CERBI TECHNICAL BRIEF

CerbiShield Platform Overview 2026

Scan. Govern. Enforce. Prove. One telemetry-governance model across source, runtime, and OTLP.

Current platform overview - August 13, 2026

Purpose

Cerbi is a logging and telemetry governance platform for teams that need a repeatable answer to four questions: what may be emitted, which policy applies, where that policy is enforced, and what evidence proves the control was active.

ACQUISITION ENTRY **Cerbi Scanner**	CONTROL PLANE **CerbiShield**	IN-PROCESS ENFORCEMENT **CerbiStream**
CENTRAL ENFORCEMENT **Cerbi Gateway**	DEPLOYMENT MODEL **Customer Azure tenant**	OBSERVABILITY BACKEND **Kept in place**

01

The platform model: Scan → Govern → Enforce → Prove

SCAN

Find logging and policy issues in source and CI before runtime.

GOVERN

Define profiles, targets, validation, deployment state, and history in CerbiShield.

ENFORCE

Use CerbiStream in-process, Gateway on the OTLP path, or both.

PROVE

Review violations, evidence, reporting, audit context, usage, and health.

Cerbi is not an observability backend replacement. Existing logging frameworks, OpenTelemetry instrumentation, dashboards, alerts, search, analytics, and retention remain useful downstream.

02

Cerbi Scanner: the lowest-friction starting point

A new governance platform is difficult to sell if the first conversation begins with a production deployment. Scanner turns the first step into evidence: run it against one repository, surface concrete logging issues, and decide whether runtime enforcement is justified.

Why Scanner matters commercially

It lets a buyer see their own problem before being asked to change a telemetry path. That makes Scanner the natural acquisition product even when CerbiShield and Gateway are the higher-value platform components.

03

CerbiShield: control + evidence plane

Overview	Governance posture, workload metrics, trend signals, attention items, and platform health.
Rules	Templates, composition/editing, compatibility checks, validation, and targeted deployment.
Deployments	Targets, environments, rollout state, status, operator context, and deployment detail.
Violations	Severity/rule trends with workload, field, profile, and time-window drill-down.
Validation	Representative payload testing before policy promotion.
Evidence + reporting	Governance evidence, coverage/score trends, operational reporting, and exports.
Audit	Actor, action, resource, version, target, status, and timestamp context.
Health + connectivity	Service health, latency, queues, databases, routing activity, errors, and dependency reachability.

The Dashboard should be demonstrated as one policy moving from definition to validation to targeting/deployment to enforcement to evidence - not as a collection of unrelated charts.

04

Two enforcement boundaries

CerbiStream - in process

Use when policy should execute before the first network hop. Strong fit for selected high-risk applications. Requires application integration; keeps the existing backend.

Cerbi Gateway - OTLP boundary

Use when workloads already emit OpenTelemetry and platform teams need a central adoption path. No Cerbi SDK required for those workloads; keeps the existing backend.

The enforcement point changes. The governance model - policy, targeting, validation, deployment state, evidence, and operational review - stays consistent.

05

Gateway security and reliability

- ✓ Customer-hosted Azure Container App.
- ✓ HTTPS OTLP/HTTP ingress restricted to a customer-approved source CIDR.
- ✓ Required signed Runtime Policy Bundle verification.
- ✓ Customer-local non-exportable P-256 Azure Key Vault signing key.
- ✓ Non-public customer-local runtime-policy storage.
- ✓ TLS downstream OTLP transport.
- ✓ Bounded memory, batch processing, queued export, and retries.
- ✓ Liveness/readiness probes and horizontal scaling.

Gateway remains labeled controlled preview until the Marketplace lifecycle and exact-candidate Dashboard acceptance gates are complete. The website should change this status only when those gates actually close.

06

Who the platform is for

Platform engineering	One policy model across application and OTLP enforcement without replacing the observability estate.
Security engineering	Reduce sensitive telemetry exposure and keep signed runtime policy inside the customer trust boundary.
DevSecOps	Move checks into source/CI, validate before rollout, and preserve controlled deployment history.
Observability teams	Govern before indexing while keeping the tools, dashboards, alerts, and retention workflows already operated.
Risk / compliance	Use bounded evidence and reporting to support control reviews without pretending product installation equals certification.

07

A practical adoption sequence

01	Scan one repository and review the actual telemetry issues found.
02	Define one governance profile that addresses a real requirement.
03	Choose the enforcement boundary: CerbiStream for an application or Gateway for an OTLP workload.
04	Run the change in non-production and validate representative telemetry.
05	Review violations, deployment state, evidence, health, and measured overhead in CerbiShield.
06	Expand only after the team can explain how the control works operationally.

Commercial principle

Start with evidence, not a platform migration. Scanner or one representative OTLP workload should prove enough value to justify the next step.