Building IPv6-Only Data Centers

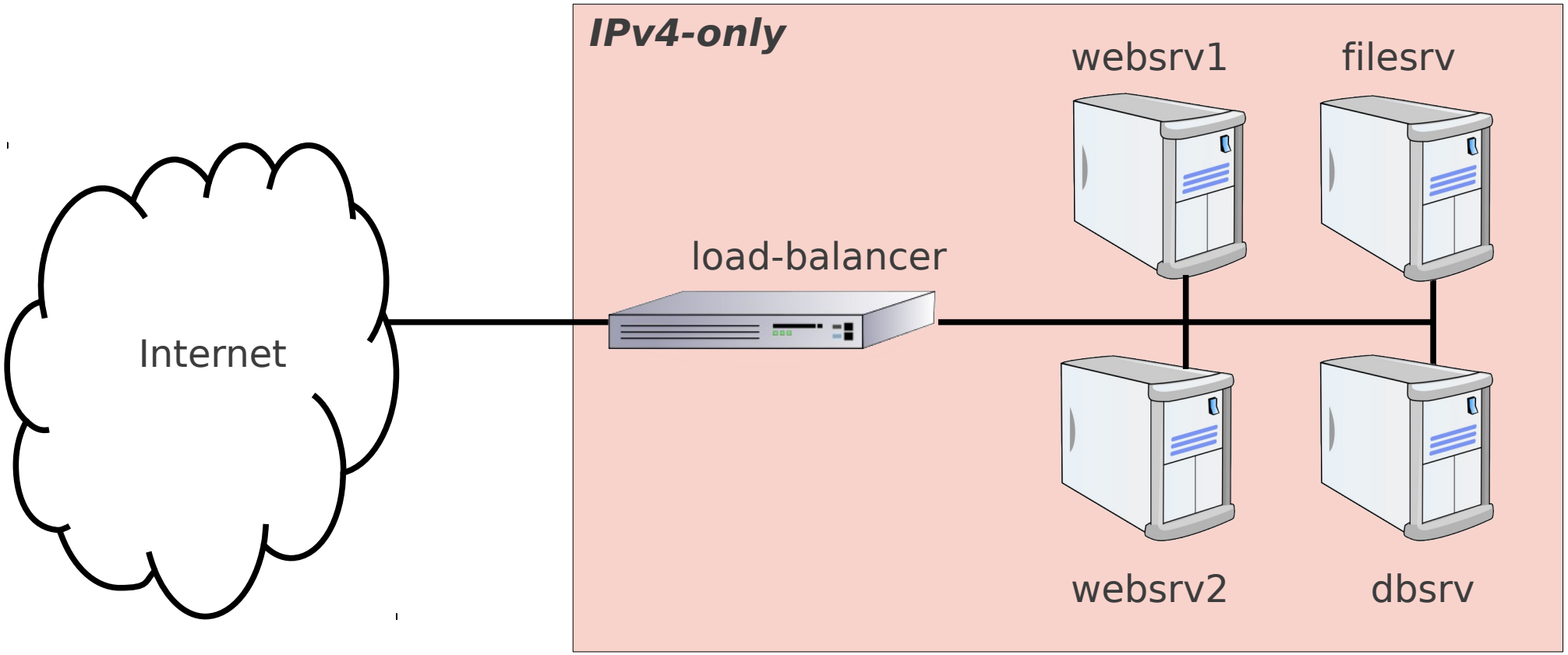
Osama I. Al-Dosary dosary@solyton.com

13th IPv6 TaskForce Meeting, May 2013

menog.org

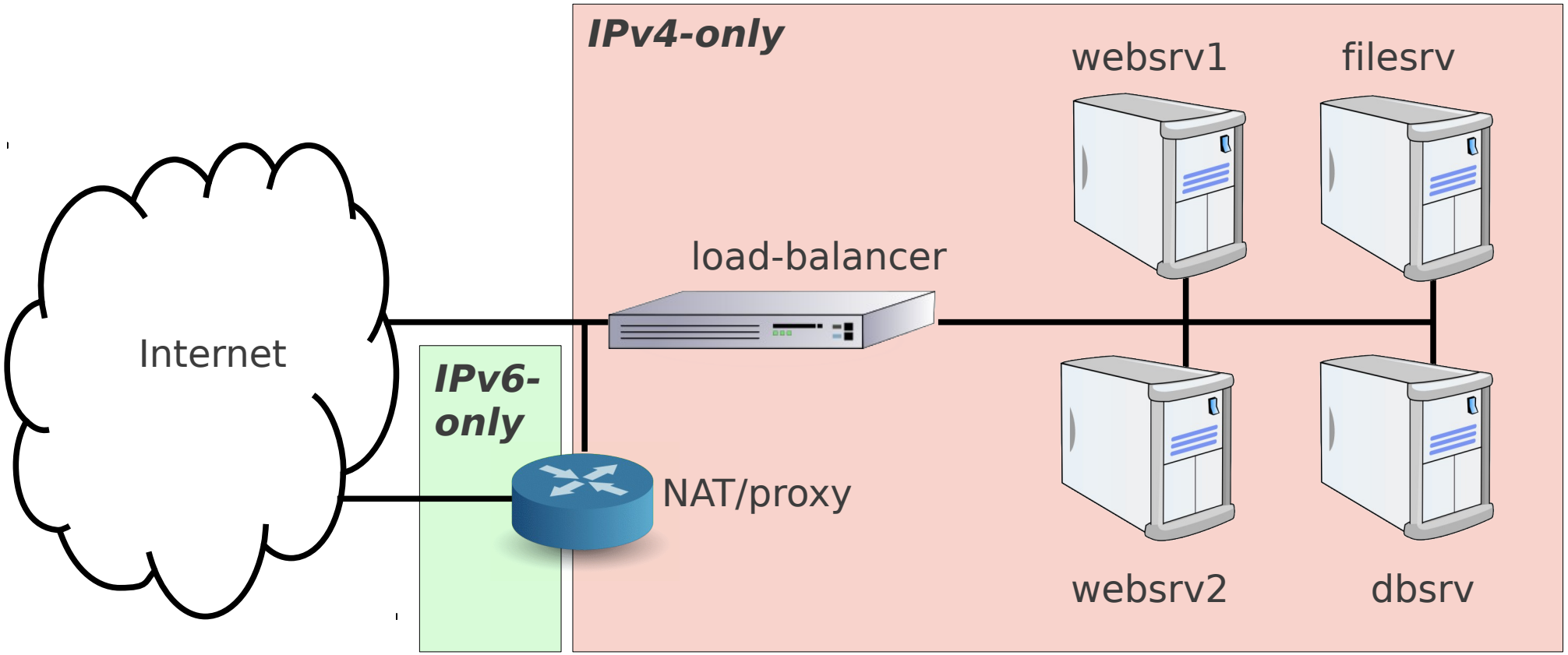
IPv6 deployment approaches

0) Traditional IPv4-only DC environment



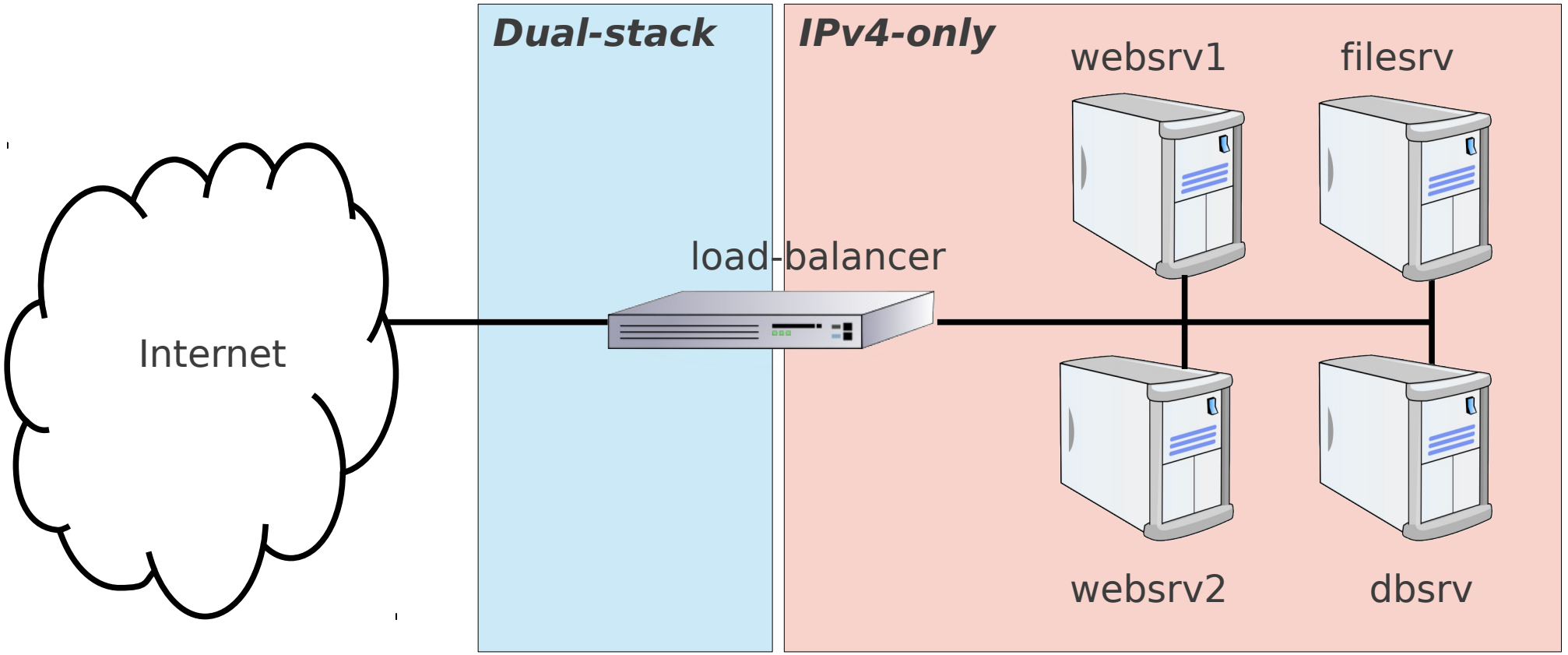
