From proprietary to fully open-source: Arm Toolchain’s adoption of LLVM technology

US LLVM Developers’ meeting 2025

Peter Smith

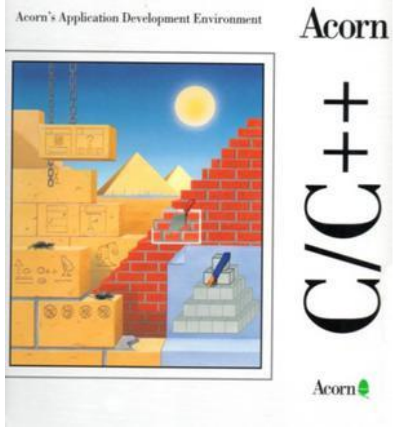
2025-10-05

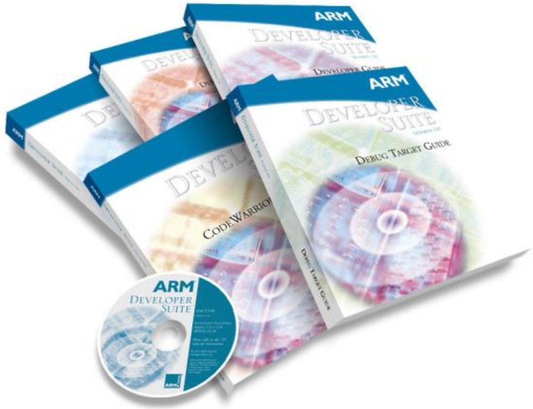
Arm history from 25 BC (before clang) AKA 1985

Acorn Archimedes

GSM mobile phone

Symbian phone





Arm Toolchain evolution

Pictures from Wikipedia, sources in references

A perfect storm for armcc. 5 BC (Before Clang)

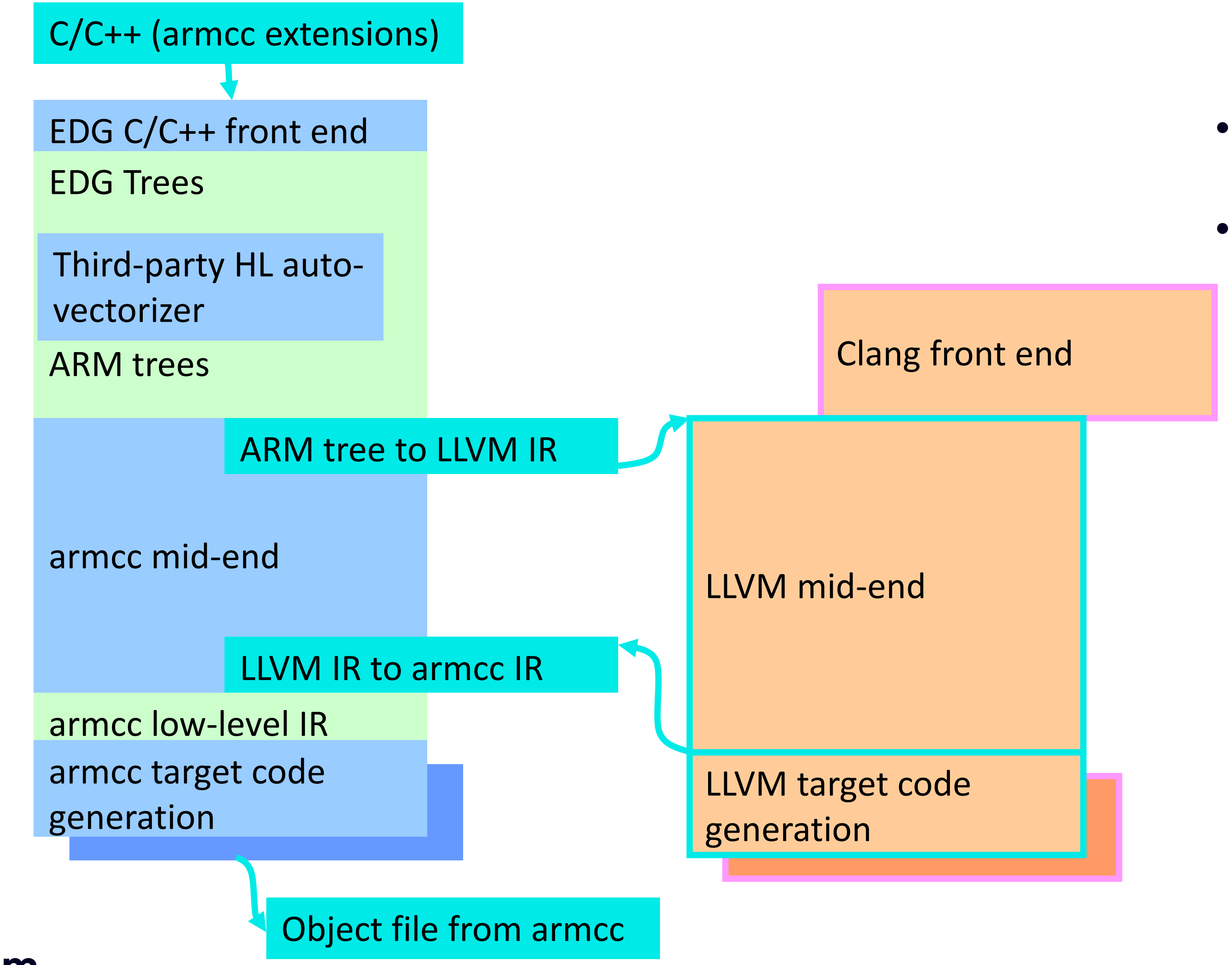
- Industry moving towards software platforms or microcontrollers.
- Out-of-order CPUs and Thumb-2 erode code-generation advantage.
- C++11 a major upgrade.
- Good GCC support for Arm necessary but not sufficient
- Proprietary nature of armcc prevents collaboration with partners.
- This LLVM thing looks interesting!

