

The case for IPv6-only data centres

...and how to pull it off in today's IPv4-dominated world

Tore Anderson

Redpill Linpro AS

IPv6 Business Conference, Zürich, June 2013

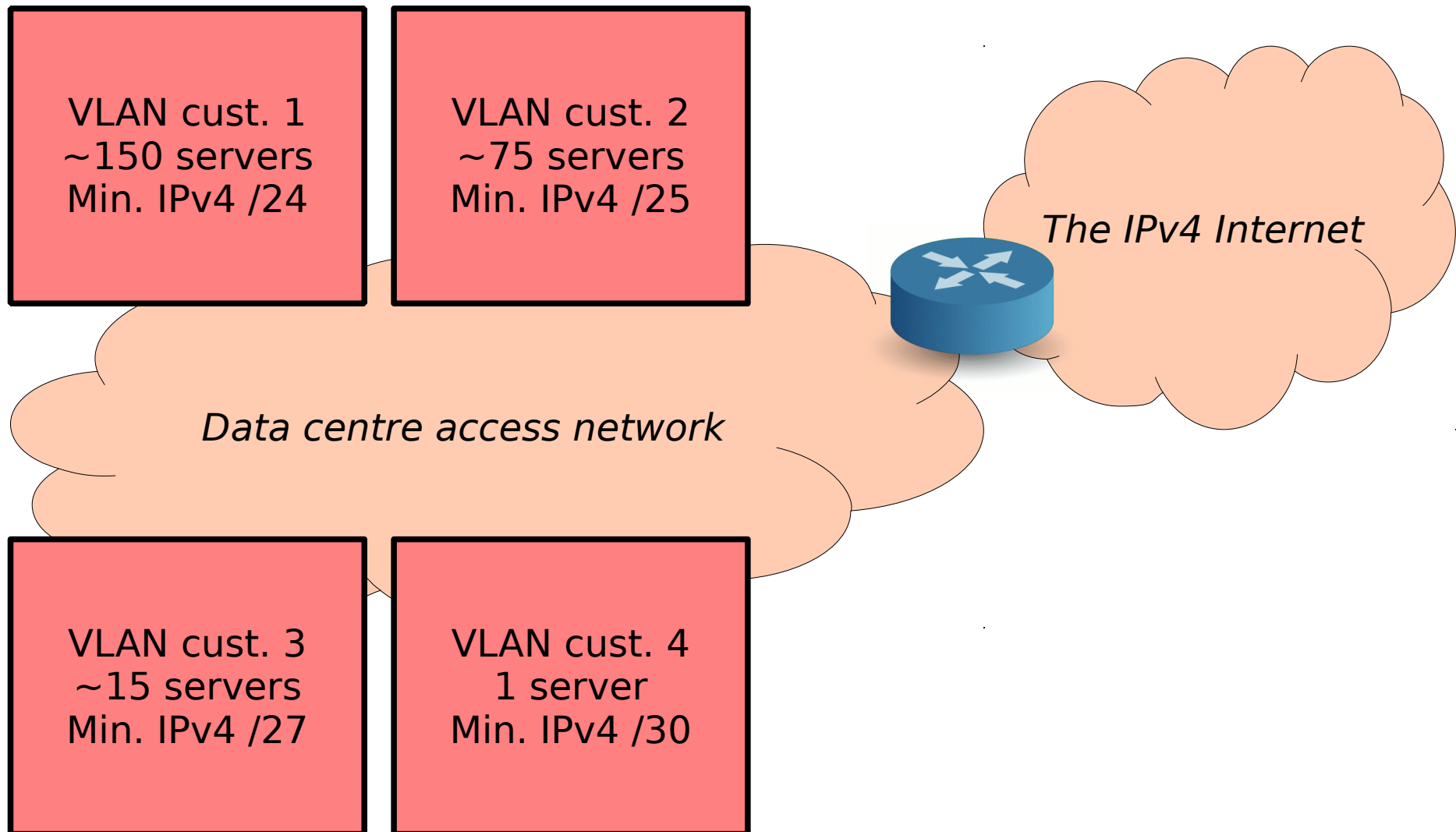


**IPv6 Business
Conference**

Organized by
SWISS
IPv6
COUNCIL

| 2013
June, 11

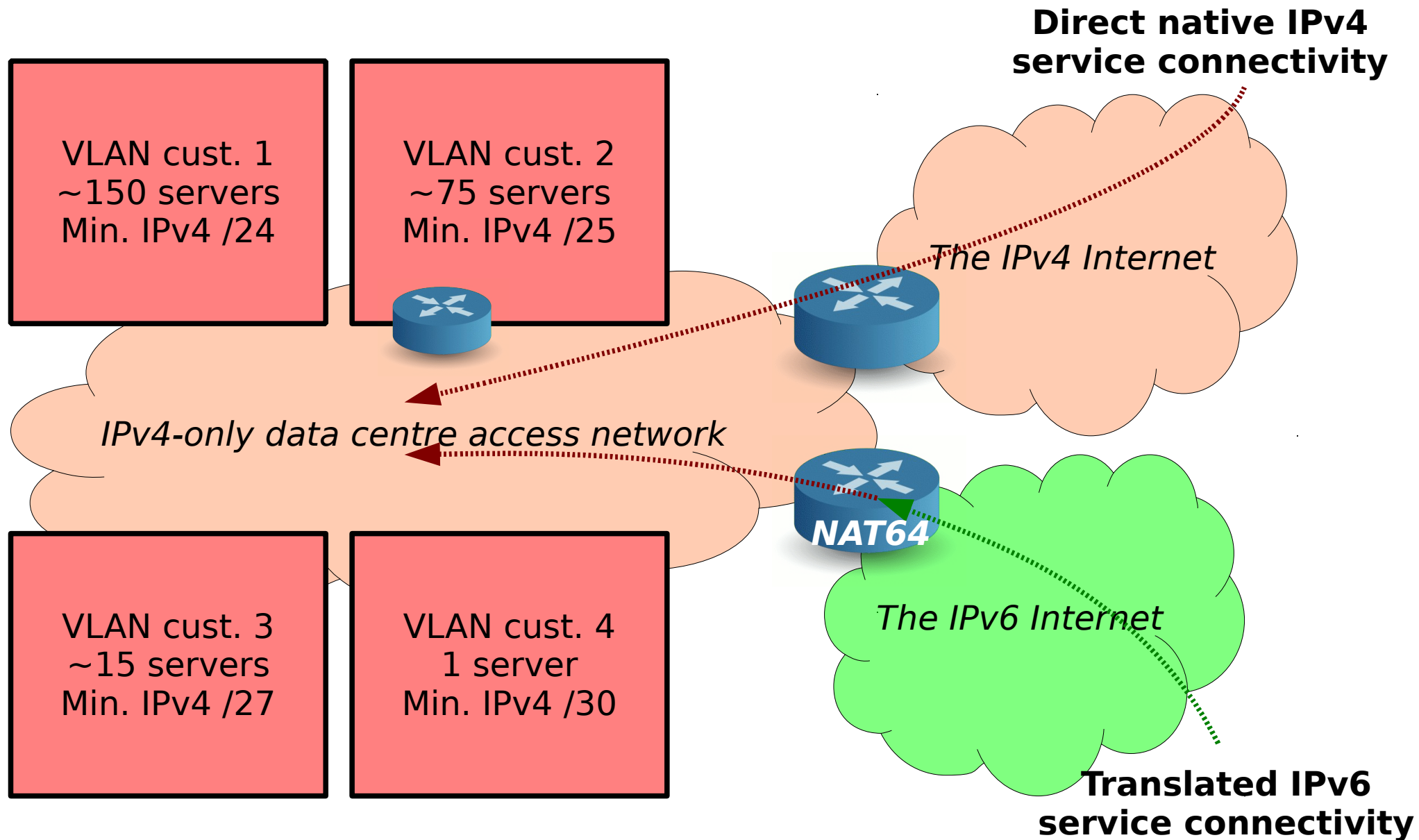
Traditional DC topology



Challenges ahead

- IPv6 deployment
- IPv4 depletion

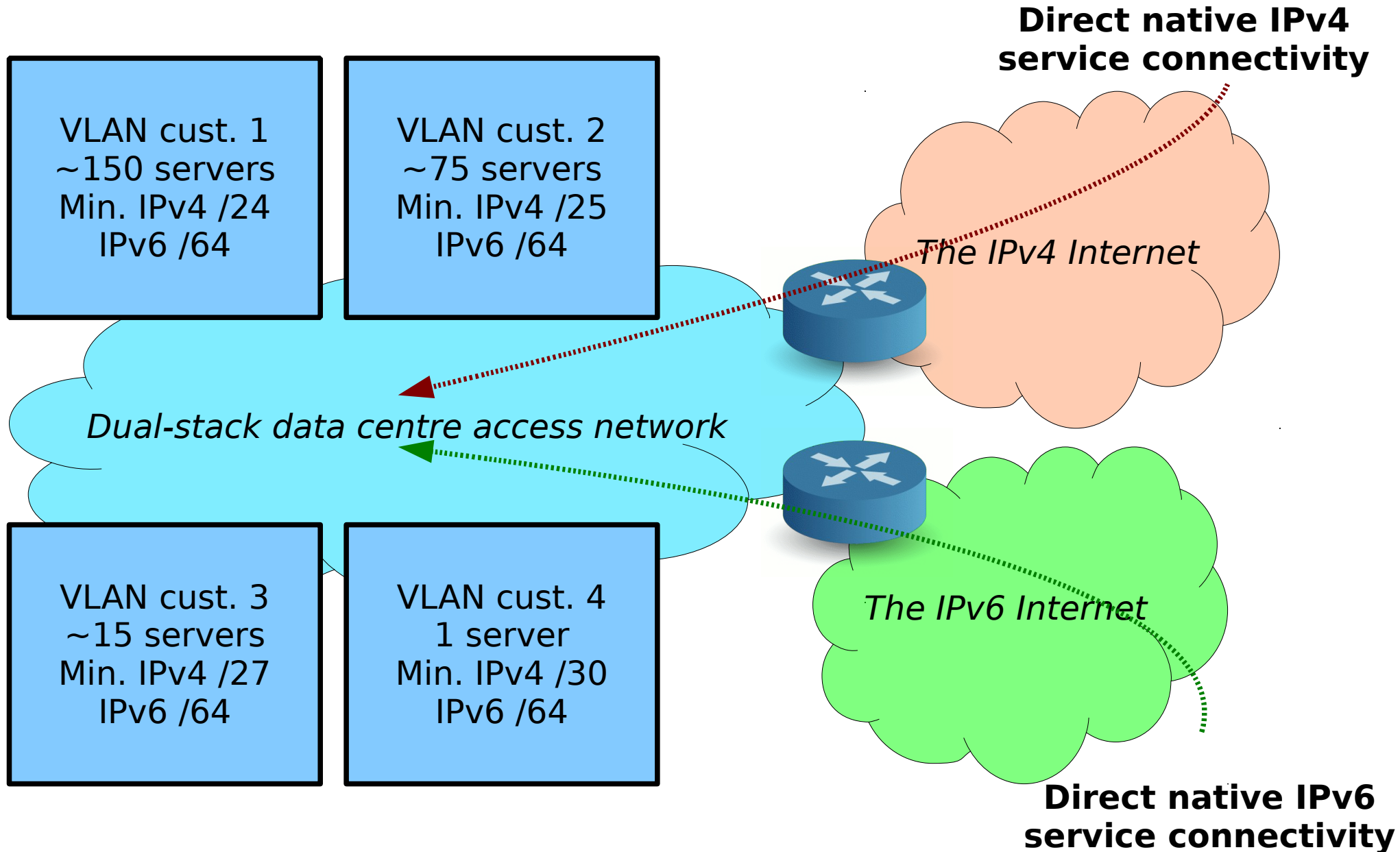
NAT64 (or proxies)



NAT64 (or proxies)

- The good:
 - Services now accessible over IPv6
- The bad:
 - Requires stateful devices - expensive, hard to scale, limits routing flexibility, and vulnerable to DoS attacks
 - Failures/fail-over breaks all sessions
 - Obscures source IPv6 address of user
 - Does not help with IPv4 depletion

