NAT64Check

Version 2

Sander Steffann
sander@steffann.nl

Jan Žorž
zorz@isoc.org

steffann

Internet Society

go6

Problem statement and real world status

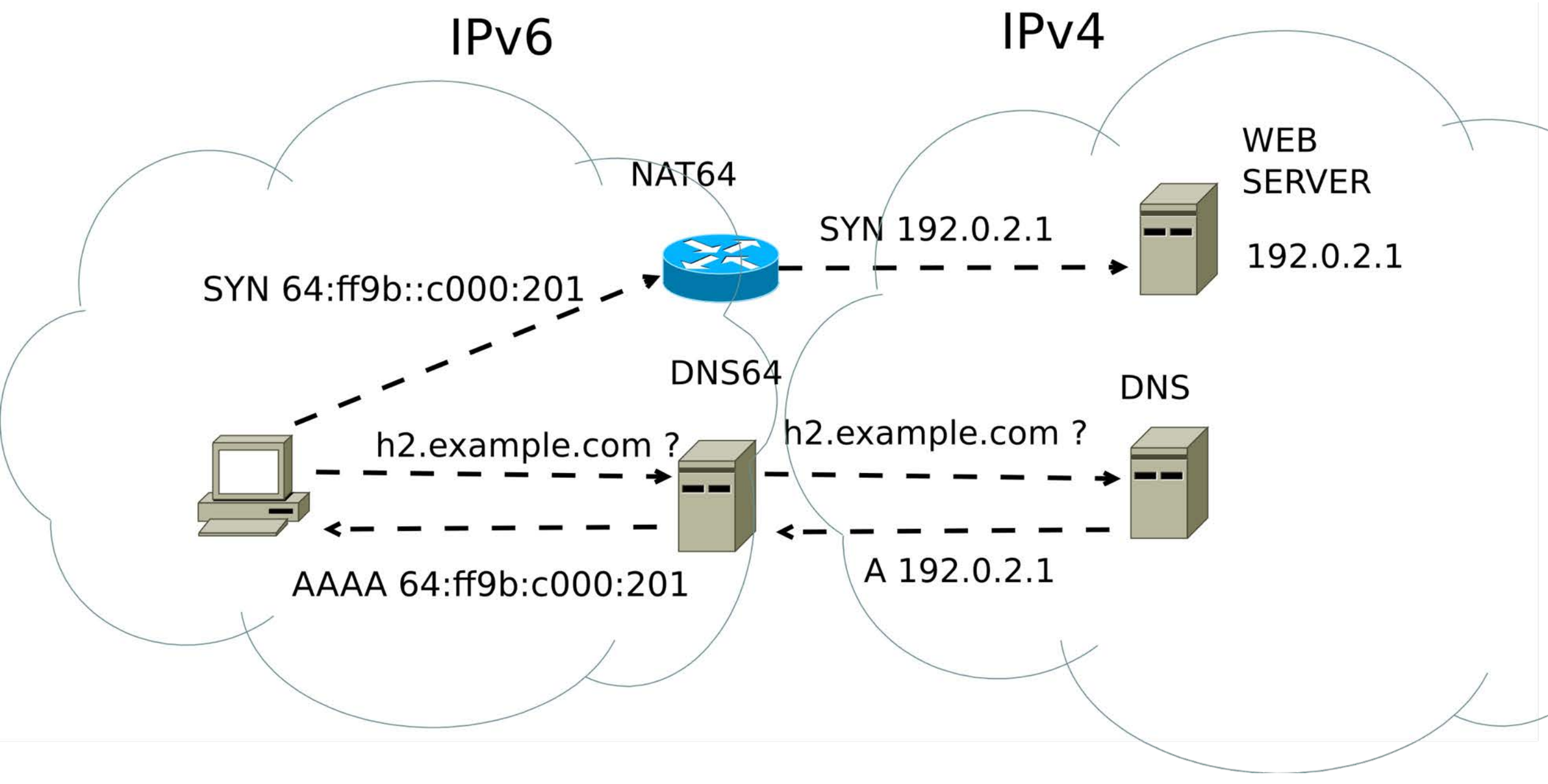
(Six Degrees of Inner Turbulence)

IPv6 and IPv4 are incompatible on the wire.

- We need transition and translation mechanisms between the two protocols
- Mobile operators are massively switching devices to IPv6-only connectivity
  - millions of users
  - using 464XLAT (Android) or plain NAT64 (Apple) as a transition tool to access IPv4 content
- Some people do weird stuff while adding AAAA to their DNS records
- Important questions:
  - Do content providers know how their content will be seen from such environments?
  - Do connectivity providers know what their users' experience on IPv6-only would be?

Some dark magic behind NAT64/DNS64

(Paralysed)



Go6lab NAT64/DNS64 public test-bed

(The Test That Stumped Them All)

Aimed at everyone who would like to test NAT64/DNS64 functionality

4 different implementations, 4 different instructions how to direct traffic there

Used by operators

- Testing the idea of providing NAT64/DNS64 and/or 464XLAT to their users

Used by application providers

- To see how their apps works in NAT64/DNS64 environment

Used by HW/SW vendors

- Testing their solutions against multiple NAT64 vendors

Gained quite some traction and momentum this days

Instructions: https://go6lab.si/current-ipv6-tests/nat64dns64-public-test/

Go6lab NAT64/DNS64 test-bed

https://go6lab.si/current-ipv6-tests/nat64dns64-public-test/

NAT64/DNS64 public test

Go6lab is hosting a variety of NAT64/DNS64 solutions, open for general Internet public for testing.

