# Open source self-hosted mesh VPN with IPv6!

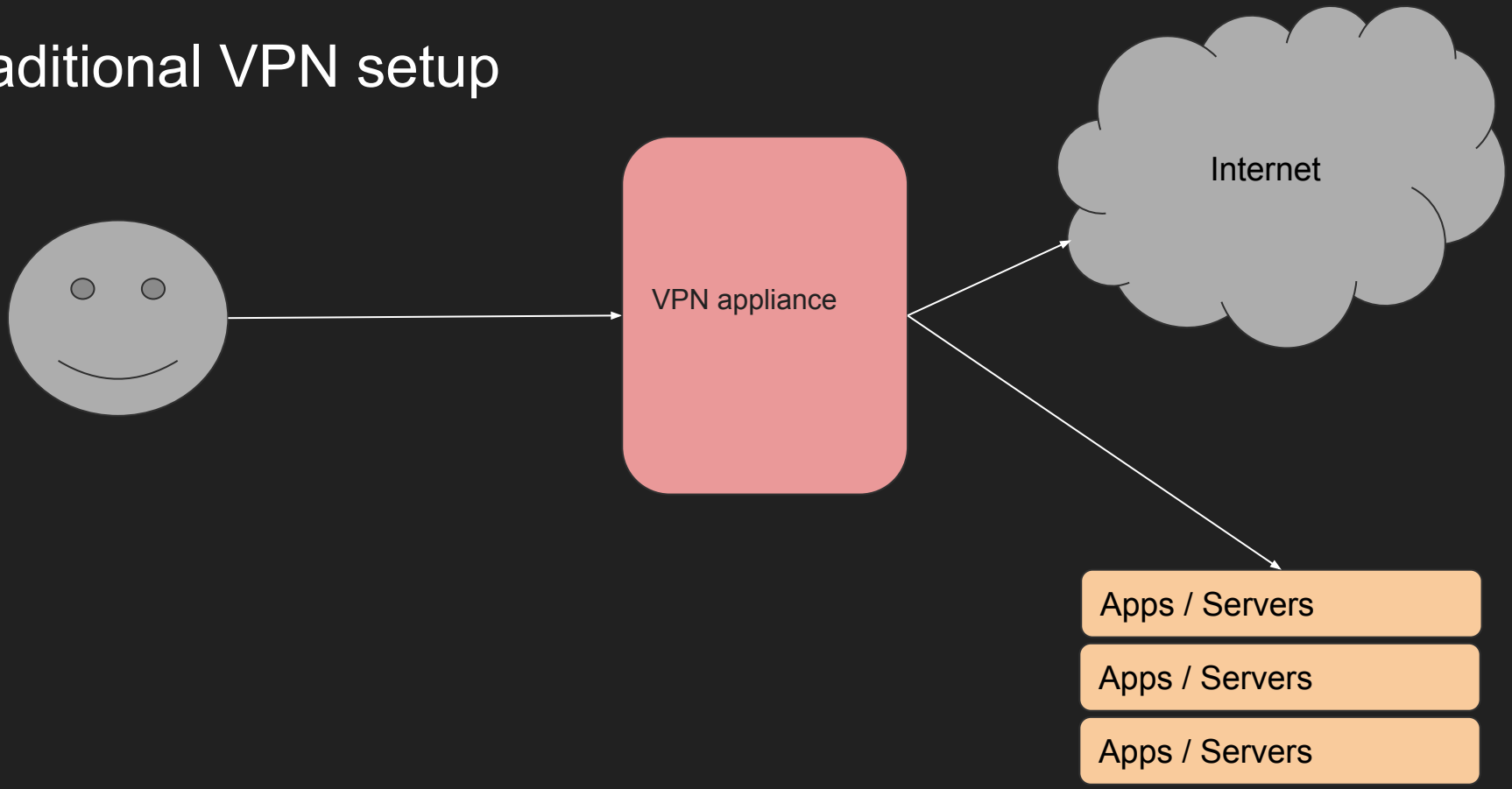
Anurag Bhatia, Hurricane Electric

# What is VPN?

# What is VPN?

- Virtual Private Network
- “Virtual” = Virtual in nature and an overlay on top of existing “physical networks”
- Can be using different technologies on layer 2 or layer 3
- Not always but mostly encrypts traffic
- Layer 3 / IP based VPN have become very popular over time
- Can be point to point or mesh or a mix