IPv6 deployment approaches

1) IPv4-only + IPv6 via NAT or proxy



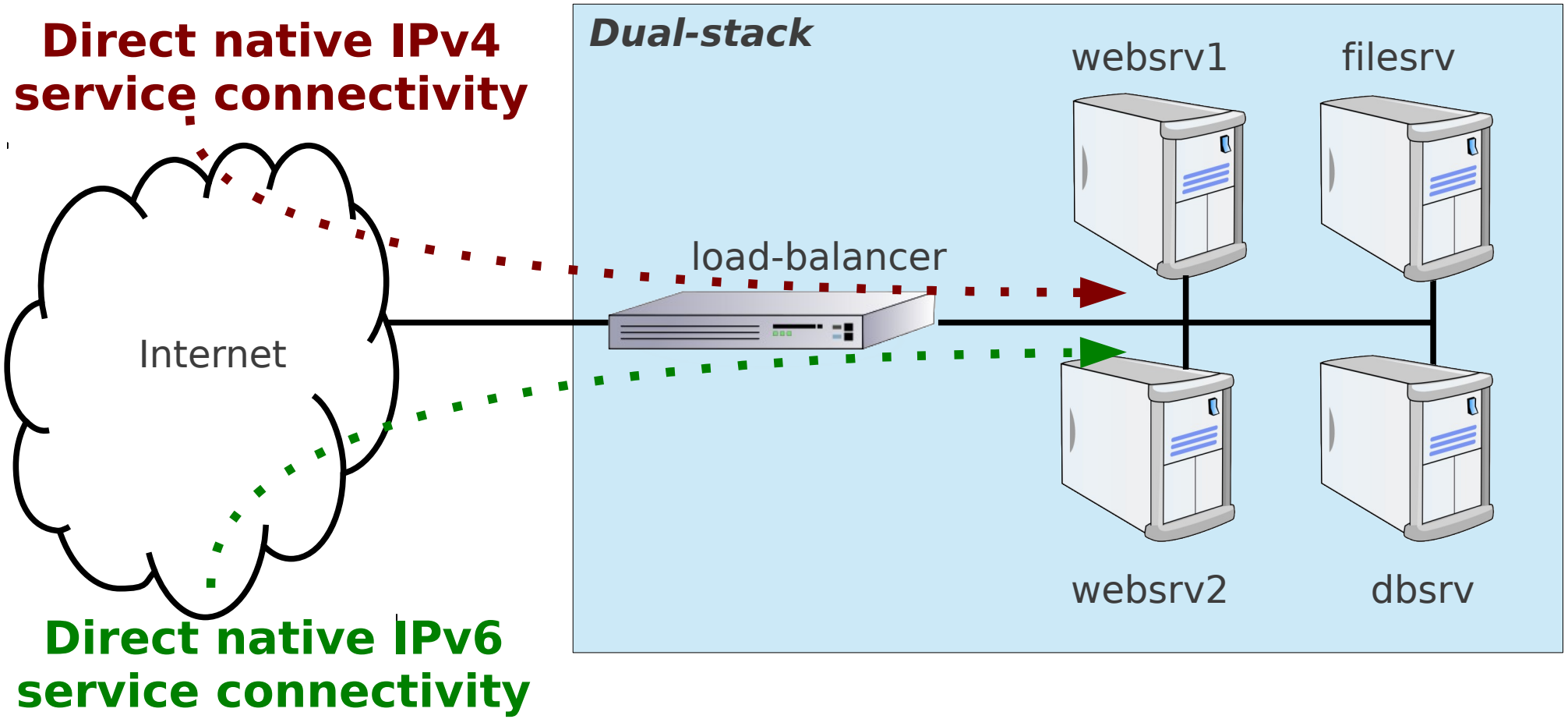
IPv6 deployment approaches

2) Dual-stacked public front-end, IPv4 BE



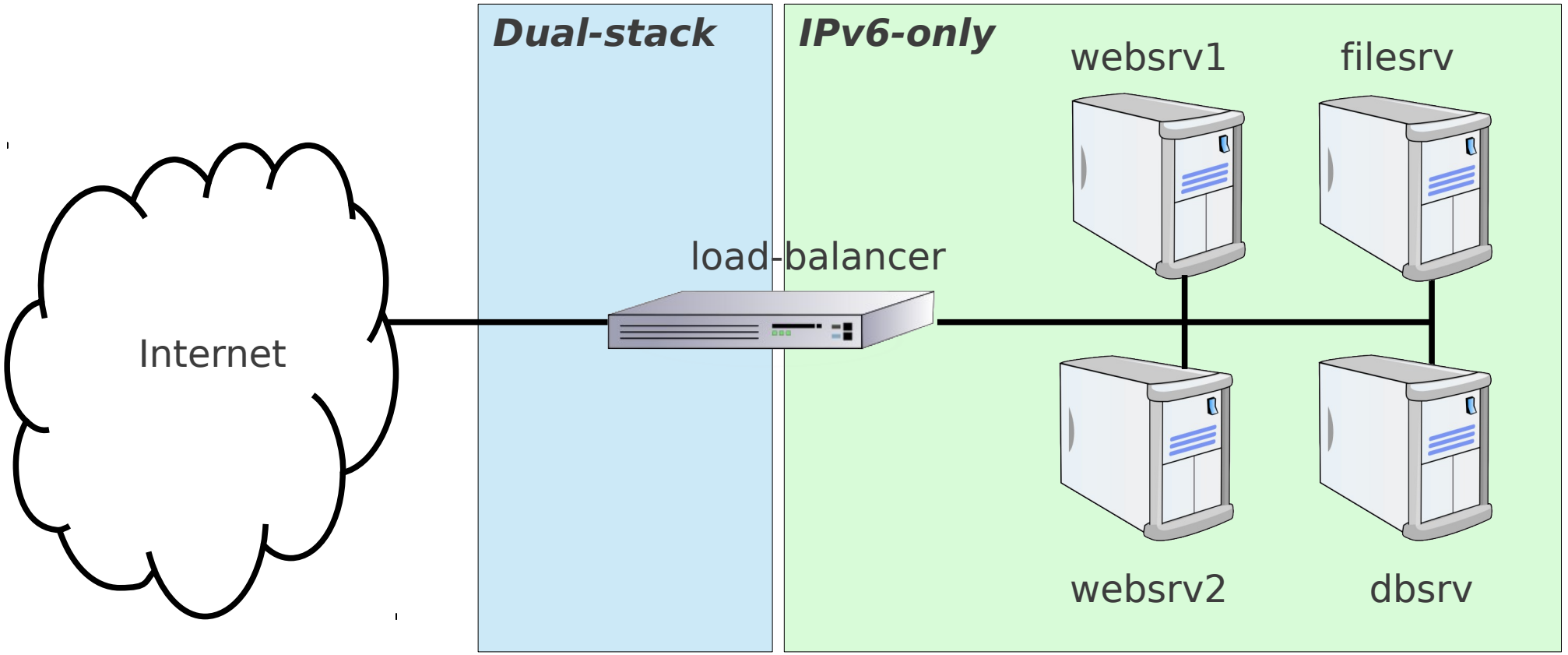
IPv6 deployment approaches

3) Full dual-stack