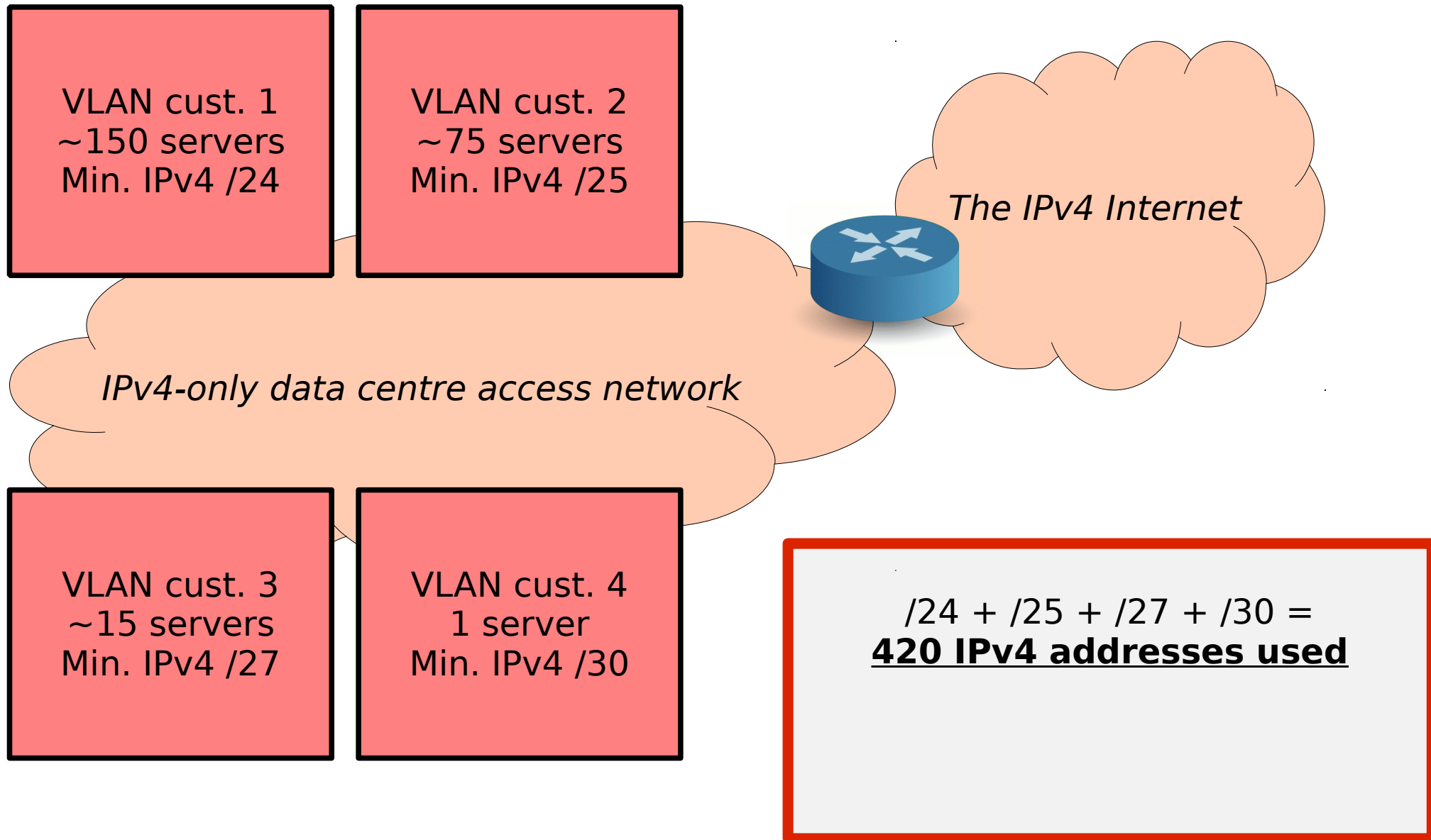
Dual stack



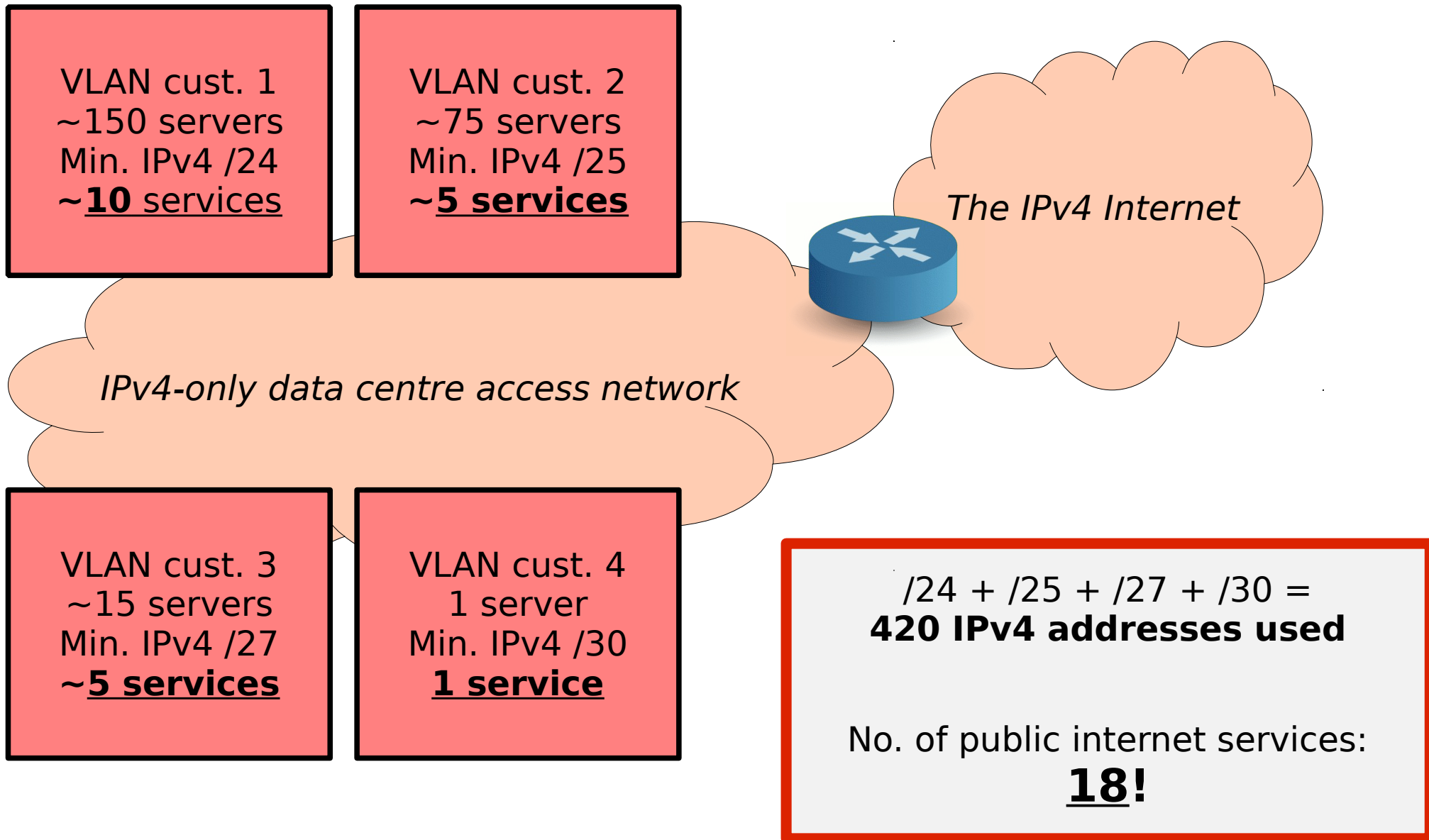
Dual stack

- The good:
 - Services now available over IPv6
- The bad:
 - Does not help with IPv4 depletion
 - Greatly increases complexity
 - Doubles the amount of ACLs, monitoring, troubleshooting, possible failure scenarios, staff training, documentation, testing, etc.

