DNS64 and NAT64

IPv6 Migration workshop for IETF and 3GPP

November 5-6, 2009
Shanghai, China

Simon Perreault

Viagénie

simon.perreault@viagenie.ca

http://www.viagenie.ca

Credentials

- Participation in the IETF BEHAVE WG
  - NAT traversal
    - Author of a STUN and TURN server (numb.viagenie.ca)
    - Editor of TURN-IPv6, TURN-TCP, ICE-TCP
  - IPv6 transition (== NAT removal?)
    - Author of three implementations of DNS64.
    - Working on three implementations of NAT64.
- Co-ported the Asterisk open-source PBX to IPv6.
- Ported the FreeSWITCH open-source PBX to IPv6.
- Consulting in IP networking at Viagénie in Québec, Canada.

Plan

- Tell me again: Why was NAT-PT deprecated?
- How are DNS64/NAT64 different?
- What about the other IPv6 transition technologies?
- ALGs: Do we really need them?
- How to deploy and scale?
- Our open-source DNS64/NAT64 implementations

Deprecation of NAT-PT

- Described in [RFC4996].
- Issues with NAT-PT that do not apply to NAT64:
  - The NAT-PT box has to be the default router to snoop DNS queries.
  - For dual-stack hosts, the NAT-PT DNS-ALG returns both native and translated IPv6 addresses.
  - The NAT-PT DNS-ALG erroneously translates responses to DNS A queries from IPv6 network (and vice-versa).
  - One NAT binding is created per converted DNS RR, which may be more than needed.
  - NAT-PT is fully incompatible with DNSSEC.

NAT-PT for 3GPP

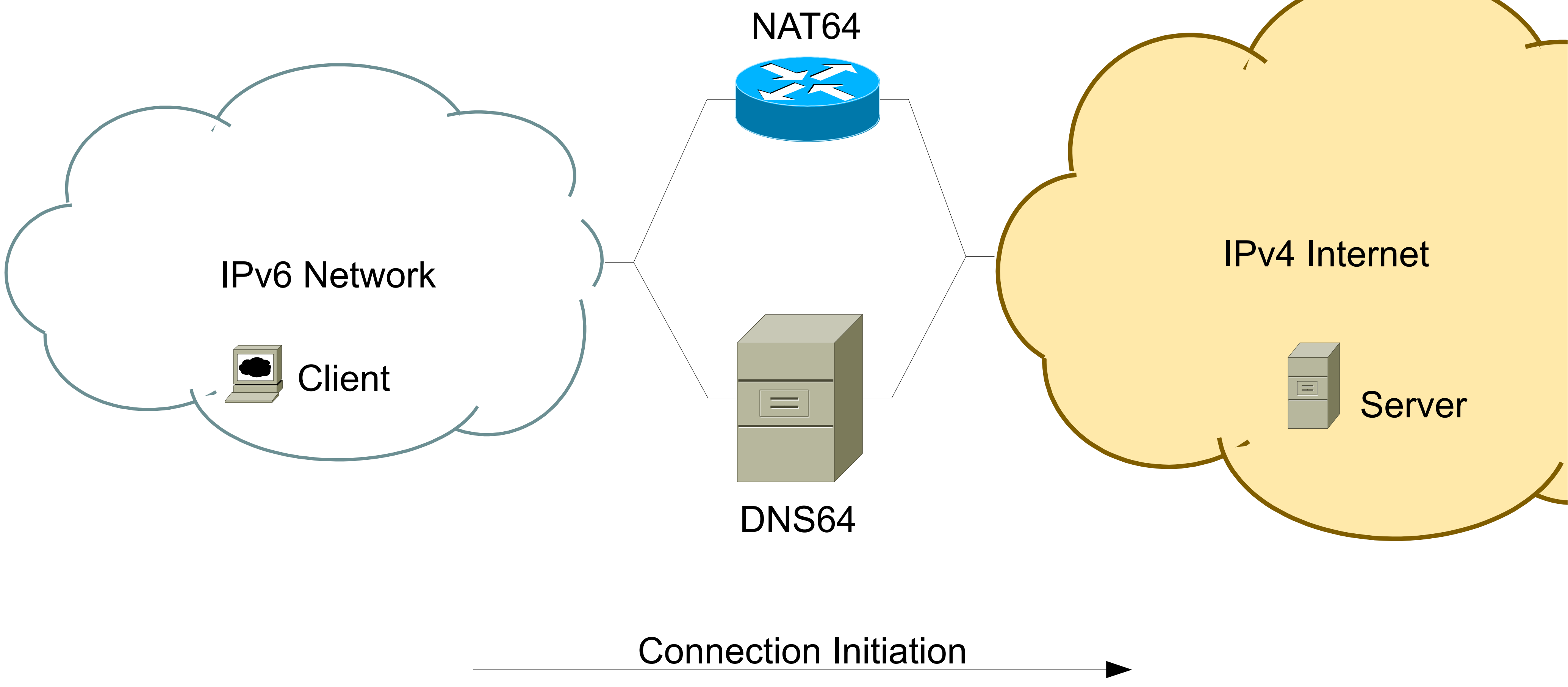
- From RFC4215:

```
Appendix A - On the Use of Generic Translators in the 3GPP Networks

[...]

   To minimize the problems associated with NAPT-PT, the following
   actions can be recommended:

      1. Separate the DNS ALG from the NAPT-PT node (in the "IPv6 to
         IPv4" case).

      2. Ensure (if possible) that NAPT-PT does not become a single
         point of failure.

      3. Allow for load sharing between different translators.  That is,
         it should be possible for different connections to go through
         different translators.  Note that load sharing alone does not
         prevent NAPT-PT from becoming a single point of failure.
```

