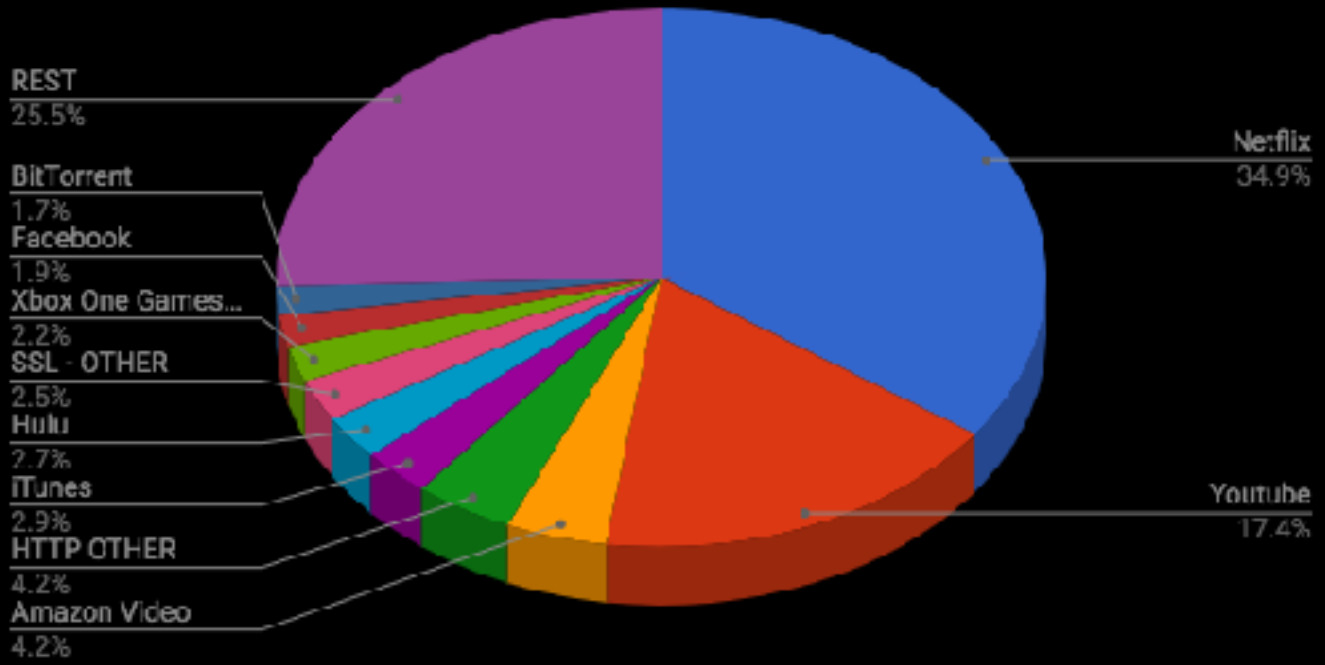
Alistair Crooks
October 25, 2017

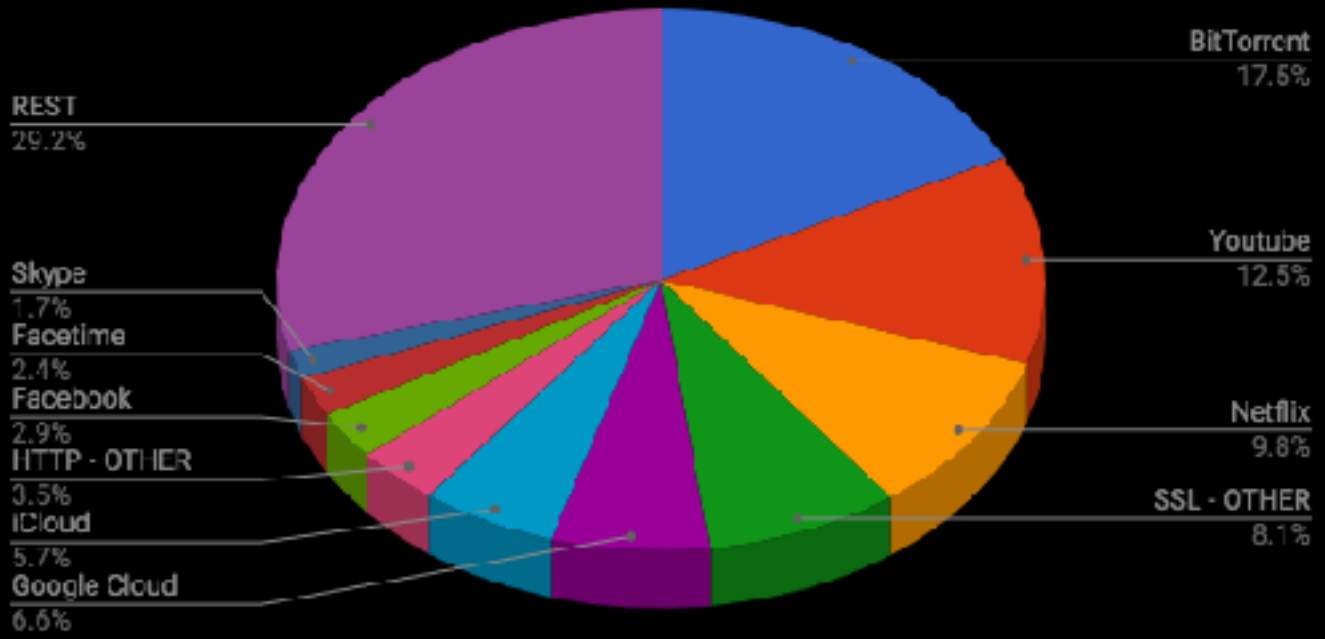
- > 100 million members
- > 190 countries
- > 125 million hours of TV shows and movies per day