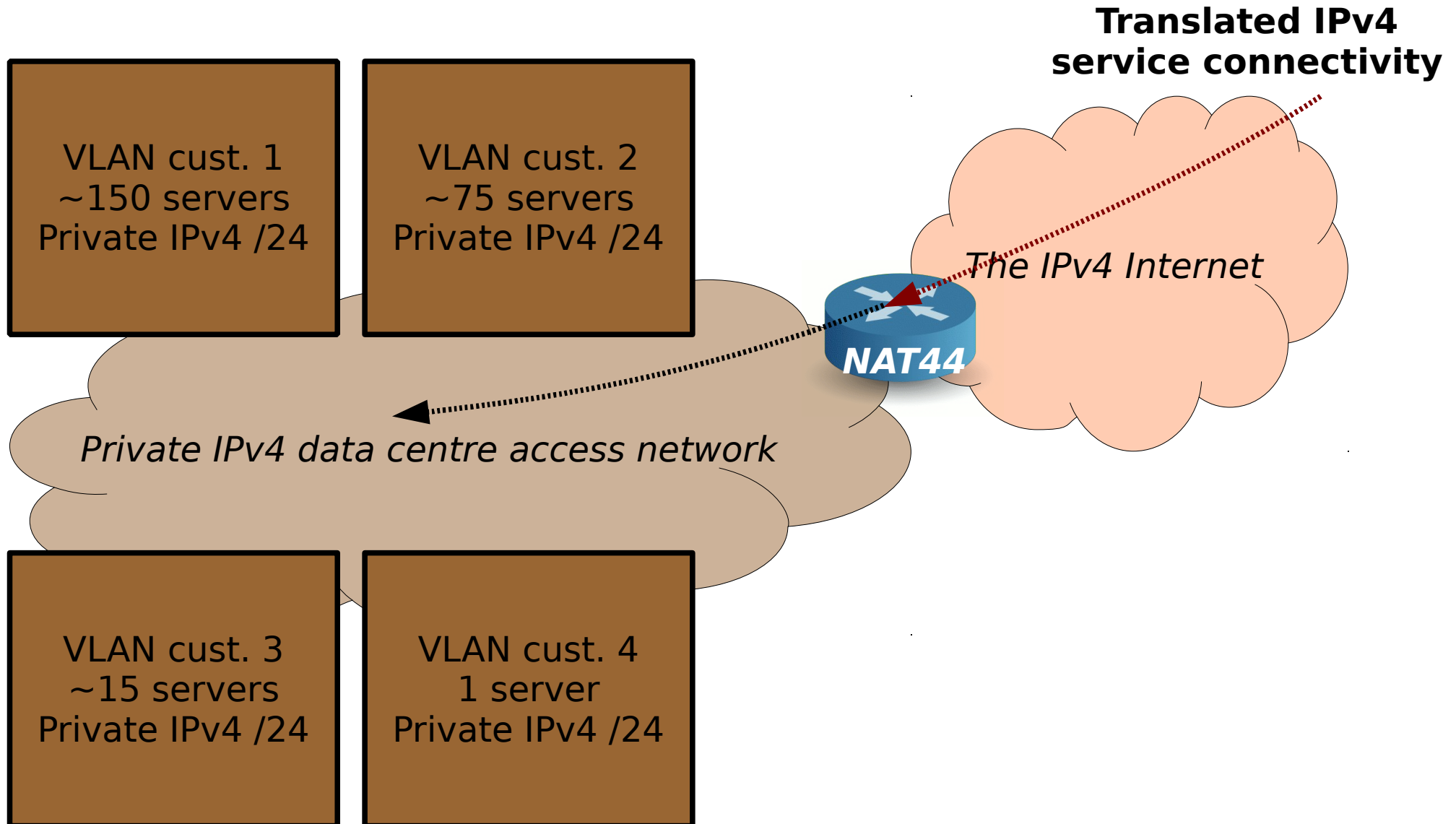
Examining IPv4 usage



Examining IPv4 usage



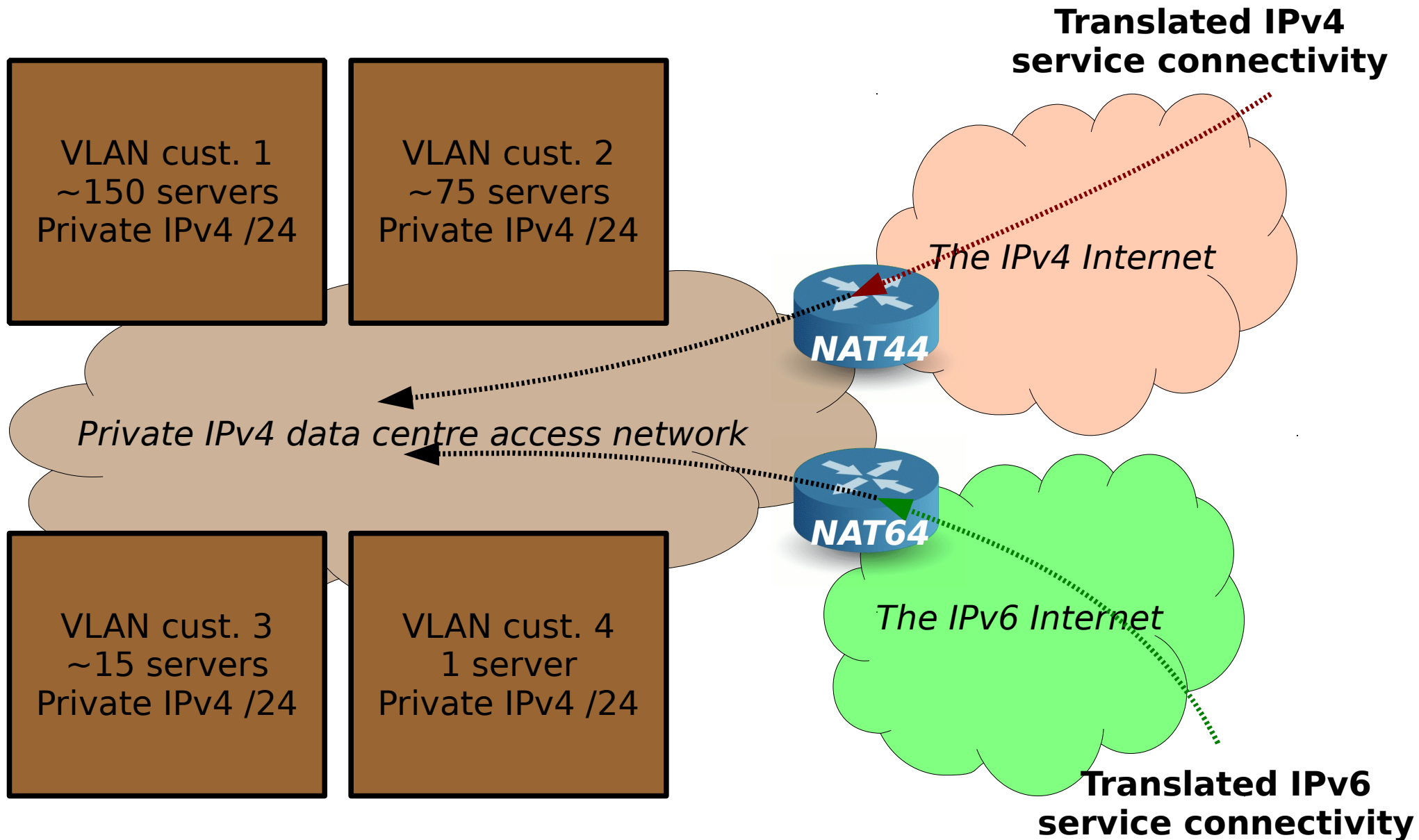
Private IPv4 + NAT44/proxies



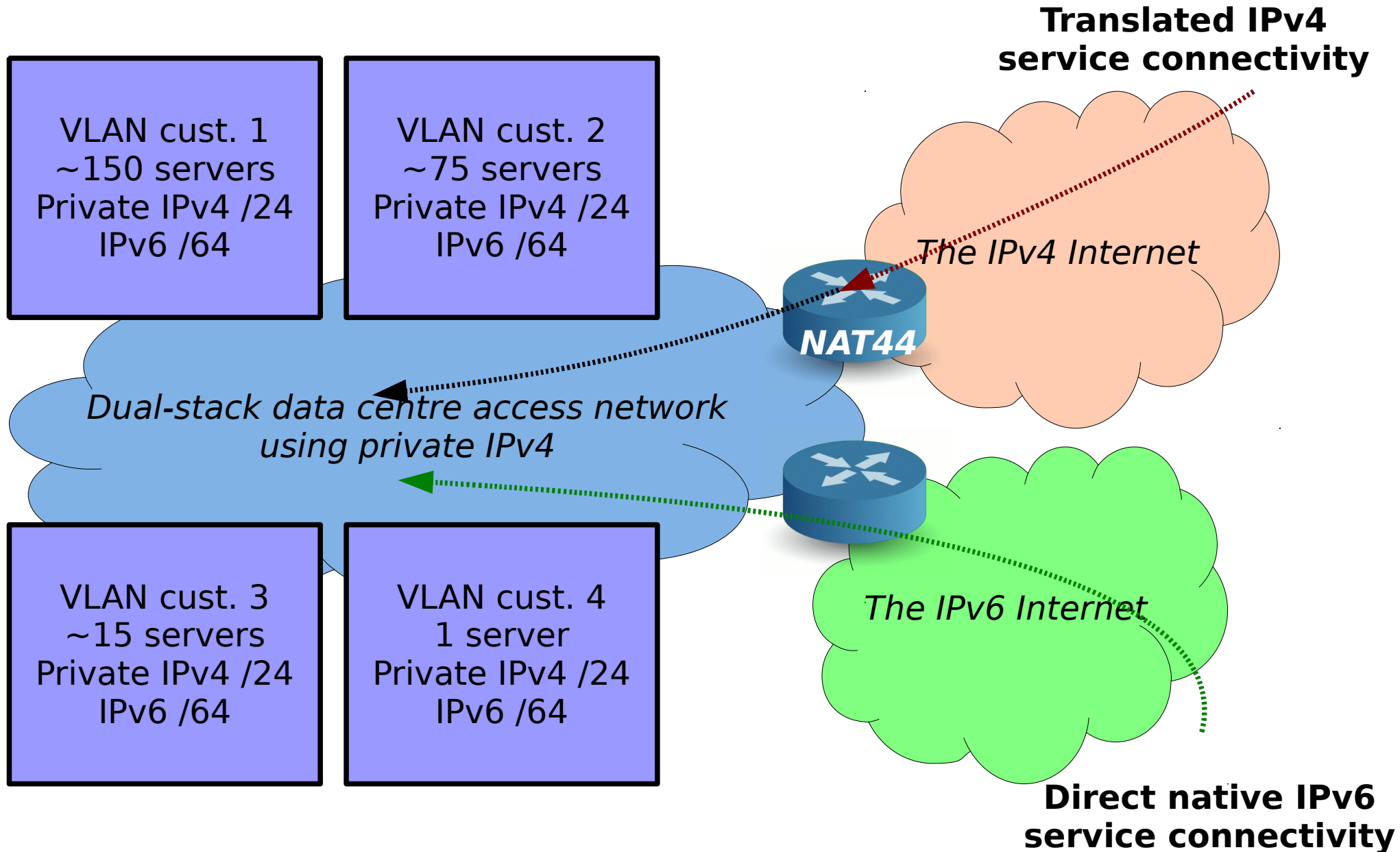
NAT44/proxies

- The good:
 - Solves IPv4 depletion, use only 1 public address per public service
- The bad:
 - Requires stateful devices - expensive, hard to