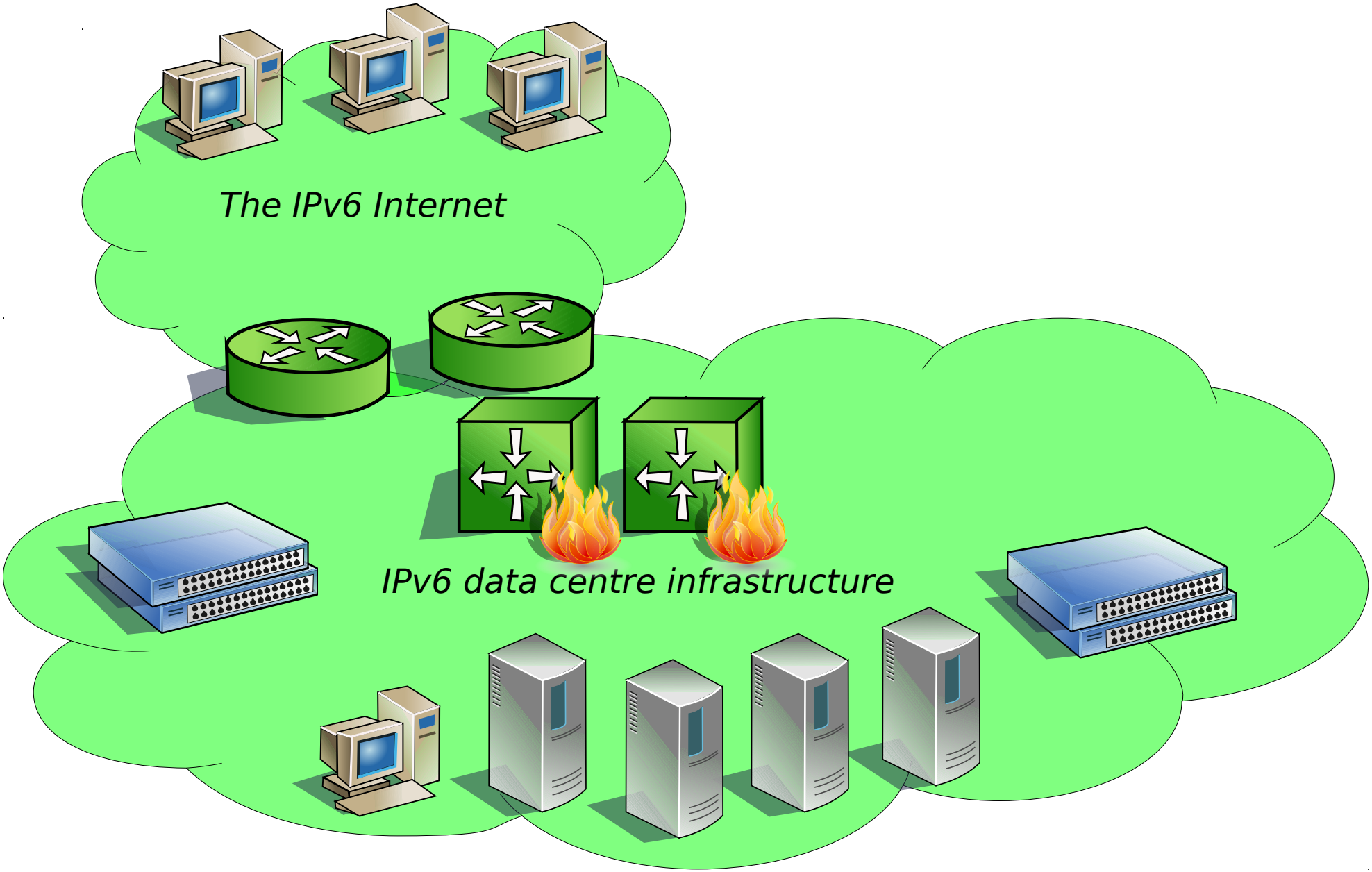
SIIT-DC: Stateless IP/ICMP Translation for IPv6 Data Centre Environments & SIIT-DC: Dual Translation Mode