# Traditional VPN setup

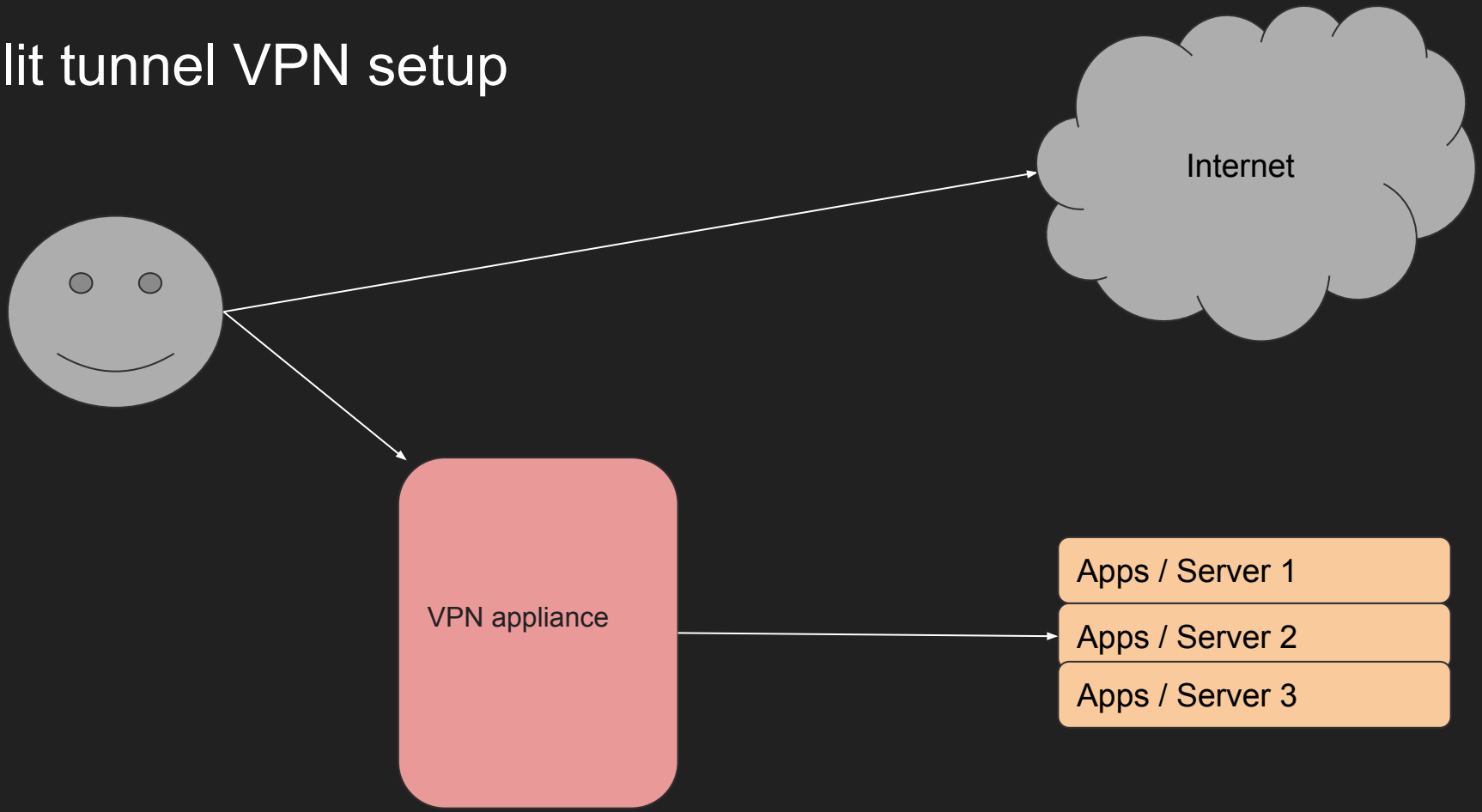


# Problems in traditional setup

- VPN server can be a bottleneck as all traffic passes through it
- Latency to the internet becomes an issue specially if server is located far away
- Does not play well with locally hosted CDNs of content players of the network operators