- Multiple tens of Tbps
- Hundreds of partners
- Multiple thousands of machines

The Internet: USA, downstream, 2016



The Internet: US, upstream, 2016

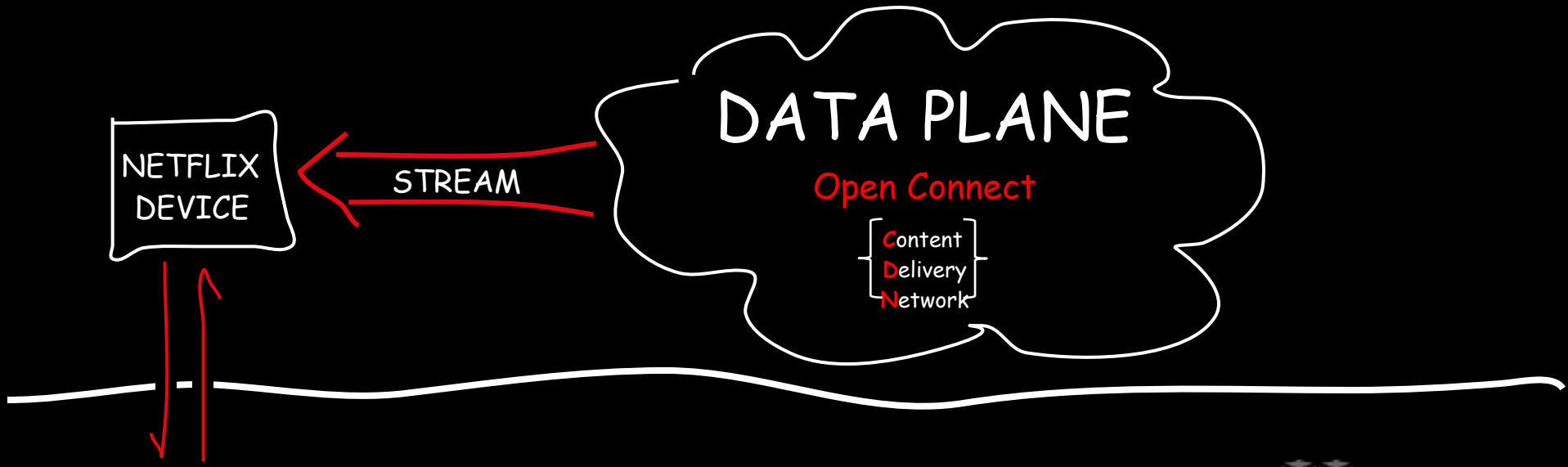



How
NETFLIX
Works


Open Connect Architecture

Use Cases

- Movie/TV Shows
- Pre-located content
- Popularity, churn and fill windows

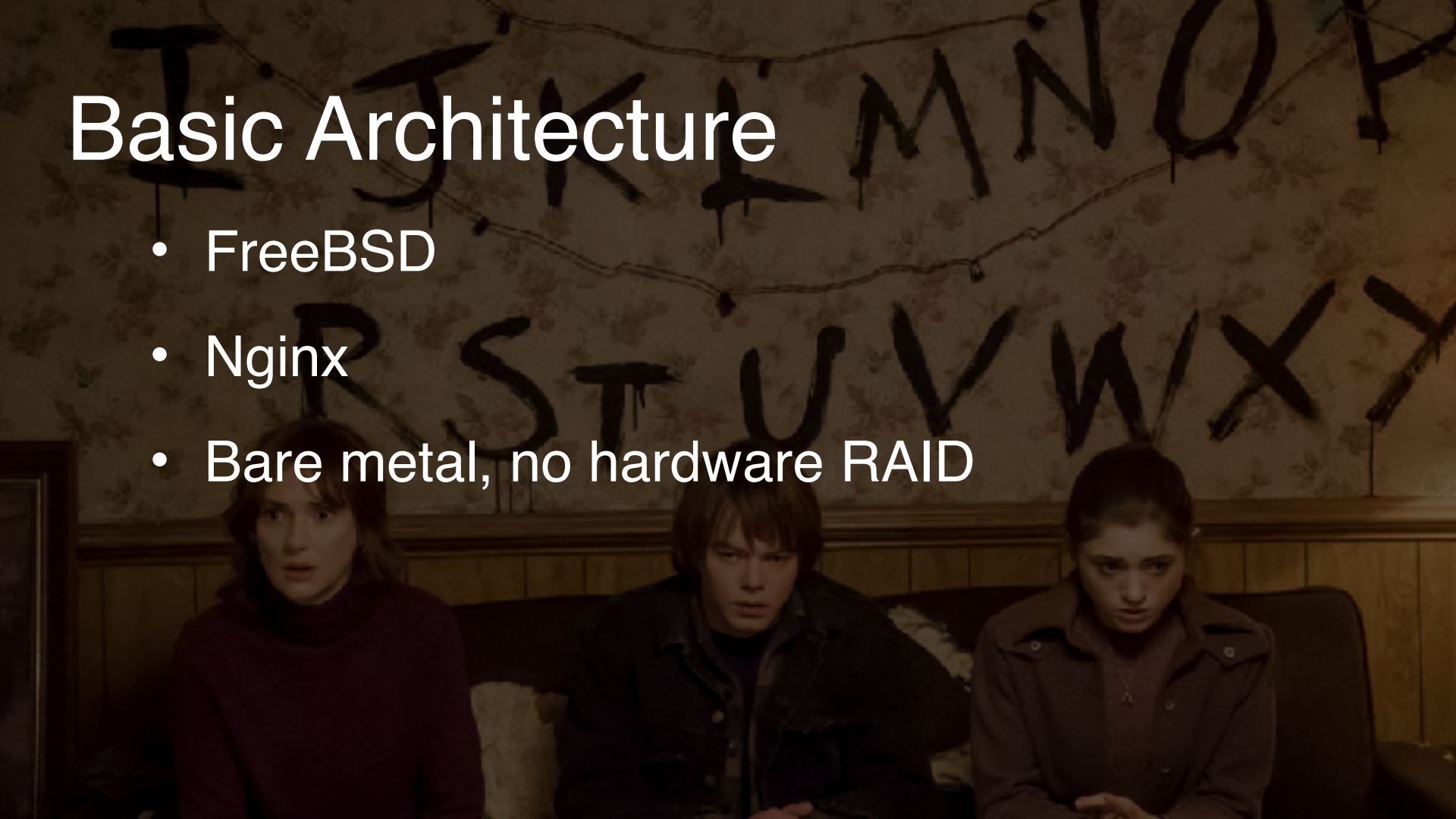


amazon web services

CONTROL PLANE

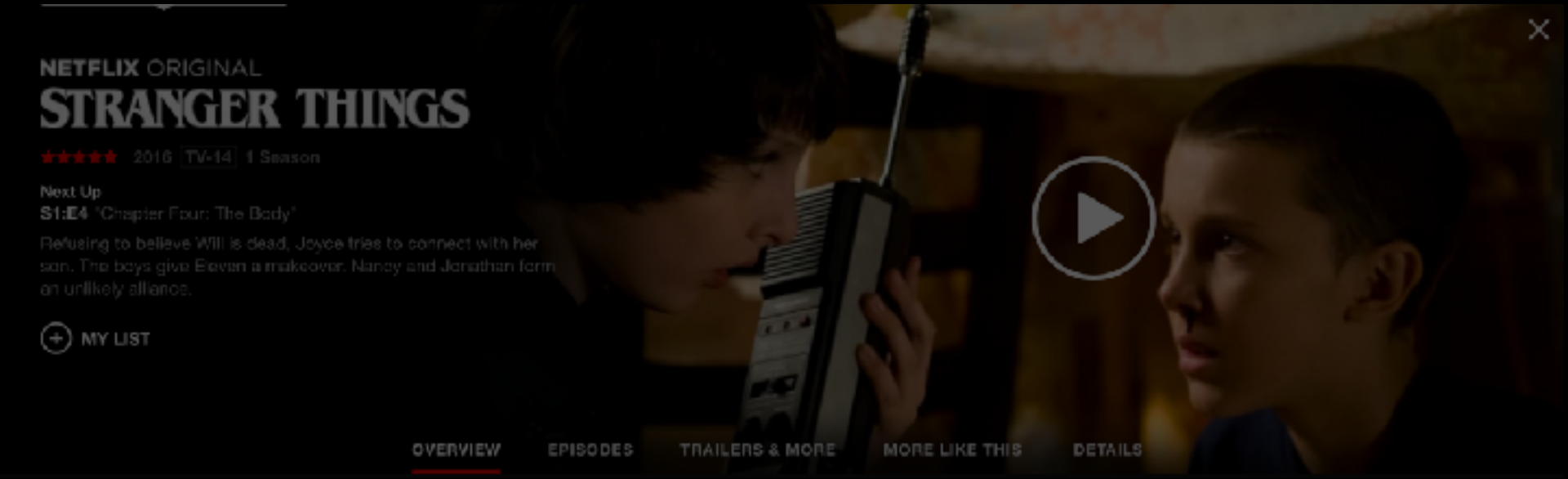
Basic Architecture

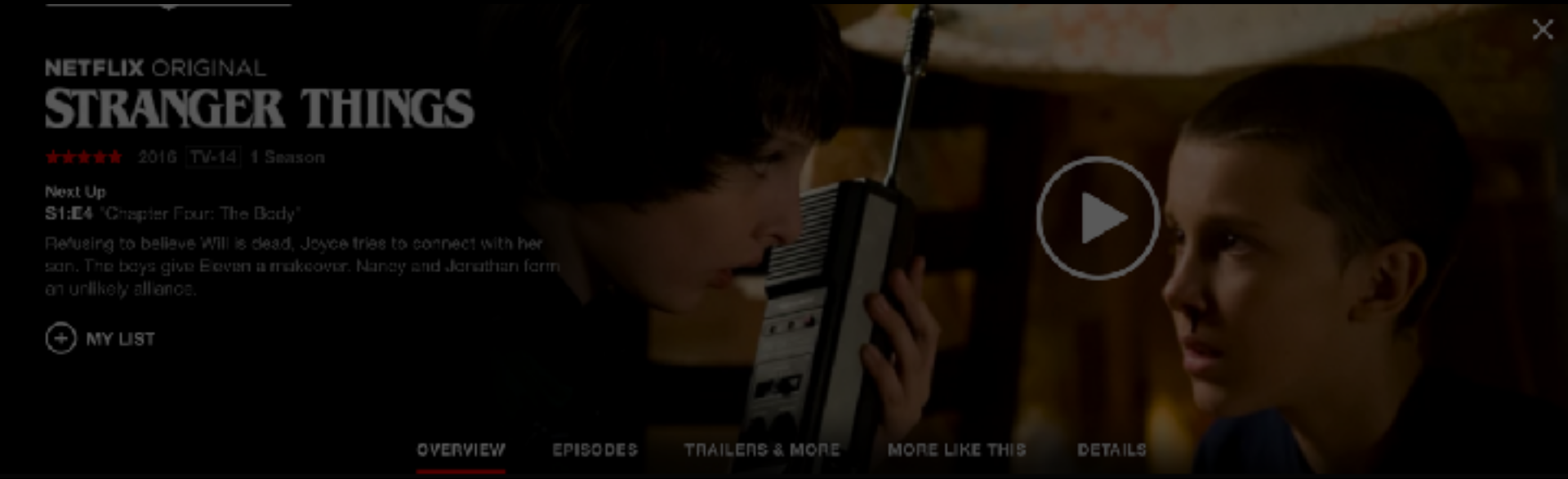
- FreeBSD
- Nginx
- Bare metal, no hardware RAID



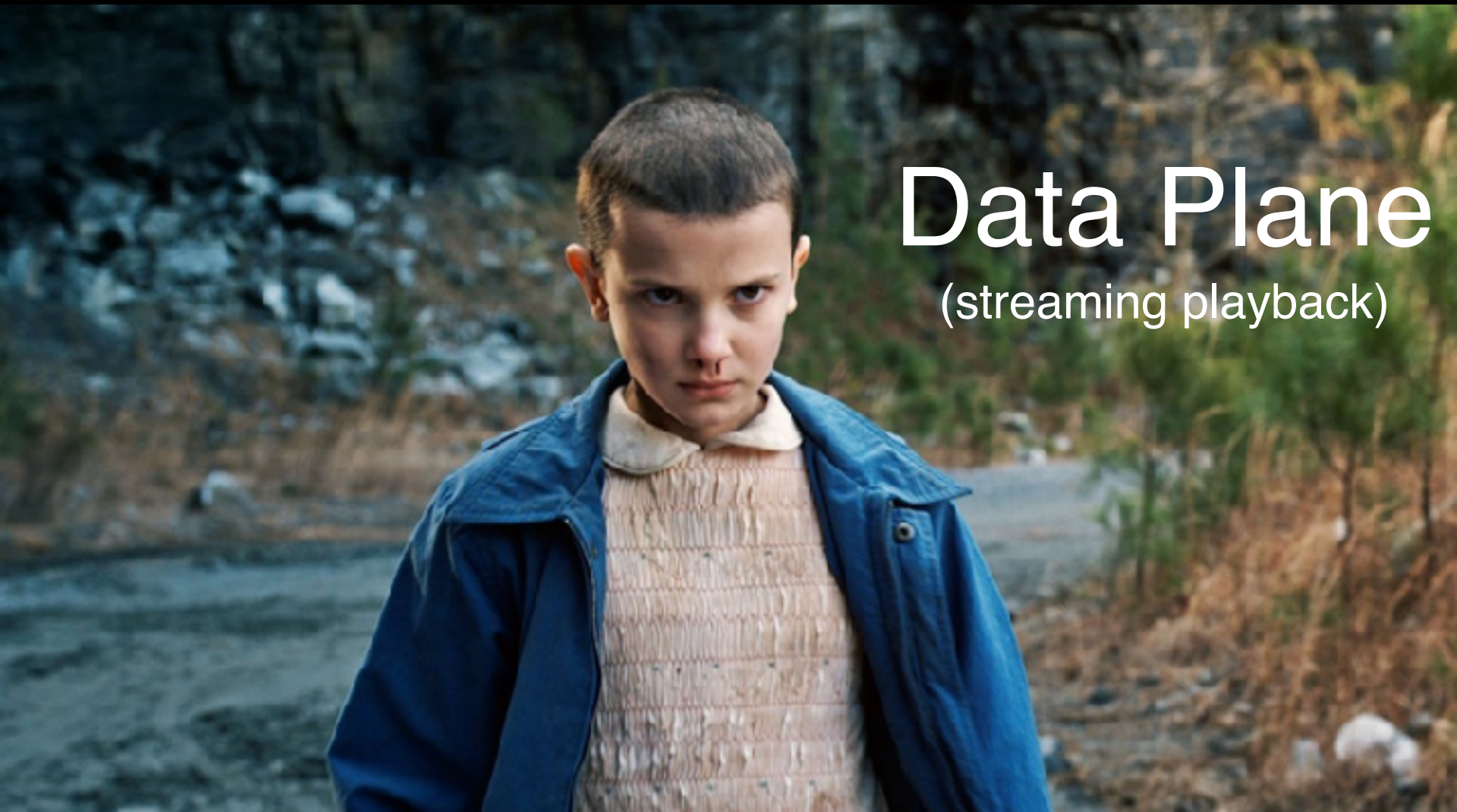

NETFLIX ORIGINAL
STRANGER THINGS
2016 TV-14 1 Season
Next Up
S1:E4 "Chapter Four: The Body"
Refusing to believe Will is dead, Joyce tries to connect with her son. The boys give Eleven a makeover. Nancy and Jonathan form an unlikely alliance.
MY LIST
OVERVIEW
EPISODES
TRAILERS & MORE
MORE LIKE THIS
DETAILS

before streaming starts = control plane =





streaming = data plane = Open Connect


Data Plane
(streaming playback)


Open Connect
A NETFLIX ORIGINAL

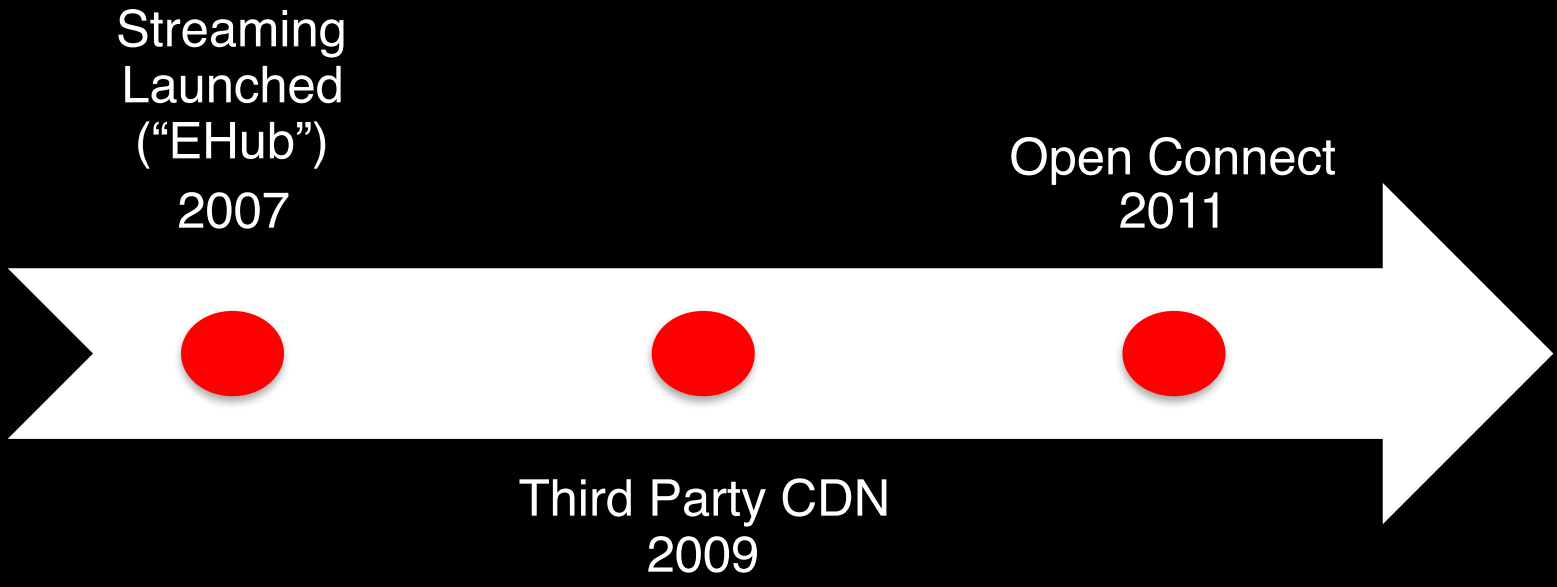
Q: What is a Content Delivery Network?