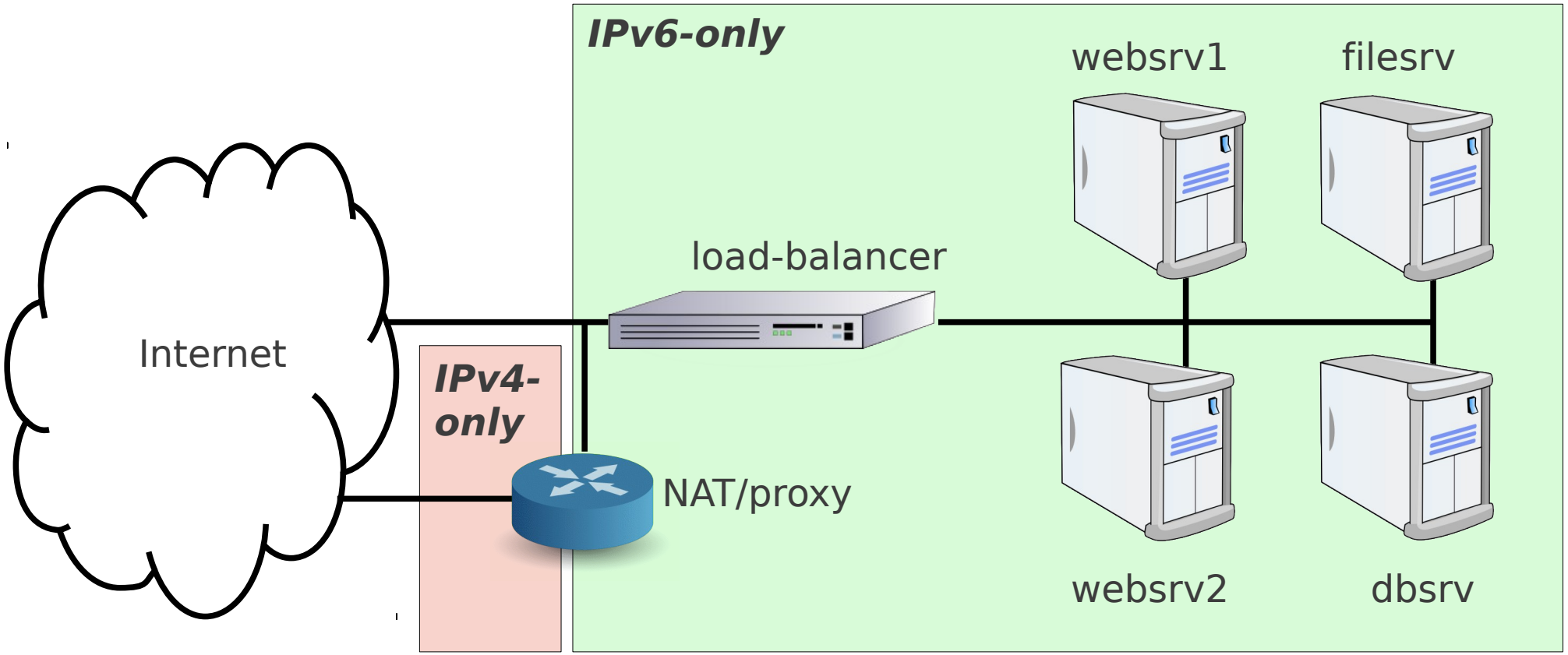
IPv6 deployment approaches

4) Dual-stacked public front-end, IPv6 BE



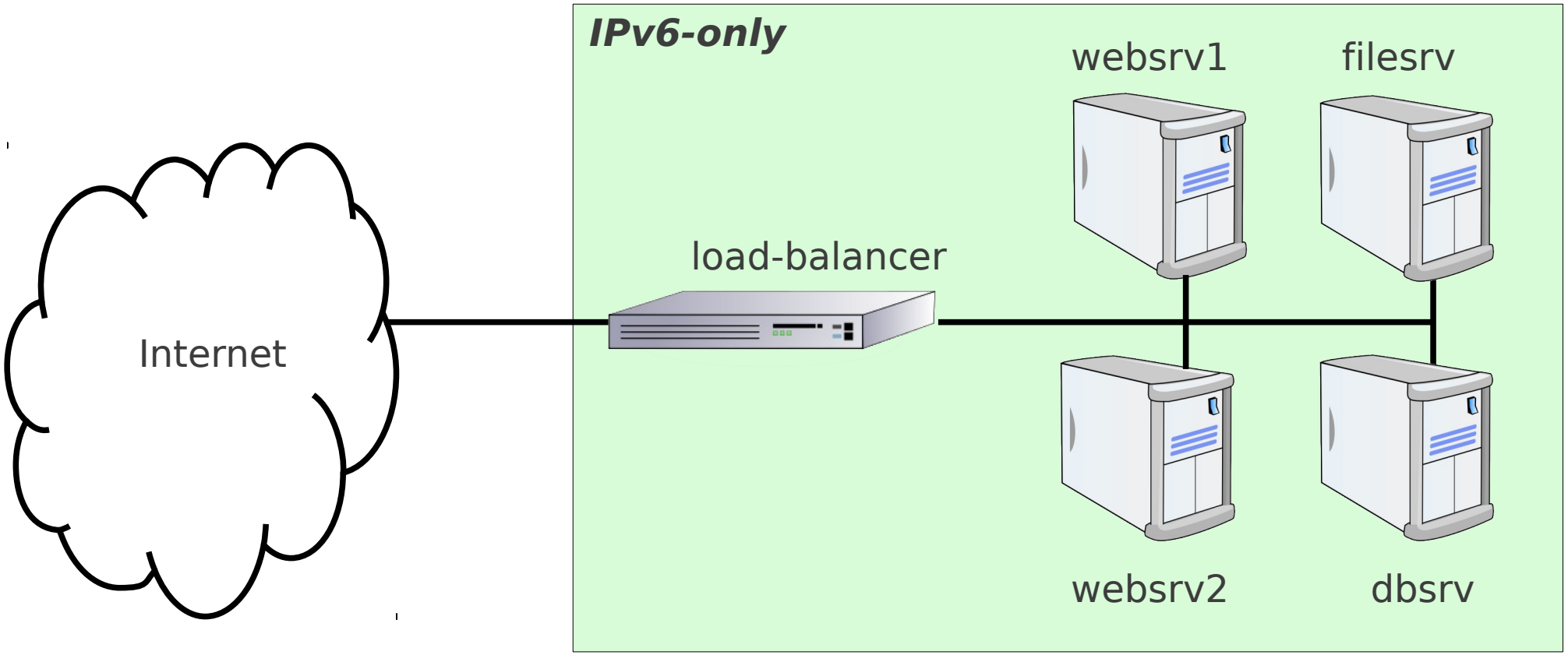
IPv6 deployment approaches

5) IPv6-only + IPv4 via NAT or proxy



IPv6 deployment approaches

6) IPv6-only, no IPv4 connectivity at all



Incremental IPv6 deployment

