

SPRINGFIELD AMAZON WEB SERVICES USER GROUP
MARCH 2021

#SGFAWS

APPLE SILICON M1: DEVELOPING ON AN ARM PROCESSOR

ABOUT SPRINGFIELD AWS

- ▶ **Meetup**

<https://meetup.com/sgfaws/>

- ▶ **YouTube**

<https://youtube.com/c/sgfaws>

- ▶ **Discord**

<https://sgf.dev/>



Springfield Amazon Web Services (SGF AWS) User Group is a community-based user group that promotes and advocates for Amazon Web Services in the Springfield Missouri region.

ABOUT JASON KLEIN

- ▶ **20+ years experience in IT (Networks and Linux) and Software Development (PHP/MySQL)**
- ▶ I have been a macOS user for 15 years, starting with Mac OS X Tiger (10.4). I began using PhpStorm IDE about 5 years ago.
- ▶ I have been deploying projects to Intel Linux for 20+ years. Red Hat (2001-'12), Debian (2012-'18), Amazon Linux (2018-'20), Amazon Linux 2 (2020-)
- ▶ Follow me!  @JasnK  @jason-klein



AGENDA

- ▶ Terminology
- ▶ History
- ▶ Why ARM?
- ▶ Ecosystem
- ▶ Apple Processor History
- ▶ Apple M1 (ARM) vs Apple Intel (x86)
- ▶ Applications, Containers, Virtual Machines

TERMINOLOGY

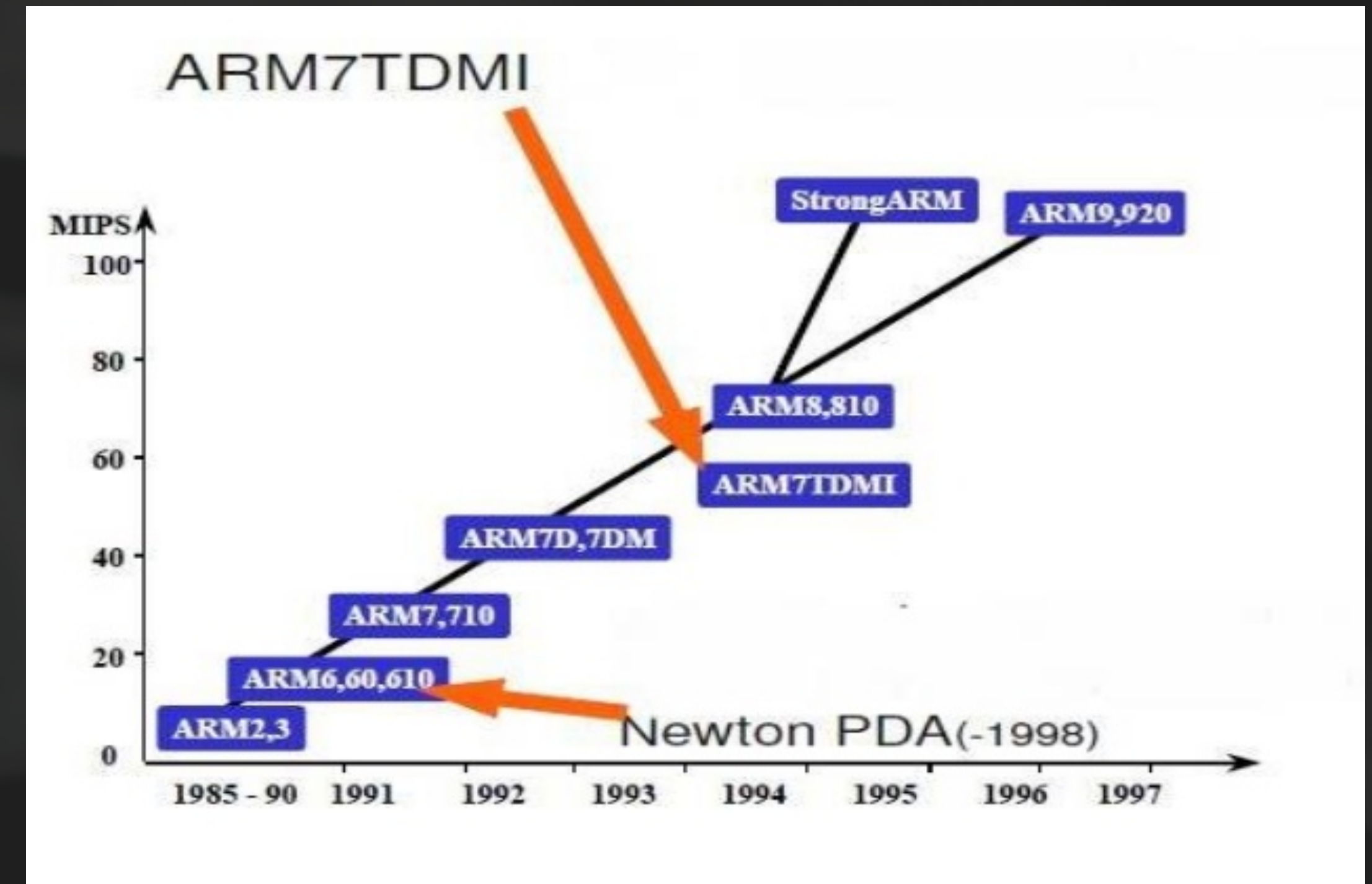
- ▶ **Advanced RISC Machine (ARM)**
- ▶ **Reduced Instruction Set Computer (RISC)**



