



# Open. Together.



**OCP**  
REGIONAL  
SUMMIT

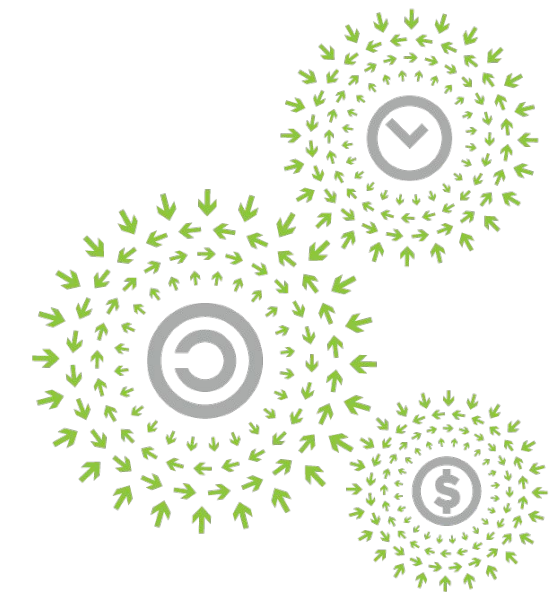


# Open System Firmware From Scratch

Jonathan Zhang, Firmware Engineer, Facebook  
David Hendricks, Firmware Engineer, Facebook



OPEN SYSTEMS  
FIRMWARE



**OPEN**  
PLATINUM™



**OCP**  
REGIONAL  
SUMMIT

Open. Together.

# Agenda

- **What is FB doing? Why?**
- The build system design
- How do we enable our partners and the community?



OPEN SYSTEMS  
FIRMWARE



# Use cases of OSF in FB

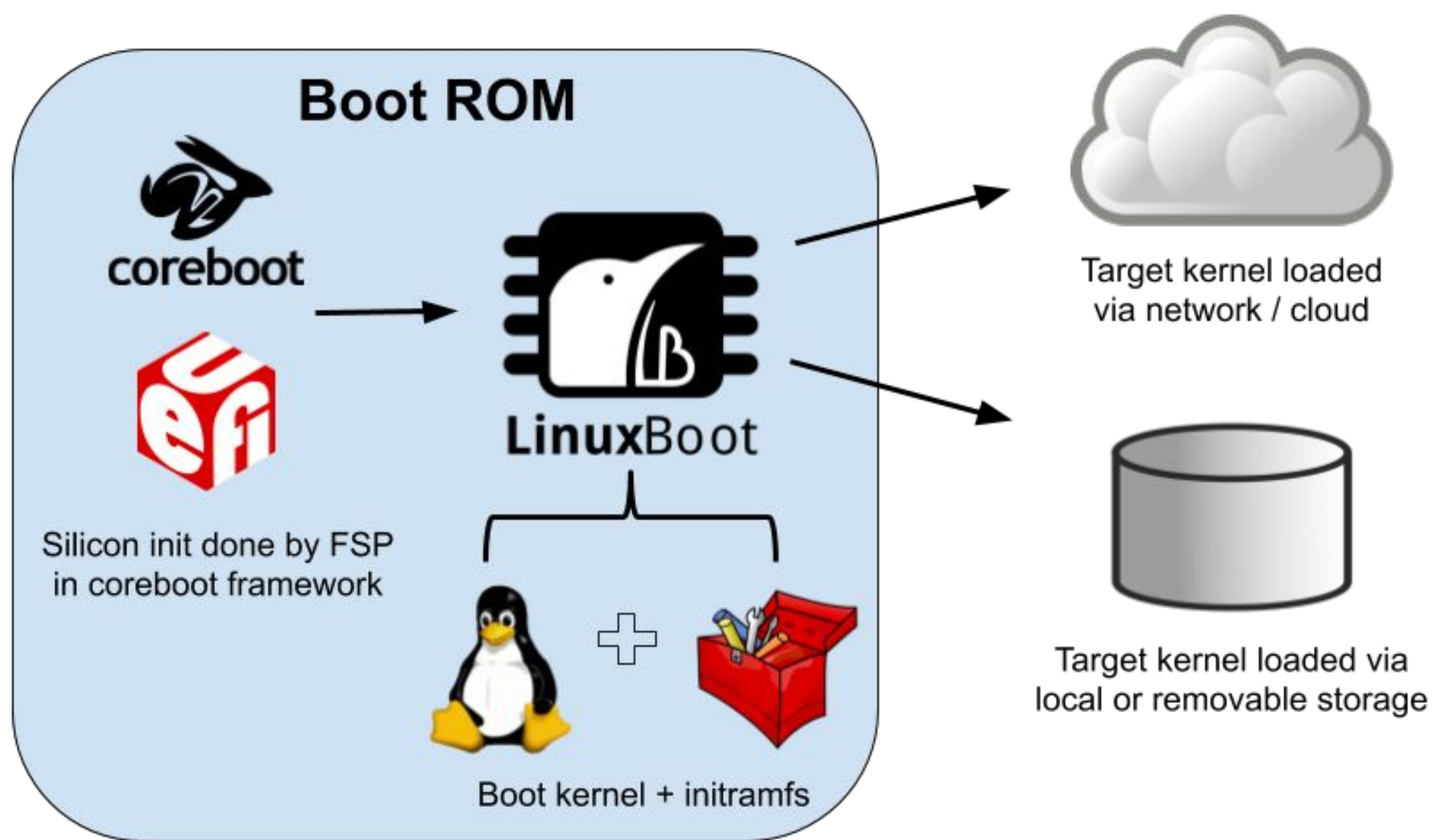
- Enabling OSF on compute/storage server fleets
- Deploying OSF on networking systems (minipack)
- Deploying OSF on telecom devices (Open Cellular)
- Deploying OSF on embedded devices

# coreboot/LinuxBoot OSF motivation

- Open source
  - Transparent
  - Operational efficiency, reactive vs. proactive for issues
- Modern architecture
  - Simple and portable, leverages Linux
  - Easy to add features and customization
- Familiarity
  - Reuse runtime components
  - Leverage engineering resources
- Fast boot time
  - Demanded by some work loads