A: Geographically distributed content servers attached to networks

+

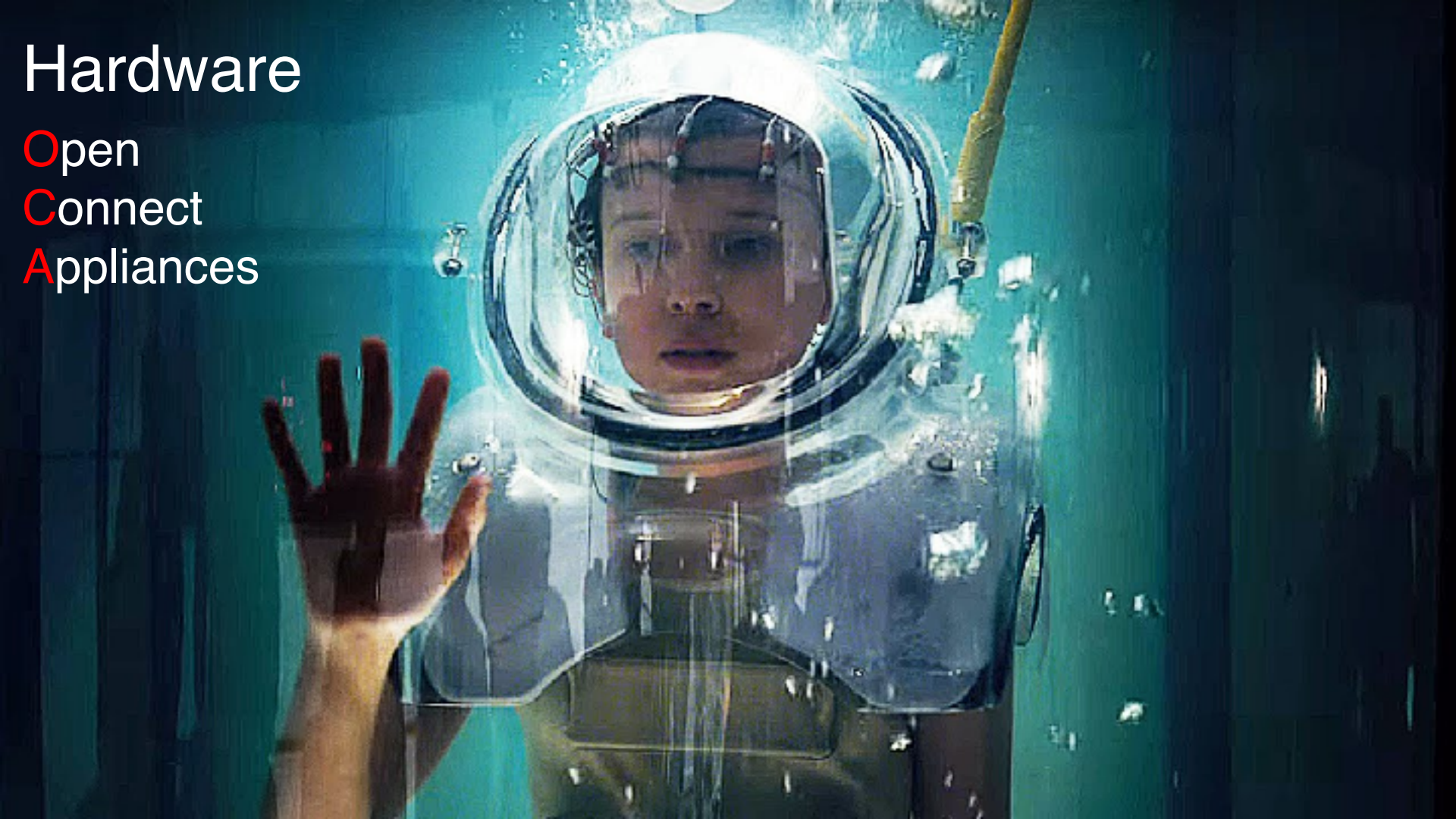
a way of routing requests to the closest (and/or best performing) server / network path

History


Streaming
Launched
(“EHub”)
2007
Third Party CDN
2009
Open Connect
2011

Hardware

Open
Connect
Appliances



Open Connect Appliance

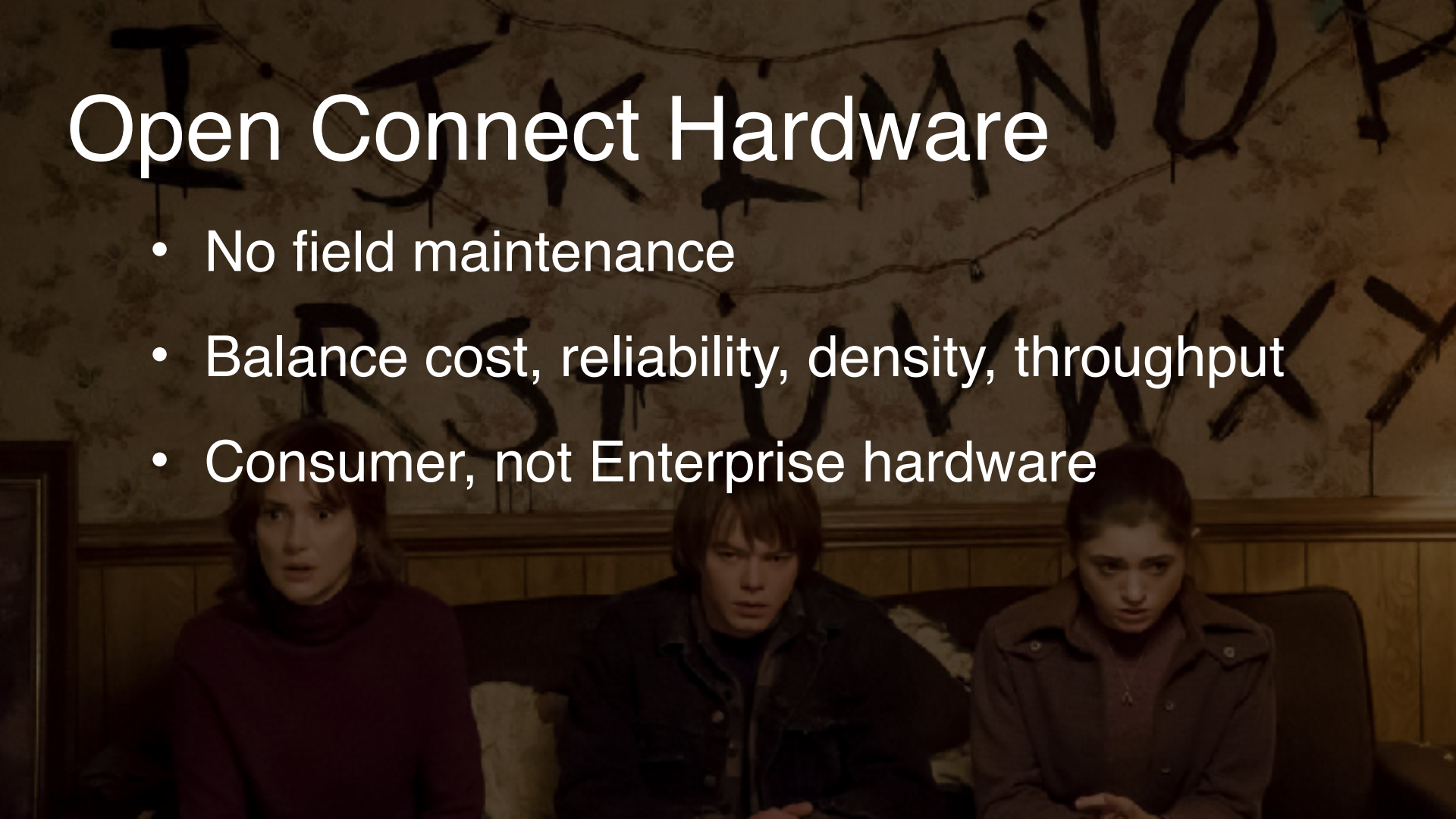
https://openconnect.netflix.com/en/appliances/

Open Connect Appliance Hardware

Type	Storage	Throughput	Use
Storage	108-288 TB HDD 6-10 TB SSD	10-20 Gbps	Hold large % of catalog, ISP sites
Flash	14 TB NVMe	40-100 Gbps	Very high traffic sites
Global	64 TB HDD 6 TB SSD	8 Gbps	Smaller ISP sites

Open Connect Hardware

- No field maintenance
- Balance cost, reliability, density, throughput
- Consumer, not Enterprise hardware