DNS64 and NAT64

- DNS-ALG and translation functions are now separate.
- DNS64 is designed with DNSSEC in mind.
- NAT64 only deals with connections initiated from IPv6 to IPv4.
  - Constraining the problem space generally results in simpler, cleaner, more robust and scalable solutions.
- NAT64 doesn't need to be the default router.
- NAT64 benefits from better knowledge of NAT.

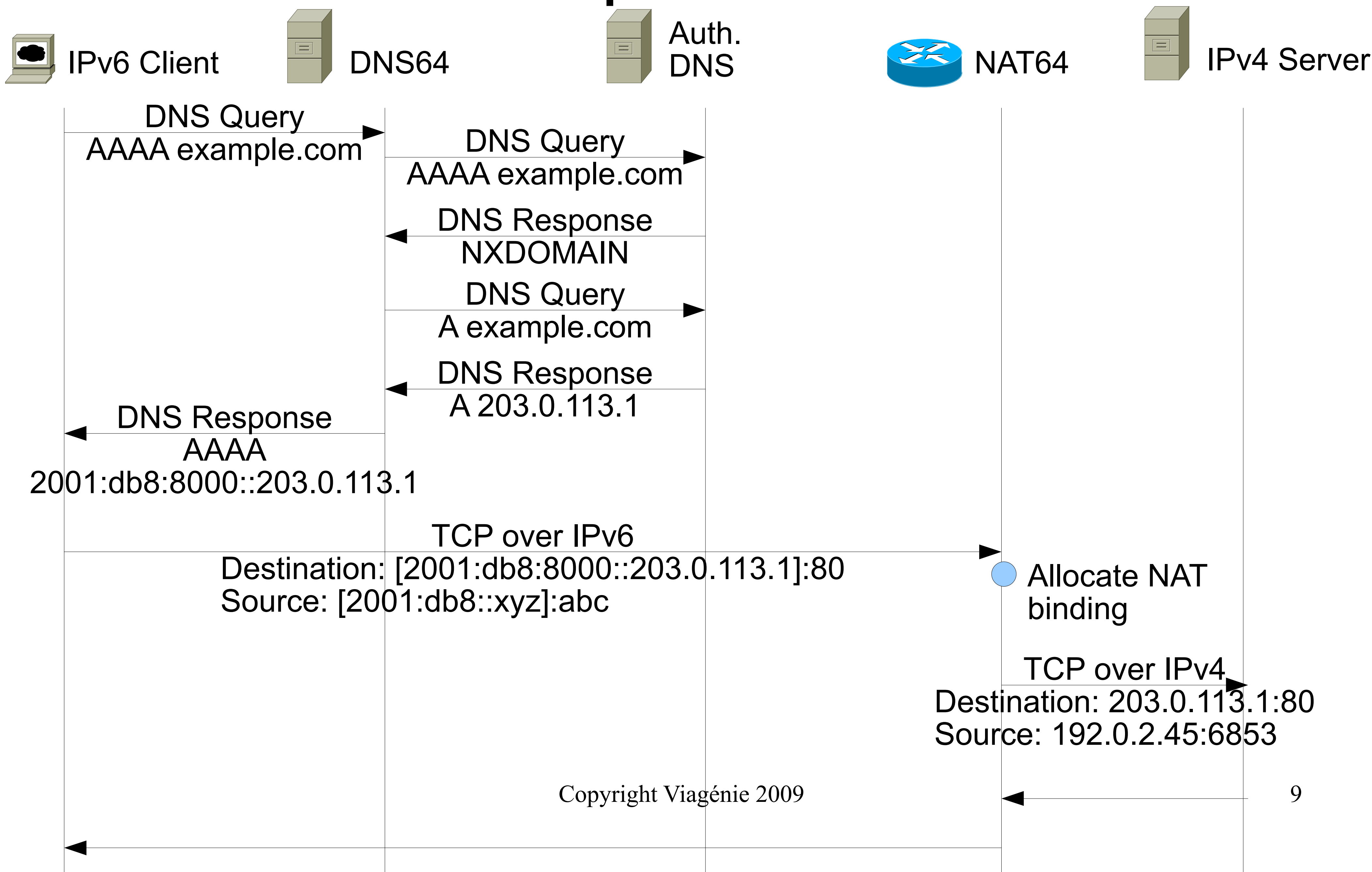
IPv6 network to IPv4 Internet



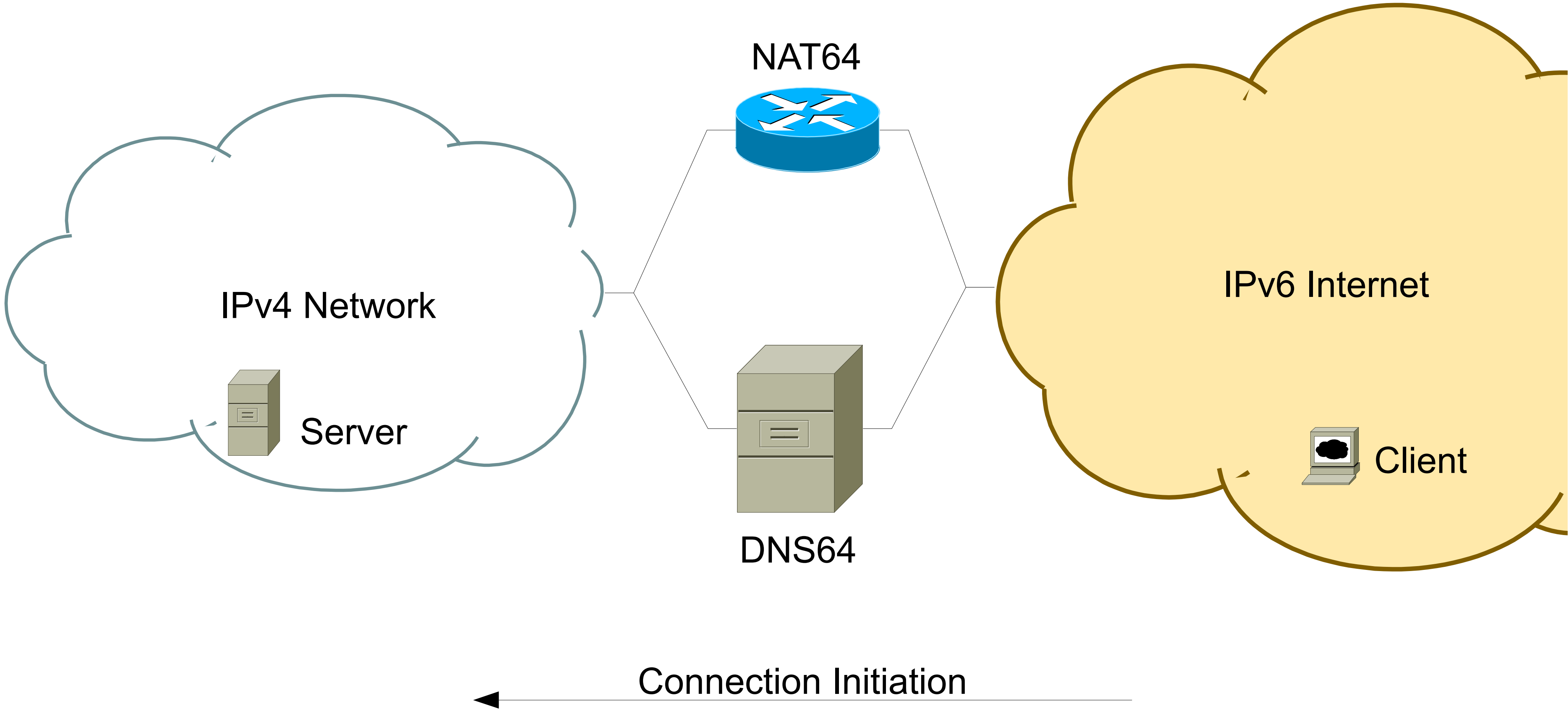
Example