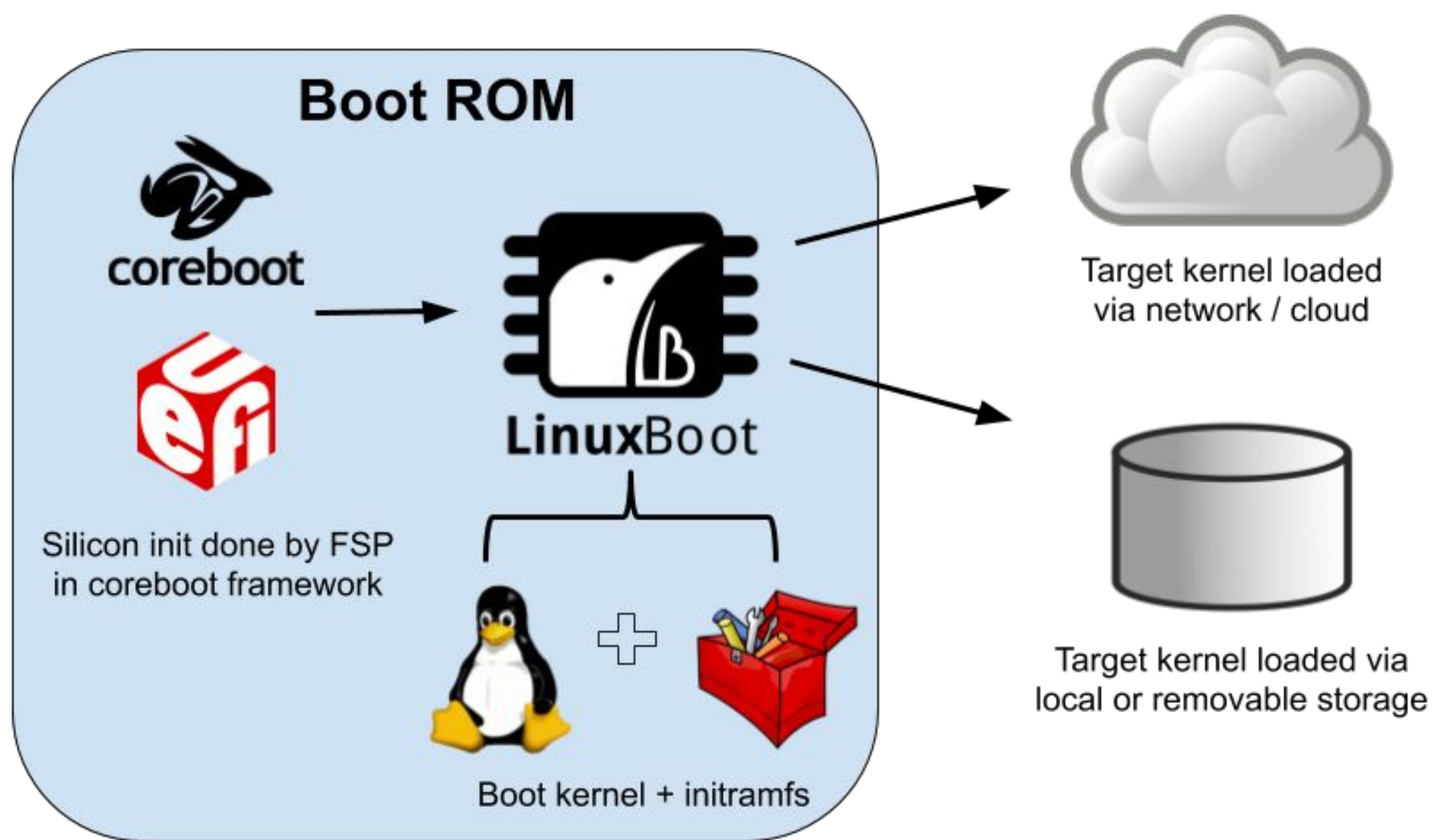
# FB OSF architecture



- Put a kernel+initramfs in boot ROM



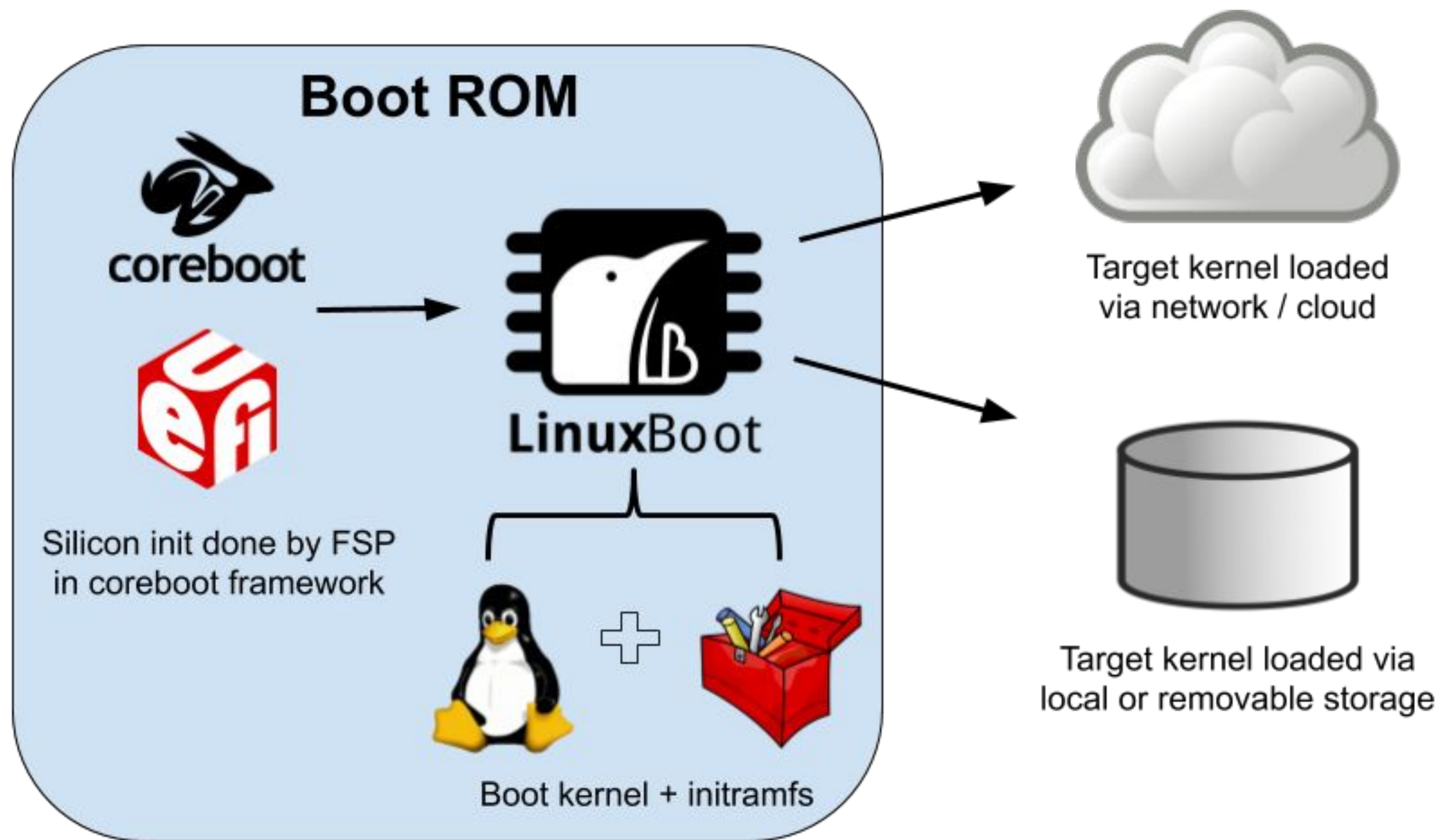
# FB OSF architecture



- Put a kernel+initramfs in boot ROM
