Bootloader Process on an M1 Pro Macbook Pro

The model of computer that I have is the Macbook Pro 14-inch with the M1 Pro chip. This computer was one of the early adopters of the Apple Silicon chipset that is now currently used in modern Apple computers. These computers marked a departure from intel based chips to ARM based. Because of this, a Mac with Apple Silicon boots in a similar fashion to an iPhone or iPad. I used the Apple support website page to learn more about this process in which it provided me with an explanation and diagram to go off of. Booting on an M1 series of Macs consists of four main stages: The Boot ROM, The Low-Level Bootloader (or LLB), iBoot, and then macOS.

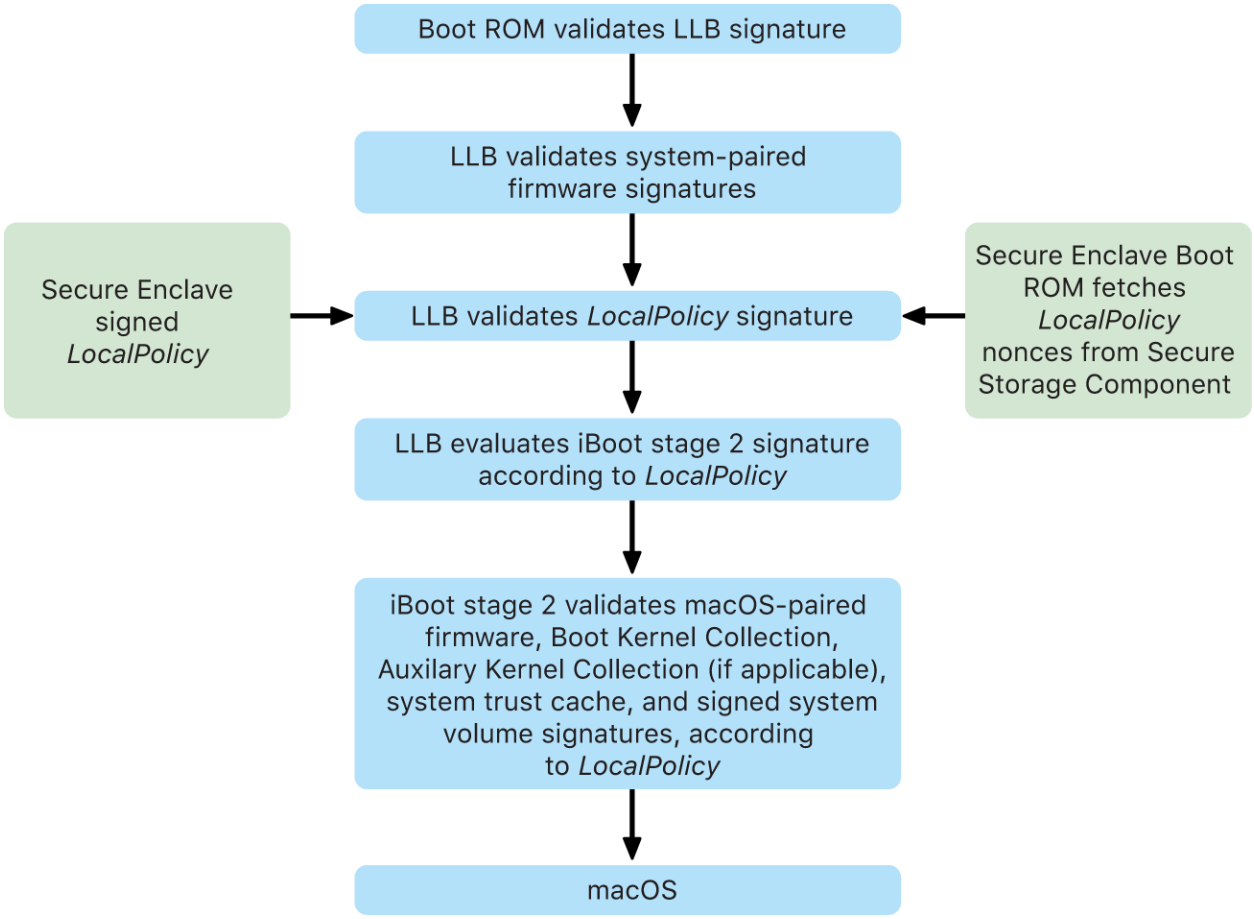
Bootloader Process Overview:

