



Porting Envoy To Windows

A Status Update

William A. Rowe Jr. - Pivotal

Yechiel Kalmenson - Pivotal

Agenda

- Who are we, and what is Envoy anyways?
- What Issues have been solved, will be solved again, and are in progress today?
- What's ahead for Windows?



Introductions

Yechiel Kalmenson



- Software Engineer at Pivotal
- Working on the Windows Containers team
- Has been working on the Windows Envoy port for about a year
- @yechielk on Slack and social media

William A. Rowe, Jr.



- Principal Software Engineer at Pivotal
- Member of the Pivotal App Suite team in the Spring / Cloud Foundry organization
- Instrumental in the Windows port of APR 2.0, the Apache Portability Runtime underneath Apache Web Server, Subversion and others
- Joined the Windows Containers team effort to port Envoy this past Spring
- @wrowe on Slack and social media

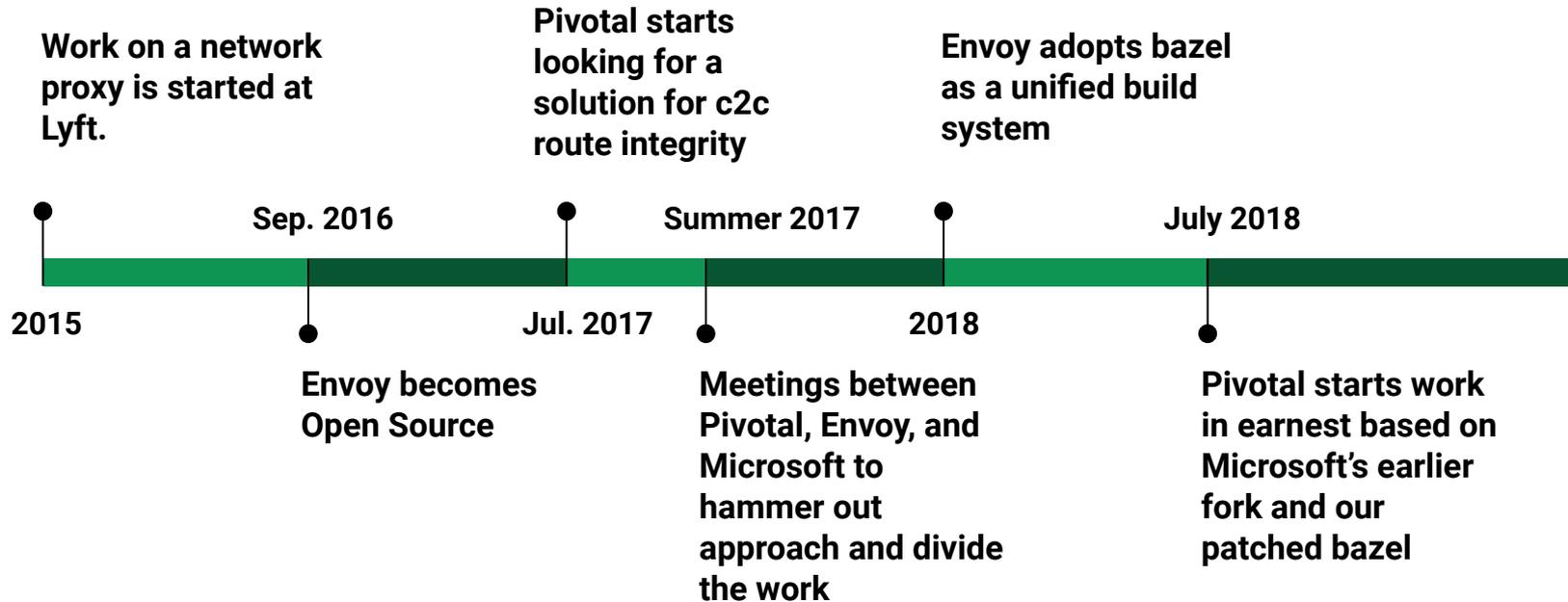
A group of people in a meeting room. A man on the left is pointing at a whiteboard. Several people are sitting on stools, listening. A man on the right is standing with his arms crossed, looking towards the whiteboard. The scene is dimly lit with a blue tint.