Hackers 1995

HISTORY OF ARM PROCESSORS

- ▶ 1981 Acorn Computer introduced "BBC Micro" CPU
- ▶ 1983 Began Acorn RISC Machine (ARM)
- ▶ 1985 Completed "ARM1" processor design
- ▶ 1990 Advanced RISC Machines Ltd. (Arm, Ltd.) established by Acorn Computer Group, Apple Computers, and VLSI
- ▶ 1991 ARM (Arm, Ltd.) created and licensed the new ARM6 microprocessor standard (Apple Newton)
- ▶ 2005 98% of all mobile phones sold used at least one ARM processor



Early ARM History (1985-1997)

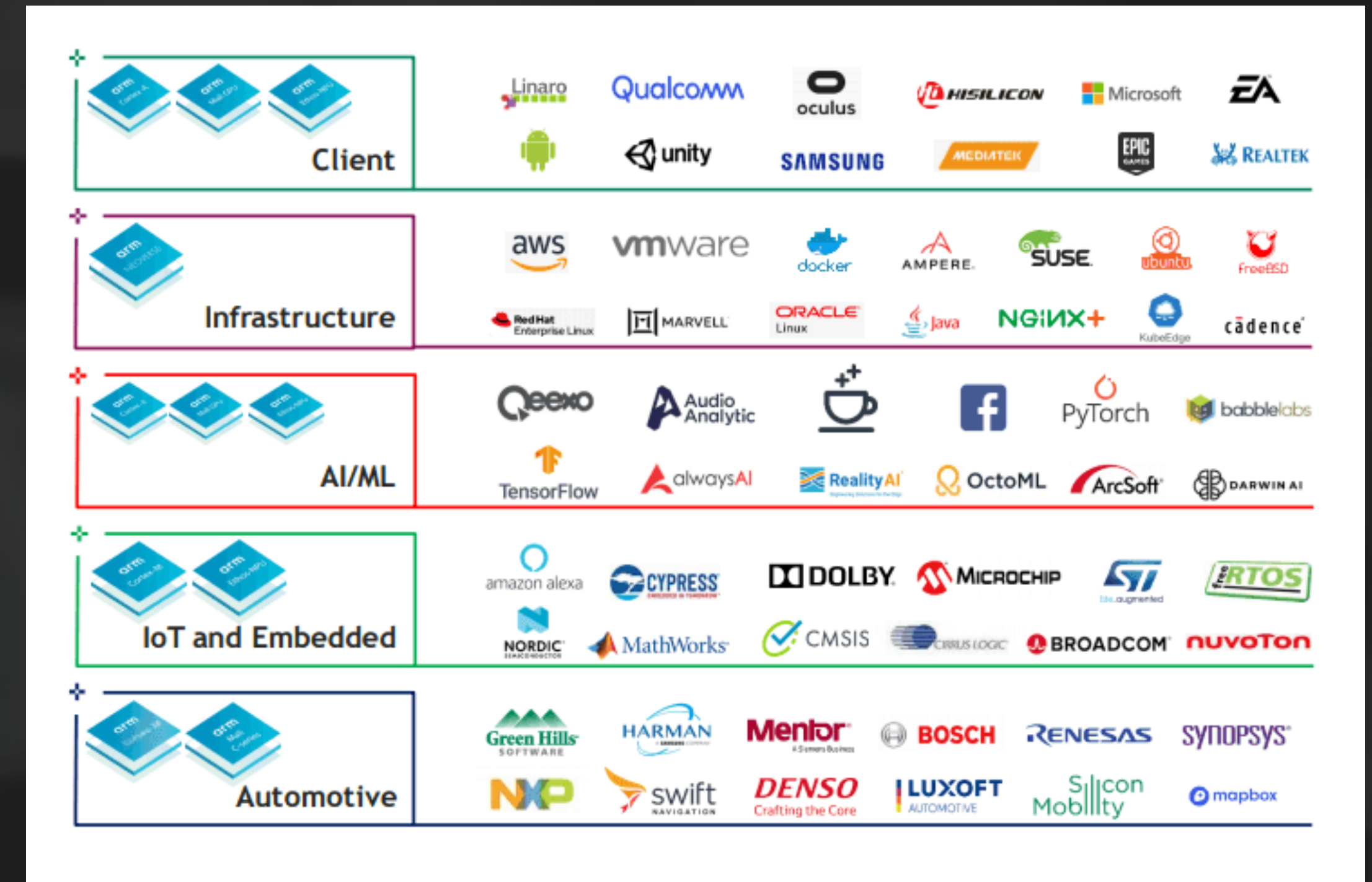
WHY USE ARM PROCESSORS?