Tore Anderson

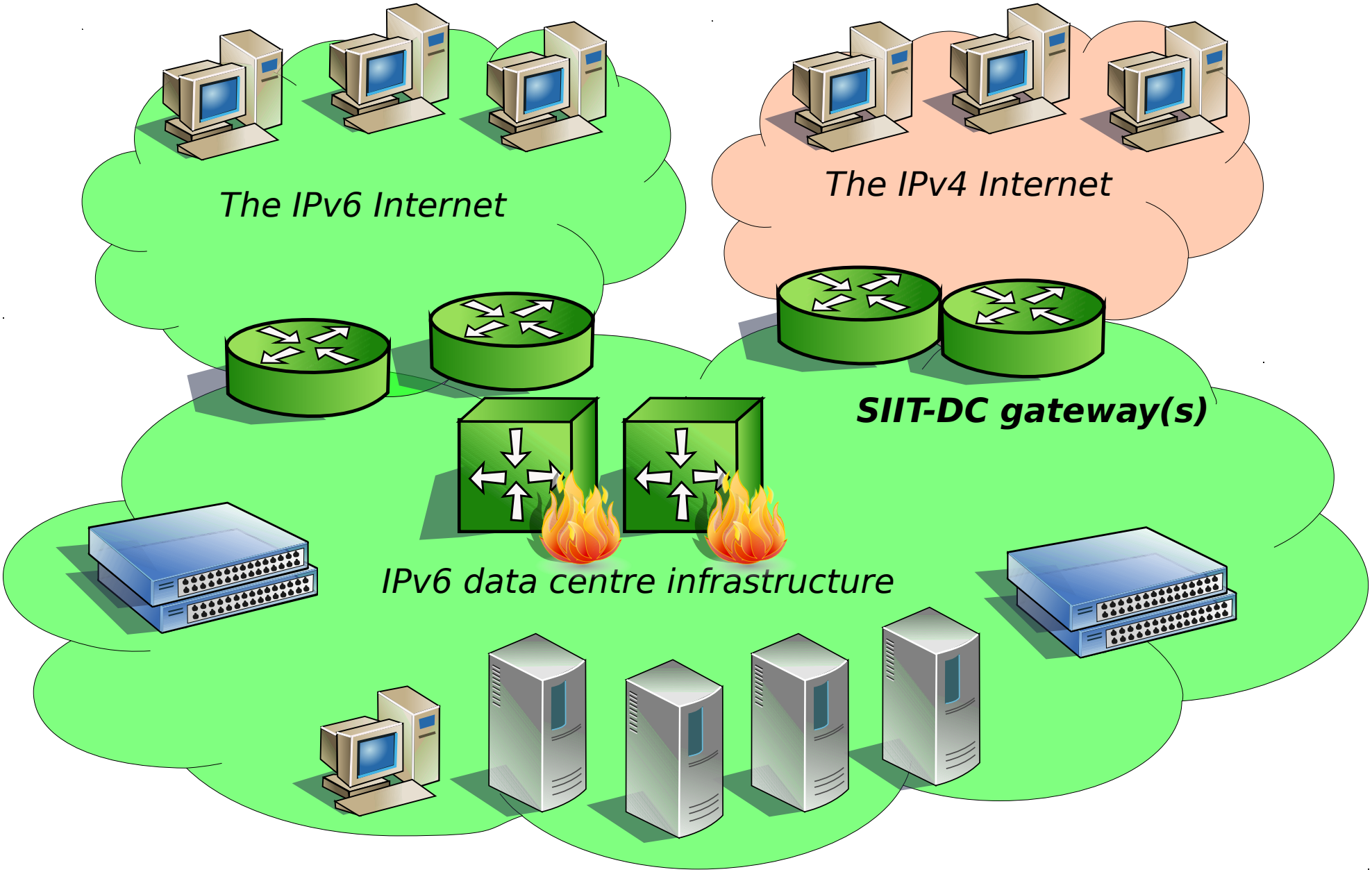
Redpill Linpro AS

RIPE 91, Honolulu, November 2014

An IPv6 data centre

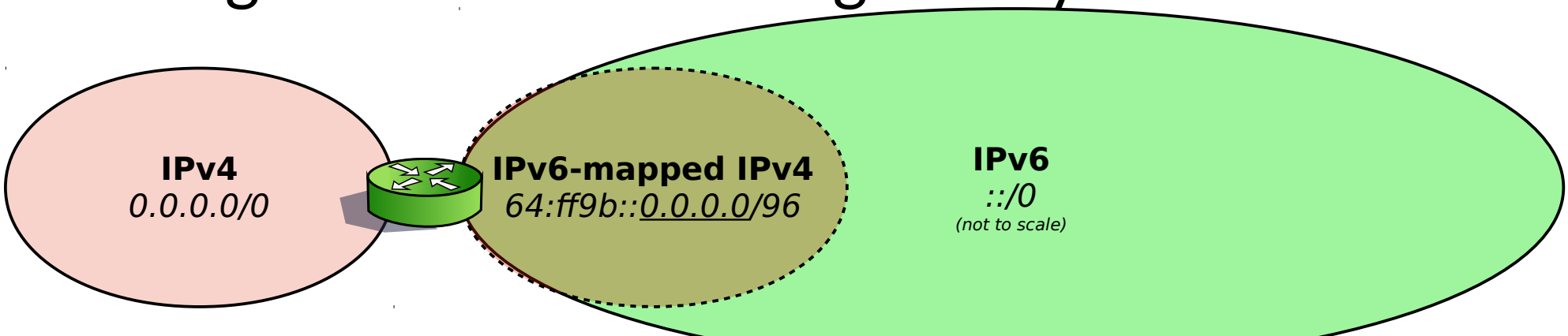