- 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?

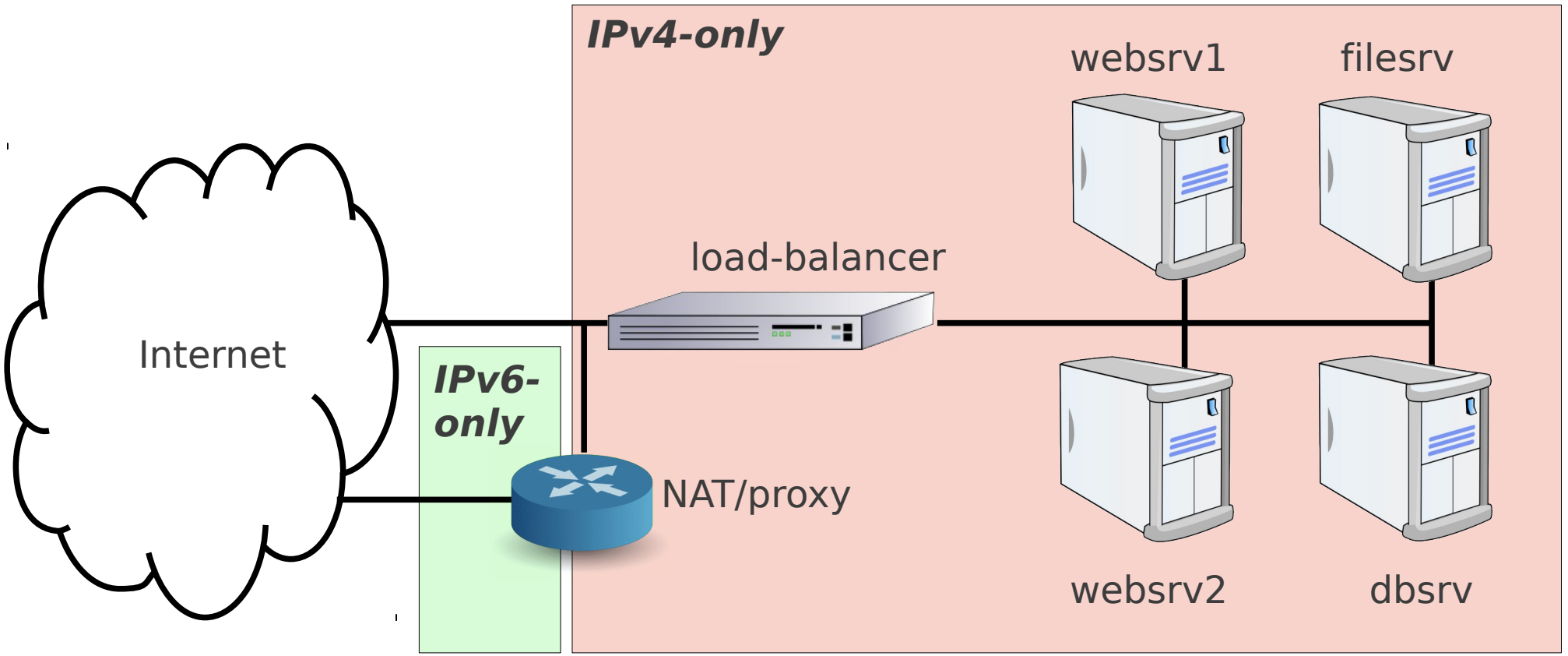
- IPv4-only
- IPv4-only + IPv6 via NAT/proxy
- Dual-stacked public frontend, IPv4 BE
- Full dual-stack
- Dual-stacked public frontend, IPv6 BE
- IPv6-only + IPv4 via NAT/proxy
- ~~IPv6-only~~