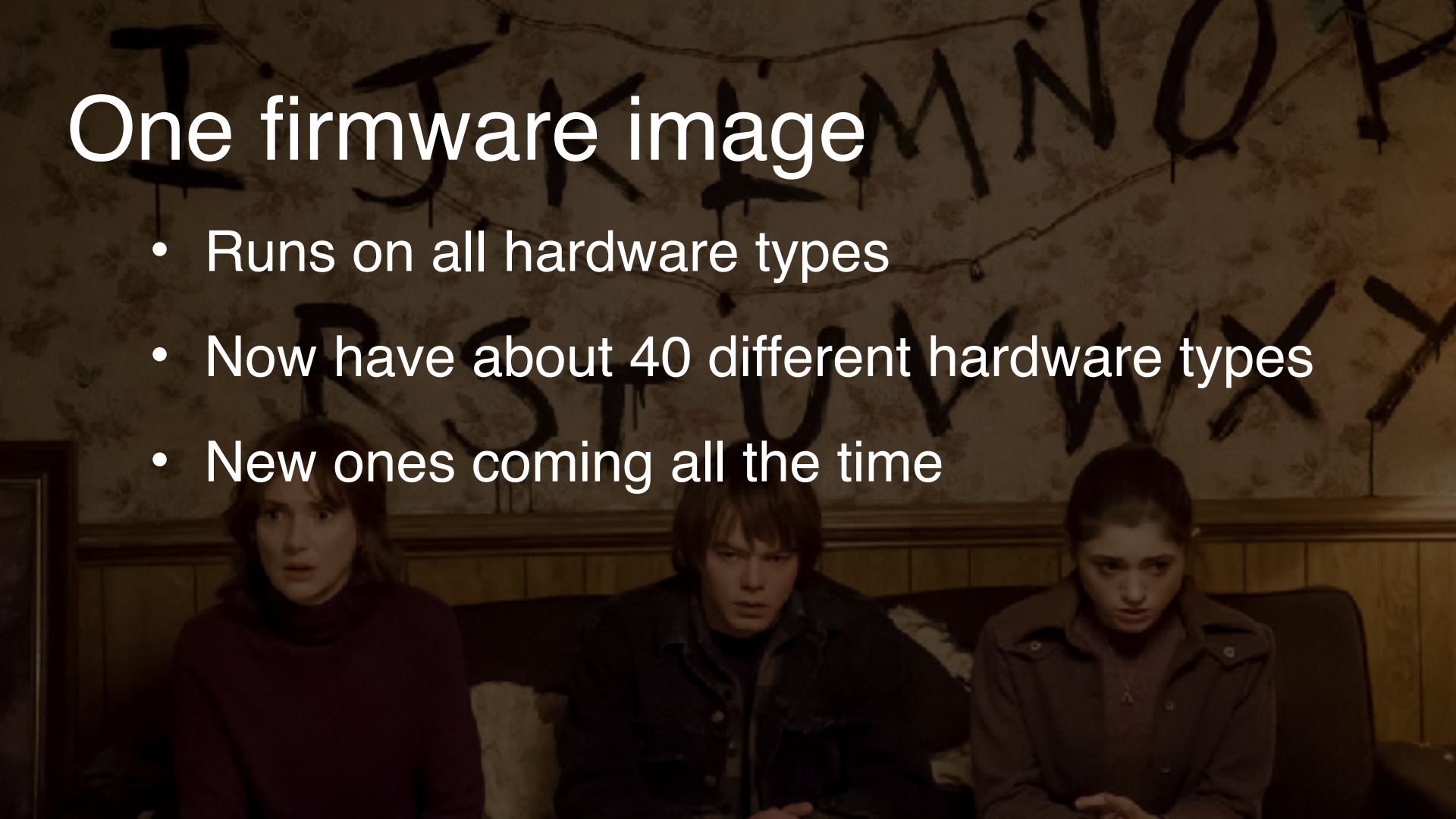
Mid 2017 Storage

Mid 2017 Global

Mid 2017 Flash

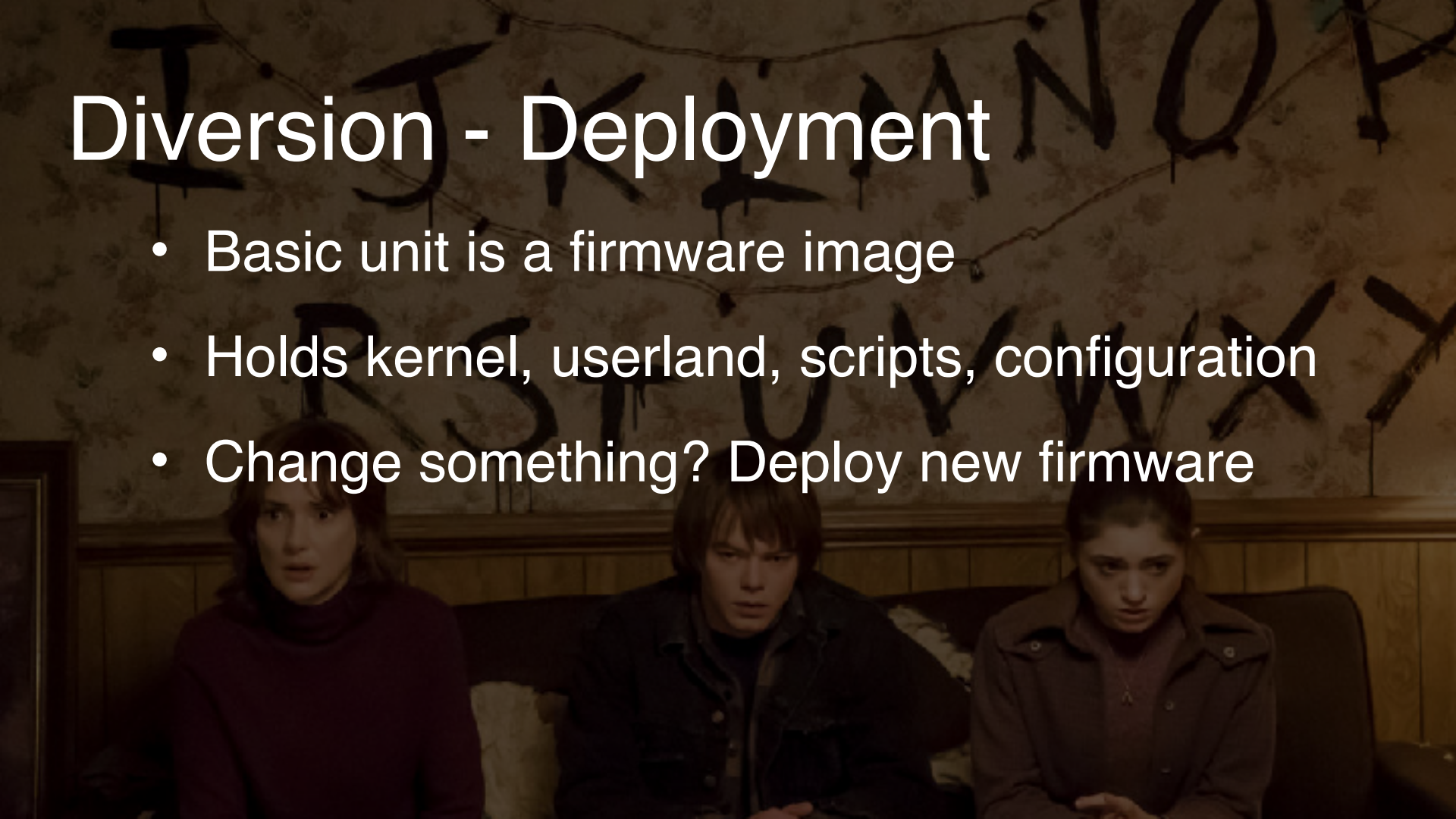
One firmware image

- Runs on all hardware types
- Now have about 40 different hardware types
- New ones coming all the time



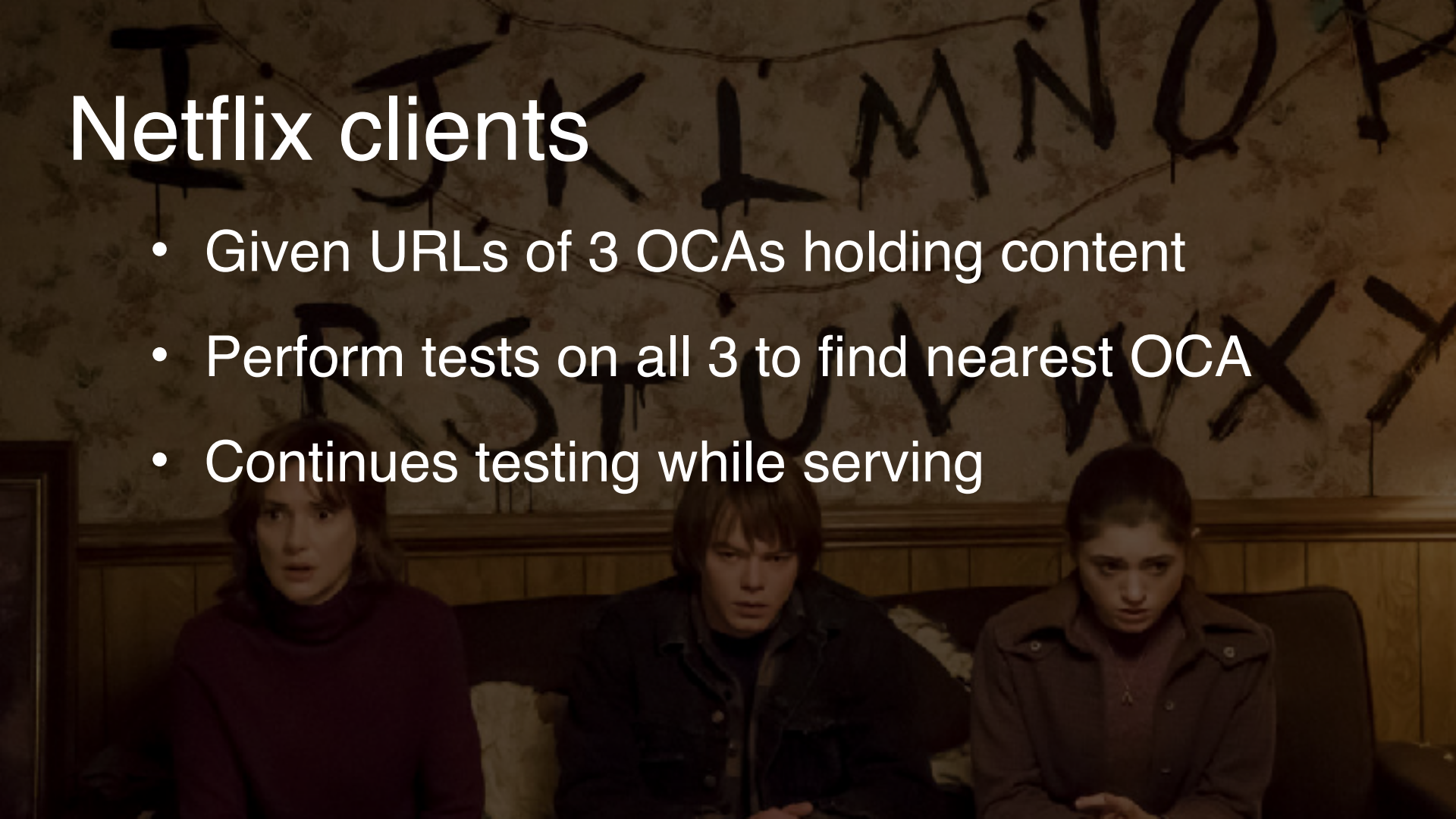
Diversion - Deployment

- Basic unit is a firmware image
- Holds kernel, userland, scripts, configuration
- Change something? Deploy new firmware



