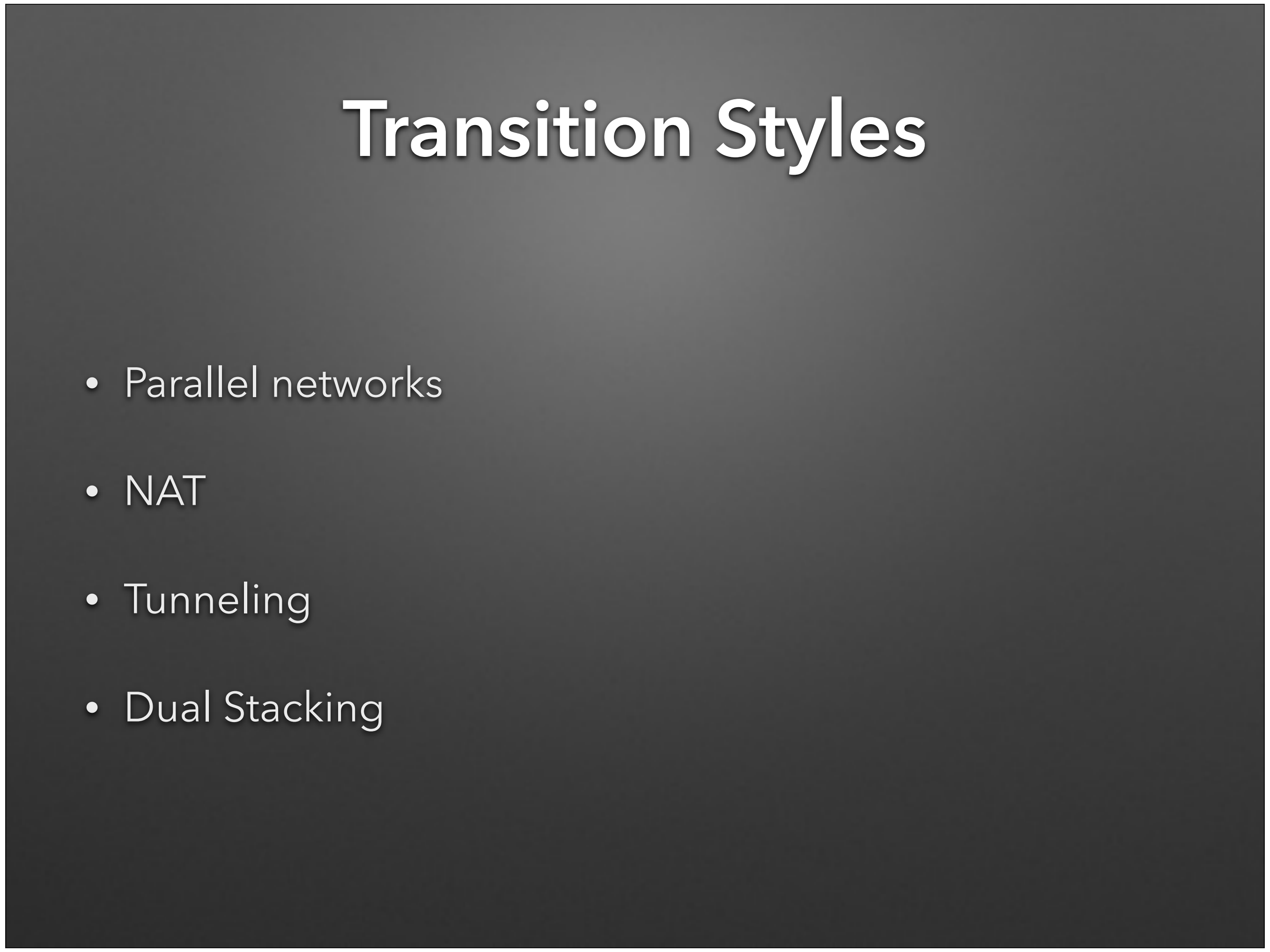
IPv6

Workshop!

Overview

- General Considerations re: using IPv6
- Systems
- Adding IPv6 to an IPv4 application
- Workshop (DNS, Web, Firewall, and Mail)

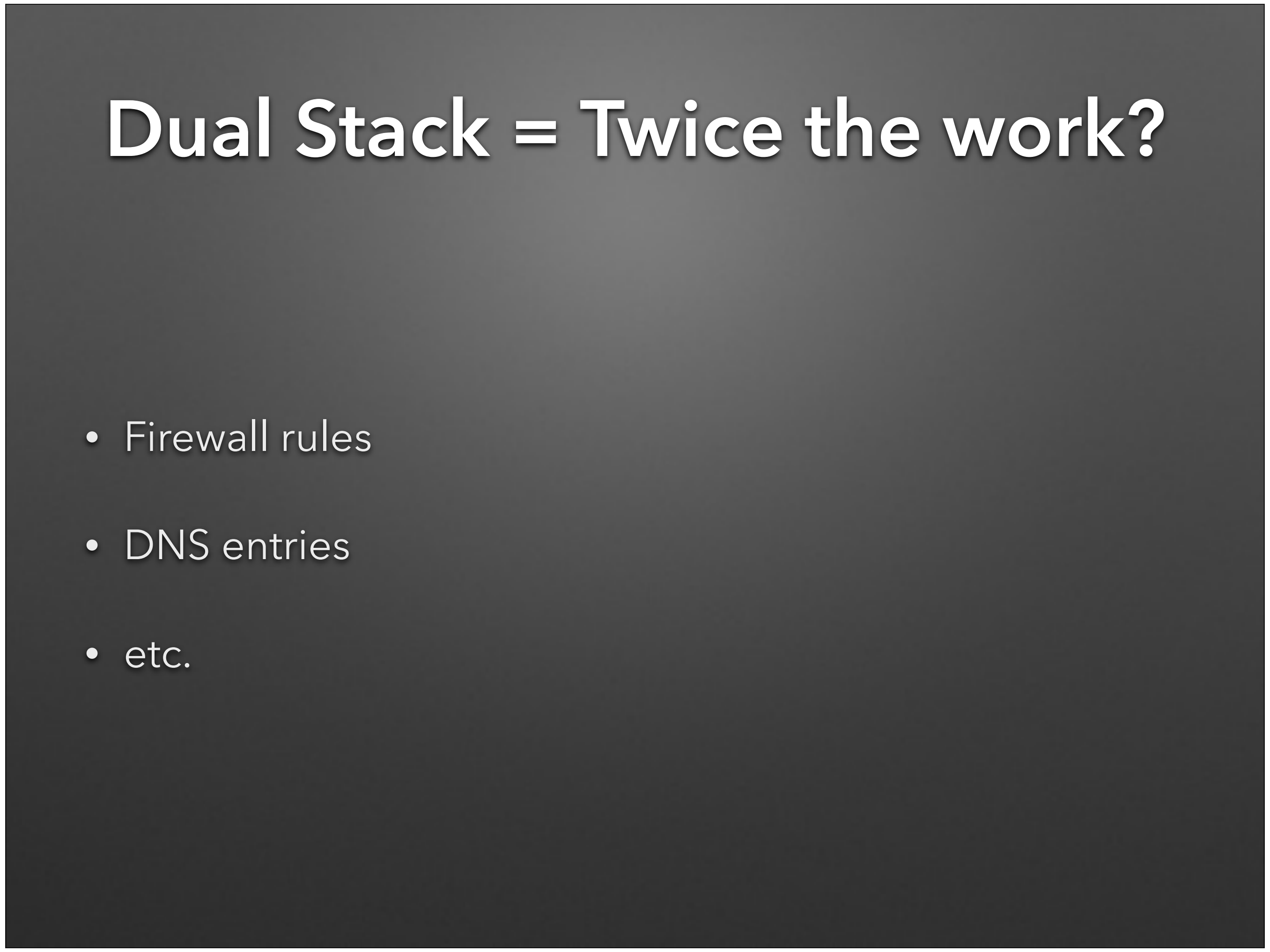


Transition Styles

- Parallel networks
- NAT
- Tunneling
- Dual Stacking

I'll be focusing on Dual Stacking - as that's what we've done at Cybera. However nearly all the concepts here will apply to whichever method you choose to use to provide IPv6 connectivity.

Parallel Networks - separate complete networks.
NAT - using address translation to bridge between the two types. (see NAT64 - allows IPv6 host to contact IPv4 hosts)
Tunneling - eg. 6in4 (eg. HE's), 4in6, 6to4.

