FreeBSD and the Foundation

2022 March SLUUG Meetup

Deb Goodkin – Executive Director, FreeBSD Foundation
Ed Maste - Sr. Director of Technology

Goals

- Explain the Foundation’s role and what we do
- Describe how the Project works
- Cover our current and future software development plans
- Answer any questions you may have

Upcoming Events

- BSDCan June 1-4, 2022 (virtual)
- 2022 Spring Developer Summit June 16-17, 2022 (virtual)

  https://wiki.freebsd.org/DevSummit/202206

- EuroBSDCon (Vienna, Austria) September 15th-18th, 2022

Who Am I?

- Joined the FreeBSD Foundation in August, 2005
- Technical background – 20 years in storage development as firmware engineer, logic designer, applications engineer, technical marketing and technical sales
- Enhancing my FreeBSD skills so I can teach others how to use and contribute to FreeBSD
- Runner and avid Pokemon Go player

Our History

Founded in March 2000

501(c)3 (non-profit public charity)

Based in Boulder, Colorado

100% Funded by donations

George Neville-Neil, Director

FreeBSD Foundation Board

Justin T. Gibbs
President and
Founder

Benedict
Reuschling,
Vice President

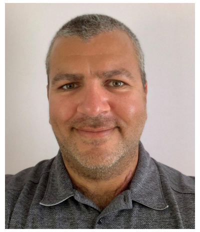
Dr. Marshall Kirk
McKusick, Treasurer

Dr. Robert N.M.
Watson, Director

Kevin Bowling,
Secretary

Dr. Hiroki Sato,
Director

Deb Goodkin
Assistant Secretary

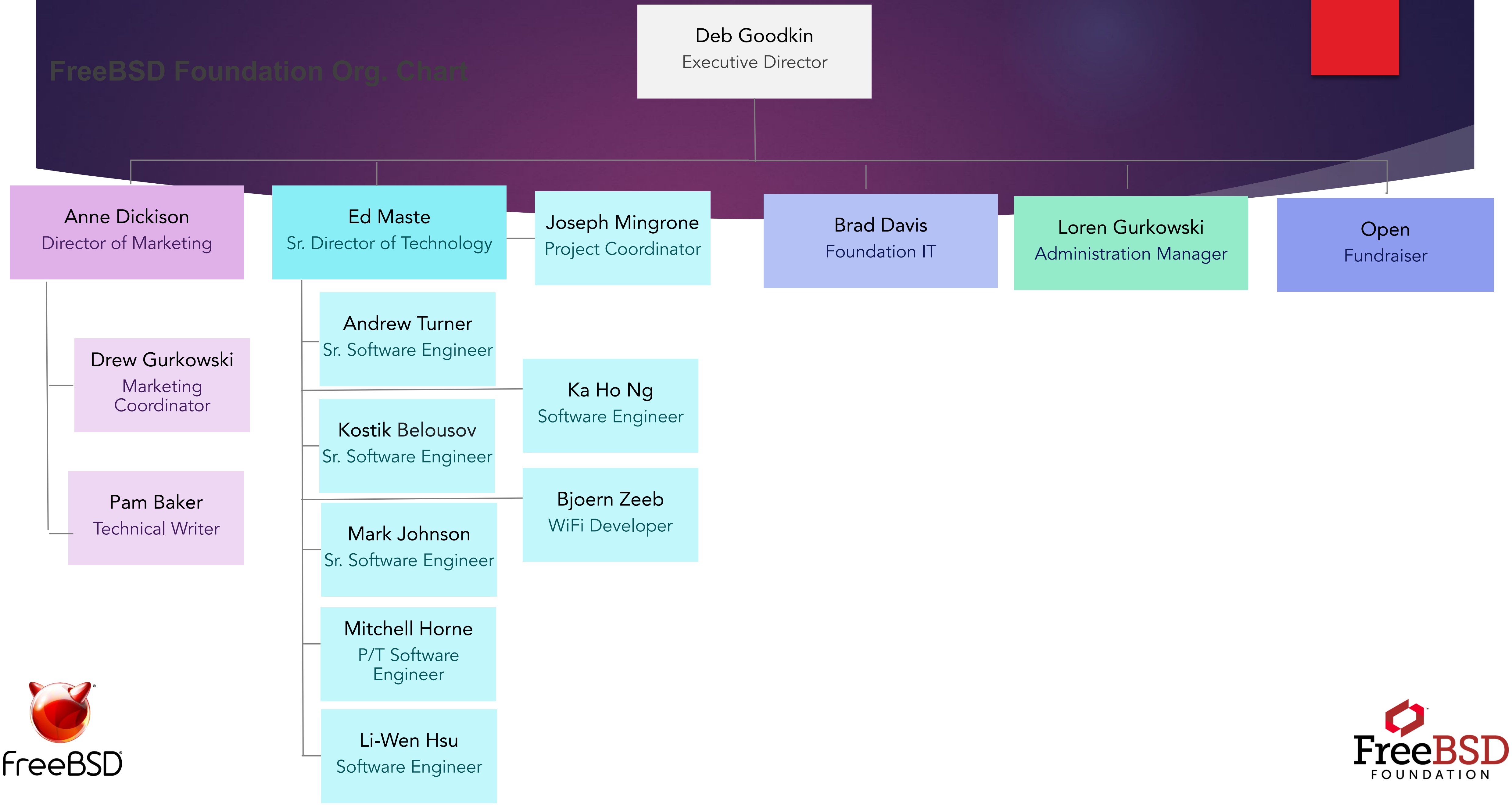


Andrew Wafaa,
Director

FreeBSD

FreeBSD
FOUNDATION

FreeBSD Foundation Org. Chart



- Deb Goodkin
  Executive Director
  - Anne Dickison
    Director of Marketing
    - Drew Gurkowski
      Marketing Coordinator
    - Pam Baker
      Technical Writer
  - Ed Maste
    Sr. Director of Technology
    - Joseph Mingrone
      Project Coordinator
    - Andrew Turner
      Sr. Software Engineer
    - Ka Ho Ng
      Software Engineer
    - Kostik Belousov
      Sr. Software Engineer
    - Bjoern Zeeb
      WiFi Developer
    - Mark Johnson
      Sr. Software Engineer
    - Mitchell Horne
      P/T Software Engineer
    - Li-Wen Hsu
      Software Engineer
  - Brad Davis
    Foundation IT
  - Loren Gurkowski
    Administration Manager
  - Open
    Fundraiser

FreeBSD

FreeBSD FOUNDATION

Our Purpose

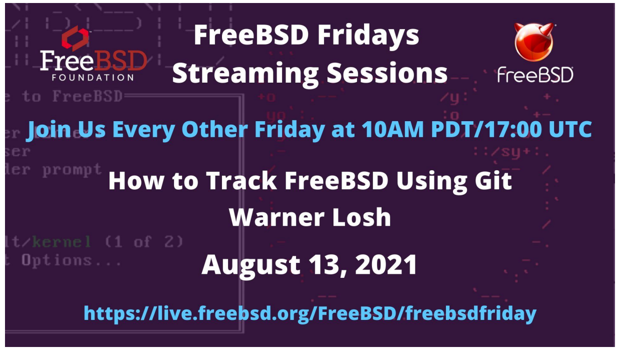
Make FreeBSD the best platform for research, products, computing, education, and more.

What We Support

```
Welcome to FreeBSD
1. Boot Multi User [Enter]
2. Boot Single User
3. Escape to loader prompt
4. Reboot

Options:
5. Configure Boot Options...
```

Software Development/OS Improvements



FreeBSD Fridays
Streaming Sessions

Join Us Every Other Friday at 10AM PDT/17:00 UTC

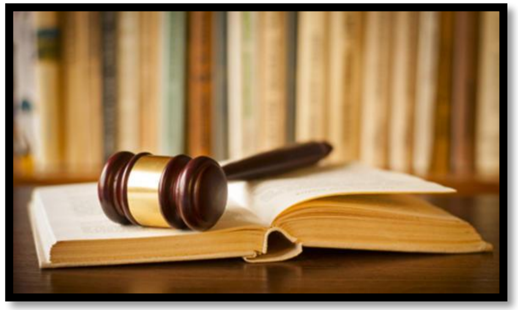
How to Track FreeBSD Using Git
Warner Losh