- IPv6 network → 2001:db8::/64
- NAT64 → 2001:db8::1
- DNS64 → 2001:db8::2
- NAT64 public IPv4 pool → 192.0.2.0/24
- Pref64::/n → 2001:db8:8000::/96

Example


IPv6 Client
DNS64
Auth. DNS
NAT64
IPv4 Server
DNS Query
AAAA example.com
DNS Query
AAAA example.com
DNS Response
NXDOMAIN
DNS Query
A example.com
DNS Response
A 203.0.113.1
DNS Response
AAAA
2001:db8:8000::203.0.113.1
TCP over IPv6
Destination: [2001:db8:8000::203.0.113.1]:80
Source: [2001:db8::xyz]:abc
Allocate NAT binding
TCP over IPv4
Destination: 203.0.113.1:80
Source: 192.0.2.45:6853

IPv6 Internet to IPv4 network

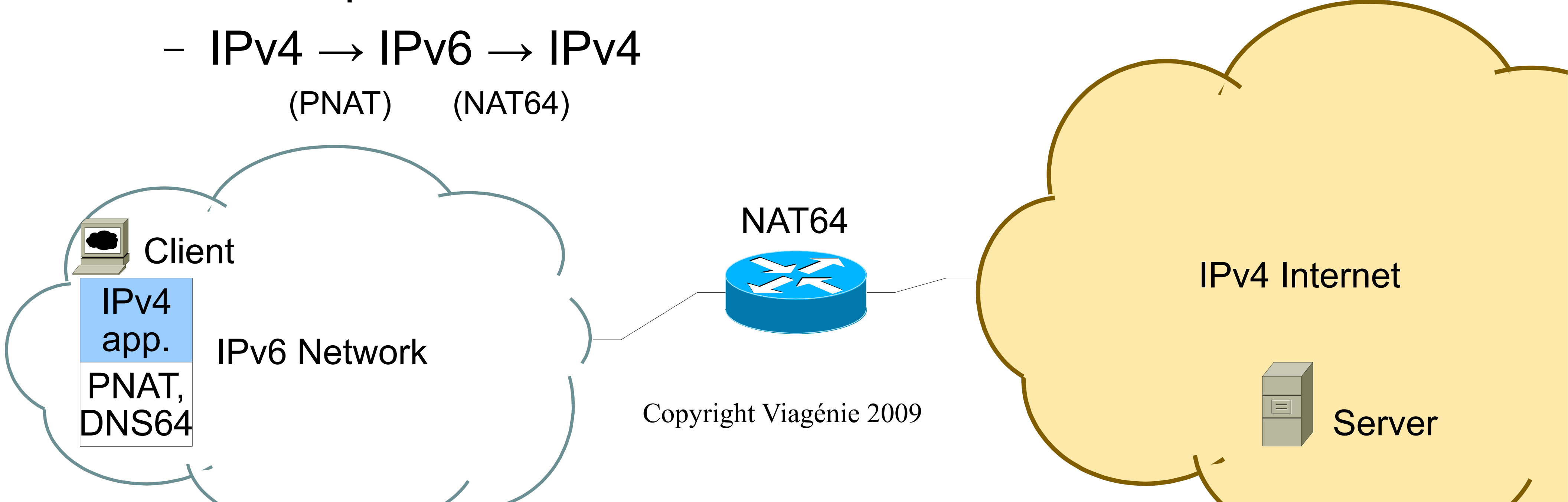


NAT64 uses the least amount of evil possible

- Builds upon years of work on IPv4 NATs by BEHAVE working group.
  - [RFC4787], [RFC5382], [RFC5508], [RFC5389], and others
- NAT mapping behavior is endpoint-independent.
- NAT filtering is optional.
  - If enabled, behavior is address-dependent.
- TCP simultaneous-open works.
- Hairpinning works.
- Result: NAT traversal works (e.g. SIP with ICE)

