ZEROTIER

Reimagining the VPN

Traditional VPNs

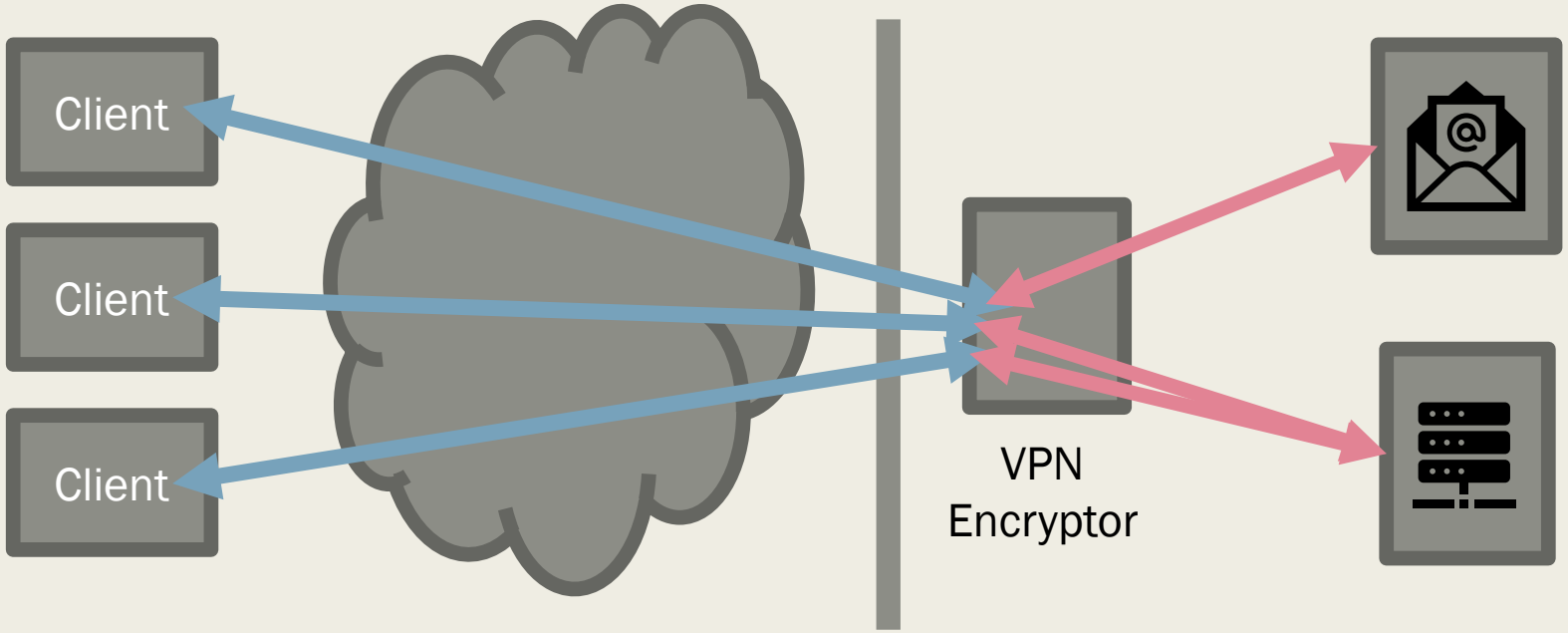
Problem

- Access to internal resources over untrusted network
- Secure Communication Between Sites

Solution

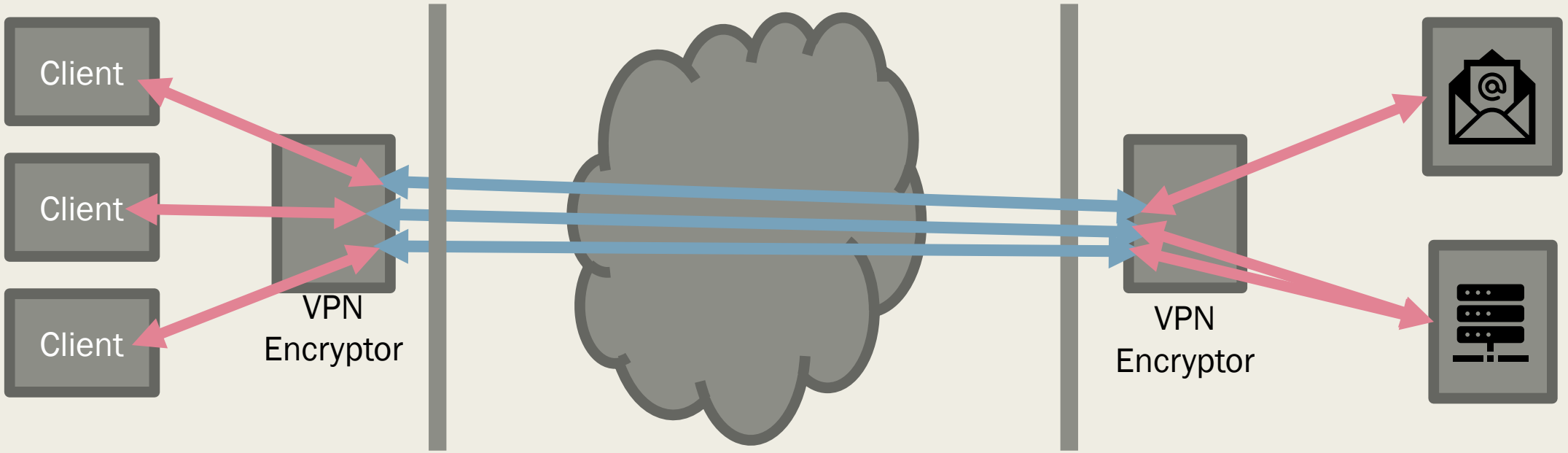
- “Road Warrior” configuration
- “Site-to-Site” configuration