- Do minimal silicon init and jump to LinuxBoot as early as possible



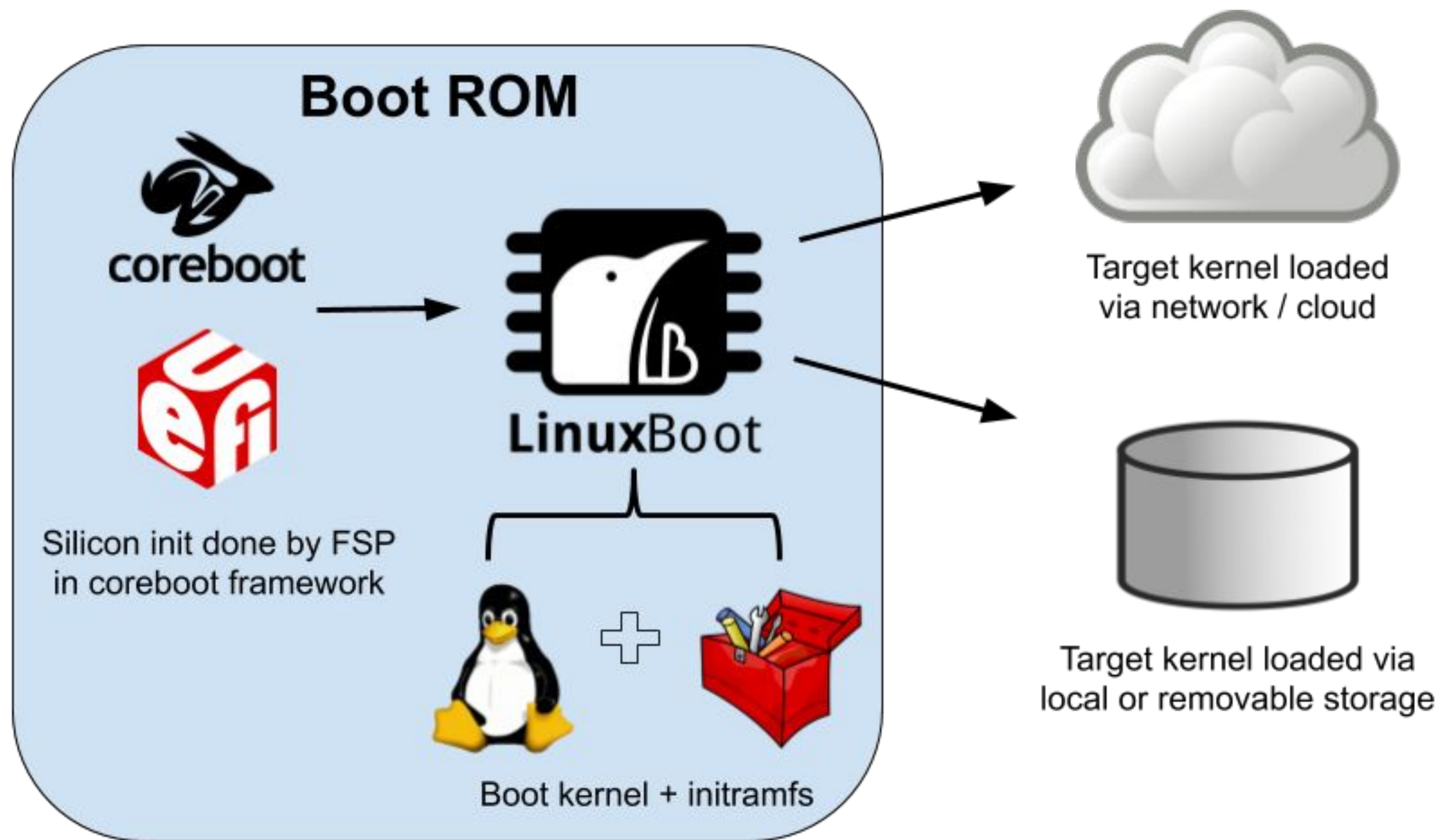
# FB OSF architecture



- Put a kernel+initramfs in boot ROM
- Do minimal silicon init and jump to LinuxBoot as early as possible
- Use Linux to boot Linux



