



Adventures in Perimeterless Homelabbing

Shea Polansky

2020-11-07

BSides Orlando

shea@bsides:~\$ whoami

- Does security for \$BIG_TECH
 - Software development + sysadmin background
 - Moved to security consulting before landing current gig
- Breaks things for money
- Hosts an unwise amount of services in his house
- Cares about **usable** security

📍 polansky.co 🐦 [@0x5ca1e5](https://twitter.com/0x5ca1e5)





SwiftOnSecurity
@SwiftOnSecurity

Rule 777:


If you don't make a system usable and secure, the user will make it usable and insecure.

<https://twitter.com/SwiftOnSecurity/status/1002383281550233601>



Quick Disclaimer

I will be talking about specific software/technologies in this talk, some of which are commercial products. I have no relationship with any of the projects in question, and any comments I make are purely based on my own tinkering, and definitely do not represent the views of anyone else, especially not my employer. This is also not meant to be a complete survey of what's out there; there are likely to be products or technologies I didn't encounter or didn't find time to include.





Background

Perimeterless? Homelab?



Homelab /hōm-lab/

- A justification for a higher power bill and a better internet connection
- A fantastic way to learn how to admin things
- 100% home-grown, organic, cruelty-free technical debt

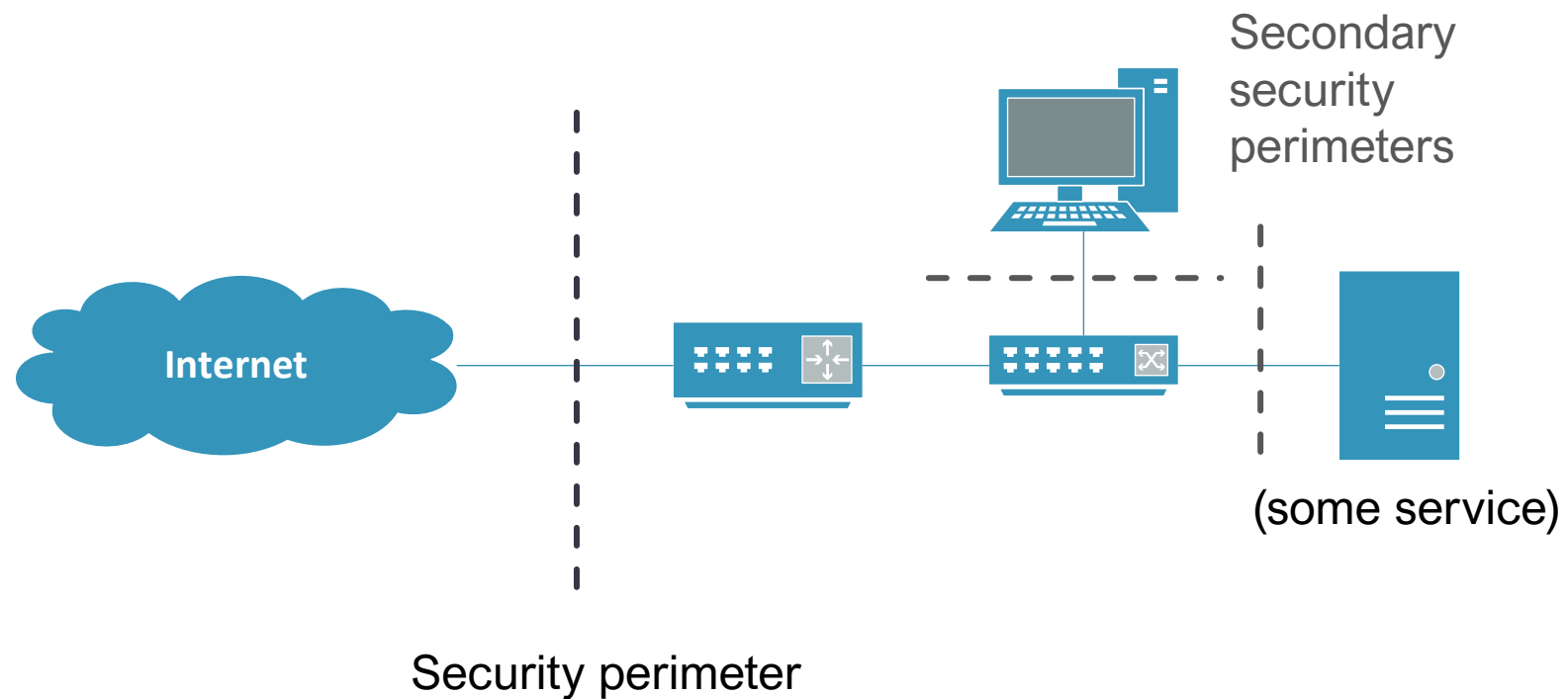


Slide borrowed from a [talk](#) I gave at [SCaLE 18x](#) with [Morgan Gangwere](#)

<https://redd.it/bsxz2e>

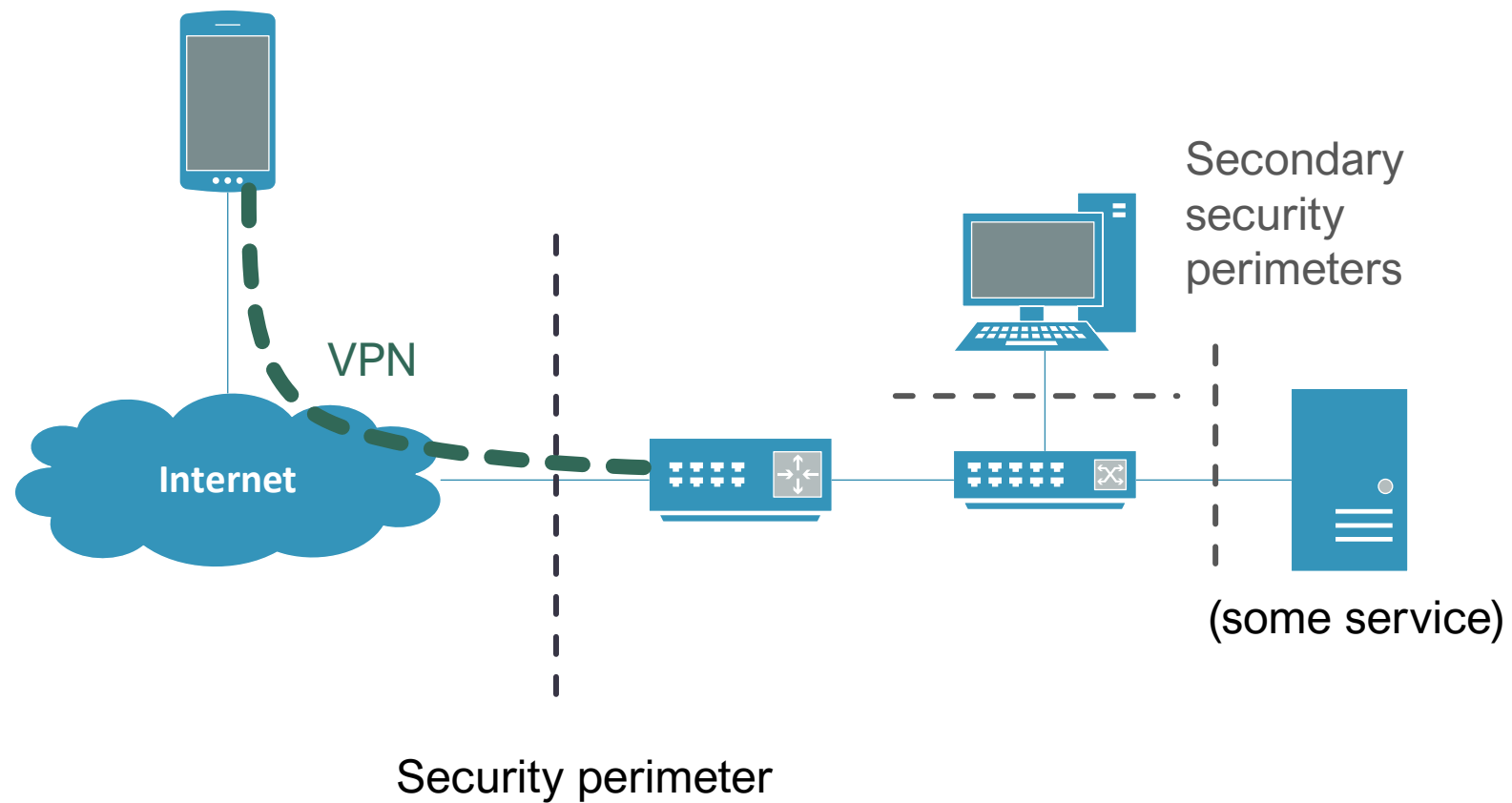


Typical Home Network w/ Homelab





Remote Access





And so VPNs solved every possible remote access problem with perfect convenience and security.