Why was Arm interested in LLVM in 2009?

- Technology and community, developing fast, but unproven for Arm.
- Experiment to integrate LLVM into armcc started in late 2009.

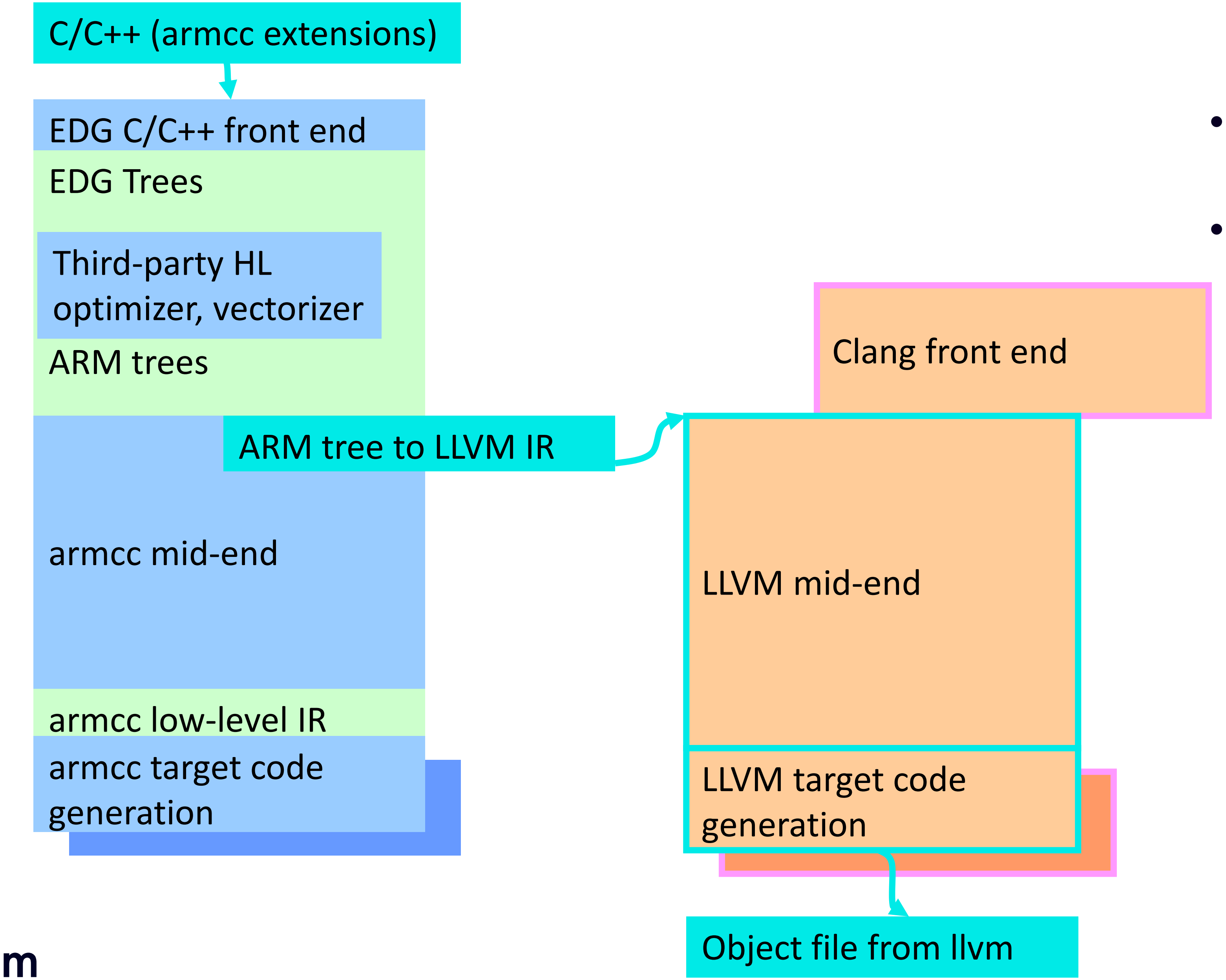
	Technology	**Community**
Strengths	• Modern C++ code-base. • Mid-end optimizations. • Arm support.	• License. • Modular design.
Weaknesses	• Auto-vectorization. • Windows support. • Arm backend-maturity.	• Self sustaining Arm community not a given.