August 13, 2021

https://live.freebsd.org/FreeBSD/freebsdfriday

Advocacy and Education

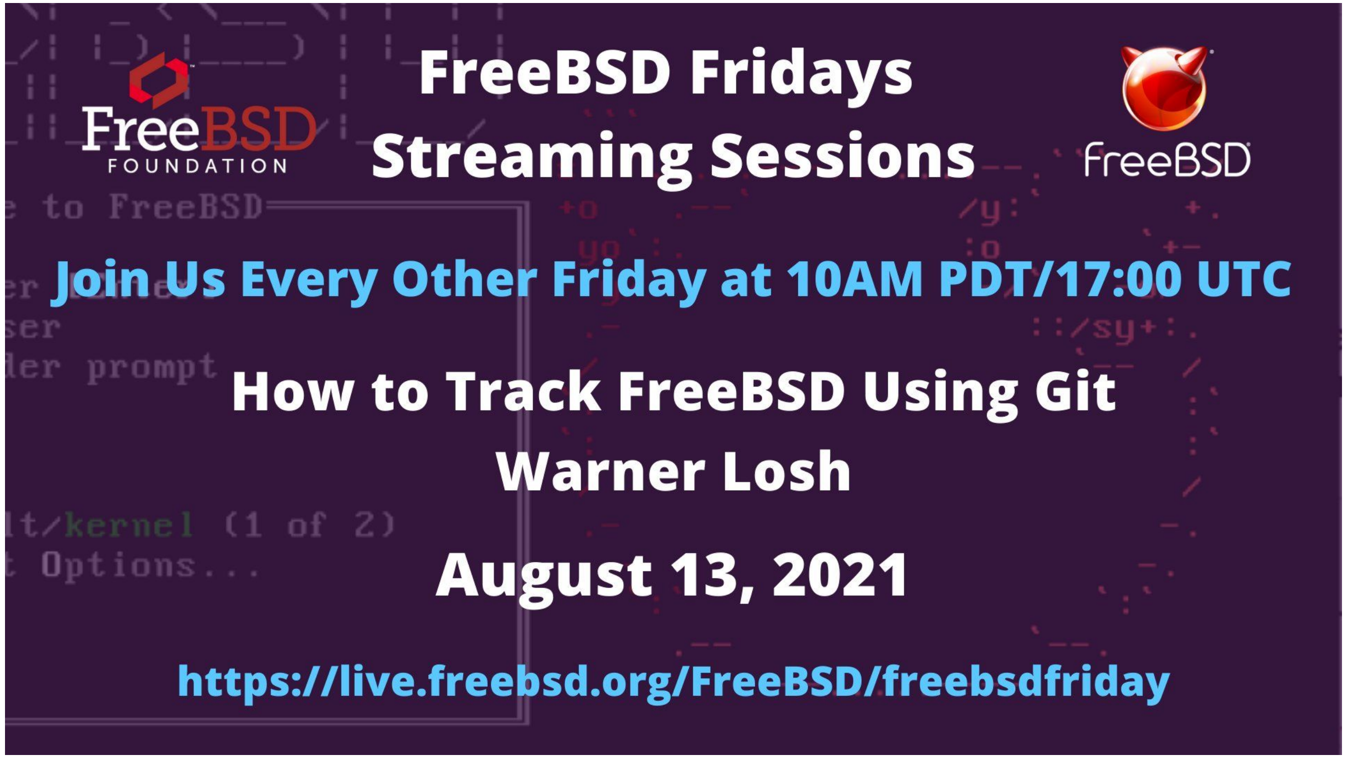
Security Team / Infrastructure / QA



Legal – Core Team and FreeBSD IP

Face-to-Face Meetings and Summits

Advocacy and Education



FreeBSD Fridays
Streaming Sessions

Join Us Every Other Friday at 10AM PDT/17:00 UTC

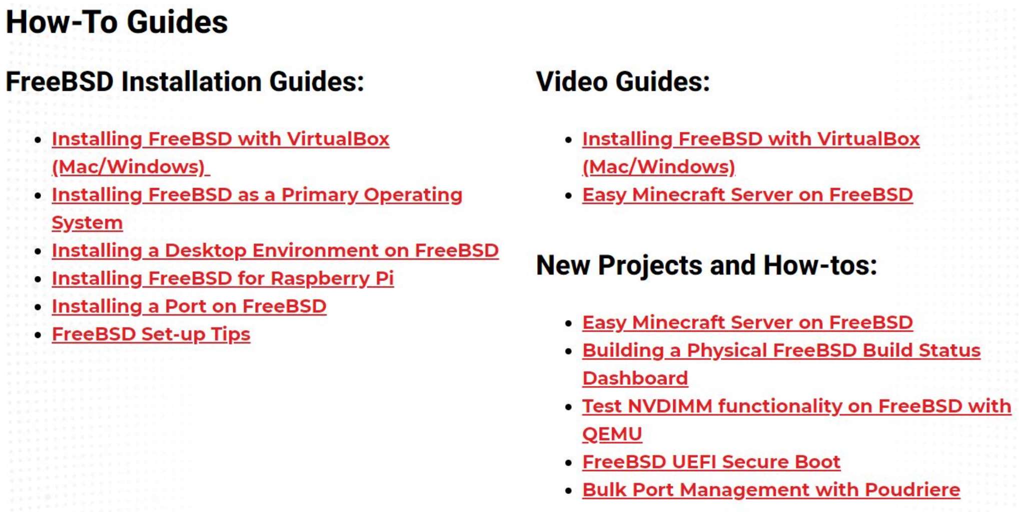
How to Track FreeBSD Using Git
Warner Losh

August 13, 2021

https://live.freebsd.org/FreeBSD/freebsdfriday

FreeBSD Advocacy/Education

- Promote FreeBSD through presentations and workshops around the world
- FreeBSD Journal
- FreeBSD Fridays Introductory Talk Series
- How-To guides and training material
- 2022 Spring Developer Summit
- FreeBSD Bootcamps
- College Level Operating Systems Curriculum



How-To Guides

FreeBSD Installation Guides:

- Installing FreeBSD with VirtualBox (Mac/Windows)
- Installing FreeBSD as a Primary Operating System
- Installing a Desktop Environment on FreeBSD
- Installing FreeBSD for Raspberry Pi
- Installing a Port on FreeBSD
- FreeBSD Set-up Tips

Video Guides:

- Installing FreeBSD with VirtualBox (Mac/Windows)
- Easy Minecraft Server on FreeBSD

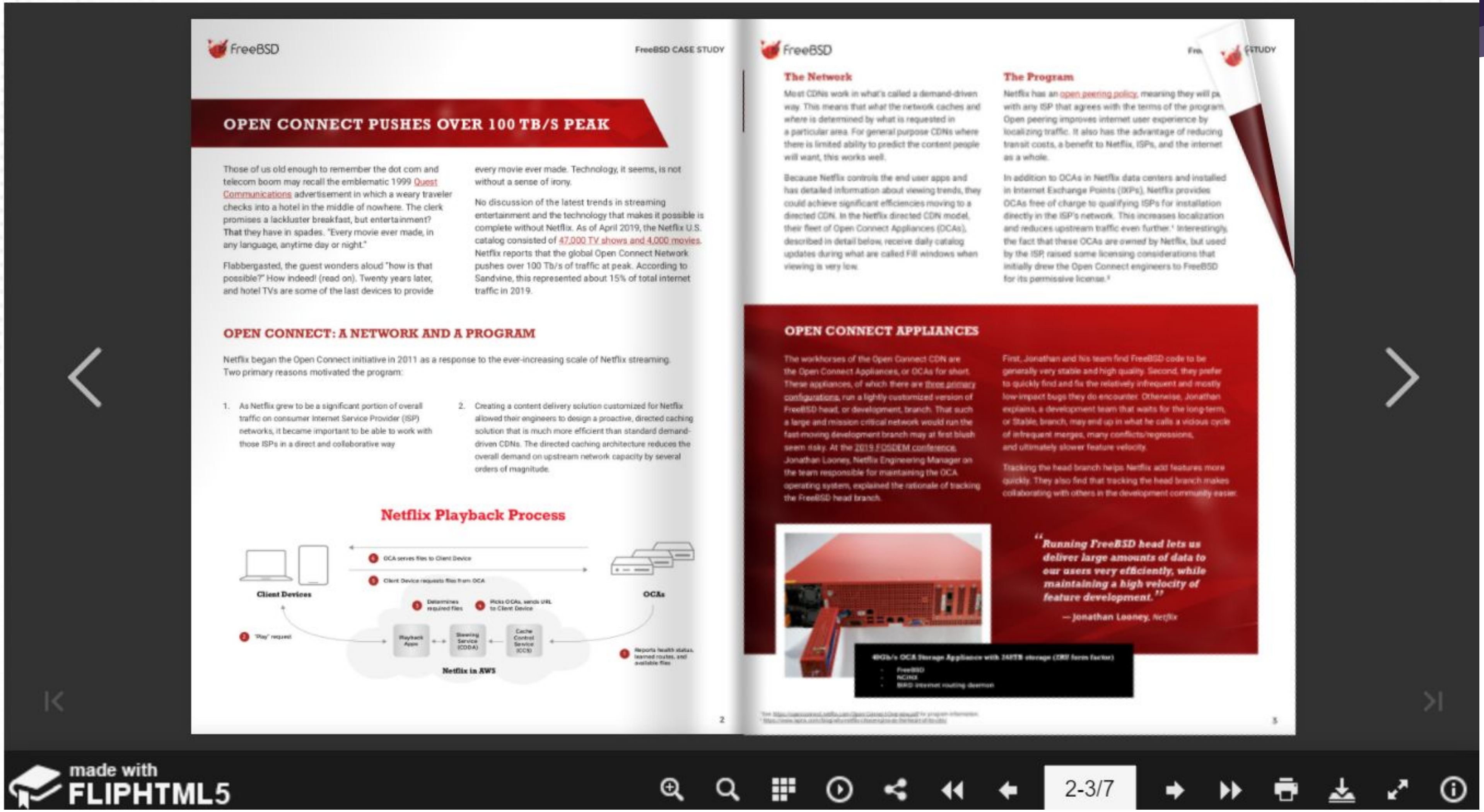
New Projects and How-tos:

- Easy Minecraft Server on FreeBSD
- Building a Physical FreeBSD Build Status Dashboard
- Test NVDIMM functionality on FreeBSD with QEMU
- FreeBSD UEFI Secure Boot
- Bulk Port Management with Poudriere

FreeBSD

FreeBSD FOUNDATION

FreeBSD Case Study: Netflix



OPEN CONNECT PUSHES OVER 100 TB/S PEAK

Those of us old enough to remember the dot com and telecom boom may recall the emblematic 1999 Quest Communications advertisement in which a weary traveler checks into a hotel in the middle of nowhere. The clerk promises a lackluster breakfast, but entertainment? That they have in spades. "Every movie ever made, in any language, anytime day or night."