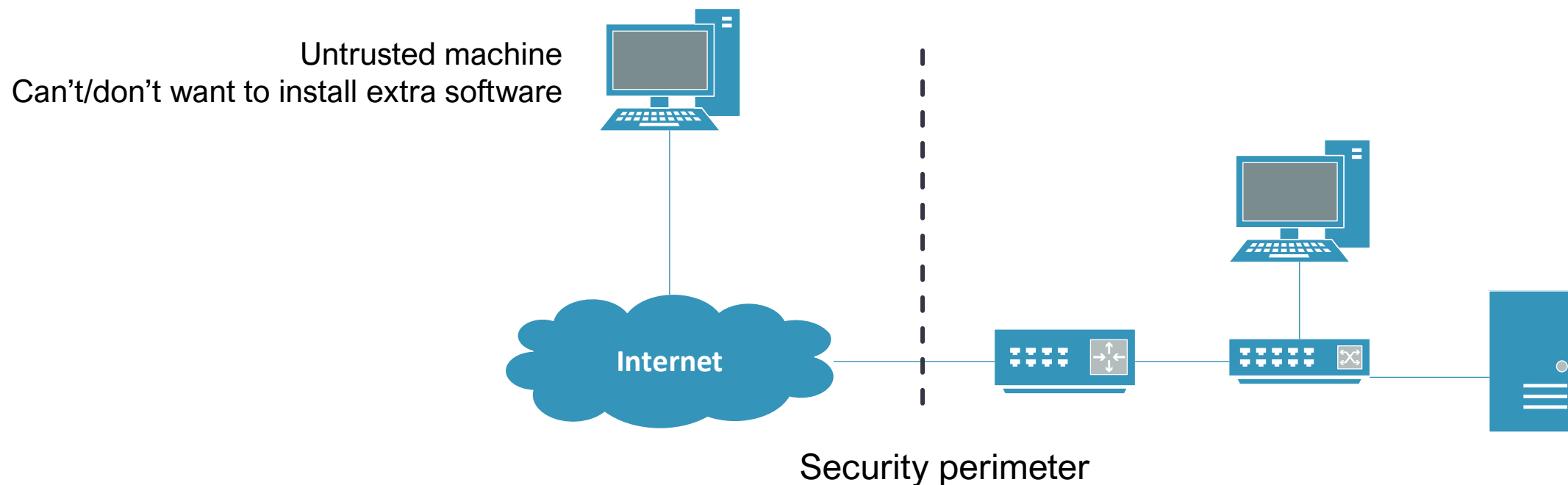
We can all go home now, problem solved.





Homelab VPN Inconvenience

What about accessing stuff on computers you don't own?





