CASE STUDY

Engineering Zero Trust from Identity to Access

Sector: Cybersecurity / Zero Trust / Enterprise Infrastructure

7K+
Customer Organisations

50M+
API Requests Daily

CASE BACKGROUND

A contractor joins an organisation for three months to troubleshoot an internal application. They need access to that application, but there is no reason for them to enter the rest of the company's network.

For that access to happen securely, several things need to work together.

The platform needs to recognise the contractor through the organisation's existing identity provider. It needs to verify the device they are connecting from. It needs to determine what application they are allowed to access, assess whether the request satisfies the organisation's security policies, establish a secure connection to the resource, and continue enforcing those controls for as long as access is required.

When the engagement ends, the contractor's access needs to disappear with it.

This is the operating model behind Zero Trust Network Access (ZTNA), and it was the foundation of a security platform we have worked on for roughly a decade.

> *Our involvement began when the company behind the platform was still at an early stage. Over the years, we worked across the security and engineering layers required to turn Zero Trust from an access principle into a working product: identity integrations, device authentication, identity-aware access, policy enforcement, secure tunnels, DNS protection, automated user provisioning and customer onboarding.*

As the product matured, those capabilities also had to operate at a very different scale. Today, the platform **supports more than 7K customer organisations** and **processes over 50 million API requests every day** across a global infrastructure.

The engineering challenge therefore evolved on two fronts: expanding what the security platform could do while ensuring those capabilities continued to work reliably as the business grew.

THE PROBLEM

Establishing Trust Before Granting Access

Identity was one of the first pieces that had to work across very different enterprise environments.

Every organisation already had its own way of managing users. One might use Google, another Microsoft Azure AD, another Okta or a different identity provider altogether. Asking customers to create and manage a separate identity environment for the security platform would introduce another layer of administration and another set of credentials for employees.

Support for SAML and OIDC allowed the platform to integrate with different identity providers while preserving the organisation's existing single sign-on experience.

But identity alone could not establish sufficient trust.

A valid employee credential being used from an unknown or compromised device still creates risk. Mutual TLS (mTLS) was therefore used to authenticate the device itself through trusted digital certificates. Access could depend on both who was making the request and whether it originated from an approved machine.

From there, the decision could become more granular.

The platform incorporated trust scores based on factors such as device health, location and behaviour. Instead of applying the same access decision to every authenticated user, policies could respond to the context surrounding an individual request.

A familiar user on a compliant corporate device could therefore be treated differently from the same credentials appearing on an unfamiliar device or from an unexpected location.

This created the foundation for contextual access rather than relying on authentication as a one-time gateway into the network.

Enforcing Access Without Extending Trust to the Network

Once identity and device trust had been established, the next requirement was controlling what the user could actually reach.

Traditional remote-access models often placed considerable importance on getting a user securely onto the corporate network. Once inside that boundary, however, the user could potentially have visibility into resources beyond the application they actually needed.

The platform introduced identity-aware proxies in front of protected applications so that access could be evaluated at the resource level.

Requests had to satisfy the applicable identity and security policies before reaching the application. This allowed an organisation to give a contractor access to one internal application without opening the surrounding network to them.

The secure connection itself also mattered.

WireGuard tunnels were configured to provide encrypted connectivity between remote users and enterprise resources. Its lightweight architecture provided a modern alternative to traditional VPN tunnelling while protecting data moving between the device and the destination.

Zero Trust controls also extended beyond private applications.

Through DNS filtering, requests to known malicious, phishing or restricted destinations could be intercepted before a connection was established. Organisations could apply their own policies around permitted or prohibited categories of internet traffic, adding another enforcement layer to the broader access environment.

Together, these capabilities allowed access to be controlled from the point at which a user authenticated through to the resource or destination they ultimately reached.

THE APPROACH

Making Zero Trust Operationally Manageable

Security controls become considerably harder to operate when every customer has to be configured manually.

Consider something as routine as an employee joining or leaving an organisation.