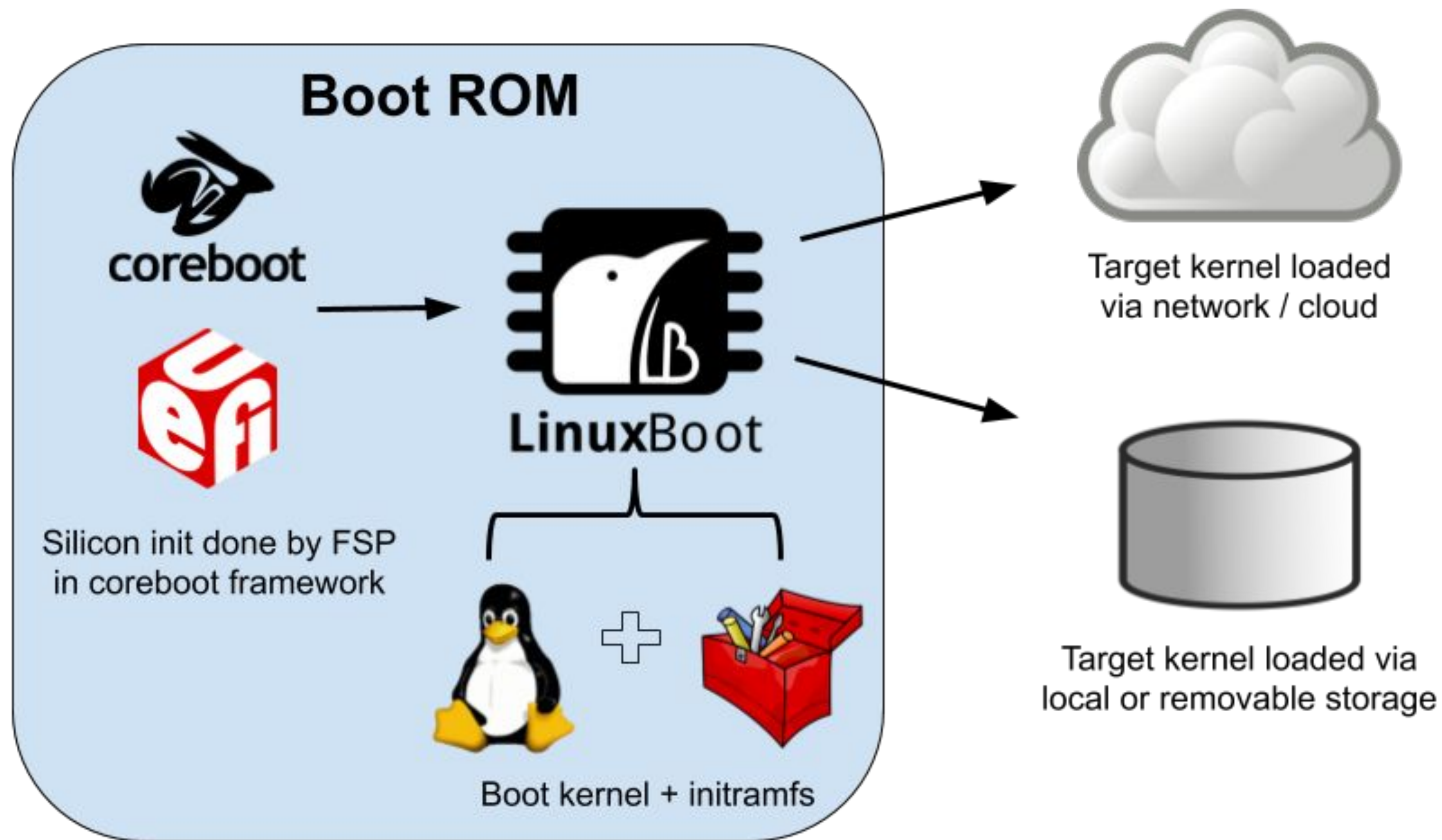
# FB OSF architecture



- Put a kernel+initramfs in boot ROM
- Do minimal silicon init and jump to LinuxBoot as early as possible
- Use Linux to boot Linux
- Add features and tools as needed