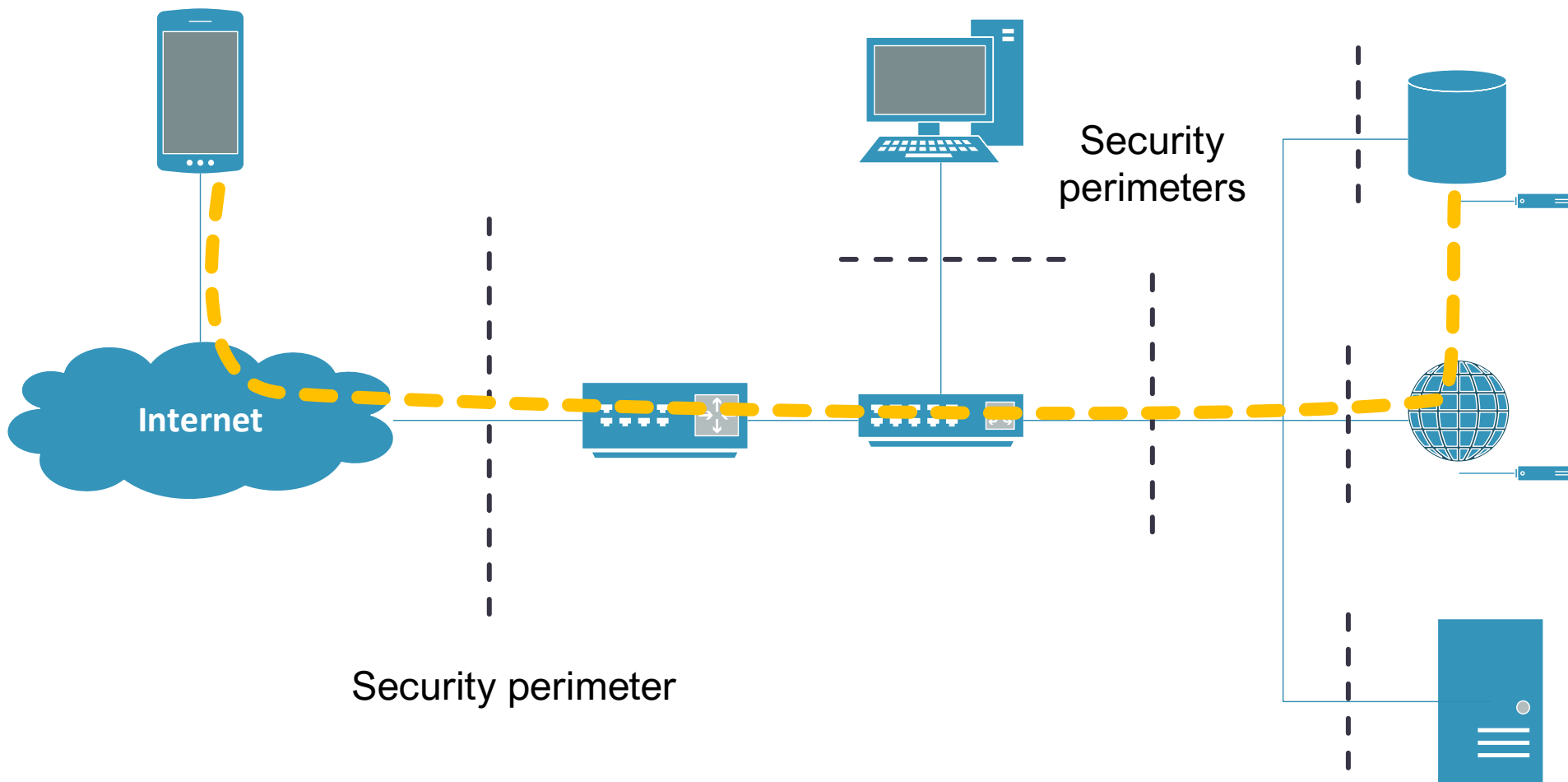
Perimeterless Networking

“Zero Trust” without the buzzwords





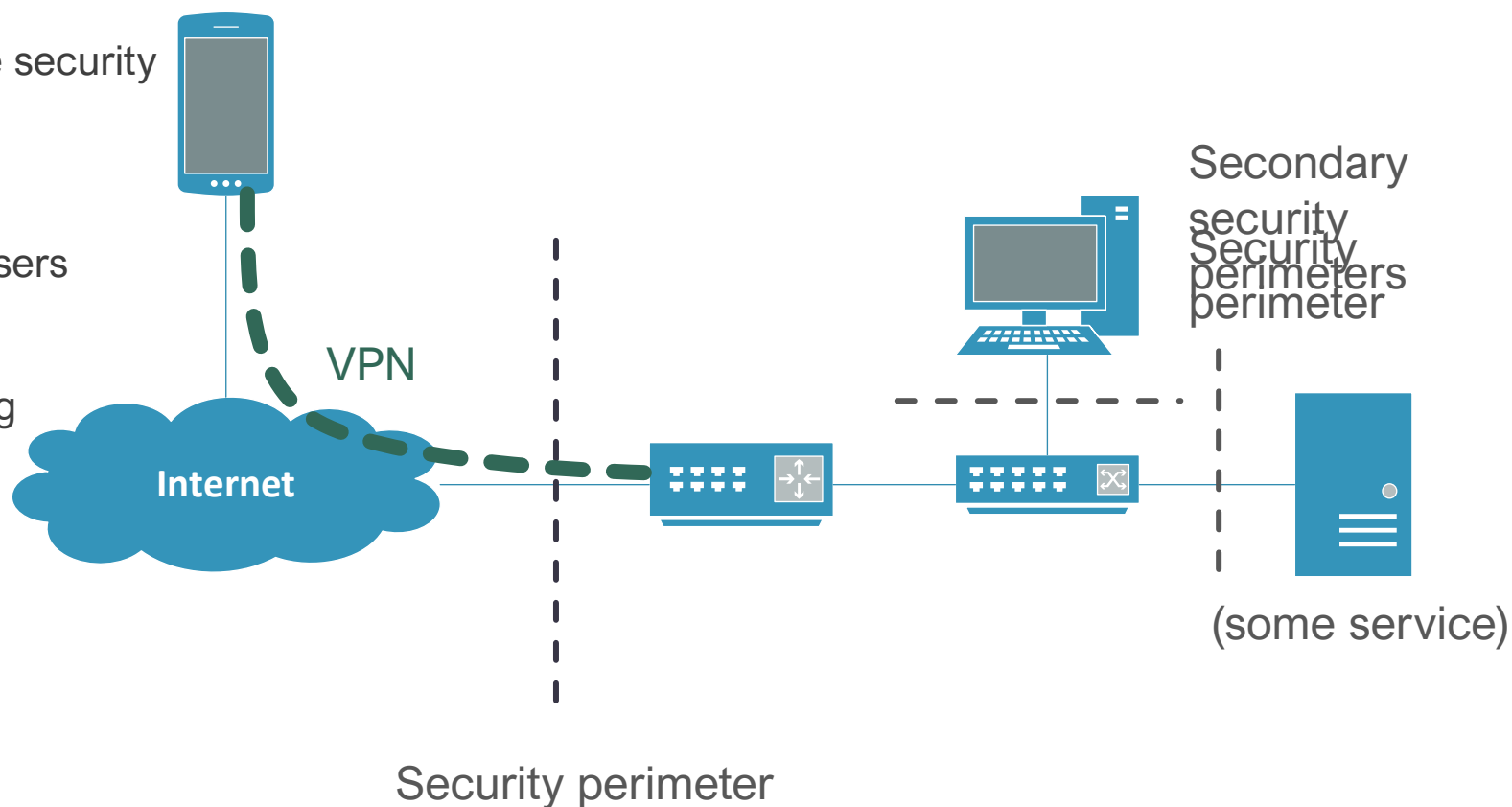
“Zero Trust” Networking





Goals for a “Perimeterless” Network

1. Provide the same security as a VPN
2. Be as convenient and invisible to users as possible
3. Work with existing on-prem infra



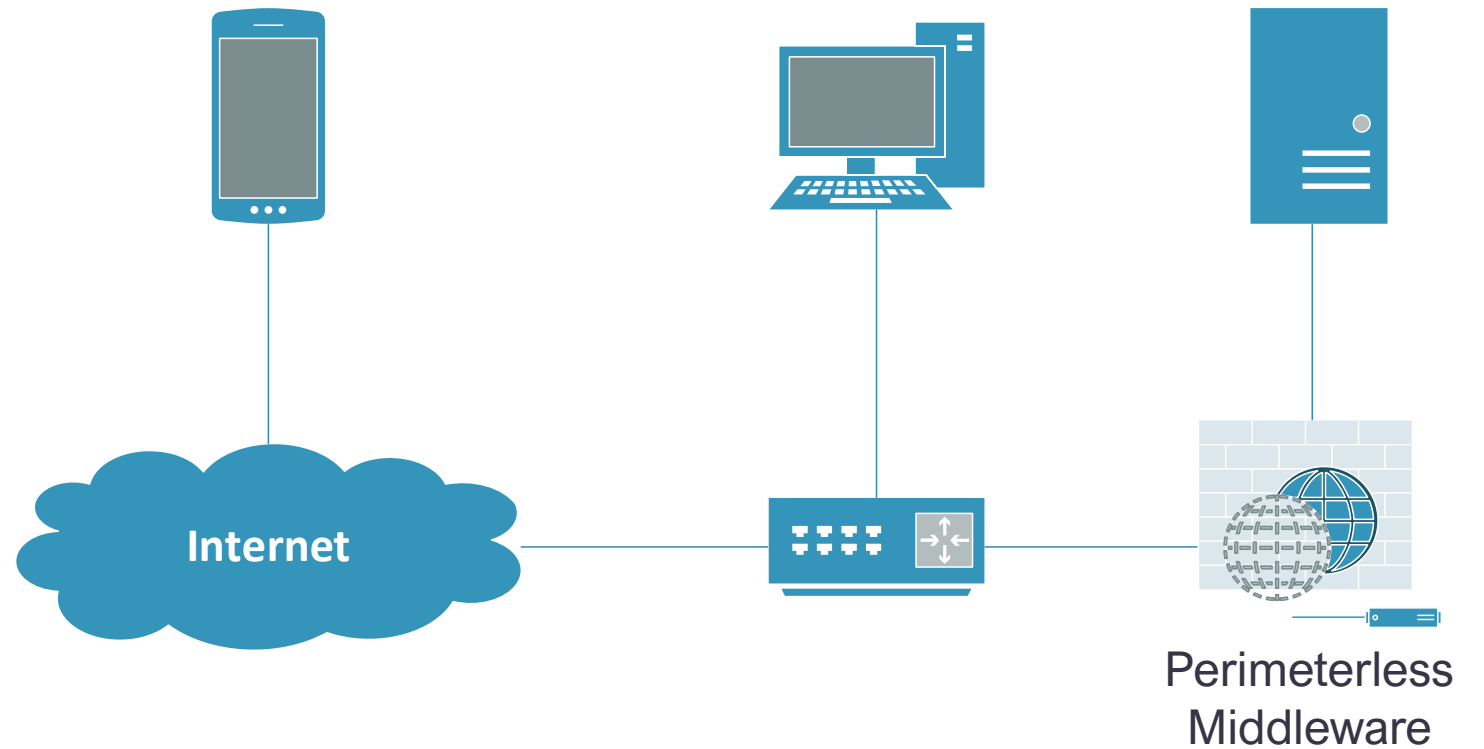
Perimeterless Solutions

Perimeterless Middleware

- Mutual TLS
- Access Proxy
- Mesh VPNs

Side benefits

- Get strong SSO by default
- Centralize access controls
- Hopefully leverage hardware security





Mutual TLS

The minimal additional software solution

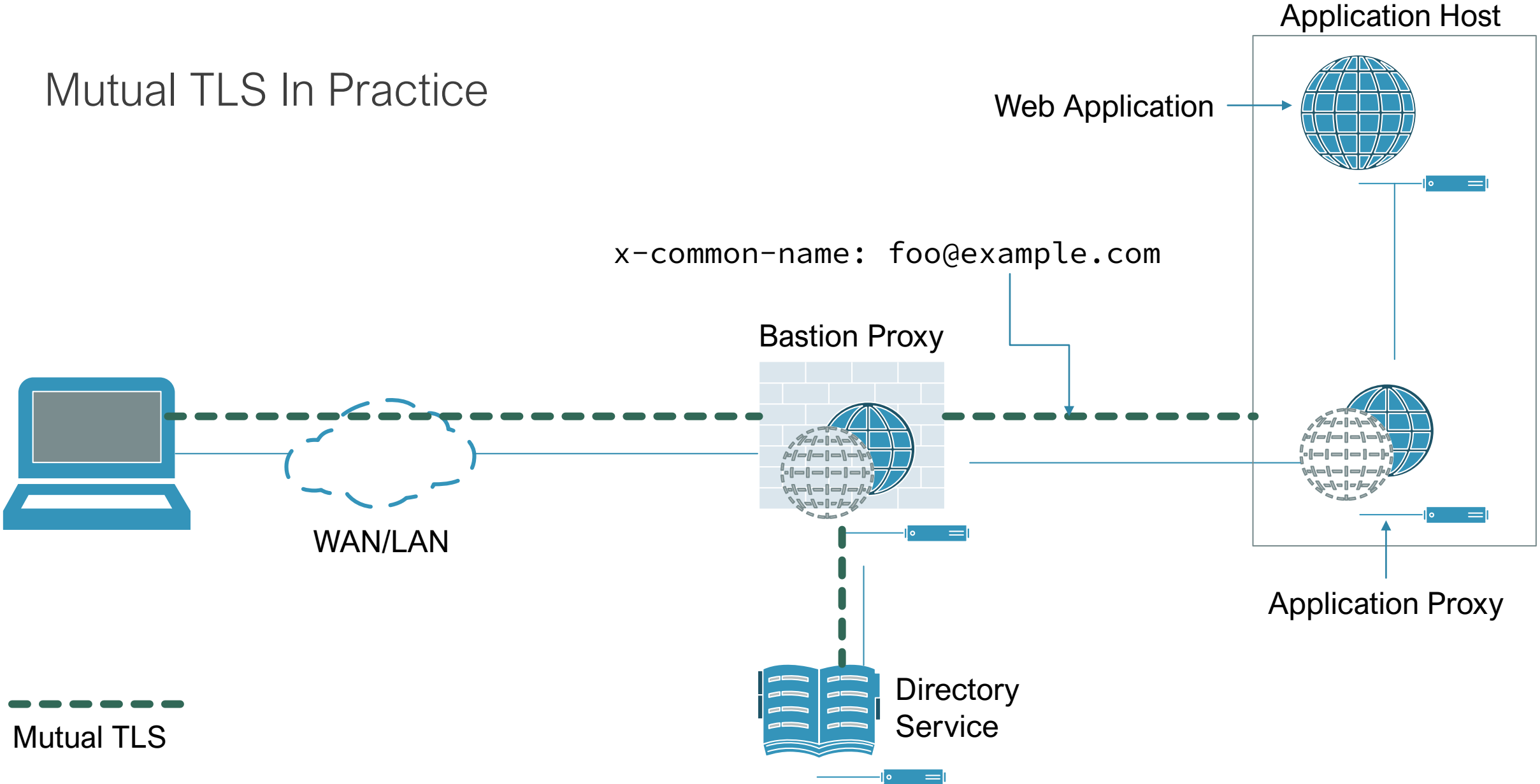


Mutual TLS as Perimeterless Middleware

- TLS supports mutual authentication
- Issue every user a certificate from your enterprise CA
- Stick a reverse proxy in front of your web applications
- Make access control decisions based on certificate properties
 - Common name
 - EKU fields (maybe even custom ones)
- Reverse proxy can forward user details to web application to enable single sign on
- Security is very good – unauthenticated attack surface is basically nothing, certificate parsing routines are pretty mature
- Hardware security is readily achievable
 - Most TLS implementations support PKCS#11
 - Smart Cards, Yubikeys, TPMs all allow you to do MFA
 - Security device won't sign the TLS challenge without PIN
 - PINs can be short – users will be happy
 - They also keep keys out of main memory, so keys can't be stolen