# Split tunnel setup...

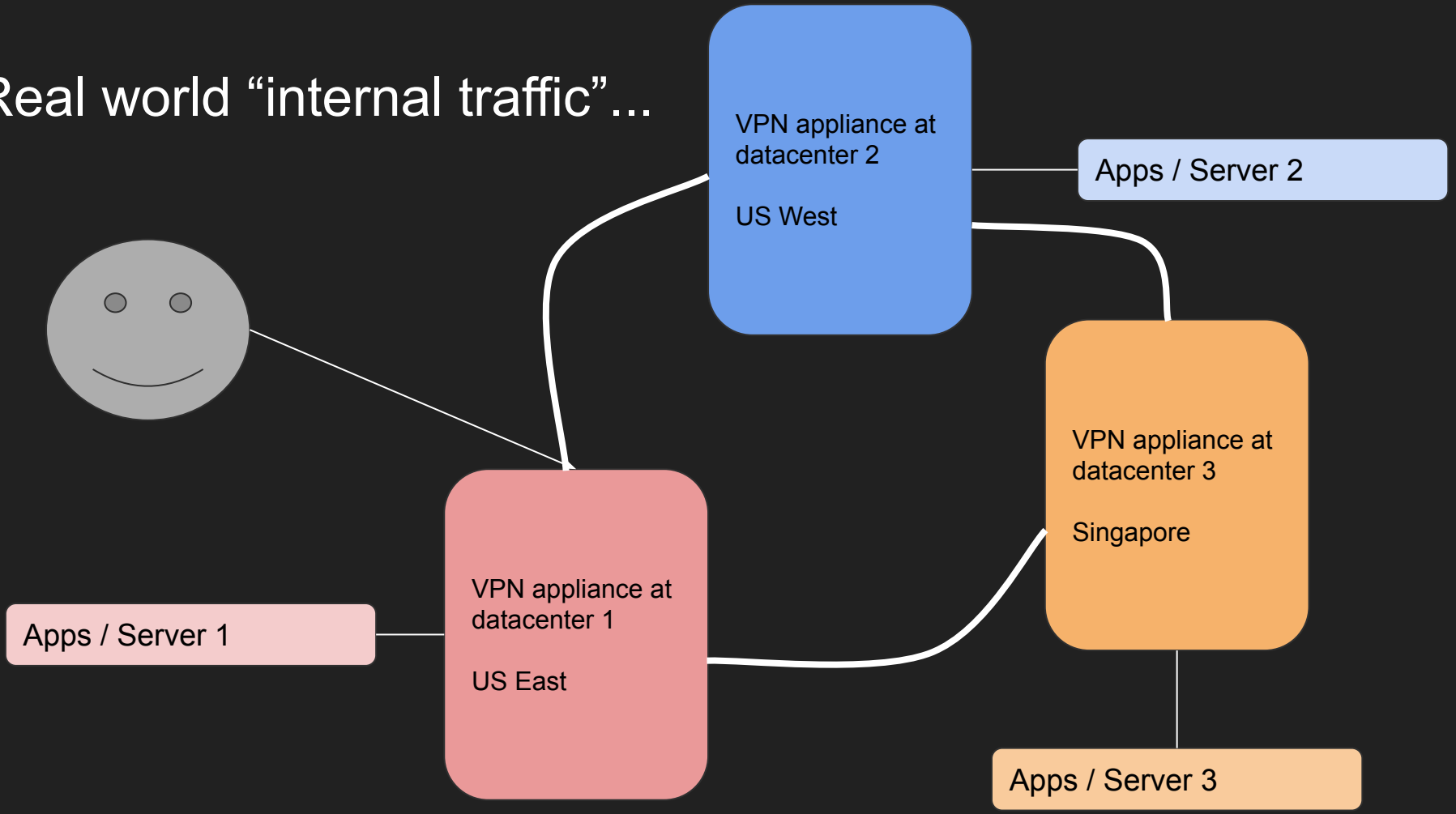
# Split tunnel VPN setup



# Good and bad with split tunnels

- Do not slow down non-internal i.e internet traffic
- Save bandwidth requirement, latency & management of VPN gateway
- Inject only required routes but dependending on type of VPN these routes can be hardcoded in client config or pushed on the fly
- Better than old tunnels but still following hub-spoke model for internal traffic, have issues in scaling up where internal apps are spread across different datacenters, cloud players...