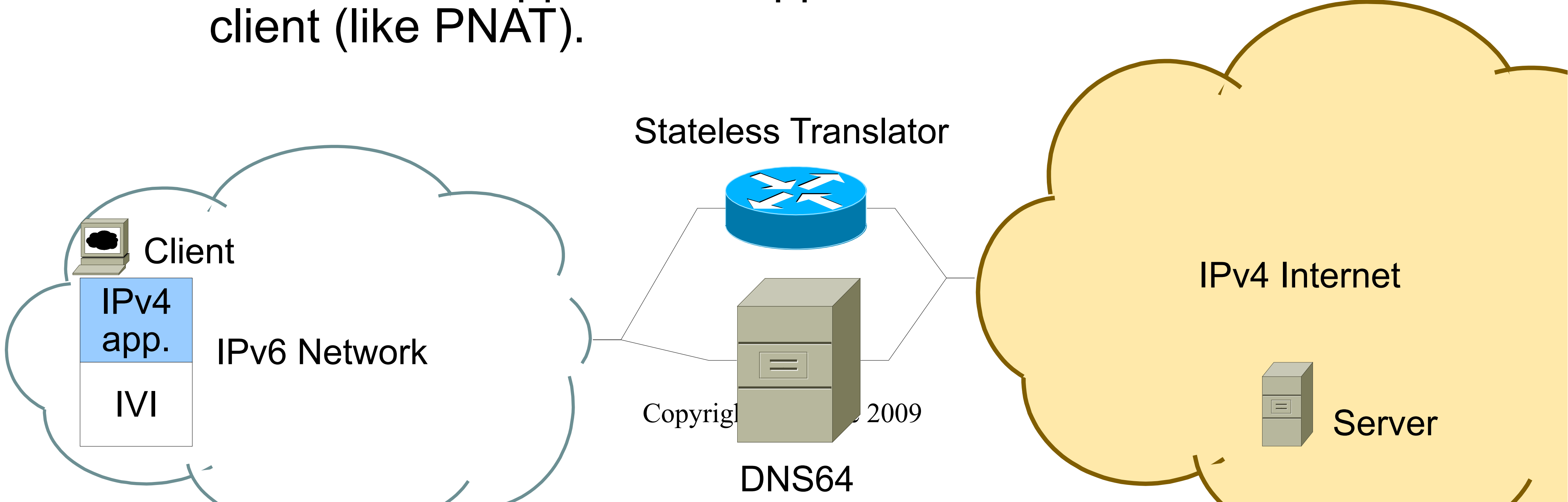
NAT64 vs. others

- PNAT [draft-huang-behave-pnat]
  - Not in conflict with NAT64.
  - PNAT is a bump-in-the-stack technology to enable IPv4 applications to use IPv6 network for communication.
  - There needs to be a NAT64 in the network (modified a little).
  - PNAT performs DNS64 in the host.
  - $\text{IPv4} \underset{\text{(PNAT)}}{\rightarrow} \text{IPv6} \underset{\text{(NAT64)}}{\rightarrow} \text{IPv4}$



