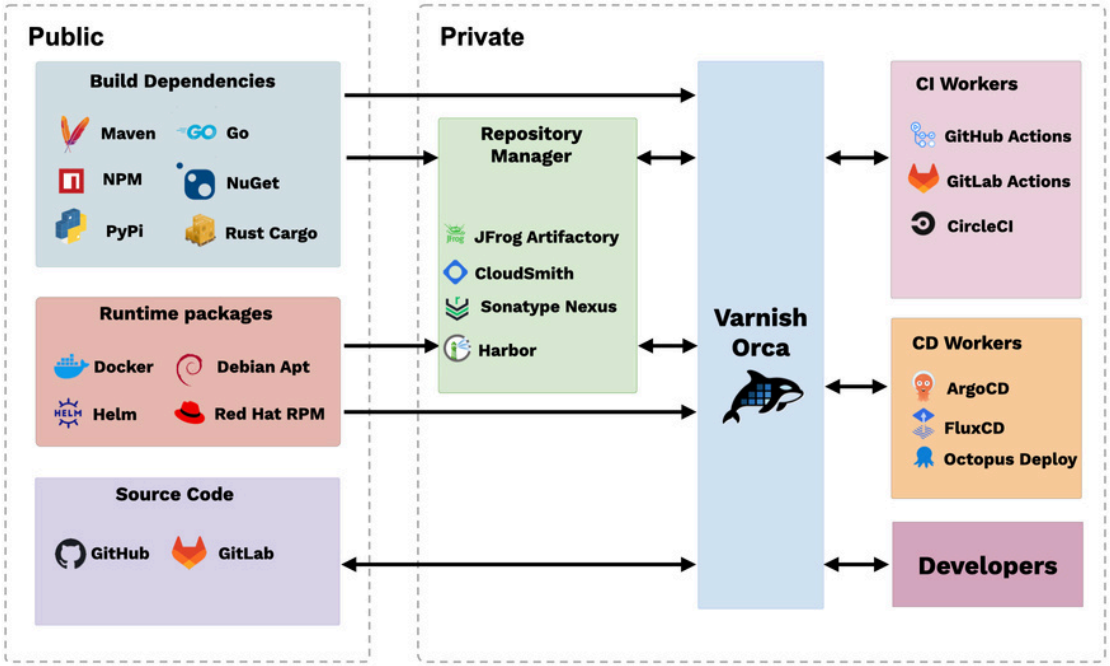
Varnish Orca caches registry HTTP responses in an implementation-native way. Whether running a *docker pull* or an *npm install*, Orca stores and serves dependencies locally or within your private network, delivering CDN-grade performance for internal builds.

Deploy Orca close to where it's needed, such as alongside CI/CD runners or build servers, to **reduce backend load**, **minimize redundant requests**, and **cut network latency** while **maintaining full audit visibility**.

With **YAML-based configuration**, teams can define routing and security policies in minutes. Authorized requests continue to rely on the origin's authentication mechanisms. Orca does not duplicate or store credentials, but efficiently reuses cached artifacts across authorized clients without compromising individual access controls. This ensures alignment with zero-trust principles.

Under the hood, Orca uses the **Varnish Massive Storage Engine (MSE)**, a persistent, disk-backed cache engineered for high-throughput, low-latency performance at multi-terabyte scale. It achieves in-memory-like speed with advanced I/O optimization and object indexing, making it ideal for artifact and image delivery workloads.

By consolidating delivery through Orca, you can unify fragmented ecosystems, eliminate redundant infrastructure, and achieve a high-performance, isolated supply chain.



Start Free, Deploy in Minutes

Getting started with Varnish Orca is simple:

- Docker image
- Helm Chart
- DEB packages
- RPM packages

Varnish Orca is configured through a *config.yaml* file that describes where to find the registries for each artifact type. It is available as a Free edition as well as a Premium version, which offers additional features such as private registry caching and persistent caching.