# Real world “internal traffic”...

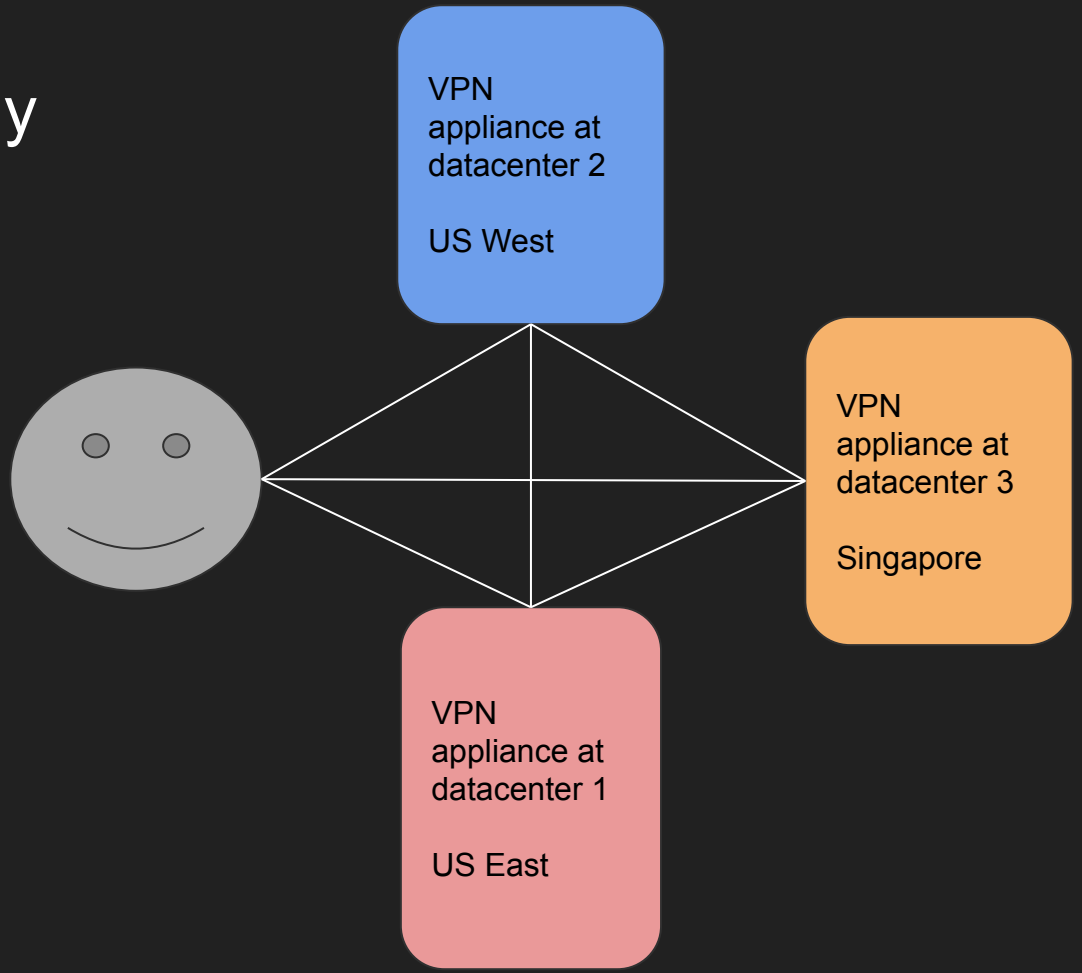


# About multiple VPN appliances....

- Usually routing over the internet has a much better path then transit a bunch of networks via appliances
- The best and ideal setup is to connect all clients to all gateways but that makes config harder
- Becomes an administrative issue to ask all clients to update config if a new region comes up
- Some clients do not even support maintaining more than one end point

# Ideal and most efficient topology...

# Ideal VPN topology




Full mesh....just like how devices would talk over the internet

# How many tunnels?

# Config complexity

1. Number of tunnels = n * (n-1) / 2 i.e for 4 devices, (4 x 3) / 2 = 6 tunnels and thus (4 * 3) 12 “endpoints” to configure for VPN

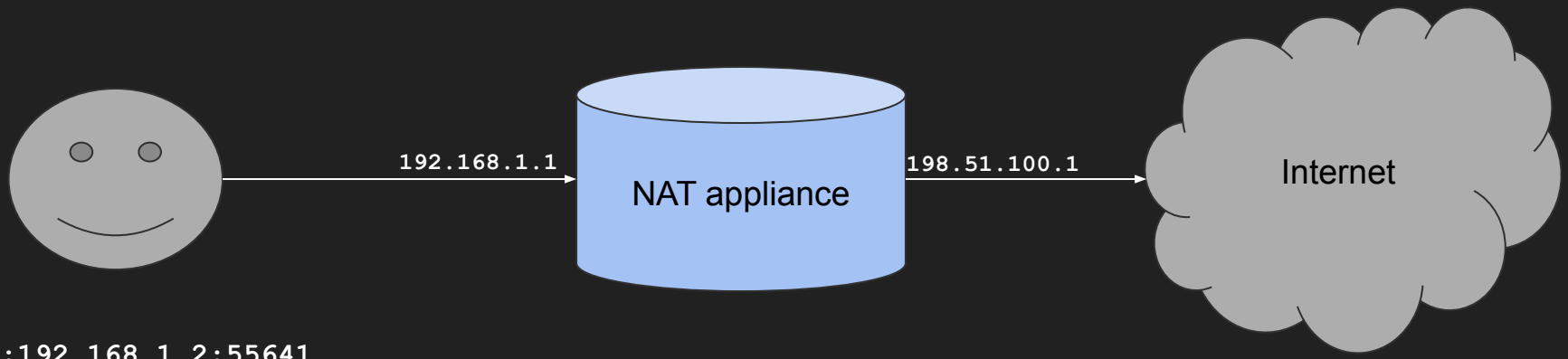
2. 12 endpoints to configure for firewall rules (if not 12, atleast 6 so that one side can initiate connection)

3. What if some clients have IPv4, some have IPv6? Setup multiple tunnels or stick to IPv4 and run on old outdated protocol?