- ▶ License architecture from Arm, Ltd., rather than purchasing processors from Intel or AMD. Licensees typically use reference architecture when designing their own Systems-on-Chips (SoC)
- ▶ Low costs, minimal power consumption, and lower heat generation than their competitors. Bonus: longer battery life!

Learn More https://en.wikipedia.org/wiki/ARM_architecture

ECOSYSTEM

- ▶ Clients (Phone, Tablet, VR)
- ▶ Infrastructure (Hosting Servers)
- ▶ AI/ML (Super Computers)
- ▶ IoT and Embedded (Smart Speakers)
- ▶ Automotive (Infotainment, Navigation)



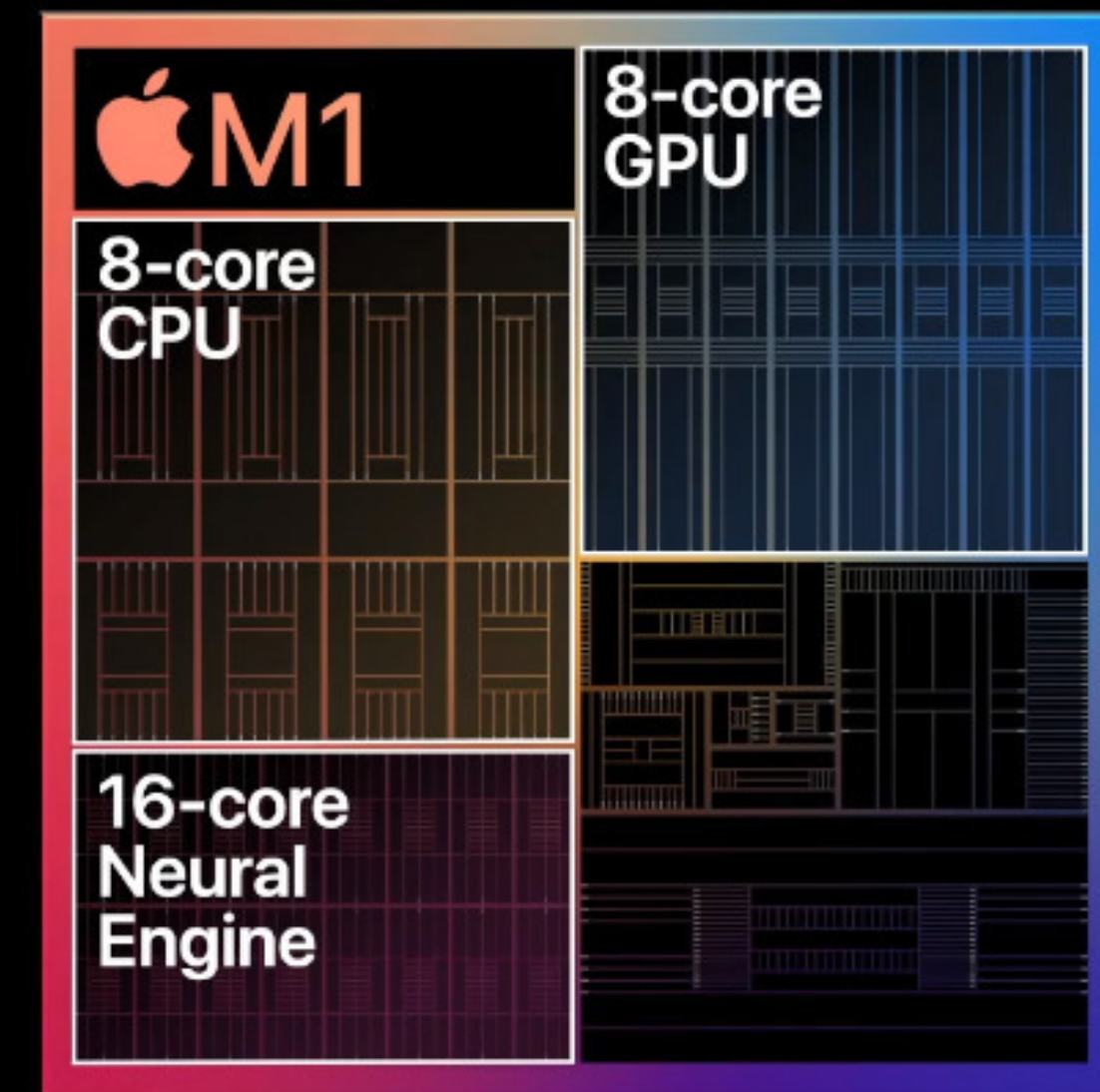
ARM Ecosystem

APPLE PROCESSOR HISTORY

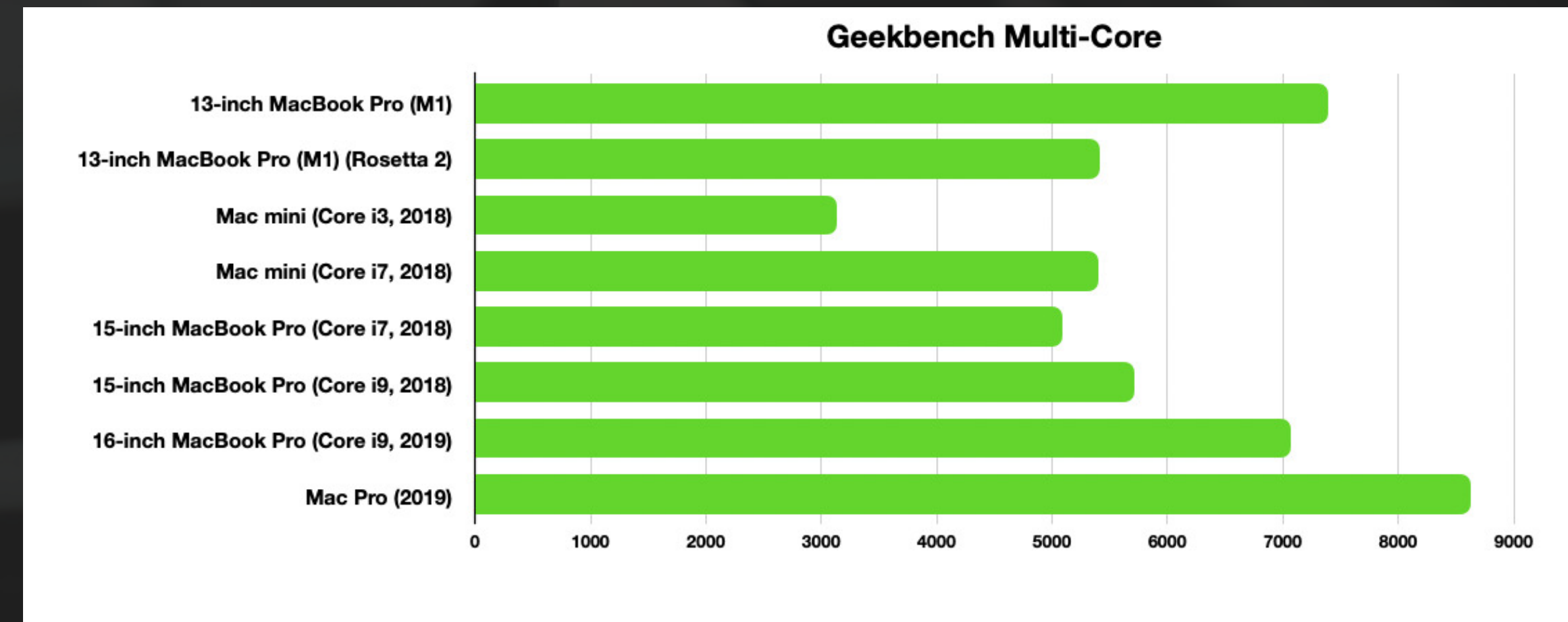
- ▶ **1984-1995 Apple Macintosh (Motorola 68000 RISC)**
- ▶ **1994-2005 Apple Power Macintosh (32-bit PowerPC G2/G3/G4 RISC)**
... Rosetta assists with transition from PowerPC to Intel
- ▶ **2006-2011 Apple Computers (32-bit Intel x86)**
- ▶ **2007-2021 Apple Computers (64-bit Intel x86)**
... Rosetta 2 assists with transition from Intel to ARM
- ▶ **2020-? Apple Computers (64-bit Apple M1 ARM)**