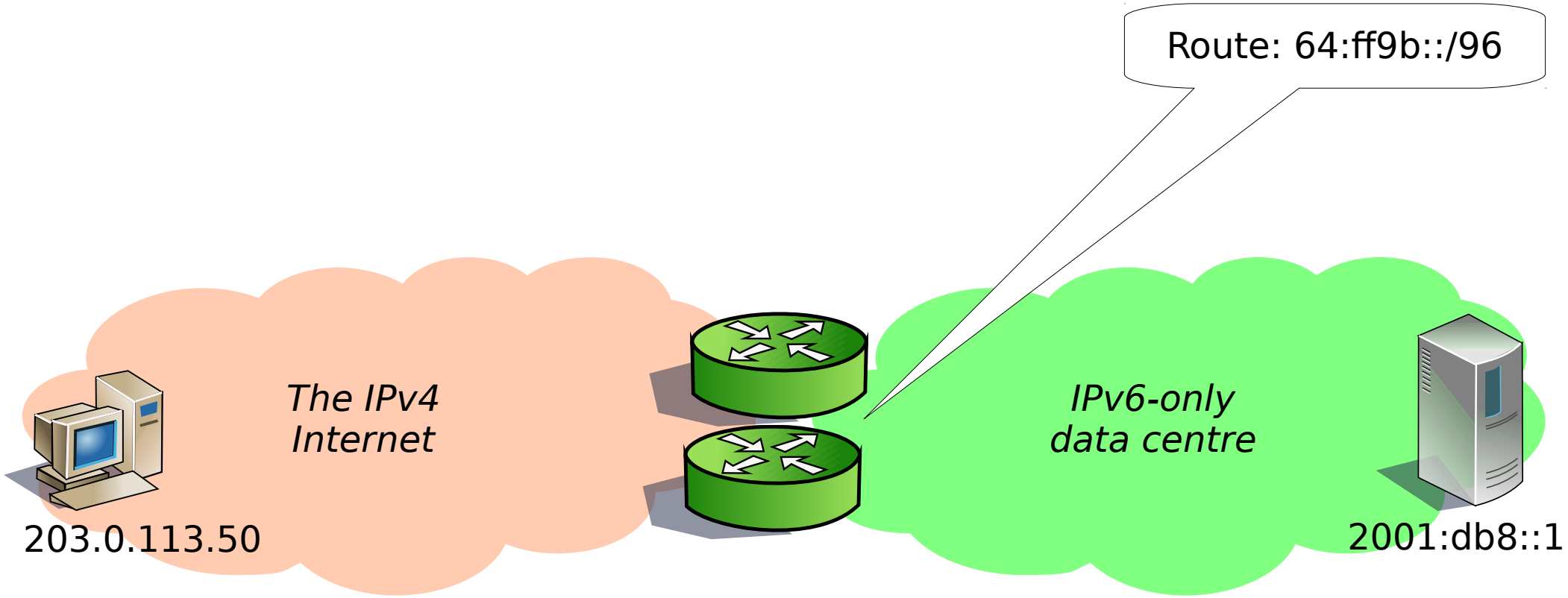
An IPv6 data centre with SIIT-DC



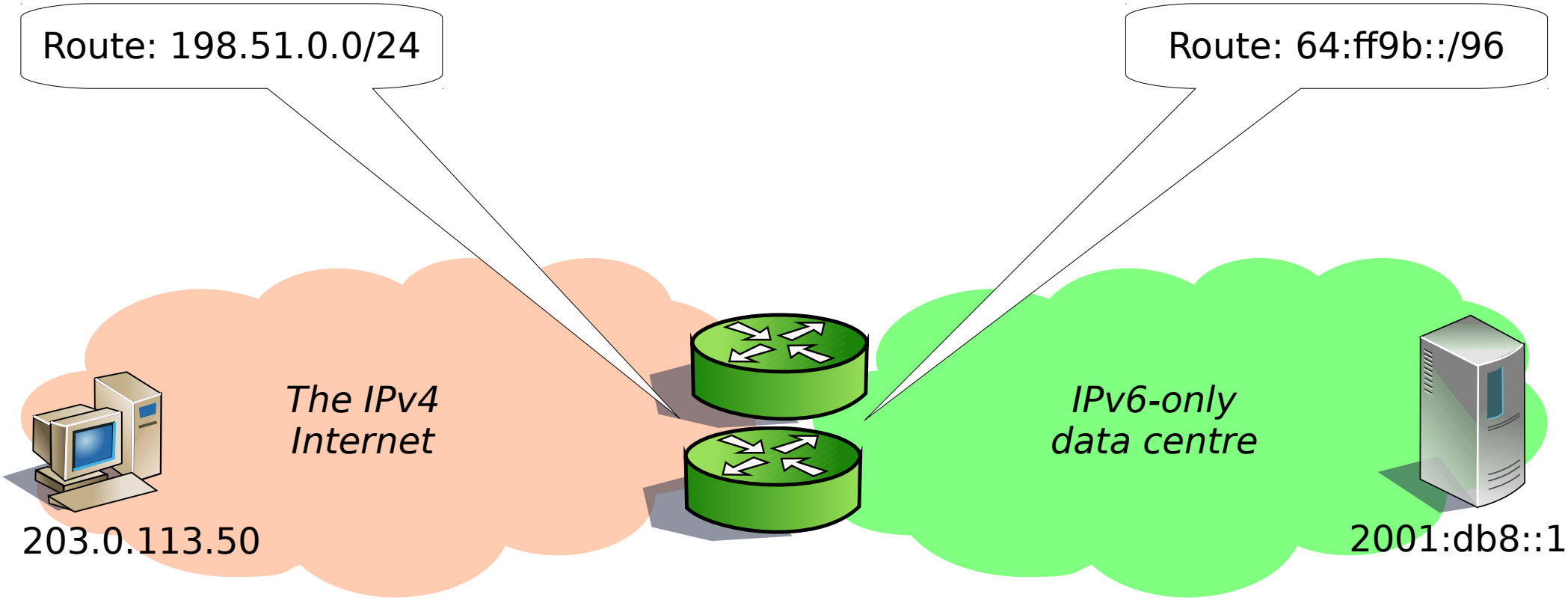
So how does it work?

