

Why This Design

A single edge host that terminates public TLS, inspects every request, and forwards only what passes inspection. This page is the reasoning behind that shape and the tool choice

Why terminate everything at one host

The alternative to a single edge is exposing services directly, each with its own public certificate and its own internet-facing surface. That multiplies the attack surface by the number of services and scatters the security configuration across all of them. Every app becomes its own front door, and every front door is one you have to lock individually.

Funnelling all public traffic through one host inverts that. Backends get no public IP addresses at all; they are reachable only from the edge. The internet sees exactly one machine. That machine is the only thing that has to be hardened against direct attack, the only place public certificates live, and the only place request inspection has to run. Concentrating the edge concentrates the work of defending it, which for a solo operator is the difference between a defensible position and an unmanageable one.

Why open-appsec on nginx

The edge runs nginx as the reverse proxy, with [open-appsec](#) attached as the inspection layer. The reasons:

- **Open-source and self-hostable.** Consistent with the rest of the environment, it runs on infrastructure under my control with no per-request cost and no traffic routed through a third-party scrubbing service. For a privacy-focused setup, sending every request through someone else's cloud WAF would undercut the point.
- **Machine-learning-based, not only signatures.** Traditional WAFs match requests against signatures of known attacks. Open-appsec adds a model that scores requests by how anomalous they look, which catches variations that no signature covers and reduces the endless rule-tuning that signature-only WAFs demand.
- **Runs as an nginx module.** It attaches to the nginx already doing the proxying rather than being a separate appliance in the path. One host, one request flow, one place to

apply controls.

- **Right-sized.** It provides real application-layer protection, the OWASP attack categories, anti-bot, schema validation, without the cost and operational weight of a commercial WAF appliance built for enterprise traffic volumes.

The tradeoffs

A single edge is a single point of failure and a single chokepoint. Every request in the environment passes through this one host, which means its performance is everyone's performance, and its uptime is everyone's uptime. In a small ecosystem this is a reasonable trade. The simplicity of one well-understood edge outweighs the availability cost, and the traffic volume is nowhere near enough to strain a single host. It must be said, though, that this does not scale. A larger or more critical environment needs the WAF running as a redundant, load-balanced tier so that no single host failing takes the environment down, and so the edge can be updated without an outage window. The design in this book is deliberately right-sized for a small ecosystem, and deliberately unsuitable for anything larger.

Open-appsec's learning model needs time and tuning. The machine-learning approach is a strength, but it is not zero-configuration. The model has to observe real traffic before it can reliably tell normal from malicious, and requires some manual review and policy tuning before enabling blocking capabilities. This is a real operational cost, covered in detail on the inspection page: the learn phase is mandatory, not optional.

Pinning nginx creates a maintenance burden. Open-appsec attaches to specific nginx versions, which means nginx cannot be allowed to upgrade freely. It has to be held at a compatible version and updated deliberately in-step with open-appsec's compatibility. That is a small, periodic chore traded for the inspection capability.

Adapt this for...

Any environment consolidating public ingress behind one inspecting edge. The tool specifics are nginx and open-appsec, but the decisions generalize: put all public services behind one hardened proxy, terminate TLS there, inspect there, and give backends no other way in. The one decision that must change with scale is the number of edge hosts. One can be right for a small ecosystem, and a high-availability tier is required for anything larger.

Revision #1

Created 2026-07-27 16:53:22 UTC by Chris Landis

Updated 2026-07-27 17:06:39 UTC by Chris Landis