scale, limits routing flexibility, and vulnerable to DoS attacks
 - Failures/fail-over breaks all sessions
 - May obscure source IPv4 address of user
 - Does not help with IPv6 deployment

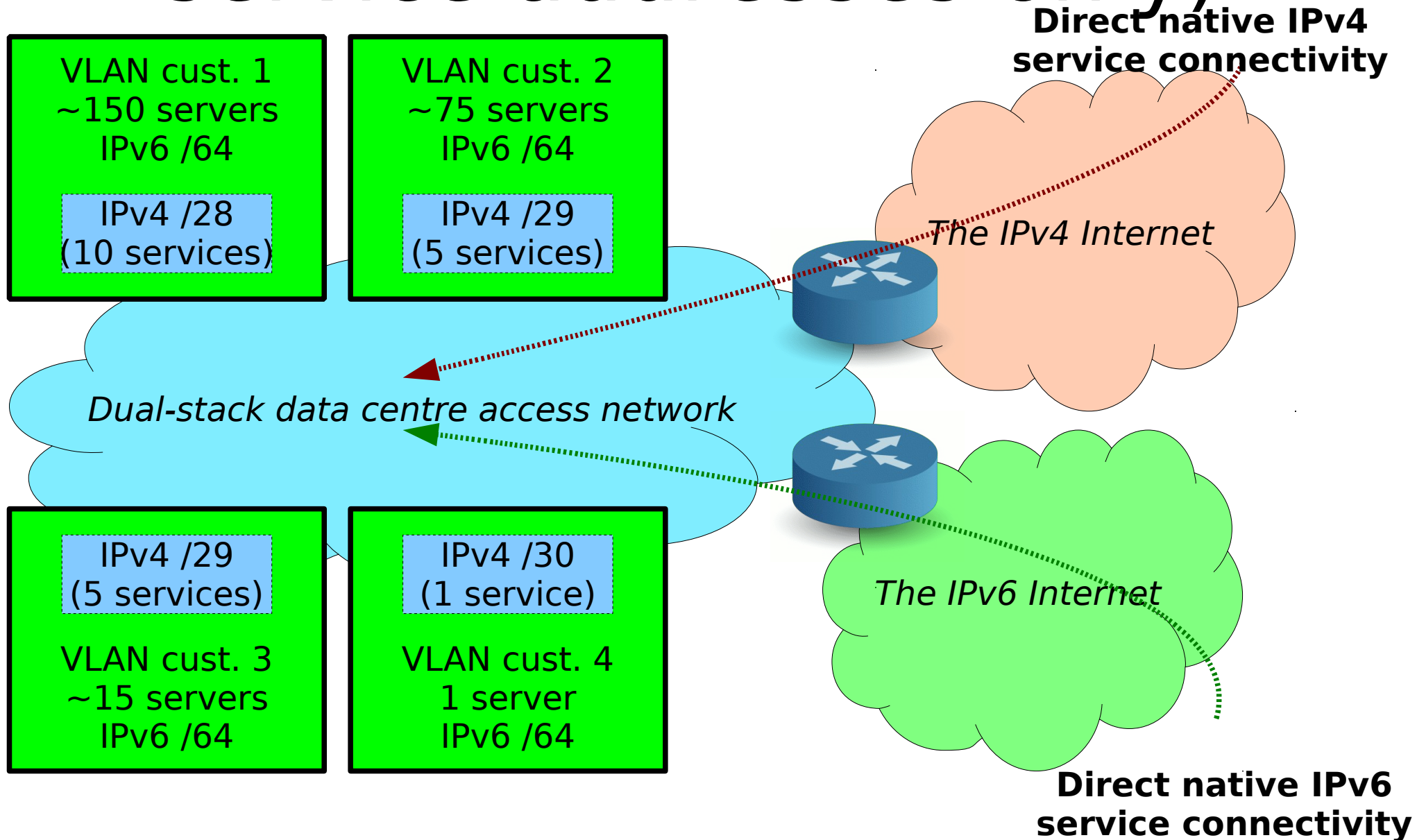
Private v4, NAT44 and NAT64



Native v6, private v4, NAT44



Partial dual stack (IPv4 for service addresses only)



Mix and match solutions?