< 1% of end-users world-wide have IPv6!

Incremental IPv6 deployment

- 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

IPv6 via NAT64 or proxy



NAT64 or Proxies Disadvantages

- Requires Stateful devices:
  - expensive,
  - hard to scale,
  - limits routing flexibility, and
  - vulnerable to DoS attacks
- Failures/fail-over breaks all sessions
- Does not help with IPv4 depletion

NAT64 or Proxies Disadvantages (Cont'd)

- Obscures source IPv6 address of user, which limits Content Provider features
  - Geo Location
  - Analysing visitor behaviour
  - Auditing
  - Security / Access Control
  - Rate Limiting

Incremental IPv6 deployment

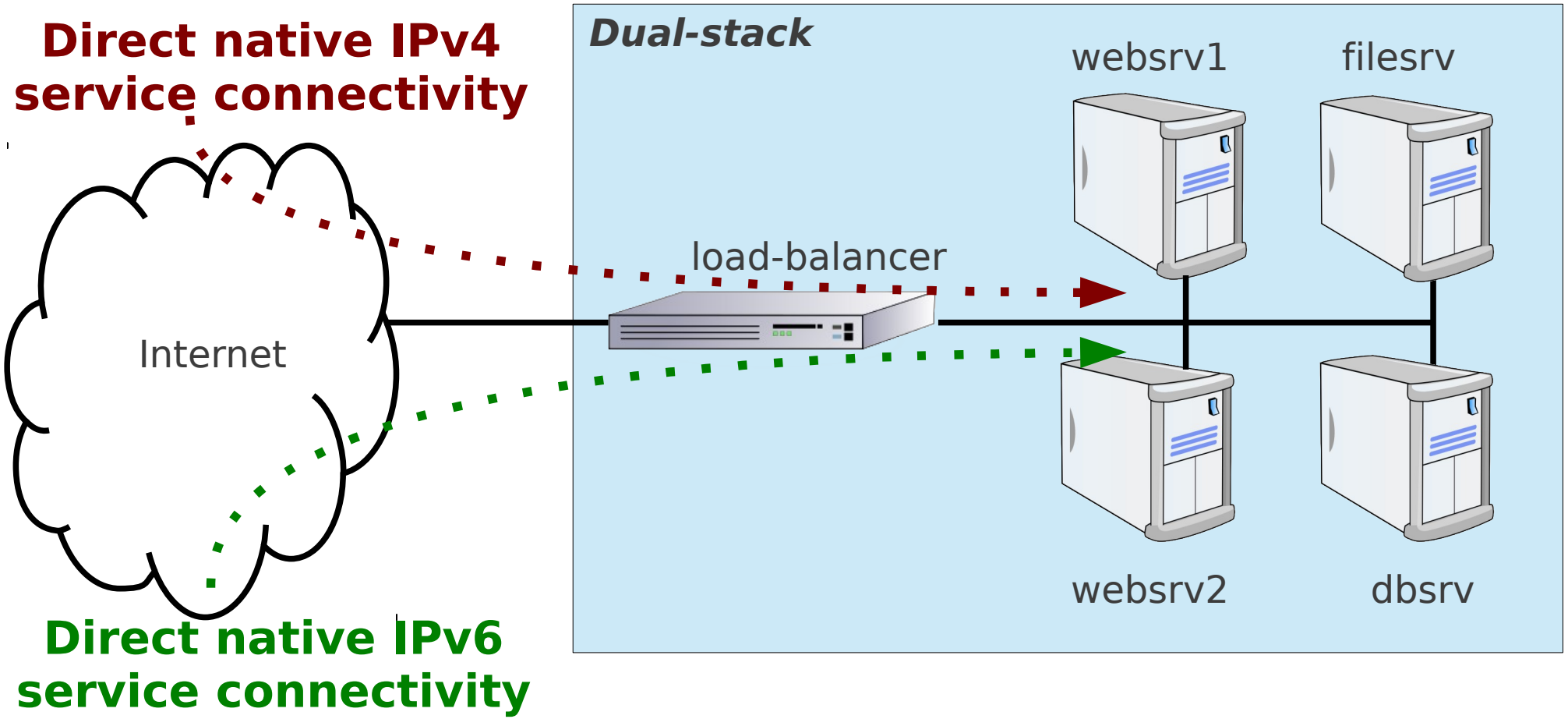
- IPv4-only
- IPv4-only + IPv6 via NAT/proxy
- Dual-stacked public frontend, IPv4 BE
- Full dual-stack
- Dual-stacked public frontend, IPv6 BE
- IPv6-only + IPv4 via NAT/proxy
- IPv6-only

Incremental IPv6 deployment

- 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~~

IPv6 deployment approaches

3) Full dual-stack



Incremental IPv6 deployment

- 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

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

Why skip dual-stack?