Mutual TLS In Practice



Mutual TLS Implementation Specifics

- Clients
 - Windows supports generating certificates on the TPM
 - Browsers will use the system cert store
 - On Linux you can give the browser a TPM PKCS#11 Module
 - On MacOS, unfortunately no way to use the T2
 - All major OSs support smart cards/YubiKeys
- Reverse proxy
 - Nginx with `auth_request` module
- Some kind of access control server
 - This is going to be somewhat custom; I recommend starting with simple LDAP to HTTP bridge

So why not just do that?

- Things that don't run in a browser won't work
 - Although this is somewhat easily solved with stunnel + SNI
 - PKIs are actually kinda hard
 - Not everything supports mutual TLS
 - Mobile browsers are wonky in particular
 - Inflexible authentication
 - MFA is not enforced by the server at all
 - Just relying on the smart card to be well behaved (and available)
 - UX is terrible for edge cases
-



mTLS User Experience

- User is prompted for their SmartCard PIN on first connection
 - Cached for some time, then prompted again later
- If they don't have their smart card, no access period
- If there's a problem, they get a generic browser error message about not being able to negotiate at TLS connection
 - Not very understandable to normal users
- Self-service is difficult to impossible
 - If services are protected by mTLS and you can't negotiate a mTLS connection, you can't self-service your mTLS credentials



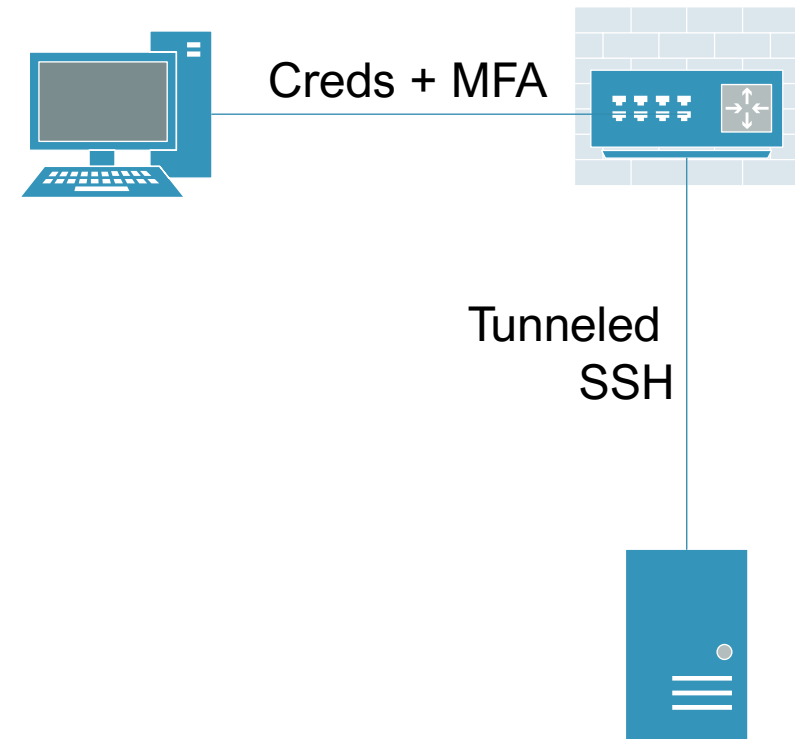
Privileged Access Management + Access Proxies

Modernized Bastion Proxies



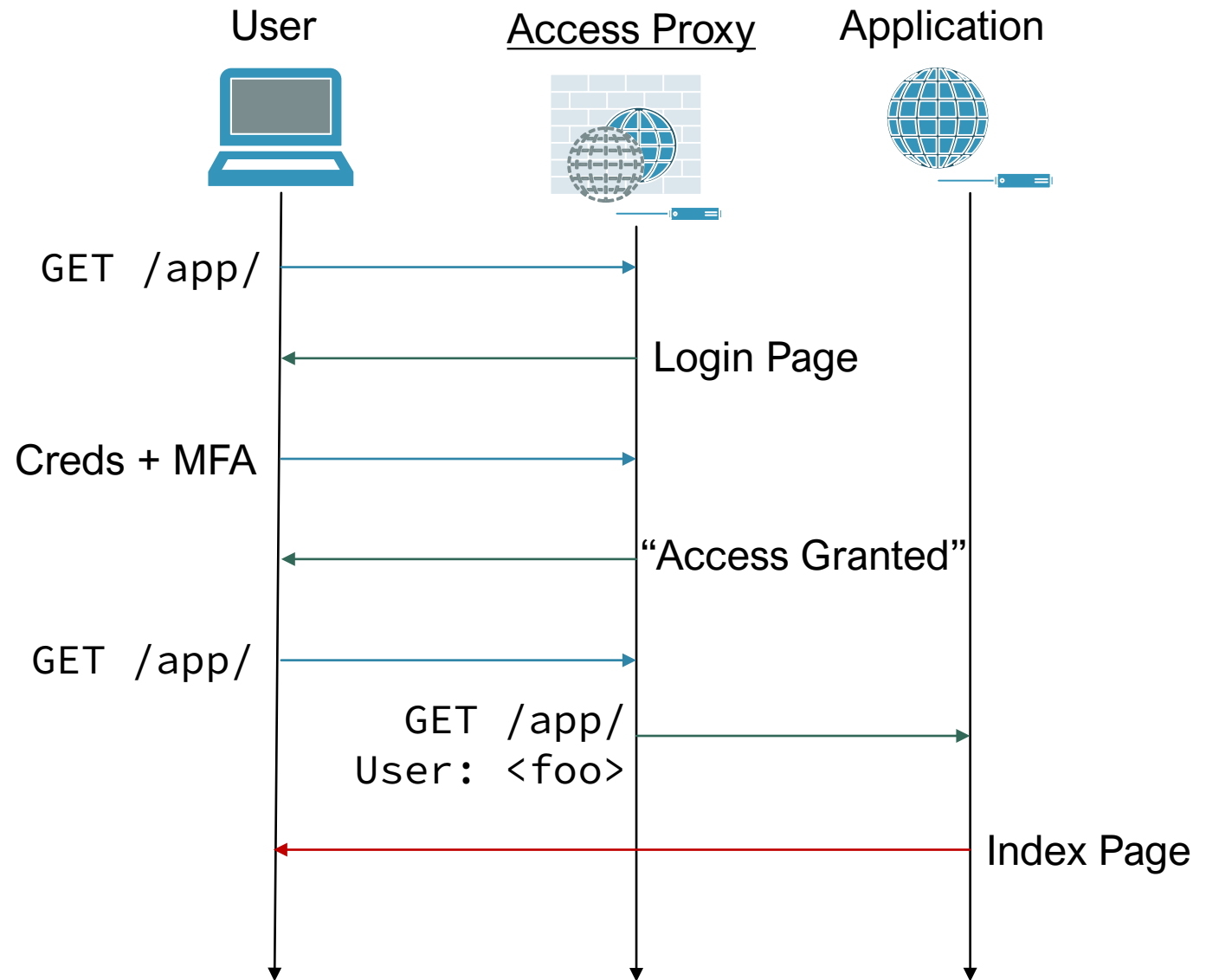
Privileged Access Management Solutions

- Very popular for cloud workloads
- Provide centralized MFA / ACLs for (typically) SSH access to hosts
- Act as a bastion through which all traffic is funneled
- Provide logging, just-in-time access to hosts



Access Proxies

- Modernized, more flexible versions of the previous mutual TLS setup
 - Sort of like the principle of a PAM solution, but applied to everything
- Proxy + integrated web application
 - Handles login, access control rules, session management, self-service
- Can allow more complex authentication scenarios
 - U2F, TOTP, hardware tokens, SMS, whatever
- Can allow very fine-grained authorization
 - Source IP, access patterns, etc.