- The good:
 - Solves IPv4 depletion, use only 1 public address per public service (or close to it)
 - Ensures IPv6 service availability
- The bad:
 - You're still dependent on IPv4 – expect more migration projects down the road
 - You require NAT44, NAT64, or proxies – all stateful devices, **or**:
 - Accept the operational complexity of running two IP versions in parallel

Deploying IPv6 incrementally

- IPv4-only

- ➔ IPv4-only + IPv6 via NAT/proxy

- ➔ Dual-stacked public frontend, IPv4 BE

- ➔ Full dual-stack

- ➔ Dual-stacked public frontend, IPv6 BE

- ➔ IPv6-only + IPv4 via NAT/proxy

- ➔ IPv6-only

What's possible today?

- IPv4-only

- IPv4-only + IPv6 via NAT/proxy

- Dual-stacked public frontend, IPv4 BE

- Full dual-stack

- Dual-stacked public frontend, IPv6 BE

- IPv6-only + IPv4 via NAT/proxy

- ~~→ IPv6 only~~

***Only a single-digit percentage of end-users world-wide have IPv6!
Switzerland is exceptionally high, but still, IPv6-only content ... no way***

Let's take a shortcut...

- IPv4-only

~~IPv4-only + IPv6 via NAT/proxy~~

~~Dual-stacked public frontend, IPv4 BE~~

~~Full dual-stack~~

~~Dual-stacked public frontend, IPv6 BE~~

IPv6-only + IPv4 via NAT/proxy

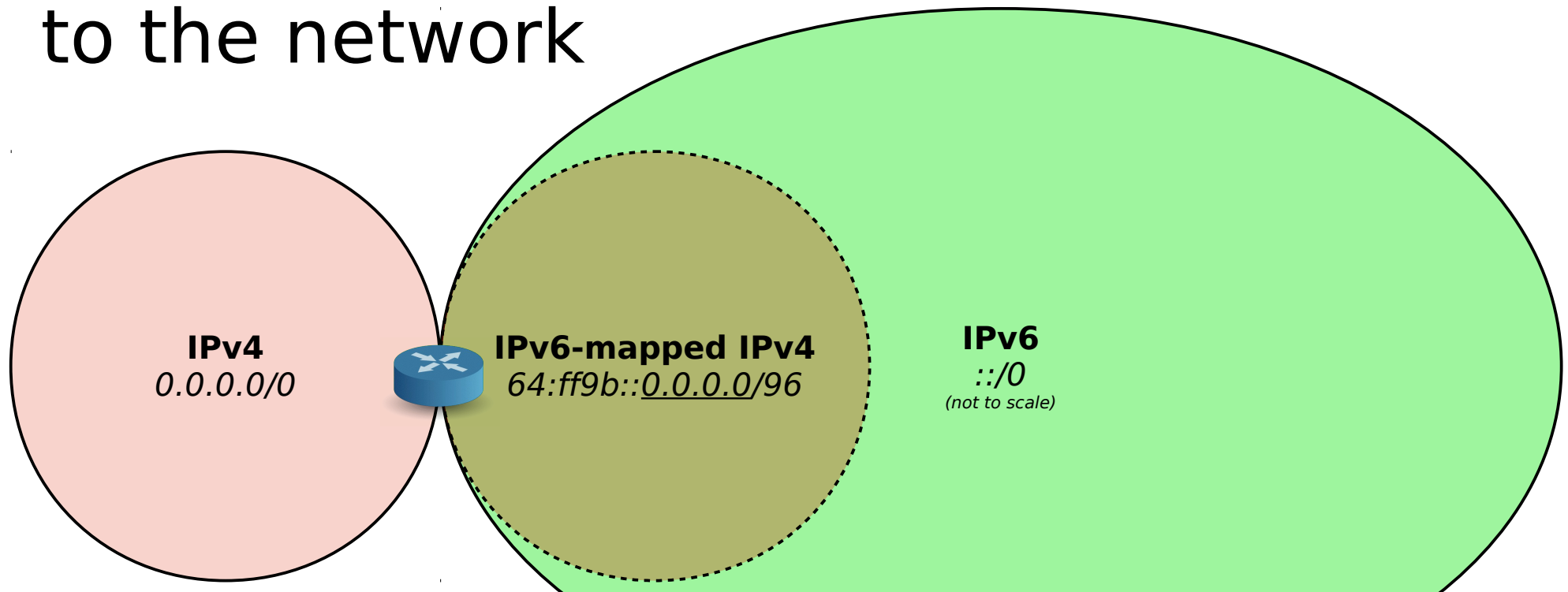
~~IPv6 only~~

Stateless IP/ICMP Translation (SIIT)

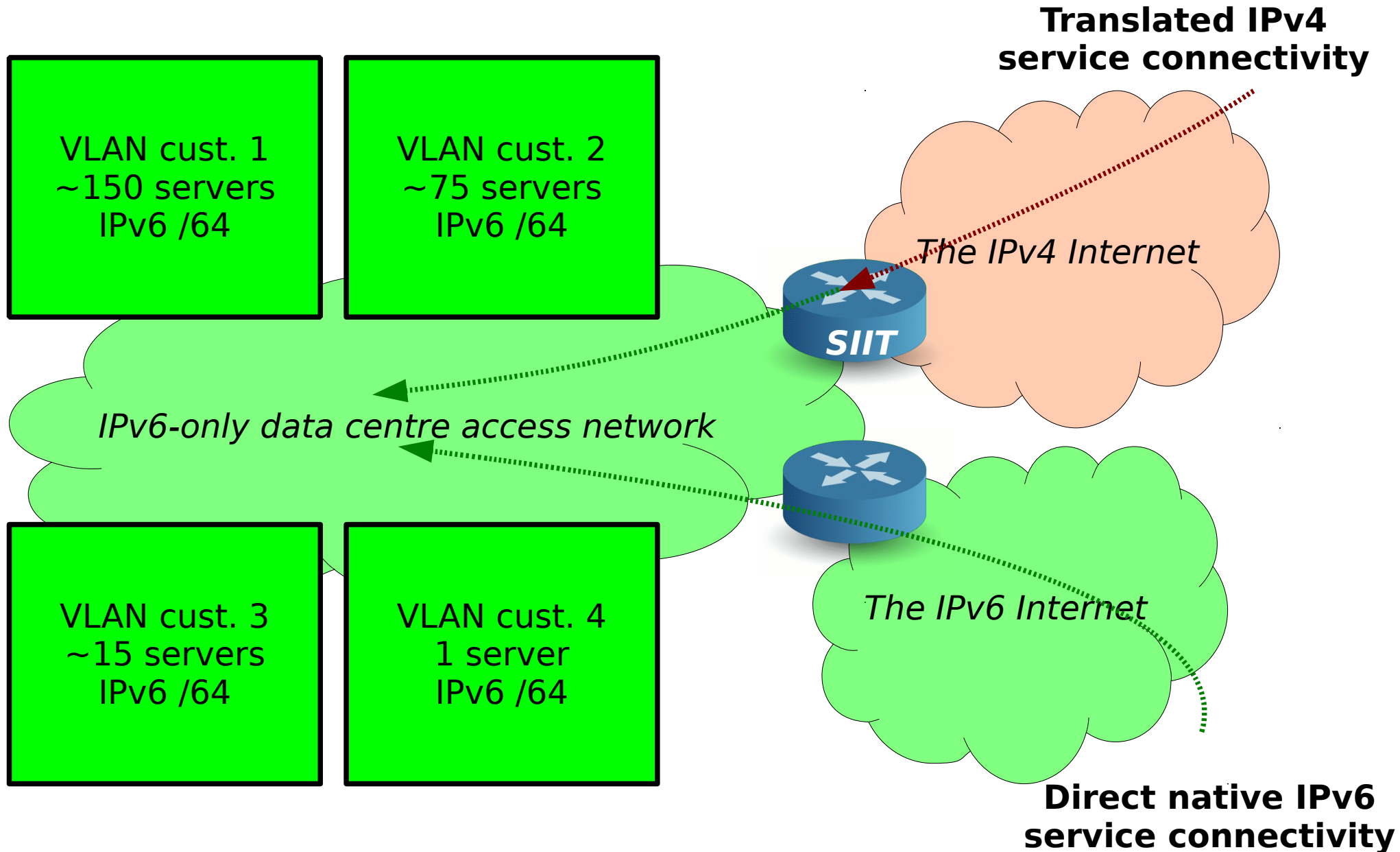
RFCs 6052, 6145,
draft-anderson-siit-dc-00

(Also known as Stateless
NAT64 andIVI)

- Maps the entire IPv4 internet into a /96 IPv6 prefix in a 1-to-1 stateless fashion
- Static stateless 1-to-1 IPv4-to-IPv6 mappings defined for each public service
- Server LANs and applications only use IPv6 – IPv4 connectivity is “outsourced” to the network