Flabbergasted, the guest wonders aloud "how is that possible?" How indeed! (read on). Twenty years later, and hotel TVs are some of the last devices to provide every movie ever made. Technology, it seems, is not without a sense of irony.

No discussion of the latest trends in streaming entertainment and the technology that makes it possible is complete without Netflix. As of April 2019, the Netflix U.S. catalog consisted of 47,000 TV shows and 4,000 movies. Netflix reports that the global Open Connect Network pushes over 100 Tb/s of traffic at peak. According to Sandvine, this represented about 15% of total internet traffic in 2019.

OPEN CONNECT: A NETWORK AND A PROGRAM

Netflix began the Open Connect initiative in 2011 as a response to the ever-increasing scale of Netflix streaming. Two primary reasons motivated the program:

1. As Netflix grew to be a significant portion of overall traffic on consumer Internet Service Provider (ISP) networks, it became important to be able to work with those ISPs in a direct and collaborative way
2. Creating a content delivery solution customized for Netflix allowed their engineers to design a proactive, directed caching solution that is much more efficient than standard demand-driven CDNs. The directed caching architecture reduces the overall demand on upstream network capacity by several orders of magnitude.

Netflix Playback Process

6 OCA serves files to Client Device
5 Client Device requests files from OCA
Client Devices
OCAs
3 Determines required files
4 Picks OCAs, sends URL to Client Device
2 "Play" request
Playback Apps
Steering Service (CODA)
Cache Control Service (CCS)
1 Reports health status, learned routes, and available files
Netflix in AWS

The Network

Most CDNs work in what's called a demand-driven way. This means that what the network caches and where is determined by what is requested in a particular area. For general purpose CDNs where there is limited ability to predict the content people will want, this works well.

Because Netflix controls the end user apps and has detailed information about viewing trends, they could achieve significant efficiencies moving to a directed CDN. In the Netflix directed CDN model, their fleet of Open Connect Appliances (OCAs), described in detail below, receive daily catalog updates during what are called Fill windows when viewing is very low.

The Program

Netflix has an open peering policy, meaning they will pe with any ISP that agrees with the terms of the program. Open peering improves internet user experience by localizing traffic. It also has the advantage of reducing transit costs, a benefit to Netflix, ISPs, and the internet as a whole.

In addition to OCAs in Netflix data centers and installed in Internet Exchange Points (IXPs), Netflix provides OCAs free of charge to qualifying ISPs for installation directly in the ISP's network. This increases localization and reduces upstream traffic even further.[1] Interestingly, the fact that these OCAs are owned by Netflix, but used by the ISP, raised some licensing considerations that initially drew the Open Connect engineers to FreeBSD for its permissive license.[2]

OPEN CONNECT APPLIANCES

The workhorses of the Open Connect CDN are the Open Connect Appliances, or OCAs for short. These appliances, of which there are three primary configurations, run a lightly customized version of FreeBSD head, or development, branch. That such a large and mission critical network would run the fast-moving development branch may at first blush seem risky. At the 2019 FOSDEM conference, Jonathan Looney, Netflix Engineering Manager on the team responsible for maintaining the OCA operating system, explained the rationale of tracking the FreeBSD head branch.

First, Jonathan and his team find FreeBSD code to be generally very stable and high quality. Second, they prefer to quickly find and fix the relatively infrequent and mostly low-impact bugs they do encounter. Otherwise, Jonathan explains, a development team that waits for the long-term, or Stable, branch, may end up in what he calls a vicious cycle of infrequent merges, many conflicts/regressions, and ultimately slower feature velocity.

Tracking the head branch helps Netflix add features more quickly. They also find that tracking the head branch makes collaborating with others in the development community easier.

> "Running FreeBSD head lets us deliver large amounts of data to our users very efficiently, while maintaining a high velocity of feature development."
>
> — Jonathan Looney, Netflix

40Gb/s OCA Storage Appliance with 248TB storage (2RU form factor)
- FreeBSD
- NGINX
- BIRD internet routing daemon

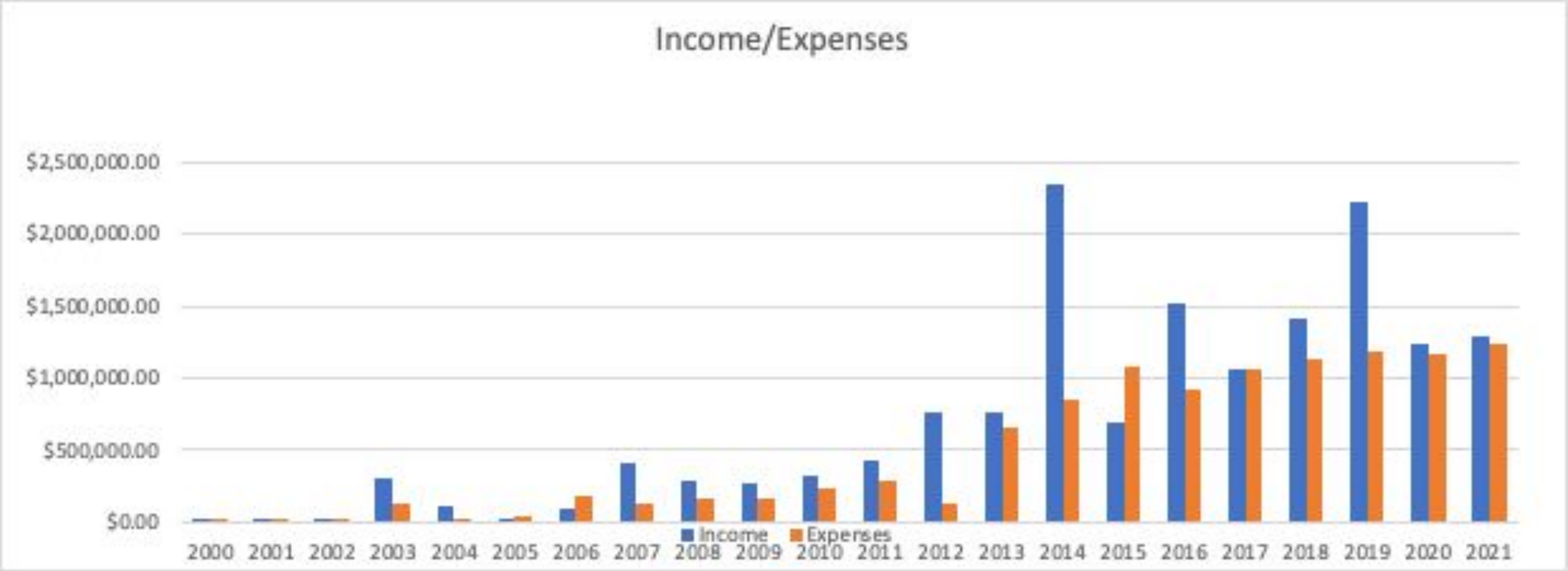
[1] See [illegible] for program information.
[2] [illegible]

Security Team / Infrastructure / QA

Legal – Core Team and FreeBSD IP

Face-to-Face Meetings and Summits
(mostly virtual in 2020 and 2021)

Fundraising



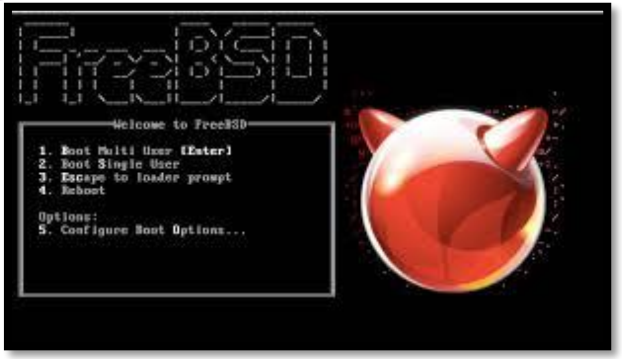
2021
Raised $235k
Budget $2M

More than half the budget goes to software development

How the Project Works

The FreeBSD World

FreeBSD is an open source Unix-like **operating system** descended from the Unix developed at the University of California, Berkeley in the 1970s.



