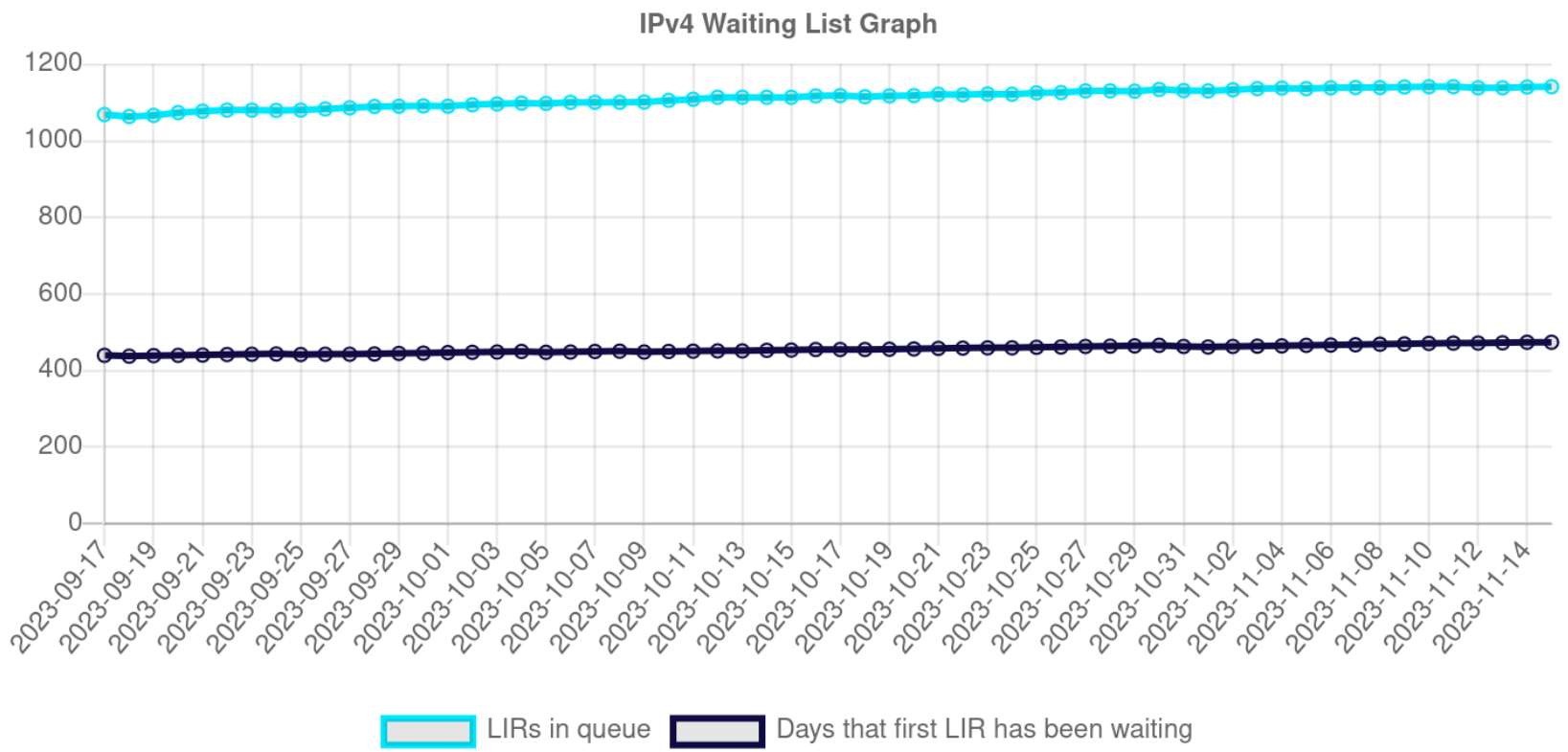
IPv6 Only: ~~no~~ customer demand

Pete Stevens
Mythic Beasts Ltd

IPv6 only – a recap

- VM get only public IPv6
- DNS resolvers set to DNS64 which generate 64:ff9b::/96
- NAT64 set up
- Free inbound proxy that proxies http/https/imaps/pop3s
- Port forwarding for ssh from the IPv4 enabled host
- By default no clat so 8.8.8.8 is unreachable
- All Mythic Beasts managed services will connect over IPv6

Why bother?


