Repo →

github.com/michaelo/bootable-applications

Bootable applications

An introduction

Michael Odden / NDC TechTown 2024
me@michaelodden.com

Who are you?

Practical info

Repo

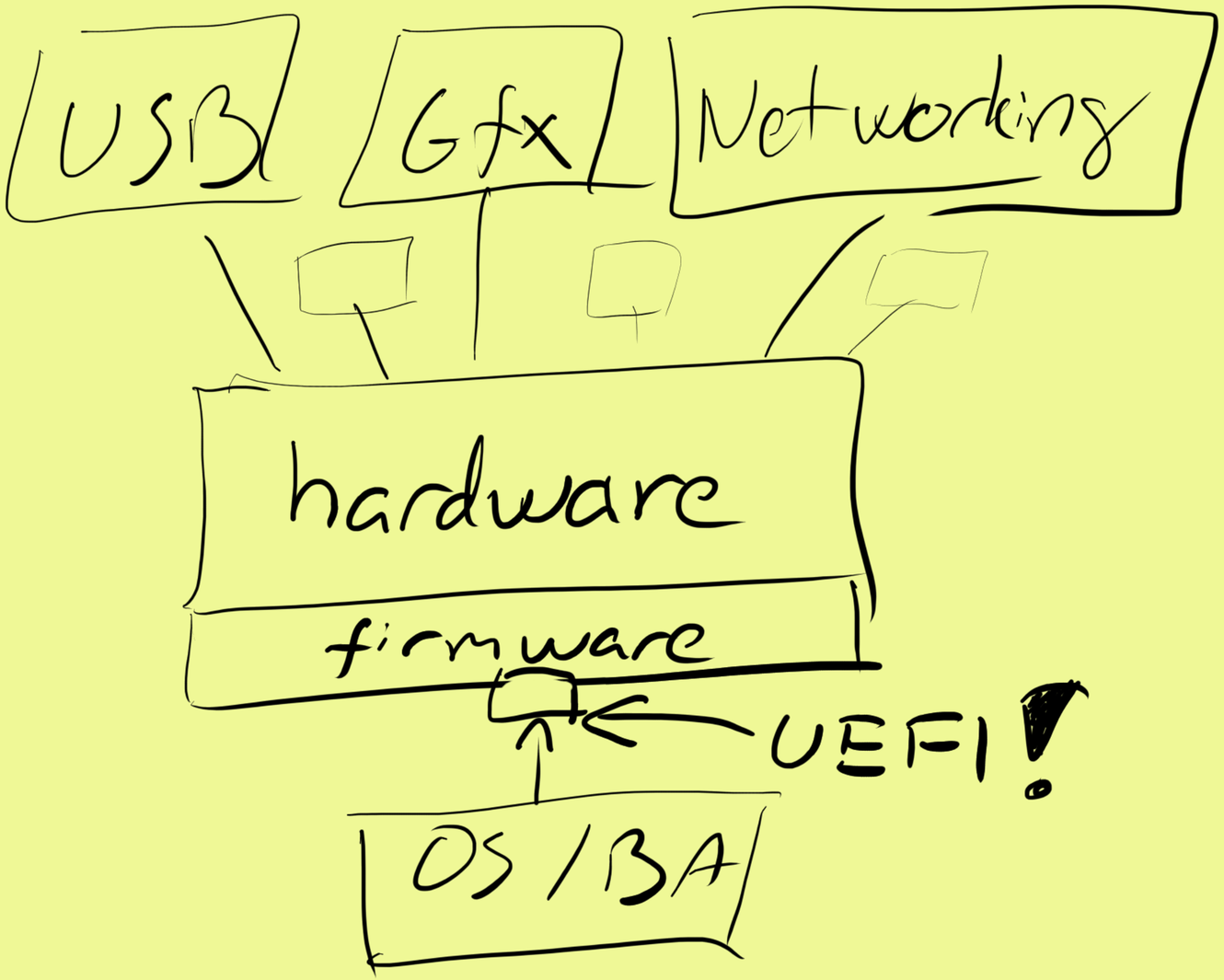
github.com/michaelo/bootable-applications

First of all... UEFI?