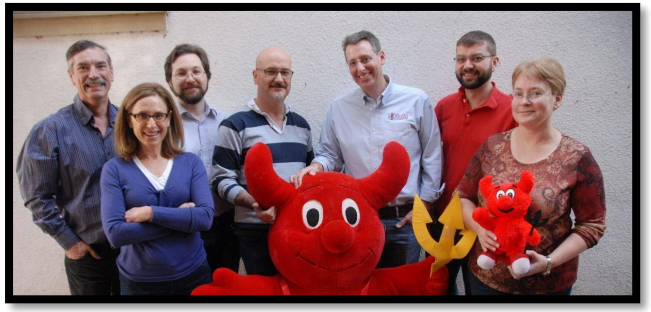
```
Welcome to FreeBSD

1. Boot Multi User [Enter]
2. Boot Single User
3. Escape to loader prompt
4. Reboot

Options:
5. Configure Boot Options...
```

The FreeBSD Project is an active open source community since 1993 with hundreds of committers and thousands of contributors around the world.

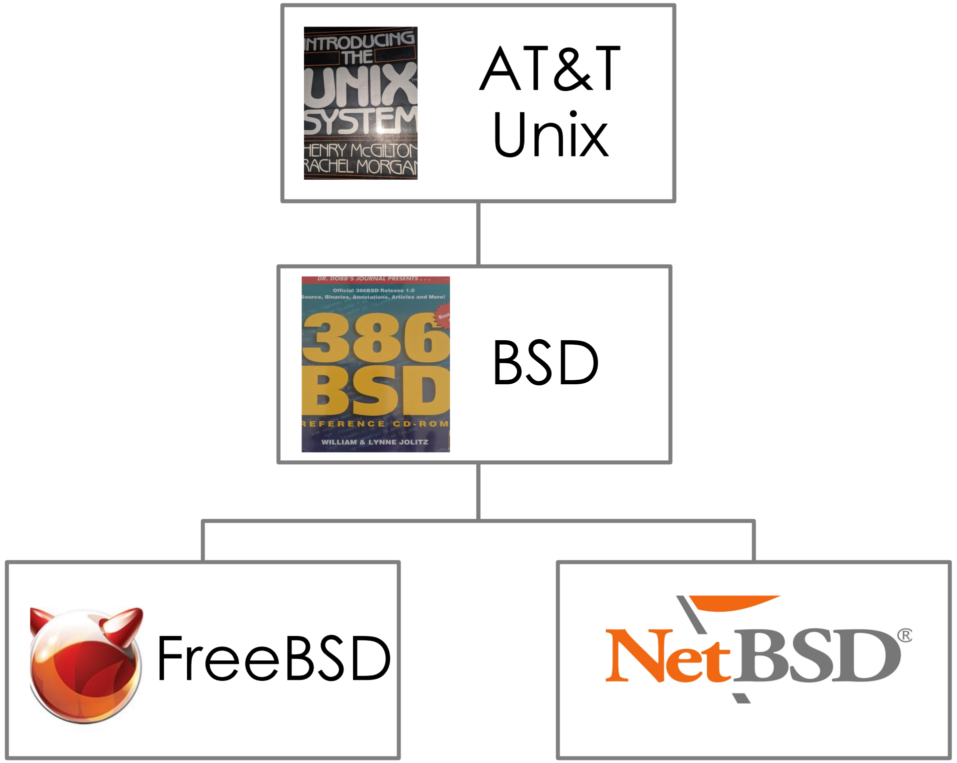
The FreeBSD Foundation is a 501(c)3 **non-profit organization** registered in Colorado, USA in 2000 dedicated to supporting the FreeBSD Project, its development and its community.



What is FreeBSD?

- Free and Open Source Computer Operating System
- Complete operating system including kernel, userland, documentation, and tools
- Descended from Berkeley Unix a descendent of the original Unix
- Used by universities, corporations, and users for over 28 years!

Abridged BSD Family Tree



AT&T Unix

BSD

FreeBSD

NetBSD

UNIX

1969

In 1969 Ken Thompson, Dennis Ritchie and others started working on a program that utilized the full capabilities of new powerful computer systems. This program was called Unix.

BSD

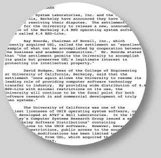
1974

In 1974 The Computer Systems Research Group at UC Berkeley started to modify and improve AT&T Research Unix. They called this modified version "Berkeley Unix" or "BSD".

386/BSD

1992

386/BSD was released in 1992. This was the first freely redistributable full BSD operating system with 100% unencumbered files.



BSDi Lawsuit

1992

BSDi found itself in legal trouble with AT&T's Unix System Laboratories, then the owners of the System V copyright, and the Unix trademark. The USL v. BSDi lawsuit was filed in 1992.

FreeBSD

1993

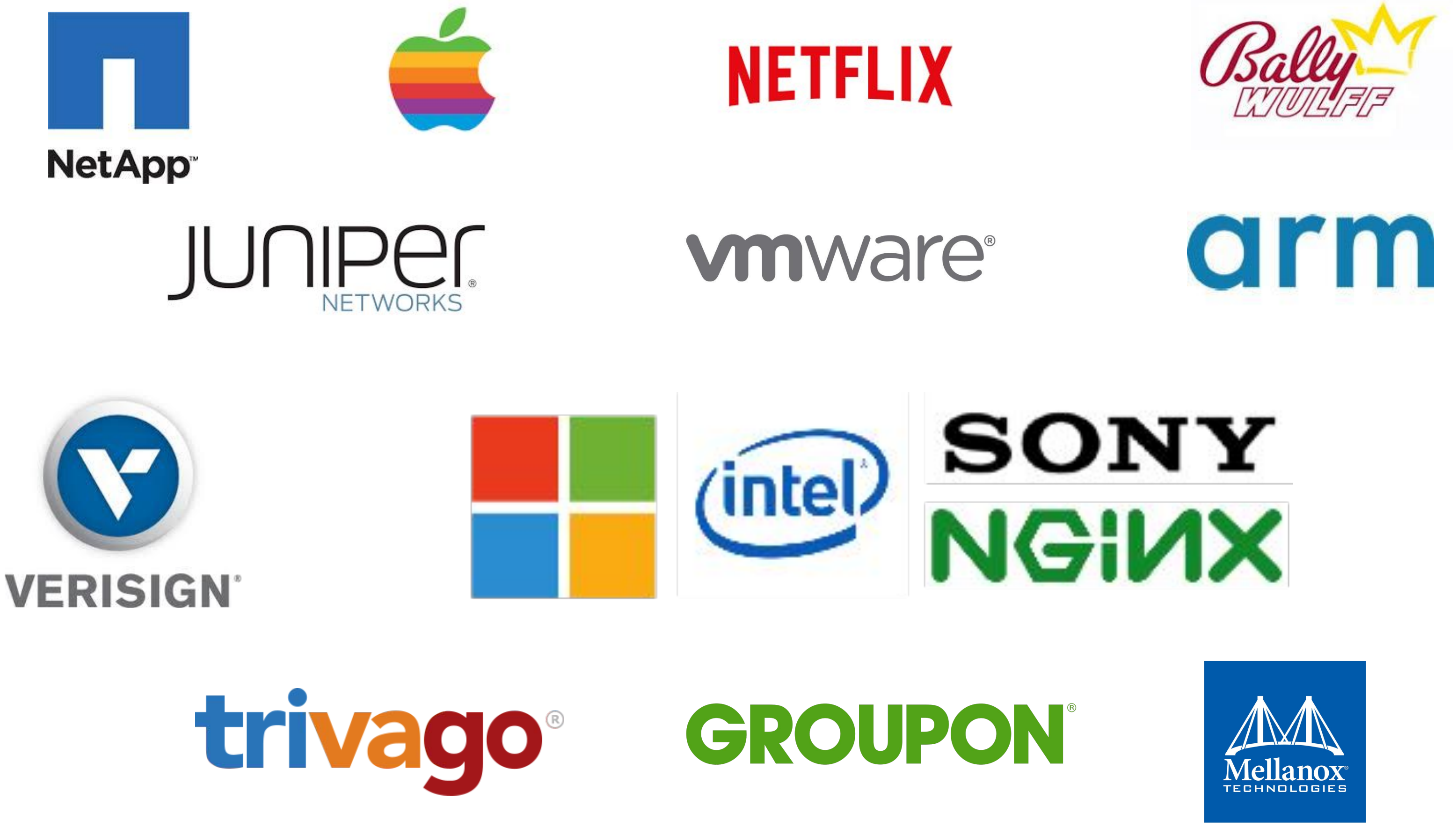
The development flow of 386BSD was slow and after a period of neglect, a group of 386BSD users decided to branch out on their own and create FreeBSD so that they could keep the operating system up to date. On 19 June 1993, the name FreeBSD was chosen for the project.

2022

28 YEARS OF INNOVATION AND GROWTH

https://www.freebsdfoundation.org/freebsd/timeline/

Who Uses FreeBSD



Most Likely You Use FreeBSD!