Disclaimer: This setup is not intended for performance testing, just to see how NAT64/DNS64 operates, what applications breaks and what are differences between different implementations. If you need to do performance testing of NAT64/DNS64 send email to <zavod@go6.si> and schedule a test session in go6lab facility.

To test different NAT64/DNS64 setups you need to disable IPv4 on your device and set an IPv6 resolving DNS server, different one for each setup. Please send us your observations, specially about which applications breaks in IPv6-only/NAT64 environment. We would like to build a list of them.

https://go6lab.si/current-ipv6-tests/nat64dns64-public-test/

A10 Networks NAT64 implementation: set your DNS to **2001:67c:27e4:15::6411**
NAT64 implementation is running on a A10 vThunder virtual appliance.
NAT64 routed prefix: 2001:67c:27e4:642::/64
Quick ping6 test if up&running: ping6 2001:67c:27e4:642::5bef:6015

PaloAlto Networks Firewall NAT64 with BIND9.9 DNS64: set your DNS to **2001:67c:27e4::64**
NAT64 implementation is running in PAN500 firewall box.
NAT64 routed prefix: 2001:67c:27e4:64::/64
Quick ping6 test if up&running: ping6 2001:67c:27e4:64::5bef:6015

Jool NAT64 with BIND9 DNS64: set your DNS to **2001:67c:27e4:15::64**
Jool NAT64 implementation is running in a virtual container on proxmox server.
NAT64 routed prefix: 2001:67c:27e4:1064::/64
Quick ping6 test if up&running: ping6 2001:67c:27e4:1064::5bef:6015

Cisco ASR1000 NAT64 with BIND9 DNS64: set your DNS to **2001:67c:27e4::60**
NAT64 implementation is running in Cisco ASR1001.
NAT64 routed prefix: 2001:67c:27e4:11::/64
Quick ping6 test if up&running: ping6 2001:67c:27e4:11::5bef:6015

Go6lab NAT64/DNS64 test-bed HW/SW

(Systematic Chaos)

Some DNS admins put “crap” in AAAA records

(The Enemy Inside)

Have you ever seen any of this values for AAAA record?

::

::1

::ffff:[IPv4_addr]

fe80::[some_value]

64:ff9b::[some_value]

2001:DB8::[some_value]

If you have seen something like this and you know who did it, talk to that people and tell them to fix it. This sort of thing isn’t useful to anybody and severely impacts user experience.

More: http://www.employees.org/~dwing/aaaa-stats/ (courtesy of Dan Wing)

When deploying in real life you need “fixes”

(Build Me Up, Break Me Down)

So, what can we do about it?

We can figure out who this people are, contact them, warn them about the issue, educate them and ask them to fix the problem.

Remember: If you are not part of solution, you are part of the problem.

At the same time we can protect our users from bad user experience and set the “exclude” rules in our DNS64 servers. IANA allocated 2000::/3 as global unicast address pool, so whatever else is used in AAAA - it’s by default bogus and we can safely ignore that.

When deploying in real life you need “fixes”

(Stream of Consciousness)

BIND9 example of DNS64 configuration in go6lab:

```
dns64 2001:67c:27e4:64::/96 {
        clients { any; };
        mapped { !rfc1918; any; };
        exclude { 0::/3; 4000::/2; 8000::/1; 2001:DB8::/32; };
        break-dnssec yes;
};
```

Explanation of “exclude” configuration parameter: If DNS64 server gets an AAAA record with a value of anything outside 2000::/3 - it ignores it and synthesizes the AAAA record from NAT64_prefix::IPv4_address

When deploying in real life you need “fixes”

(Stream of Consciousness)

Unbound example of DNS64 configuration in go6lab:

```
server:
        module-config: "dns64 validator iterator"
        dns64-prefix: 2001:67c:27e4:64::0/96
        private-address: 0::/3
        private-address: 4000::/2
        private-address: 8000::/1
        private-address: 2001:DB8::/32
#        private-address: 64:FF9B::/96
#        private-address: ::ffff:0:0/96
#        private-address: ::1/128
#        private-address: ::/128
        interface: 2001:67c:27e4::64
```

Project history

(Falling into infinity)

- Version 0:
  - Jan wrote some Bash scripts
- Version 1:
  - Sander wrote a Python/Django application
  - Corinne designed a web interface