IPv4 Waiting List Graph
1200
1000
800
600
400
200
0
2023-09-17
2023-09-19
2023-09-21
2023-09-23
2023-09-25
2023-09-27
2023-09-29
2023-10-01
2023-10-03
2023-10-05
2023-10-07
2023-10-09
2023-10-11
2023-10-13
2023-10-15
2023-10-17
2023-10-19
2023-10-21
2023-10-23
2023-10-25
2023-10-27
2023-10-29
2023-10-31
2023-11-02
2023-11-04
2023-11-06
2023-11-08
2023-11-10
2023-11-12
2023-11-14
LIRs in queue
Days that first LIR has been waiting

No customer demand

- The customers know and understand IPv4 and **everybody knows they don't understand or want IPv6**
- We already run lots of standard managed application IPv6 only
- Now starting to do bigger customer written applications

Riskwise

- Web service for property companies that manages risks
- Tracks fire safety documents, compliance status, accident and incident management …
- As a risk management company they believe you should
  - Move very slowly and don’t break anything ever
- Second application to move in, we did an internal application first on the same IPv6 only stack

Traditional web stack

- MySQL database (~100TB)
- File storage (~10TB)
- Memcached
- PHP
- Apache
- HAProxy

Private cloud

- VM hosts run our standard cloud setup but hardware dedicated to a single customer
  - No per VM costs
    - Unless you need a chargeable IPv4 address
  - Allows a scheduled approach to VM breakout security issues, microcode updates and so on
  - Only the firewall VM has an external link, completely private virtual network for the others

Central Management Facilities

- Security updates
- Backups
- Performance graphing
- Monitoring
- All these are external to Riskwise using Mythic Beasts shared infrastructure and tooling

Management requires direct access

- Update process requires our update server to log into the customer server to update packages
- Backup process requires large data outflow to the offsite backup servers avoiding NAT64
- Graphing polls all the services directly
- Monitoring checks all the component services directly
- This is much easier if **every server has a globally unique identifier**

Restrictive firewall

- Firewall rules allow
  - the update server direct access to ssh
  - Graphing server direct access to munin
  - Monitoring server direct access to monitored services
  - Servers can send wireguard packets to each other

Public/Encrypted network

- We don’t run a ‘private’ network
  - All access to a protected resource goes over a wireguard tunnel with ULA
  - Every webserver has a 1:1 tunnel to the database/memcached and so on
    - Even if they’re VMs on the same physical host
  - This is a lot of config!
  - ansible to the rescue

Encrypted network

- Used to use SSL
  - setup/teardown too high latency
  - Application SSL support ismostly terrible
    - Common not to authenticate certs
    - Updating the cert not graceful
    - Client certificates? What are they?
- Wireguard separates this out

IPv6 single stack

- Almost all traffic behind the firewall is IPv6
  - We don’t trust the network at all, every internal connection is SSL or wireguard
  - Exception is v4 on the dual stack load balancer at the front of the site
- All outbound traffic starts IPv6
  - If the destination is IPv4 only it gets translated by haproxy/NAT64

External services

- DeployHQ for deploying applications updates with ssh
  - Full IPv6 support so open ssh in the firewall
  - No port forwarding!
  - Routing issue meant they dropped IPv6 mid deploy
    - ssh doesn’t implement happy eyeballs so this was fixed immediately

Bugs!