- iPhone or Apple computer
- Streaming Netflix
- Planning your next vacation
- Sony PlayStation 4&5

Getting an awesome deal!

Why Use FreeBSD?

- Friendly and Approachable Community
- Excellent Documentation
- Good Tooling and Modern Compilers
- Consistent Development and Release Processes

- Wide Variety of Architectures Supported
- 2-clause BSD license - Does not restrict what you can do with your own code!
- Secure, Stable, and Reliable

Why Companies Use FreeBSD?

- History of innovation
- High performance
- Great tools
- ABI stability within major releases – Remember POLA
- Mature release model
- Excellent documentation
- Business Friendly License
- ZFS
- Open community
- Smaller footprint than most operating systems

"We choose FreeBSD for many of our internal services and product service offerings because we know we can rely on its consistent reliability and performance. Its portability not only allows us to run it on almost any commodity or enterprise server, but allows for the possibility to move a hard drive from one server to another, boot, and get back to normal operation with minimal fuss."

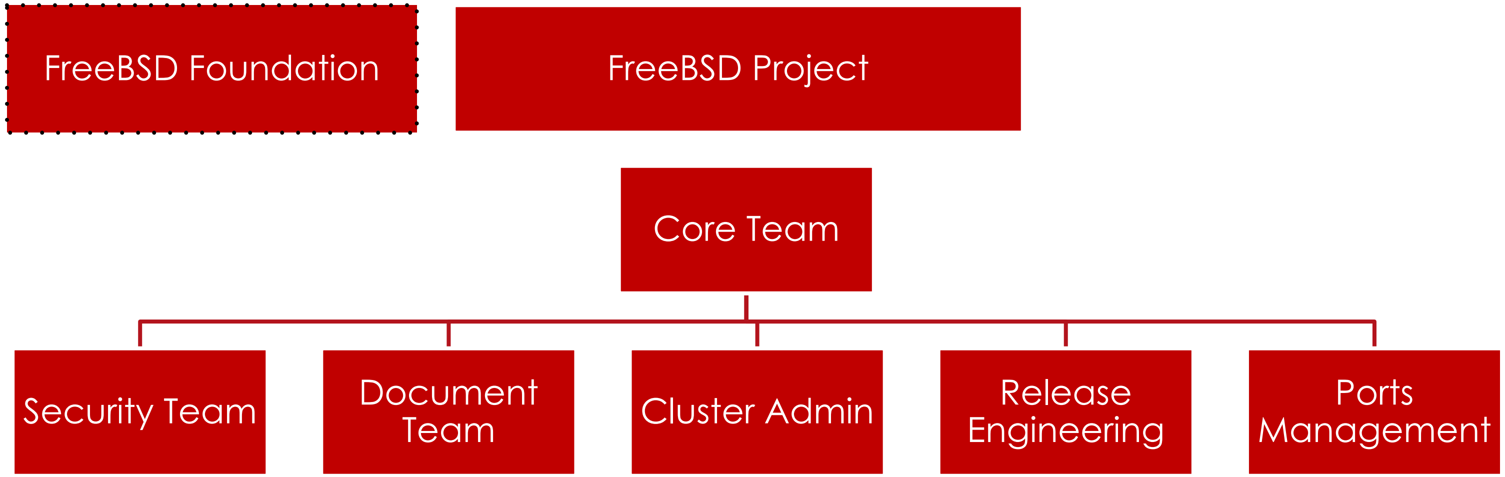
FreeBSD Project Goal

Provide a stable and fast general purpose operating system that may be used for any purpose without strings attached.

FreeBSD Project Model

- FreeBSD followed the model set up at Berkeley, improving over the years.
- Thousands of contributors/developers who maintain, write documentation, and make improvements who can submit changes and improvements as PRs or through committers.
- Hundreds of committers who can submit changes and improvements to the source tree
- Nine member elected core team who governs and leads the Project.
- Strong mentorship culture, where a committer will mentor a new contributor
- We have no "benevolent" dictators for life, meaning anyone can make a huge impact.

FreeBSD Project Org Chart


FreeBSD Foundation

FreeBSD Project

Core Team

Security Team

Document Team

Cluster Admin

Release Engineering

Ports Management

Other Teams include:

- Ports Secteam
- Security Officer
- Bugmeisters

- Continuous Integration Testing Admins
- Postmaster Team
- Webmaster Team
- Phabricator Code Review Administration

- Core Team - 9
- Committers - ~400
- Contributors - Thousands

We need your help!

FreeBSD Core Team

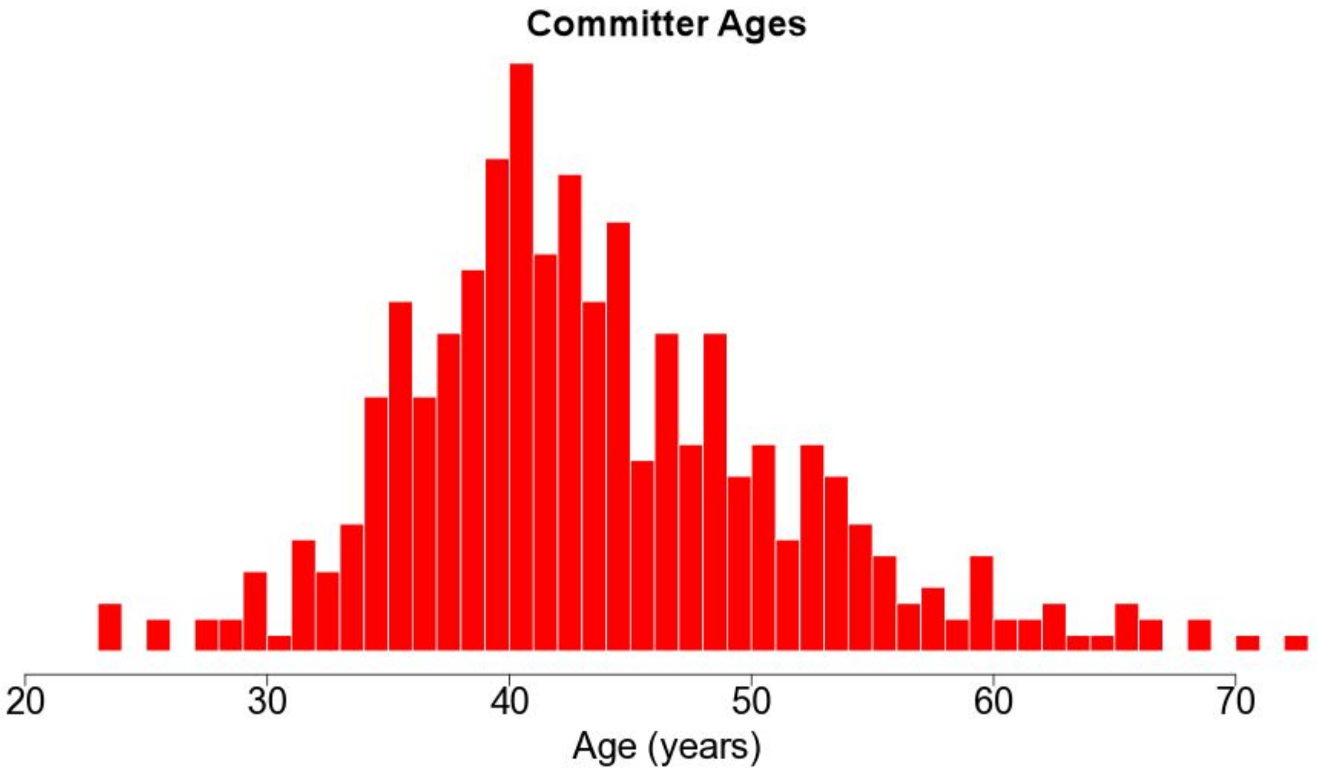
9-member elected management body

- Elections held every two years
- Active committers vote for core members
- Non-voting core team secretary is selected by the core team

- Responsibilities
  - Administrative (commit bits, hats, team charters)
  - Strategic (project direction, coordination, cajoling)
  - Rules, conflict resolution, enforcement

Who are the FreeBSD committers

- Locations
  - 34 countries
  - 6 continents
- Ages
  - Oldest: 73 (born in 1948)
  - Youngest: 23 (born in 1998)
  - Average: 44
  - Data* from circa Nov 2021



FreeBSD Releases

FreeBSD Operates on the Principle Of Least Astonishment (POLA):
Don’t break things that work!
Upgrades are generally painless even across major releases.

Two Types of Releases:

Major Release
(Dot Release) –13.0
Around every two years
(supported for 5 years)

Point Release – 12.2
Around every 9 months –
ABI/API compatibility

Two Types of Branches:

Current – Head
All changes to base system committed here. Dot releases built from here.

Stable –
After testing, most changes in current moved here. Point releases built from stable.