Traditional VPNs – Road Warrior

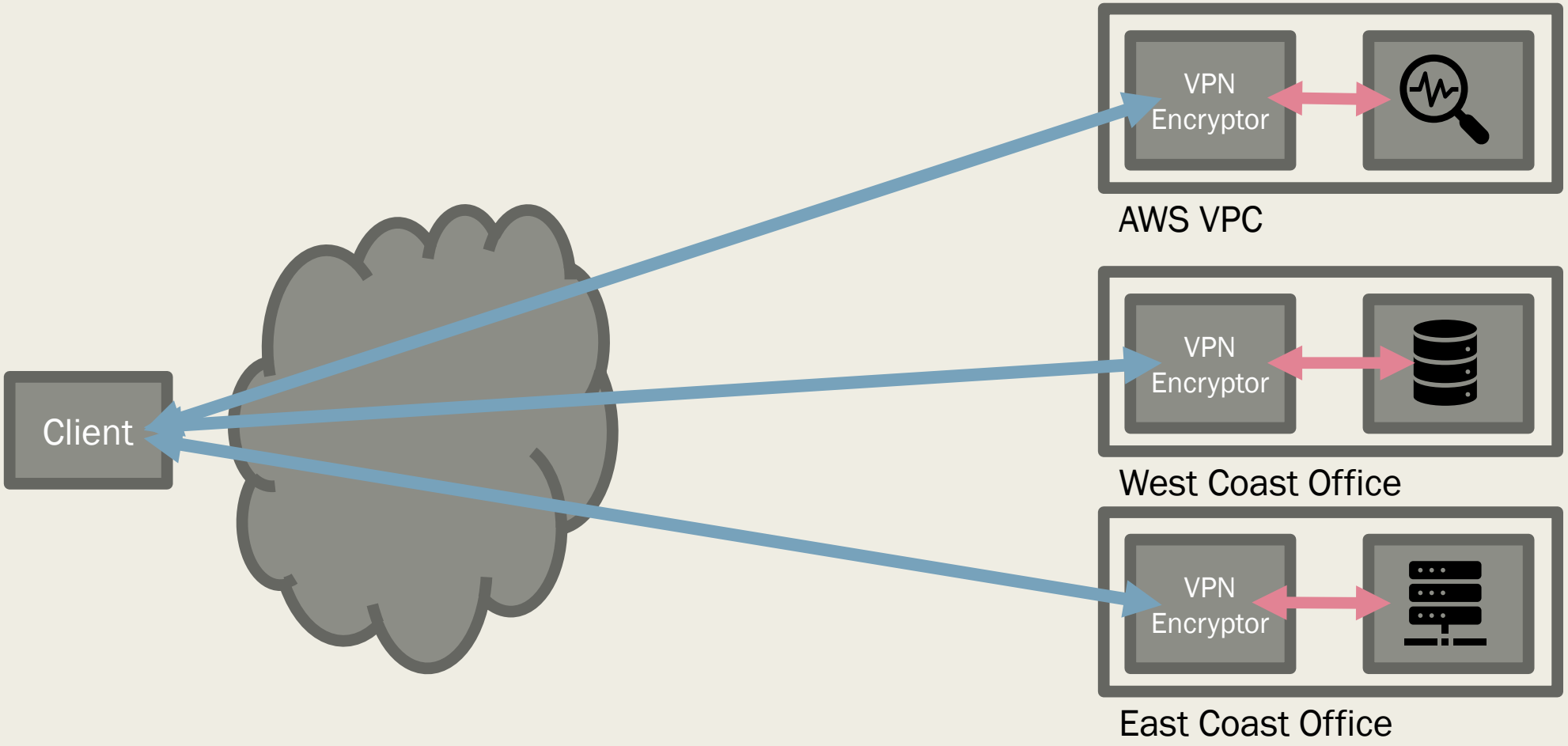


Q: Where are the standard symbols for those network entities? A: In the copy of Visio you're going to buy me. :)

