CLOUDFLARE

SOLUTION BRIEF

An overview of Internet-native architecture

Infrastructure capabilities that simplify Zero Trust security adoption and lead to great end-user experiences

There's a contradiction lurking at the heart of Zero Trust security. Users, corporate applications, and data face a strong gravitational pull to the public Internet. Yet many enterprises are justifiably wary of exposing their network traffic to the public Internet's insecurity and unreliability.

The answer to this contradiction? Organizations need a network architecture that provides the Internet's interconnectivity and adaptability without compromising on security, data protection and privacy, or reliability. We call this **Internet-native architecture.**

Read on to understand what it looks like — and how its features support Zero Trust adoption and better end-user experiences.

Characteristics of Internet-native architecture

At the highest level, we define Interent-native architecture as a broadly dispersed network of servers which can deliver corporate network functions for a hybrid (i.e. distributed and heterogeneous) collection of users, endpoints, applications, offices, and data centers. Such architecture has:

	Meaning	Example benefit
Global reach and interconnectivity	Direct 'any-to-any' connectivity between offices, users, apps, ISPs, Internet exchange points, cloud instances, etc.	✓ Low latency between user in New York and server in Tokyo. ✓ No limits on which type of application can connect.
Fundamentally interwoven security and networking	All services live on a single homogenous network fabric, requiring no backhauling.	✓ Both forward and reverse proxy services – SWG, ZTNA, CASB, RBI, DLP and more – live on every server.
'Composability' and future-proofing	Updatable with new features as networking needs change over time — rather than constrained by prior architectural decisions.	✓ Compatibility with an as-yet-undeveloped security service. ✓ Adoption of a future Internet protocol in weeks, not years.

What do these high-level characteristics look like at the feature level, and how do those features simplify Zero Trust adoption?

Global reach and interconnectivity

- **A large number of network locations**, which live all over the world. This supports Zero Trust by putting policy enforcement, proxying, and packet transport close to users anywhere, improving reliability.
- **Connectivity to many ISP and interconnection peers.** This supports Zero Trust adoption by giving user requests from anywhere a shorter journey to any resource — improving and simplifying the user experience.
- **Dual-stack or IPv6-only networking functionality.** This supports Zero Trust by letting users connect to resources over any ISP connection.
- **Connectivity with any flavor of existing network infrastructure.** This supports Zero Trust by letting you keep certain legacy hardware without impeding other adoption efforts. It also lets you comply with data privacy regulations that may mandate certain types of infrastructure for data storage, while still using the network for other Zero Trust capabilities.
- **Data and control planes encrypted between endpoints and the network.** This supports Zero Trust adoption by keeping user requests secure from snooping by an on-path attack.

Fundamentally interwoven security and networking

- **Security and performance services run on every server in every network location,** rather than certain services requiring specialized infrastructure. In this way, the architecture is both a global policy enforcement platform and a global traffic transport network — rather than just one or the other. This supports Zero Trust by reducing traffic backhauling and capacity constraints for end-users. It also simplifies onboarding of new Zero Trust services — since IT teams do not have to connect to disparate infrastructure every time or tradeoff performance for security.
- **Anycast functionality**, in which any network location can respond to user requests. This supports Zero Trust by reducing routing latency, improving the end-user experience — and by reducing routing maintenance, improving the administrative experience.

- **True cloud architecture** using microservices or serverless functions, rather than hardware-derived virtual machines imitating cloud functionality. In the latter, an entire service (or multiple services) run in one virtual machine instance, so when one function of one service needs to scale, an entire new instance needs to be spun up. Avoiding this supports Zero Trust by avoiding associated traffic capacity limitations or high costs, and by letting developers build, test, and update features more quickly.
- **Traffic travels on a virtual private backbone** which can both encrypt traffic and route it over the fastest network paths. This supports Zero Trust by keeping user requests secure, giving IT teams easier visibility and logging of requests, and protecting users from congestion on the public Internet.

Composability and future-proofing

- **Programmability,** meaning it's straightforward to add customized routing rules, access policies, and code — ideally using a single development platform across every service. This supports Zero Trust by enabling unique needs and exceptions that inevitably arise during the adoption journey.
- **Homogeneous network infrastructure,** i.e using the same underlying servers everywhere, using the same single control and management planes, and designing services that do not require specialized hardware. This means future Zero Trust services will be able to run everywhere, and be integrated cleanly into the network.

Next steps

To learn more about what Internet-native architecture looks like in practice check out our reference architecture. You can also request an architectural workshop to discuss your specific organization's needs:

- **Reference architecture:** cfl.re/architecture-reference
- **Architecture workshop:** cloudflare.com/products/zero-trust/plans/enterprise/