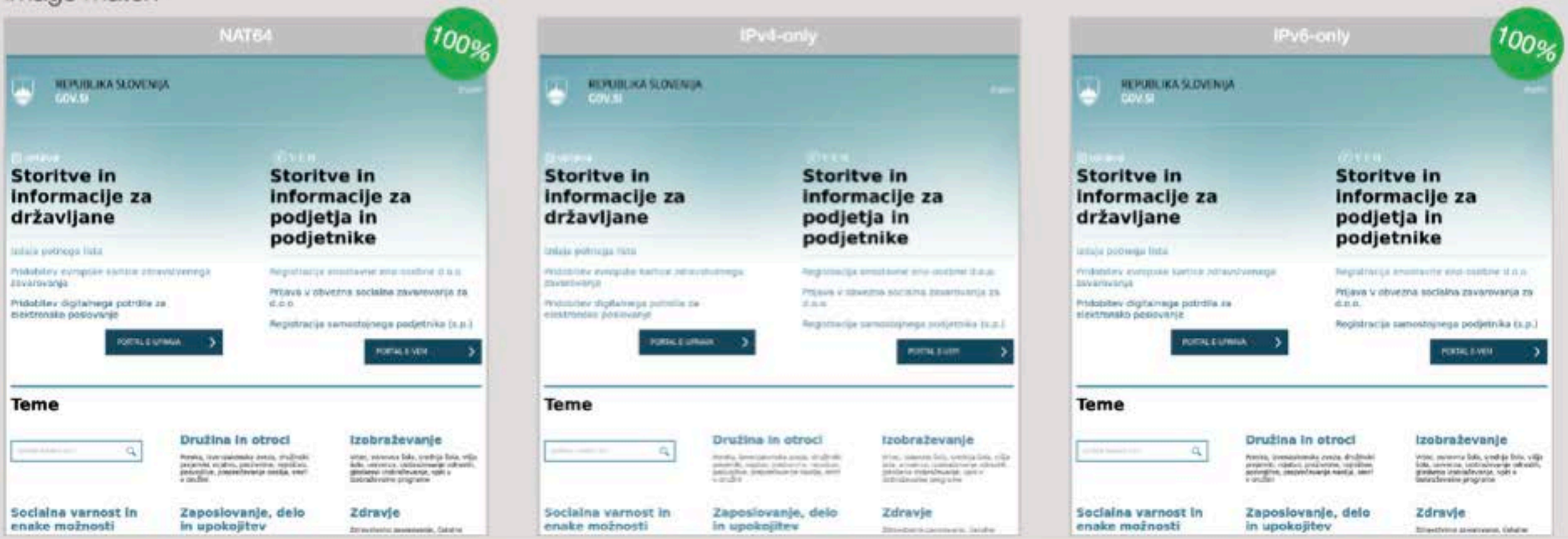
nat64check

www.gov.si

Test again

Last test finished:
6 September 2018, 12:07 CEST

Image match



THE GOOD

(The mirror)

nat64check

textfiles.com

Test again

Last test finished:
6 September 2018, 12:18 CEST

Image match

NAT64	IPv4-only	IPv6-only
100%		

THE BAD

(The mirror)

nat64check

foxnews.com

Test again

Last test finished:
28 May 2018, 10:26 CEST

Image match

NAT64

IPv4-only

IPv6-only

THE UGLY

(The mirror)

Version 1 flaws

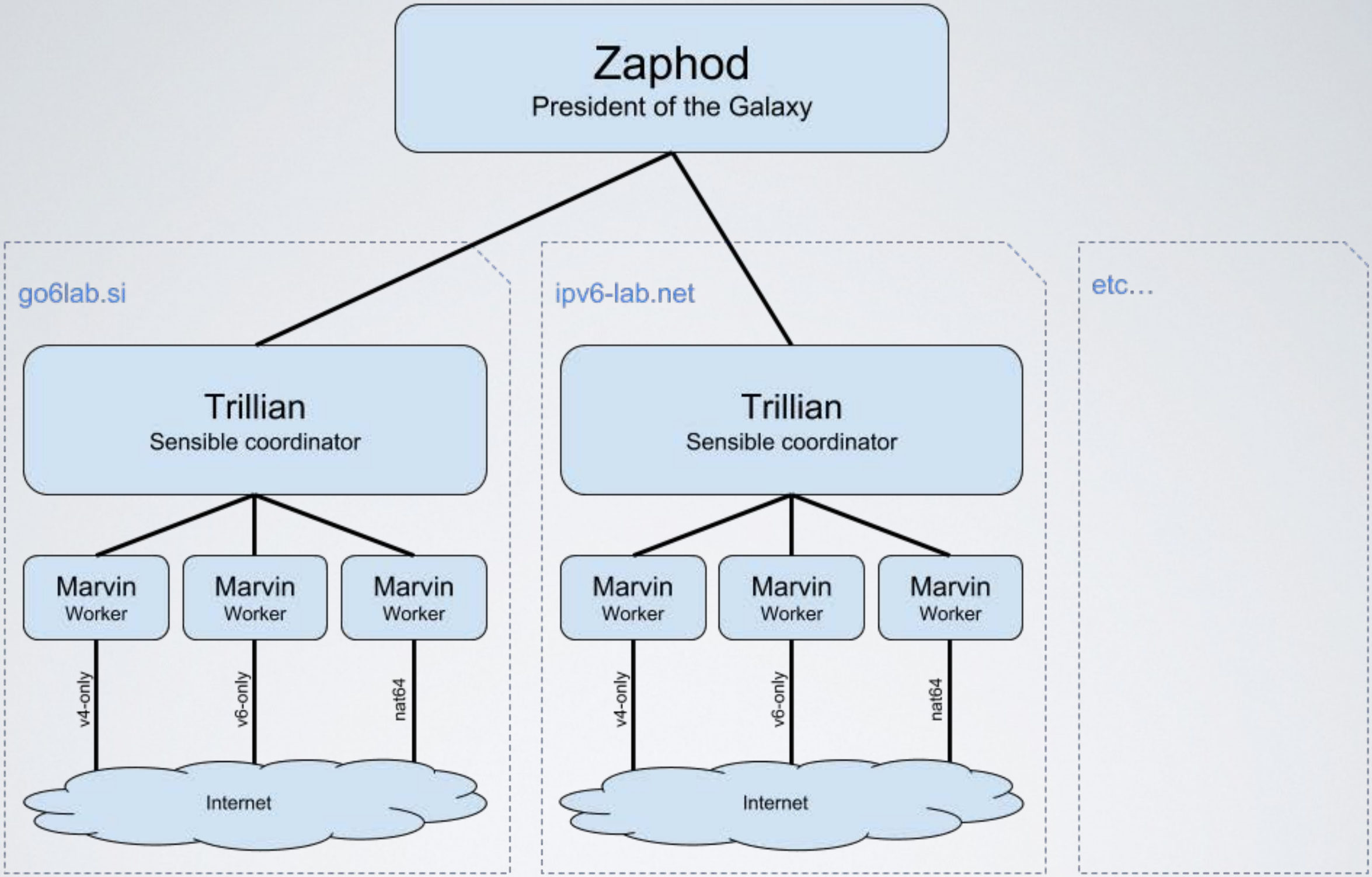
(A change of seasons)