COTS Access Proxies

POMERIUM — Identity Aware Proxy

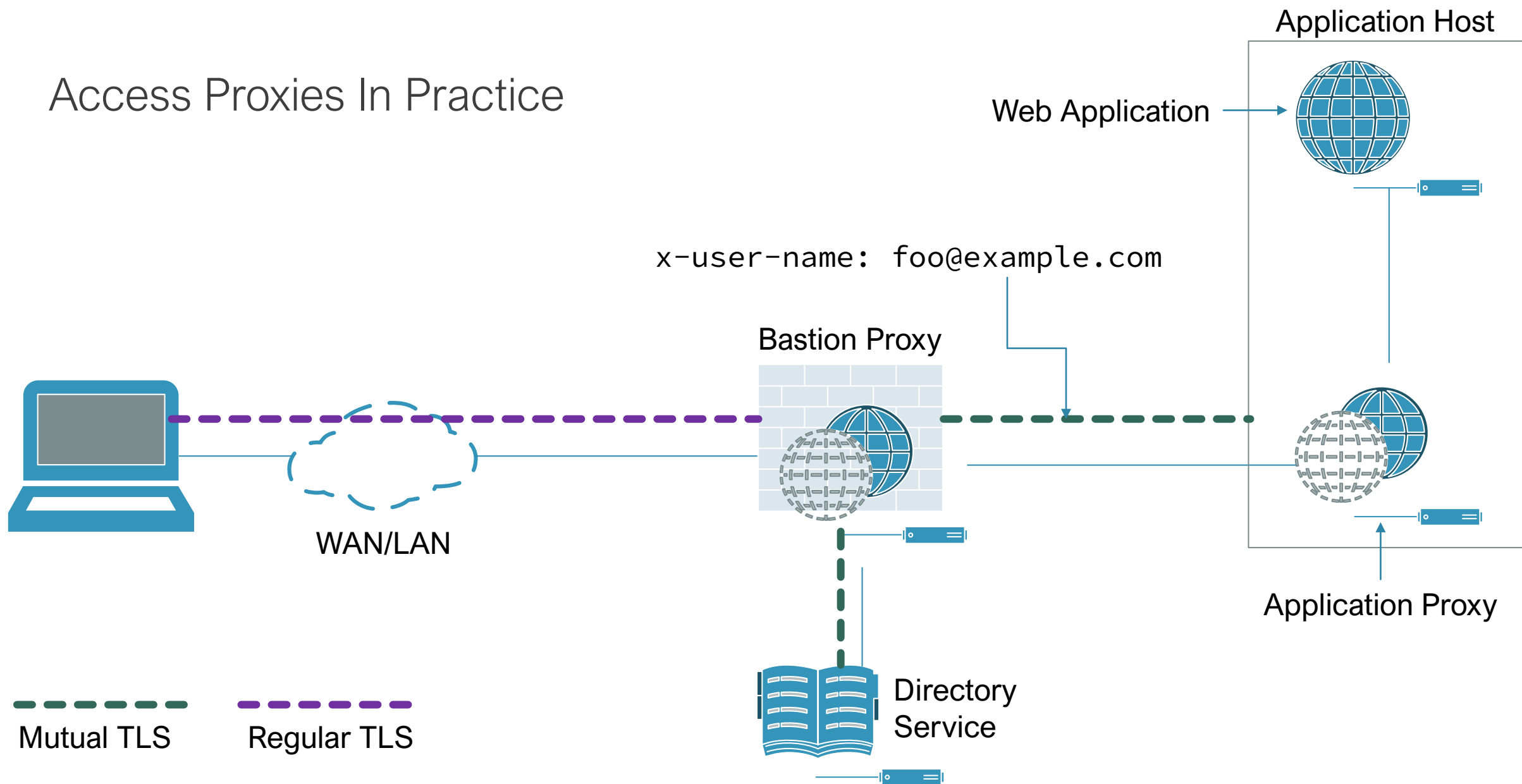
- Sits directly in-line with HTTPS requests
- Handles both actual proxying and access control
- User login is handled via OIDC
- Open Source with “contact us for pricing”
enterprise version coming Soon™
 - No current free version limitations
- Support WebAuthn authentication
- Support complex access control rules based on path, groups, etc.
- Homelab friendly!

authelia — Authn/Authz Server

- Serves as the access control component only
 - Must be integrated into your existing reverse proxy
- Users are from LDAP or local database
- Handles first+second factor itself
- Free and Open Source



Access Proxies In Practice





Access Proxy Benefits Over Pure mTLS

- More flexible authentication
- Self-service supported
- More flexible authorization
 - E.g., prompt users to re-login if suspicious access patterns, accessing from different country
- Doesn't rely on browser support for mTLS, OS support for hardware tokens
- UX is overall *way* better
 - Can be much more flexible on authentication
 - Much easier route to self-service credential management
 - Can have intelligible error pages to users

Access Proxy Caveats

- Similar to mTLS in some ways
 - Hard to integrate with non-browser solutions
 - Can hack things together with custom cookie readers
- Basically doesn't work at all for non-HTTP protocols (e.g., SSH)
 - Some enterprises (e.g., Google) just use a custom SOCKS over HTTPS solution (AKA a VPN)
 - Others use a separate PAM solution (which is more work and more complex)
- Plus you now are exposing a web app to the internet, so better make sure it's secure
 - See: Pulse Secure VPN



Mesh VPNs

The future of perimeterless (and zero trust) networking

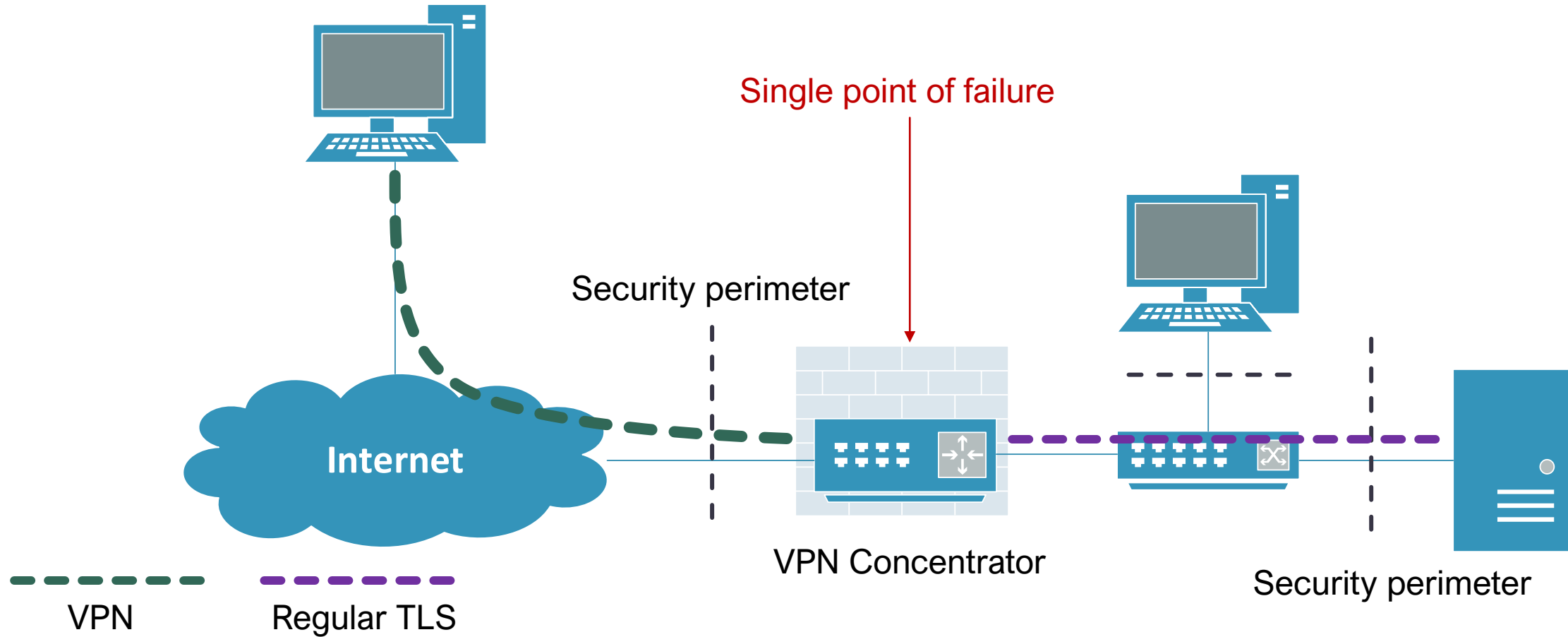


Wait, I thought this
was about getting rid
of VPNs!