APPLE M1 (ARM) vs APPLE INTEL (x86)

- ▶ System on Chip (M1) vs Components (Intel)
- ▶ Shared CPU and GPU Memory
- ▶ High Performance + High Efficiency CPU Cores
- ▶ 15X Machine Learning Performance (e.g. face detection in Mac Photos?)
- ▶ User Experience similar to iPad/iPhone (e.g. Instant On, near instant app loading/response)



M1 Architecture

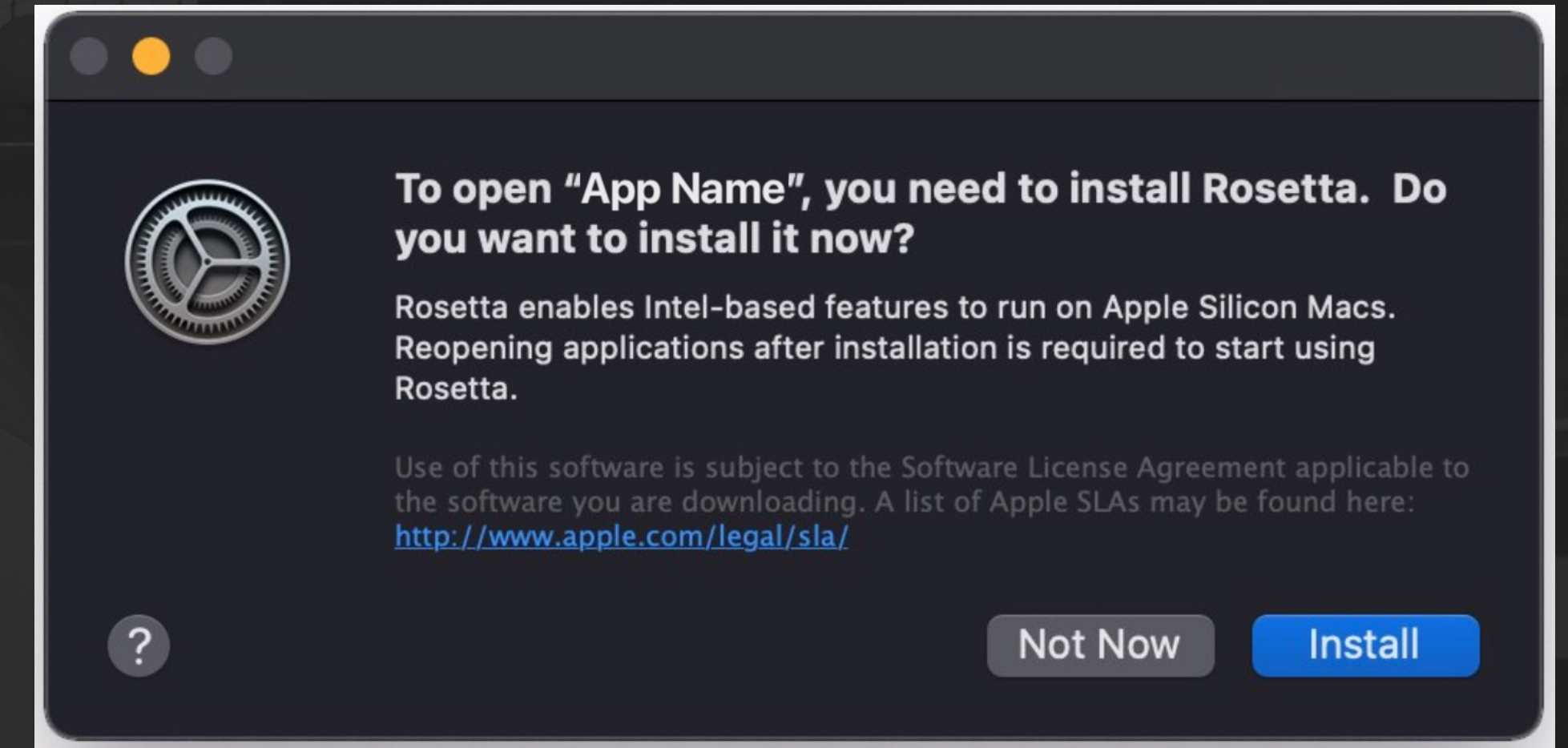


Geekbench Multi-Core (M1 vs Intel)