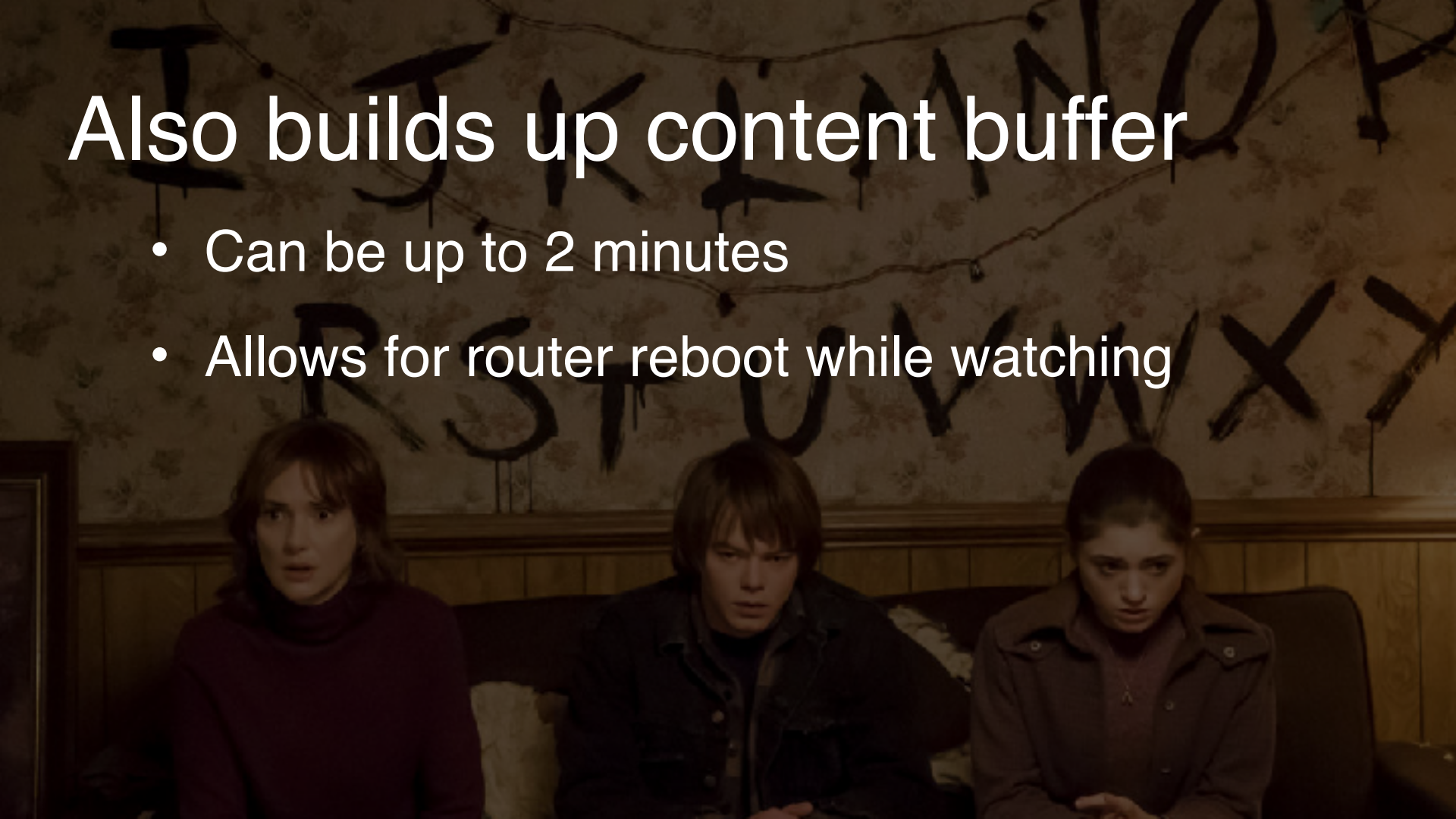
Netflix clients

- Given URLs of 3 OCAs holding content
- Perform tests on all 3 to find nearest OCA
- Continues testing while serving



Also builds up content buffer

- Can be up to 2 minutes
- Allows for router reboot while watching

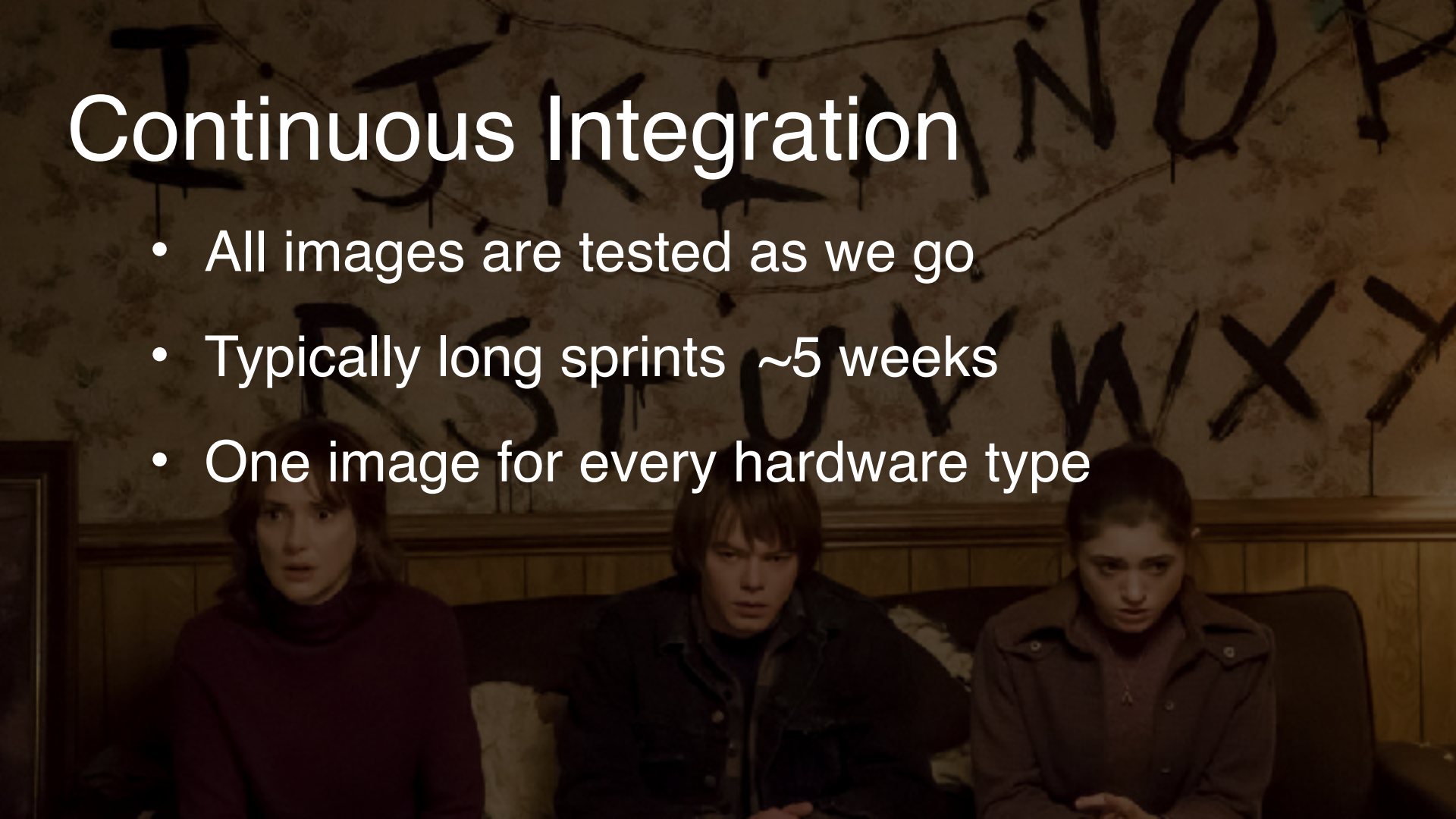


No test network

- All testing in production
- AB testing done across organisation
- Instant idea if a change is good/bad/neutral

Continuous Integration

- All images are tested as we go
- Typically long sprints ~5 weeks
- One image for every hardware type

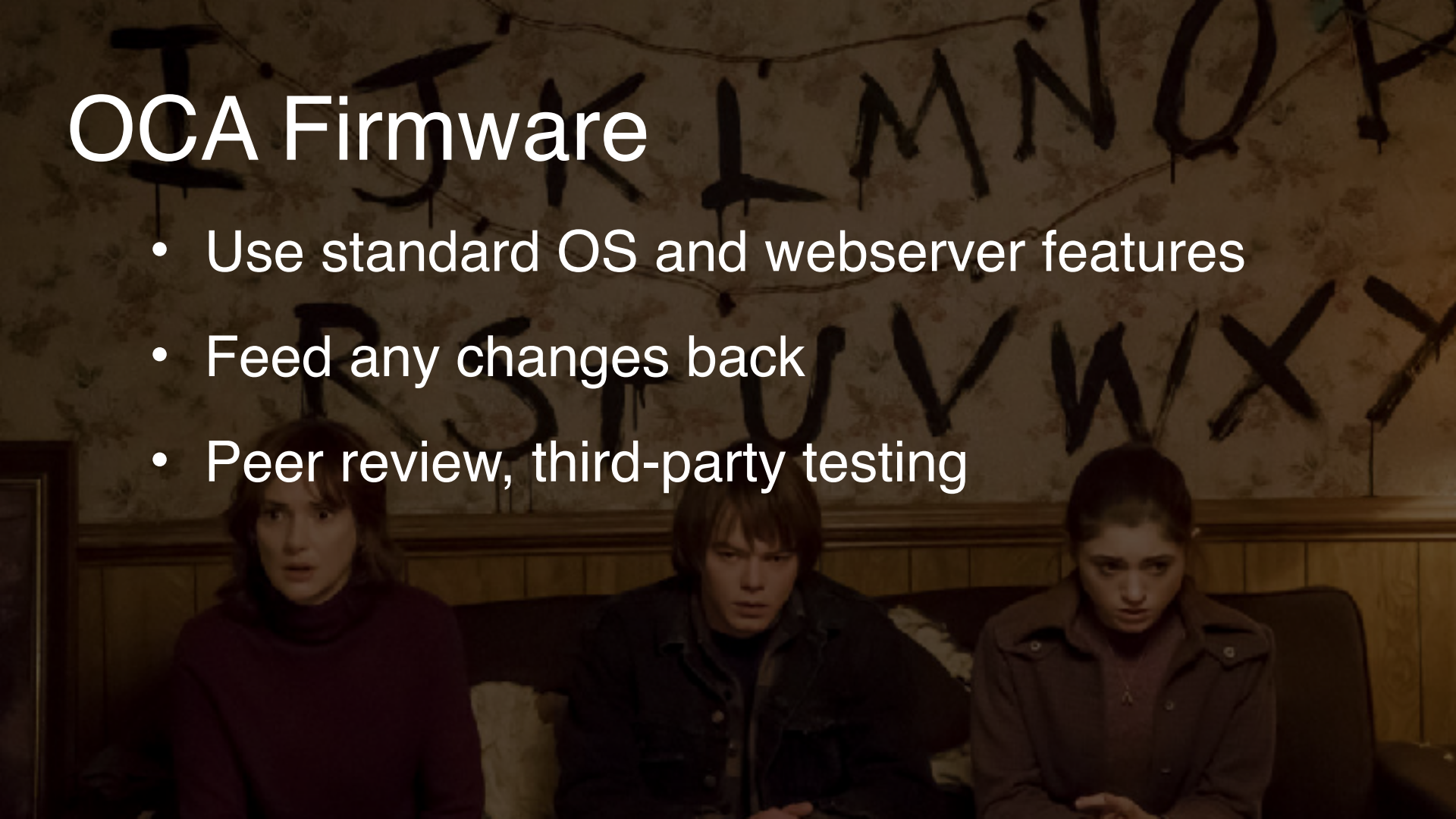


Control Plane

- OCAs ask control plane for desired firmware
- Will download and boot once
- Control multiple OCAs

OCA Firmware

- Use standard OS and webserver features
- Feed any changes back
- Peer review, third-party testing



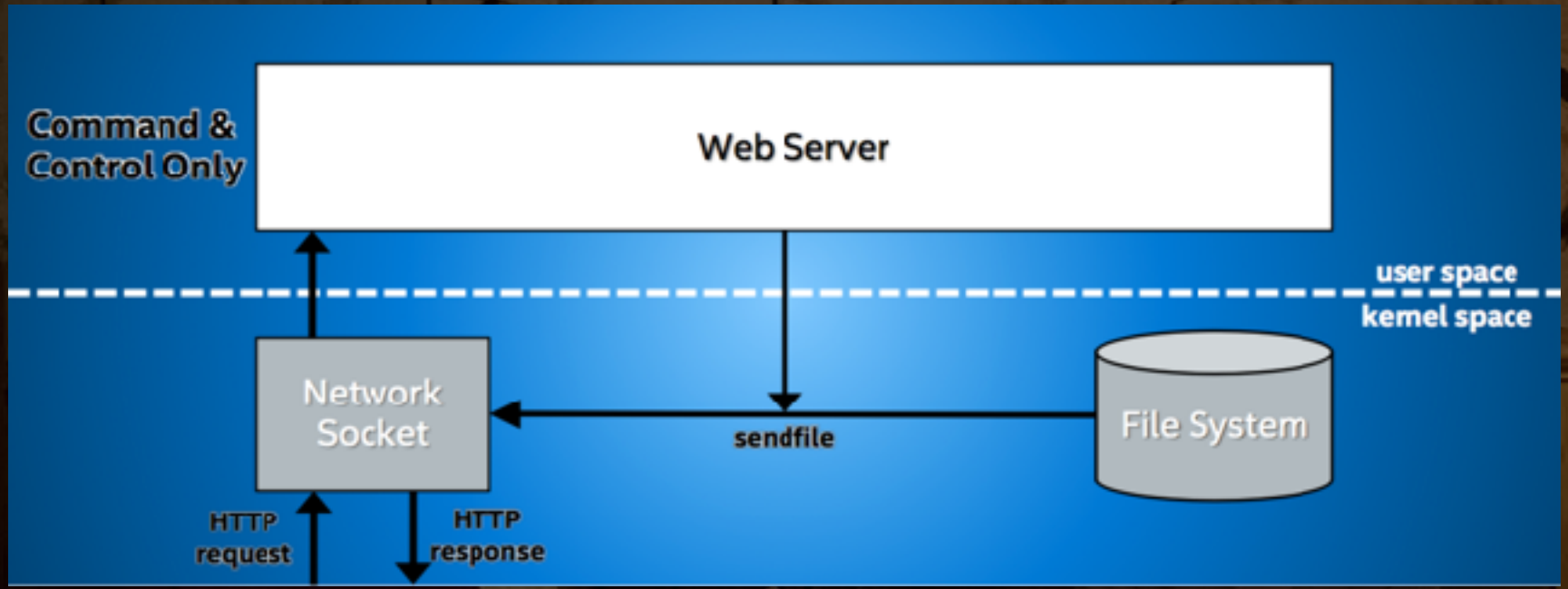
OCA Firmware Implications

- try not to make minor changes
- upstream changes by developer
- bring in changes with next OS sync