- DS implies lots of unwanted complexity
  - More ACLs, monitoring, troubleshooting, possible failures, training, documentation, ...
- Saves lots of precious IPv4 addresses
  - 1 IPv4 per service, rather than 1+ per server
- Forces sysadmins to learn and use IPv6
  - They resist change and new technologies
  - Provision dual-stack and they'll use IPv4 only
- Perform a single IPv6 migration project

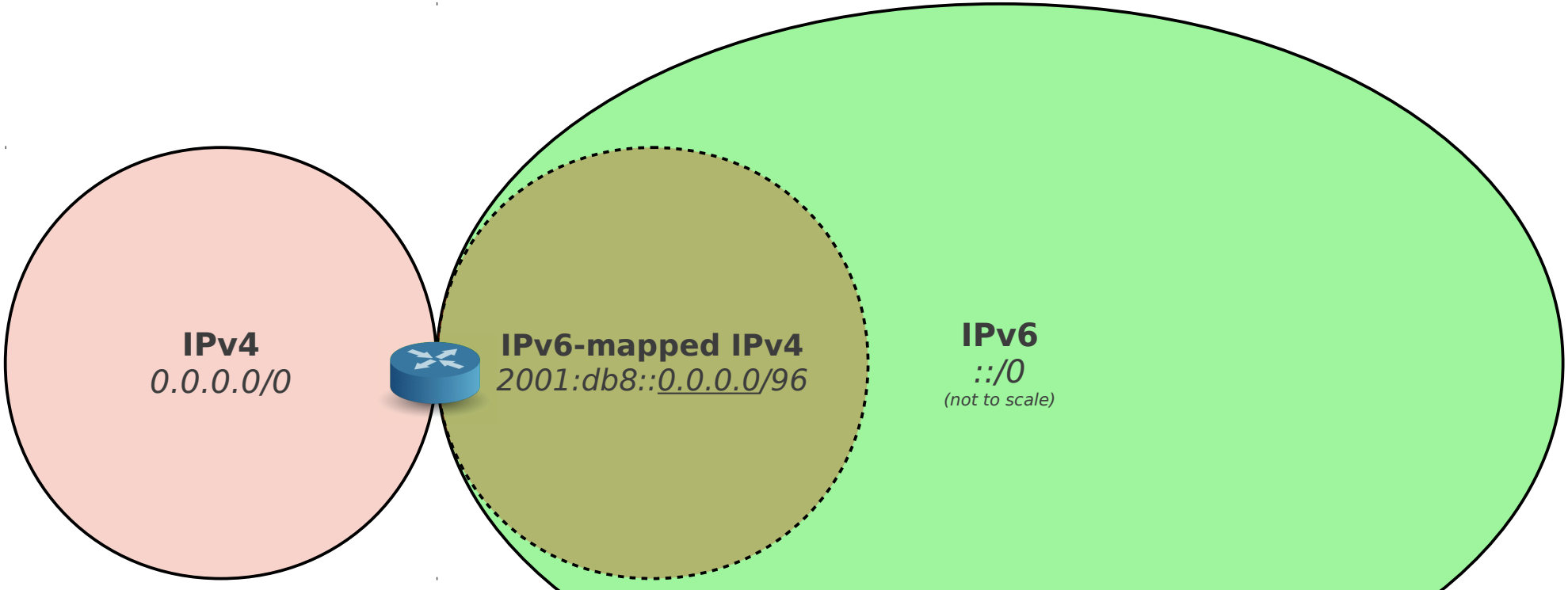
Stateless IP/ICMP Translation (SIIT)

RFCs 6052, 6145

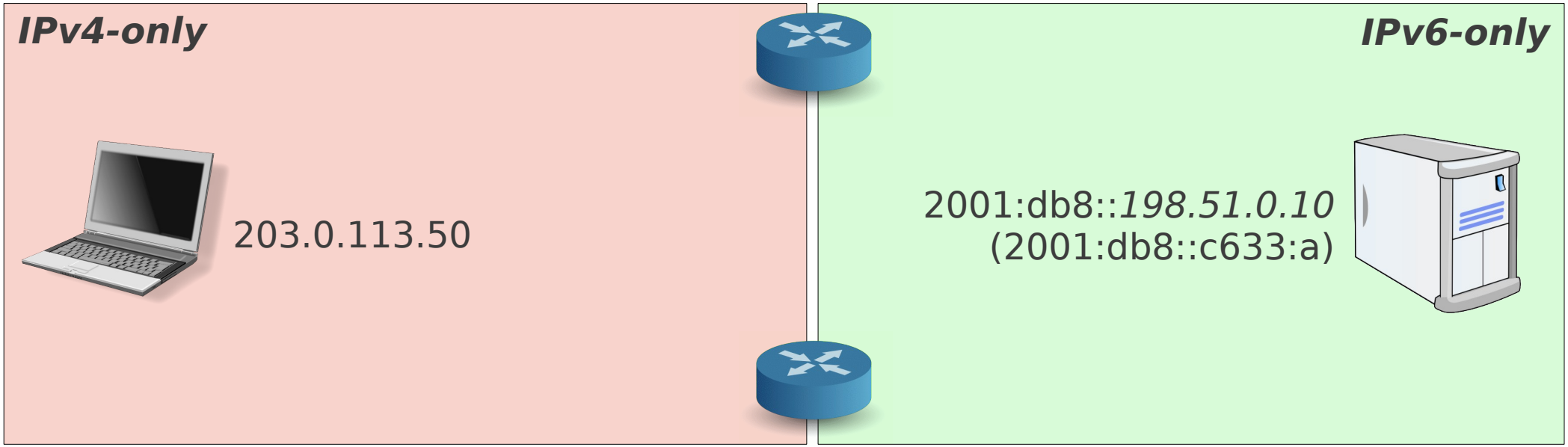
(Also known as Stateless NAT64 and IVI)

SIIT in a nutshell

- Maps the entire IPv4 address space into an IPv6 prefix from the SP's address space
- Translates IPv4/ICMPv4 headers into IPv6/ICMPv6 headers, and vice versa