Learn More <https://appleinsider.com/articles/20/11/17/m1-benchmarks-proves-apple-silicon-outclasses-nearly-all-current-intel-mac-chips>

INTEL APPLICATIONS

- ▶ Rosetta 2 install automatically suggested the first time you run an Intel application
- ▶ Ahead-of-Time Translation (AOT) is performed the first time an Intel application is launched, translating Intel code to ARM code
- ▶ Just-in-Time Translation (JIT) is also available (e.g. if you run an Intel based browser with a JIT JavaScript engine)
- ▶ Overall, magical experience! Everything just works (including IDE, build tools, etc). Very surprised and impressed.

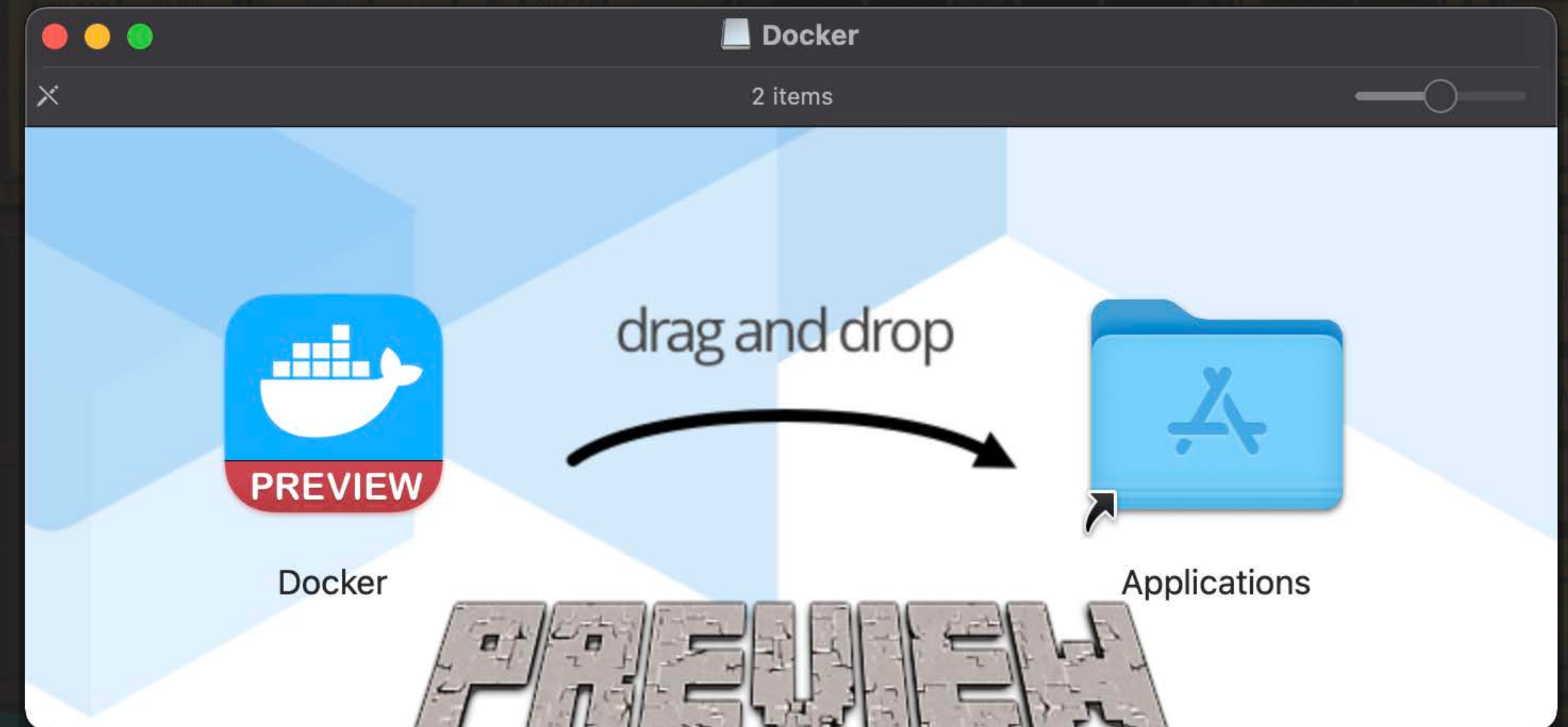


Rosetta 2 prompt when installing Intel application



INTEL CONTAINERS

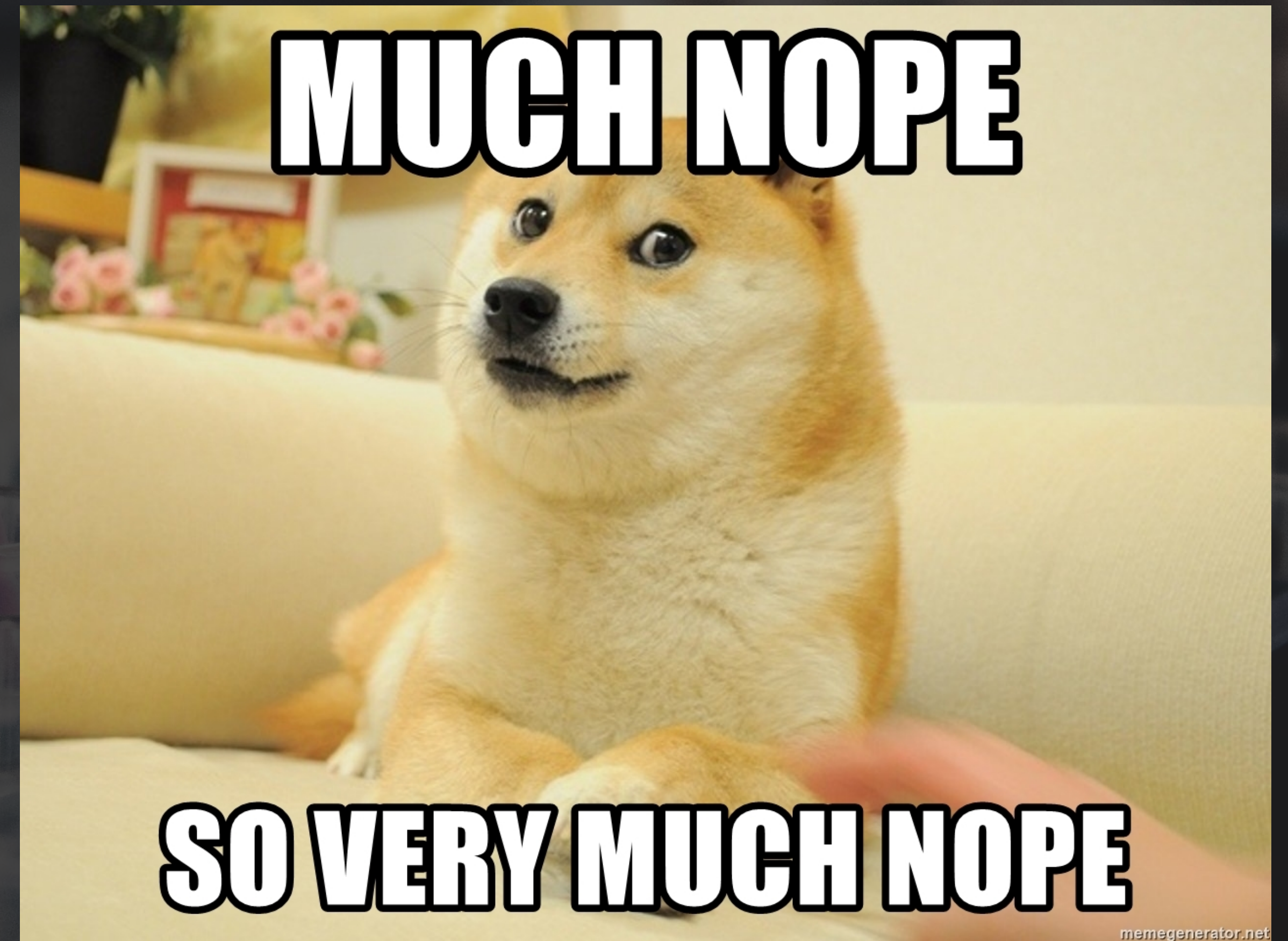
- ▶ ARM64 containers “are also faster and use less memory than Intel-based containers”
- ▶ Not all images are available for ARM64 (e.g. MySQL). Use “`--platform linux/amd64`” to run an Intel image under emulation
- ▶ You may need to build/test/deploy images to Intel and ARM. Use Docker Buildx (Cross CPU Architecture Builds) to build Intel or BOTH Intel and ARM...
`docker buildx build --platform linux/amd64,linux/arm64 --push -t example/example .`