Weekly snapshots available for current and stable branches

How to Contribute to FreeBSD

- Code, writing documentation, maintaining ports, and advocacy.
- Easy to get started contributing.

https://www.freebsd.org/projects/newbies/

Some Suggestions:

- Start by translating or improving our documentation
- Pick one of the many ports to maintain or add
- Go through the PR list and fix some bugs

Check out FreeBSD Fridays and our How-To Guides for more getting started information

https://freebsdfoundation.org/freebsd-fridays/

https://freebsdfoundation.org/freebsd-project/resources/

FreeBSD Deep-Dive

Keeping Current

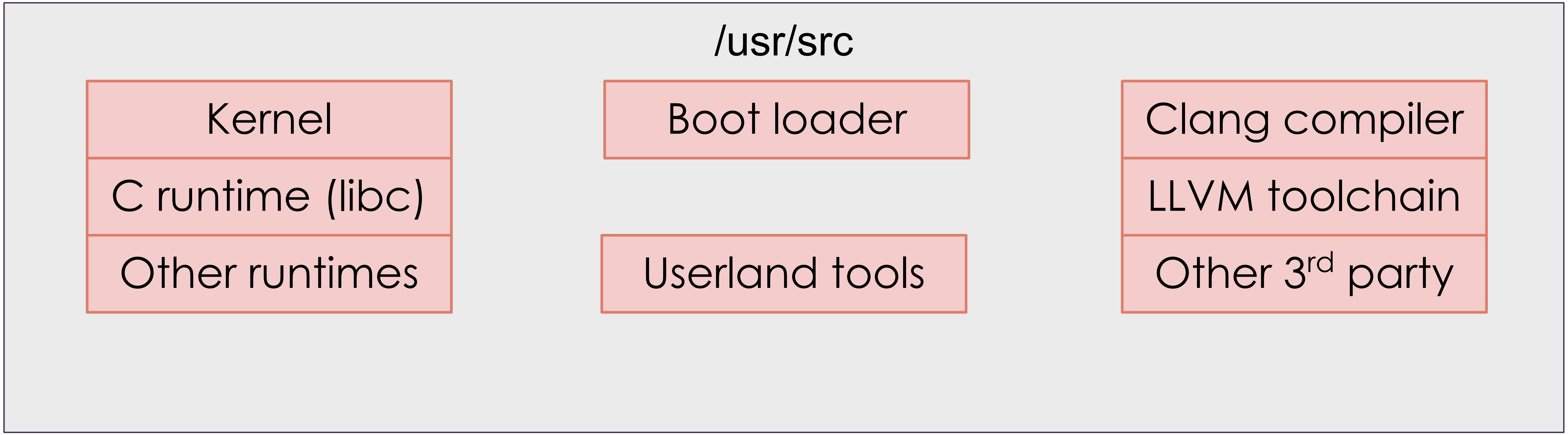
- svn to git migration - src, doc, ports
- FreeBSD doc tree migration docbook/sgml to asciidoc
- 40,000+ Binary packages - package manager redesign in 2010
- Merged our code into the OpenZFS repository! Now a variety of improvements and features will be available to FreeBSD.
- Improving desktop experience – wifi, graphics, latest hardware support, support obs, Audacity, video conferencing webapps
- Removed obsolete GCC 4.2.1, binutils 2.17.50, gdb 6.1

https://freebsdfoundation.org/blog/project-update-toolchain-modernization/

Desktop Distributions and Derivatives

- MidnightBSD
- GhostBSD
- NomadBSD
- helloSystem
- airyxOS

FreeBSD Project Base System



/usr/src

Kernel

C runtime (libc)

Other runtimes

Boot loader

Userland tools

Clang compiler

LLVM toolchain

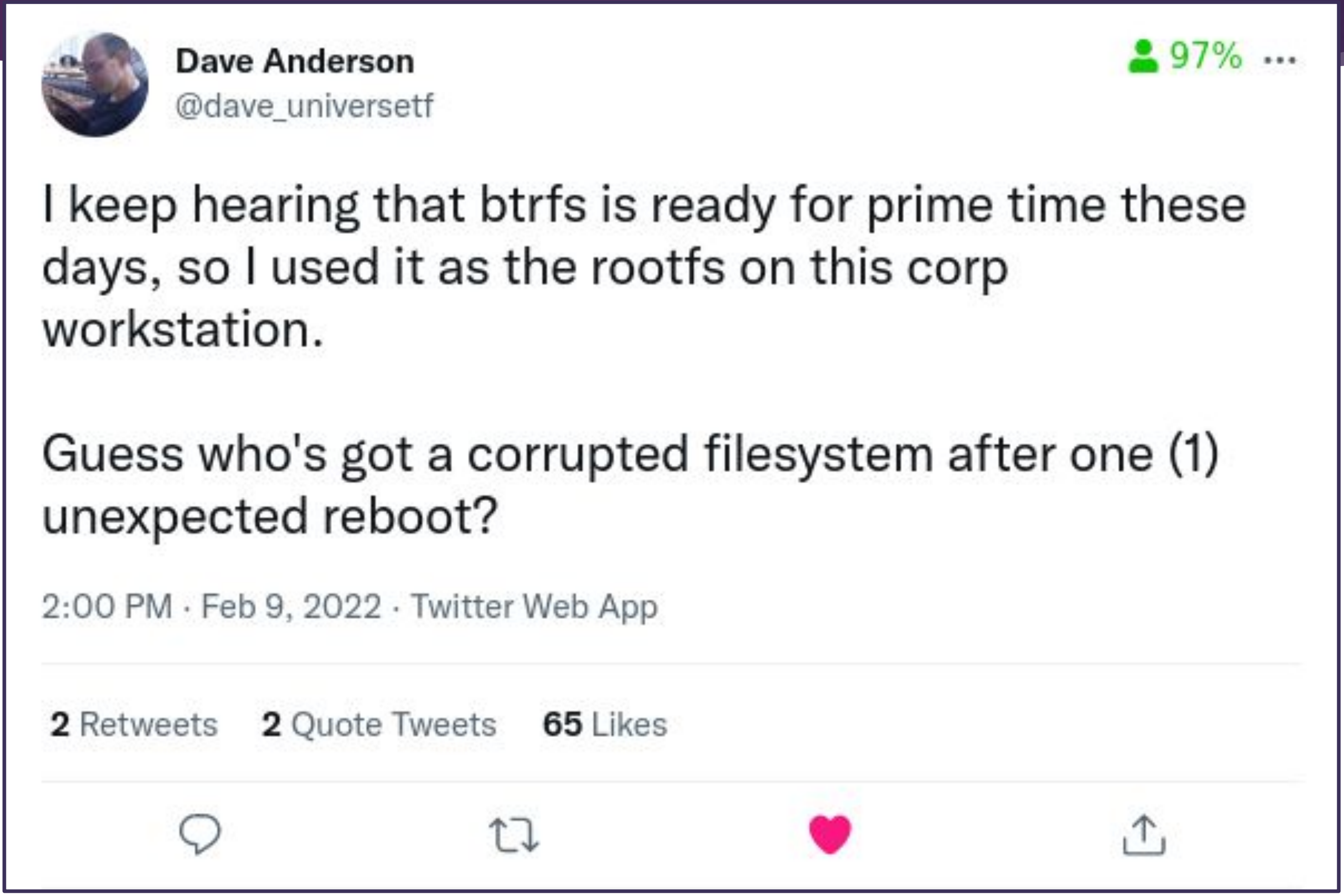
Other 3rd party

FreeBSD Features

- **Robust file systems** including UFS and ZFS (Active work happening on ZFS)
- **DTrace** - an advanced event-based performance analysis and troubleshooting tool. DTrace can help you identify and quantify the root cause of virtually any performance issue, in both user-level and kernel code. It can be executed using custom and powerful one-liners and scripts.
- **Jails** – Lightweight virtualization added to FreeBSD in the early 2000s.

- **bhyve** – Full-blown hypervisor. This hypervisor supports a number of guests, including FreeBSD, OpenBSD, Microsoft Windows, and many Linux distributions.
- **TCP/IP** was originally developed on BSD and FreeBSD remains the reference implementation for several network protocols.
- **Capsicum** – Capsicum is a lightweight OS capability and sandbox framework developed at the University of Cambridge Computer Laboratory. Capsicum extends the POSIX API, providing several new OS primitives **to support object-capability security** on UNIX-like operating systems

OpenZFS



Dave Anderson
@dave_universetf

I keep hearing that btrfs is ready for prime time these days, so I used it as the rootfs on this corp workstation.

Guess who's got a corrupted filesystem after one (1) unexpected reboot?

2:00 PM · Feb 9, 2022 · Twitter Web App

2 Retweets 2 Quote Tweets 65 Likes

Containers

- FreeBSD pioneered containers with **Jails**
- **Linuxulator -** provides binary compatibility with Linux®
- **bhyve** – Full-blown hypervisor. This hypervisor supports a number of guests, including FreeBSD, OpenBSD, Microsoft Windows, and many Linux distributions.
- **Pot** – Another container framework based on jails, to run FreeBSD containers on FreeBSD
- **Bastille** - is an open-source system for automating deployment and management of containerized applications on FreeBSD.
- **Iocage** – Convenient, lightweight, and easy container management

Leading Edge

- Capsicum
- CHERI/CheriBSD
- Netflix streaming 400GB/s from single server

The Power to Connect – Excerpt from Netflix Case Study

Netflix Open Connect Appliance
2RU 40Gb/s Storage Appliance with 248TB storage

Application

Open Connect is the name of the global network that delivers Netflix TV shows and movies to members world-wide.