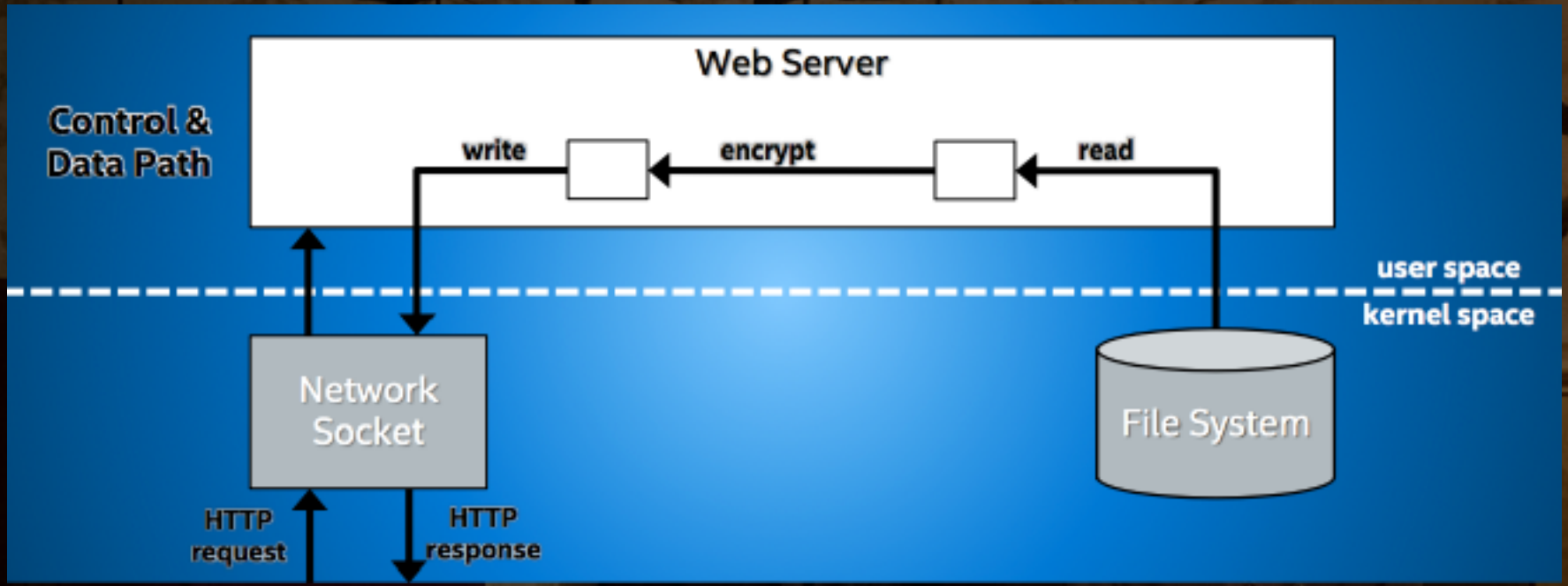
Implications of Implications

- have upstream write access via developers
- regular upstream syncs
- incompatible changes minimised

Sendfile System Call

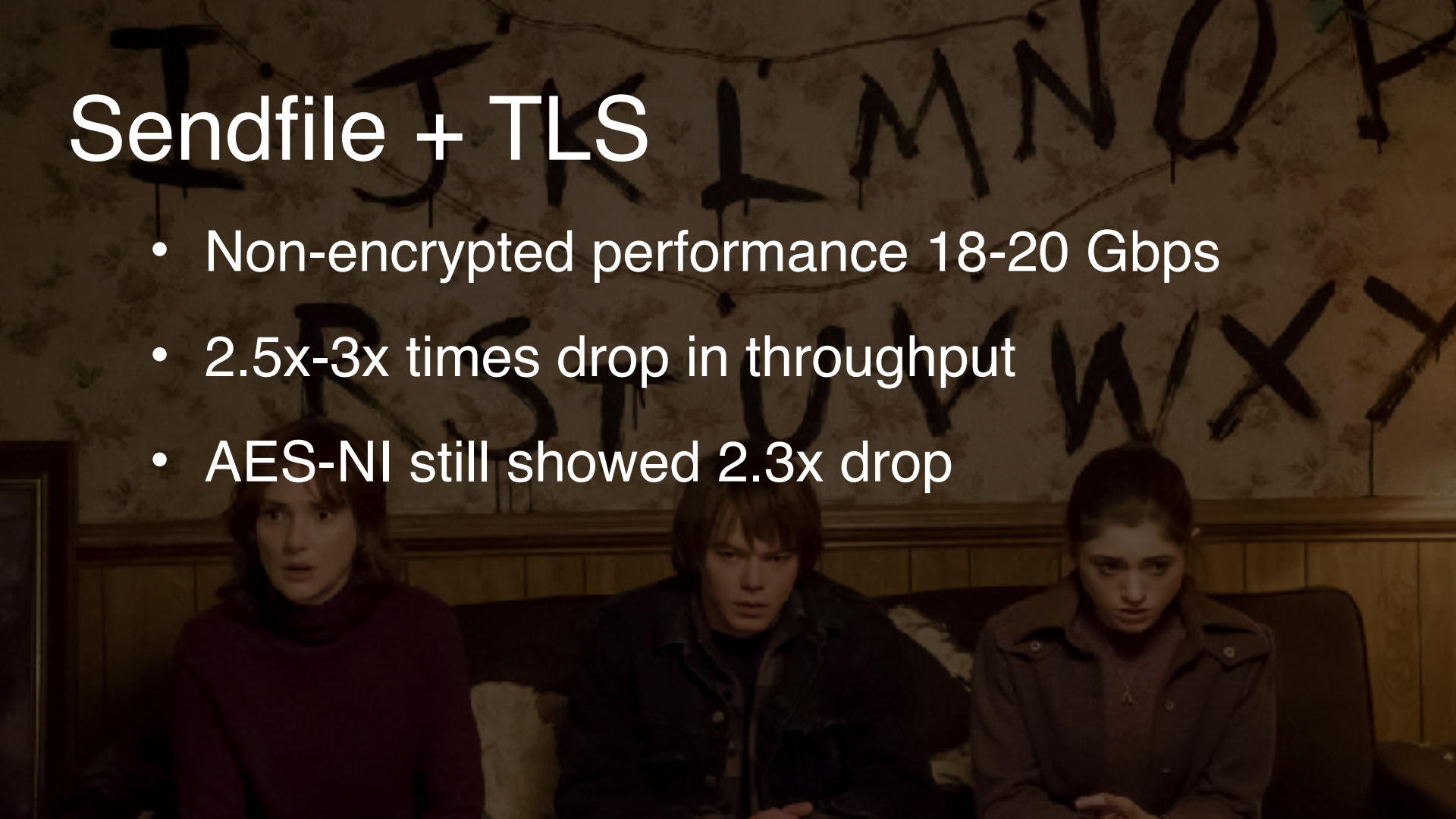


Sendfile + TLS

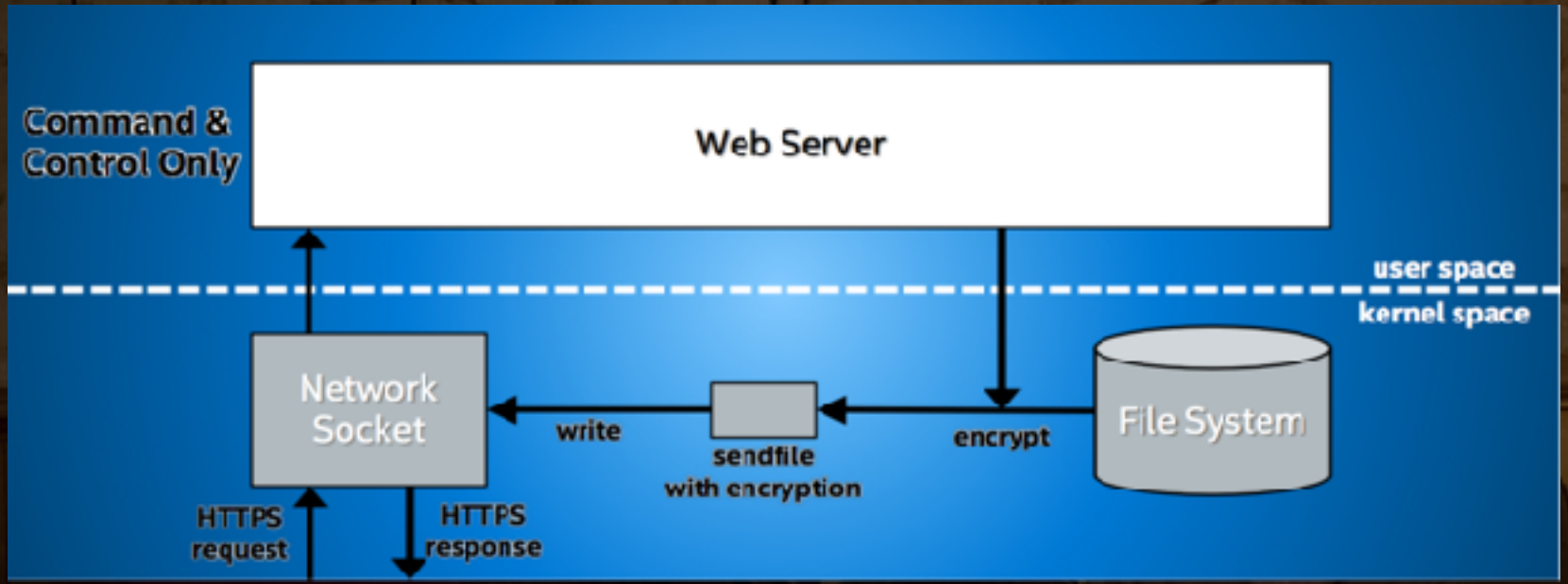


Sendfile + TLS

- Non-encrypted performance 18-20 Gbps
- 2.5x-3x times drop in throughput
- AES-NI still showed 2.3x drop



