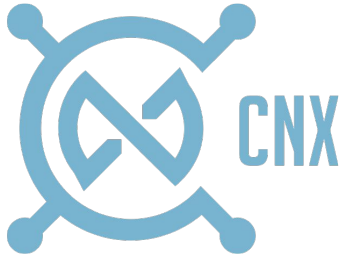
API Gateway

A unconventional IXP service



Mike Gaertner CNX @ APIX #32 2025

What is the problem?

- CNX has been working with ISP and Data centers for years, and we have the largest eyeball network in country
- but we made no headway with service providers (like banks, wallets etc.)
- we started talking to service providers about why they do not peer locally
- Rather (un)surprising answer:

Security

Financial service providers said

- There is no local WAF provider in country
- Our risk manager said we must have protection for our API
- Yes we understand 99% our our traffic is local

We talked to the regulator

CNX: Having banking traffic routed outside the country for digital payments is not ideal?

Regulator: Yes we agree

Regulator: We do not think we can do anything about it

We went back to the Banks

Q: What do you actually need from your WAF solution?
Let's focus on the mobile app — the #1 tool for banks to interact with customers.
The website is almost exclusively B2B and represents barely 1% of traffic.

After a few meetings we got it down to 3 core issues

- Exposure of Origin
- DDoS against Origin
- Identifying valid sessions

CNX

Well … can we build this?

What capabilities do we have to mitigate against DDoS?

- We are the IXP — we got more capacity than anyone else
- We got co-locations in data centers
- We have our own IP space and we could deploy a anycast network

How to address origin and client side identification ?

- Identify your client → Mutual TLS with the mobile app
- Hide your origin → Local reverse proxy at the IXP
- Absorb the DDoS → Anycasted reverse proxy in multiple DCs

CNX

Let's think about it

How to structure this setup :

- **Reverse Proxy** — good old *HAProxy* at the IXP
- **Incoming TLS handshake** from mobile app
- **No certificate?** → Close connection
- **Certificate present** → Negotiate TLS, check **client cert signer**
- **Signer invalid?** → Close connection
- **Valid signer** → Establish new TLS session with Bank API
- **Forward traffic** to bank's origin

Result:

❌ All non-app traffic dropped before hitting the bank

✅ **Identify your client:** *Solved*

✅ **Hide your origin:** *Solved*

How Much Capacity Do We Need?

How Much Capacity Do We Need?

- **Population:** 17M total
- **Online payment users:** ~5M (major centers)
- **Usage assumption:** 4 transactions/day/user
- **Transaction size:** ~300 bytes (TLS encrypted)

Daily volume (transaction only):

- 5M × 4 = **20M transactions/day**
- 20M × 300B ≈ **6 GB/day**

Full session overhead (~10×):

- Balance check, account selection, vendor lookup, QR processing
- 6 GB × 10 ≈ **60 GB/day**

Traffic rate:

- Avg ≈ **7 Mbps**
- Peak (×20 burst) ≈ **140 Mbps**

With full session load, **<150 Mbps** can serve the *entire* country's mobile banking API traffic

CNX

Worst-Case Domestic DDoS (Sizing Targets)

Attack mix assumed

- L4 SYN/ACK flood + L7 HTTPS request flood
- Domestic bot pool only (mobile + fixed CPE/PC)

Botnet sizing (domestic only)

- Mobile: ~1% of 17M
  ≈ **170k** devices × **0.5 Mbps** each ⇒ **~85 Gbps**
- Fixed CPE/PC: ~0.2% of ~3.5M households
  ≈ **7k × 10 Mbps ⇒ ~70 Gbps**
- Compromised servers on local hosting
  ⇒ **~20 Gbps** buffer
- **Credible worst case: ~175 Gbps**

Packet-/request-level view

- L4 small-packet flood @ 64B:

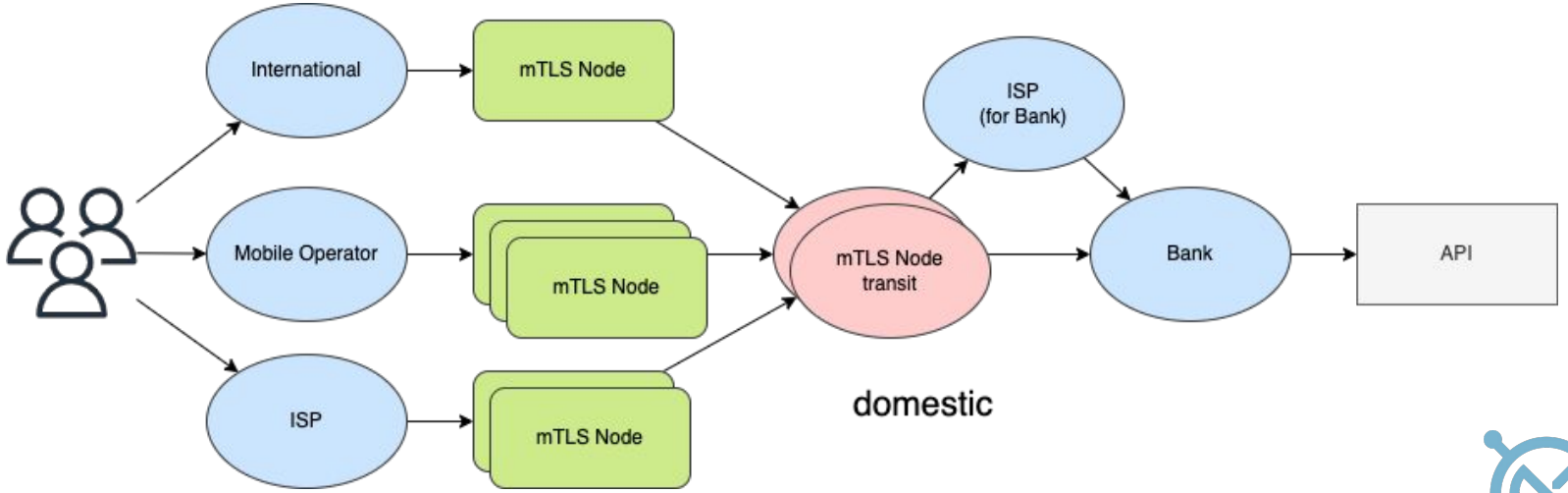
  1 Gbps ≈ 1.488 Mpps
  175 Gbps ≈ 260 Mpps

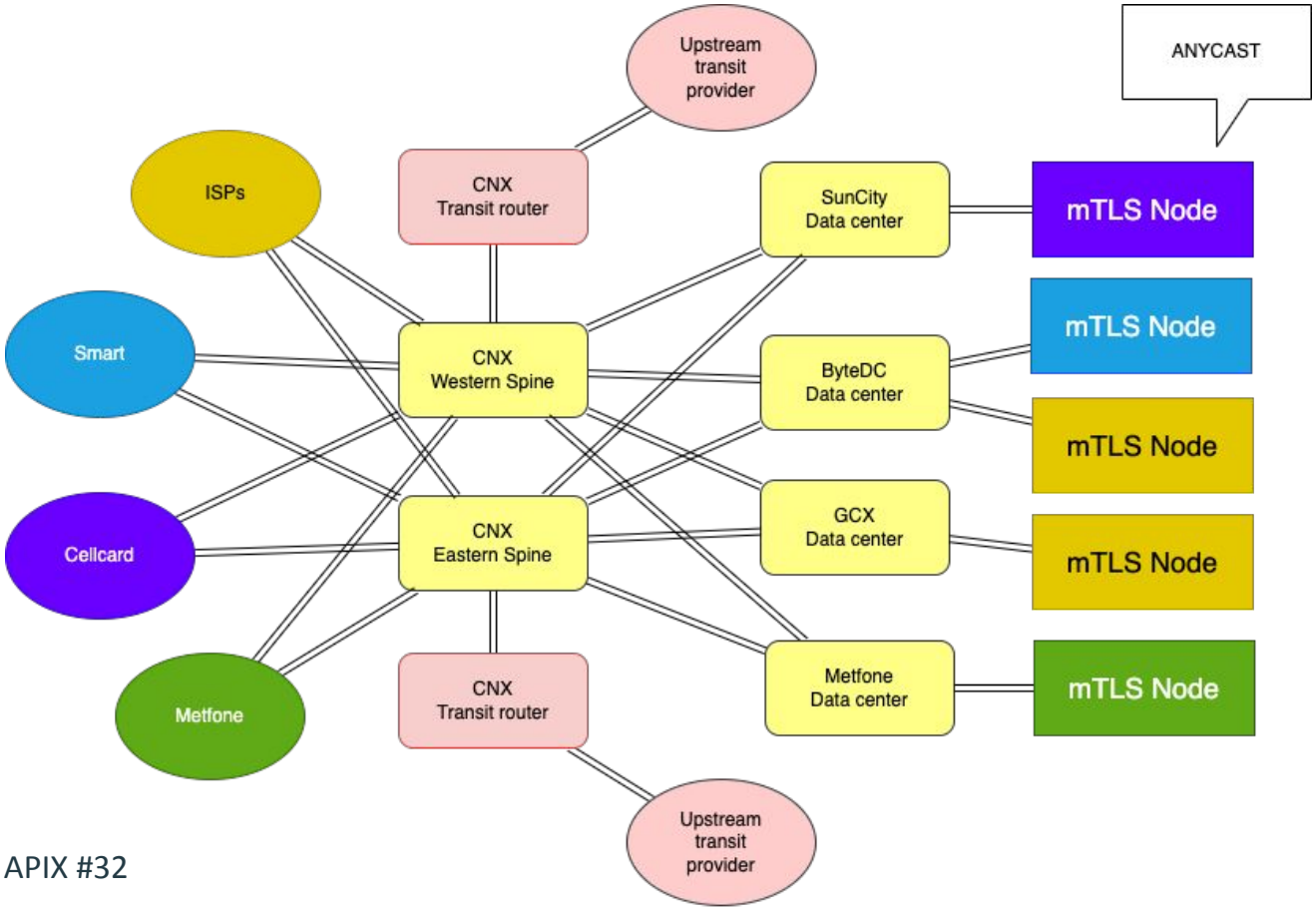
- L7 HTTPS GET flood (1 KB req effective over TLS):