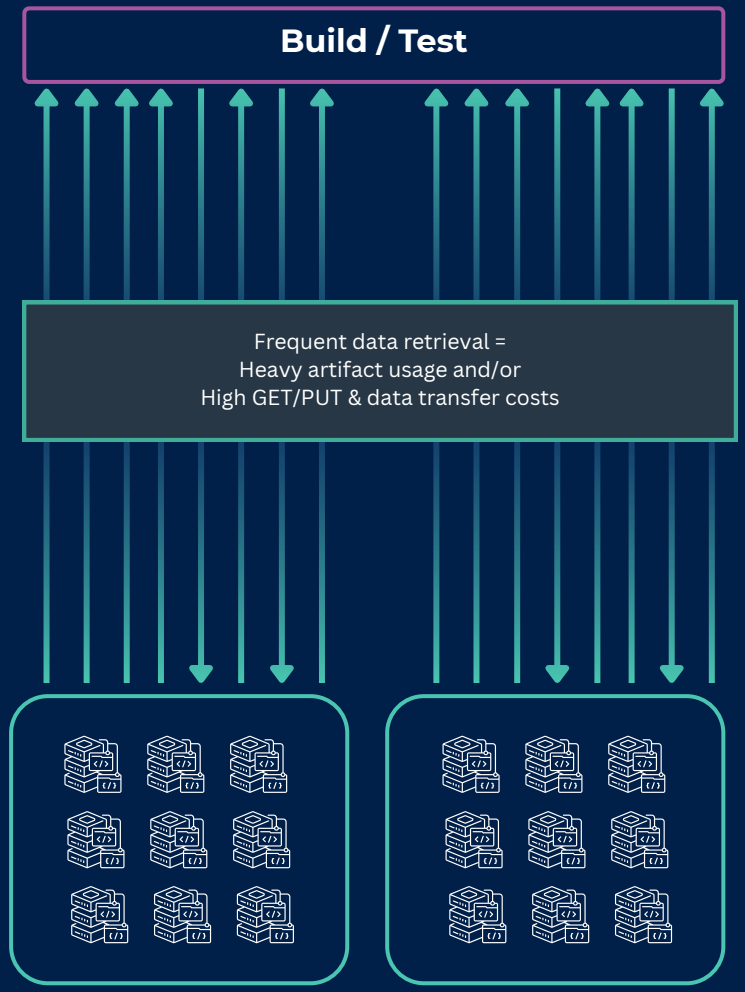
Get Started

Supported Ecosystem and Integration Points

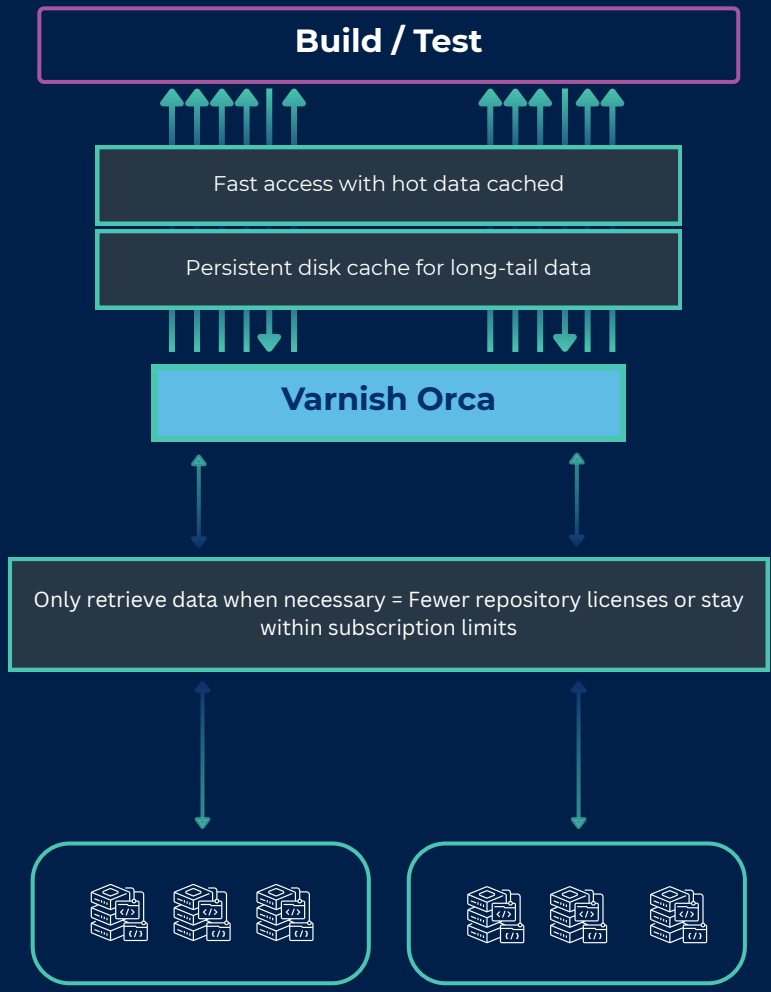
Category	Common Tools and Platforms	How Varnish Fits
Source Control	GitHub, GitLab, Bitbucket, Gitea	Caches repository data and packfiles for faster clones and fetches while preserving authentication and access controls.
Artifact Repositories / Universal Registries	JFrog Artifactory, Sonatype Nexus, CloudSmith	Internal caching tier for packages and OCI artifacts, reducing origin load and egress.
Platform Registries	AWS ECR, Azure Container Registry, Google Artifact Registry, AWS CodeArtifact	Accelerates delivery from cloud-native registries, reducing latency and shielding against rate limits across builds.
Service Registries & Package Sources	Docker Hub, npm, PyPI, Maven Central, Go proxy	Private, compliant internal mirrors that eliminate rate limiting from public sources.
Object Storage	Amazon S3, Google Cloud Storage, Azure Blob Storage, on-prem S3-compatible systems	High-throughput caching for binaries/models.
CI/CD Pipelines	Jenkins, GitHub Actions, GitLab CI/CD, Azure DevOps	Transparent acceleration for dependency/image fetches.
Developers & Consumers	On-site engineers, remote teams, and hybrid environments	Automatic speed-up. No changes to tools.