Traditional VPNs – Site to Site



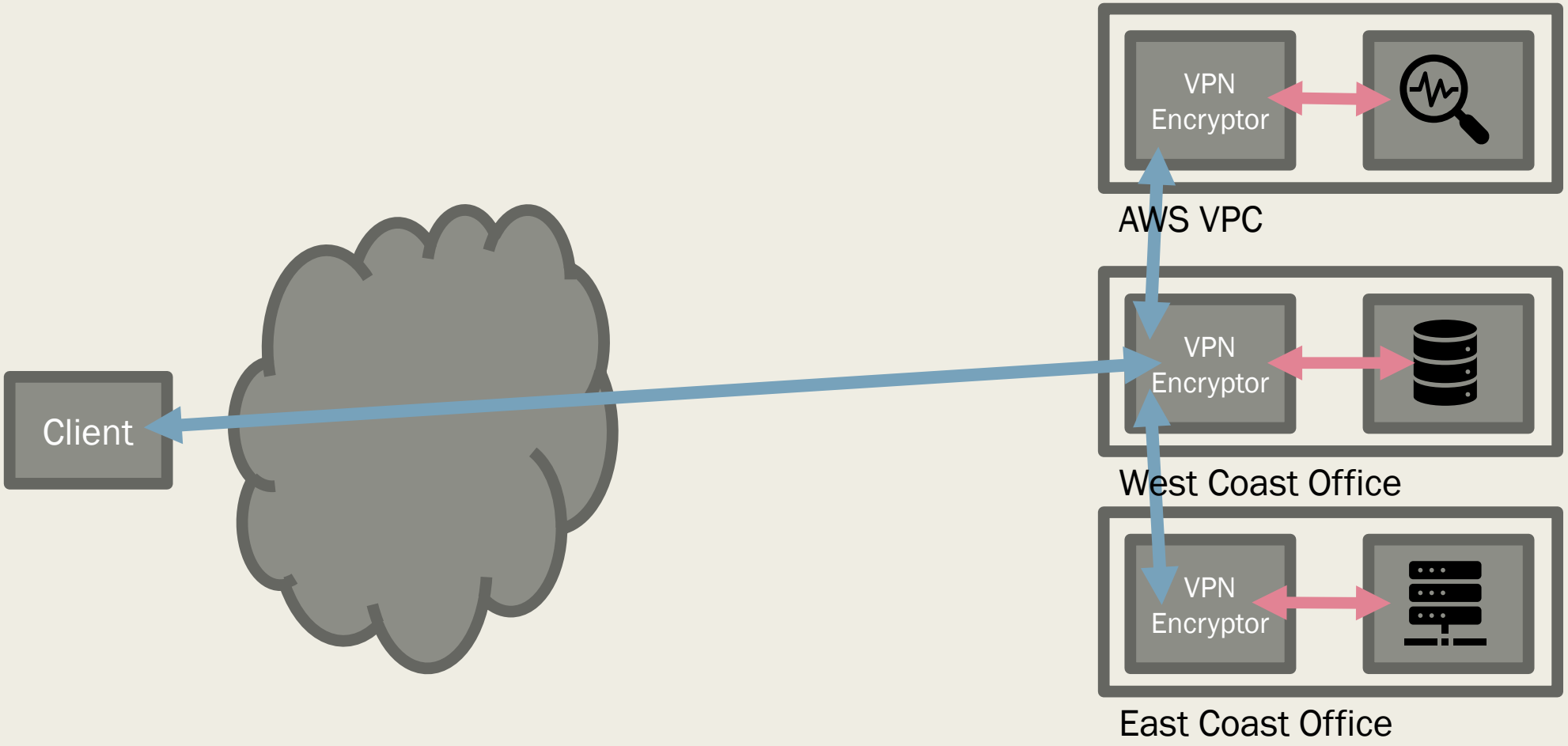
Problem #1: Multi-Site Road Warrior



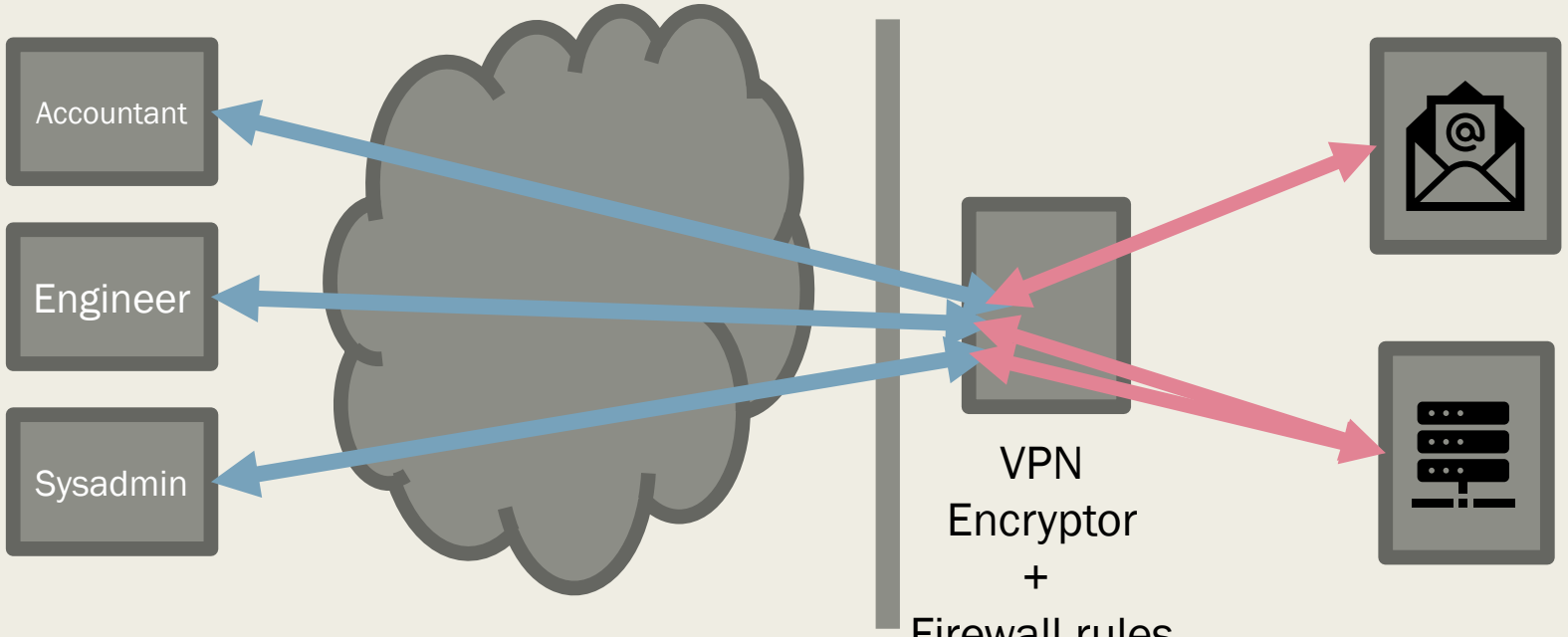
Not actually possible

*Except with WireGuard. But WG doesn’t scale, and isn’t finished.