EDG to LLVM Bridge experiment 2009 - 2011



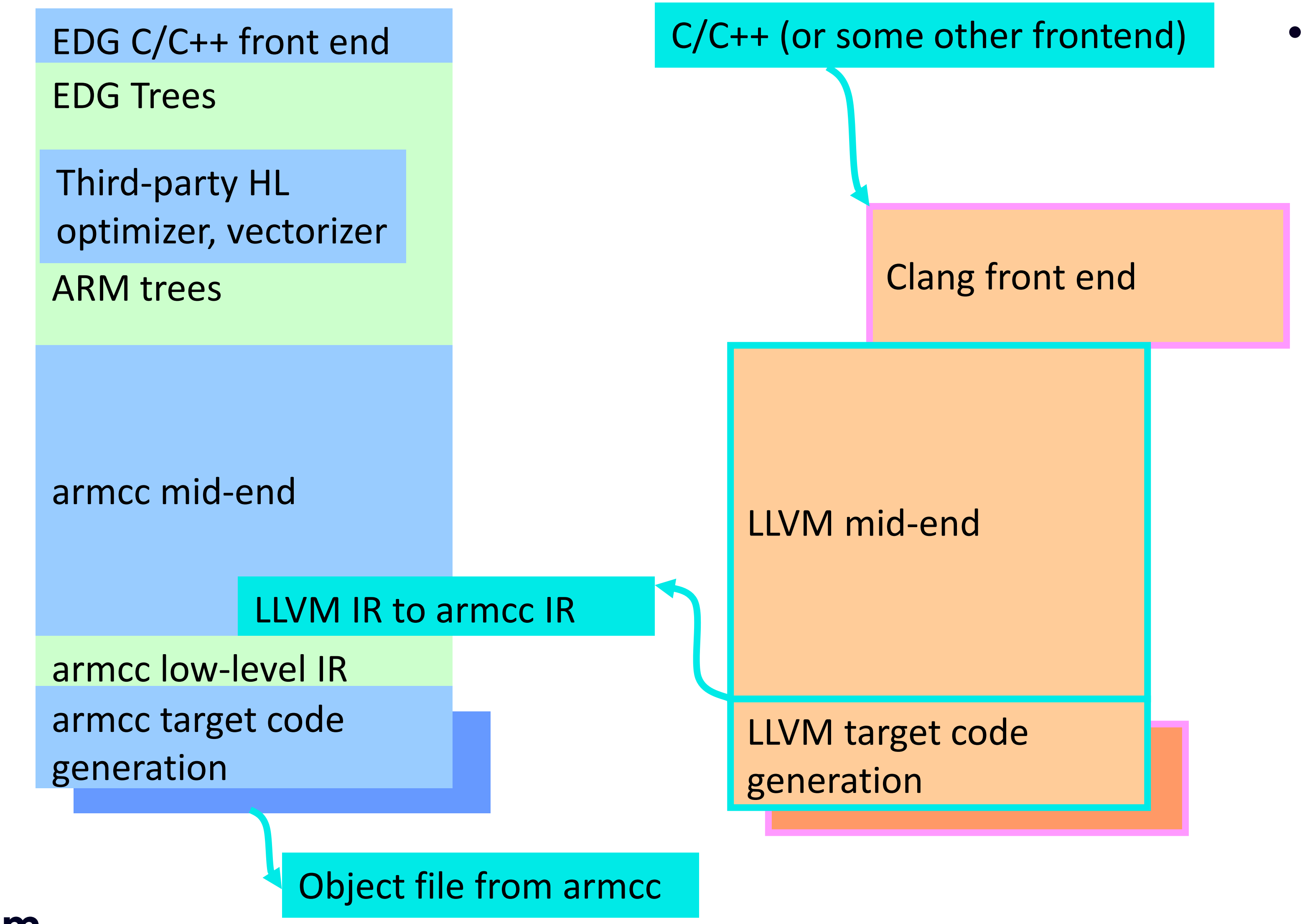
- EDG a licensed C++ front-end.
- Use LLVM as a replacement mid-end.