- IPv4 packets are *statelessly* translated to IPv6 and vice versa by the SIIT-DC GWs [RFC 6145]
- The end user's IPv4 source address is 1:1 mapped into a 96-bit IPv6 prefix [RFC 6052]
- The service's IPv4 destination address is rewritten according to a 1:1 IPv4:IPv6 mapping configured in the SIIT-DC gateways

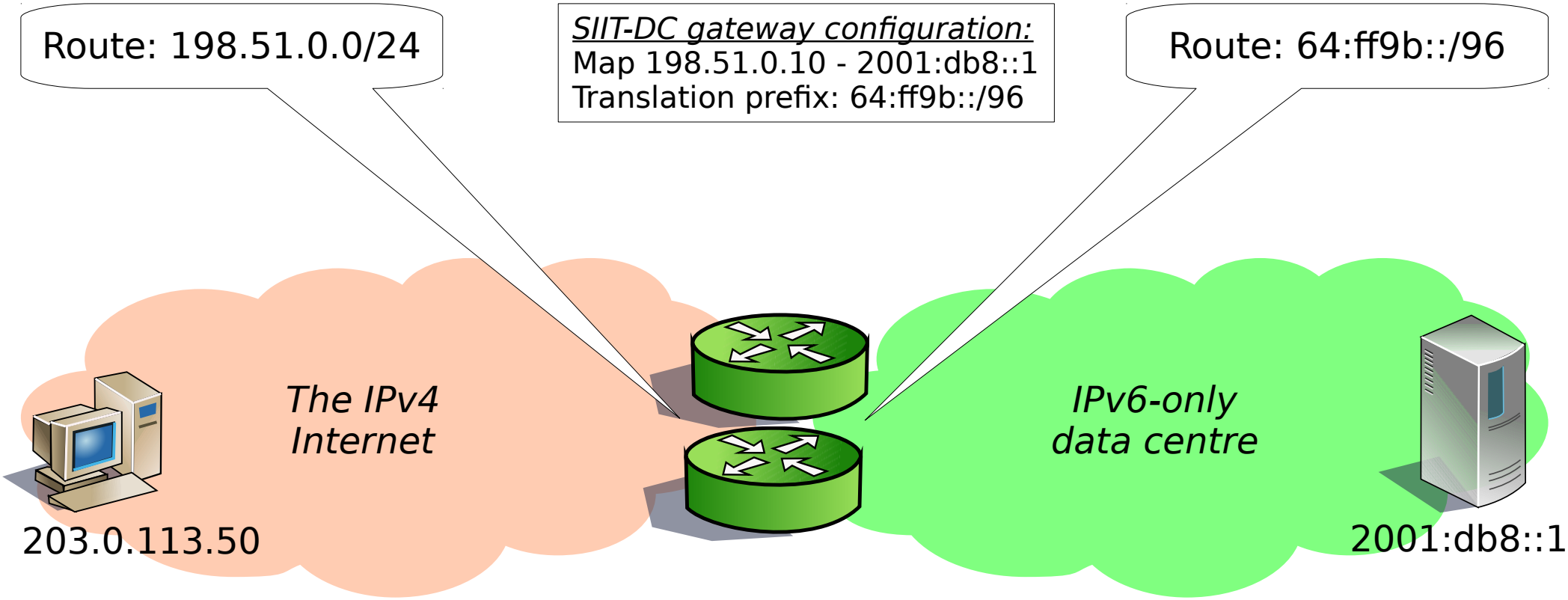




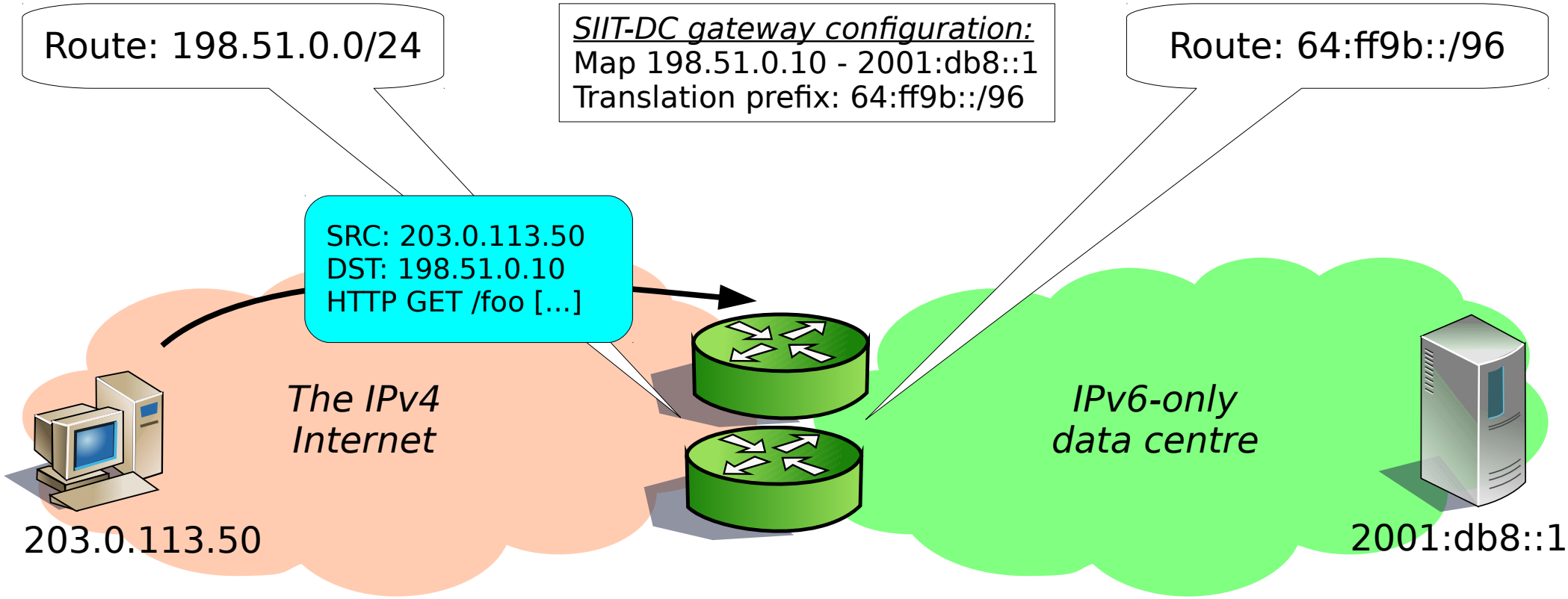