Sendfile + kernel TLS

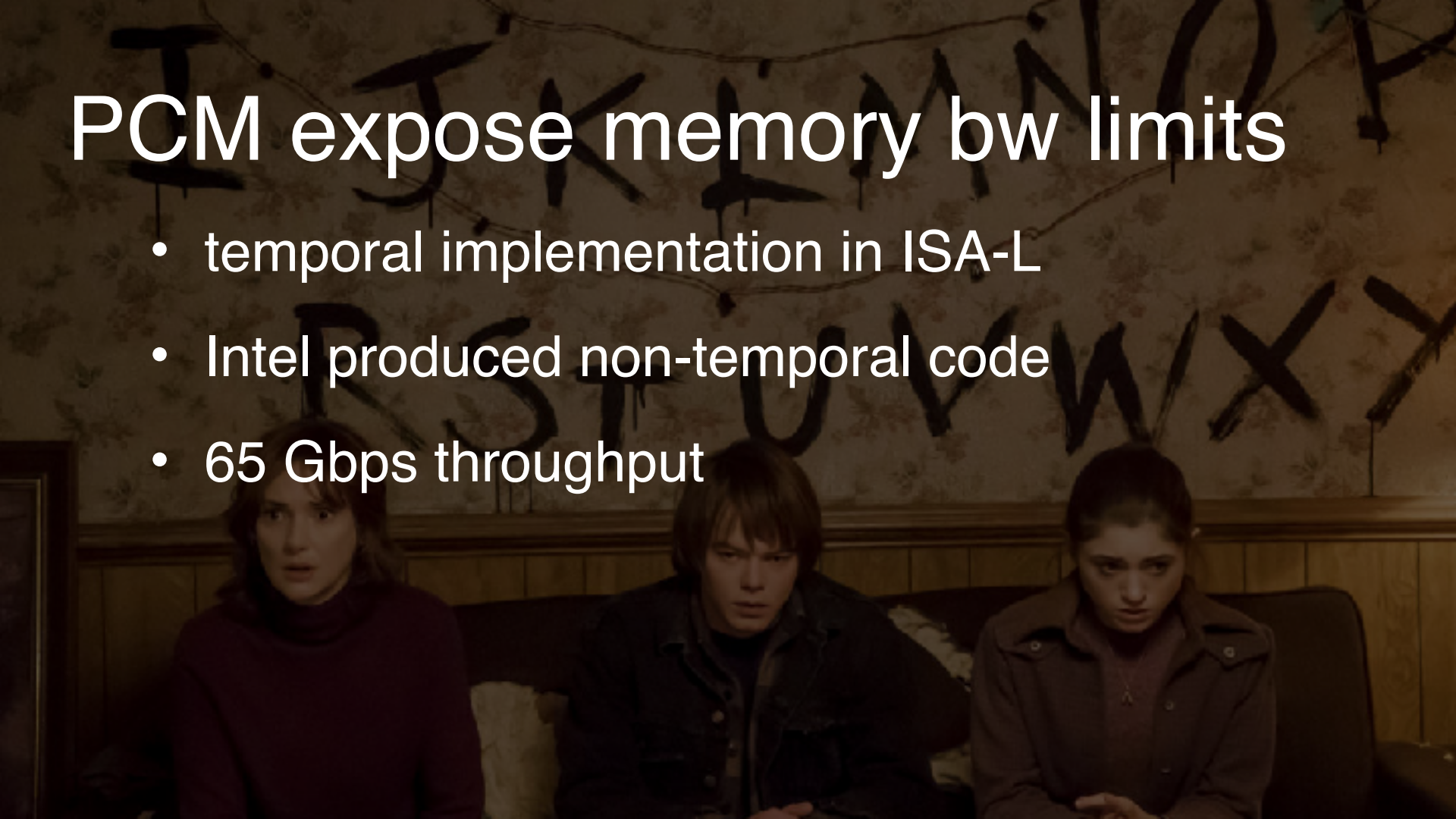


Step 1- Encrypted throughput

- Hardware - NVMe storage
- 18-20 Gbps with SSDs
- 58 Gbps with NVMe

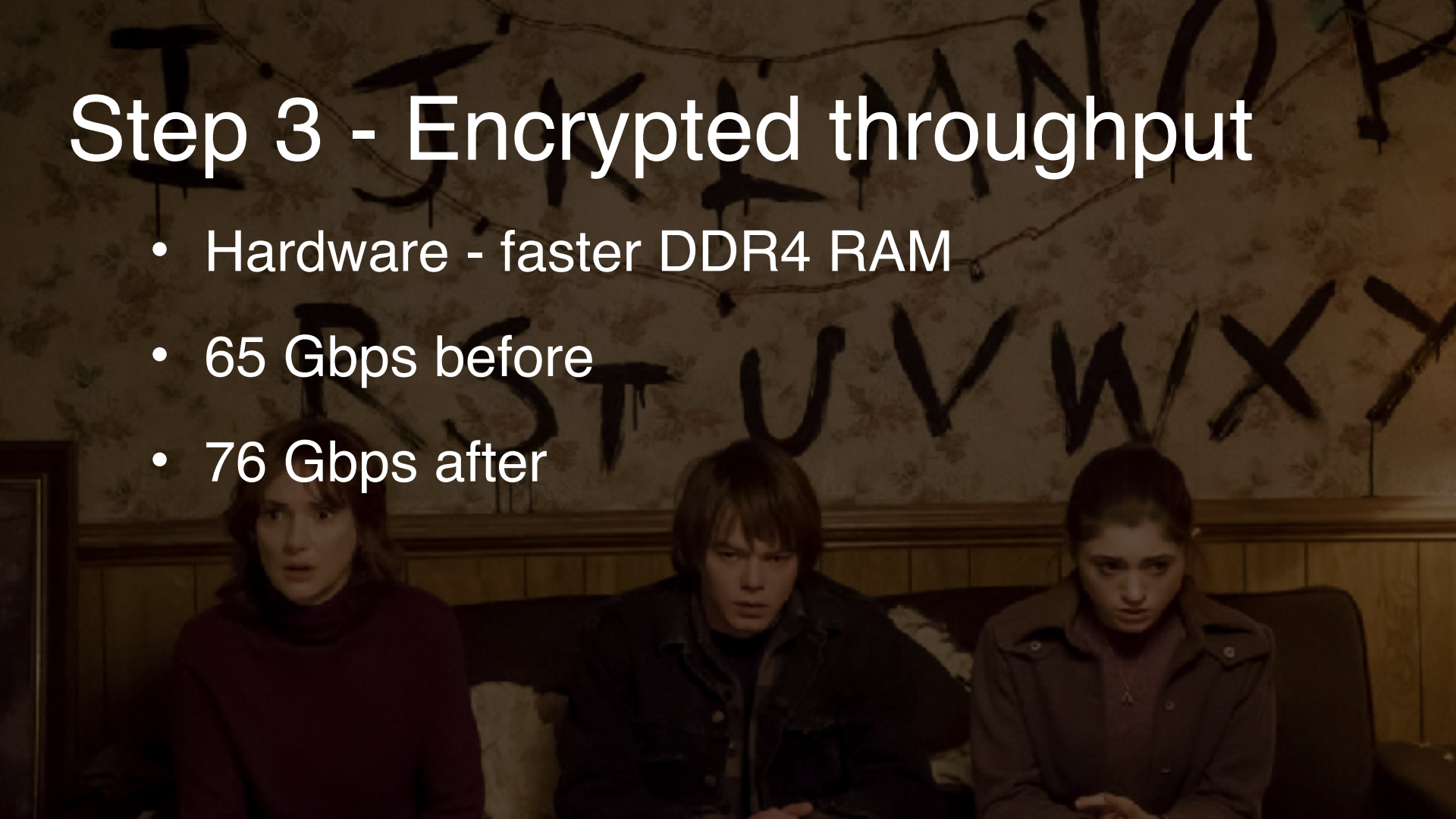
PCM expose memory bw limits

- temporal implementation in ISA-L
- Intel produced non-temporal code
- 65 Gbps throughput