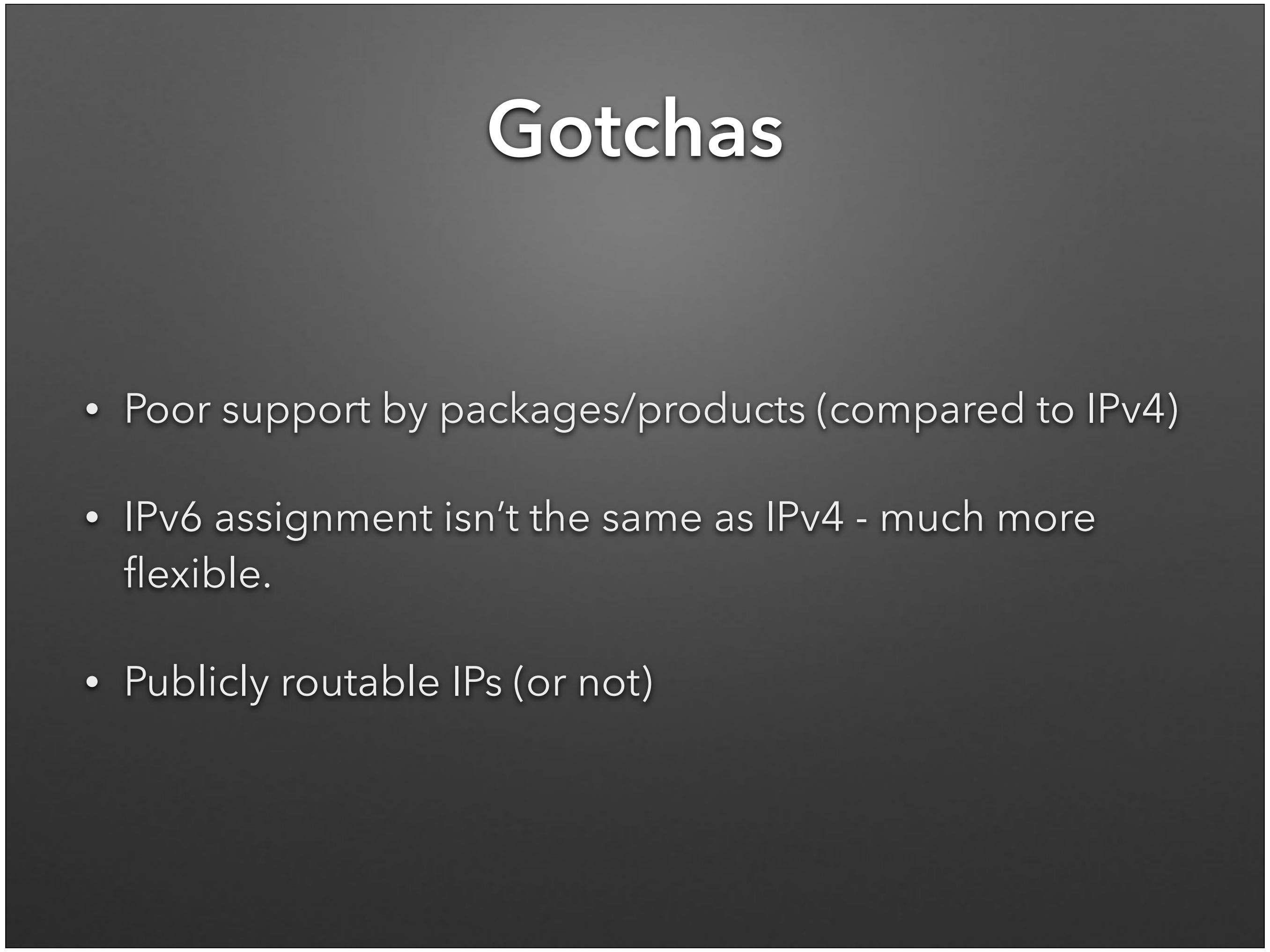


Dual Stack = Twice the work?

- Firewall rules
- DNS entries
- etc.

Uh. Yup. But it’s not that bad.

Ultimately the point is to make sure you’re running the same service on both IP versions. There are lots of variables that crop up from how the OS, app, or even network equipment times out on a connection that may or may not exist.



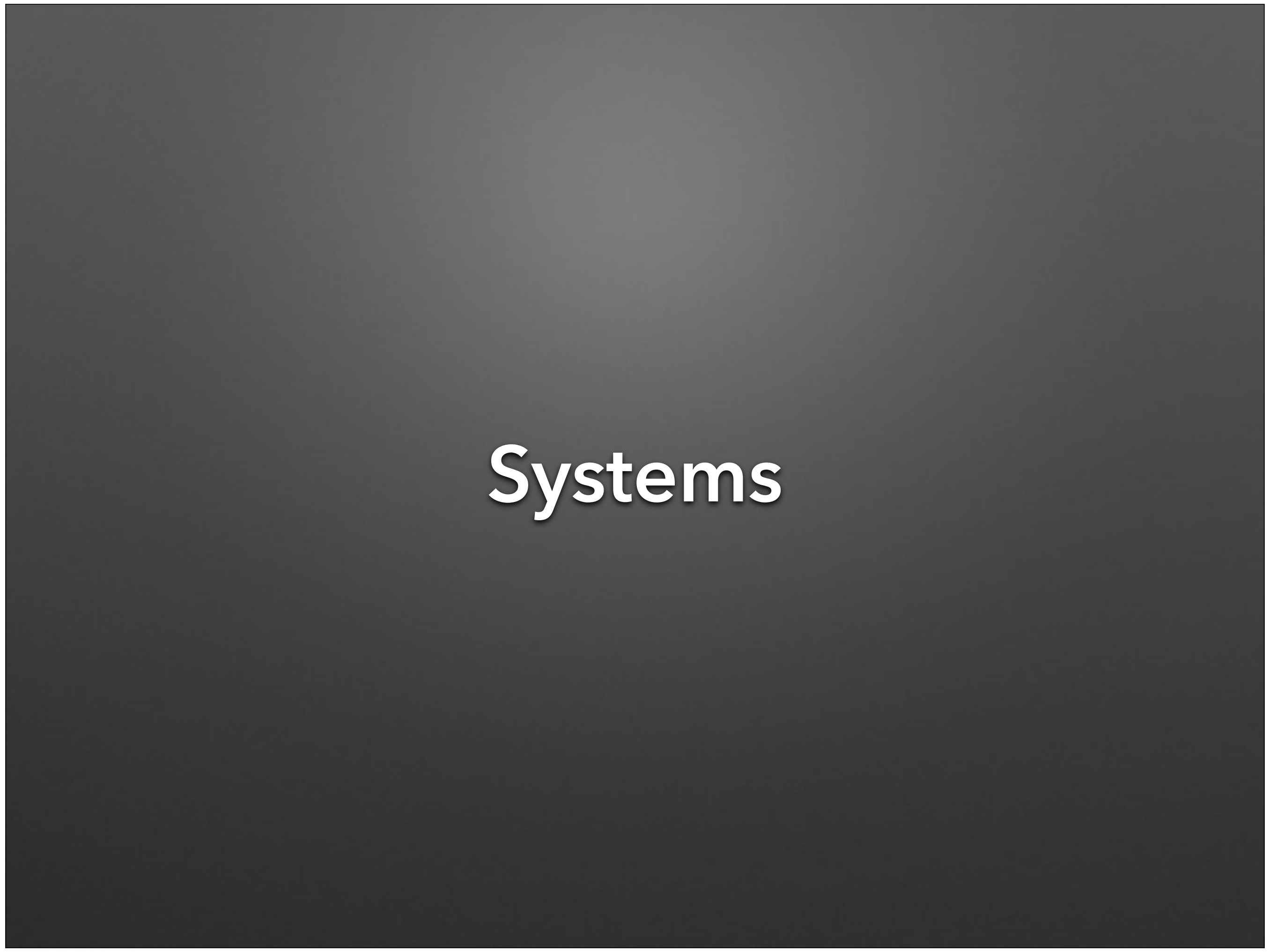
Gotchas

- Poor support by packages/products (compared to IPv4)
- IPv6 assignment isn’t the same as IPv4 - much more flexible.
- Publicly routable IPs (or not)

Lots say they do, but testing is not as thorough. Lots of exceptions.

IPv6 assignment - it’s very flexible. You don’t need to tie yourself to mimicking your IPv4 setup

If you’re used to having a NAT and not being able to get a publicly routable IP it can be very different.



Systems

Linux, OS X, Windows: https://wiki.bc.net/atl-conf/display/BCNETIPv6LAB/How+to+Configure+Your+Laptop+for+IPv6

Can have multiple addresses.

RFC 6555 - Happy Eyeballs

Happy Eyeballs is how to dual with dual stack hosts.

How to deal when the IPv6 path is broken and IPv6 is preferred.

Try IPv6 first - quick timeout and then use IPv4 from then on.

OS X / iOS

IPv6 first, then IPv4. It will prefer IPv6 whenever it can.