Native IPv6 + IPv4 via SIIT



Technical walkthrough

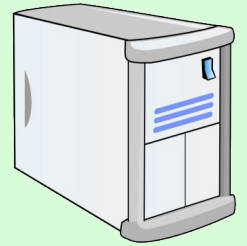
IPv4-only user



203.0.113.50



IPv6-only web server



2001:db8::1

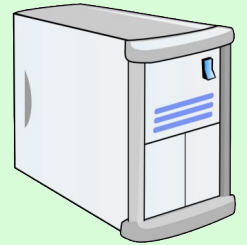
IPv4-only user



203.0.113.50



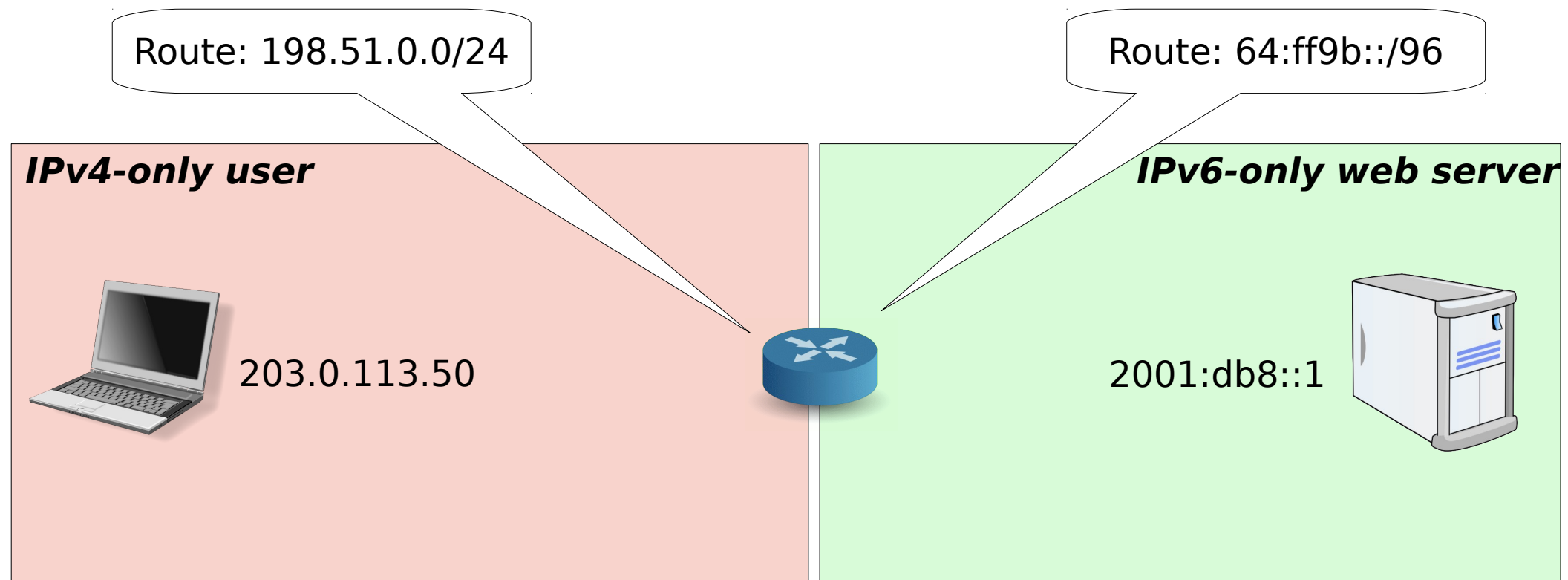
IPv6-only web server



2001:db8::1

Route: 64:ff9b::/96

- An IPv6 /96 prefix is assigned as the translation prefix representing the IPv4 internet and routed to the SIIT gateway



- An IPv6 /96 prefix is assigned as the translation prefix representing the IPv4 internet and routed to the SIIT gateway
- A pool of IPv4 service addresses are assigned and routed to the SIIT gateway

SIIT gateway configuration:
Map 198.51.0.10 - 2001:db8::1
Translation prefix: 64:ff9b::/96

Route: 198.51.0.0/24

Route: 64:ff9b::/96

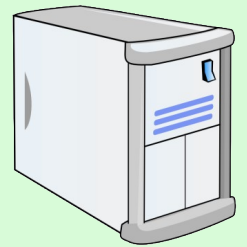
IPv4-only user



203.0.113.50



IPv6-only web server



2001:db8::1

- The SIIT gateway is configured with static IPv4 mappings for each IPv6 service
- The IPv6 /96 prefix is configured as a default rule (used if no static map match)
- IPv4 (IN A) records are added to DNS

SIIT gateway configuration:
Map 198.51.0.10 - 2001:db8::1
Translation prefix: 64:ff9b::/96

Route: 198.51.0.0/24

Route: 64:ff9b::/96

IPv4-only

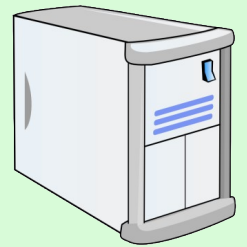
SRC: 203.0.113.50
DST: 198.51.0.10
HTTP GET /foo [...]

203.0.113.50



IPv6-only

2001:db8::1



- The client looks up the service's IPv4 address in DNS, and connects to it like it would with any other IPv4 address
- The IPv4 packet is routed to the SIIT gateway's IPv4 interface

SIIT gateway configuration:
Map 198.51.0.10 - 2001:db8::1
Translation prefix: 64:ff9b::/96

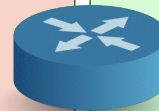
Route: 198.51.0.0/24

Route: 64:ff9b::/96

IPv4-only

SRC: 203.0.113.50
DST: 198.51.0.10
HTTP GET /foo [...]

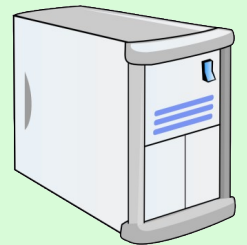
203.0.113.50



IPv6-only

SRC: 64:ff9b::203.0.113.50
DST: 2001:db8::1
HTTP GET /foo [...]

2001:db8::1



- The SIIT GW translates the packet to IPv6
 - DST address rewritten according to static map
 - SRC address gets the /96 prefix prepended (as it does not match any static maps)
- Layer 4 payload is copied verbatim

SIIT gateway configuration:
Map 198.51.0.10 - 2001:db8::1
Translation prefix: 64:ff9b::/96

Route: 198.51.0.0/24

Route: 64:ff9b::/96

IPv4-only

SRC: 203.0.113.50
DST: 198.51.0.10
HTTP GET /foo [...]

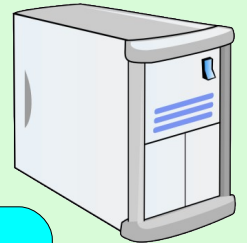
203.0.113.50



IPv6-only

SRC: 64:ff9b::203.0.113.50
DST: 2001:db8::1
HTTP GET /foo [...]

2001:db8::1



SRC: 2001:db8::1
DST: 64:ff9b::203.0.113.50
HTTP 200 OK [...]

- The server responds to the packet just as it would with any other IPv6 packet
 - Server needs no specific support for SIIT
- The original IPv4 source address isn't lost
- Response is routed back to the SIIT GW

SIIT gateway configuration:
Map 198.51.0.10 - 2001:db8::1
Translation prefix: 64:ff9b::/96

Route: 198.51.0.0/24

Route: 64:ff9b::/96

IPv4-only

SRC: 203.0.113.50
DST: 198.51.0.10
HTTP GET /foo [...]

203.0.113.50

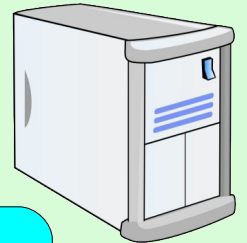
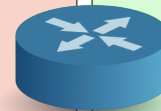
SRC: 198.51.0.10
DST: 203.0.113.50
HTTP 200 OK [...]

IPv6-only

SRC: 64:ff9b::203.0.113.50
DST: 2001:db8::1
HTTP GET /foo [...]