Problem #1: Multi-Site Road Warrior

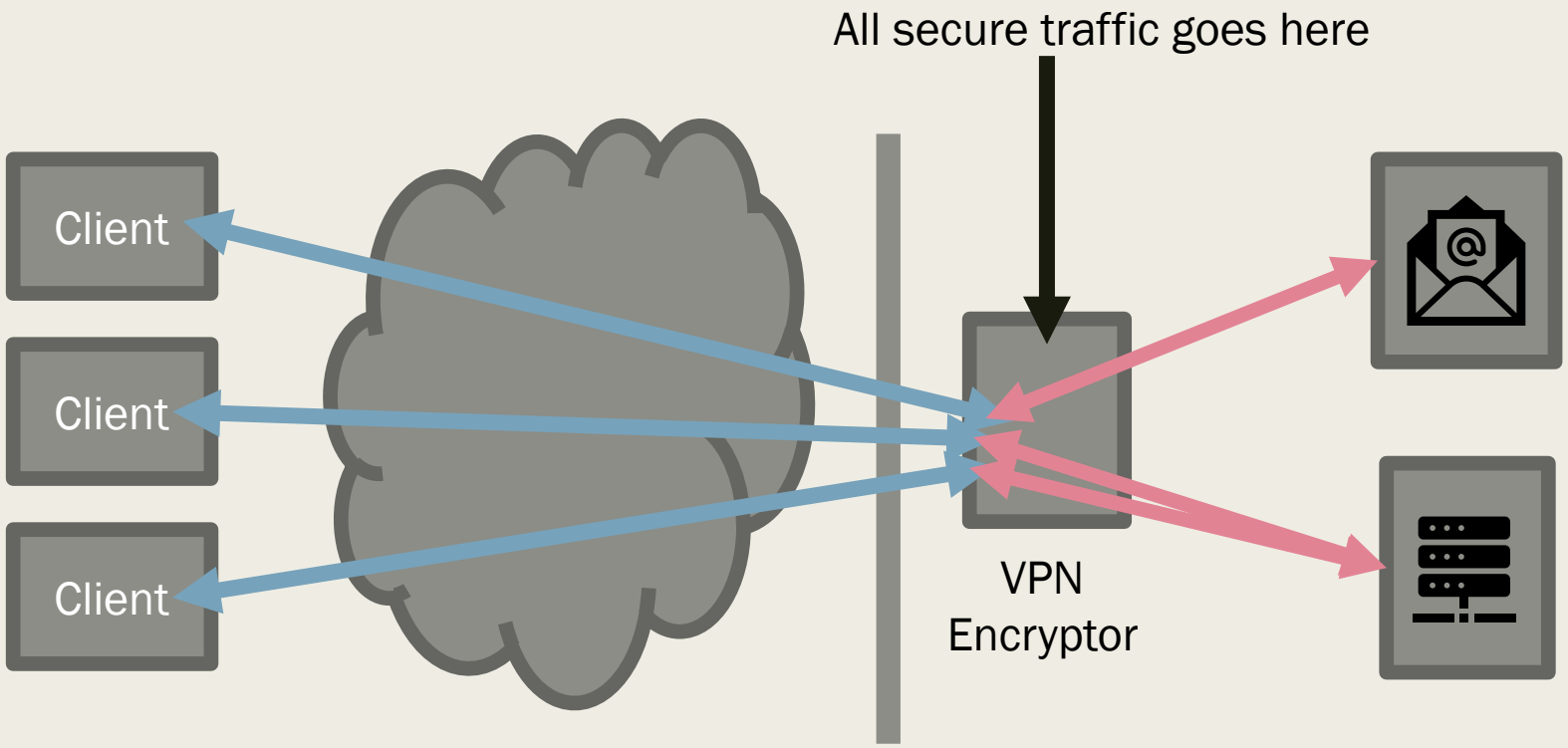


Problem #2: Access Control



- Gets complicated fast
- Granularity is hard

Problem #3: Single Point of Failure



Problem #3: Single Point of Failure

Security

- Lots of trust placed in the encryptor
- Can talk to everything
- Is implicitly trusted by internal network, external clients
- Own it = win
- (See: Pulse Secure VPN)