- Kinda! In my homelab, I don't care about risks from unknown devices (since it's just me)
- For enterprises, employees are accessing from a limited set of devices, either employer owned or subject to MDM policies
- But let's talk about the other disadvantages of a VPN

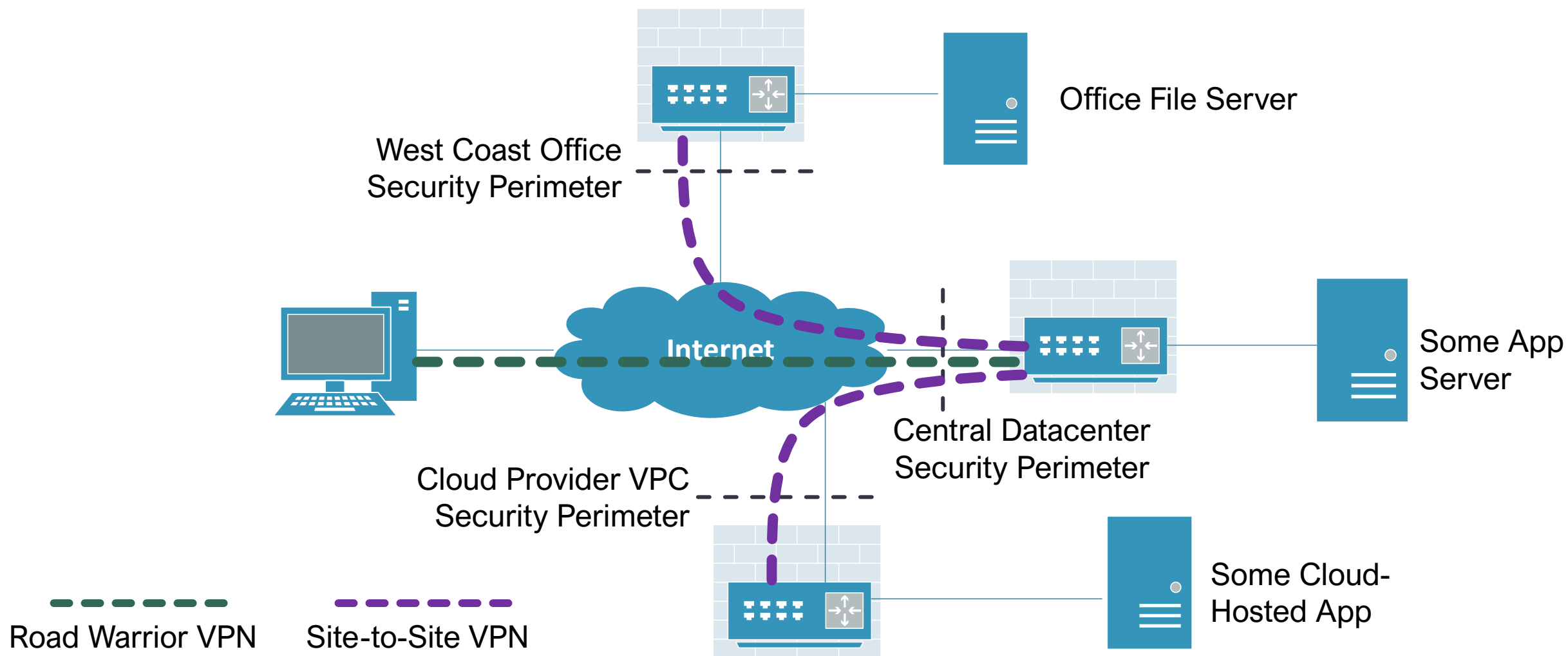


Typical Road Warrior VPN Configuration



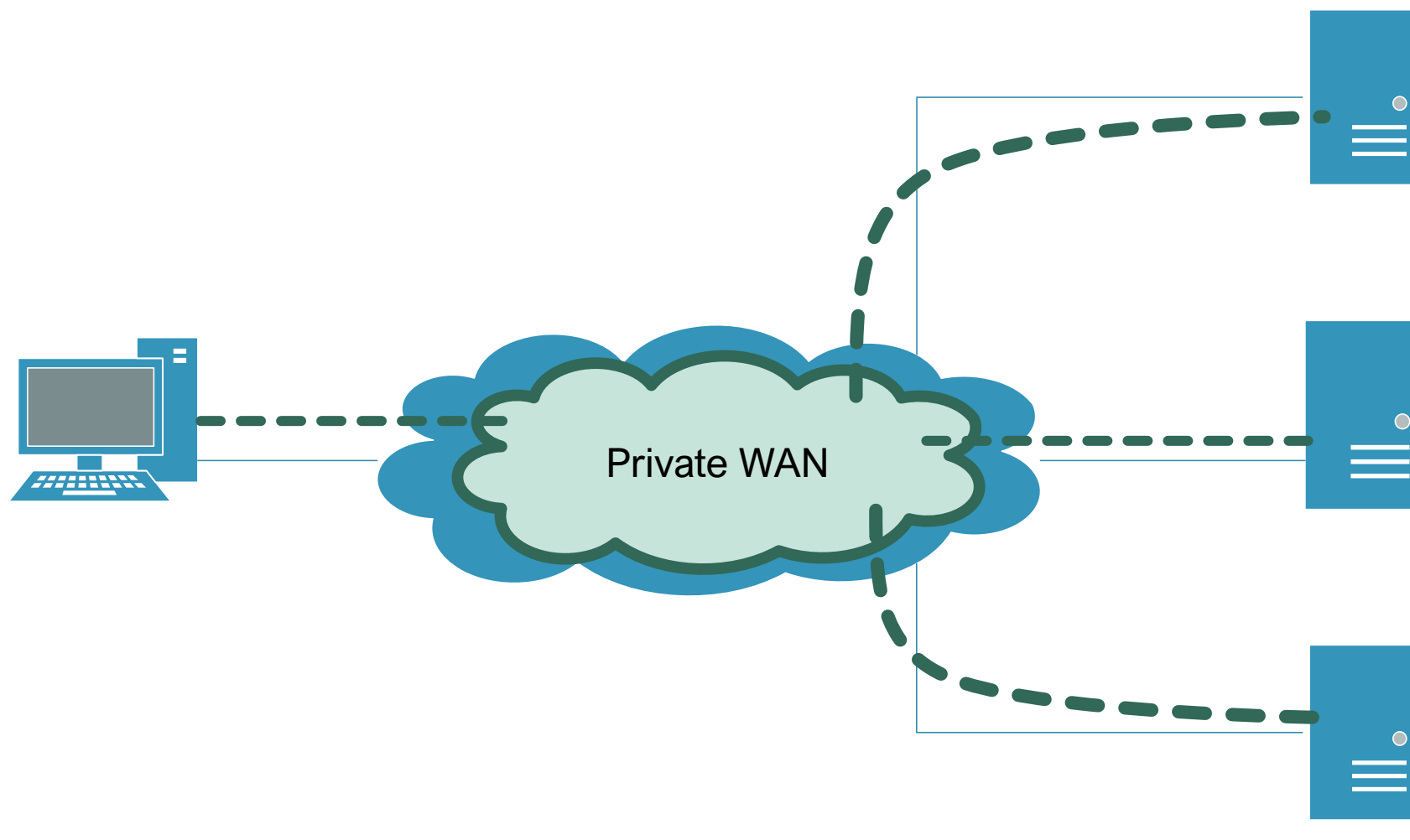


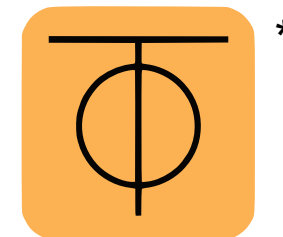
Road Warrior VPNs in Complex Enterprises





Mesh VPNs in Complex Enterprises





ZeroTier: “A Global Ethernet Switch For Planet Earth”

- All clients connect to a global “WAN” identified by their public key
- You join a network by its controller’s public key
- The controller issues ACLs
 - Simple: Allow from x to y
 - Complex: Can assign tags, data to clients and make decisions based on that
- Network ACLs are enforced by each endpoint
 - ACLs + client data + IP address ownership are signed by controller to prevent spoofing
- Hypothetically, you could bind IPs to LDAP and then make access control decisions based on those
 - Since they’re unspoofable, they’re equivalent to a unique user identity (as long as you only listen on the ZT interface)

* Not a cult logo, I promise



Potential ZeroTier Benefits

- Network ACLs are automatically distributed to all users + enforced cryptographically
 - Spoofing attacks impossible
- Tunnels any layer 3 protocol
- “Just Works” across complex network topologies
- Could bind IP addresses to LDAP for user identification
 - Incoming connection on ZT interface → LDAP lookup IP → make access control decision by user groups



So why not
ZeroTier (yet)?