What is Envoy?

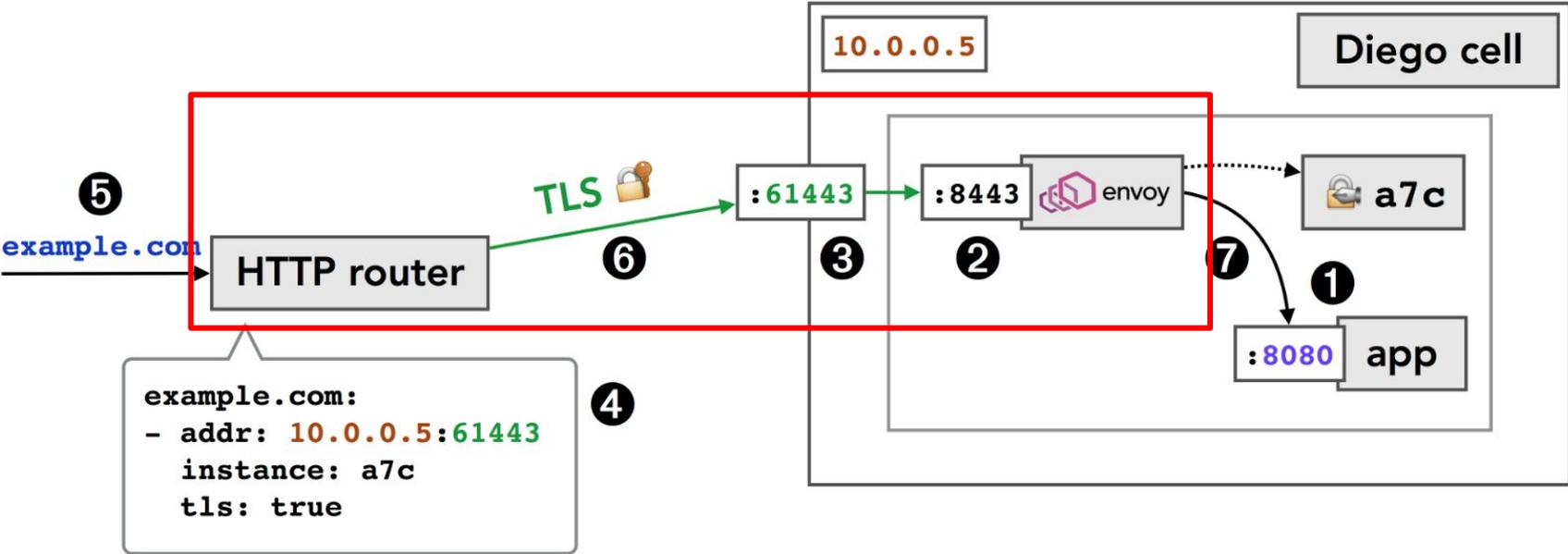
“The goal of Envoy is to be an abstraction that allows higher layer systems to be built that make the network transparent to end users.”