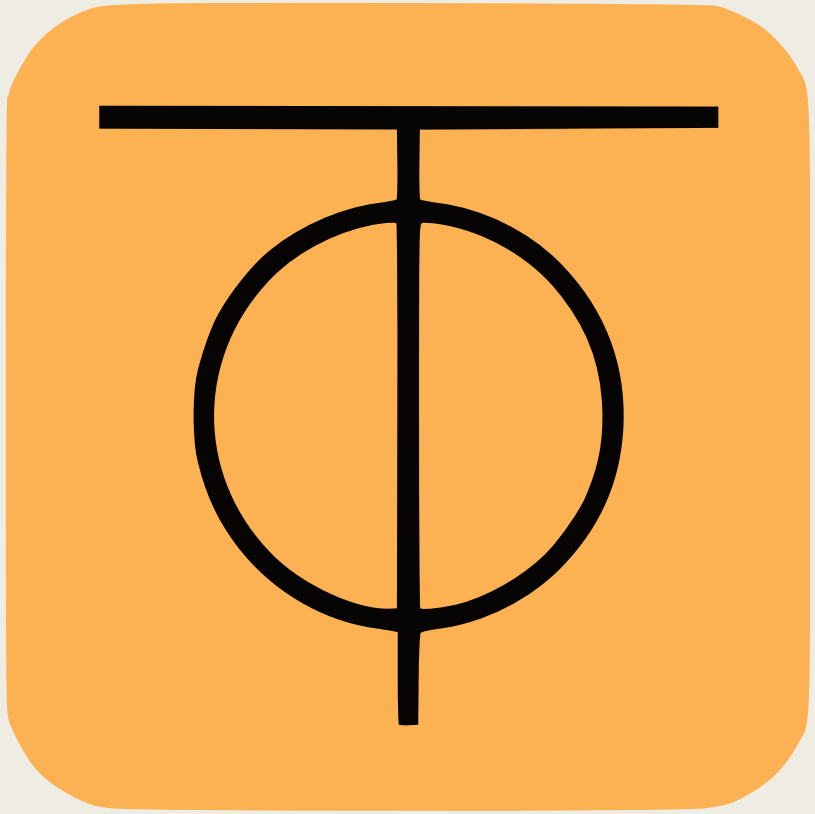
Availability

- If it breaks, no VPN for anyone
- If it's being used as a multi-site relay, breaks a lot more things

Not End-to-End Authenticated/Encrypted

- VPN Encryptor terminates the connection
- Services behind the VPN aren't guaranteed that the client is real without additional authentication

Solution: ZeroTier

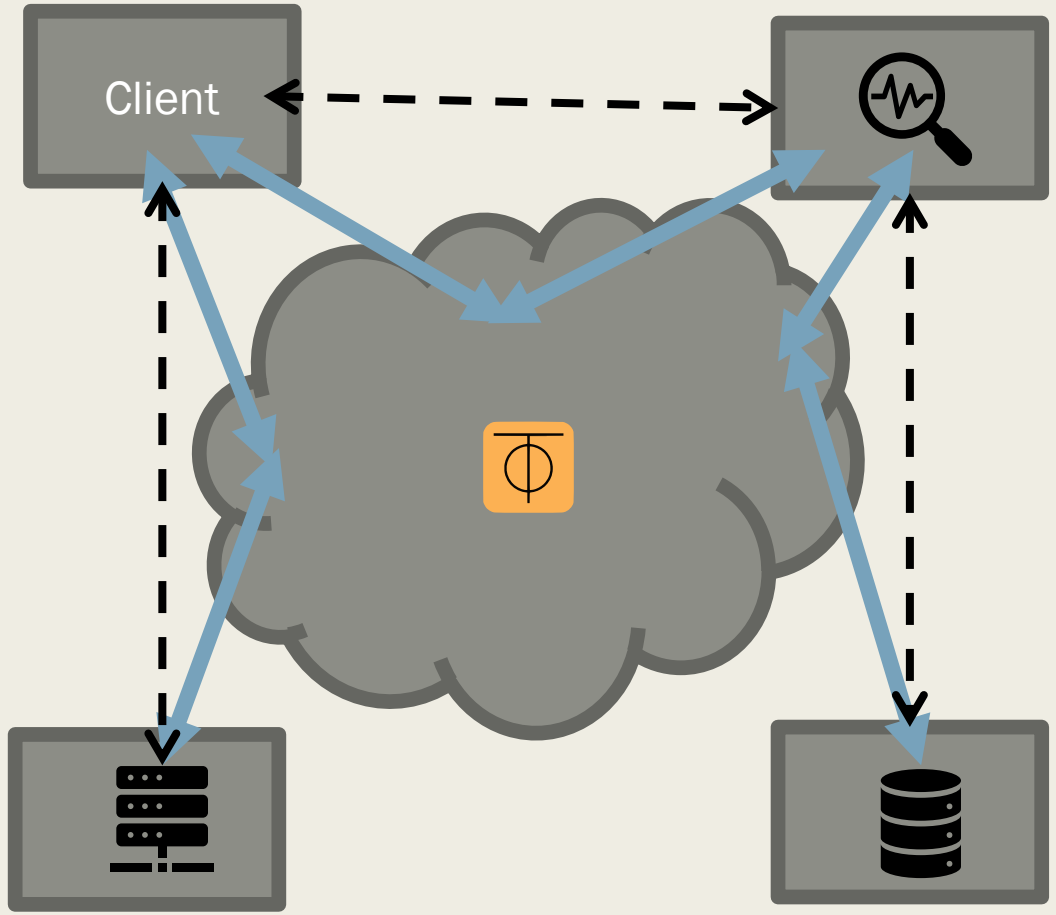


“A global Ethernet Switch for Planet Earth”

(It’s not the logo for a cult, I promise)

ZeroTier Basics

- Dynamic Peer-to-peer VPN
- All nodes (across all users) inhabit the same *global* address space
  - *Your address is your public key*
- Nodes (clients) join networks (if permitted) identified by the same address system
- Packets are inherently signed and encrypted using that public/private key
- Global servers handle client introductions, but all communication is direct
- Can bridge to a real network, centrally distribute route



Network Controllers and Rules

Controller

- Each network has a controller
- Controller has an address the same as a regular node’
- Network is addressed by controller address+identifier
- Controller signs your pubkey to allow you onto the network
- Controller also assigns IP addresses, security metadata