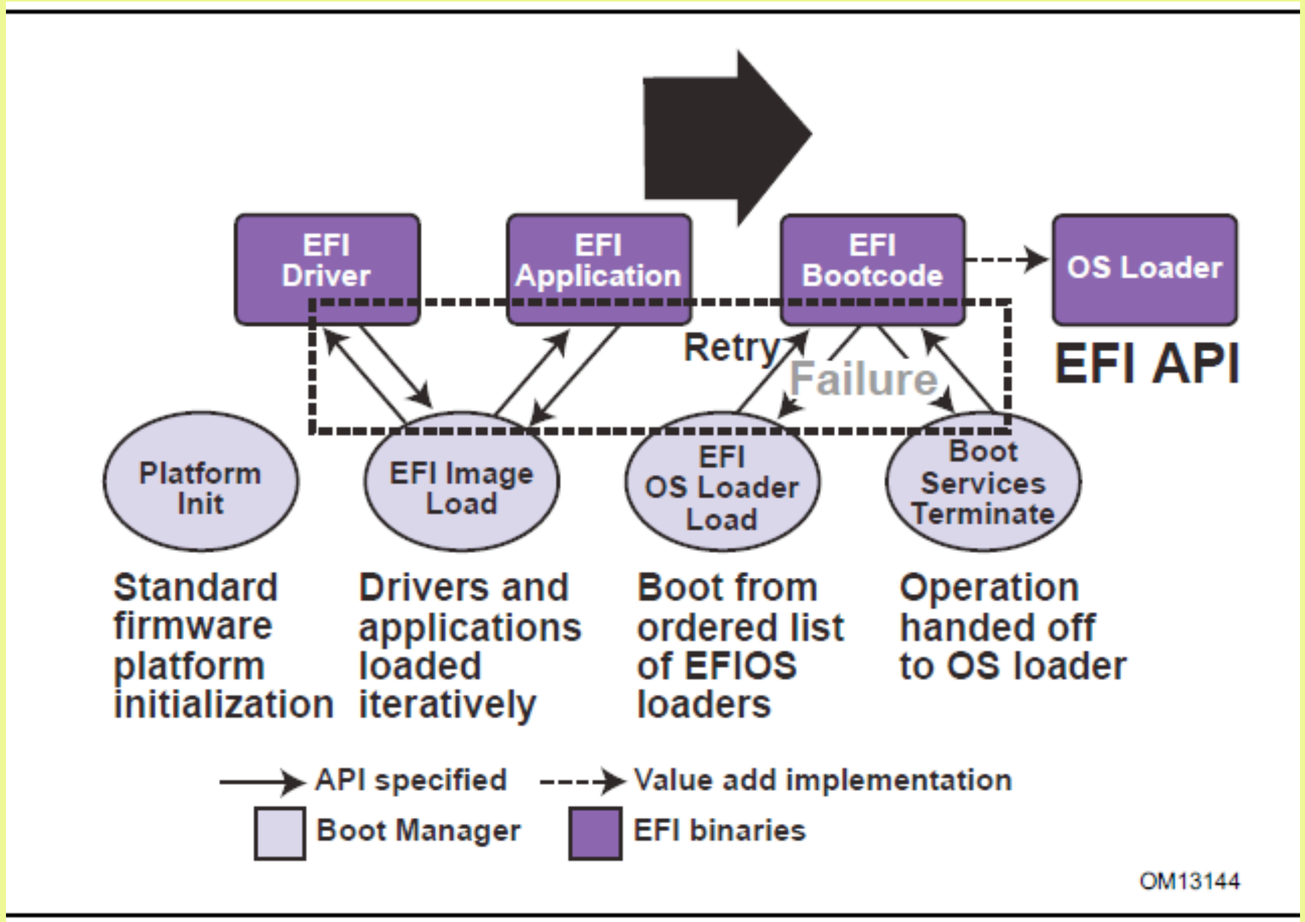
... - Intel - Microsoft - ARM - AMD - Apple - ...
... - IBM - nvidia - Cisco - Broadcom - ...

Scope

... why!?

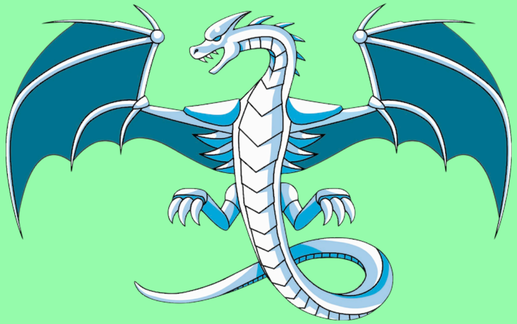

USB
Gfx
Networking
hardware
firmware
UEFI!
OS / BA

https://uefi.org/specs/UEFI/2.10/



4. EFI System Table
4.1. UEFI Image Entry Point
4.1.1. EFI_IMAGE_ENTRY_POINT
4.2. EFI Table Header
4.3. EFI System Table
4.4. EFI Boot Services Table
4.5. EFI Runtime Services Table
4.6. EFI Configuration Table & Properties Table
4.7. Image Entry Point Examples
5. GUID Partition Table (GPT) Disk Layout
6. Block Translation Table (BTT) Layout
7. Services — Boot Services
8. Services — Runtime Services
9. Protocols - EFI Loaded Image
10. Protocols – Device Path Protocol
11. Protocols — UEFI Driver Model
12. Protocols — Console Support
13. Protocols – Media Access
14. Protocols — PCI Bus Support
15. Protocols — SCSI Driver Models and Bus Support
16. Protocols — iSCSI Boot
17. Protocols — USB Support
18. Protocols — Debugger Support
19. Protocols — Compression Algorithm Specification