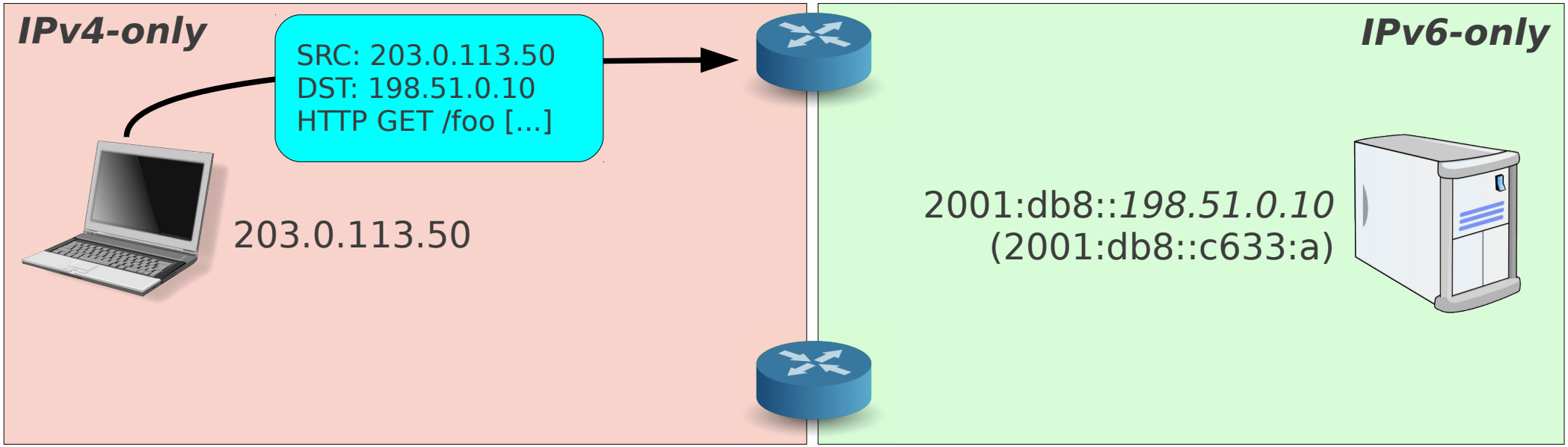


SIIT theory



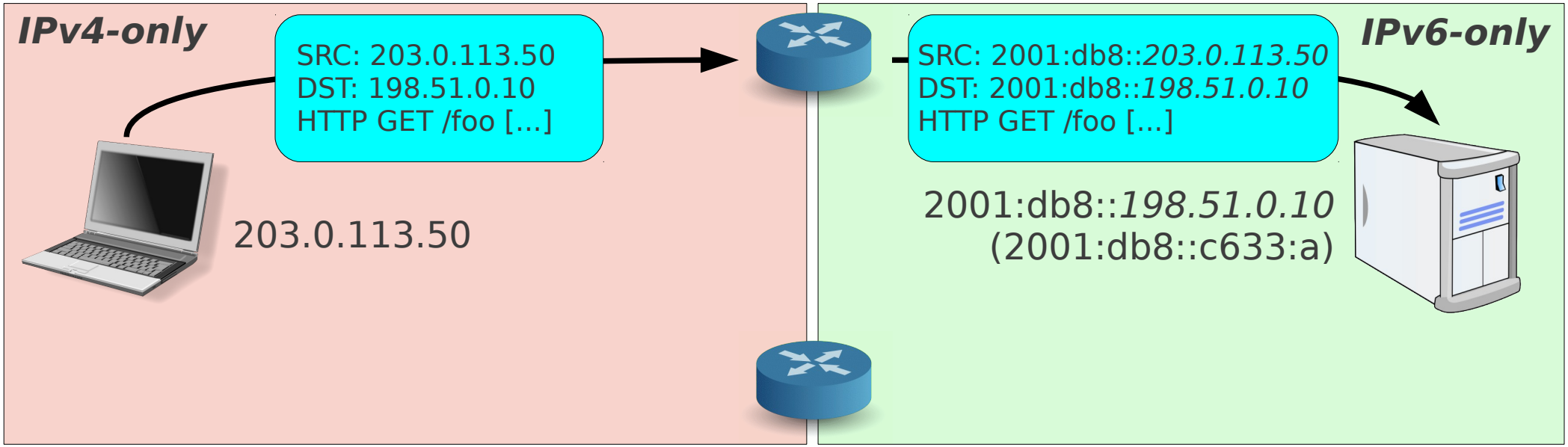
- An IPv4-translatable IPv6 address from a pre-defined /96 prefix (that represents the IPv4 internet) is configured on the server
- This address is routed to the server using regular IPv6 routing techniques

An IPv4 client connecting



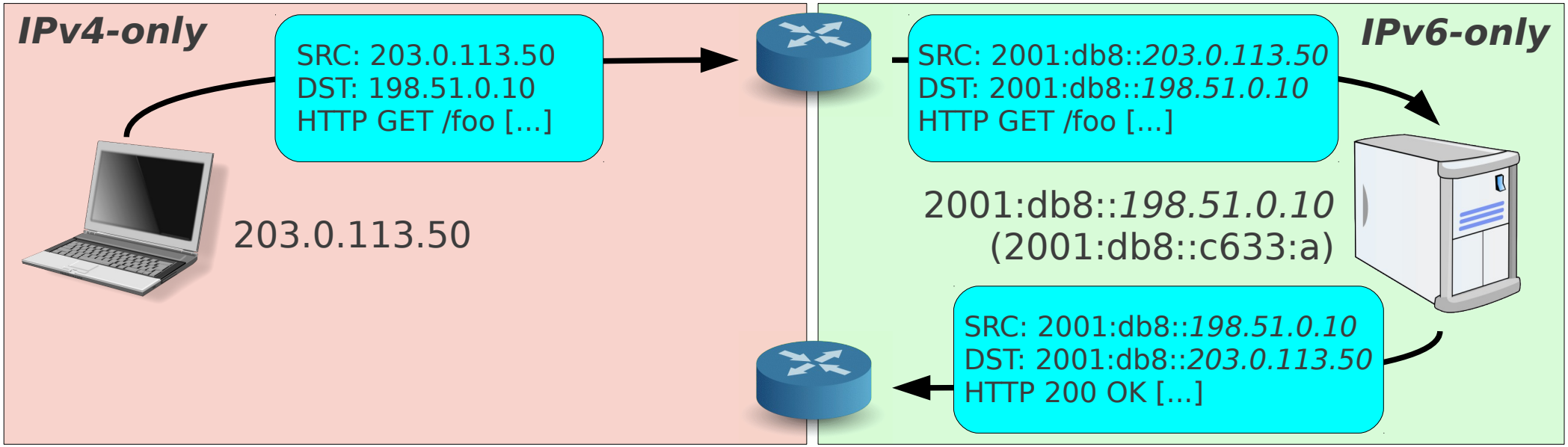
- The IPv4 service address is published as a regular A record for the service in DNS
- It's routed to the provider's SIIT gateways using standard IPv4 routing techniques
- IPv4 clients connect to it in a normal way