- Unclear what the cause of a failure is
- Bad error detection
- Two different systems:
  - ipv6-lab.net in NL
  - go6lab.si in SI
- No coordination between them
- Still online at https://nat64check.go6lab.si/

VERSION 2

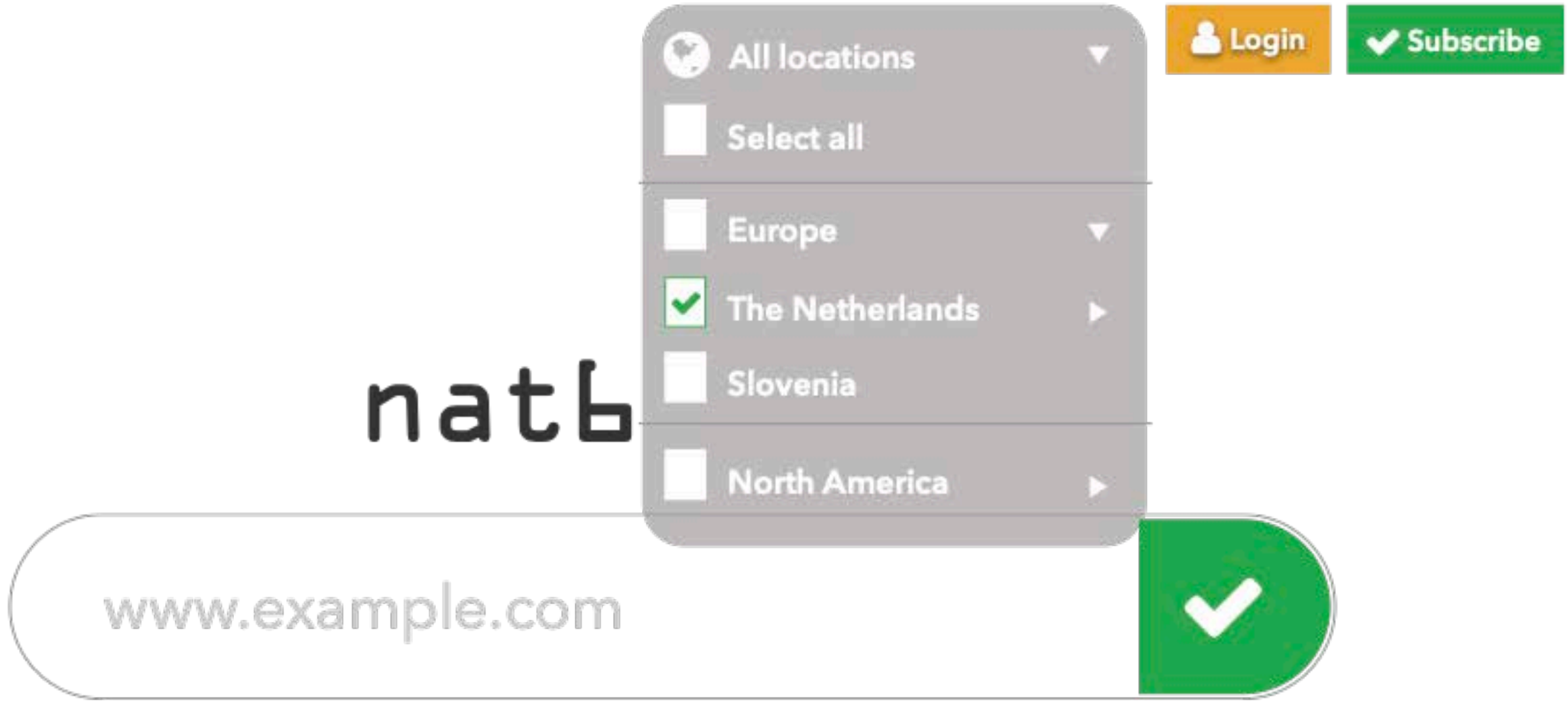
(Systematic Chaos)

- Distributed system
  - Central web interface
  - Many test-locations possible
  - Easy to install if you want to run your own test-location
- Better error detection and feedback
- Extendability for different tests