- Rate limiting on the login form was a varchar, rather than an IP type which truncated IPv6 addresses leading to overly broad lockout (trivial fix)
- Some queries broke MySQL GTID usage which kept breaking replication (not IPv6, application fix)
- Connections kept lingering causing the browser to run out and hang the application (http2 config error from apache/haproxy, not IPv6)

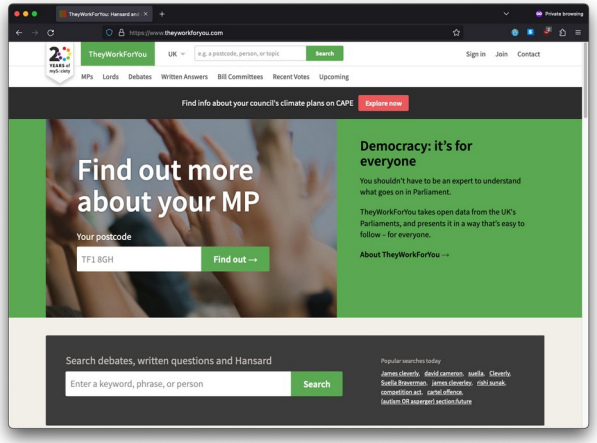
And now for something completely different

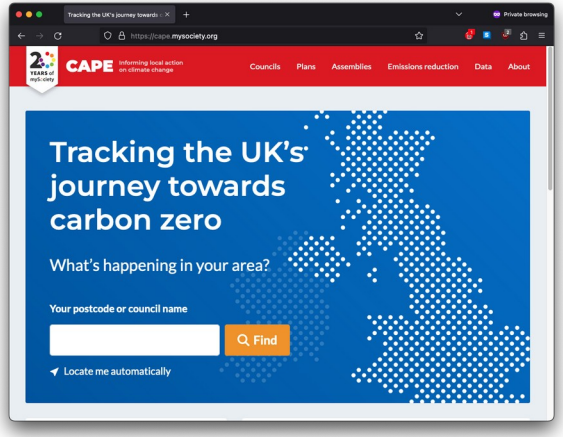
This is Sam.

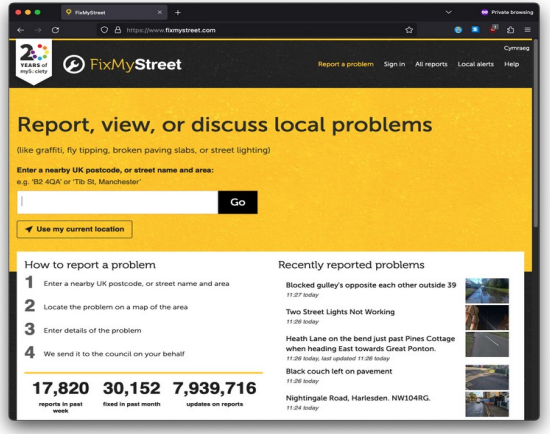
He doesn’t build networks.

Established in 2003, mySociety is a not for profit group pioneering the use of online technologies to empower citizens to take their first steps towards greater civic participation.

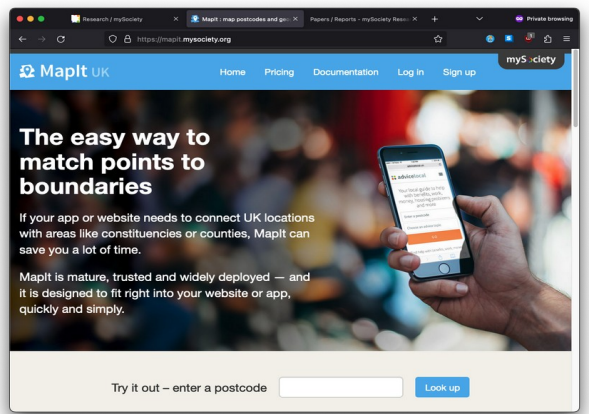
We help people be active citizens with technology, research and data that individuals, journalists, and civil society can use, openly and for free.