- The building blocks are purpose-built Open Connect Appliances (OCAs).
- FreeBSD was selected as the operating system for OCA because of its balance of stability and features, strong development community, staff expertise, and license.

Results

- **Delivers over 100 Tb/second globally at peak**
- **90 Gb/s from an OCA – using commodity parts and FreeBSD**
- FreeBSD is central to pushing this much content **cost-effectively**. By minimizing kernel to userspace copies, data stays in the kernel as long as possible
- Async Sendfile, a Netflix and NGINX innovation, is available to all FreeBSD users
  - Web server tells kernel to send this chunk of this file out over this socket
  - Kernel returns to userspace so the web server can do other things
  - Kernel continues in background sending files to users

https://freebsdfoundation.org/blog/freebsd-case-study-netflix/

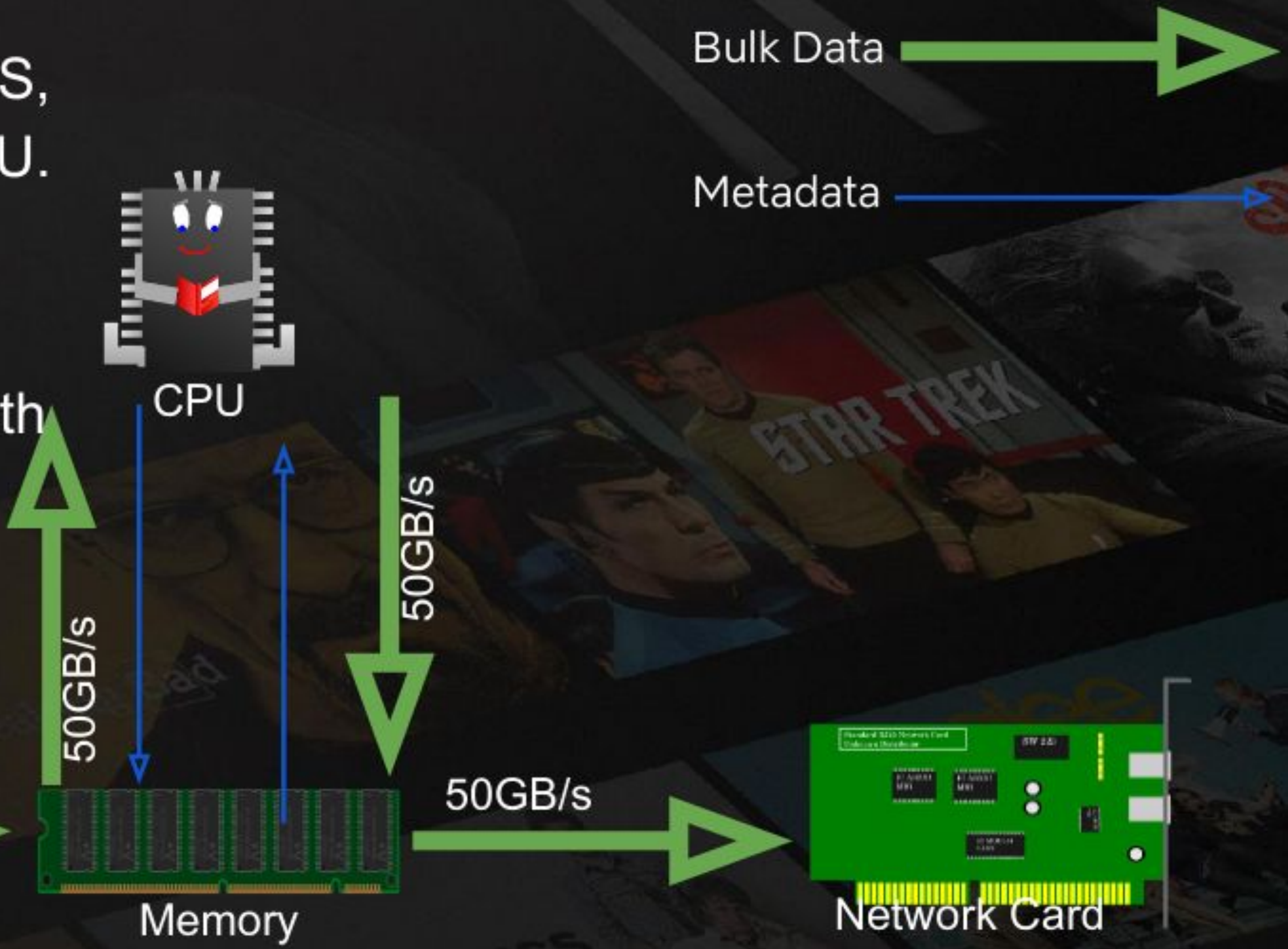
NETFLIX

Netflix 400Gb/s Video Serving Data Flow

Using sendfile and software kTLS, data is encrypted by the host CPU.

400Gb/s == 50GB/s

~200GB/sec of memory bandwidth and ~64 PCIe Gen 4 lanes are needed to serve 400Gb/s



https://people.freebsd.org/~gallatin/talks/euro2021.pdf

Proactive Capsicum

- Lightweight capability and sandbox framework
  - “Practical Capabilities for UNIX”
- Capabilities + Capability Mode
- Inspiration for Linux Landlock, see also Google Fuchsia

https://www.cl.cam.ac.uk/research/security/capsicum/

Proactive Security - Capsicum

- Lightweight capability and sandbox framework
  - "Practical Capabilities for UNIX"
- Capabilities + Capability Mode
- Inspiration for Linux Landlock, see also Google Fuchsia

https://www.cl.cam.ac.uk/research/security/capsicum/

CHERI

What is CHERI?

CHERI (Capability Hardware Enhanced RISC Instructions) extends conventional hardware Instruction-Set Architectures (ISAs) with new architectural features to enable fine-grained memory protection and highly scalable software compartmentalization.

What is Morello?

What's exciting is ARM and University of Cambridge are collaborating on an experimental CHERI-extended multicore, superscalar ARMv8-A processor (*the instruction-set architecture used in almost all mobile devices in the world*), that will run CheriBSD – Morello Program.

The program may radically change the way Arm designs and programs processors in the future to enable better built-in security

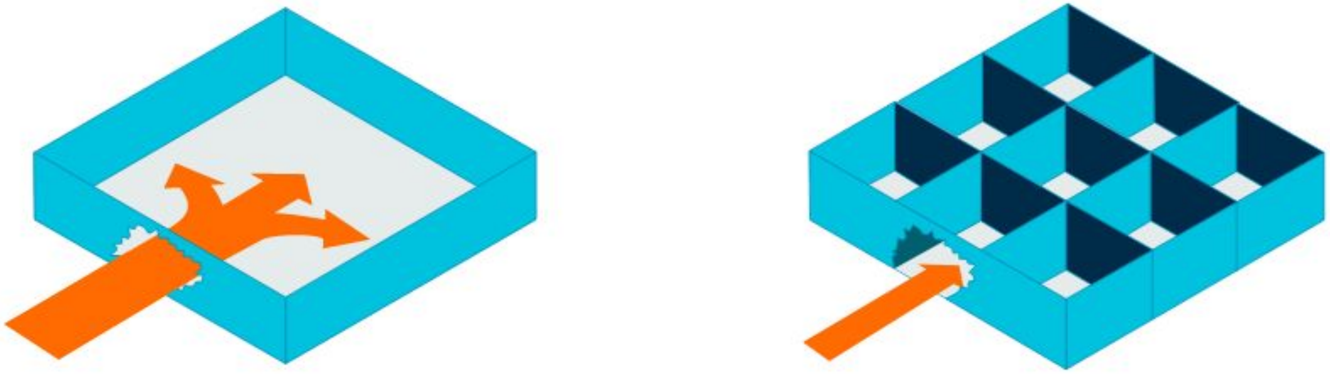
What is CheriBSD?