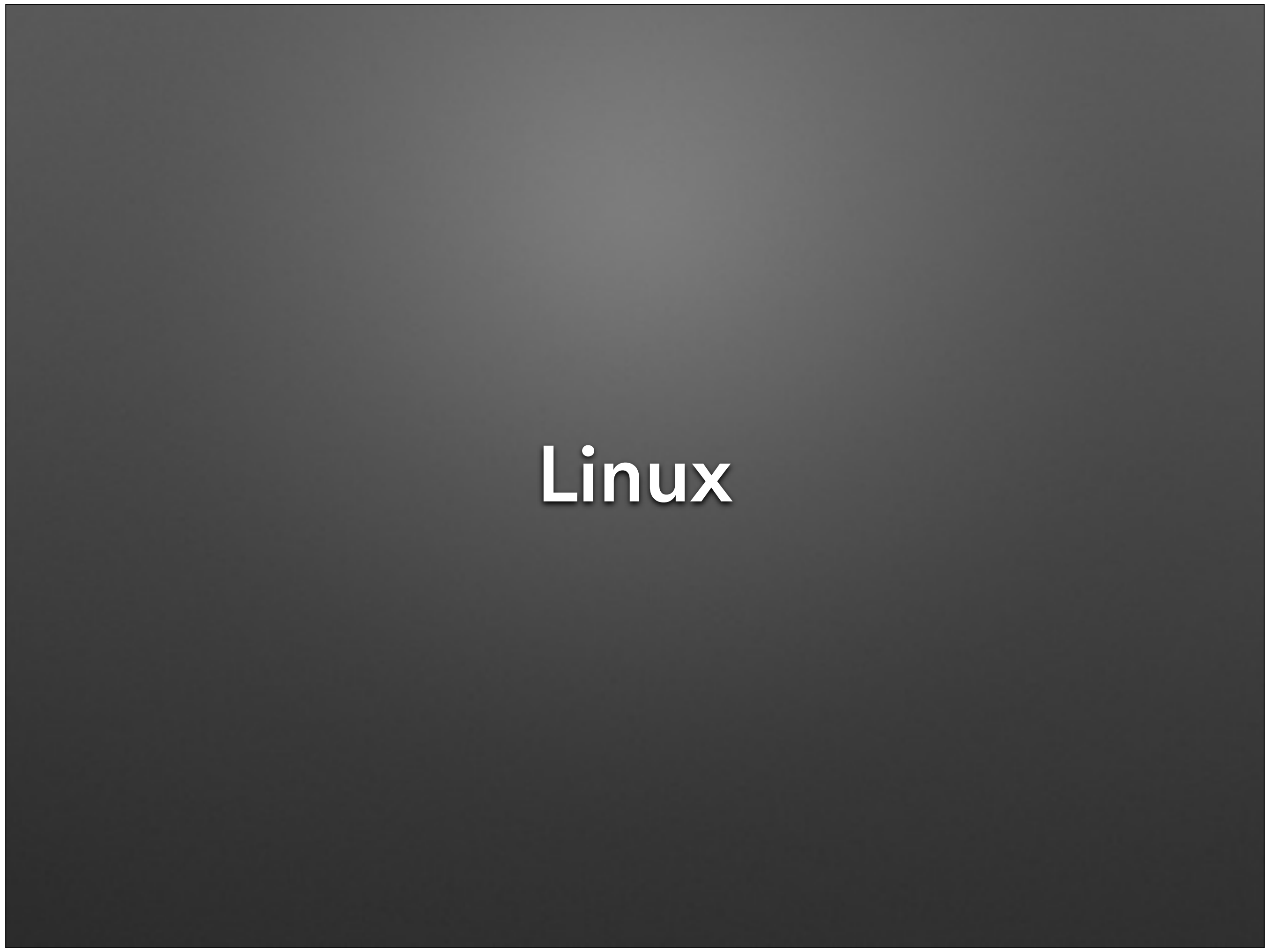
Supported IPv6 since 10.2

Supports Happy Eyeballs since 10.7

Windows

IPv6 first, then IPv4. It will prefer IPv6 whenever it can.

Windows 7 and prior doesn’t support Happy Eyeballs



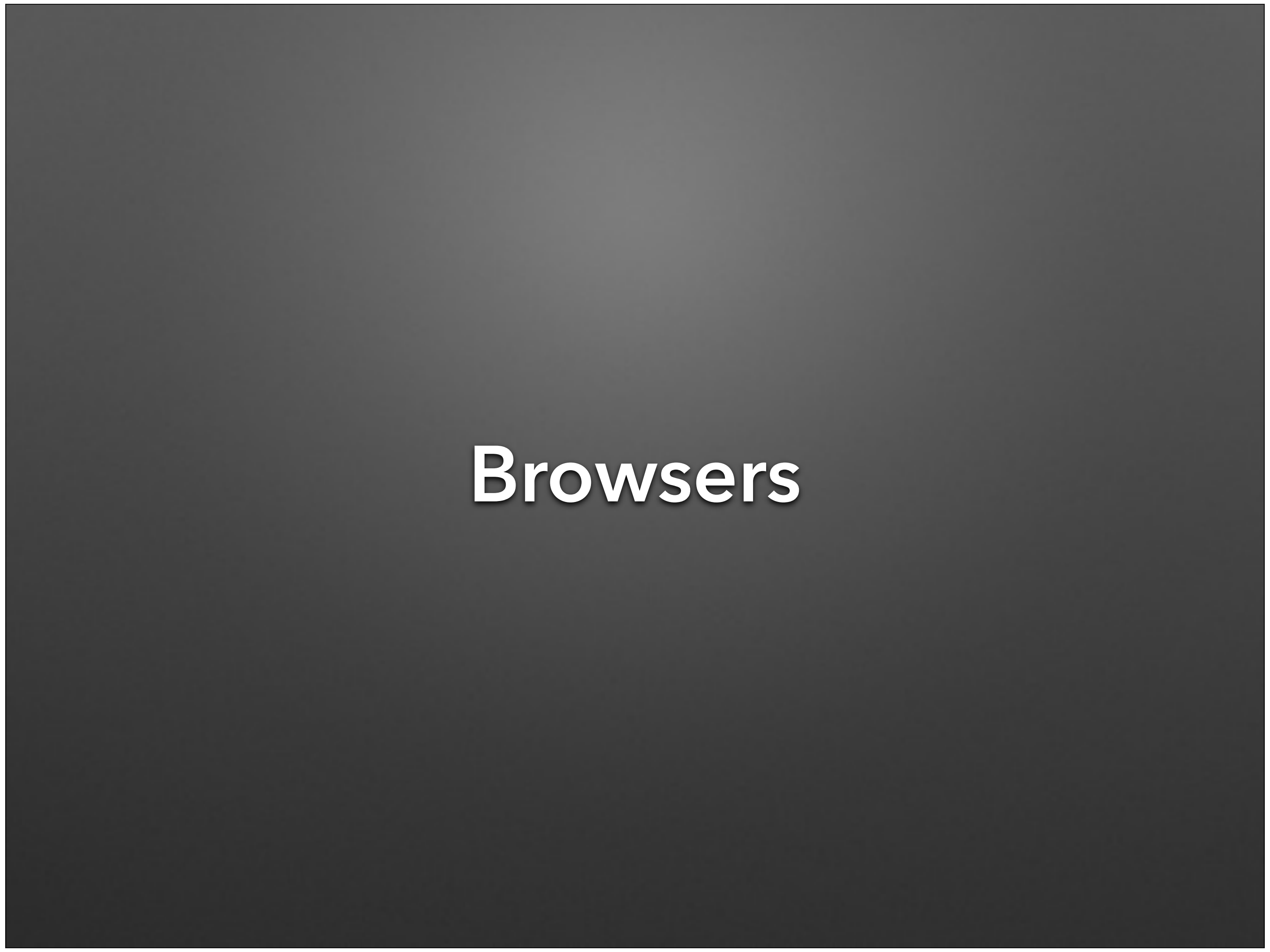
Linux

Has had an IPv6 ready kernel since ~2002/2003. (2.4+)

Prefers IPv6 if available.

Applications?

Applications are a whole different bowl of wax. You'll need to test. If it's written so the OS takes care of things you're pretty safe.



Browsers

Browsers are a whole different beast.

Happy Eyeballs - if one IPv6 connection > 300ms falls back to IPv4.

Mozilla Firefox

Happy Eyeballs - sticks with IPv6 if possible (tries all possible IPv6 addresses before falling back).

network.http.fast-fallback-to-IPv4 -> concurrent test

Internet Explorer

Nearly identical to Firefox: tries all IPv6 address/connections first and then falls back.