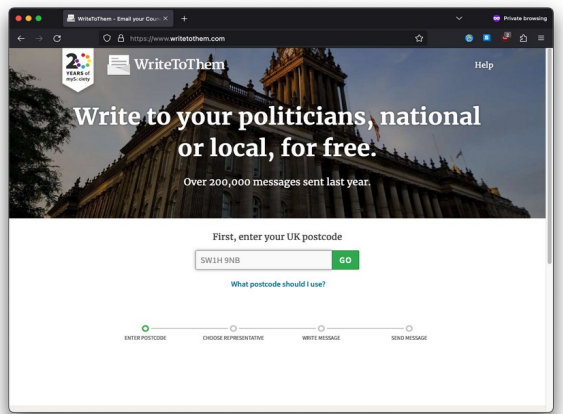


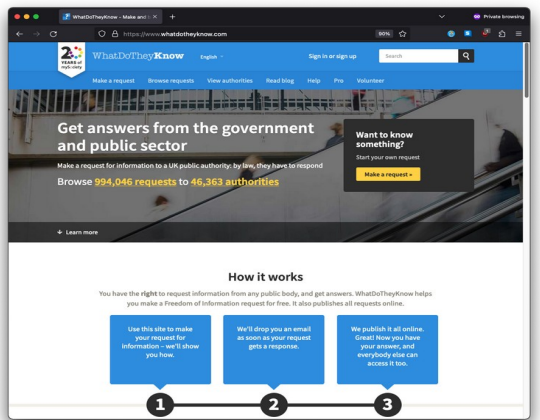




We run a number of relatively well-known web services, including TheyWorkForYou, WhatDoTheyKnow and FixMyStreet, used by millions of people across the UK.







Some of our systems are operated on a commercial basis by our wholly owned subsidiary, SocietyWorks, and generate income for the charity. These are subject to commercial SLAs with our clients.

We have a small technical team with only one full-time role dedicated to systems and infrastructure.

In total we currently run approximately 30 production web services plus a number of internal, staging and development systems.

Many of our services have been operating since the mid 2000s and use a range of languages and frameworks including Perl, PHP, Ruby on Rails and Python Django.

Regardless of the language and framework most services are organised as fairly traditional three-tier web applications. Boring? Maybe, but it works.

Nginx and Varnish sit at the edge on dual-stack systems. We also run our own SMTP and DNS.

Our application tier run middleware process managers suitable for the framework in use and are horizontally scalable.

Our data tier includes PostgreSQL, MySQL, Redis and Xapian.

For systems management and monitoring we use a combination of Puppet, Ansible, Icinga, Prometheus and Grafana together with various Perl and Python homebrew tools.

Before we moved into a private cloud environment at Mythic Beasts we operated our services on a number of large dual-stack physical machines at a now defunct UK hosting provider.

We had a small presence in some large public cloud and platform-as-a-service providers but we found that controlling cost was a significant challenge.

Given time and resource constraints refactoring our well established services to be “microservices” or more “cloud native” in an effort to chase cost-savings wasn’t a realistic option.

Plus we were cautious of the risk of vendor lock-in and proliferation of infrastructure providers leading to further complexity.

But we needed more resources and flexibility for growth and redundancy and we had to get out of our old provider.

We already knew Mythic Beasts and were very interested in the potential of their private cloud offering.

This offered us a good balance between dedicated resources, costs and flexibility together with high technical skills.

But the private cloud assumes that the majority of your systems will be IPv6 only. Could we make this work?

Mythic Beasts take care of NAT64 as part of their private cloud platform, essentially making this requirement work transparently.

We also had a head-start as we’d been running fully dual-stack for many years.

We encrypt the majority of traffic between our systems and our firewall automation only needed some simple changes to cater for systems without IPv4 addresses.

A number of our monitoring checks defaulted to IPv4 but this was also relatively simple to change due to our existing automation.

In fact we found only a single critical application, the Xapian project’s replication server, that didn’t support IPv6 at all.

We resolved this using a transparent Nginx TCP streaming proxy bound to the IPv4 loopback interface. This had the added benefit of making it easy to encrypt the traffic.

I’d estimate that supporting IPv6-only was no more work than that needed to support changes between major OS versions. (Which we did at the same time as our migration into the Private Cloud.)

In the end the biggest challenge wasn’t supporting IPv6 but rather the task of moving everything with the minimum disruption to our users which would be true of any wholesale hosting migration.