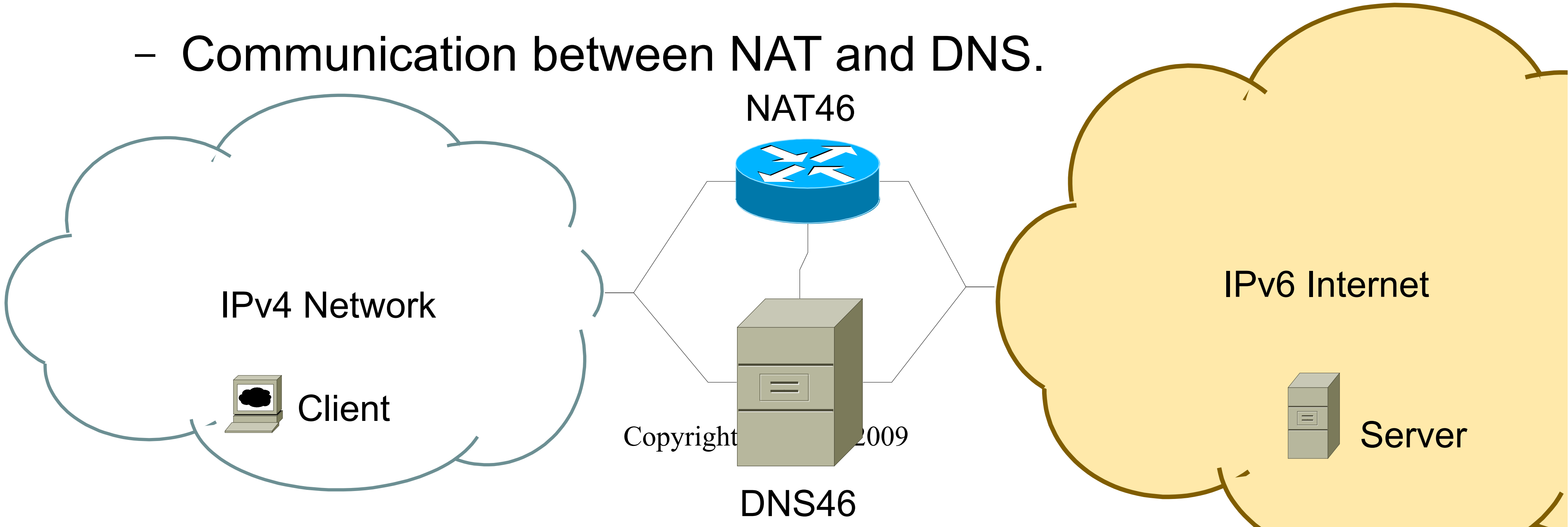
NAT64 vs. others

- IVI [draft-xli-behave-ivi]
  - Not in conflict with NAT64.
  - Constraint: IPv4 public address pool big enough to assign one IPv4 address to each IPv6-only client.
  - Stateless.
  - Double IVI: Support IPv4 applications on IPv6-connected client (like PNAT).



NAT64 vs. others

- Virtual IPv6 connectivity [draft-vogt-durand-virtual-ip6-connectivity]
  - Not in conflict with NAT64.
  - Different use case: IPv4 network to IPv6 Internet.
  - Support legacy IPv4 client devices when IPv6 Internet is well developed.
  - Stateful.
  - Communication between NAT and DNS.



ALGs

VIAGĒNIE

- From [RFC4924], “Reflections on Internet Transparency” (from Internet Architecture Board):

```
No matter how well an ALG is implemented,
barriers to transparency will emerge over
time, so that the notion of a "transparent
ALG" is a contradiction in terms.
```

- At the moment, an FTP ALG draft is being considered for adoption in the BEHAVE working group.
  - IPv6 client behind NAT64 thinks it is talking to IPv6 server and send an EPASV or EPORT command.
  - Server is really IPv4 and doesn't understand EPASV or EPORT.

ALGs

- SIP transition plan is to use ICE and TURN. [draft-ietf-sipping-v6-transition]
  - IPvX-only client asks TURN server for both IPv4 and IPv6 address allocations.
  - TURN server relays between IPv4 and IPv6.
  - This method could be applicable to many other protocols.
- URLs with IPv4 address literals
  - 2.38% of Alexa's top 1 million websites contain them.
  - HTTP proxy is more resource-intensive than NAT64.
  - Workaround: proxy auto-config file with regular expression. [draft-wing-behave-http-ip-address-literals]