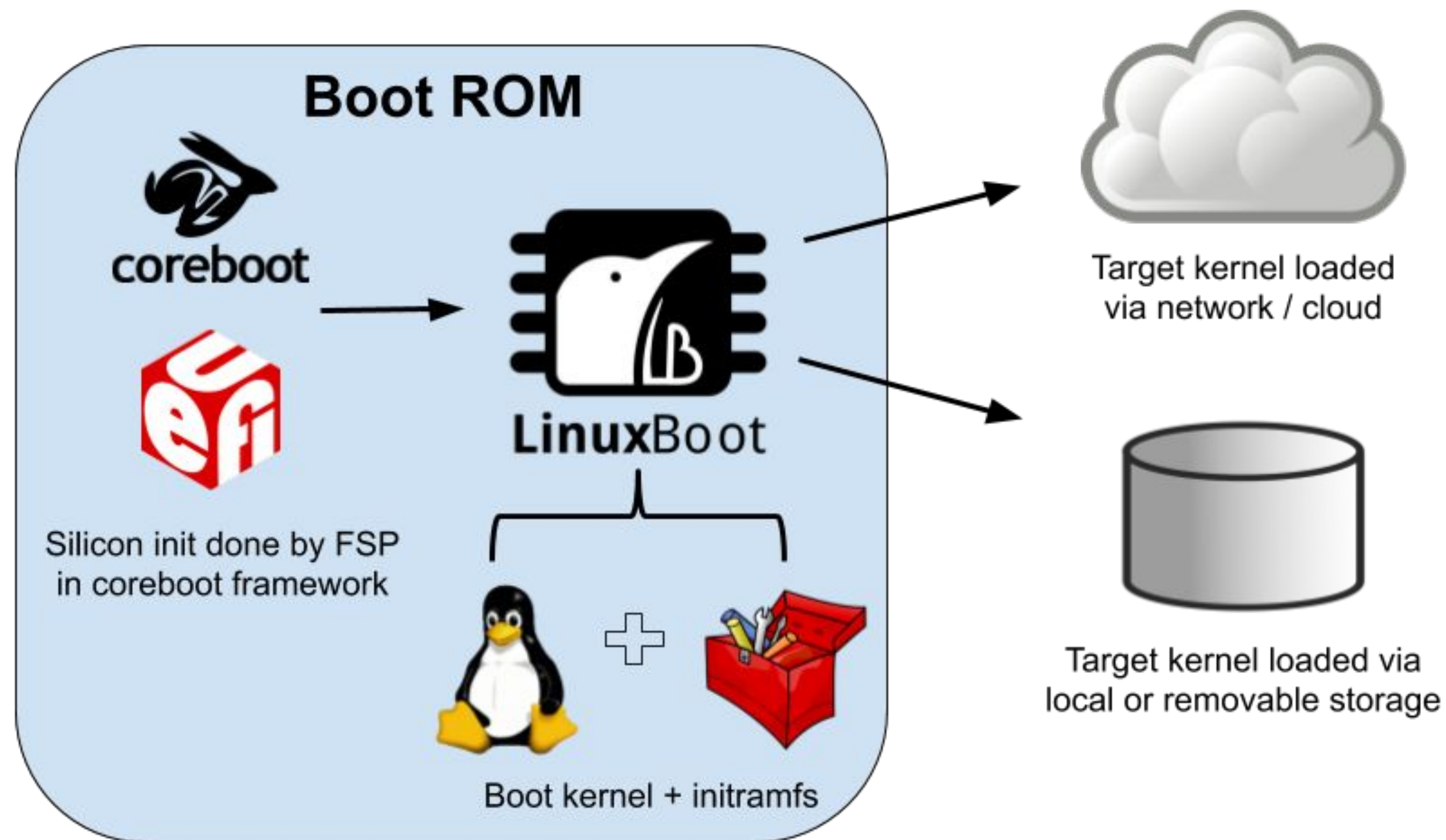
# FB OSF architecture



- Put a kernel+initramfs in boot ROM
- Do minimal silicon init and jump to LinuxBoot as early as possible
- Use Linux to boot Linux
- Add features and tools as needed
- Debug, build, and deploy on our schedule.



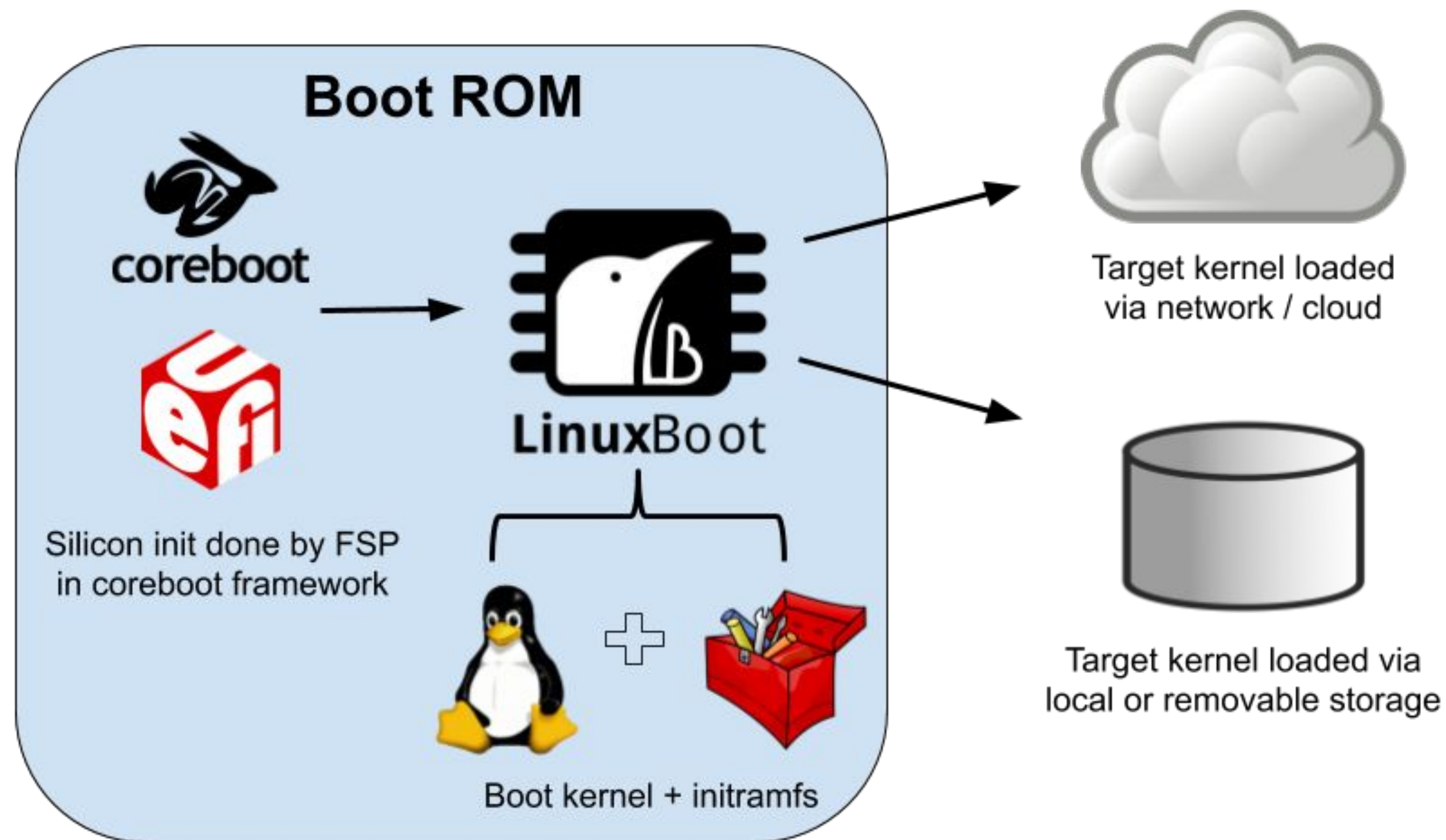
# FB OSF architecture



- Put a kernel+initramfs in boot ROM
- Do minimal silicon init and jump to LinuxBoot as early as possible
- Use Linux to boot Linux
- Add features and tools as needed
- Debug, build, and deploy on our schedule.
- Transparent and reproducible



# FB OSF architecture



- Put a kernel+initramfs in boot ROM
- Do minimal silicon init and jump to LinuxBoot as early as possible
- Use Linux to boot Linux
- Add features and tools as needed
- Debug, build, and deploy on our schedule.
- Transparent and reproducible
- Bring modern, open-source development practices to firmware