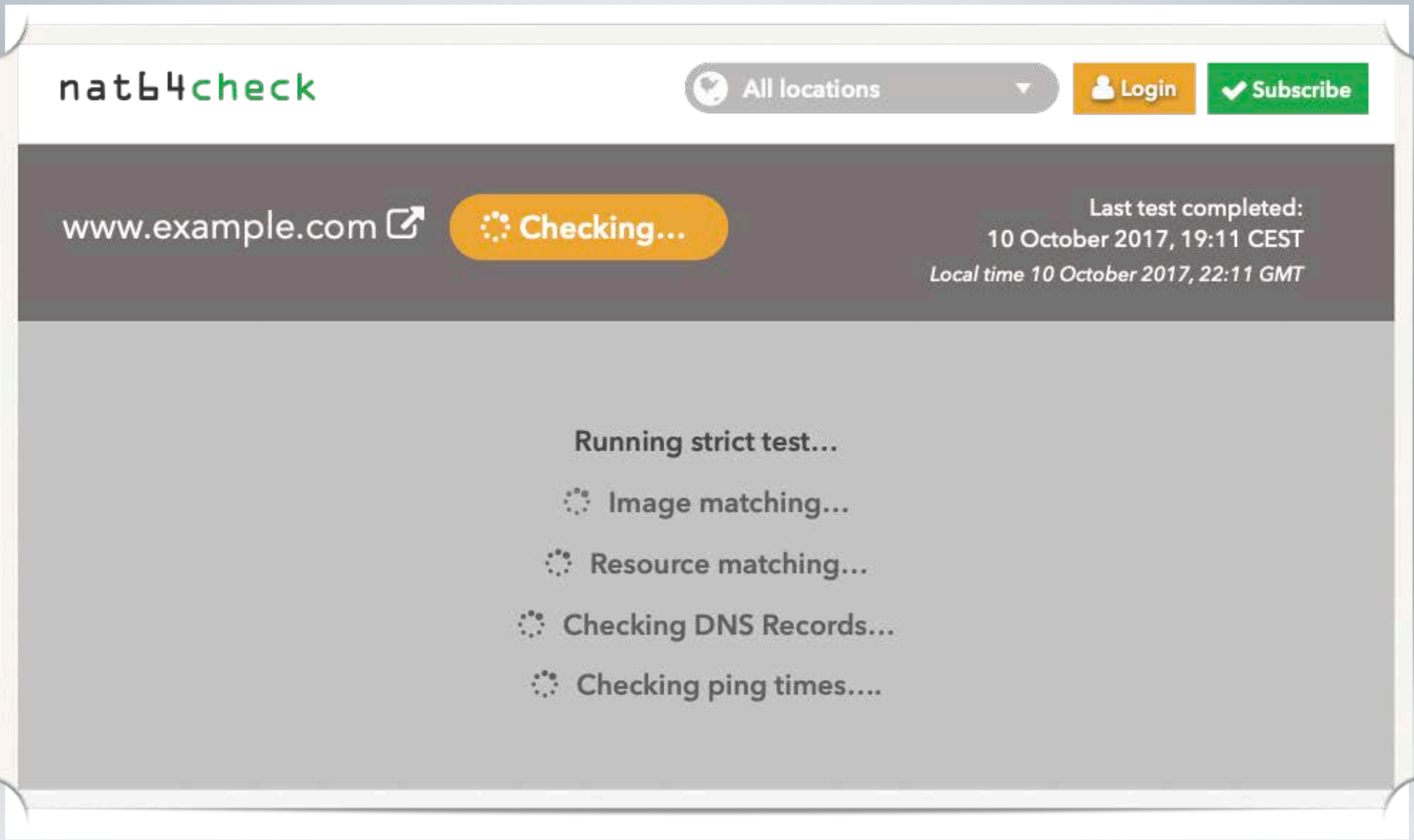
TECHNICAL DESIGN

(Illumination Theory)