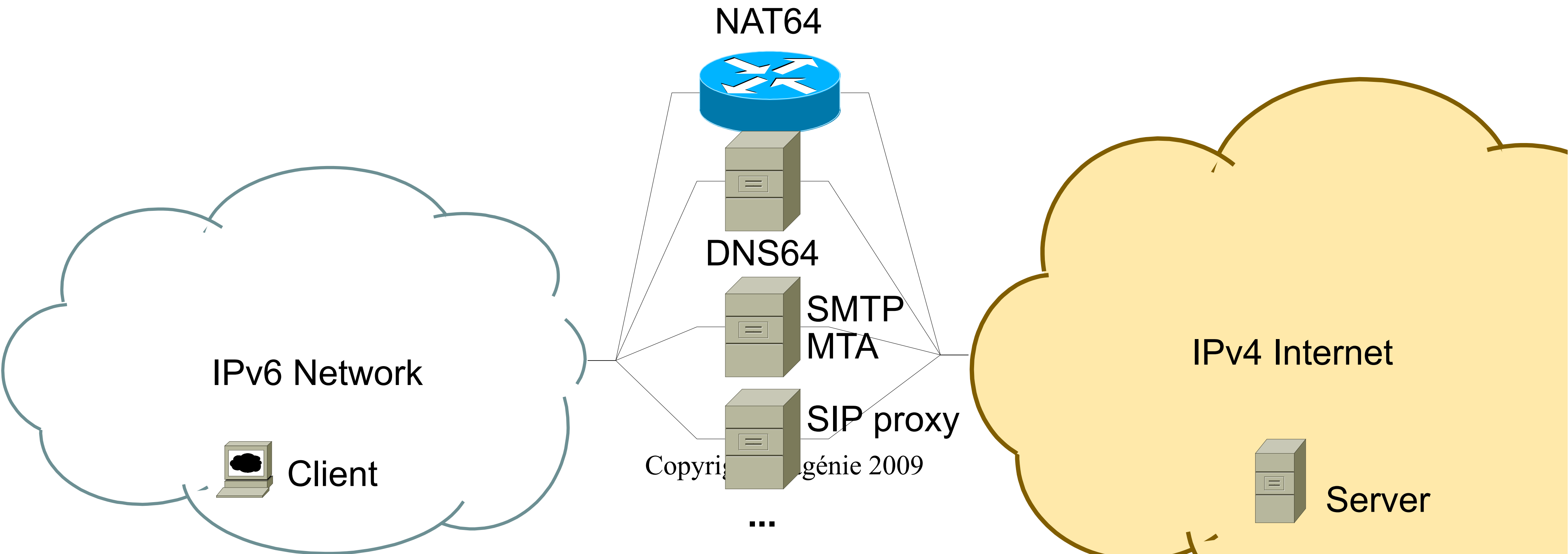
Deployment

- Scaling
  - Same mechanisms as NAT44
    - Cold standby, hot standby (e.g. VRRP)
    - Synchronizing state (e.g. pfsync)
    - See e.g. [draft-xu-behave-nat64-standby]
  - DNS64-based load balancing
    - Multiple NAT64 boxes, each with its own Pref64::/n.
    - Sems to be safe:
      - Choose a Pref64::/n based on DNS query destination address.
    - Seems to not be so safe:
      - Choose a Pref64::/n based on DNS query source address.

Deployment

- From [draft-ietf-behave-v6v4-framework]:

```
As a general rule, a simple operational
recommendation will work around many
application issues, which is that there
should be a server in each domain or an
instance of the server should have an
interface in each domain
```



Ecdysis: Open-Source DNS64 and NAT64

- Funded by NLnet and Viagénie.
- Ecdysis refers to the molting of the cuticula in arthropods, as an analogy of IPv4 molting into IPv6. After molting, the arthropod is fresh and ready to grow! Arthropods is also the expertise of the 5 years old son of one of the project leads...

Ecdysis: Open-Source DNS64 and NAT64

- Three open-source implementations of DNS64
  - A stand-alone implementation written in Perl for experimentation purposes.
  - A patch for Bind.
  - A patch for Unbound.
  - Available now at **http://ecdysis.viagenie.ca**
- Three open-source implementations of NAT64
  - A stand-alone implementation using libpcap for experimentation purposes.
  - A patch for OpenBSD's pf.
  - A patch for Linux's Netfilter (iptables).
  - Available soon...

Implementation Considerations

- The “good NAT” behavior is different from that of pf and Netfilter. Needs separate state data structures.
- NAT64 changes the whole headers at once. pf works in two separate phases: destination first, then source.
  - Can't fit in the translation model pf is based on.
- Unbound is modular, Bind is monolithic.
- Found issue with TTL of synthetic AAAA records. Solution now part of the spec.
  - TTL(synth. AAAA) = min(TTL(A), TTL(SOA))
- IPv4 access for DNS64 server not needed.
  - Is this useful? Maybe not.

Conclusion

- NAT64 is a part of your IPv6 transition toolbox.
- Don't over-engineer it. It's only for transition.

Questions?

simon.perreault@viagenie.ca

This presentation: http://www.viagenie.ca/publications/

References

- Open-source DNS64 and NAT64: http://ecdysis.viagenie.ca
- NAT64: [draft-ietf-behave-xlate-stateful]
- DNS64: [draft-ietf-behave-dns64]