- An IPv6 /96 prefix is assigned as the translation prefix representing the IPv4 internet and routed to the SIIT-DC gateways



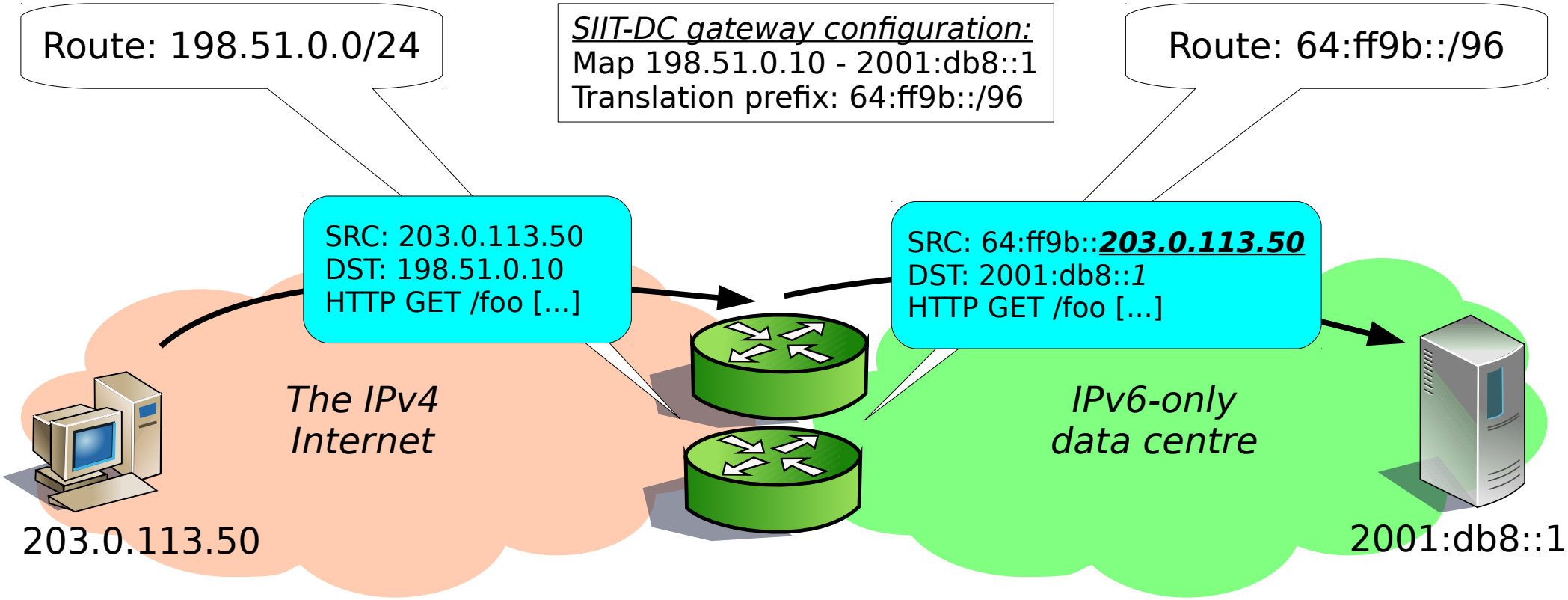
- An IPv6 /96 prefix is assigned as the translation prefix representing the IPv4 internet and routed to the SIIT-DC gateways
- A pool of IPv4 service addresses is assigned and routed to the SIIT-DC gateway