Creating an account manually across systems introduces effort. Failing to remove that account when the employee leaves introduces risk.

> *The platform therefore incorporated SCIM-based provisioning, allowing user information to remain synchronised with an organisation's identity environment. Users could be created, updated or removed as their status changed without relying on separate manual administration within the security platform.*

The same principle applied when organisations themselves joined the platform.

Migrating from an existing VPN environment to Zero Trust could involve provisioning infrastructure, connecting identity providers and establishing the relevant access policies. Automating significant parts of that onboarding process reduced the amount of manual configuration required to bring new customers onto the platform.

This became increasingly important as the commercial footprint changed.

An operating model built around manually configuring a relatively small number of large enterprises could not remain unchanged when the platform began supporting thousands of organisations.

Automation therefore became part of the security architecture itself. It allowed the platform to preserve granular customer-specific controls without requiring operational effort to rise proportionally with every new customer and user.

When the Customer Base Moved Beyond 7K Organisations

The growth of the platform introduced another architectural problem.

Certain services in its earlier architecture had been deployed at an organisation level. That model was manageable across a smaller enterprise customer base. ***Once the number of organisations moved beyond 7K, every customer-specific service, connection, configuration and policy environment became part of a much larger infrastructure footprint.***

The customers themselves also had different requirements.

Some large enterprises needed dedicated deployments. Some required infrastructure within particular cloud environments. Thousands of smaller organisations could operate efficiently within shared infrastructure.

The architecture had to accommodate these models without turning every deployment into an individual engineering exercise.

> ***The platform's core, built using Go and MySQL, operates across GCP and AWS, with customers served through more than a dozen global points of presence.***

This distributed model allowed infrastructure to be aligned with customer and regional requirements while the underlying Zero Trust capabilities remained consistent across environments.

As usage patterns and customer distribution changed, existing services were revisited, infrastructure models evolved, and some mature components were completely rewritten. These changes had to preserve years of existing behaviour while the live platform continued serving customers, and the rewrites were completed without reported production issues.

THE OUTCOME

Making Millions of Security Decisions
Without Slowing the User Down

The platform now processes more than **50 million API requests every day** and over 350 million in a week.

Average throughput is around 600 requests per second, *increasing* to approximately **850 requests per second** at peak.

Those numbers matter because access control sits directly in the user's path.

Identity checks, policy evaluation and other security decisions cannot add enough latency to make employees feel that security is slowing down their work.

The platform currently maintains a median response time of around *30 milliseconds*, while the vast majority of requests complete within approximately 200 milliseconds. Its error rate remained below 0.2% during the supplied recent production period.

At the current pace, the platform is handling more than 20 billion requests annually.

Maintaining that performance meant engineering the security capabilities and the architecture around them together. Identity integration, policy enforcement or secure connectivity may each solve a different security requirement, but the user ultimately experiences them as one access journey.

KEY TAKEAWAY

What Changed in Practice

Over roughly a decade, the platform evolved from an early-stage Zero Trust product into a security environment supporting thousands of organisations across multiple regions and cloud environments.

Our role evolved with it.

We worked across the capabilities required to establish and enforce Zero Trust in practice, from SAML and OIDC integrations, mTLS device authentication and identity-aware proxies to trust-based policy enforcement, WireGuard connectivity, DNS filtering, SCIM provisioning and automated onboarding.

As adoption increased, that work extended into the architecture underneath those capabilities: scaling Go-based services, supporting different deployment models across GCP and AWS, redesigning mature components and helping the platform sustain tens of millions of daily API requests with low latency.

The outcome is therefore larger than the platform's ability to support 7K+ organisations.

> *It is the ability to carry the same security principle through identity, device, policy, connectivity and lifecycle management, and continue doing so as the number of customers, environments and access decisions grows.*

That is what this project really solved.

A centralised, governed, searchable system – built to assist rather than obstruct.

www.joshsoftware.com | contact@joshsoftware.com | +91 7887889902