We got access to our new Private Cloud in early October 2021 and were deploying production services there by November.

We had completed the vast majority of the migrations and upgrades (we updated our base builds from Debian Stretch to Bullseye at the same time) by July 2022.

We currently operate approximately 60 virtual machines in two separate locations and are looking at at expanding to a third.

Of these systems, only 7 are dual stack.

We have adequate capacity and redundancy and can predict and manage our cost base more efficiently.

Moving away from IPv4 and private networking might seem a huge project, but it doesn’t have to be.

You need good testing and automation and it helps to have an experienced and reliable hosting partner!

MySociety

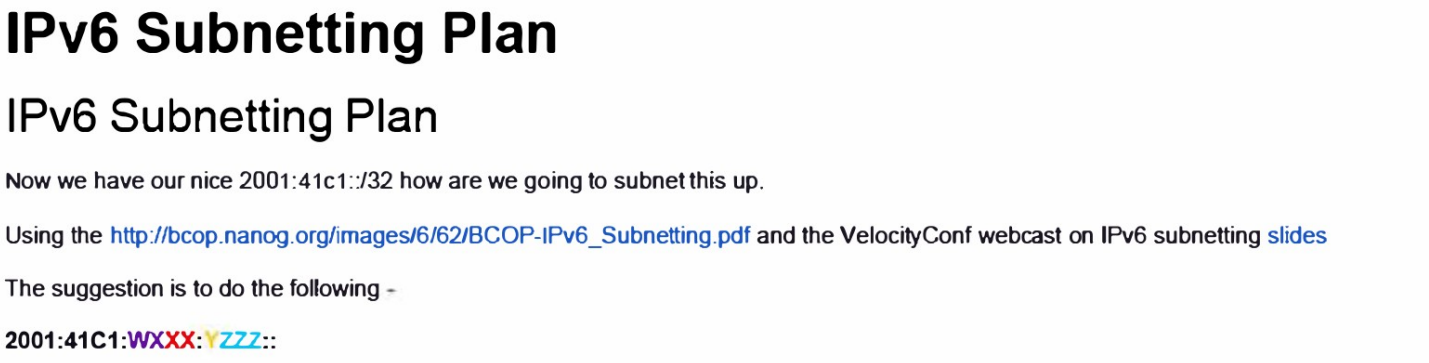
- Sam has undersold how awesome MySociety are
  - They built the petitions site for No10 that forced the government to apologise to Alan Turing
  - Eleanor Shaikh used ‘What do they know’ to uncover the Horizon postmaster scandal
  - A million searchable FOI requests gives you a lot of data to hold the authorities to account

FOI

Liao Bai 18 April 2022 — Delivered

Dear British Broadcasting Corporation,

Please provide a copy of your policy documents and implementation plans for providing the BBC's internet services over IPv6.



IPv6 Subnetting Plan

IPv6 Subnetting Plan

Now we have our nice 2001:41c1::/32 how are we going to subnet this up.

Using the http://bcop.nanog.org/images/6/62/BCOP-IPv6_Subnetting.pdf and the VelocityConf webcast on IPv6 subnetting slides

The suggestion is to do the following -

2001:41C1:WXXX:YZZZ::

IPv4 still isn't quite dead!

- Some external sources want to firewall to just our source IP address
- So NAT64 needs to originate from an IPv4 address not shared with another customer
- Per customer NAT64 :-(

DNS/NAT64

- Private NAT64 doesn’t need private DNS64
- Configure per customer prefixes on central DNS64 servers
  - When we upgraded bind in 2023 we discovered it now enforces RFC6052 (from 2010) so bits 64-71 of your prefix must be now be zero
  - We had to retire prefix:6464:6464:<embedded-v4> (oops)

The End

- Questions?
- https://www.mythic-beasts.com/blog
- https://www.mysociety.org/
- https://social.mythic-beasts.com/@beasts
- https://www.facebook.com/mySociety
- @Mythic_Beasts and @mysociety at a social network that thinks IPv6 is harder than rocket science