No MFA support

Poor integration with existing
directory systems

Somewhat immature software



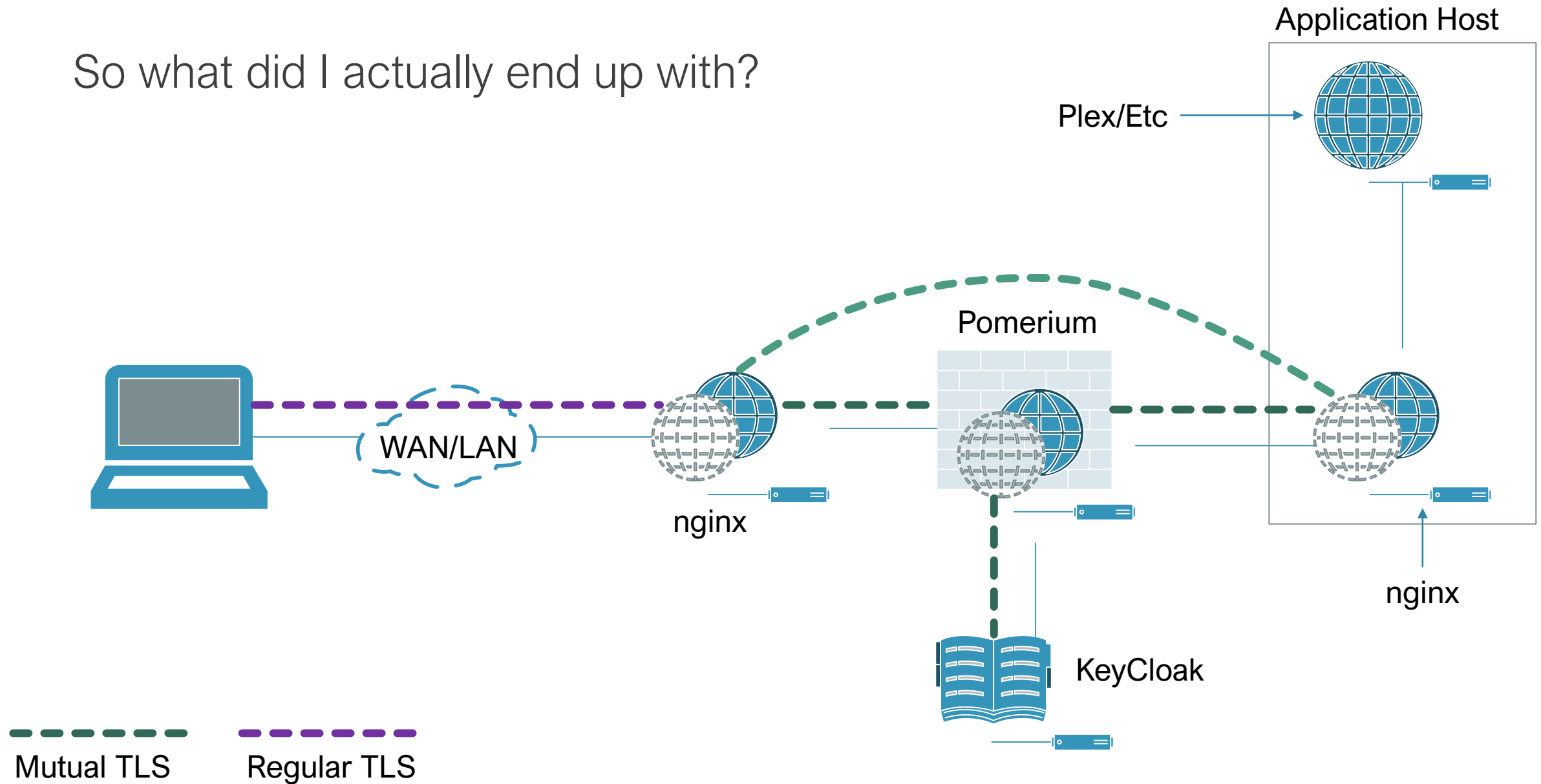
So what did I actually end up using?

Bringing this back to my actual homelab.





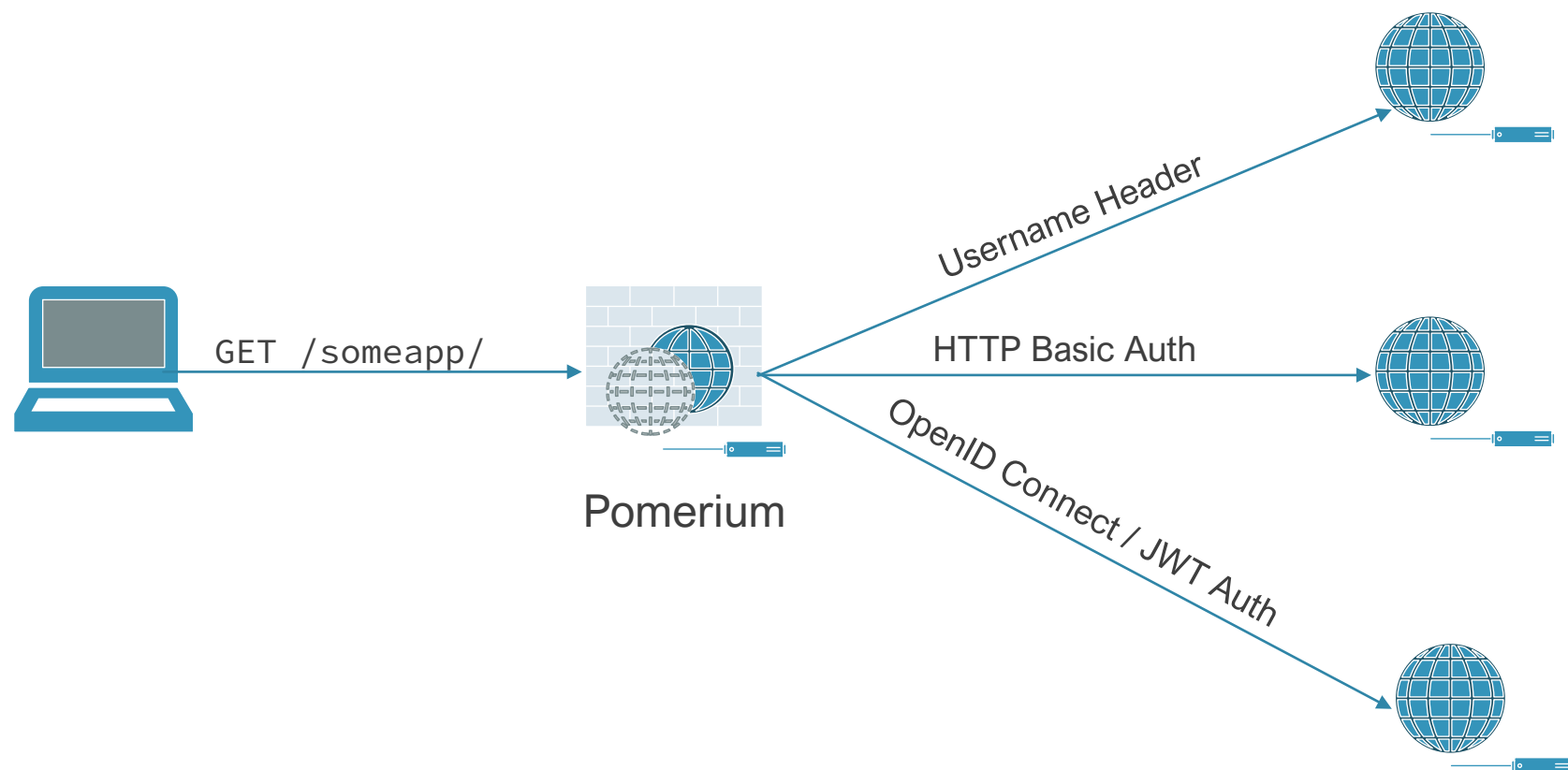
So what did I actually end up with?





Grafting Zero-Trust Auth Onto Existing Services

Happy path: Browser app and backend with external auth support





Grafting Zero-Trust Auth Onto Existing Services

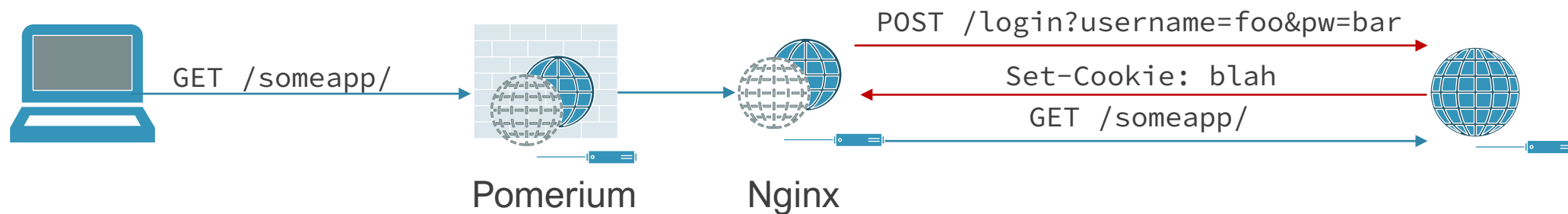
Alternate happy path: Browser and single-user backend