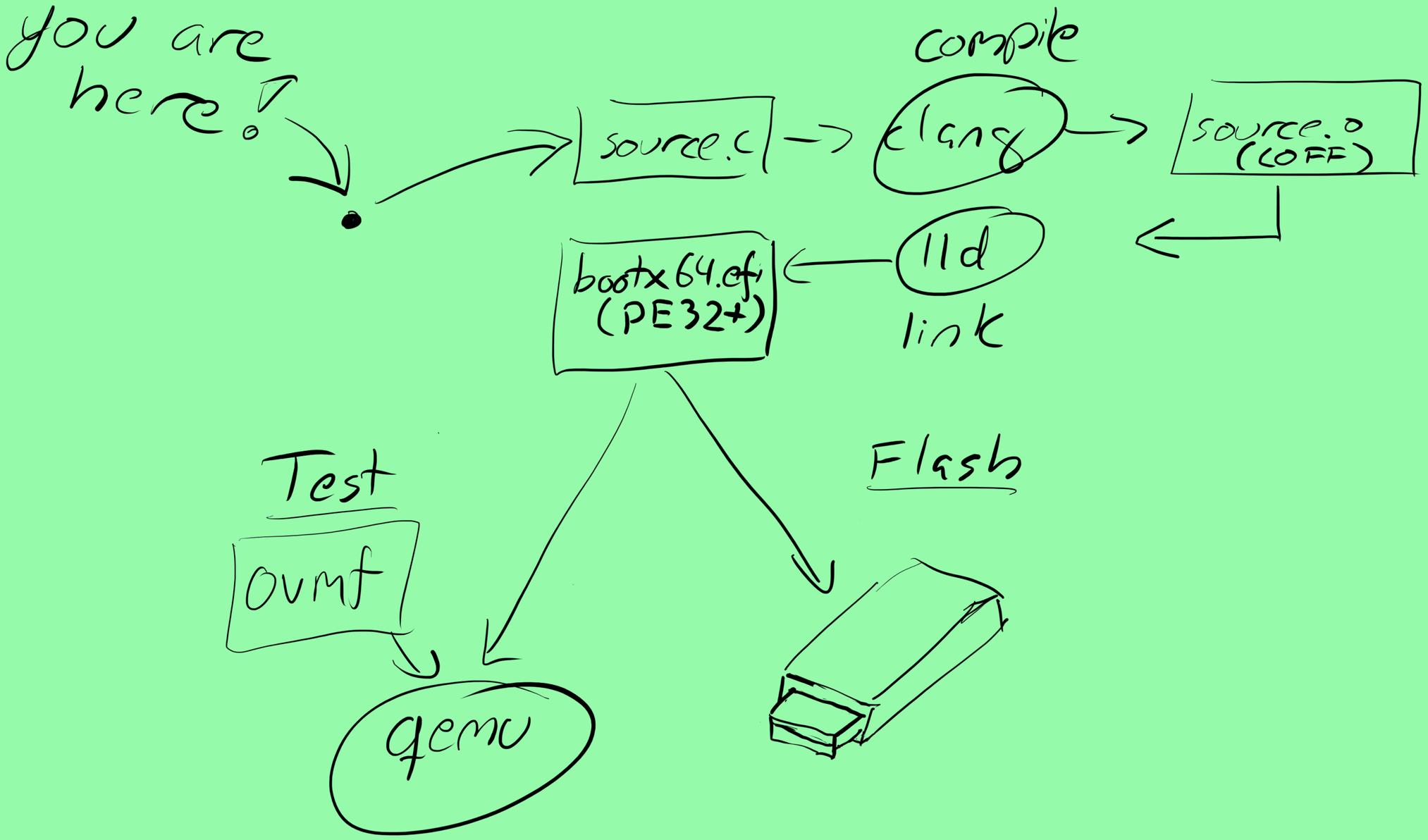
The tools



lil_uefi

QEMU

The process


you are here!
compile
source.c
clang
source.o (COFF)
lld
link
bootx64.efi (PE32+)
Test
ovmf
qemu
Flash

Let’s dive in!

Further reading

- UEFI specification: https://uefi.org/specifications
- lil_uefi by Allen Webster and Ryan Fleury: https://handmade.network/p/308/lil-uefi/
- osdev.org: http://wiki.osdev.org/UEFI
- TianoCore / EDK II - https://www.tianocore.org/ / https://github.com/tianocore/edk2
- uefi-simple: https://github.com/pbatard/uefi-simple

Repo

Now go do something ~~good~~ fun!