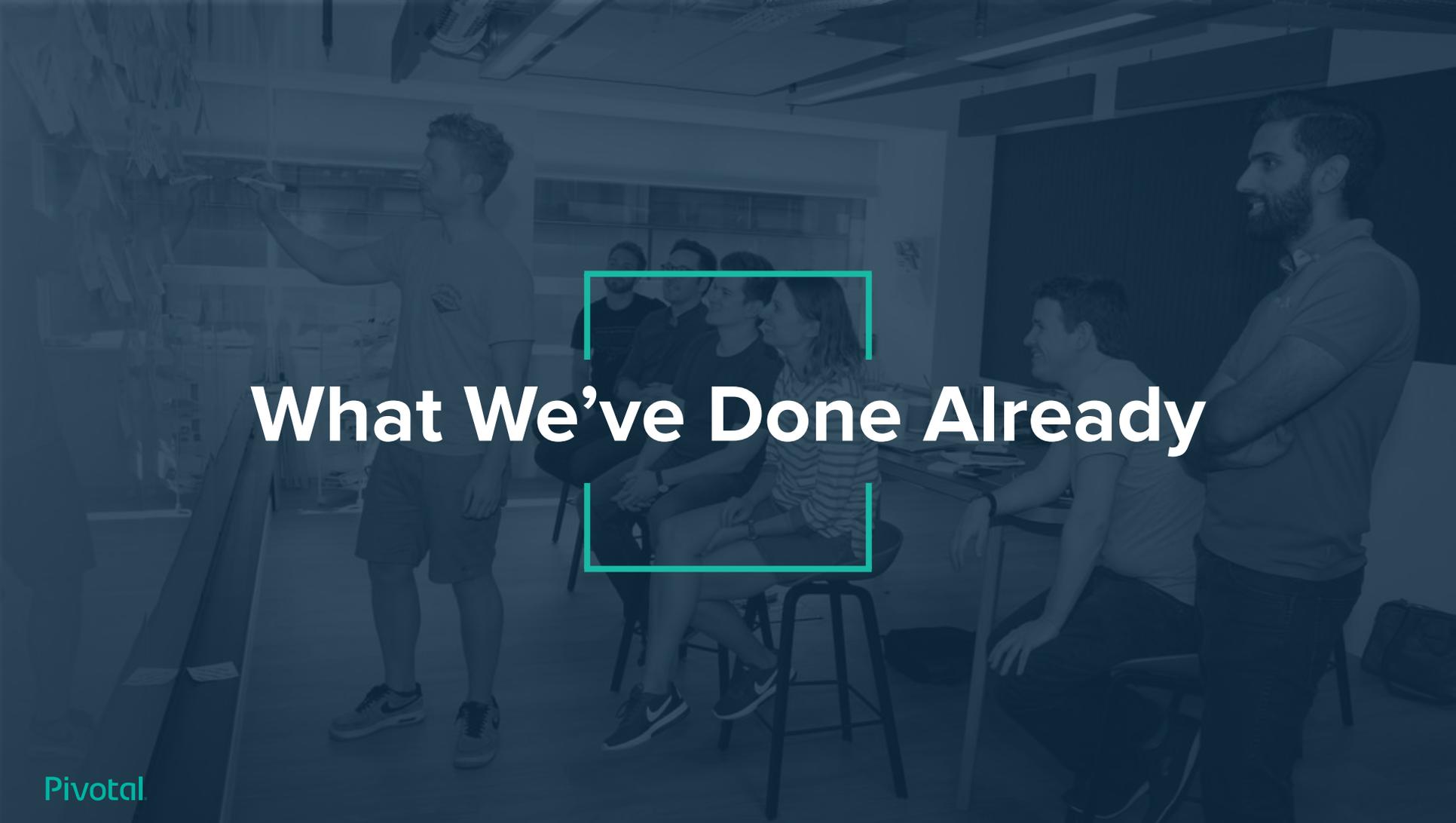
- Matt Klein, Lyft

Timeline

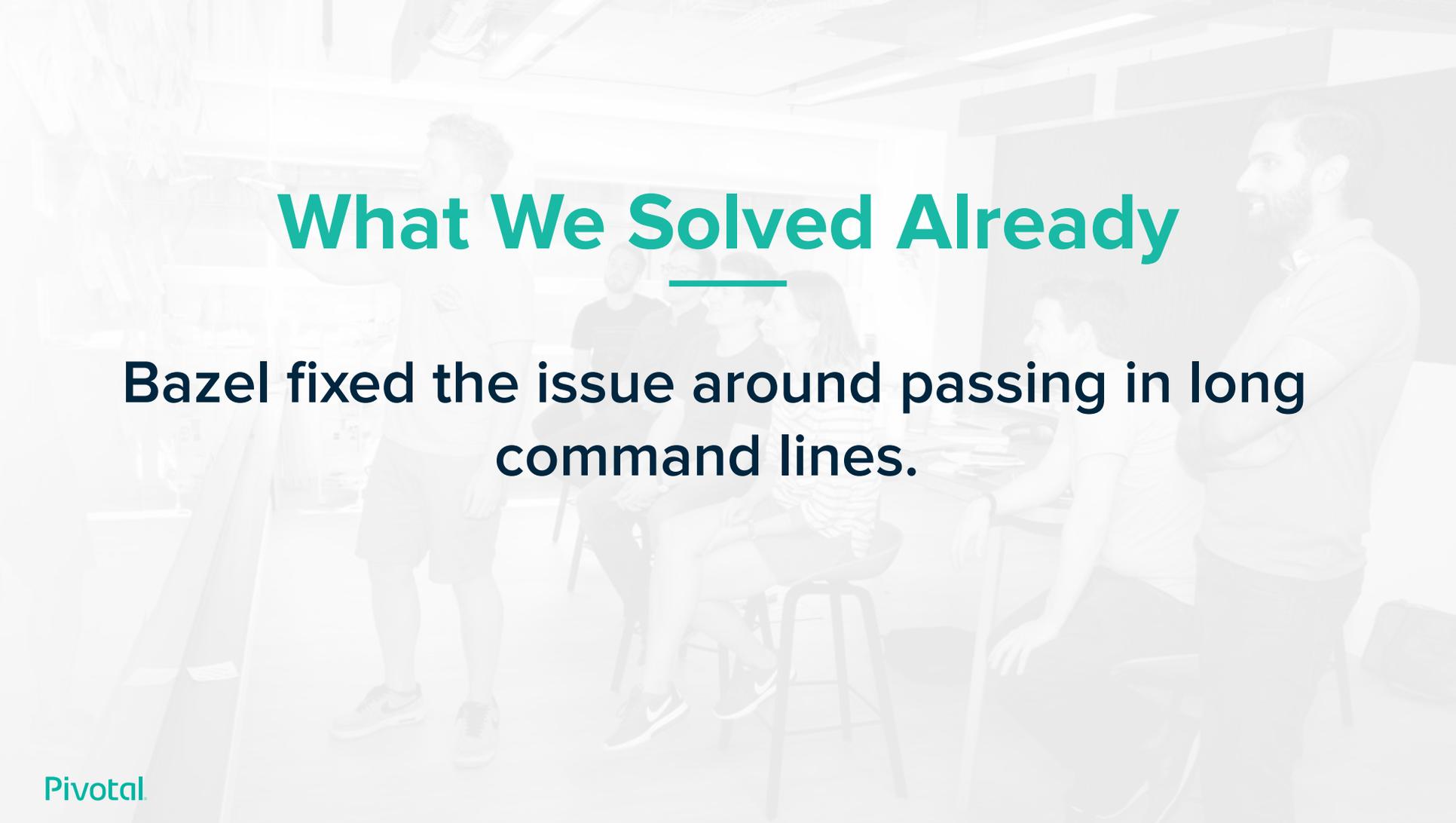


Pivotal Use Case



A group of people in a workshop or classroom setting. A man on the left is pointing at a wall covered in papers. A group of four people is seated in the center, highlighted by a teal square. A man on the right is standing and talking to a seated man. The background shows a chalkboard and shelves.

What We've Done Already

A group of people in a meeting room, with one person standing and pointing at a screen.

What We Solved Already

Bazel fixed the issue around passing in long command lines.



What We Solved Already

Many Linux-specific socket issues were resolved by OS/X work around BSD sockets

What We Solved Already

`<windows.h>` `#define`'s many common keywords that had to be `#undef`'ed in every file where windows headers were included

```
4  #include <windows.h>
5
6  // <winsock2.h> includes <windows.h>, so undef some interfering symbols.
7  #undef TRUE
8  #undef DELETE
9  #undef ERROR
10 #undef GetMessage
11
```

A group of people in a meeting room, with a man standing and pointing at a whiteboard.