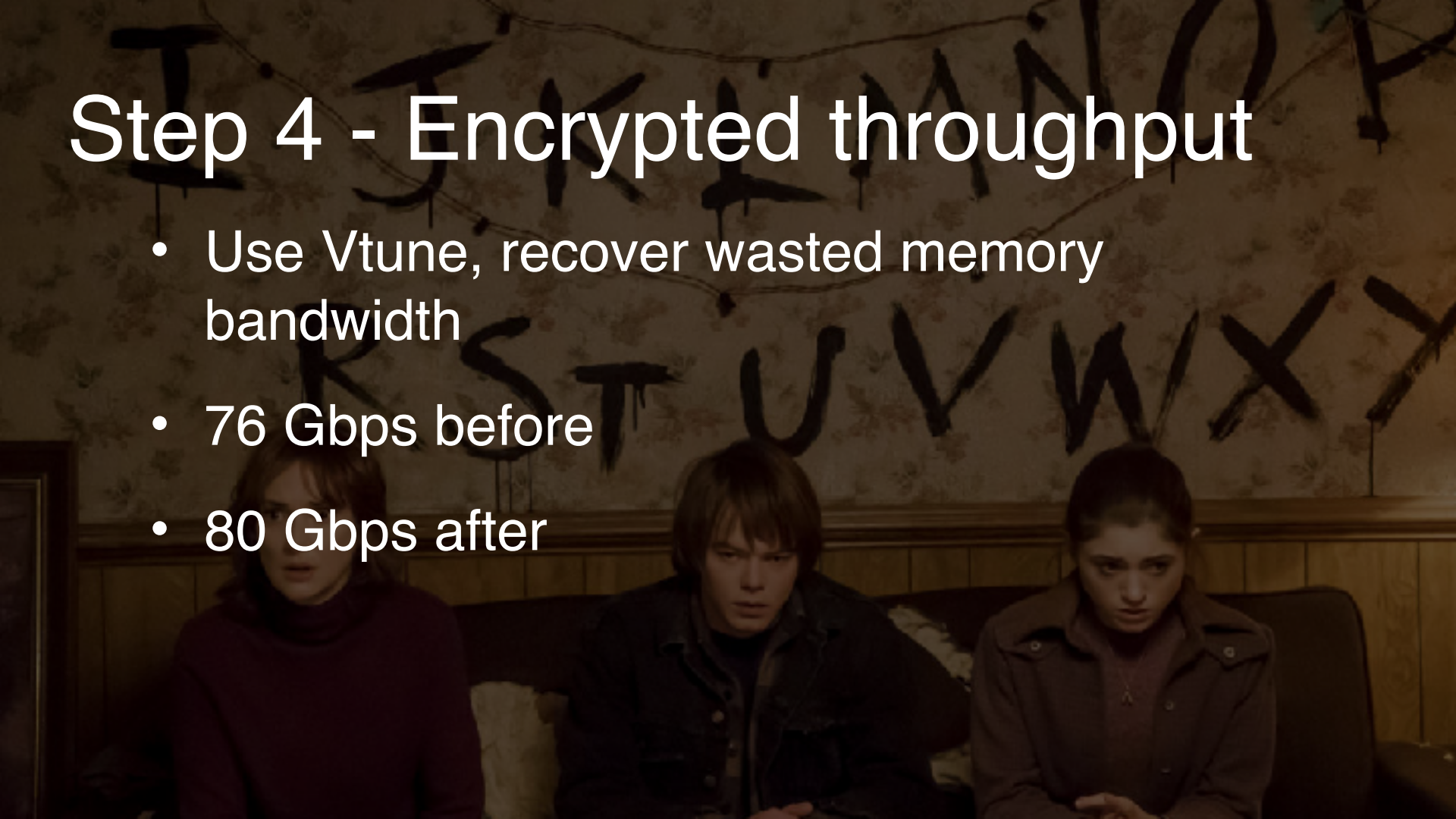
Step 3 - Encrypted throughput

- Hardware - faster DDR4 RAM
- 65 Gbps before
- 76 Gbps after

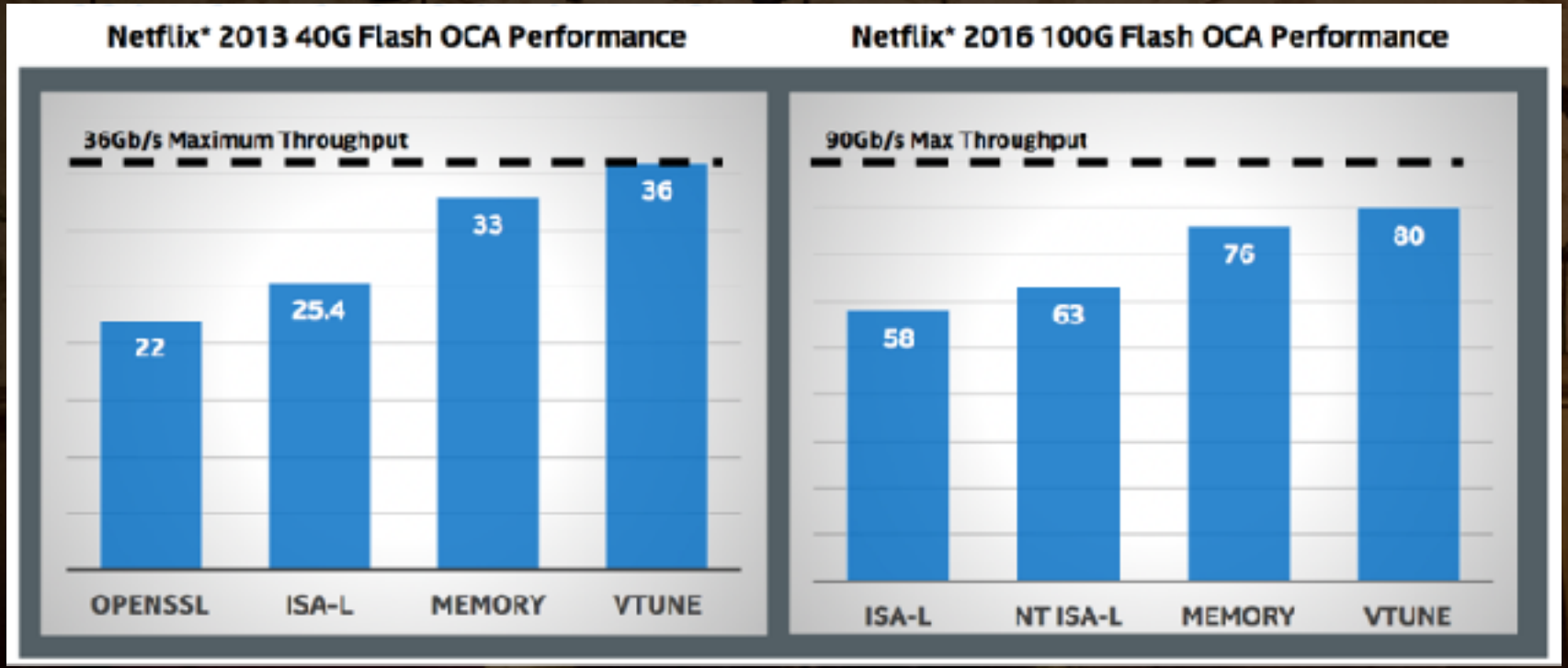


Step 4 - Encrypted throughput

- Use Vtune, recover wasted memory bandwidth
- 76 Gbps before
- 80 Gbps after

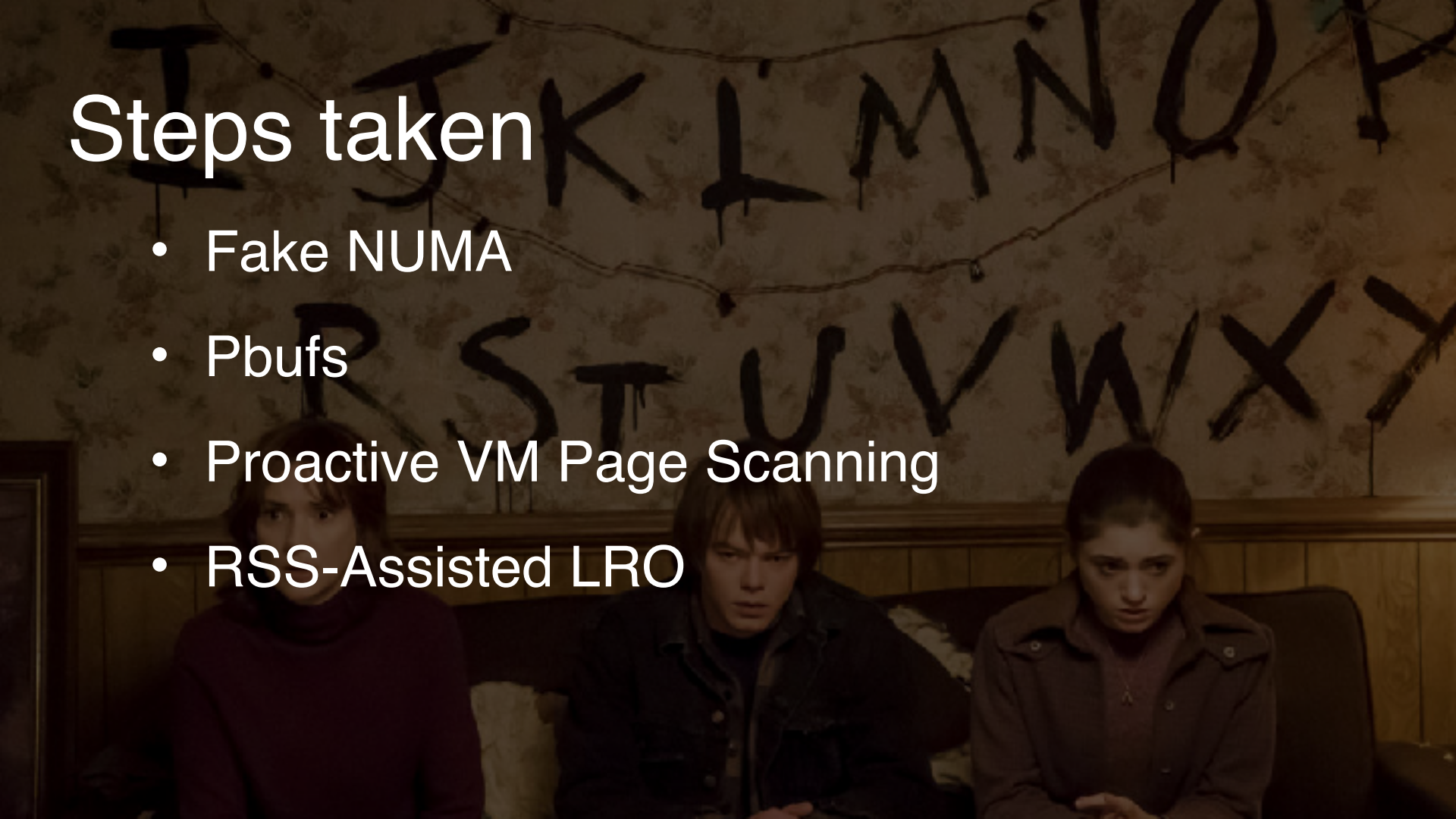


Sendfile + kernel TLS Performance



80 Gbps is so 2016...

- https://medium.com/netflix-techblog/serving-100-gbps-from-an-open-connect-appliance-cdb51dda3b99
- Blog post detailing steps to improve throughput



Steps taken

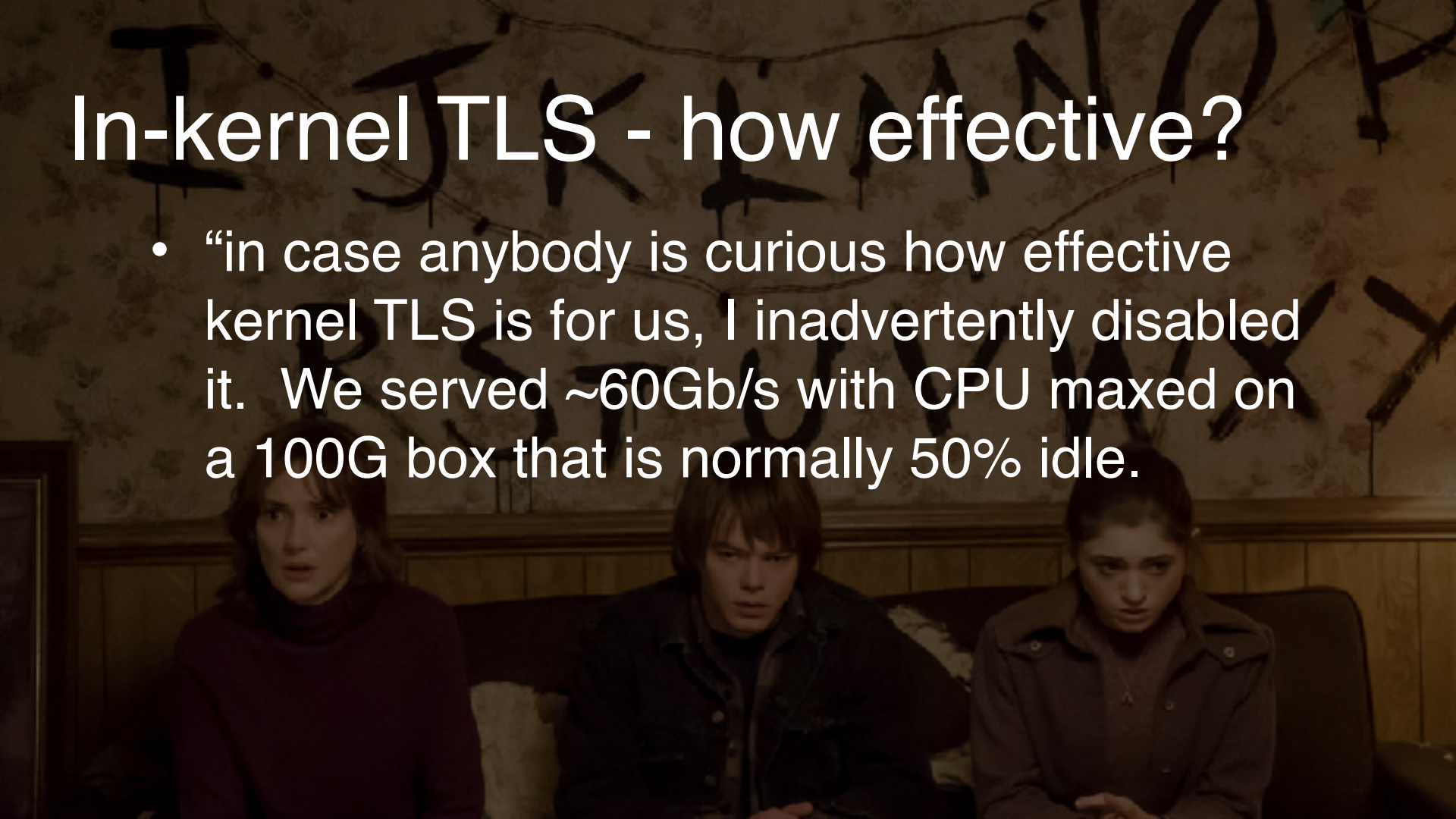
- Fake NUMA
- Pbufs
- Proactive VM Page Scanning
- RSS-Assisted LRO

TLS at 100 Gbps

- More VTune-driven optimisations - m_ext
- Getting out of our own way
- Mbuf page arrays
- “At this point, we’re able to serve 100% TLS traffic comfortably at 90 Gbps using the default FreeBSD TCP stack.”

In-kernel TLS - how effective?

- “in case anybody is curious how effective kernel TLS is for us, I inadvertently disabled it. We served ~60Gb/s with CPU maxed on a 100G box that is normally 50% idle.




Questions?