EDG to LLVM Bridge experiment 2009 - 2011



- Armcc front-end only.
- Implemented in prototype

EDG to LLVM Bridge experiment 2009 - 2011



- LLVM as a bridge to armcc backend.

LLVM community progress during prototype

Welcome to the LLVM Blog

Clang Self hosts

Intro to MC

Introducing libc++

Clang builds

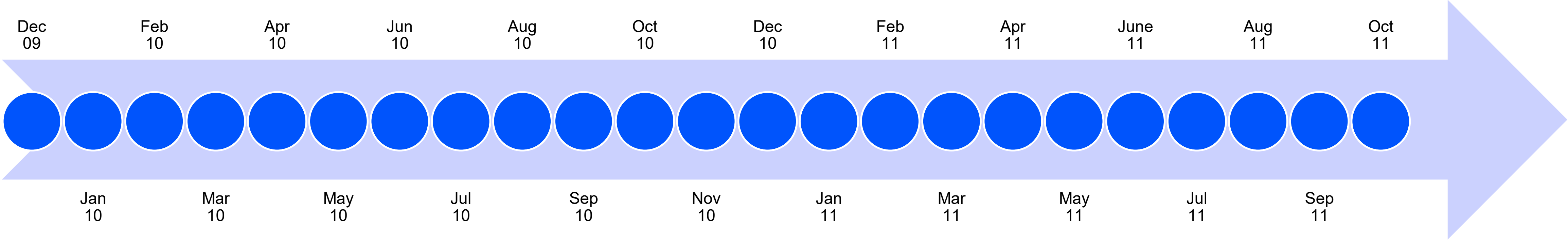
boost

Improvements to Arm backend

Greedy register allocator

European user group meeting (Proto EuroLLVM)

LLVM community



EDG to LLVM IR bridge starts

Prototype 1st release

EEMBC, Spec, Plum Hall

Prototype closes

Arm internal

2009 LLVM Developer Meeting, maintainers round table

> The volatility of the C++ API is **intentional**:
>
> - it allows for faster evolution of the design.
> - it is **a strong encouragement** for LLVM users to **contribute** their improvements to the project: any change not contributed is likely to break at the next release and will increase maintenance cost.