CheriBSD is a FreeBSD-based operating system that companies and universities will turn to, if they want to explore and use Morello, because it is the only OS that fully integrates CHERI support today and for several years to come.

https://www.cl.cam.ac.uk/research/security/ctsrd/cheri/ https://freebsdfoundation.org/freebsd-fridays/

CHERI

Compartmentalization



Breach has full access

Breach is contained to a specific area

https://www.arm.com/blogs/blueprint/morello

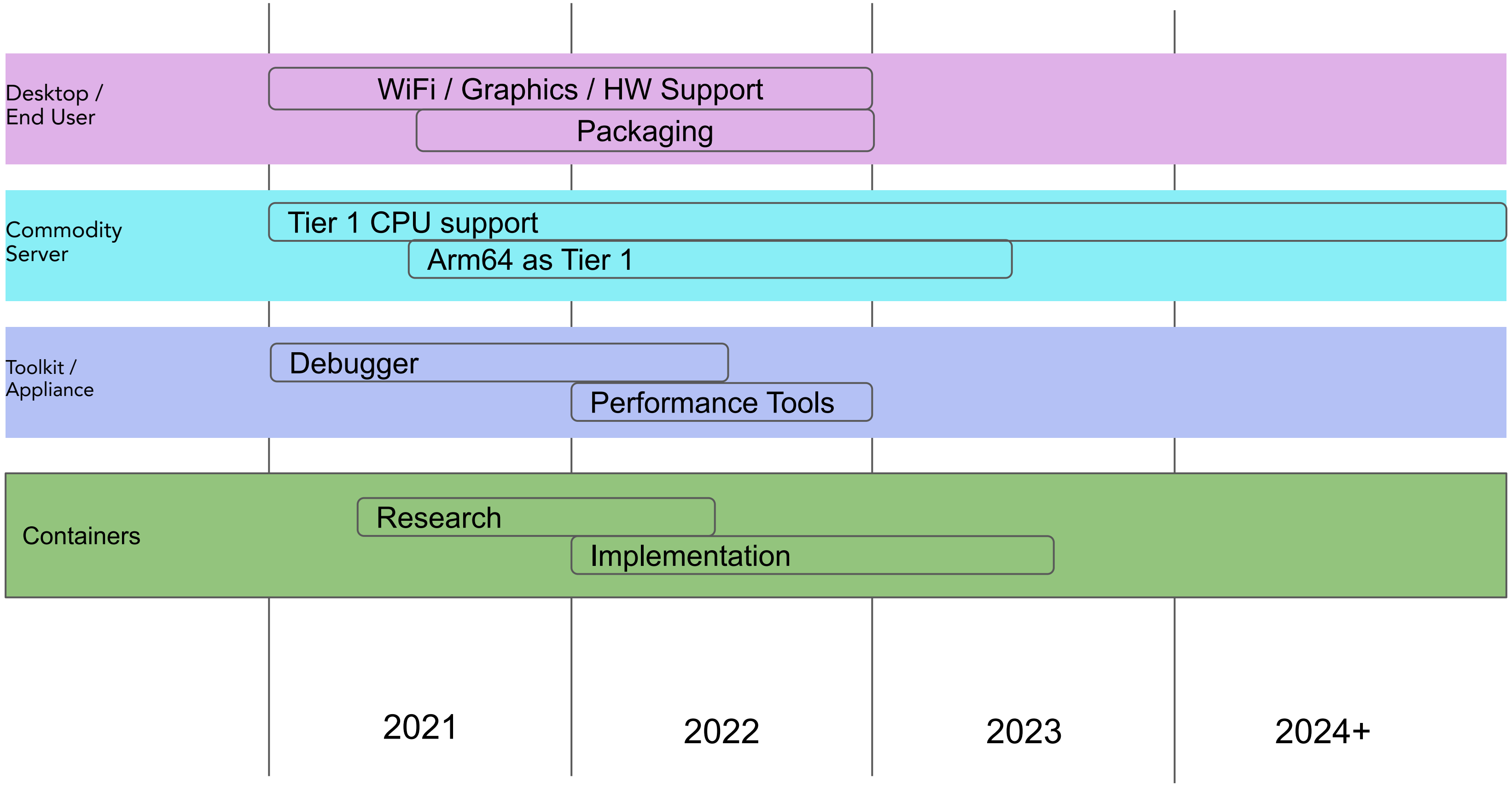
Proactive Security - Syzkaller

- Unsupervised code-coverage-driven kernel syscall fuzzer
- 111 commits to date from Syzkaller-reported issues
- Important stability improvements e.g. socket locking
- KASAN and KMSAN improve kernel assertions / diagnostics
- Working on improving ease of deployment in other environments
- 1 FF employee

https://syzkaller.appspot.com/freebsd

https://github.com/google/syzkaller/blob/master/docs/freebsd/README.md

2021-2024+ Technology Group Roadmap



Desktop / End User

WiFi / Graphics / HW Support

Packaging

Commodity Server

Tier 1 CPU support

Arm64 as Tier 1

Toolkit / Appliance

Debugger

Performance Tools

Containers

Research

Implementation

2021

2022

2023

2024+

FreeBSD

FreeBSD FOUNDATION

https://freebsdfoundation.org/blog/technology-roadmap/

Now What?

Why Linux and FreeBSD Should Work Together

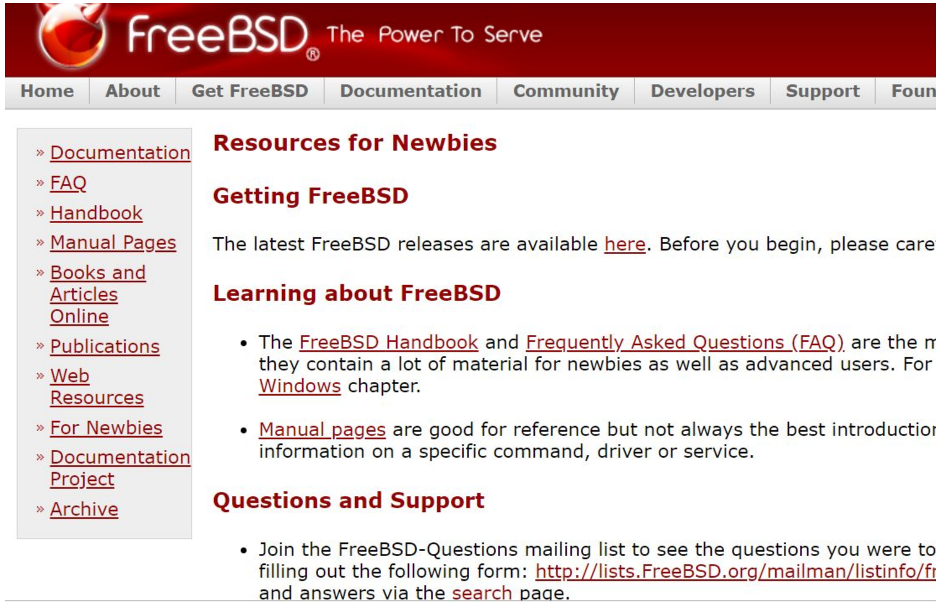
- May work on multiple operating systems during your employment
- Learn from each other. We both have successes and failures.
- Different coding methodologies and philosophies – Understanding the reasons for both.
- FreeBSD's smaller code base makes it a great reference platform.
- "Using and learning FreeBSD made me a better Linux admin and systems engineer."

FreeBSD

FreeBSD FOUNDATION

Why Contribute to FreeBSD

- Be part of an inclusive and welcoming community with a strong mentoring culture
- Great way to learn systems programming and study operating systems.
- The size of the project allows for a greater chance for anyone to make a notable impact.
- Some of the most notable BSD and FreeBSD Founders are still involved in the Project – And, they are approachable!
- Democratically run open source project allowing committers to commit their changes directly to the source tree without having to go through hierarchy of lieutenant model.



FreeBSD® The Power To Serve

Home | About | Get FreeBSD | Documentation | Community | Developers | Support | Foun

- » Documentation
- » FAQ
- » Handbook
- » Manual Pages
- » Books and Articles Online
- » Publications
- » Web Resources
- » For Newbies
- » Documentation Project
- » Archive

Resources for Newbies

Getting FreeBSD

The latest FreeBSD releases are available here. Before you begin, please care

Learning about FreeBSD

- The FreeBSD Handbook and Frequently Asked Questions (FAQ) are the m they contain a lot of material for newbies as well as advanced users. For Windows chapter.
- Manual pages are good for reference but not always the best introductior information on a specific command, driver or service.

Questions and Support

- Join the FreeBSD-Questions mailing list to see the questions you were to filling out the following form: http://lists.FreeBSD.org/mailman/listinfo/fr and answers via the search page.

How to get started with FreeBSD!

- Go to Newbies page - https://www.freebsd.org/projects/newbies/
- Read Contributing to FreeBSD (https://www.freebsd.org/doc/en_US.ISO8859-1/articles/contributing/
- Read The FreeBSD Handbook https://www.freebsd.org/doc/handbook/book.html
- Learn about the history of FreeBSD here: https://www.mckusick.com/history/
- FreeBSD Foundation's resource page with how-to guides! https://freebsdfoundation.org/freebsd-project/resources/
- FreeBSD Fridays Introductory Series https://freebsdfoundation.org/freebsd-fridays/
- Install FreeBSD on a virtual machine by following the instructions here: https://freebsdfoundation.org/freebsd-project/resources/installing-freebsd-with-virtualbox/
- LPI BSD Certification provides good learning sequence to follow: https://www.lpi.org/our-certifications/exam-702-objectives
- Have a question? There are many resources to get help:
  - https://www.facebook.com/groups/FreeSBD
  - freebsd-questions@freebsd.org
  - Join Mailing Lists Forums, Mailing Lists, IRC and Events (https://www.freebsd.org/community.html)

FreeBSD

FreeBSD FOUNDATION

Questions?