  100 Gbps ≈ 10–12.5 M req/s
  175 Gbps ≈ 17–22 M req/s

Design

Logical data path

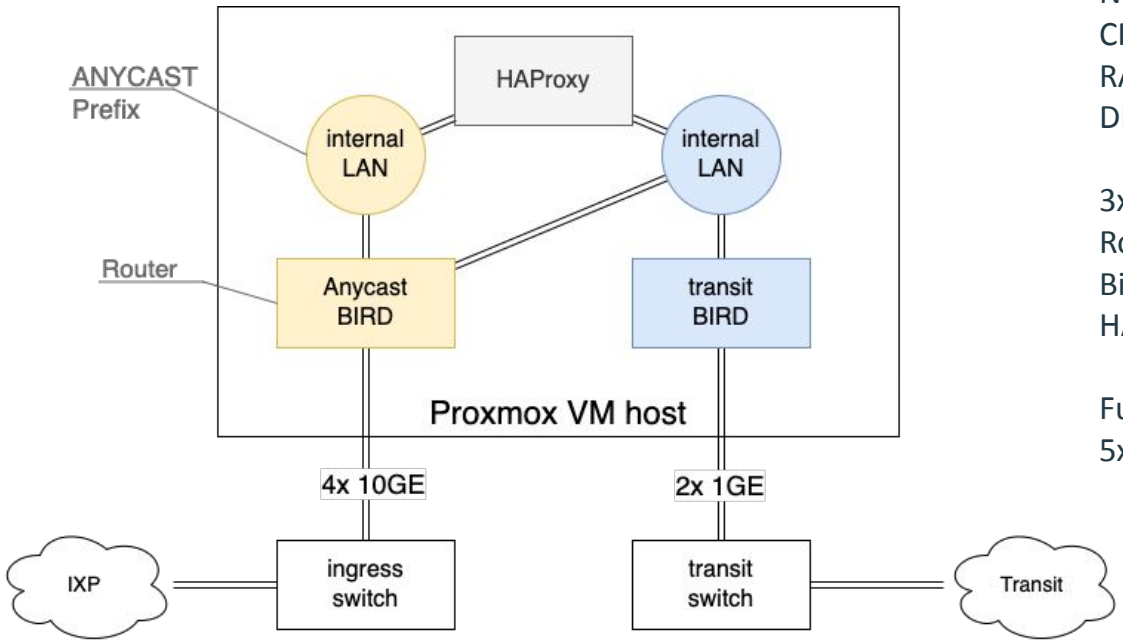




Nodes are “paired” with providers via targeted pre-pending

Upstream announcement can be cut at any time in case of in-coming DDoS, and is rate limited at the transit router

System node design (mTLS)