Lessons learned from the prototype

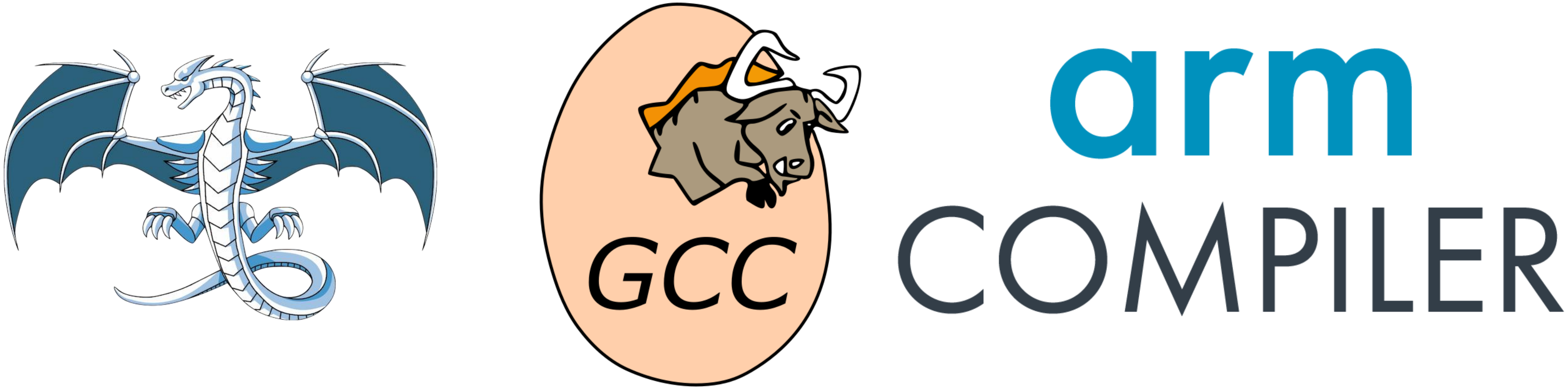
- Across the whole toolchain the community will move faster than you can.
- It pays to align yourself as closely with upstream with technology changes.
- Confirmation of hypothesis that LLVM stronger at mid-end optimizations, but back-end immature.

Arm joins the open-source LLVM community. Our priorities

- First class support for Arm in LLVM
  - Arm build bots set up, maintained and responded to.
  - Improve completeness and correctness of assembly/disassembly.
- Build and coordinate a community around LLVM for Arm
  - Sponsor and organize initial European Developer Meetings.
- Add support for AArch64
- Increase Arm's LLVM expertise.
- Participate and build influence in the community.

Come in armcc, your time is up.

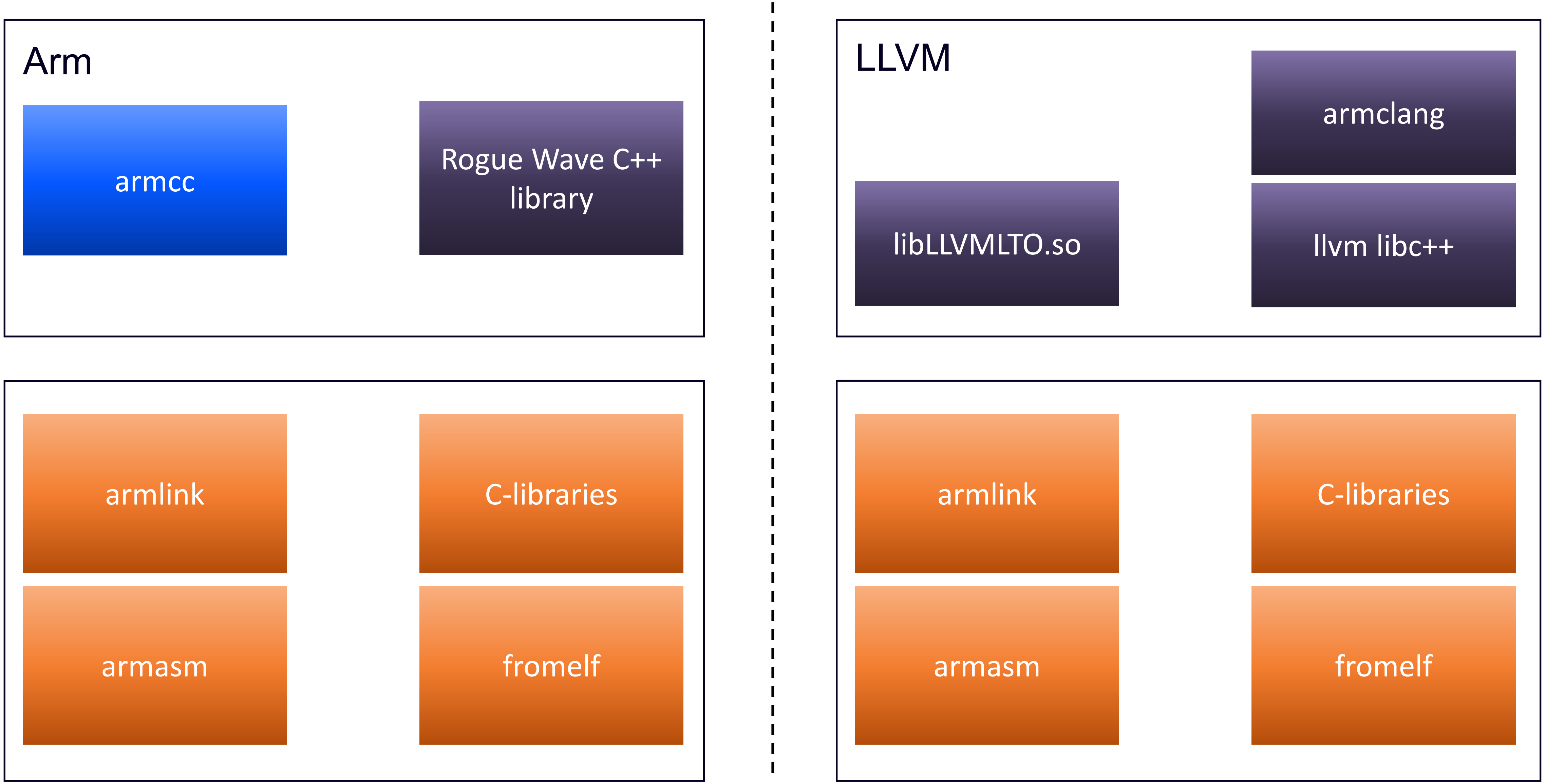
- First class support for Arm architecture needed in both LLVM and GNU ecosystems.
  - Must support the platform compilers for the dominant software platforms.
  - Maintaining 3 compiler technologies at an Arm sized company is not an option.
- Over several years, replace proprietary compiler with LLVM
  - License and technology make LLVM the preferred option.
  - Expect several years of development to be competitive with armcc.
- Maintain, proprietary toolchain during transition.