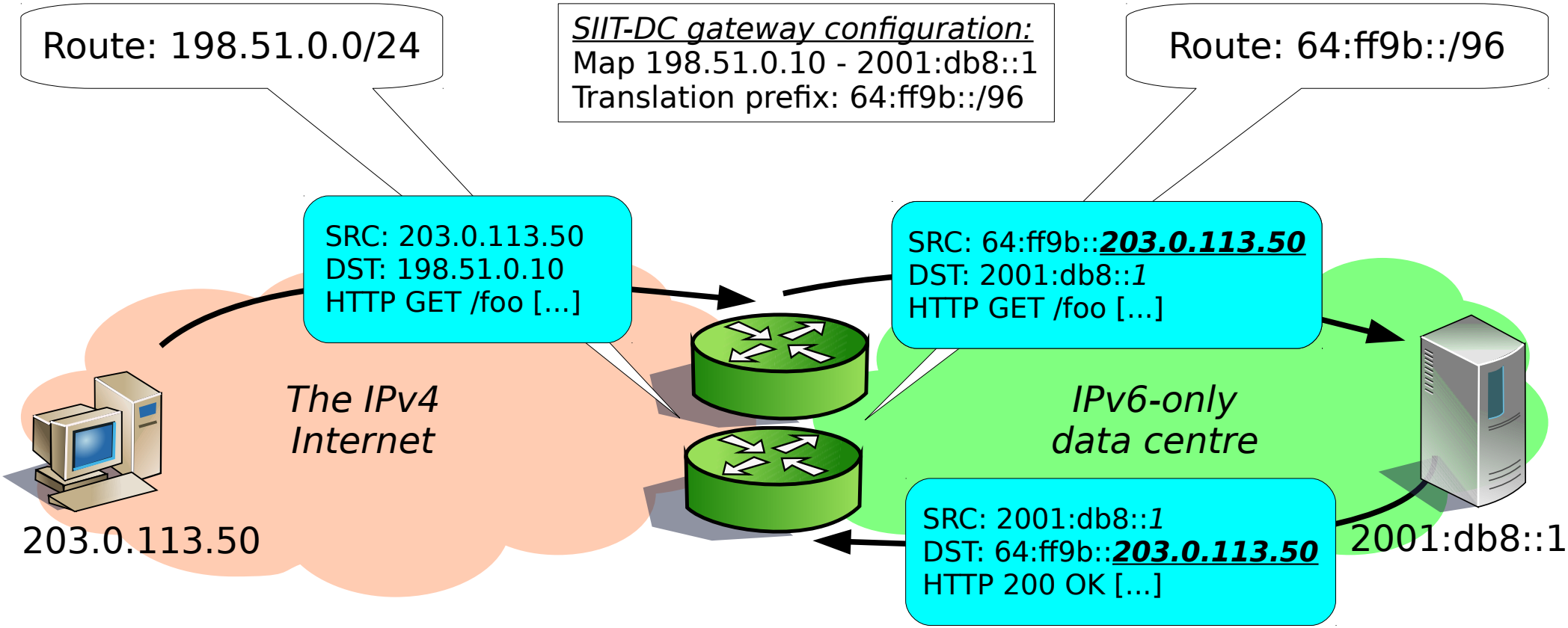
- The SIIT-DC gateway is configured with static IPv4 mappings for each IPv6 service
- The IPv6 /96 prefix is configured as a default rule (used if no static map match)
- IPv4 (IN A) records are added to DNS