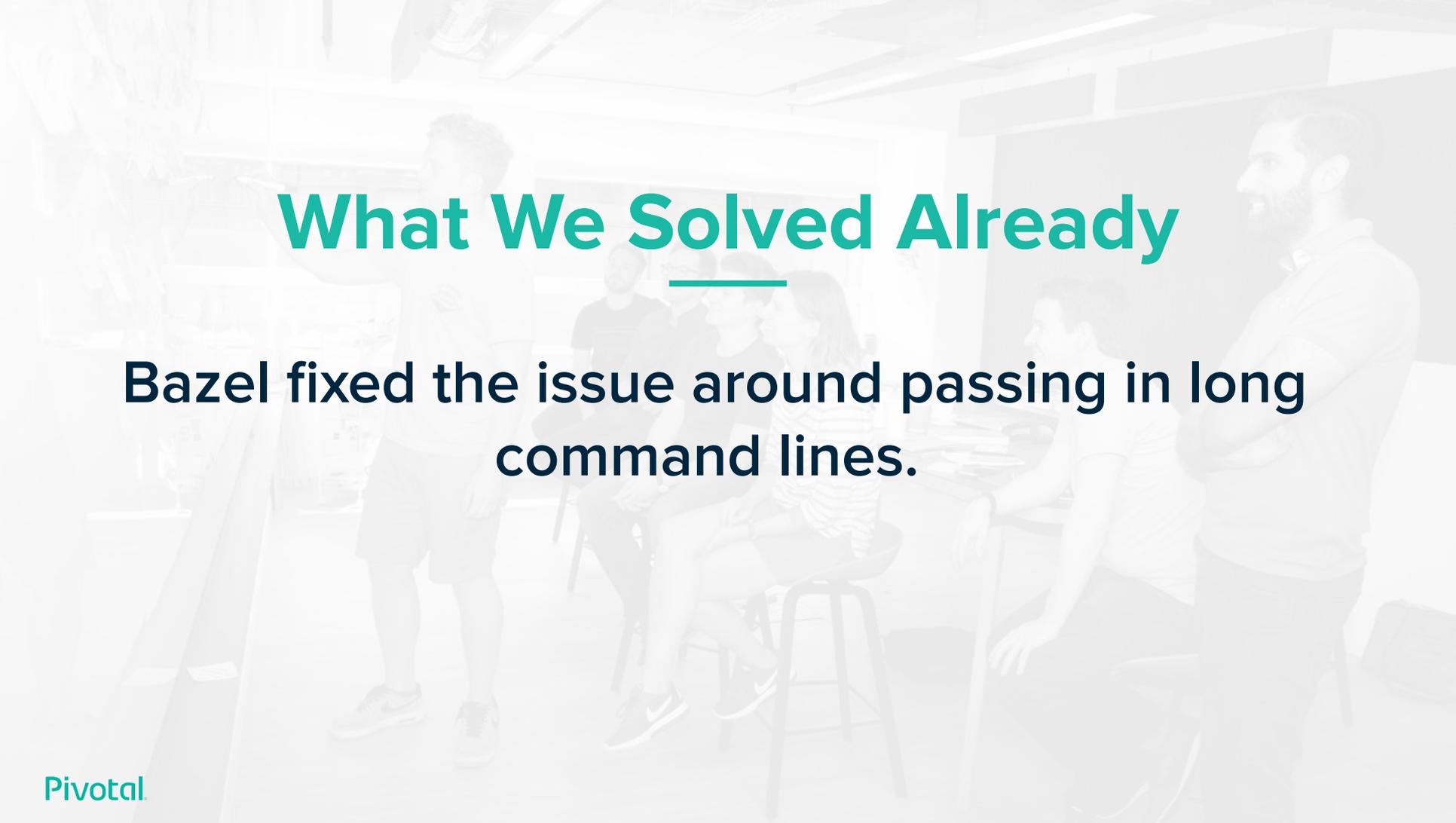
What We Solved Already

bazel_foreign_cc has been greatly simplified for our CMake dependencies



What We Solved Already

All external dependencies are bundled into
`envoy-static.exe`

A group of people in a meeting room, with one person standing and pointing at a screen.