IPv4->IPv6 translation



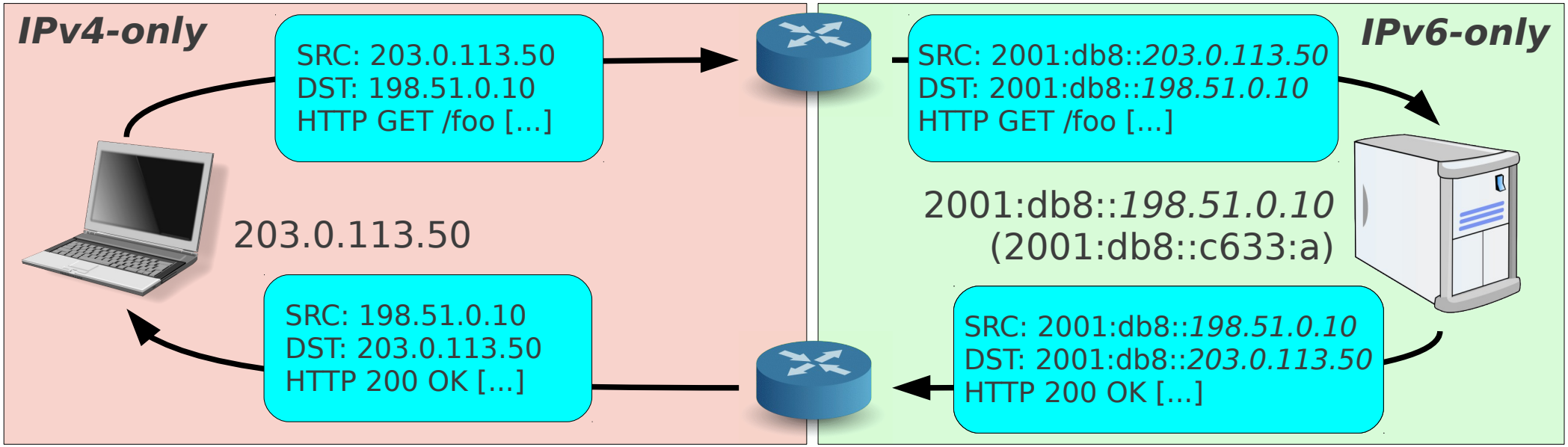
- The pre-defined /96 prefix is prepended to the IPv4 packet's SRC and DST fields
- Layer 4 payload is copied verbatim
- The packet is then routed to the server as a completely ordinary IPv6 packet

IPv6 server processing



- The server responds to the packet just as it would with any other IPv6 packet
- The original IPv4 source address isn't lost
- The /96 prefix (equivalent to the IPv4 default route) is routed to a SIIT gateway

IPv6->IPv4 translation



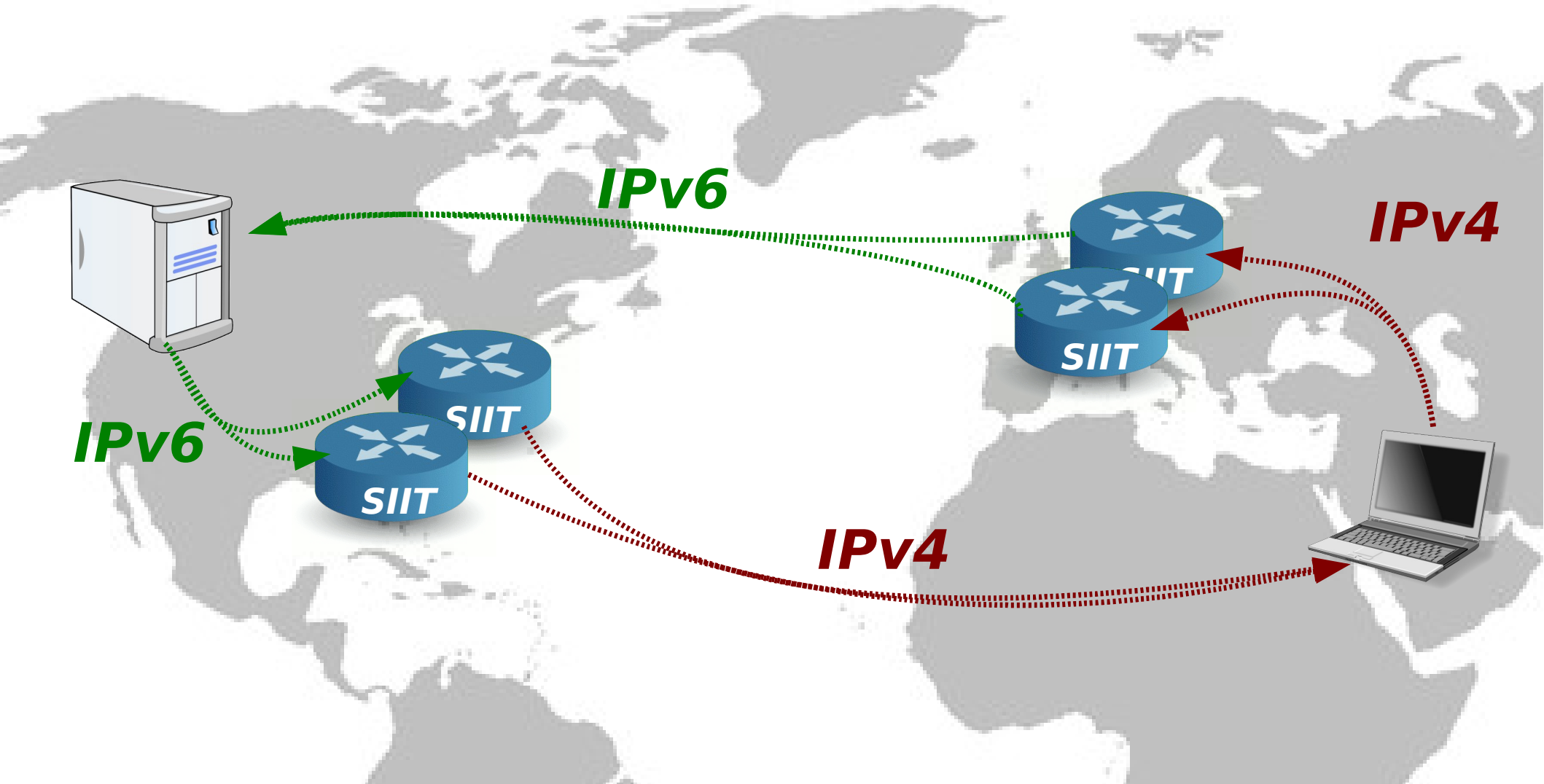
- The /96 prefix is stripped from the IPv6 packet's SRC and DST fields
- Layer 4 payload is untouched
- The resulting IPv4 packet is returned to the client which processes it normally

SIIT highlights

- Maximum conservation of IPv4 addresses
  - 1 address used per public service, none lost to infrastructure or subnet ^2 overhead, etc.
- Stateless per-packet operation
  - You can use anycast, ECMP load balancing, ...
  - Does not require flows to flow bidirectionally across a single translator
  - Concurrent flow count and fps are irrelevant for performance (unlike NAT44 and proxies)

Anycast, high availability, and ECMP (load balancing)

No problem thanks to the stateless nature of SIIT



SIIT highlights, cont.

- The original IPv4 address remains known
  - Applications may apply on IPv4 users: geo-location; auditing; access control; visitor tracking, etc.
- Single-stack applications and server LANs
  - No additional complexity, unlike dual stack
- 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 on roadmap only

Questions?
dosary@solyton.com

Presentation by Tore Anderson, delivered at RIPE 64