# Agenda

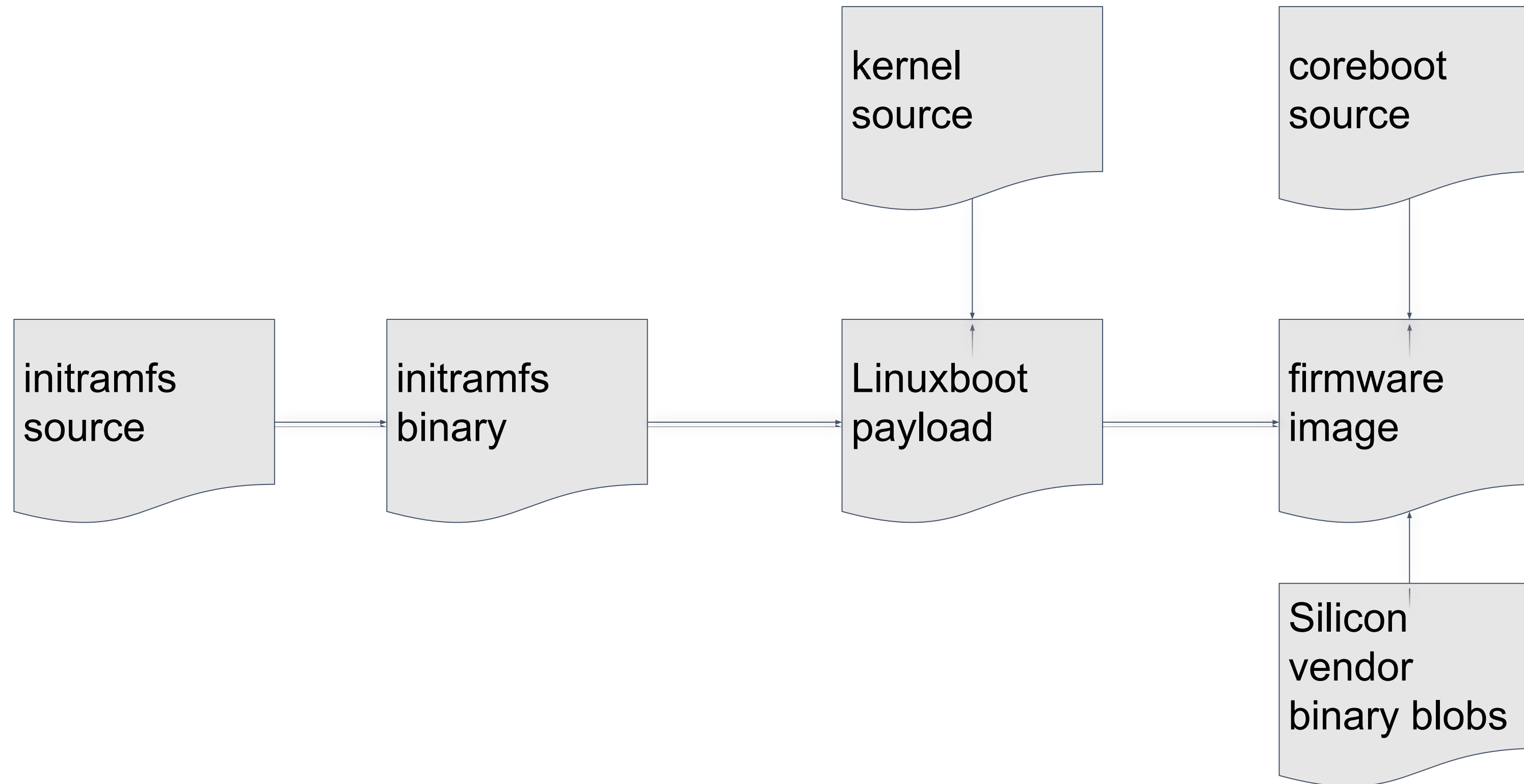
- What is FB doing? Why?
- **The build system design**
- How do we plan to enable our partners and the community



OPEN SYSTEMS  
FIRMWARE



# Build steps





# Components -- build environment

- coreboot cross toolchain build for elf
  - <COREBOOT\_ROOT>/util/crossgcc
- Tool chain for other components
  - GCC version needs to be no older than 5.0.0
- Use docker to ensure identical build environment



# Components - initramfs (FB solution)

- go
  - Modern systems software programming language with pointers, concurrency, IPC, rich libraries, etc.
  - Built-in testing facilities
  - <https://golang.org/>
- u-root
  - A fully Go userland with Linux bootloaders
  - Can be source-based like Perl Linux, or binary like Busybox
  - <https://github.com/u-root/u-root/>
- Binaries
  - Can include C binaries such as flashrom, dmidecode, etc.



# Components - Linux kernel

- Same kernel used in runtime environment
  - Production-quality drivers, familiar interfaces
  - Minimized configuration for size considerations
- Kconfig
  - Intuitive and familiar configuration system
  - `make menuconfig`



# Components -- coreboot

- Kconfig
  - Same as Linux
- Firmware Support Package (FSP)
  - Encapsulates Platform Initialization (PI) modules for general consumption
  - Used by coreboot, u-boot, Slim Bootloader, etc.
  - Detailed at another presentation: “Open System Firmware support for multi socket OCP platforms - coreboot POC on Tiogapass and Skylake SP”
- Flexible - Able to integrate vboot and other features
- Robust community