What We Solved Already

Bazel fixed the issue around passing in long command lines.

A group of people in a meeting room, with a teal square highlighting a group of four people. The scene is dimly lit with a blue tint. A man on the left is pointing at a wall covered in papers. A group of four people (two men and two women) are seated in the center, looking towards the left. A man on the right is standing and talking to another man seated at a table. The overall atmosphere is professional and collaborative.

What Still Needs To Be Done

Issues Which Can Be Reintroduced By Other Contributors

The size of an 'integer' or 'long' on Windows is and will remain 32 bits, while pointers are 64 bits (we refer to this as a 64P architecture.)

On Linux, OS/X and other *nix variants, the 'int', 'long' and pointer types are all 64 bits (64ILP architecture).

This will frustrate contributors to the Envoy project, which prefers fixed-size types such as `int64_t`, `uint32_t` etc. The Standard C++ library interfaces must conform to the actual size of the platform's 'int', 'long' types, etc.

Issues Which Can Be Reintroduced By Other Contributors

The Windows MSVC compiler began and never completed a full Posix implementation. Many commonly used Posix data types and functions aren't available, or differ in name or implementation details.

A prime example is the lack of `'ssize_t'` type on Windows, versus the ISO/C++ `'ptrdiff_t'` type which is part of the language standard.

Issues That Need Collaboration

Many small bugs need to be patched, because Microsoft, the GNU gcc project, and the Llvmlang clang communities all interpret the C++17 language standard in slightly different ways.

E. G. Alternative operator representations such as 'not', 'and', 'compl' etc. which were alternatives to '!', '&&', '~' etc. on gcc and clang, but were never supported by Microsoft's cl.exe.

Issues That Need Collaboration

In solving these issues, the question has come up whether clang-cl.exe, the LlvM compiler project's Microsoft cl.exe portability interface, will prove to be a better solution than MSVC.

We are exploring both in parallel and expect to support both, long term. Both will be in the project's CI for evaluating patch submissions.

Issues for Windows In Progress

Many earlier "solutions" to bazel BUILD configuration need to be simplified.

Different approaches were taken at different points in the development of the bazel build environment, the final clean solutions need to be deployed.

Issues for Windows In Progress

Many tests are based on invoking bash and specific GNU toolchain utilities, which differ from the Windows toolchain, need to be adjusted.

Issues for Windows In Progress

Envoy is sensitive to file changes (such as configuration files), a different implementation of file change notification needs to be completed

Issues for Windows In Progress

Process control primitives are very different on Windows vs. *nix OSs. Signal processing code for Windows conventions is still needed, and new code is needed to run Envoy as a background process (service.)

Issues for Windows In Progress

Windows SOCKET and HANDLE objects won't fit into an 'int' type as Unix file descriptors do. These are being abstracted into a IoHandle class.

Issues for Windows In Progress

The windows socket API is level triggered, not edge triggered, and this results in poor poll-based performance on Windows. Windows has its own APIs such as I/O Completion Ports, so an alternate state engine may be necessary.