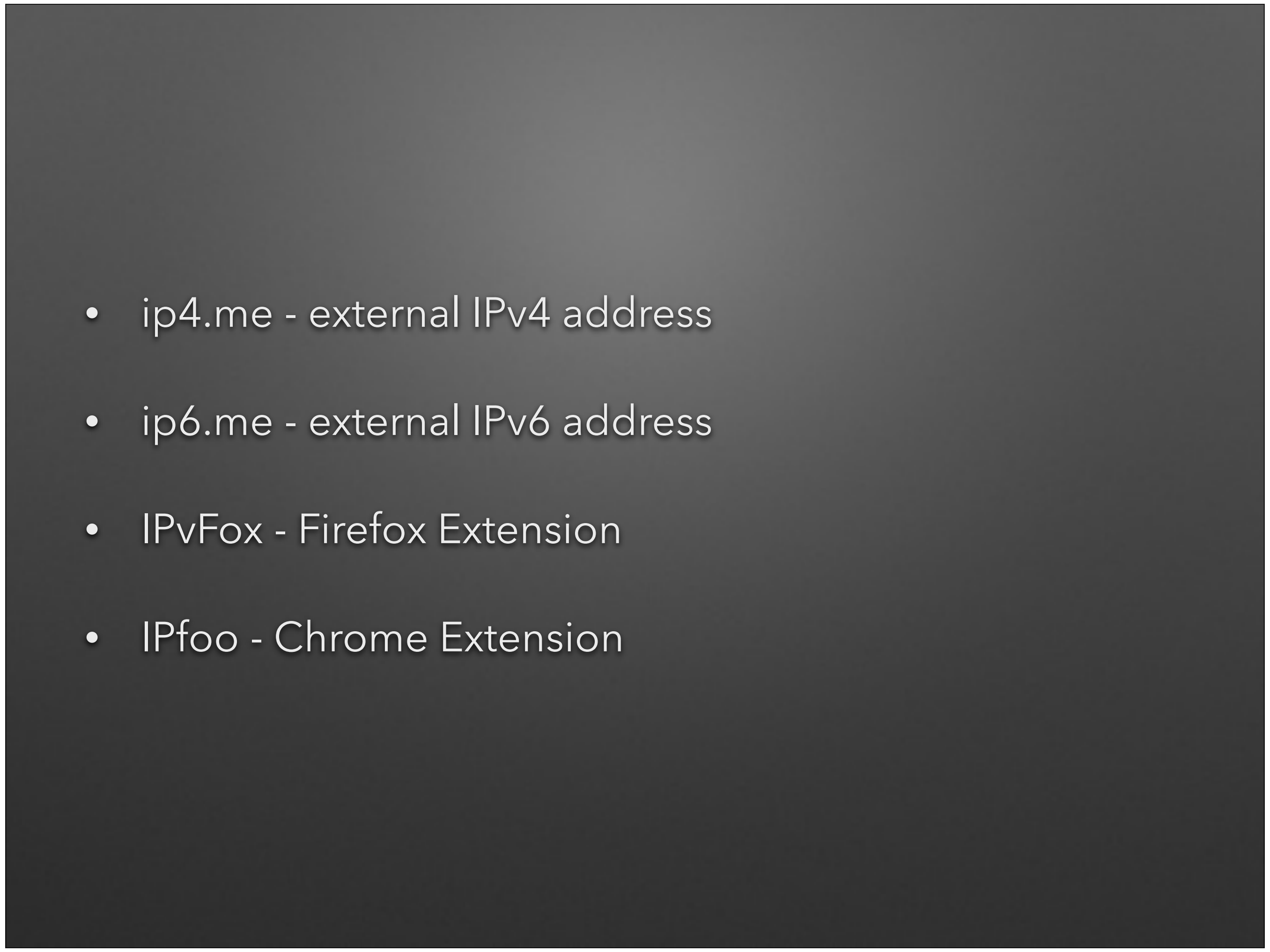
Safari (OS X or iOS)

Uses the OS' implementation of Happy Eyeballs

www.v6address.com

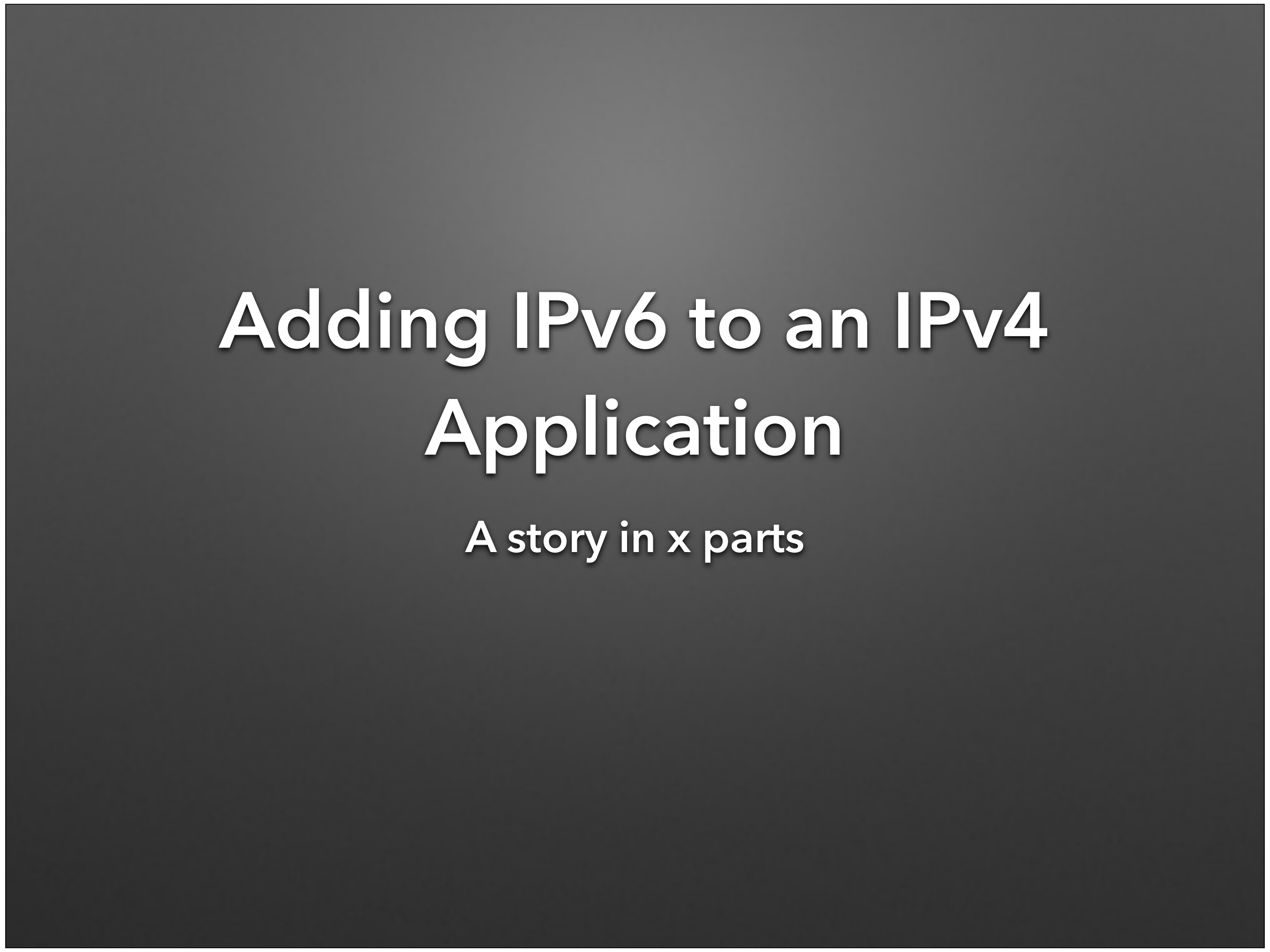
ip6.me

Curious to know? Try one of the above sites.



- ip4.me - external IPv4 address
- ip6.me - external IPv6 address
- IPvFox - Firefox Extension
- IPfoo - Chrome Extension

Other tools to use to debug/find out what’s up.