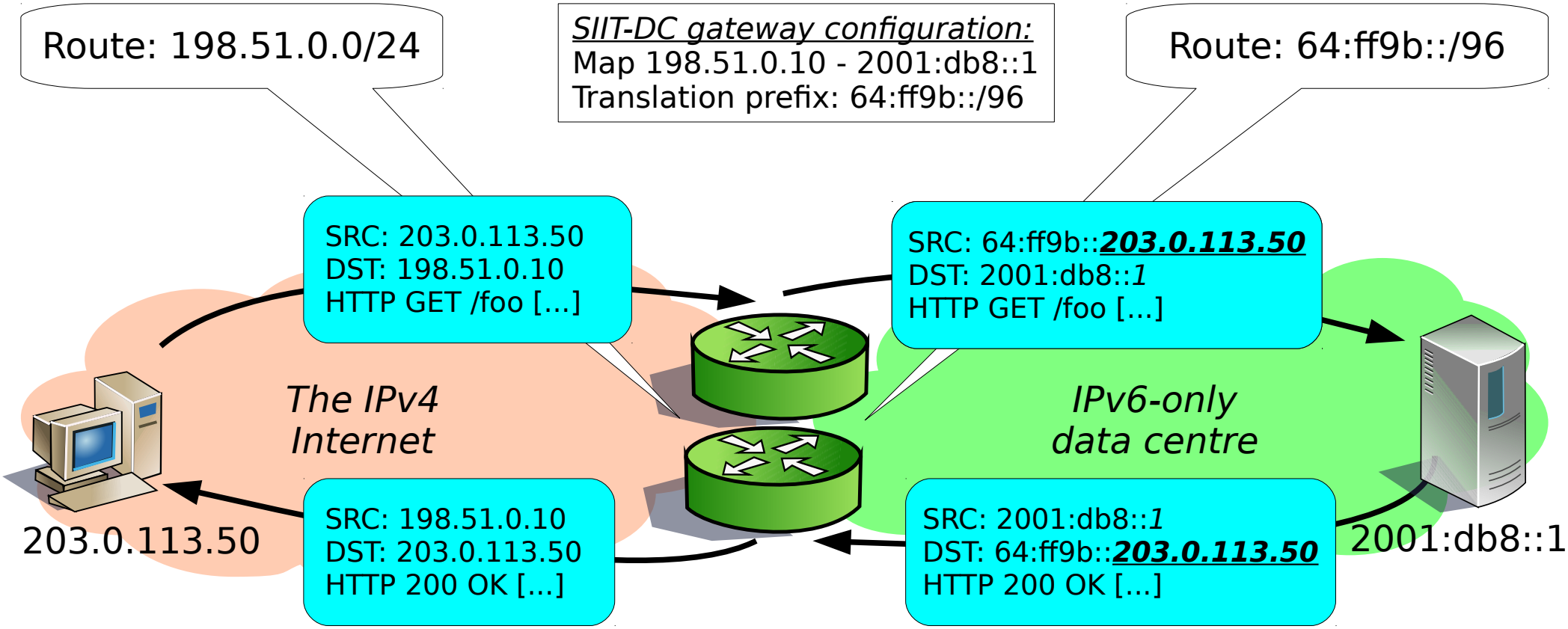
- The client looks up the service's IPv4 address in DNS and connects to it like it would with any other IPv4 address
- The IPv4 packet is routed to the SIIT-DC gateway's IPv4 interface



- The SIIT-DC gateway translates the packet to IPv6
  - DST address is 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



- The server (or load balancer) responds to the packet just as it would with any other IPv6 packet
  - The server / LB requires no SIIT-DC support or awareness
- The original IPv4 source address is not lost
- Response packet is routed back to the SIIT-DC GW



- The SIIT-DC gateway 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

Rationale for the static mapping

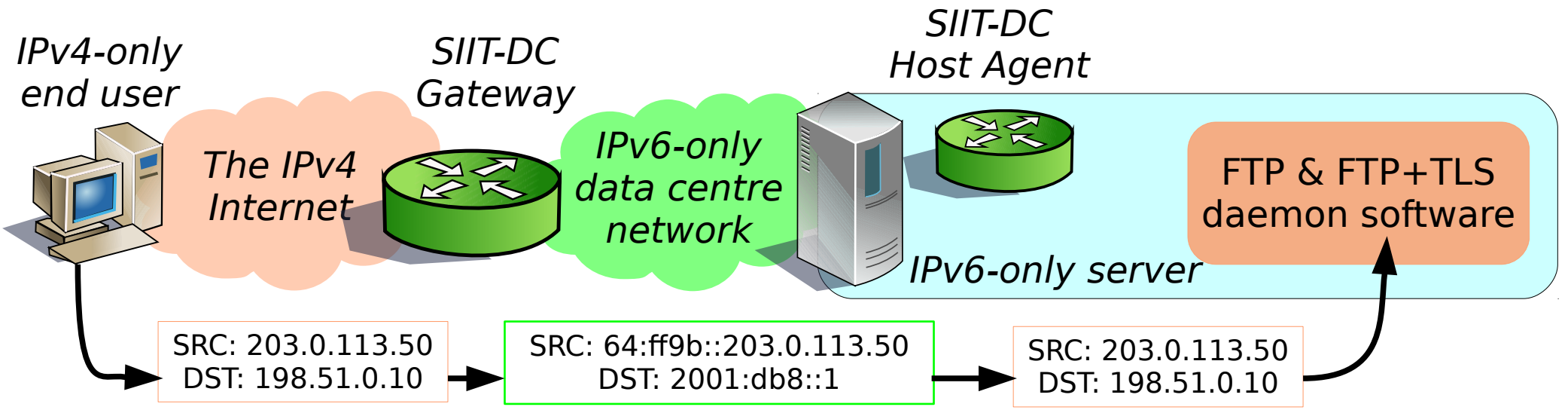
- When using RFC 6052 mapping for SRC *and* DST:
  - Servers/applications/LBs must be set up with special IPv4-mapped IPv6 addresses (in addition to their primary IPv6 address)
  - These must be routed as /128s throughout the IPv6 network and cannot be aggregated
  - Must be treated specially in ACLs, monitoring, debugging, etc.
  - Distributes complexity relating to IPv4 backwards compat across the IPv6 network and applications
  - «Dual-stack lite» - separate treatment of “IPv4” and IPv6
- When using static mappings for DST address:
  - All complexity contained within a small number of SIIT-DC GWs
  - Server admins do not make any config changes to get an IPv4 frontend («can I have an external IPv4 address for 2001:db8::1?»)