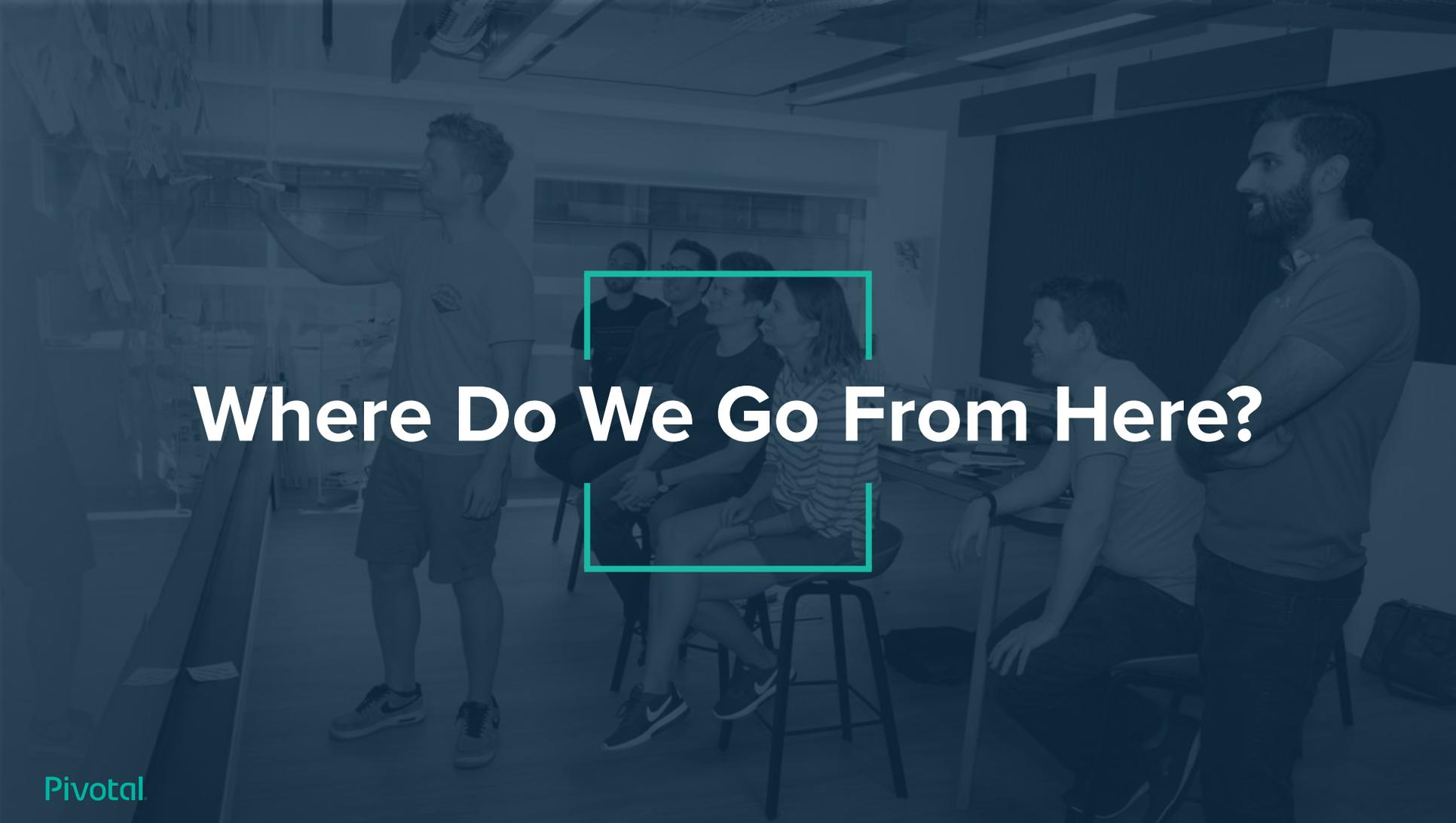
Issues for Windows In Progress

Envoy's engine is based on the libevent project. Alternatives include libev and libuv which may be better matches for Windows.

Windows Progress and Extensions

Envoy is built from a collection of extensions

- The bulk of Envoy consists of extensions to the core processing engine.
- Some examples include HTTP/1 vs HTTP/2 provided by the nhttp2 library vs QUIC provided by the quiche implementation.
- Other categories include logging, routing, filtering, and monitoring.
- During porting, we've focused on that core functionality and set the vast majority of extensions aside to focus on the core server.
- We are asking for additional contributors to take on one or a group of related extensions and finish porting that functionality, so that the Windows port is as expansive as the Linux and OS/X ports.

A group of people in a workshop or classroom setting. A man on the left is pointing at a wall covered in papers. A group of four people in the center is highlighted with a teal square. A man on the right is standing and talking to a seated man. The background shows a chalkboard and shelves.