Varnish Orca integrates across the full DevOps ecosystem, from code to containers, without requiring workflow changes.

Without Varnish



With Varnish



Varnish delivers instant access to hot artifacts while persisting long-tail data efficiently on disk, reducing origin hits across the full access curve.

Enterprise Architecture & Observability

Vendor-Neutrality

Varnish Orca fronts any artifact or source platform, maintaining a consistent caching layer without vendor lock-in. Teams can:

- Move between tools while keeping a uniform delivery layer.
- Define cache and policy logic programmatically.
- Cache content at the network edge (same room, rack, or VPC segment as build systems).

Observability and Evidence

All transactions are logged with full context. Varnish provides rich observability through unified logs, metrics, and traces, all exportable in standardized formats (e.g. JSON, Prometheus, OpenTelemetry). These integrate with SIEM, monitoring, and audit systems to provide continuous, verifiable compliance evidence.

Pipeline Reliability and Resilience

Varnish Orca is engineered for pipeline uptime, insulating your build environment from upstream chaos:

- **Insulation from failures**: Orca can serve the last known good copy of an artifact even if the origin registry is down, preventing build failures due to service outages.
- **Rate limit protection**: By absorbing redundant requests, Orca shields backend registries from being overwhelmed, avoiding costly API throttling.
- **Vendor stability**: Fronting all platforms standardizes delivery, mitigating risks associated with regional latency spikes and sudden changes in vendor models.

Network Isolation and Transparent Proxying for CI/CD

Granting CI runners direct internet access to fetch public packages is a major security liability that increases your attack surface. Varnish Orca serves as a secure bridge, isolating build workers from public registries like GitHub Packages or Docker Hub.

- **Zero public egress**: CI runners communicate only with the internal Orca node. Orca manages all outbound, audited fetches, removing the need for build servers to have direct internet access.
- **Transparent redirection**: Eliminate the pain of reconfiguring package managers. By using a Layer 3/4 change, traffic is redirected to Orca instantly without modifying a single pipeline script.

By disconnecting runners from the internet and unifying artifact delivery through a single auditable point, Orca makes CI/CD more secure and operationally efficient.

Compliance and Assurance

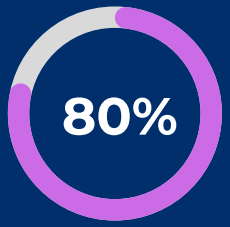
Varnish Software has been ISO 27001 certified since 2023 and is audited regularly by an accredited external audit firm. This certification supports compliance with emerging European frameworks including NIS2 and the DORA Act, reinforcing Varnish's position as a trusted infrastructure partner for regulated industries. Reviews by major global banks and technology partners have validated the robustness of our architecture and security controls. Varnish also maintains secure development and release practices aligned with industry frameworks.

Measurable Results

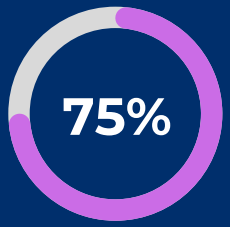
JFrog Artifactory

Cache artifacts and OCI blobs close to CI runners for fewer origin requests, faster dependency resolution, reduced egress. Real-world data from a Tier-1 Enterprise:

Lower CPU load on Artifactory



Fewer disk reads from Artifactory



Less outbound network traffic

Accelerate and Scale Artifact Workflows (Video)

Git over HTTP/S

Caches packfiles and range requests while maintaining authenticated access, for faster clones and fetches, fewer timeouts under load.

Faster clones with local caching (12s → 2s), with near-instant shallow clone fetches (2s → 0.5s)

Speeding up Git with HTTP Caching

Docker & OCI Registries

Cache image layers by digest, validates manifests, and coalesces concurrent pulls. Reliable, deterministic builds with no public registry exposure.