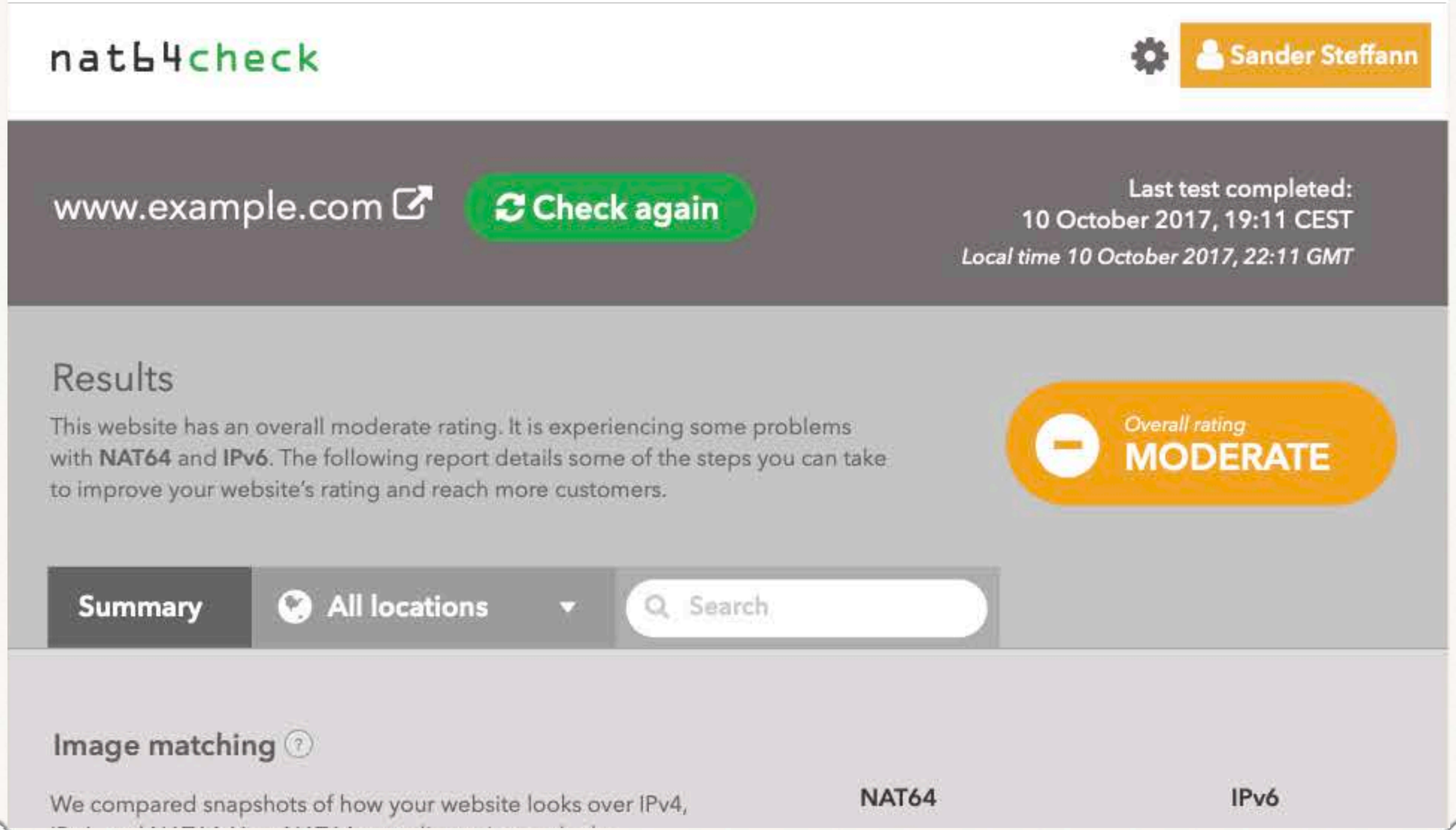
COUNTRY SELECTION

multiple choices allowed
(The Astonishing)