DELL R640

NIC: 4x 10GE
NIC: 4x 1GE
CPU: Intel Gold 6246
RAM: 128GB
DISK: 2x SSD 2TB

3x VM
Rocket Linux 9.0
Bird 3.1.2
HAPROXY: 2.4

Full system capacity
5x 40GE = **200 gbps**

Design notes on ingress

Ingress Router (VM, BIRD on Rocky 9.0)

- Disable conntrack on (anycast) IPs (notrack) to keep floods from exhausting state tables
- Apply early L4 stateless filtering at **netdev/ingress** hook (nftables)
- Per-source rate limits (meter/limit) for SYN, UDP, ICMP
- SYN proxy on TLS ports to complete handshake before HAProxy
- Monitor PPS/Gbps counters separately from bytes — floods are packet-rate bound
- Log counters via nft monitor or flow export, not per-packet logging during events

Design nodes on proxy

HAProxy (Rocky 9.0)

- Terminate TLS and enforce mutual TLS — reject invalid/missing client certs before app logic
- Restrict accepted CAs to bank-controlled signers
- Enable connection reuse / session resumption to reduce TLS handshake load
- Drop idle connections aggressively (low timeout client/timeout server)
- Monitor active connections, TLS handshakes/sec, and backend response times
- Keep HAProxy process pinned to CPU cores with predictable load (avoid scheduler thrash)
- Pin vCPUs to dedicated physical cores on the hypervisor to ensure consistent performance under load.

HAProxy Routing Logic

- **Dual-network design**: one interface in the **anycast network** (service traffic) and one in the **transit network** (backend & connection to origin).
- **Anycast interface**: handles all API service IPs — every query arriving here is answered **on the same network** to avoid asymmetric routing.
- **Transit interface**: used for backend connections to bank origins, management access, monitoring, and updates.
- **Strict separation**: prevents service traffic from leaking into the transit network and keeps control-plane/admin channels off the public API path.
- **Routing enforcement**: HAProxy binds listeners to anycast IPs only; backend and health checks use the transit gateway exclusively.

Will it work?

So, Can we withstand a DDoS?

1:1 Anycast Node ↔ Telco – 40 GE Each

- **Direct pairing**: Each telco-facing anycast node connects directly to the telco at **40 GE**.
- **Capacity match**: Link speed is equal on both sides — no artificial bottleneck before the proxy.
- **Ingress protection**: Stateless nftables filtering + SYN proxy at the edge drops floods before HAProxy.
- **Traffic segregation**: Anycast interface handles API requests only; backend & management use transit link.
- **DDoS resilience**: Attack traffic stays local to the telco's segment — no spillover to other nodes or transit.

Close, but more work is needed

Mobile infection rate Short-term peaks up to **5%** in specific regions after a major malicious app campaign, before app stores or carriers intervene. Operators without aggressive filtering or user education, **0.3%–1%** sustained.

Assumptions: infected handset can push **~0.3–0.5 Mbps** of 64B SYNs without killing UX.

1% infected (40,000 devices)

- Bandwidth: **12–20 Gbps**
- Packet rate: **~18–30 Mpps**

A **1%** mobile botnet is enough to push **~12–20 Gbps (~18–30 Mpps)** at the edge of what we can sustain

Resilience Status & Next Steps

- Current design: Anycast edge per telco, stateless L4 filtering, mTLS at HAProxy
- Estimated to handle domestic floods up to **20–30 Mpps (~12–20 Gbps)** without link saturation
- Link saturation risk at **>40 GbE** flood from a single source network
- Next steps for high-confidence DDoS resilience:
  - Add upstream mitigation (Flowspec / RTBH) agreements with each telco
  - Consider **2×40 GbE** or additional nodes for high-traffic carriers
  - Continue performance testing under simulated large-scale floods
- We are **not far off** from withstanding massive targeted attacks

Strategic Benefit for the IXP

- Brings **banks' APIs into the IXP fabric** — creating application-level peering relationships
- New role as a **Layer-7 interconnect**, expanding from L2/L3 peering into secure API delivery
- Strengthens the IXP's position as **neutral, trusted ground** for critical services
- Diversifies service portfolio — opens path to hosting other secure, localised applications
- Increases stickiness: banks and telcos both benefit from being members

Strategic Benefit for the Country

- **Localises critical banking services** — keeps traffic inside national infrastructure
- Improves **resilience** during international outages or geopolitical network disruption
- Reduces exposure to cyber threats
- Improves customer experience through **lower latency** and **fewer failure points**
- Sets precedent for other sectors (government, healthcare) to localise services
- Contributes to national **digital sovereignty** and **financial network stability**