4. What about client to client communication who are behind NATs?

# Let’s talk about the elephant in the room....

# NAT - Network Address Translation




`Src:192.168.1.2:55641`

`Dst:216.218.236.2:80`

`Src:198.51.100.1:55641`

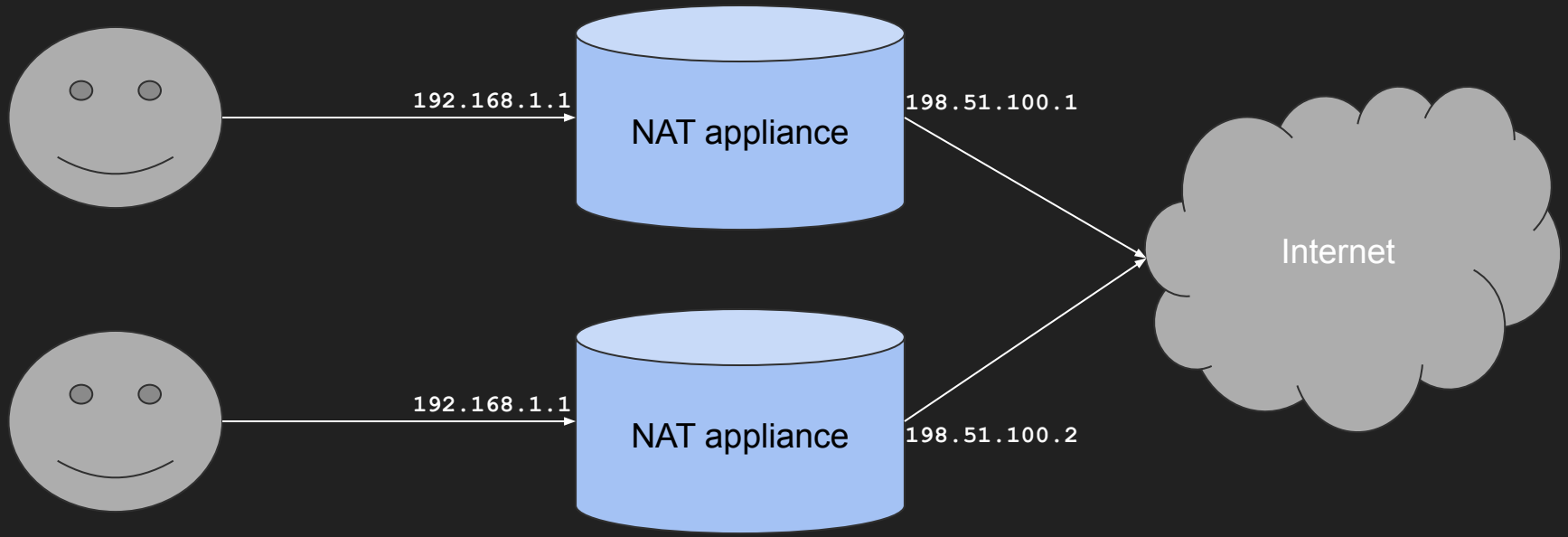
`Dst:216.218.236.2:80`

# Misc points about NAT

- Helping in keeping internet running while operators deploy IPv6 under acute IPv4 shortage
- There are max-theoretical limits due to number of ports
- Makes end to end connectivity much harder due to use of double NAT i.e one NAT by carrier (CGNAT) and one at the end user
- Is supposed to (fingers crossed) disappear eventually once everyone supports IPv6
- Client server communication is easily possible when client behind NAT initiates a connection with server which is not behind NAT

Is peer to peer communication even possible when clients are behind NAT?