Application requirements

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

Supporting IPv4-only applications

- A Host Agent reverses the SIIT-DC Gateway's translations before passing data to the application
- Application handles IPv4 traffic on an IPv4 socket
- Very similar to the CLAT component in 464XLAT
  - End-to-end IPv4 address transparency; referrals work

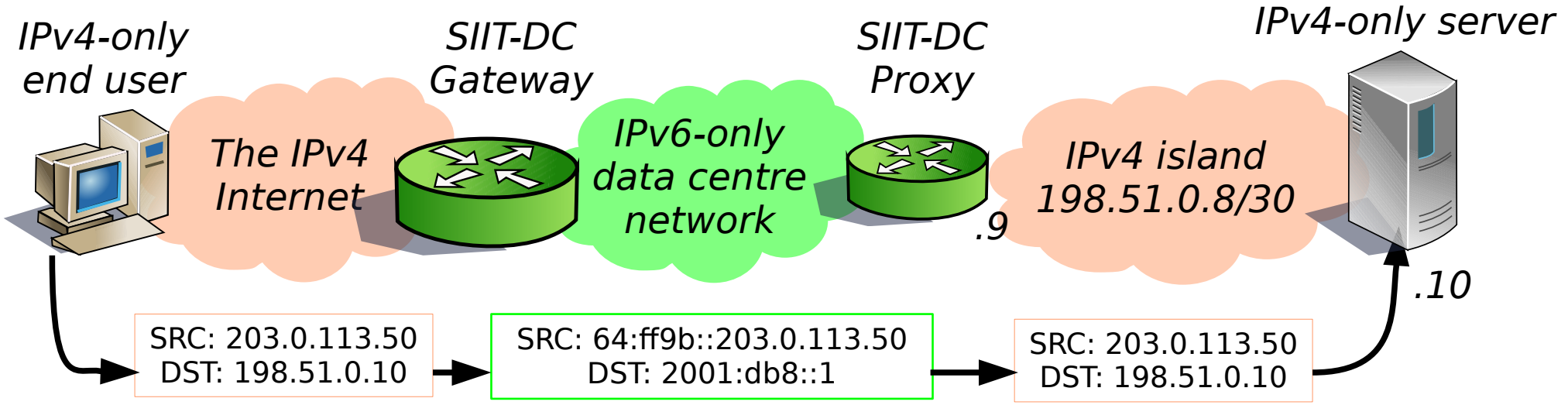


The TODO list

- Static mappings are not checksum neutral (ref. RFC6052 section 4.1) [Andrew Yourtchenko]
- Fragment Identification may be (partially) lost during translation [Andrew Yourtchenko]
- Hairpinning between IPv4-only apps/nodes behind SIIT-DC Host Agents [Andrew Yourtchenko]
- Improve (or remove) text regarding migration from dual-stack using DNS round robin [Andrew Yourtchenko]

TODO: SIIT-DC Proxy [Ray Hunter]

- A SIIT-DC Host Agent running in a “CPE” device
- Allows for supporting IPv4-only devices
- Or IPv6-capable OS-es w/o Host Agent support, which runs IPv4-only/NAT-unfriendly apps/software
  - The proxy could potentially bridge native IPv6 traffic
  - «Bump In The Wire» for IPv4 traffic



There's even running code

- TAYGA for Linux (open source)
  - http://www.litech.org/tayga/
- Cisco ASR1K
  - Requires IOS XE v3.10
- Brocade ServerIron ADX (not tested by me)
- F5 BIG-IP LTM (not tested by me)
- https://github.com/toreanderson/clatd
  - SIIT-DC Host Agent for Linux (uses TAYGA)

Questions for the WG

- Name?
  - SIIT-DC derives from SIIT (RFC 6145)
  - But people seem to think of it as «Stateless NAT46» or «Stateless NAT64» (both of which are available)
    - On the other hand, «NAT» is usually associated with port rewriting and stateful tracking, which is not done
- Anyone interested in being a co-author?
- Is it ready for WG adoption?

Questions & discussion!
Thank you for listening!

Further reading:

RFC 6052 - IPv6 Addressing of IPv4/IPv6 Translators

RFC 6145 - IP/ICMP Translation Algorithm

draft-anderson-v6ops-siit-dc - Stateless IP/ICMP Translation in IPv6 Data Centre Environments

draft-anderson-v6ops-siit-dc-2xlat – SIIT-DC: Dual Translation Mode

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-DC gateway (eating my own dog food)