CleanStart

DATASHEET

Clean Libraries

Verified language dependencies, decided before install

Clean Libraries controls which open source components are allowed into your codebase at the moment they're requested, whether the request comes from a developer or a coding agent. Every request returns one signed verdict. Each verdict is backed by components rebuilt from pinned source and checked for origin, behaviour, and equivalence to upstream. Downstream scanning then confirms a clean result instead of discovering a bad one.

The problem

AI coding agents resolve and install entire dependency trees in seconds, faster than any review process can evaluate them. One deliberate dependency pulls in hundreds of others that nobody chose or reviewed.

The first check runs at build time, after the component is already merged. So every finding turns into rework at the most expensive moment.

The verdict model

Each request returns one verdict. It's signed, machine-readable, and recorded with its reason and the policy version that produced it.

CLEAN : Admit

Origin verified, behaviour verified, equivalence to upstream shown.Trusted coordinates returned, and installation proceeds.

FLAGGED : Review

An anomaly surfaced: new capability, a maintainer signal, or a provenance weakness. Policy decides whether it proceeds.

BLOCKED : Refuse

Fails policy. Installation is refused with the reason attached, and a governed alternative is offered where one exists.

INSUFFICIENT_DATA : Queue

Verification isn't finished yet. The component is queued for build on demand rather than passed through as if it were safe.

What gets verified

Origin	Components are built from source pinned by commit hash, in a hardened, isolated environment.	SLSA build provenance, signed attestations, and an SBOM in SPDX and CycloneDX.
Behaviour	Each component runs in a sandbox at install and import. Its capabilities are diffed against the prior version, and maintainer signals are analyzed.	Behavioural report, a capability change record, and maintainer risk signals.
Equivalence	An independent rebuild and binary diff confirm the component matches upstream, and the upstream test suite runs against the rebuilt artifact.	Reproducibility record, test-suite results, and any reported deviation from upstream.

Integration surfaces

Surface	Where It Applies	Behaviour
VS Code extension	As the dependency is written	Inline verdict before the import becomes a commit
CLI	Admin, policy governance and package management	Generates per-language config (npm, pip, cargo and more), enforces policy, and drives auditing and governance: allow, flag or deny per verdict
Multiple Language SDKs	In-code use across Go, Python, Rust and JavaScript	Call CleanLibraries from application code, scripts and internal tooling. Identical verdict to CI, so local and pipeline never disagree
MCP server	Any MCP- compatible coding agent	Called before install; refuses hallucinated and squatted names. Verified with Claude Code, Cursor and Gemini CLI
CI/CD enforcement	Build and merge gates	Hard gate on policy; failures carry the reason and a governed alternative
Catalog delivery	Existing repository infrastructure	Verified artifacts across eight language ecosystems

Deployment options

Managed
CleanStart runs the pipeline and catalog. You consume verdicts and artifacts.

Self-hosted
The full stack runs inside your own infrastructure and network boundary.

Air-gapped
Complete function with no outbound connectivity, via controlled sync. A first class mode, not degraded

Sovereign
Data residency stays within your jurisdiction, with no export of dependency metadata.

Specifications

Catalog
Eight ecosystems covering the languages most common in enterprise development, rebuilt from upstream source pinned by commit hash.

Unverified and blocked components
Unverified components are queued for build on demand. Blocked components come with a governed fallback. CVE remediation targets follow the service agreement.

Evidence set
SLSA build provenance with signed attestations. SBOMs in SPDX and CycloneDX, generated at build time. OpenVEX statements for triaged findings. Standard formats only, exportable on exit.

Agent integration
A standard MCP server connects any compatible agent by configuration. Native plugins for GitHub Copilot and AWS Kiro are in development.

Compliance alignment

EU Cyber Resilience Act.
Reporting starts 11 Sep 2026, with a 24-hour warning and 72-hour notification. The full regime begins 11 Dec 2027. Provenance and SBOMs are generated continuously.

Agentic AI guidance.
Supports the approved-component-source model in the CISA / NSA advisory of 1 May 2026, with agent decisions linked to an identity.

FedRAMP / CMMC / FIPS.
Cryptographic module validation carries across the image and library boundary under one chain.

Book a demo

To evaluate your own dependency tree