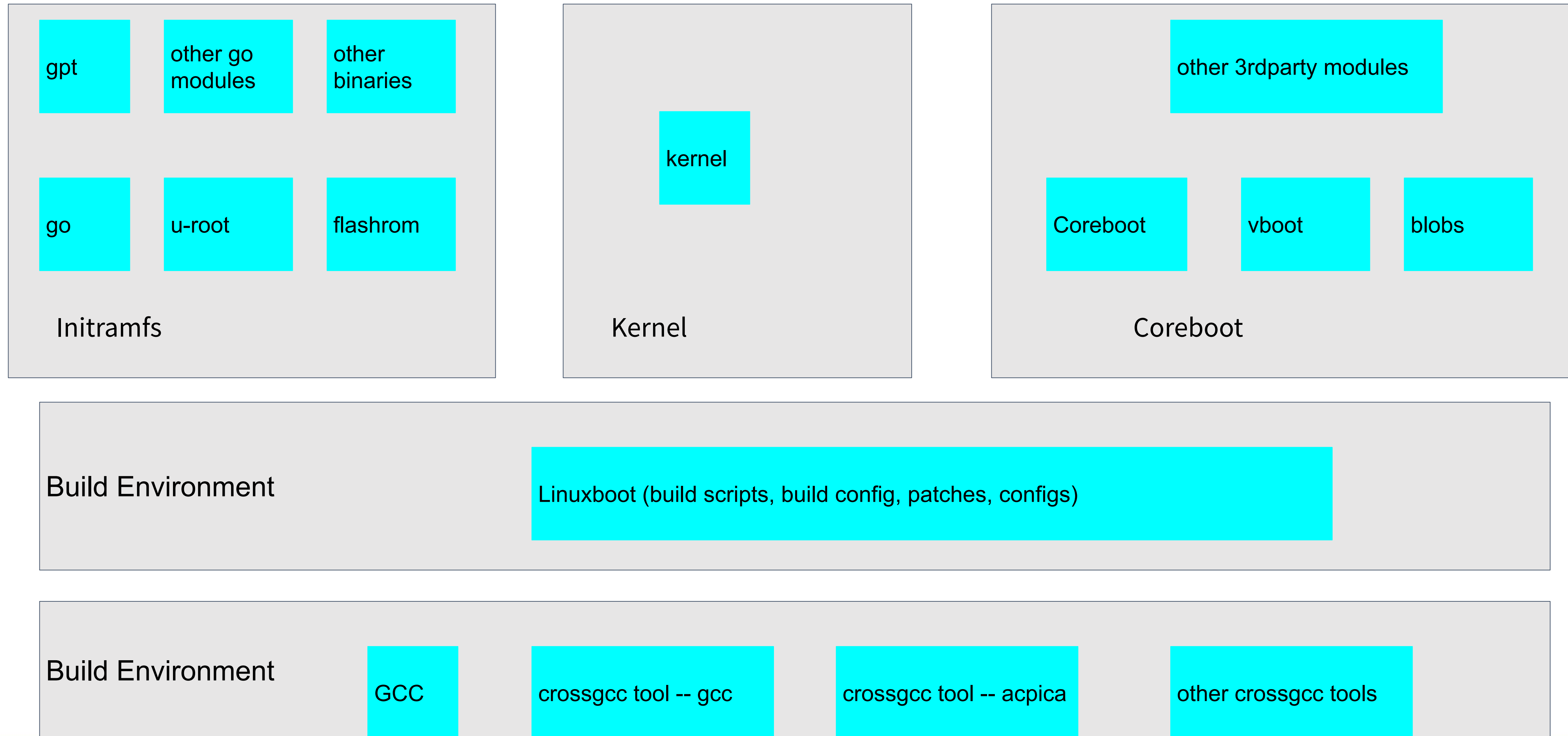


# Components -- SIP binary blobs

- IFD
- ME
- Microcode



# Components -- Put them together





# Build configuration file

- JSON format describing:
  - What components are included
  - Where to get a component
  - Public repos, Internal repos, tar balls, etc.
  - Which snapshot for a component
  - The directory structure



# Configurations

- ChromeOS Viral Product Data (VPD) binary blob
  - <https://chromium.googlesource.com/chromiumos/platform/vpd/>
- FSP configurations
- Boot log verbose level.
- Version info:
  - Overall version -- RO
  - Internal versions -- RO



# Facebook Linuxboot Repo

- Includes:
  - a. Build scripts.
  - b. Build configurations.
  - c. Kernel Kconfig, coreboot config
  - d. Patches (in rare case)
- Will be upstreamed
- Build steps:
  - a. Clone Linuxboot repo.
  - b. `cd linuxboot`
  - c. `PLATFORM=<platform> ./build.sh`



# Scenarios

- Q: Given an image, how is it build?
  - A: Run vpd tool to get internal\_Versions VPD variable.
- Q: Given a target, how do I debug?
  - A: Boot the target into u-root shell or OS shell, run vpd command to get VPD variables.
  - A: run flashrom command to dump the image.
- Q: Given build info, how do I replicate the build?
  - Clone Linuxboot repo according to the build info, and build



# Agenda

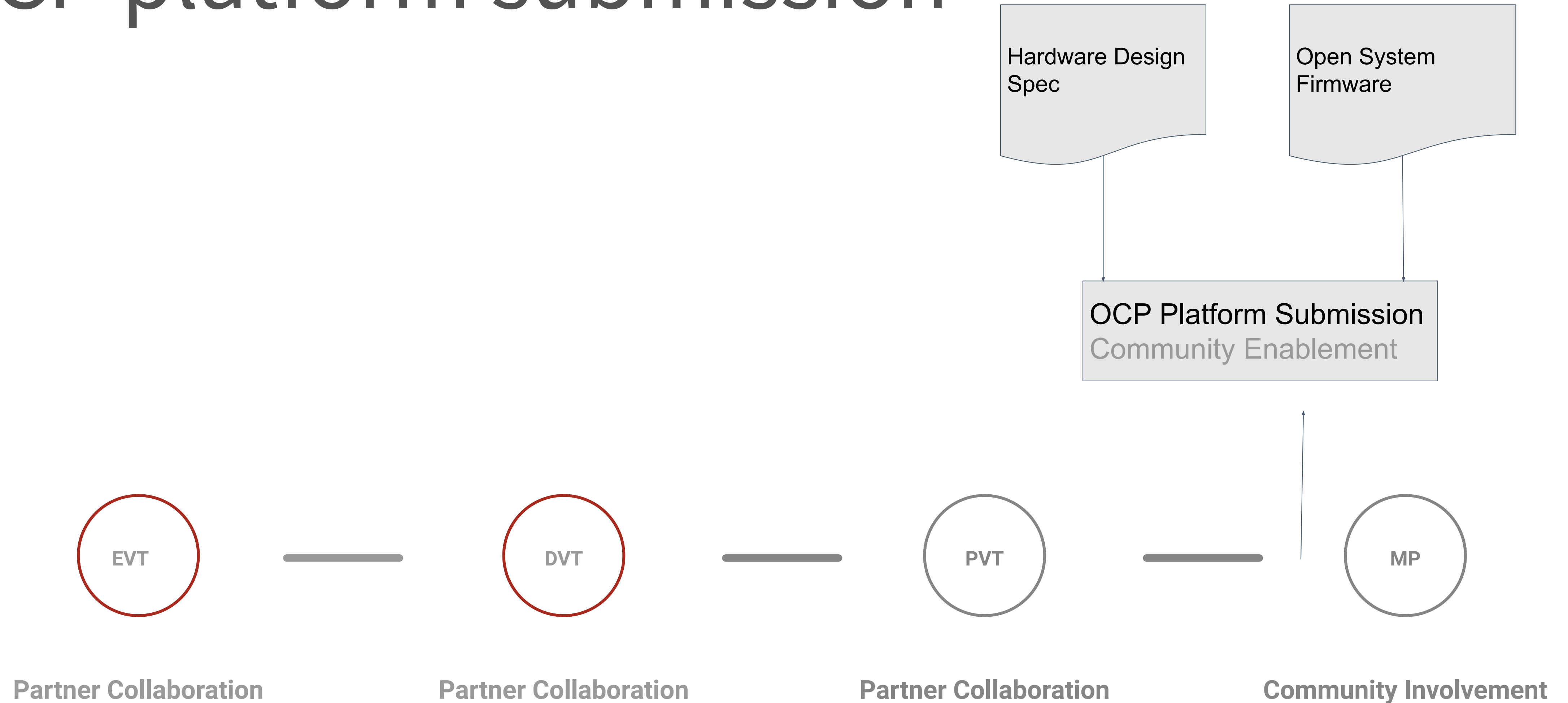
- What is FB doing? Why?
- The build system design
- **How do we plan to enable our partners and the community?**