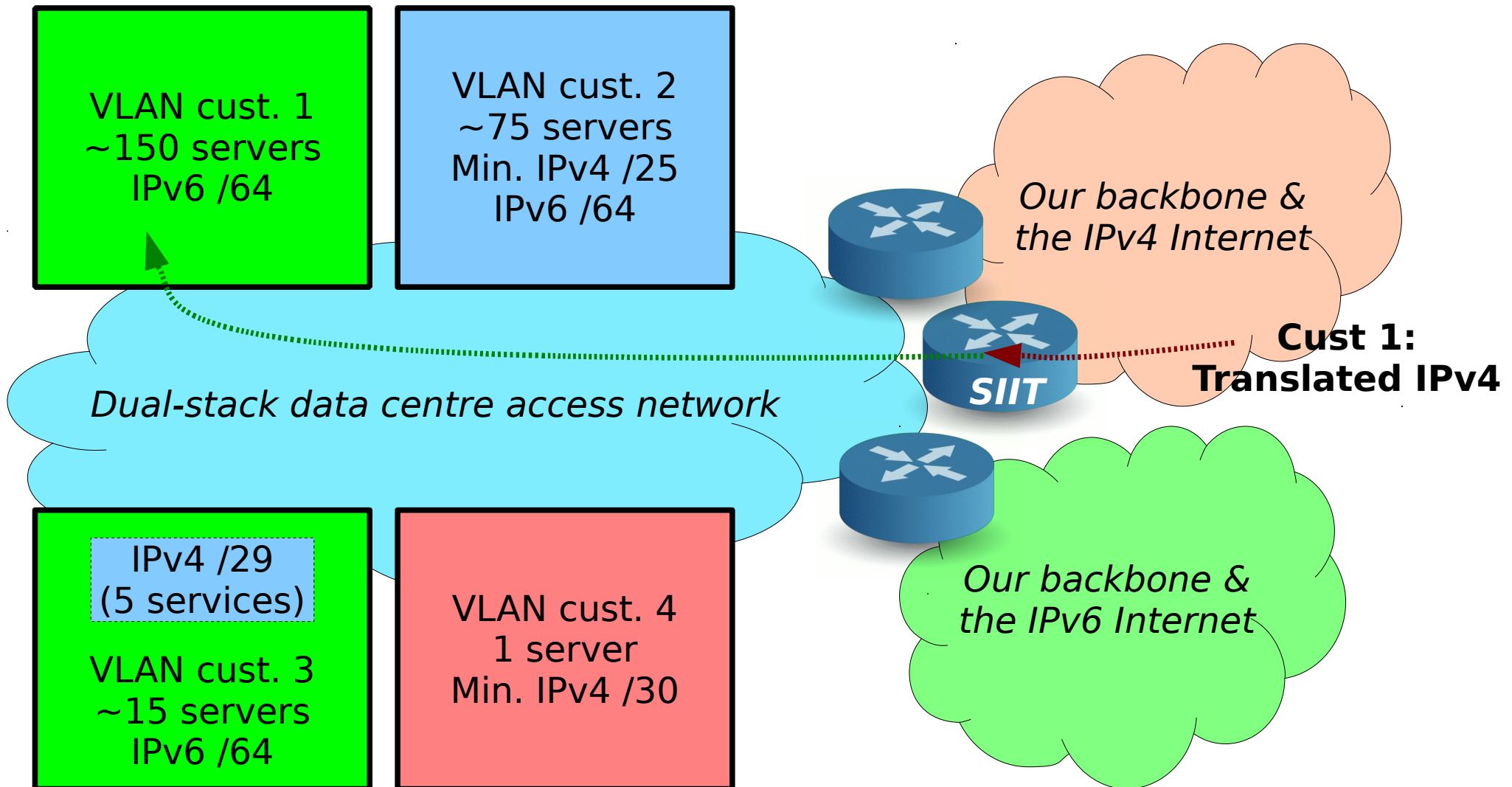
2001:db8::1

SRC: 2001:db8::1
DST: 64:ff9b::203.0.113.50
HTTP 200 OK [...]

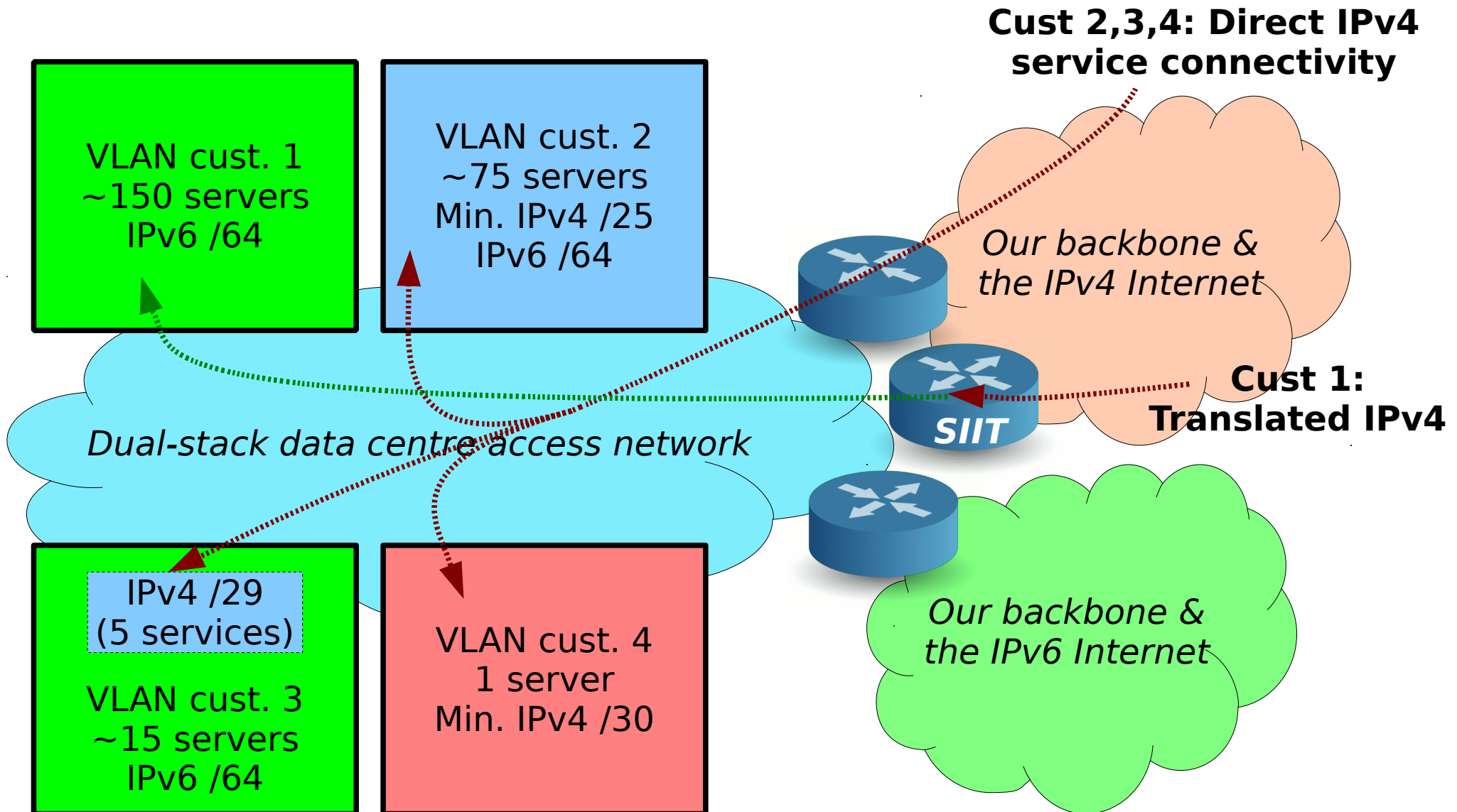


- The SIIT GW translates back to IPv4:
 - SRC address according to static mapping rule
 - DST address doesn't match any static map, so it only gets the /96 prefix stripped
- Response packet is routed back to client