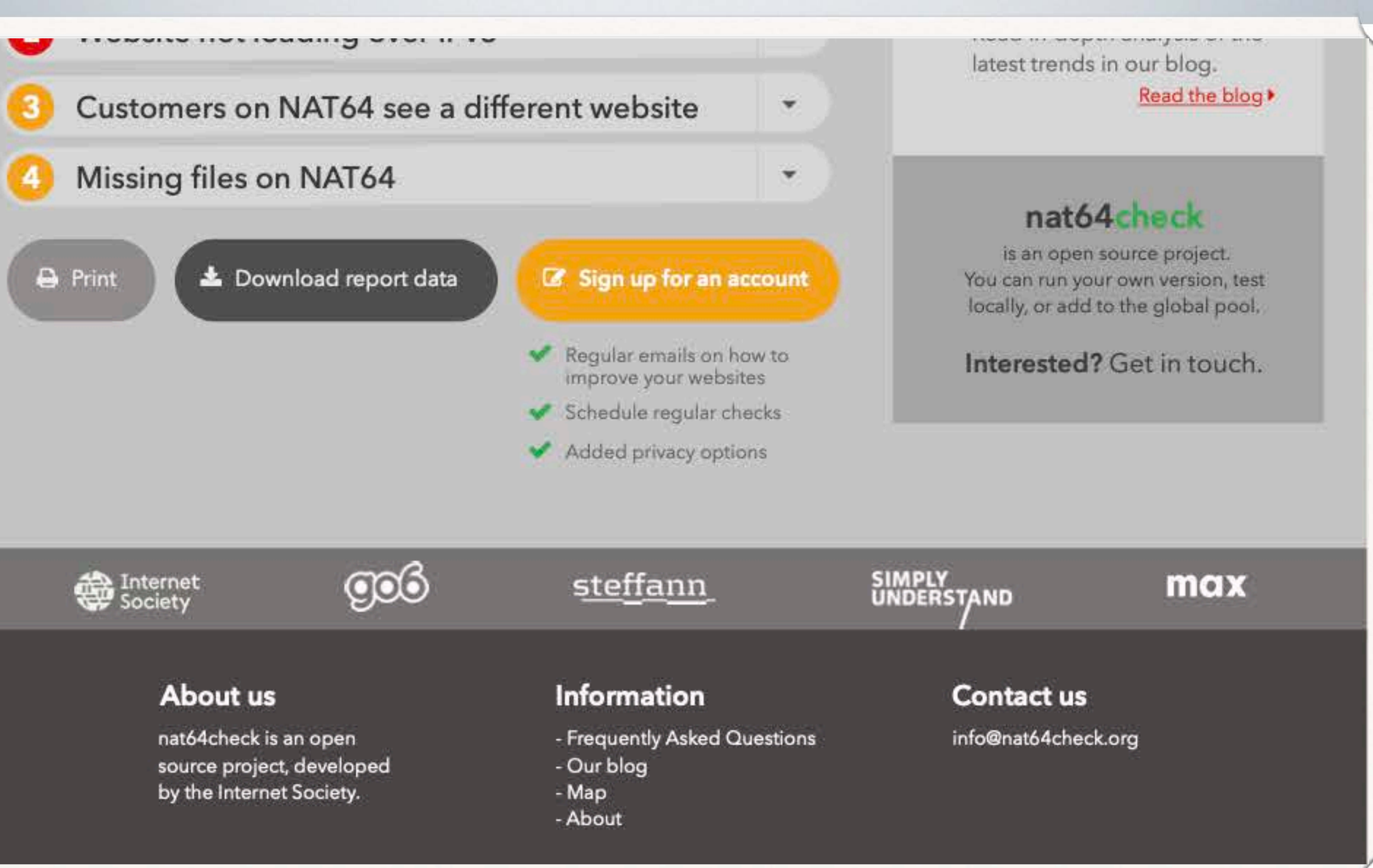
BETTER FEEDBACK

while testing
(Octavarium)



CLEAR SUMMARY

all test locations combined



CLEAR SUMMARY

all test locations combined

www.example.com

Check again

Last test completed:
10 October 2017, 19:11 CEST
Local time 10 October 2017, 22:11 GMT

Results

This website has an overall moderate rating. It is experiencing some problems with **NAT64** and **IPv6**. The following report details some of the steps you can take to improve your website's rating and reach more customers.

Overall rating
POOR

Summary | steffann.nl | Search

Image match

IPv4

NAT64 42%

IPv6

DETAILED VIEW

shown per test location
(Looking glass)

1 Broadcasting false IPv6 addresses

2 Website not loading over IPv6

3 Customers on NAT64 see a different website

4 Missing files on NAT64

Print | Download report data | Sign up for an account

- Regular emails on how to improve your websites
- Schedule regular checks
- Added privacy options

Frequently Asked Questions

- How does your country fare? Read in-depth analysis of the latest trends in our blog.

Read the blog

nat64check

is an open source project.
You can run your own version, test locally, or add to the global pool.

Interested? Get in touch.