Adding IPv6 to an IPv4 Application

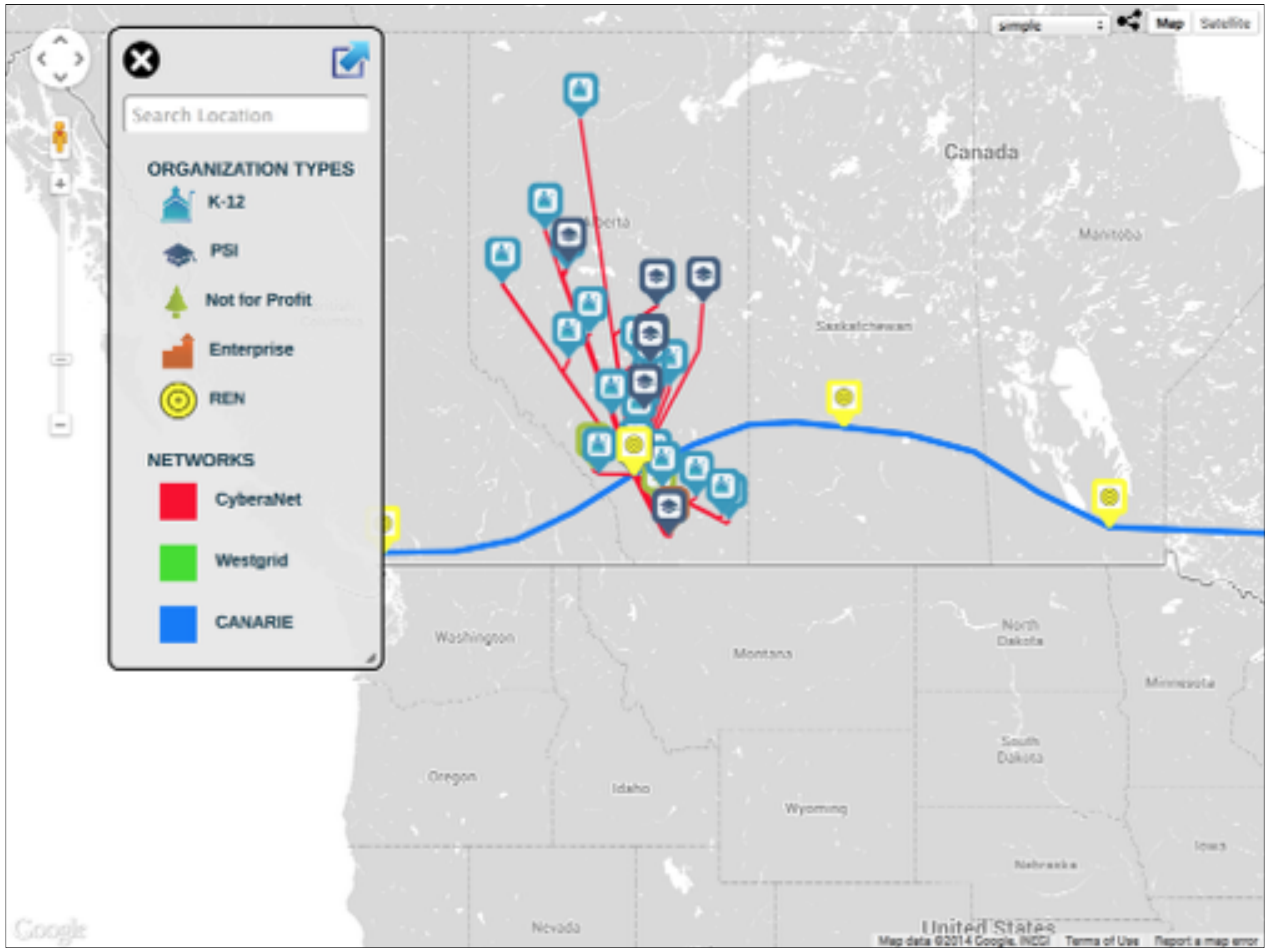
A story in x parts

Last before we get our hands dirty with the workshop is how we took an IPv4 application and made it happy with IPv6 and the steps necessary. Everything I did to do this helped create the workshop.

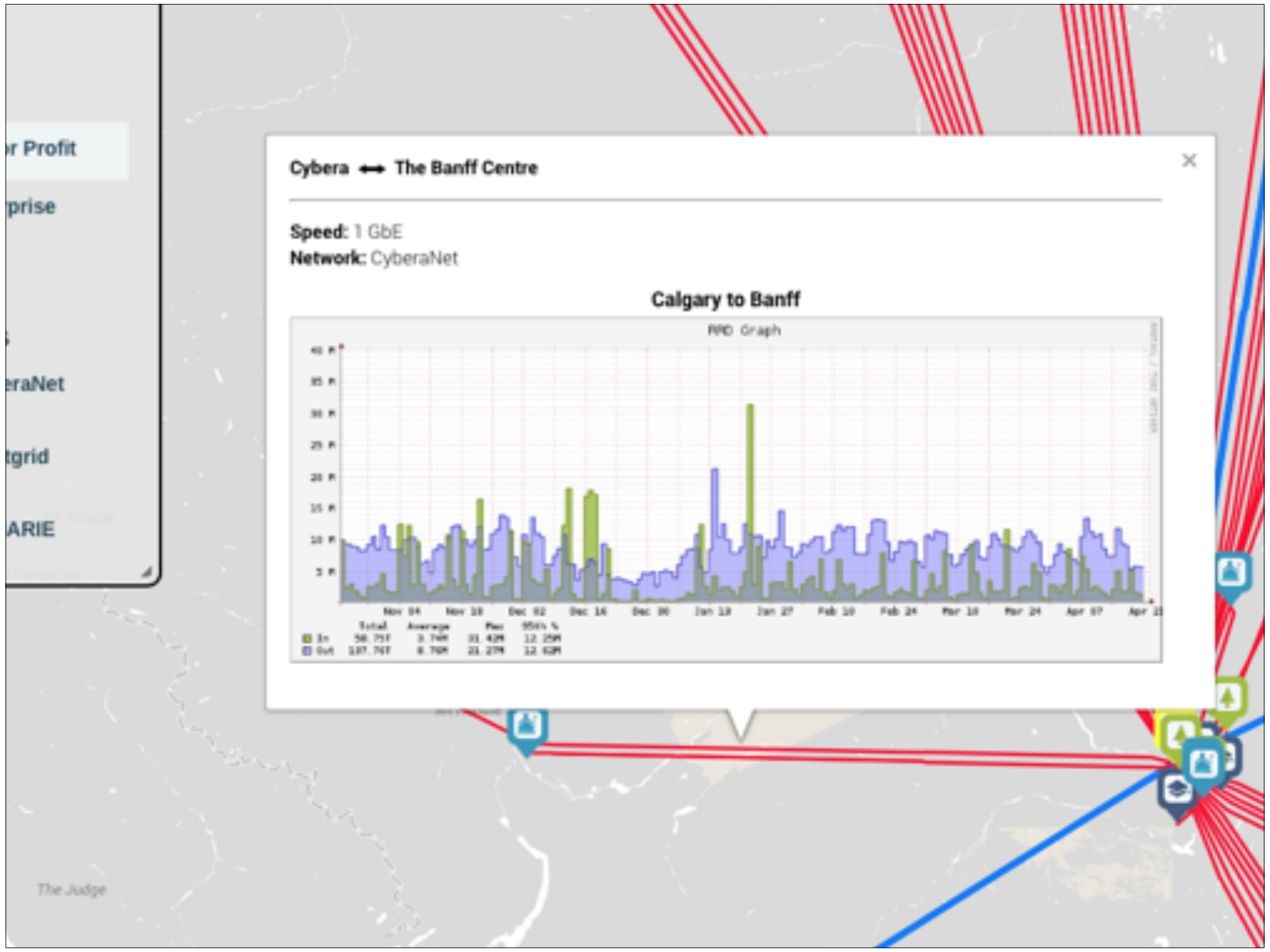
Network Mapping Tool

http://netmap.cybera.ca

Other much smarter people than I made an app and handed me the app to deploy in time for BCNET.



Other much smarter people than I made an app and handed me the app to deploy in time for BCNET. The app is designed to show the connections we provide…