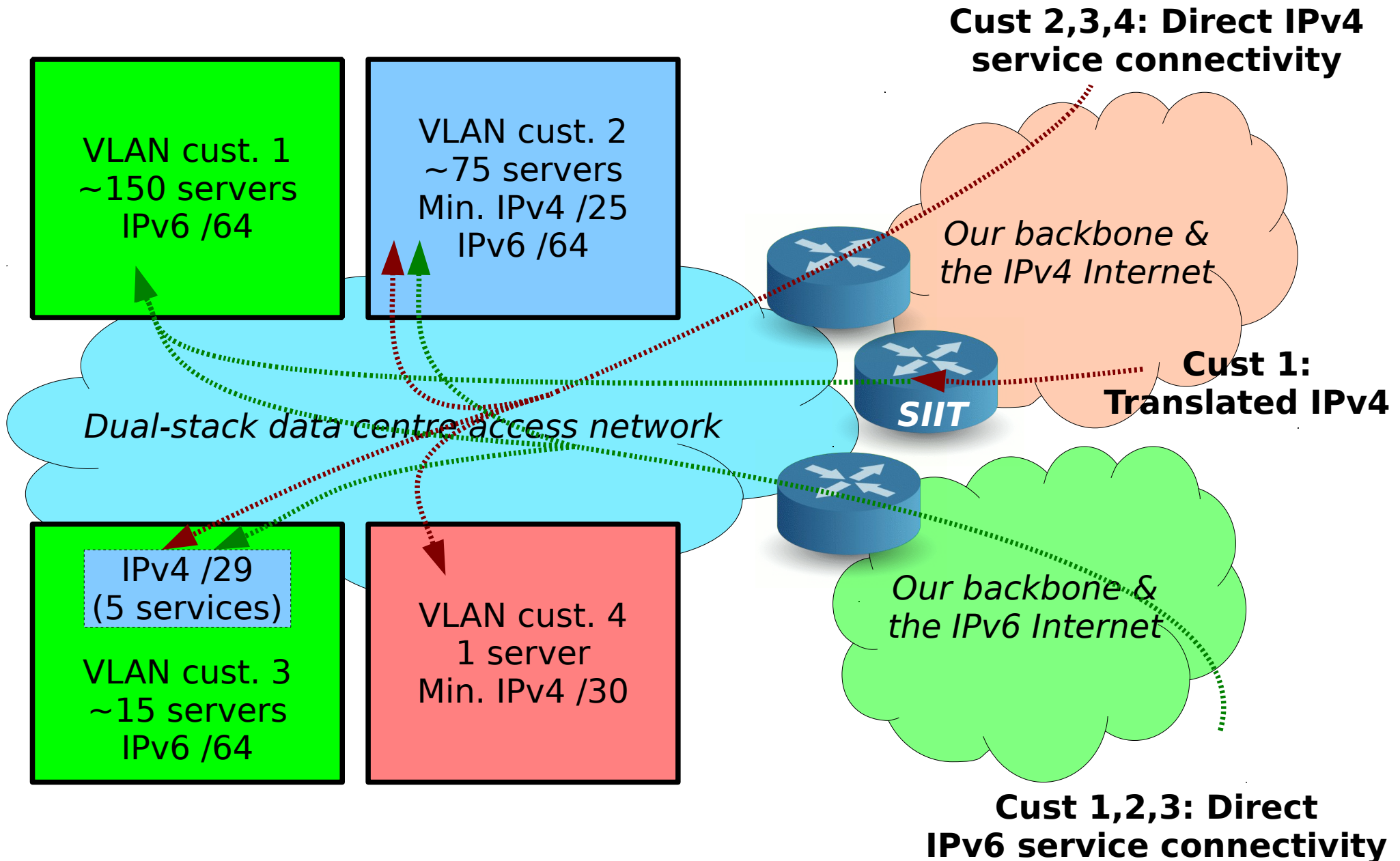
SIIT isn't all or nothing



SIIT isn't all or nothing

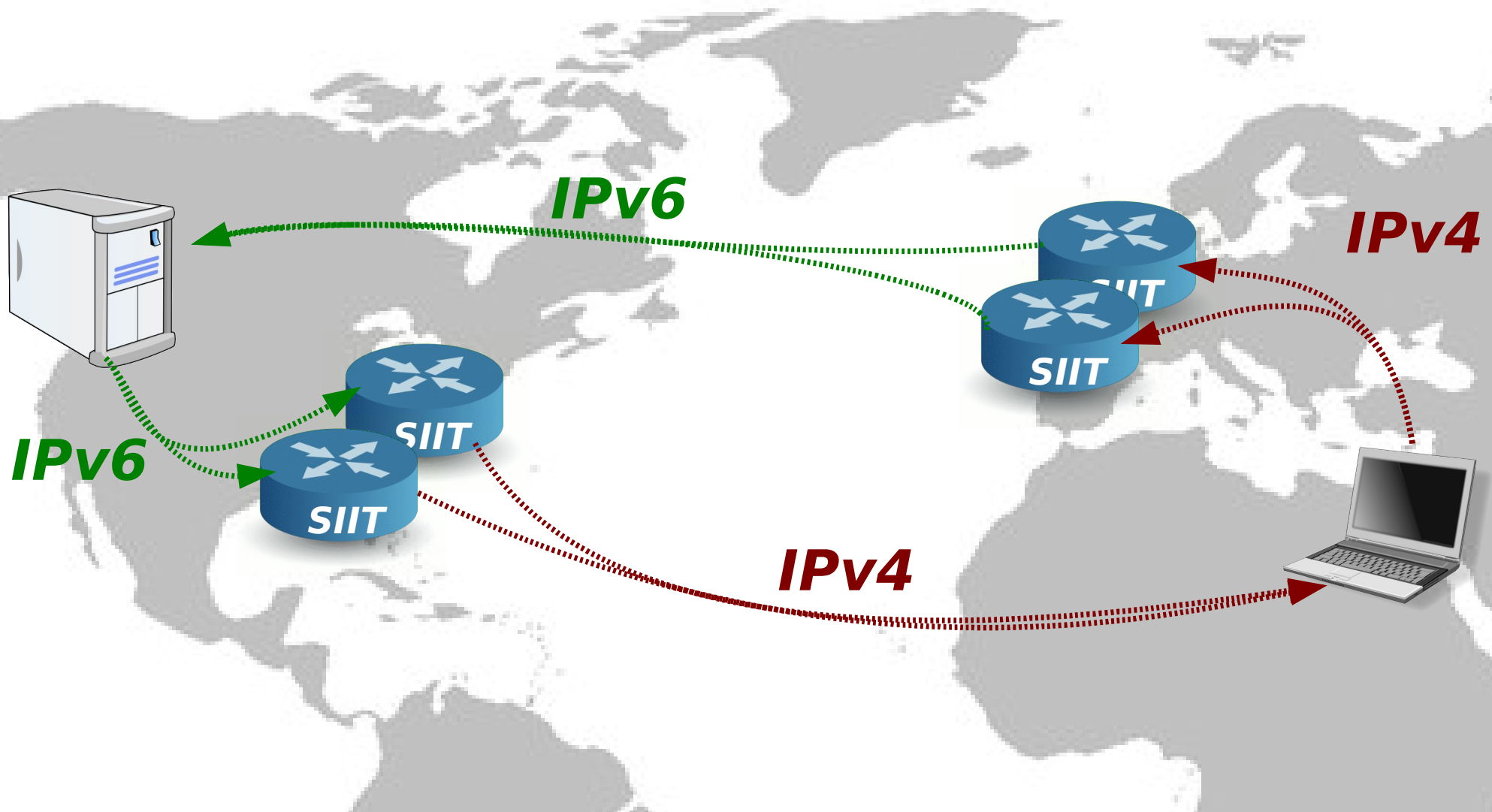


SIIT isn't all or nothing



Anycast, high availability, and ECMP (load balancing)

No problem thanks to the stateless nature of SIIT



SIIT highlights

- Stateless per-packet operation
 - You can use anycast, ECMP load balancing, ...
 - Does not require flows to bidirectionally traverse a single translator box
 - Doesn't need to be the servers' default route, may be placed anywhere in your network
 - Concurrent flow count and fps are irrelevant for performance (unlike NAT44 and proxies)
- The original IPv4 address remains known
 - Applications may geolocate IPv4 users

SIIT highlights, cont.

- Maximum conservation of IPv4 addresses
 - 1 address used per public service, none lost to infrastructure or subnet 2 overhead, etc.
- Single-stack applications and server LANs
 - No additional complexity, unlike dual stack
- Application stack is independent of IPv4
 - Forget about further IPv6 migration projects
 - When IPv4 has become irrelevant, remove IN A records and shut down SIIT gateways - done

Application requirements

- If the application doesn't work through NAT44, it will likely not work with SIIT
 - e.g., FTP (uses IP literals in Layer 7 payload)
- If the application does work with NAT44, it will likely work with SIIT as well
 - e.g., HTTP and HTTPS
- The servers' OS and application stacks must fully support IPv6

Existing implementations

- TAYGA for Linux (open source)
 - <http://www.litech.org/tayga/>
- Cisco ASR1K
 - Static mapping feature implemented in IOS XE version 3.10 (due to be released in July)
- Others? Let me know!

It works in practice too :-)

Browser used IPv4 to connect

This is Tore Anderson's personal home page (or rather a sorry excuse for one).

I've worked quite a bit with IPv6 in recent years. You can find an archive of all talks/presentations I've done on the subject [here](#), and I am also generating a [daily report page](#) detailing IPv6 end-user capability in Norway (based on traffic statistics from [VG](#) and [A-pressen Digitale Medier](#)).

My [Curriculum Vitae / Resume](#) (Norwegian).

- [LinkedIn](#)
- [Twitter](#)

You can get in touch with me by sending an e-mail to tore@fud.no or by calling/texting my mobile at +47 95931212

Some rather technical info about your HTTP request follows:

```
Remote addr: 2a02:c0::46:0:54d1:c206 (translated from IPv4 address 84.209.194.6)
Remote host: 2a02:c0::46:0:54d1:c206
Local addr: 2a02:c0:1001:100::145
User-Agent: Mozilla/5.0 (X11; Linux i686 (x86_64)) AppleWebKit/537.22 (KHTML, like G
Random ID: 901037772
```

Web server saw it as an IPv6 connection

**0x54 . 0xD1 . 0xC2 . 0x06
=
84.209.194.6**

Questions?

Thank you for listening!

Further reading:

RFC 6052 - IPv6 Addressing of IPv4/IPv6 Translators

RFC 6145 - IP/ICMP Translation Algorithm

RFC 6219 - The China Education and Research Network (CERNET) IVI Translation Design

draft-anderson-siit-dc-00 - Stateless IP/ICMP Translation in IPv6 Data Centre Environments

<http://toreanderson.no> - My personal home page (contact info, social media links, slides from this and earlier talks)

<http://redpill-linpro.com> - My employer and sponsor of this project

Note: IPv4 traffic to both of the above URLs is routed through a SIIT gateway (eating my own dog food)



**IPv6 Business
Conference**

Organized by
SWISS
IPv6
COUNCIL

2013
June, 11