Installing Docker Preview for M1

INTEL VIRTUAL MACHINES

- ▶ Expect virtualization products for Apple M1 from Parallels and VMware, NOT VirtualBox
- ▶ Virtualization Products != Intel Emulation
- ▶ Apple M1 can run Linux ARM, Windows ARM, and macOS ARM in a Virtual Machine
- ▶ Windows ARM can only run ARM apps. MS working on emulator for 64-bit Intel apps
- ▶ Apple M1 (ARM) would need an EMULATOR (e.g. QEMU) to run Linux Intel, Windows Intel
- ▶ CrossOver for Mac can run some Windows Intel apps



Nope!

BONUS: HOMEBREW

- ▶ Native support for Apple M1 in Homebrew 3.0.0 as of February 2021
- ▶ Not all Homebrew packages are available for ARM
- ▶ You can install Homebrew ARM (/opt/homebrew) and Homebrew Intel (/usr/local/bin) side-by-side since they live in separate directories

APPLE M1 #WTF

- ▶ Several Intel Applications displayed an error at end of installation, but they ran fine after install.
- ▶ Upgrade from macOS 11.1 to 11.2 failed, stuck in looped boot sequence. Unable to repair or recover. Unable to boot to safe mode. Wiped and reinstalled. Unclear if this was M1 specific. Details...
<https://www.reddit.com/r/MacOS/comments/lgbkx7/>
- ▶ Docker Desktop Preview and Laravel Valet not compatible at first, resolved in latest Docker Preview.
<https://github.com/laravel/valet/issues/1031>
- ▶ Why only two USB-C ports?? Two Ports + Dongle Hell = GRRR!
- ▶ Adobe has NOT released native Acrobat, Photoshop, Illustrator, etc. Almost all my other apps are native!
- ▶ Why is the Microsoft Intel emulation for Windows ARM not ready? I could run Windows Intel apps on Windows ARM in a VM now if this was complete. Microsoft Surface ARM users must be disappointed!

APPLE M1 #FTW

- ▶ User Interface response time seems much more like an iOS device. Instant on. Instant load.
- ▶ Intel applications are fast. Native applications are MUCH faster!
<https://twitter.com/JasnK/status/1356650130804535298>
- ▶ Always cool/cold, except ONE time Adobe Core Sync was misbehaving and ran 4 processes at 100% CPU each for a LONG time. #WhatIsThatNoise #OhItDoesHaveAFan #HotHotHot #WTF
https://www.reddit.com/r/macbookpro/comments/lhks71/after_just_5_hours_of_work_the_battery_is_down_to/gn0sod0/
- ▶ Still surprised I have not encountered issues running Intel apps on M1. Kudos Rosetta 2 team!
- ▶ Super impressed I am able to build Intel Docker images. We moved all of our containers to ARM, so this is no longer relevant. Join us next month for details!

RESOURCES

- ▶ Mac Application Compatibility with M1
<https://isapplesiliconready.com/>
- ▶ Docker Desktop for M1 (preview)
<https://docs.docker.com/docker-for-mac/apple-m1/>
- ▶ Parallels Desktop for M1 (preview)
<https://www.parallels.com/blogs/parallels-desktop-apple-silicon-mac/>
- ▶ Use built-in macOS Virtualization Framework to boot an Ubuntu Linux VM
<https://github.com/kendfinger/virtual>
- ▶ Boot an Apple M1 computer directly to Linux <https://corellium.com/blog/linux-m1>

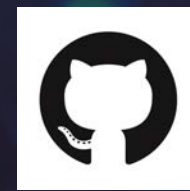
Questions?

Discuss in our Springfield Devs Discord channel .. #aws

▶ Follow me!



@JasnK



@jason-klein

SPRINGFIELD AMAZON WEB SERVICES USER GROUP
MARCH 2021

#SGFAWS

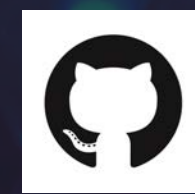
APPLE SILICON M1: DEVELOPING ON AN ARM PROCESSOR

Thank you!

► Follow me!



@JasnK



@jason-klein

SPRINGFIELD AMAZON WEB SERVICES USER GROUP
MARCH 2021

#SGFAWS

APPLE SILICON M1: DEVELOPING ON AN ARM PROCESSOR