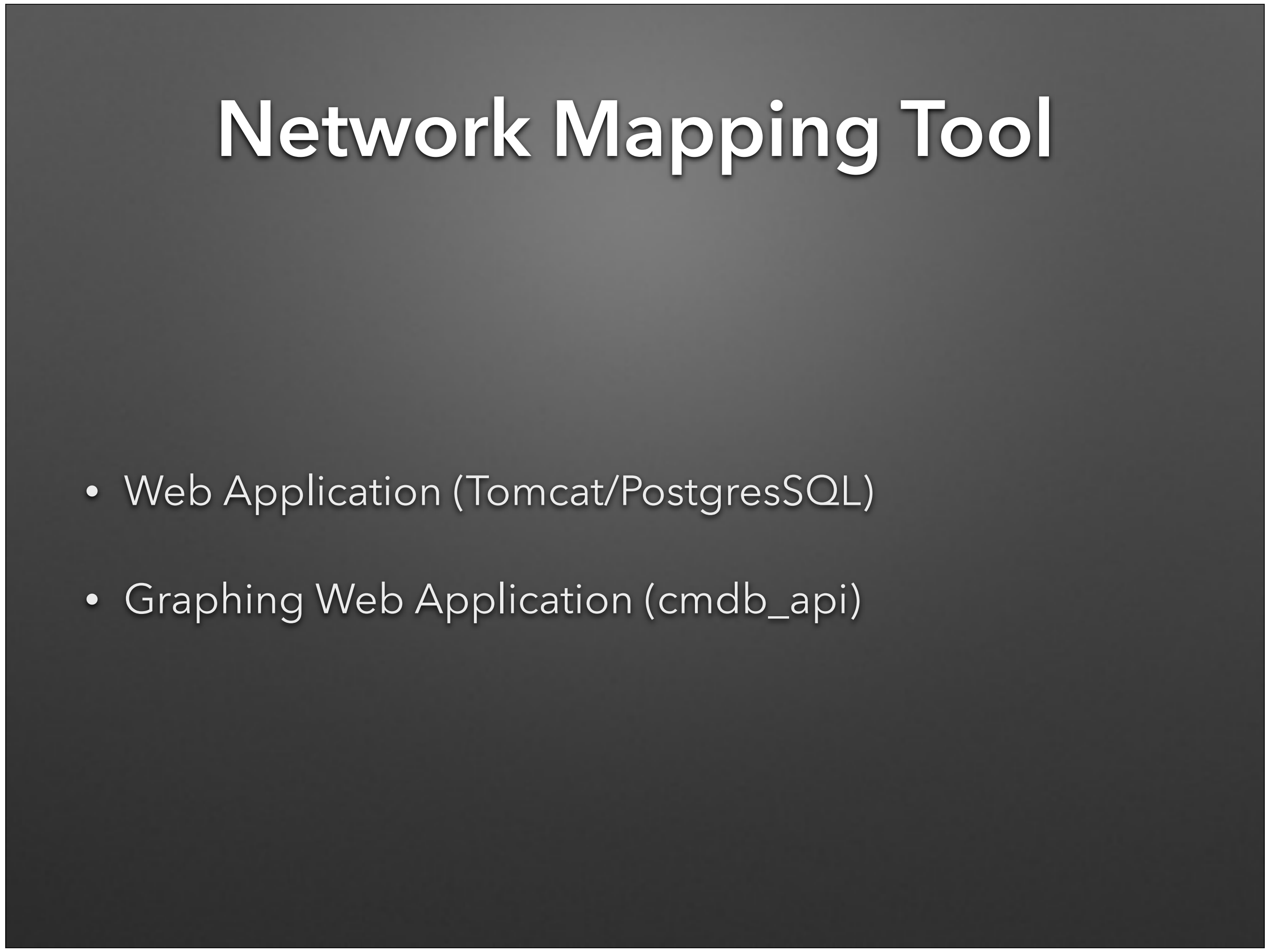
Cybera ⟷ The Banff Centre

Speed: 1 GbE
Network: CyberaNet

Calgary to Banff

	Total	Average	Max	95th %
🟩 In	58.75T	3.74M	31.42M	12.25M
🟦 Out	137.76T	8.76M	21.27M	12.62M

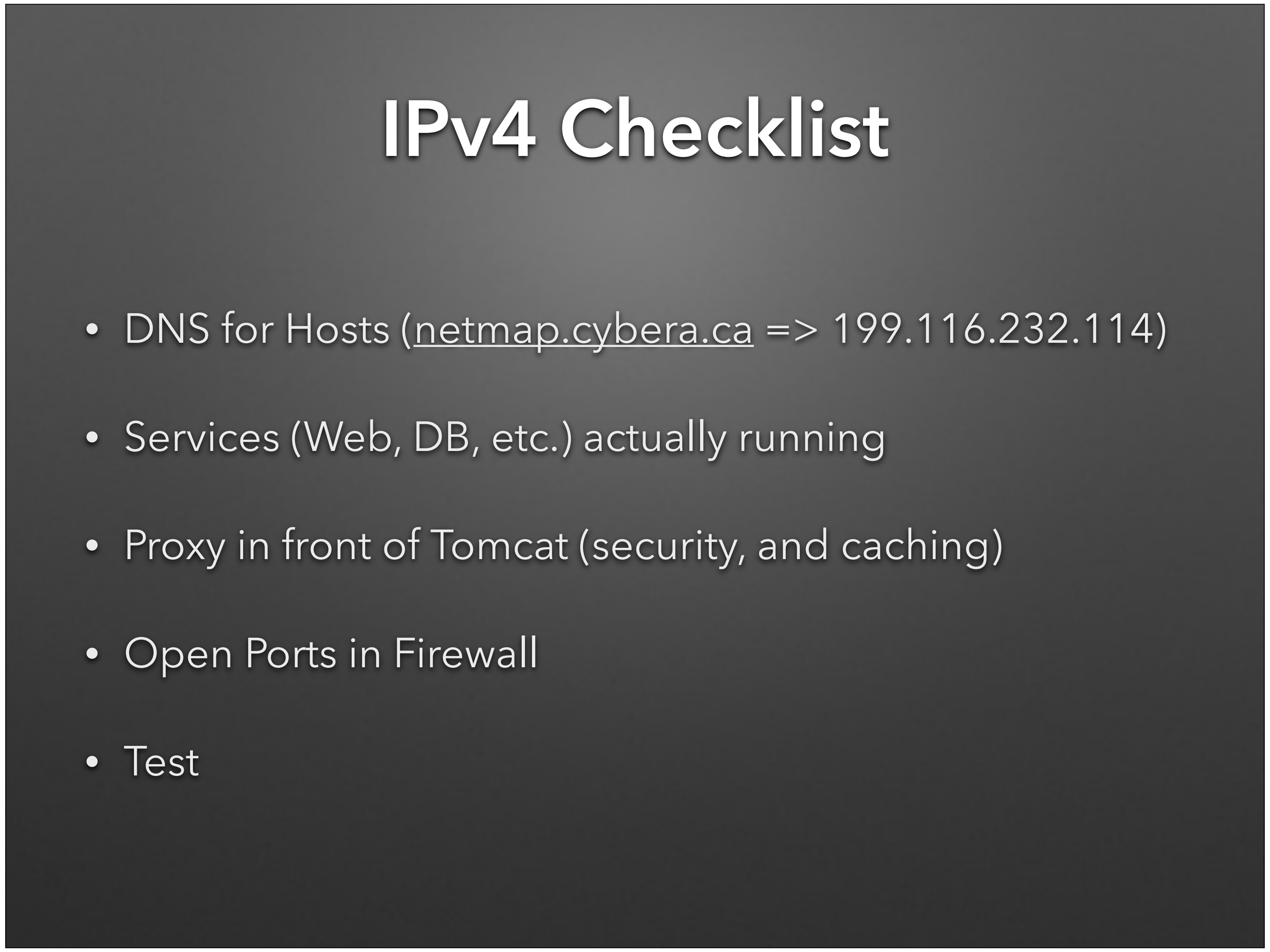
and if you click on a connection - a graph of bandwidth use.



Network Mapping Tool

- Web Application (Tomcat/PostgresSQL)
- Graphing Web Application (cmdb_api)

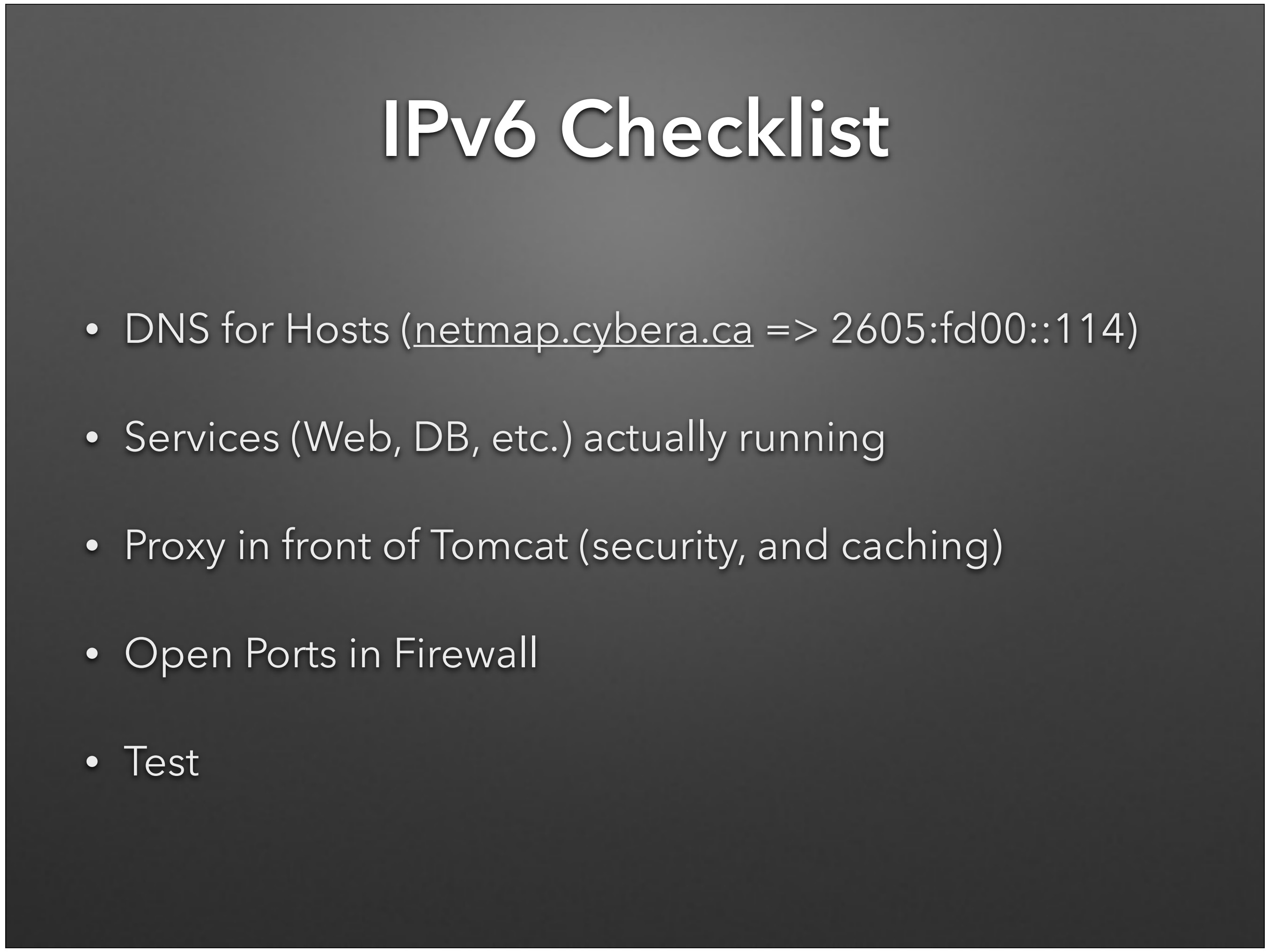
So it’s made of two parts - what’s next?