Grafting Zero-Trust Auth Onto Existing Services

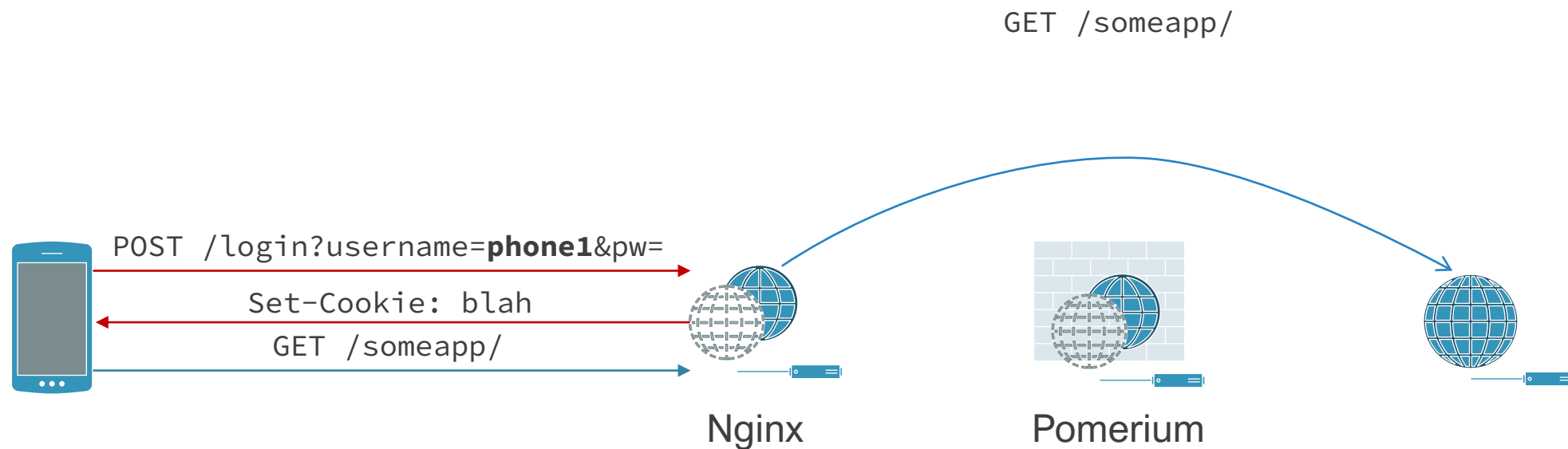
Unhappy path: Browser and app *without* external auth support





Grafting Zero-Trust Auth Onto Existing Services

Unhappy path: Thick client and app with external auth support



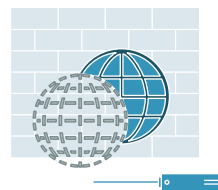


Grafting Zero-Trust Auth Onto Existing Services

Very unhappy path: Non-HTTP Service



?



Pomerium

?





User Experience Demo

Bringing it all together



Aside: WebAuthn

- Strong cryptographic authentication with universal browser support
- Challenge/response mechanism includes intended origin — **nearly phishing proof**
- Two authenticator types
 - “roaming” authenticators (dongles like Yubikeys)
 - “platform” authenticators (system secure element – TPM/TrustZone/T2/etc.)
- Can be used as a second factor, or first + second with user verification enabled
 - Every major OS supports it in both modes via platform authenticators
 - Windows Hello for Business also integrates it into AD/Kerberos/Enterprise PKI
- **Please consider adopting, your users will thank you :)**



How about applying this for real?

Let's get *enterprise*.



Users will be happier

- No more VPN
- Less/nearly no more time spent typing passwords in

Increased Security

- Cryptographic authn available on most platforms
- Fine-grained access controls on all apps

Reduced Support Costs

- No more dealing with weird VPN issues
- Reduced licensing headache

The Pitch

Overall Rollout Strategy

Step zero: buy-in from management

SSO

- If you don't already have SSO available you aren't doing any of this, so start there.

Cover web applications first

- Nearly all perimeterless options are built for the Web first

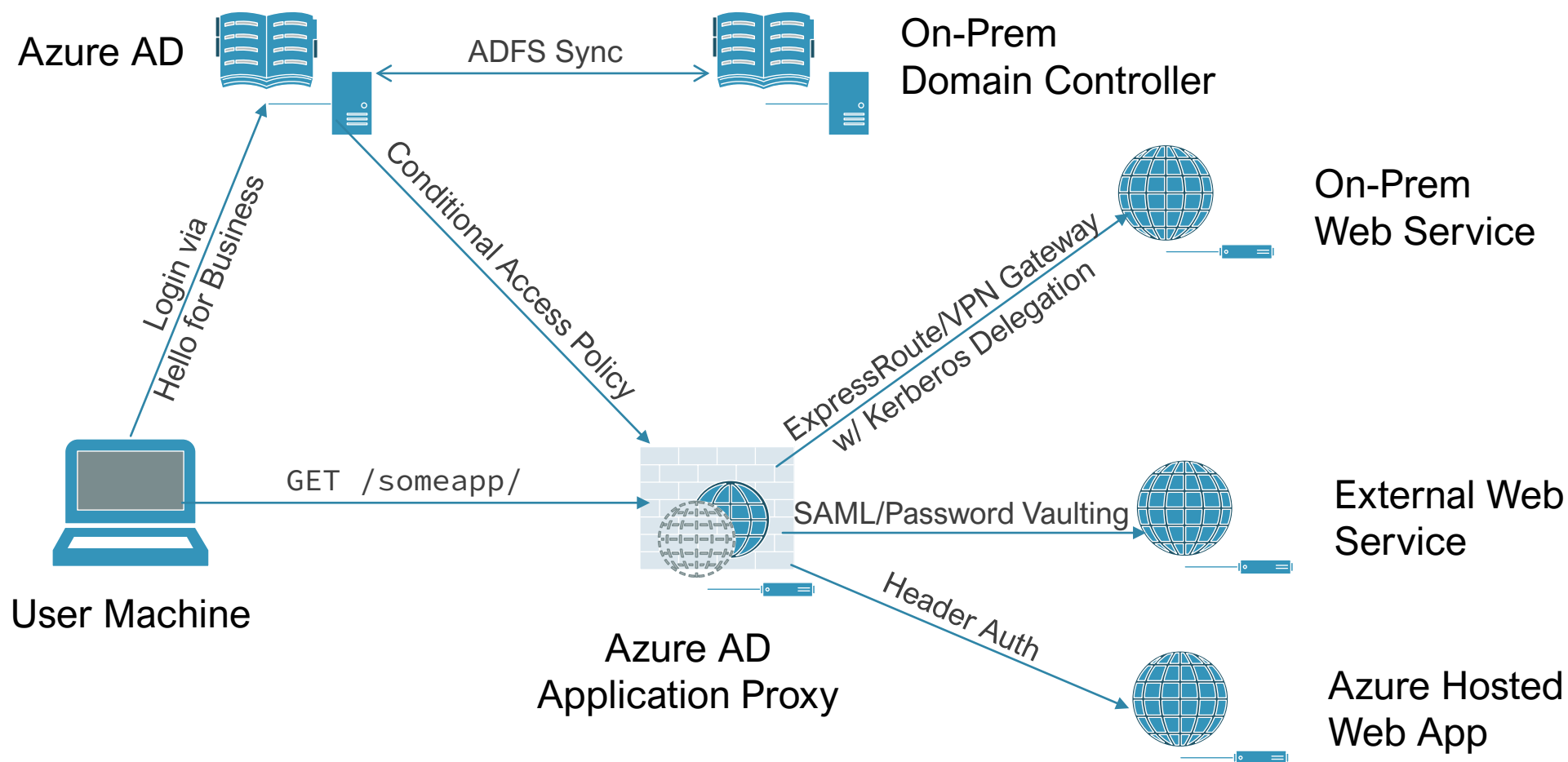
Expand coverage to non-web apps

- Use custom tunnels or custom ingress logic

Iterate over time



Example Enterprise Implementation — Full MS/Azure



Implementation Notes

- No matter what you do, this will be *very, very* custom
 - Your enterprise almost certainly has services that will require custom logic and implementation weirdness
- Use a commercial system if you can — implementing yourself is a lot of work
 - Azure AD with Conditional Access and their application proxy is expensive but incredibly powerful
 - Cloudflare also just launched a similar option
- Use it as an opportunity to deploy modern authentication mechanisms
 - Windows Hello For Business + U2F Authentication
 - Prevents phishing, makes users happy about reduced password complexity rules

Thank you.

Questions?

Links / References

- <https://www.beyondcorp.com/> — Original paper that started me on this idea
- <https://polansky.co/blog/tpm-backed-certificates-windows/> — TPM-backed certs under Windows
- <https://github.com/tpm2-software/tpm2-pkcs11> — TPM PKCS#11 stack for Linux
- https://nginx.org/en/docs/http/nginx_http_auth_request_module.html — nginx module to provide access controls based on external requests
- <https://www.pomerium.com/> — Identity aware proxy
- <https://www.authelia.com/> — Forward auth server
- <https://www.keycloak.org/> — OpenID Connect / SAML server with extensive options / federation support
- <https://www.zerotier.com/> — Mesh VPN / SD-WAN
- <https://www.microsoft.com/security/blog/2020/04/30/zero-trust-deployment-guide-azure-active-directory/>
- <https://www.cloudflare.com/teams/access/> — Cloudflare Access Proxy