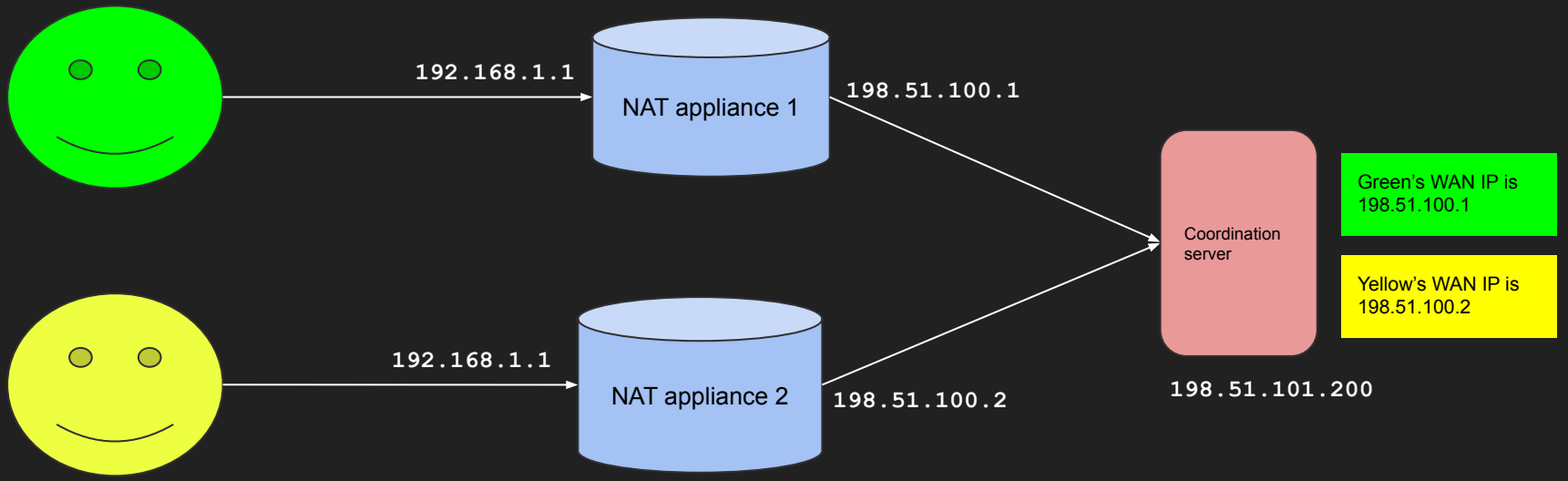
# NAT - Network Address Translation



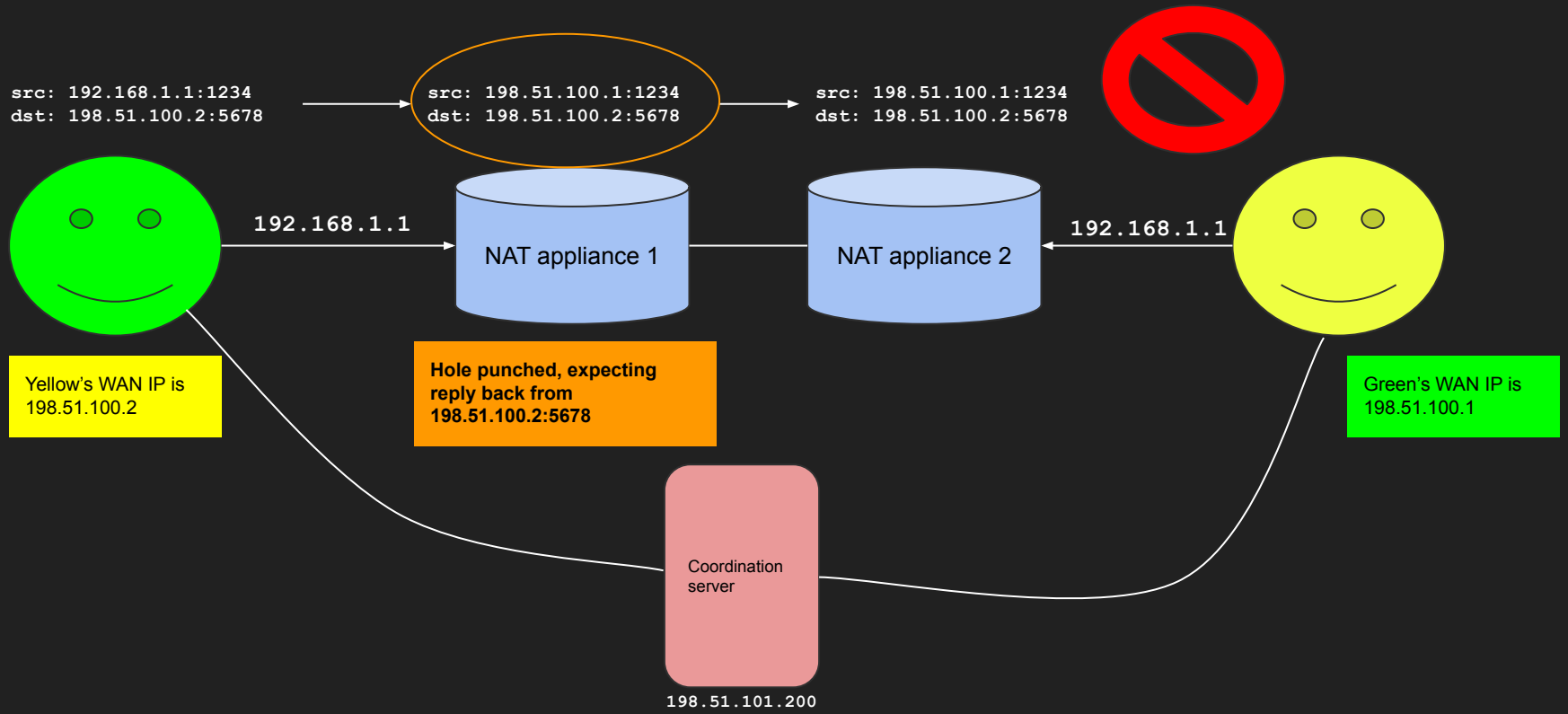
# Yes, more complex to setup but possible!