IPv4 Checklist

- DNS for Hosts (netmap.cybera.ca => 199.116.232.114)
- Services (Web, DB, etc.) actually running
- Proxy in front of Tomcat (security, and caching)
- Open Ports in Firewall
- Test

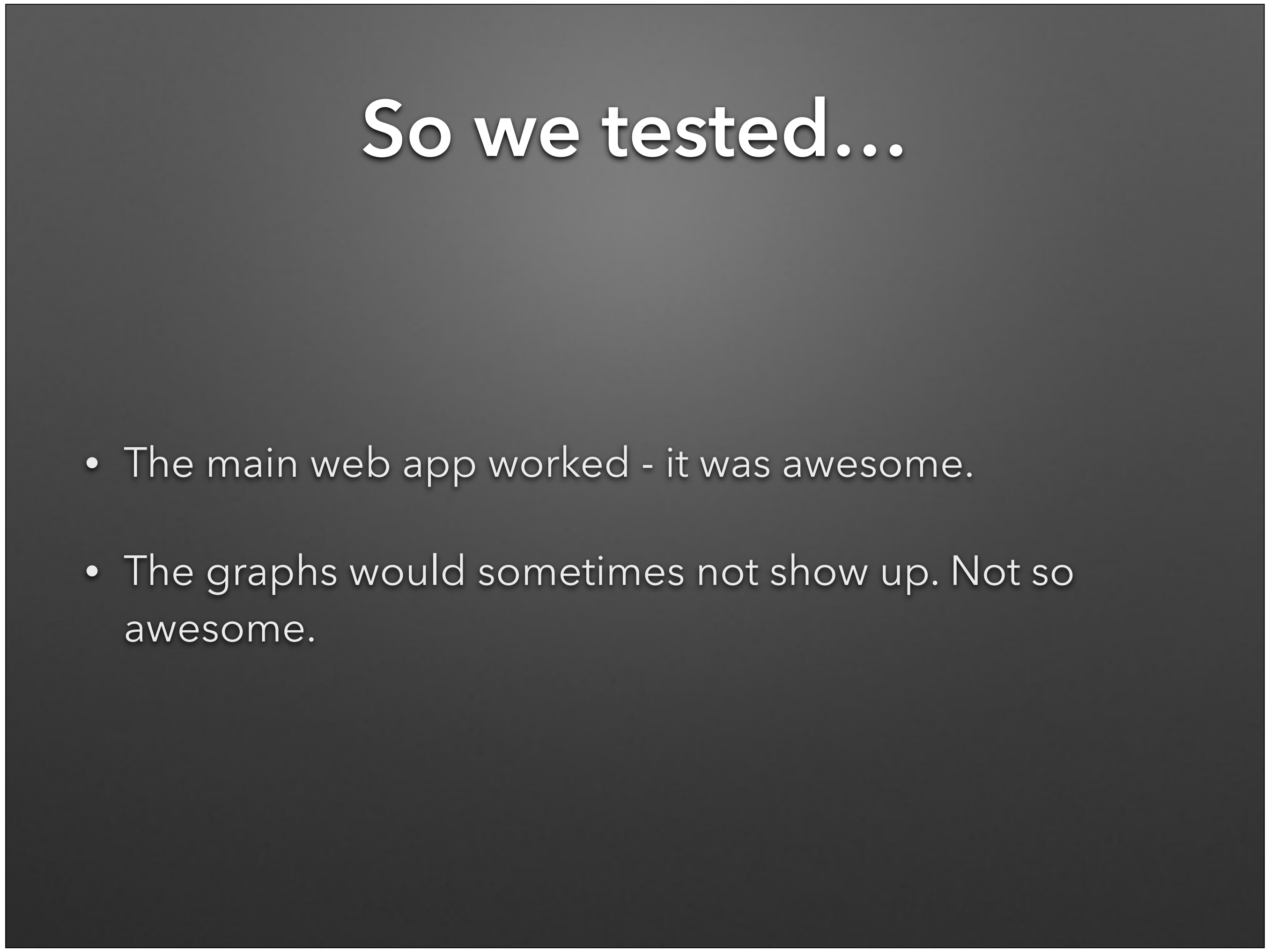
So us being awesome ops people we do what we normally do - we deploy it, test it, and make sure it stays running.



IPv6 Checklist

- DNS for Hosts (netmap.cybera.ca => 2605:fd00::114)
- Services (Web, DB, etc.) actually running
- Proxy in front of Tomcat (security, and caching)
- Open Ports in Firewall
- Test

Let’s add IPv6 as our servers already have IPv6 enabled. An IP is just another IP right?

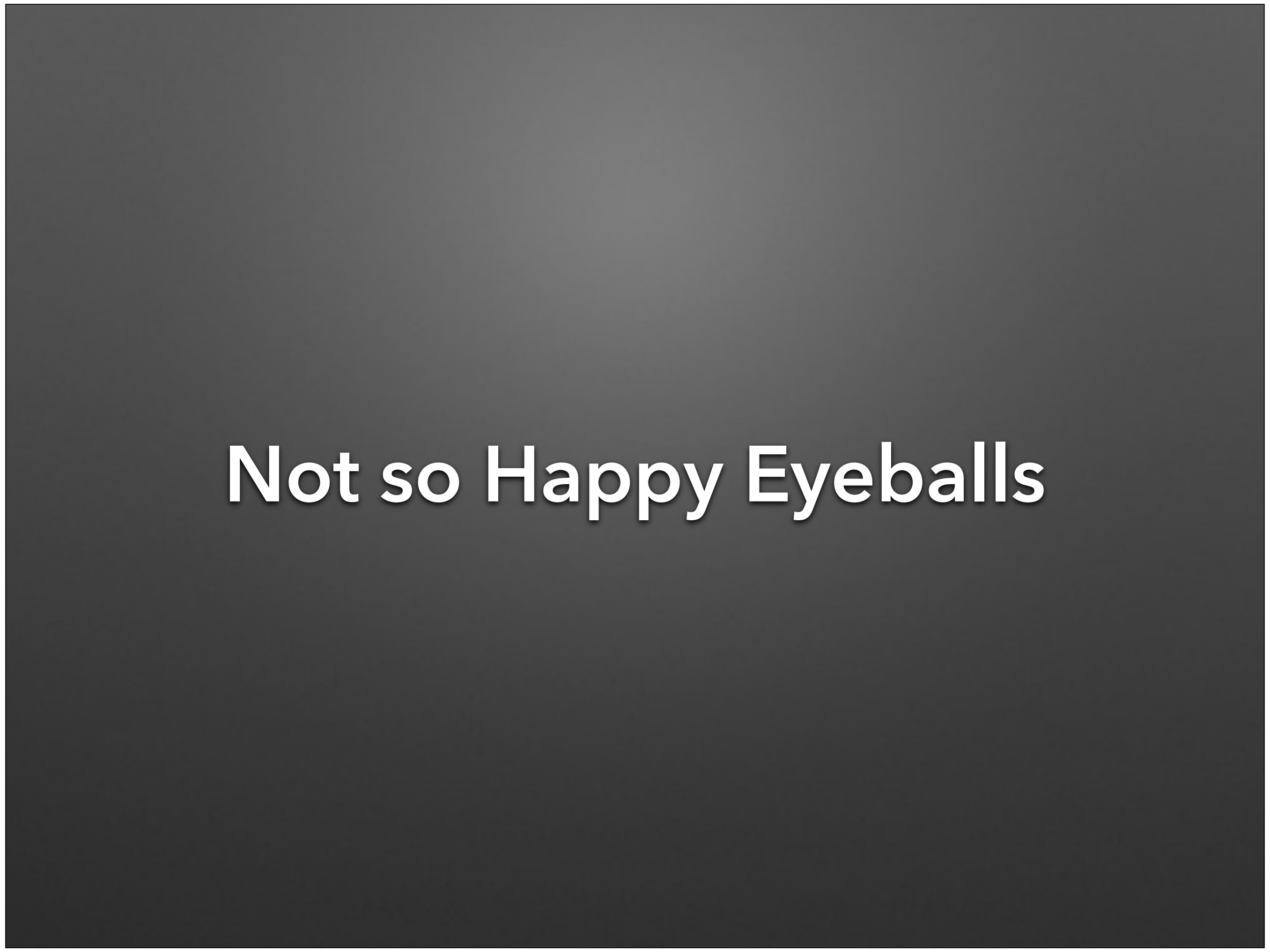


So we tested…

- The main web app worked - it was awesome.
- The graphs would sometimes not show up. Not so awesome.