Cache hit rates for images

Faster job start times

Scaling Docker Image Delivery with Varnish

Business Impact Snapshot

Area	Baseline Pain	With Varnish Orca	12-Month Outcome
Developer Wait Time	10–20 min per cold build	30–70% faster fetch times	**+1–3 hrs/dev/week saved**
Origin Load	CPU/I/O saturation at peak	60–90% fewer origin hits	**Deferred hardware scale-out**
Egress Cost	Frequent external pulls	30–50% reduction	**$200k+ annual savings typical**
Release Reliability	Builds fail under load	Grace/keep modes for uptime	**Fewer failed pipelines**
Audit & Evidence	Fragmented logs	Centralized, structured trail	**Faster audits, less toil**

ROI Model

Savings = (Dev time saved * loaded rate) + (Avoided origin scale-out) + (Egress reduction) + (Incident reduction)

Payback (months) = (One-time + Year-1 cost) / (Savings / 12)

Example Inputs
Dev time saved: 1.5 hr/week × 300 devs × $110/hr ≈ $1.98M/yr
Egress reduction: $18k/mo = $216k/yr
Avoided hardware: $250k deferred 12–18 months
Incident reduction: 6 fewer P1s × $20k = $120k

Indicative Outcome: Payback within 6–9 months.

Case Study: Large Global Financial Institution

Challenge
A leading multinational bank with globally distributed engineering teams was experiencing high latency and infrastructure strain during CI/CD operations. Artifact and container fetches between regions averaged 12 seconds, and legacy Squid proxies were unable to scale with growing concurrency and data volumes.

Solution
Varnish was deployed across regional sites, creating local cache tiers for DevOps traffic. The deployment required no workflow changes and operated entirely within the institution's internal compliance perimeter.

Results

- **50% lower CPU load on repository servers**
- **75% traffic offload from backends**
- **80% fewer disk reads**
- **Faster, more consistent, and secure workflows across global teams**

Impact

- **Developer Experience**. Higher throughput, lower wait times.
- **Platform Reliability**. Governed, reproducible, and compliant artifact delivery across regions.
- **Business Efficiency**. Significant infrastructure cost savings through reduced load and egress.

Conclusion and Next Steps

With Varnish Orca, performance and security reinforce each other. By embedding a secure caching and delivery layer inside your infrastructure, you remove latency without removing control. For financial, healthcare, and critical-infrastructure teams, it means:

- Faster, more reliable builds
- Lower infrastructure and network overhead
- Proactive supply chain defense and network segmentation for CI/CD

ISO 27001 Certified | Trusted by Global Financial Institutions | OpenTelemetry-Ready

Proof of Concept

Validate quickly using **Varnish Orca Free Edition**, the fully functional Virtual Registry Manager for small-scale deployments.

Getting started with Varnish Orca is very easy, with a Docker image, a Helm Chart, and Linux operating system DEB packages and RPM packages. Varnish Orca is configured through a *config.yaml* file that describes where to find the registries for each artifact type.

Varnish Orca Premium unlocks advanced features:

- Persistent Cache
- Private Registry Cache
- Programmability
- Git Mirror
- OpenTelemetry Tracing
- Enterprise Support

Start free, scale when ready.

Download the free edition or request a premium trial license.

```
varnish:
  https:
    - port: 443
acme:
  email: orca@example.com
  domains:
    - orca.example.com
virtual_registry:
  registries:
    - name: dockerhub
      default: true
      remotes:
        - url: https://docker.io
        - url: https://mirror.gcr.io
    - name: quay
      remotes:
        - url: https://quay.io
    - name: ghcr
      remotes:
        - url: https://ghcr.io
    - name: k8s
      remotes:
        - url: https://registry.k8s.io
    - name: npmjs
      remotes:
        - url: https://registry.npmjs.org
    - name: go
      remotes:
        - url: https://proxy.golang.org
    - name: github
      remotes:
        - url: https://github.com
    - name: gitlab
      remotes:
        - url: https://gitlab.com
```

About Varnish Software

Varnish Software is the global leader in high-performance content and data delivery. Our technology powers the world's most demanding digital platforms, from streaming and content distribution to API acceleration, edge computing, and DevOps artifact delivery.

Built on the proven Varnish Cache engine, Varnish Enterprise delivers unrivaled speed, scalability, and control. This foundation underpins solutions like Varnish Orca, which is specifically engineered to accelerate CI/CD pipelines. Orca enables organizations to operate secure, compliant, and high-performance delivery infrastructures across any environment—on-premises, in private clouds, or across sovereign and public cloud regions.

Trusted by leading enterprises in finance, media, telecommunications, healthcare, and critical infrastructure, Varnish Software provides the programmable performance layer that underpins modern digital experiences and internal delivery systems alike.

Headquartered in Stockholm, Sweden, with offices and partners worldwide, Varnish Software continues to advance the state of caching, edge intelligence, and secure software delivery for a global customer base.

VARNISH SOFTWARE

Email :
info@varnish-software.com

Website :
www.varnish-software.com