Internet Society | go6 | steffann | SIMPLY UNDERSTAND | max

About us

nat64check is an open

Information

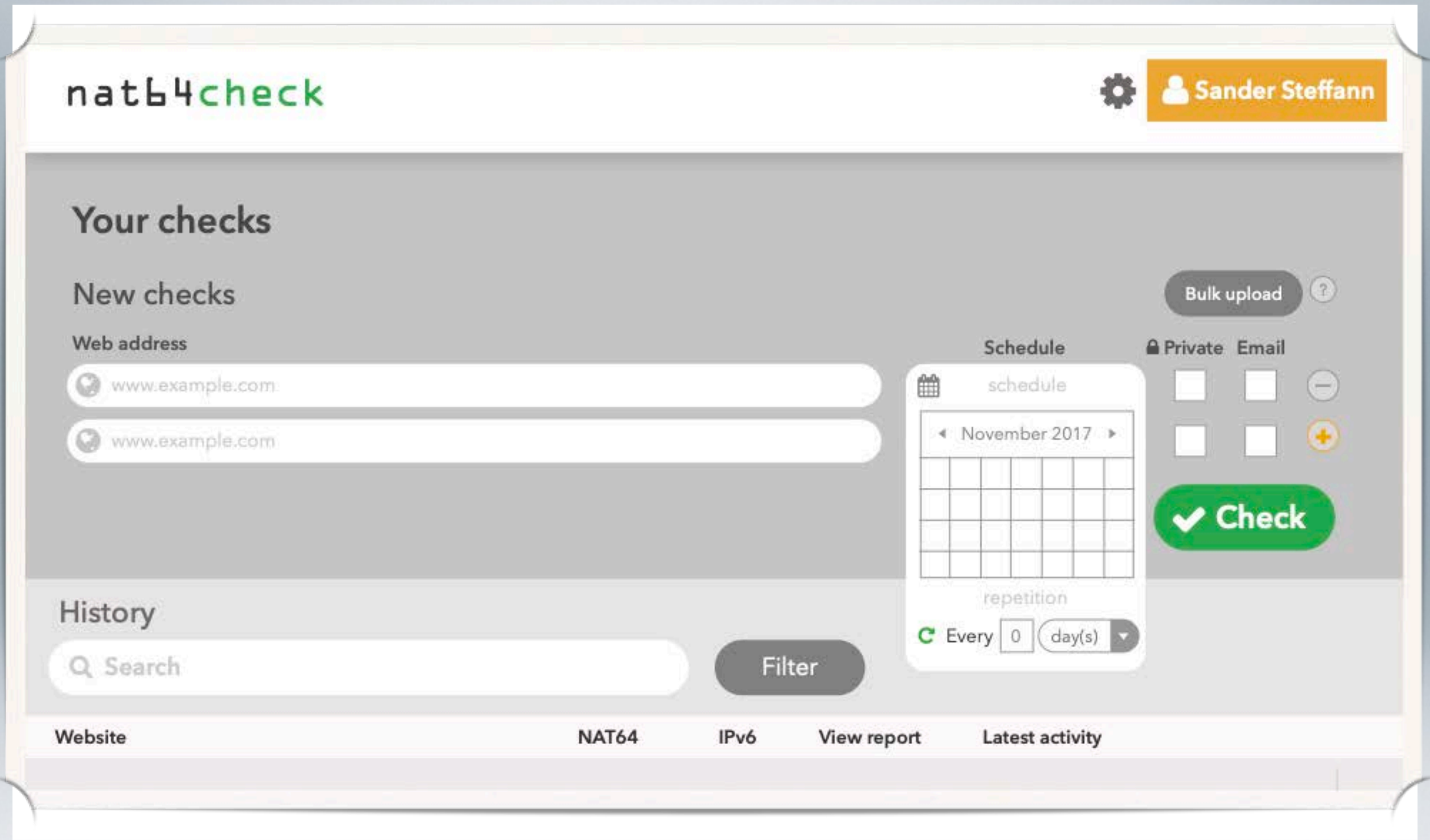
- Frequently Asked Questions

Contact us

info@nat64check.org

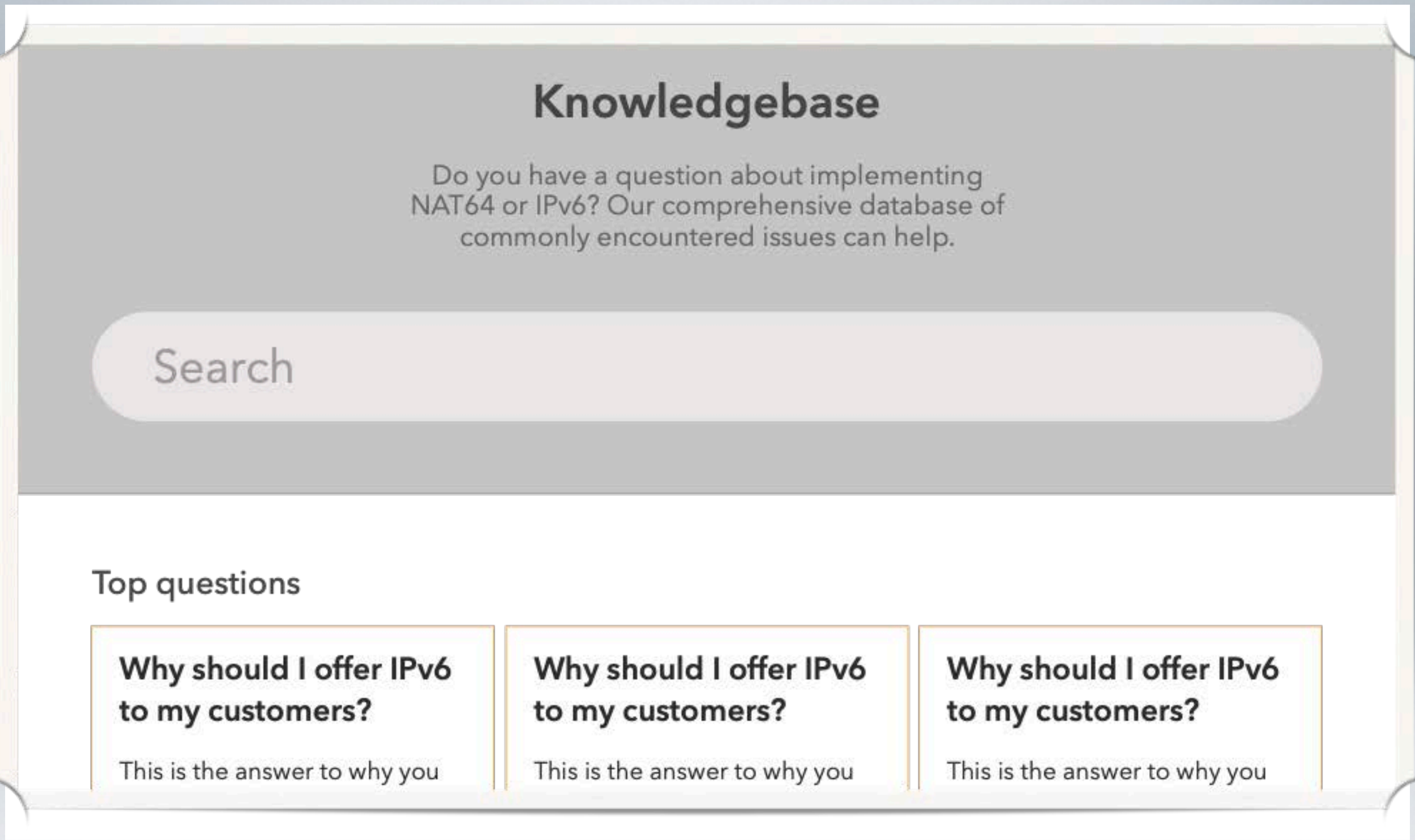
DETAILED VIEW

shown per test location
(Looking glass)



RECURRING TESTS

get alerts when things break



BACKGROUND INFORMATION

learn more about IPv6, NAT64 and the modern internet
(Train of Thought)

Credits / blame

(Images and Words)

- Project chaser:
  - Kevin Meynell
- Technical design:
  - Jan Žorž
  - Sander Steffann
- Marvin:
  - Prototype:
    Musa Stephen Honlue
  - Final implementation:
    Sander Steffann
- Trillian:
  - Implementation:
    Sander Steffann
- Zaphod:
  - Design:
    Corinne Pritchard
  - Front-end implementation:
    Internetbureau Max
  - Back-end implementation:
    Sander Steffann

Version 2 is now out!

(Learning To Live)

We need VM's around the world!
...as many as possible, so the platform grows.

https://www.nat64check.org/

Many thanks to ISOC
for sponsoring the development

steffann

Internet Society

go6