Early Arm Compiler 6 design decisions 0 BC (AKA 2014)

- Only new toolchain would get Armv8 (AArch64) and C++ 11 library support
    - Remove competition from old toolchain.
    - Toolchain migration only a small amount of extra work.
- Live at head. Continuously merge from upstream.
    - Permit upstream first development.
    - Ease upstreaming of new architecture support when made public.
    - Get benefits of latest upstream features.
- Minimize implementation of armcc specific extensions in clang
    - Emulate with existing clang features.
- Optimize and tune LLVM components for embedded market
    - Embedded benchmarks.
    - Code-size.
- Share linker, binutils and libraries with proprietary toolchain
    - LLD development in 2014 not looking promising.
    - LLVM binutils aimed at toolchain developers not users.
    - LLVM libc didn’t exist.

From Arm Compiler 5 (armcc) to Arm Compiler 6 (armclang)



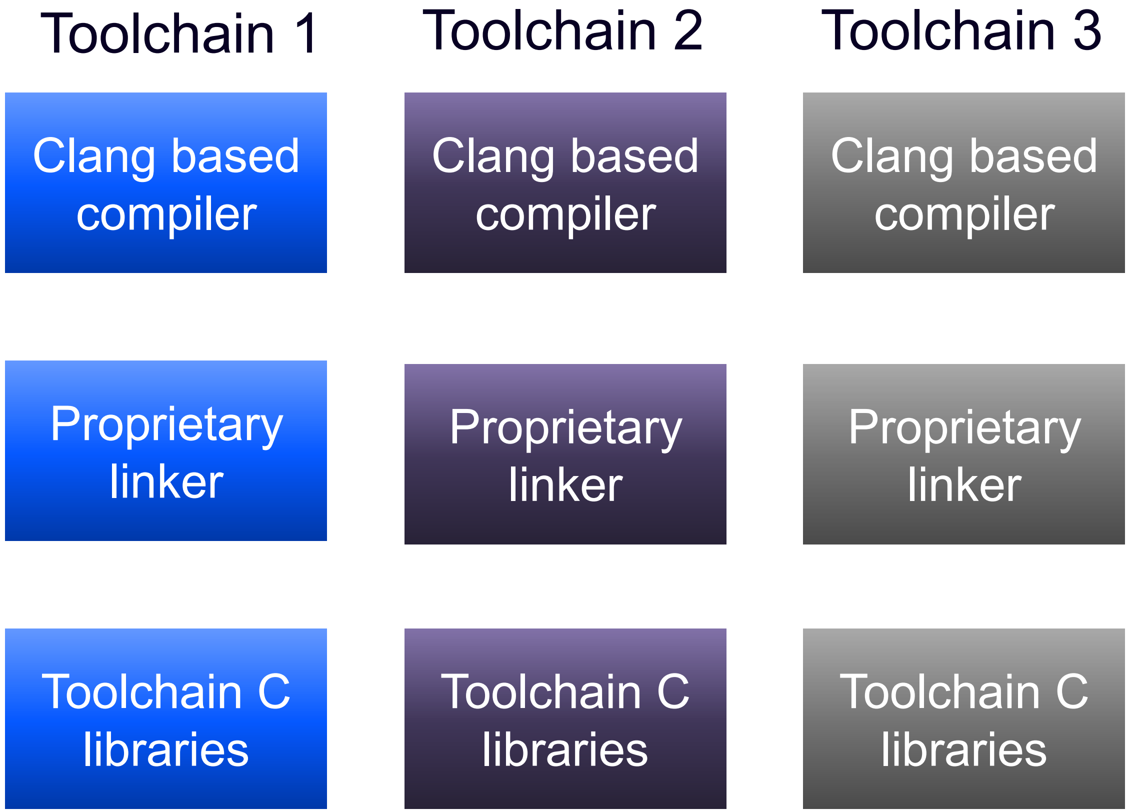
Aside; why the proprietary binutils and downstream changes?