So we tested:

- The main web app since we had a proxy in front of it just needed to be told to respond to IPv6 requests.
- CMDB_API would break. Why?



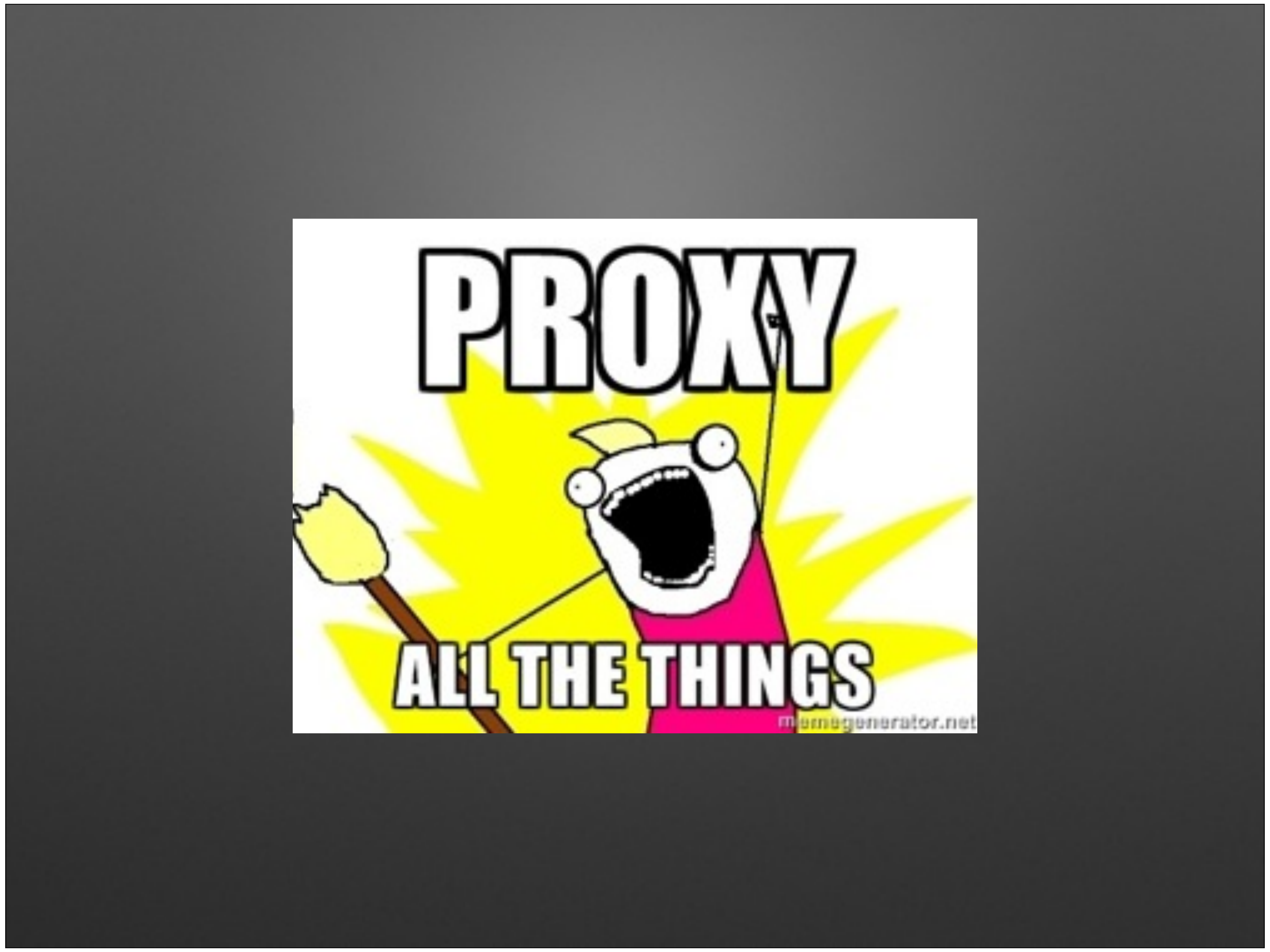
Not so Happy Eyeballs

This is what can happen when the IPv6 side of the equation is improperly set up:

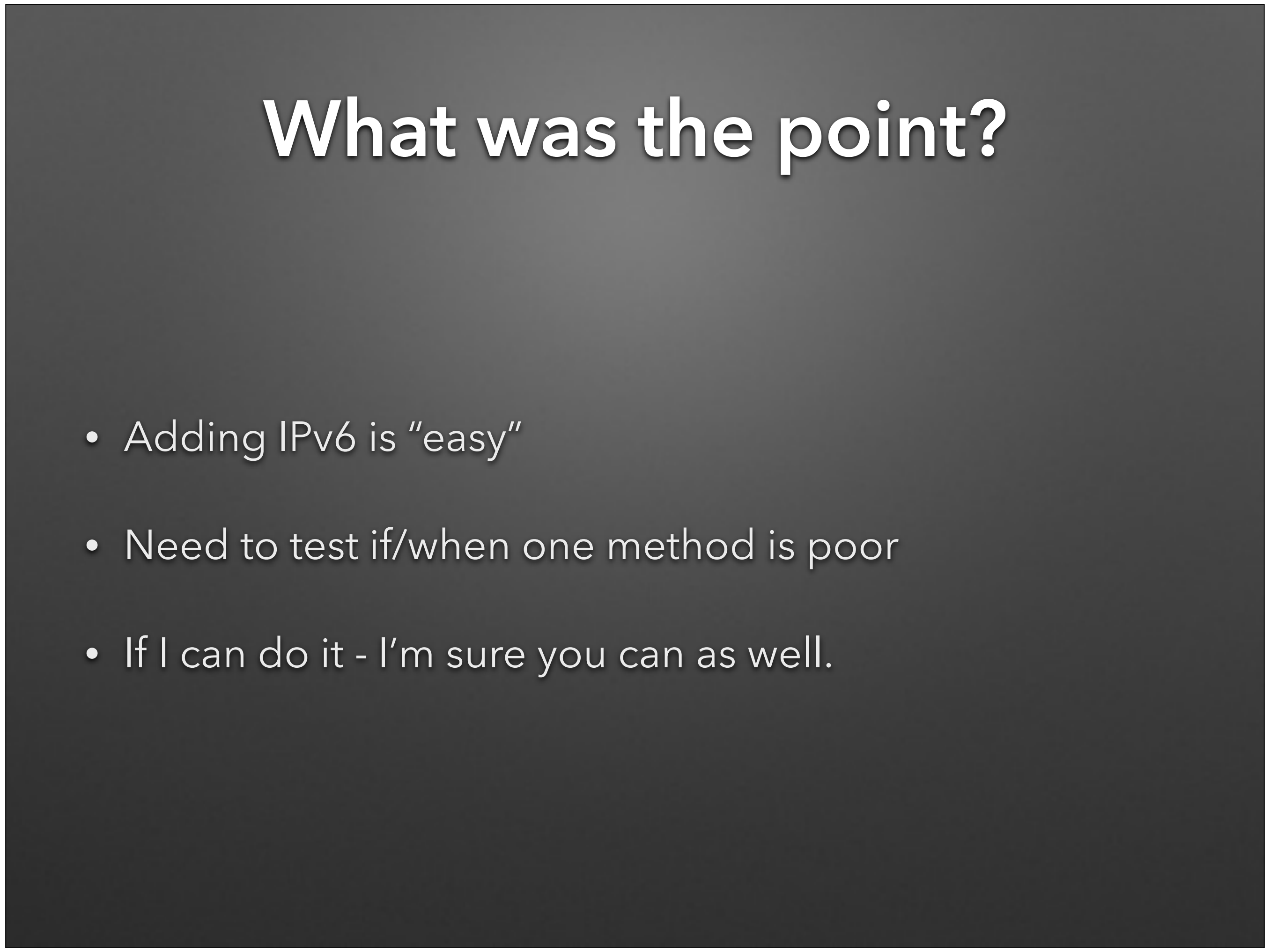
- Graphing portion would respond on IPv6 sometimes (since that server also hosts other items) but wasn’t actually running correctly.
- Connecting to netmap.cybera.ca via 4 or 6 did not guarantee what we connected on to the graph server.

Only 4 or only 6?

Digging further into the graphing application - it turns out it was configured to only listen via IPv4 - and Sinatra can’t do both at the same time.



Throw our proxy into the mix - this time proxying IPv4 and IPv4 requests and caching them as necessary.



What was the point?

- Adding IPv6 is “easy”
- Need to test if/when one method is poor
- If I can do it - I’m sure you can as well.

What’s the point of that anecdote?

Workshop Time

http://github.com/cybera/ipv6-system

If there’s anything you take away from this workshop it’s not how to configure the services - it’s that they’re ready. Go. Start. Now.

That’s the end of the slides.