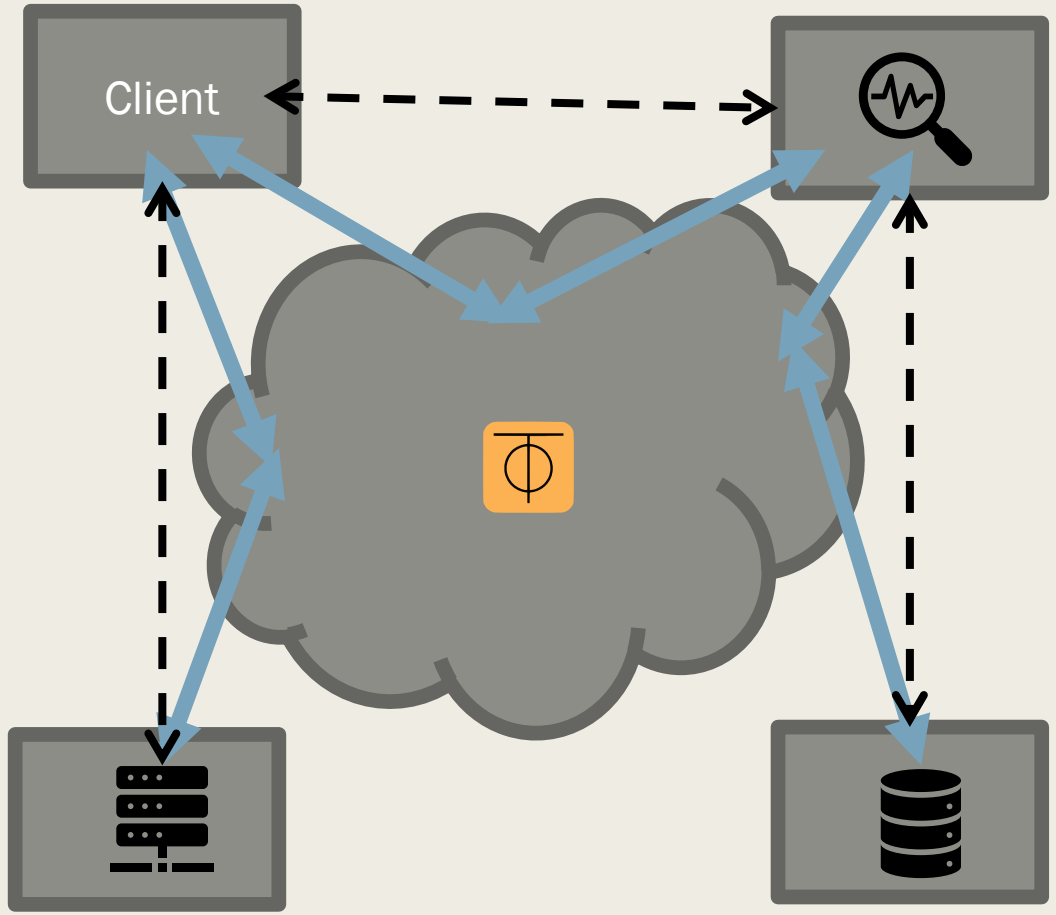
Rules

- Controller issues a global set of rules to all nodes
- Simplest form - akin to firewall rules
- Nodes enforce the rules automatically
  - *Your application won’t ever see disallowed traffic*

Basic Rules Example

1. Allow HTTPS from client to application to application
2. Allow CIFS from client to file server
3. Allow SQL from application to database
4. Allow established connections

Scaling? Traditionally subnets.



Tags

- Controller can assign tags to nodes
  - *When admitting to network, issue credential including tags*
- Tags can be numbers, bitfields, enumerations
- Can use them in rules
- Obvious use: VLANs
  - *Allow from tag x to tag x*
- Comparisons are allowed
  - *Allow where tag matches*
  - *Allow from higher tag to lower tag*
- Nodes can have an arbitrary number of tags, each tag only once
- If you have a tag called “department” that has “engineering” and “HR”, can’t have both
- What if you need greater flexibility?

Capabilities

- Tiny, named rulesets
- Issued to nodes when joining
- Evaluated by recipient with tags, global rules
- Peer-to-peer stateless cryptographic proof of authorization on every packet

Example:

- Tag for department
- Capability for departmental file server access
  - *Attached rule: allow CIFS if department tag matches*
- Can selectively grant access to file server
- Need cross department access? Make a capability for it, attach to needed node

Capabilities Example

Capability – FileServerAccess:

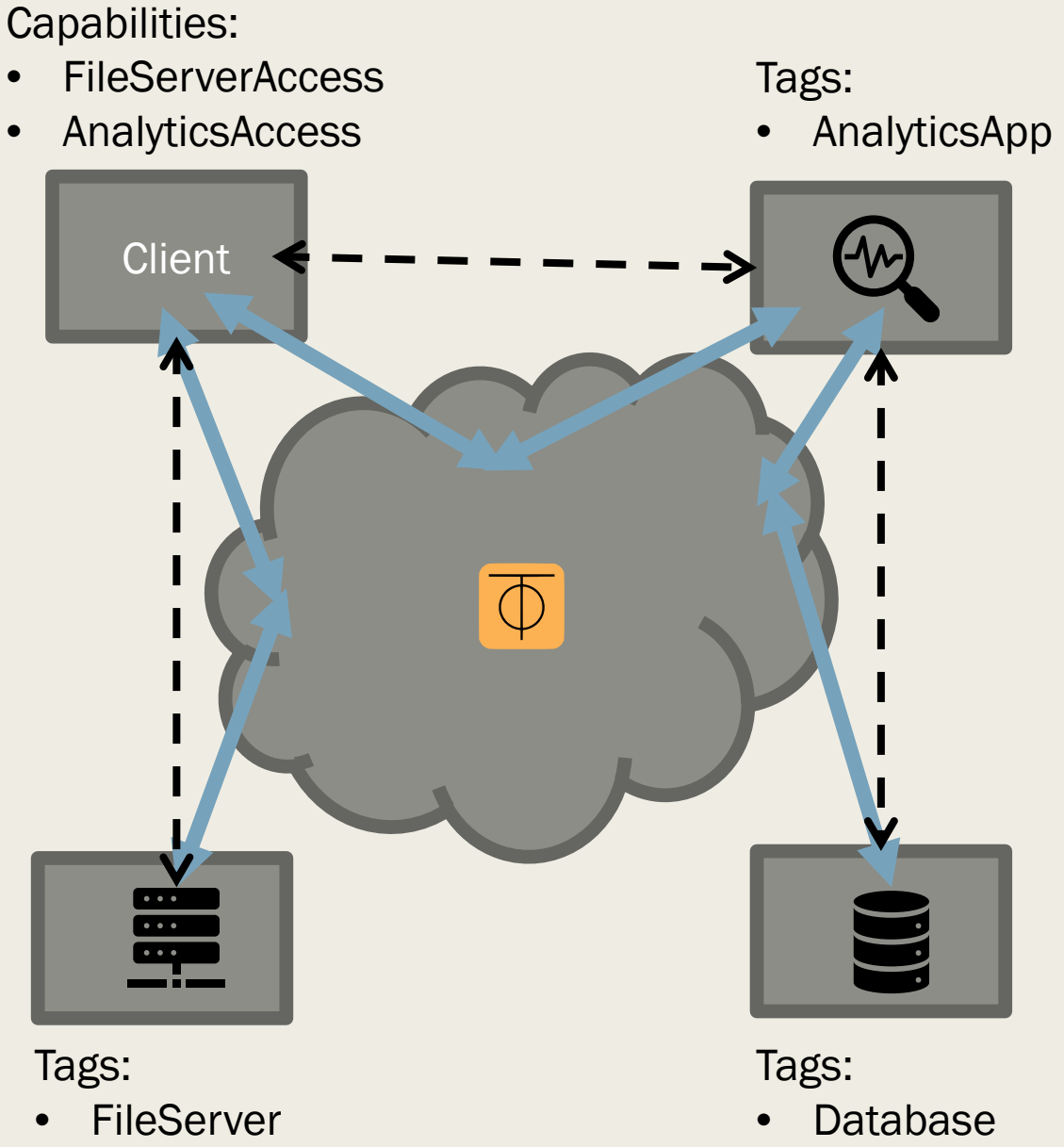
1. Allow CIFS to tag: FileServer

Capability – AnalyticsAccess:

1. Allow HTTPS to tag: AnalyticsApp

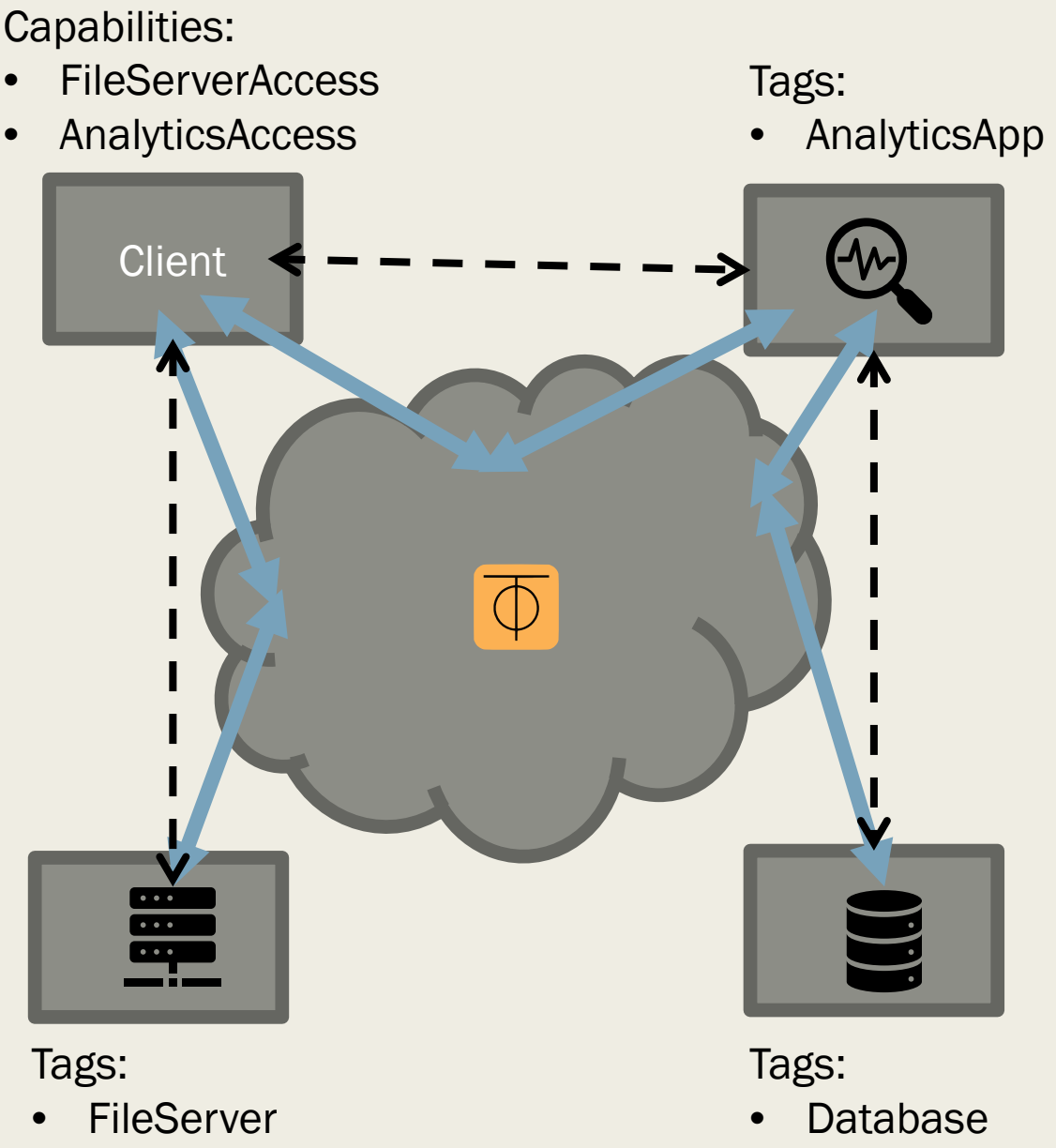
Global Rules:

1. Allow tag: AnalyticsApp to tag: Database
2. Allow established connections



What did we gain?

1. Perfectly granular controls over all traffic
2. All traffic is *guaranteed* to adhere to firewall rules without redundancy/anti-spoofing measures
3. All traffic is inherently encrypted, authenticated
4. Adding new servers is easy – failover analytics app gets the same tag and that's it
5. "Zero Trust" network – no perimeter, so no perimeter breaches



Downsides

- Main one: slightly immature, getting there
  - *Some issues on Windows with firewalls*
  - *Android app likes to disconnect for no reason*
- Requires internet access to bootstrap unless you do some extra self hosting
- Fully leveraging it requires you to pretty much throw out your existing network
- “Least friction” path is to use their central service for management
  - *Not free after 100 devices*
  - *Gives third party network control*
  - *Offer selfhosted version, but price is “call for quote”*
- You *can* host your own controller, but the documentation is sparse and there’s no UI (just a JSON API)
  - *Third party frontend exists, but is incomplete*

QUESTIONS?

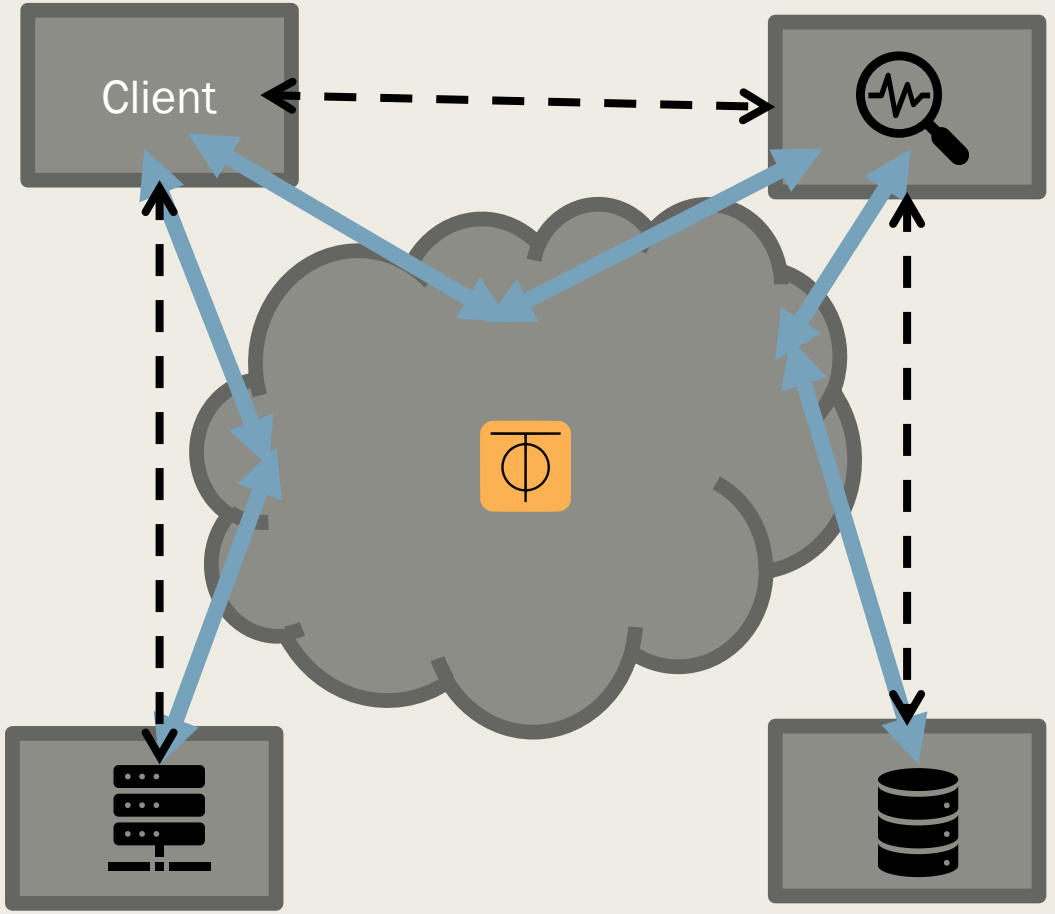
Thank you

Capabilities:

- FileServerAccess
- AnalyticsAccess

Tags:

- AnalyticsApp



Tags:

- FileServer

Tags:

- Database