# Understanding NAT traversal

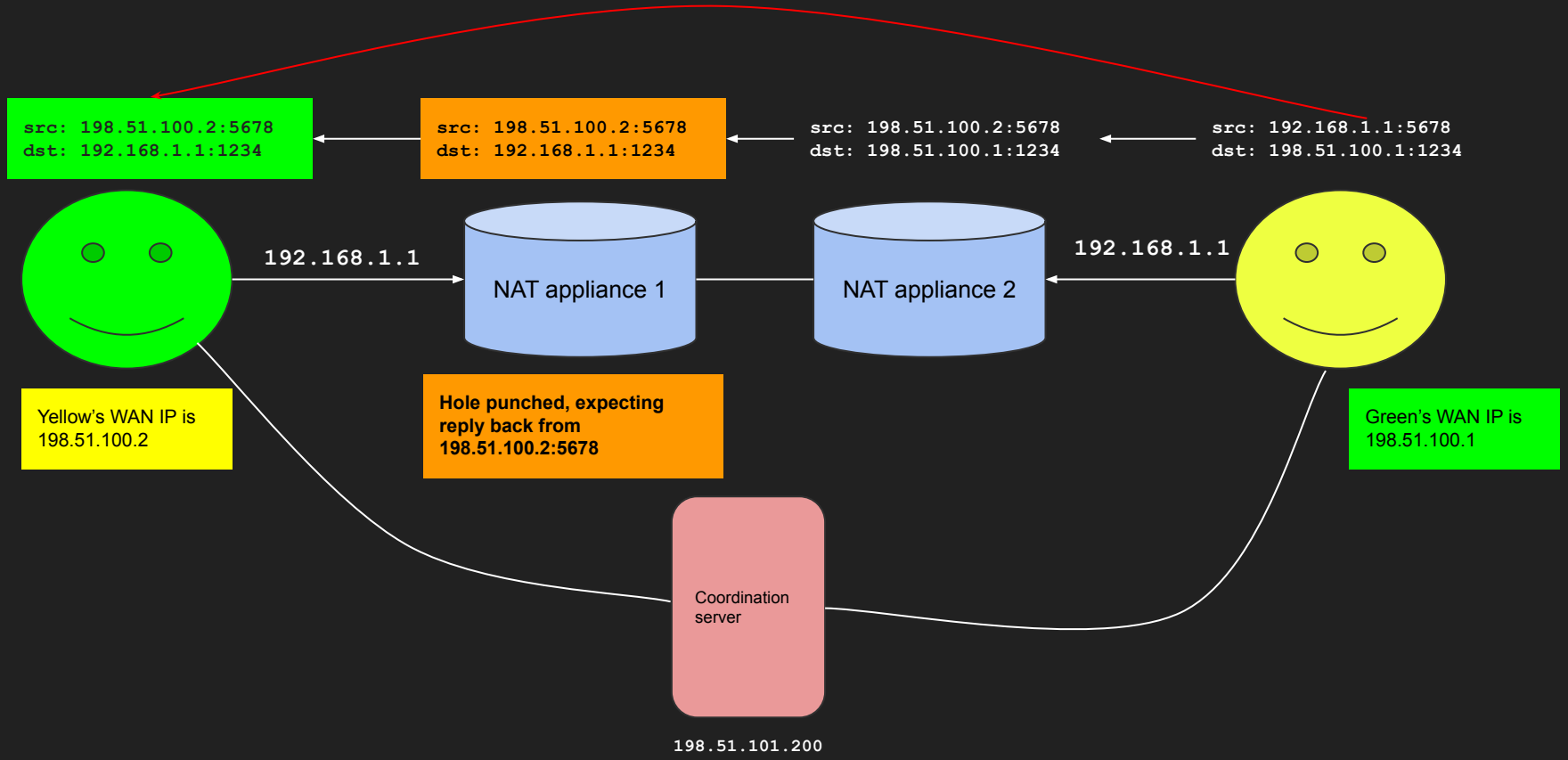
# Typical setup - both users behind NAT



# Typical setup - both users behind NAT - Green contacts Yellow



# Typical setup - both users behind NAT - Yellow contacts Green



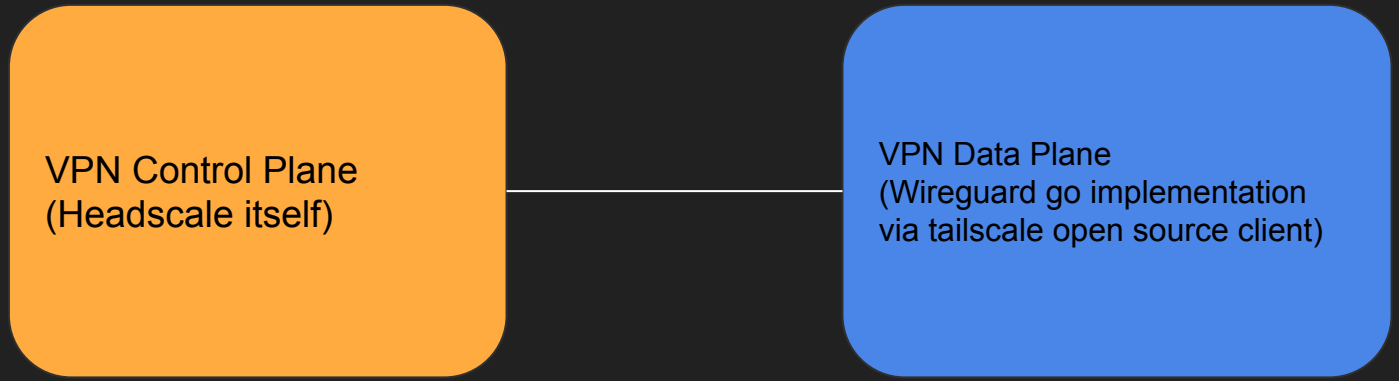
Same in reverse i.e get two way UDP communication working...

# Introduction to headscale

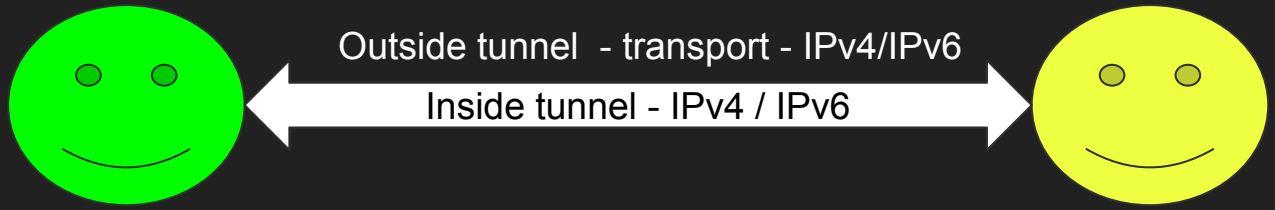
# Headscale

- Open source implementation of (closed source) tailscale control server
- Is self-hosted on a public IP ideally behind https
- VPN clients connect to headscale & get authenticated, they share their public keys with the control server
- Private keys never leave client
- Headscale shares config with everyone - creating mesh impact
- Traffic is peer to peer in majority of cases & control traffic is in few bits per second
- System includes concept of DERP servers to manage cases where NAT traversal is impossible
- Supports “advertisement of prefixes” by a participating VPN client with “approval system”

# Headscale


VPN Control Plane
(Headscale itself)
VPN Data Plane
(Wireguard go implementation
via tailscale open source client)

# Working of IPv6 in Headscale




Uses IPv4 - 100.64.0.0/10

Uses IPv6 - fd7a:115c:a1e0::/48

# Working of IPv6 in Headscale

- IPv6 is preferred on the transport (outside tunnel) as long as both endpoints have IPv6 available
- Dual stack available inside the tunnel. IPv4 and IPv6 works inside the tunnel irrespective of available protocol on transport
- Enables migration of transport to IPv6 instantly & avoiding CGNATs while it takes time to migrate internal network to IPv6

# Demo video...

4 devices - Singapore, Amsterdam, London and Manila (my laptop!)

4 x (4-1) = 12 endpoint configurations

4 x (4-1)/2 = 6 VPN tunnels

# References


1. https://tailscale.com/blog/how-nat-traversal-works
2. https://en.wikipedia.org/wiki/STUN
3. https://github.com/juanfont/headscale
4. https://tailscale.com/blog/how-tailscale-works/#encrypted-tcp-relays-derp
5. https://github.com/netbirdio/netbird
6. https://www.wireguard.com

# Questions?

anurag@he.net