1. **Boot ROM**
    - The Boot ROM is embedded in the hardware and is immutable
    - Boot ROM initiates the chain of trust by verifying and loading the LLB
    - If verification fails, the system moves to DFU (Device Firmware Update) mode, which awaits a connection to another Mac to recover
2. **Low-Level Bootloader (LLB)**
    - The LLB is stored in flash memory and performs verification steps
    - It validates the LocalPolicy signature, which is a security configuration signed by the Secure Enclave Processor (SEP)
    - Based on LocalPolicy the LLB determines the system's security mode:
        - *Full Security:* Similar to IOS, Boots only the latest macOS with verified signatures
        - *Reduced Security:* Allows older macOS versions and kernel extensions
        - *Permissive Security:* Allows custom kernels signed locally
3. **iBoot**
    - Referred to as the firmware, iBoot then continues the chain of trust by verifying cryptographic hashes, including Signed System Volume (SSV) root hash.
    - It prepares the kernel environment by loading macOS firmware such as the Secure Neural Engine and Always-On Processor.
    - iBoot ensures that key components like the Auxiliary Kernel Collection (AuxKC) are verified and loaded into memory.
4. **MacOS Kernel**

- The kernel assumes control from iBoot, initializes the hardware, mounts the file system, and loads kernel extensions (kexts) as required.
- macOS completes the startup, bringing the system to the login screen.

Architectural Differences: ARM vs. Intel

1. Integrated Security: ARM-based Apple Silicon systems incorporate secure boot policies at every stage. Intel Macs rely on UEFI firmware which is more complex and challenging to secure.
2. Hardware Dependency: M1 Macs use a highly minimal firmware that depends on internal SSD components to complete the boot process. Intel Macs are able to support fully external boot disks with more extensive driver support in firmware.
3. Recovery Mechanisms: ARM-based Macs default to DFU mode for recovery mode, while Intel Macs are able to use external recovery media or hardware-based diagnostics. With ARM architecture, the reliance on the SSD causes the Mac to be unusable after SSD fails.



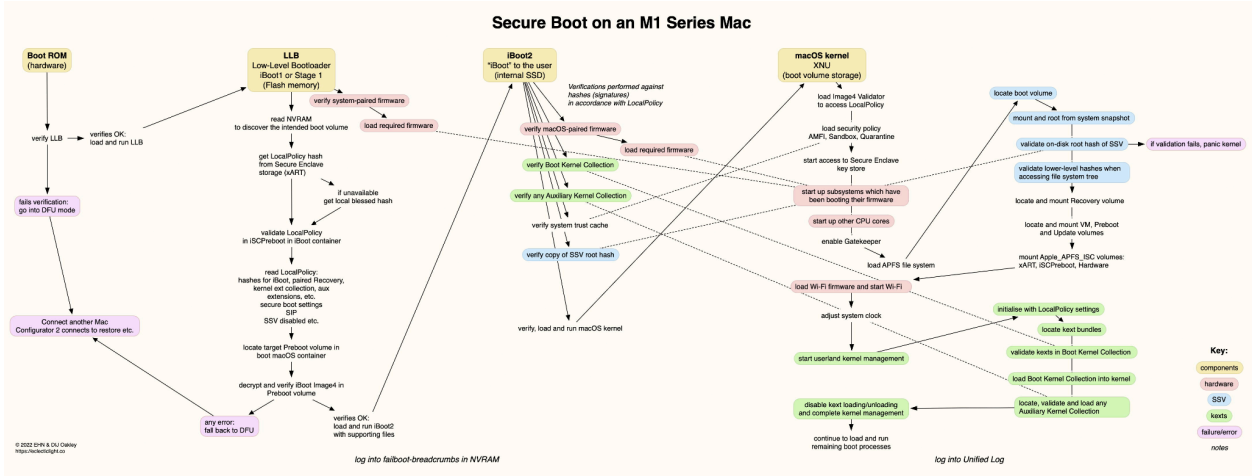


Diagram from Apple Website:
https://support.apple.com/guide/security/boot-process-secac71d5623/web
Diagram from Electric Light Company Website:
https://eclecticlight.co/2022/01/04/booting-an-m1-mac-from-hardware-to-kexts-1-hardwa

re/#:~:text=Booting%20an%20M1%20series%20Mac,await%20a%20connection%20over%20USB.