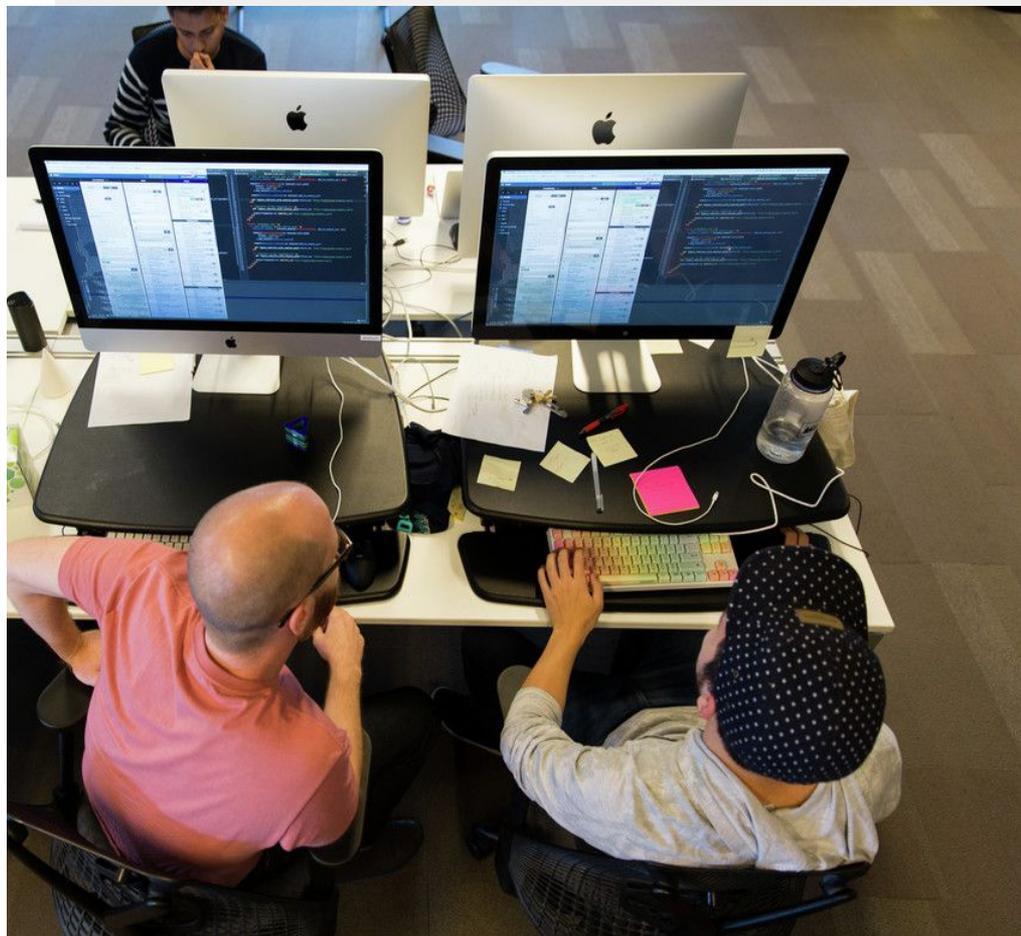
Where Do We Go From Here?

What the Future Looks Like

- **Pivotal remains committed to advancing and maintaining this Windows port.**
- The Envoy community of developers has been very welcoming and supportive of efforts to port Envoy to Windows.
- Microsoft has recently expressed renewed interest in contributing to this port.
- Envoy remains a very flexible platform, so optimizations and features addressing specific platforms and use cases remains very approachable.

How Can I Get Involved?

- Join the `#envoy-windows-dev` channel on Envoy Slack to follow or help with porting and development
- See the Envoy Project on GitHub for open issues tagged [Help Wanted]
- Watch the Building and Installation page in the Envoy Docs for word that Windows binaries dropped
- Join `#envoy-users` channel for peer support once Windows has arrived



A group of people in a meeting room. A man on the left is pointing at a whiteboard. A group of people is sitting on stools in the center, listening. A man on the right is standing with his arms crossed, looking towards the group. The room has a whiteboard, a desk, and a window in the background.

Q & A