OPEN SYSTEMS  
FIRMWARE

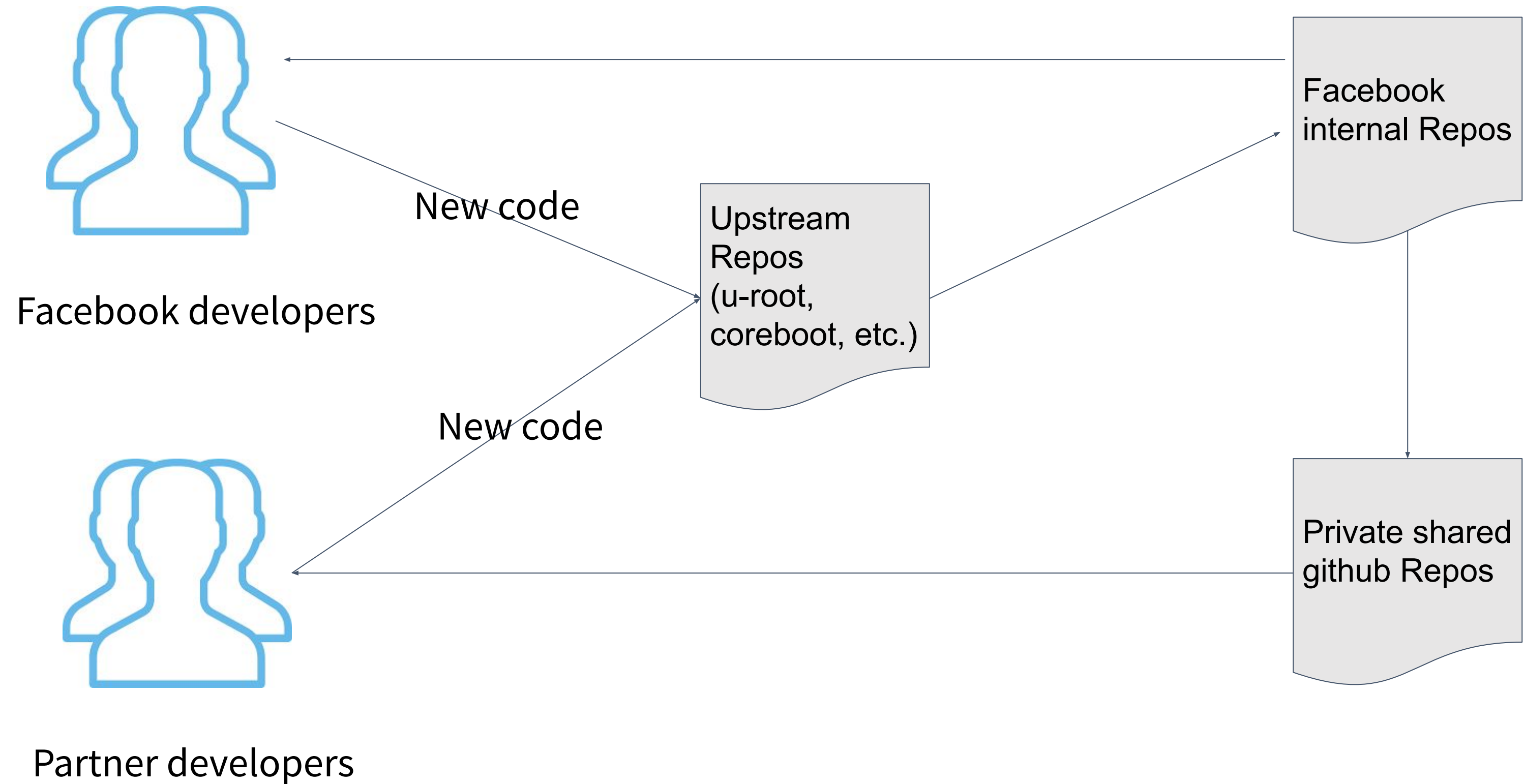


# OCP platform submission





# Partner Collaboration -- pre OCP submission





# Community Enablement -- post OCP submission

- FB is a community member.
- All pieces are downloadable and redistributable
  - Linuxboot, coreboot code
  - Silicon Vendor binaries
- CIT enabled
- OCP members can customize for their infrastructure as needed:
  - community support + professional support



# Test framework -- ConTest

- Continuous, on-demand, per-diff tests
- Plug-ins to accommodate test infrastructures.
- Will be open-sourced



# Call to Action

- OCP OSF project
  - <https://www.opencompute.org/projects/open-system-firmware>
- OSF community
  - Linuxboot: <https://www.linuxboot.org/>
  - u-root: <https://github.com/u-root/u-root>
  - coreboot: <https://www.coreboot.org/>
- Play with off-the-shelf OCP platform (such as Winwynn TiogaPass)





**OCP**  
REGIONAL  
SUMMIT

# Open. Together.

OCP Regional Summit  
26–27, September, 2019