- We’re all encouraged to contribute our changes back to upstream
  - “any change not contributed is likely to break at the next release and will increase maintenance cost.”
- What if upstream doesn’t want your contributions?
  - Community must take on maintenance burden for an unbounded amount of time.
  - Increase of implementation complexity needs to be justified.
- Upstream may not have experience in the domain
  - Do I understand the change, can I test it?
  - Who uses this anyway?

Embedded toolchains and upstreaming 2014 to 2020

- Several embedded toolchains using clang as compiler.
- Similar toolchain integration, optimization for code-size and extensions.
- No common standard that can be upstreamed.
- No one size fits all solution.
- No software platform driving convergence.

- At least not yet!



Toolchain 1	Toolchain 2	Toolchain 3
Clang based compiler	Clang based compiler	Clang based compiler
Proprietary linker	Proprietary linker	Proprietary linker
Toolchain C libraries	Toolchain C libraries	Toolchain C libraries

Arm Embedded software development needs open-source LLVM

Open-source software platforms and libraries emerge

- Open-source embedded software designed for open-source tools.
- LLVM has better code-generation for new Arm microcontrollers.
- Qualified compiler for functional safety environments.
- Requirements for use of clang security features in firmware.

Zephyr®

freeRTOS

CMSIS

yocto PROJECT

TrustedFirmware .org

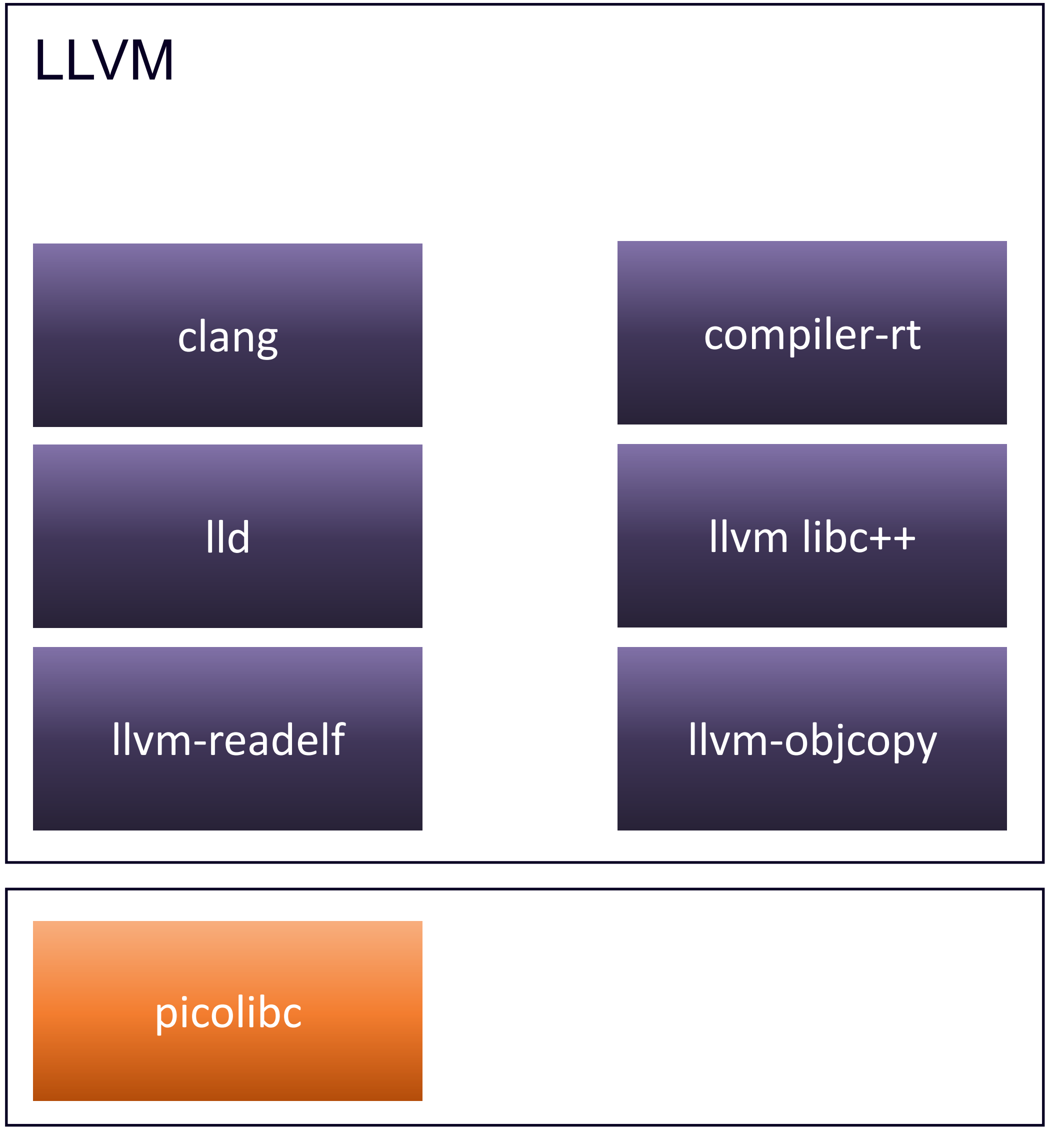
Convergent evolution towards GNU compatibility in linker and binutils

- GNU compatible linker scripts in lld and compatible binutils a requirement.
- Able to hitch a ride with larger adjacent communities
  - BSD, Sony, Google and Clang built Linux drive improvements in GNU compatibility.
  - Linux kernel; one of the world’s most demanding embedded systems can be built with llvm.

Year	Event
2014	FreeBSD 10 adopts LLVM Chrome Linux Builds use LLVM
2015	LLD ELF
2016	llvm-objcopy
2017	Google Summer of Code binutils project
2019	Clang Built Linux with clang and llvm binutils
2020	BareMetal Driver development pace increases.

LLVM Embedded Toolchain for Arm (2022)

- An additional toolchain to Arm compiler 6.
- Open-source repository of build scripts.
  - Checkout and patch llvm-project and picolibc.
- Aimed at microcontrollers (M-profile)
- Lack of multilib support a key weakness
  - Config files used for library variants.
  - 33 config files used for selection.



Finding and growing the LLVM embedded community

Community call

Monthly call started in 2022-03-03

Coordinate reviews, RFCs.

Now replicated for many communities.

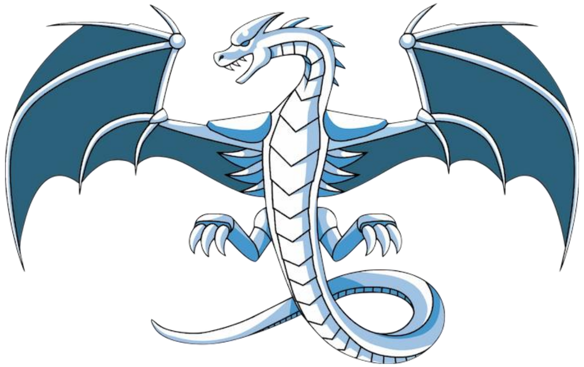
Conferences

On 3rd Embedded Systems Workshop at US Developer Meeting.

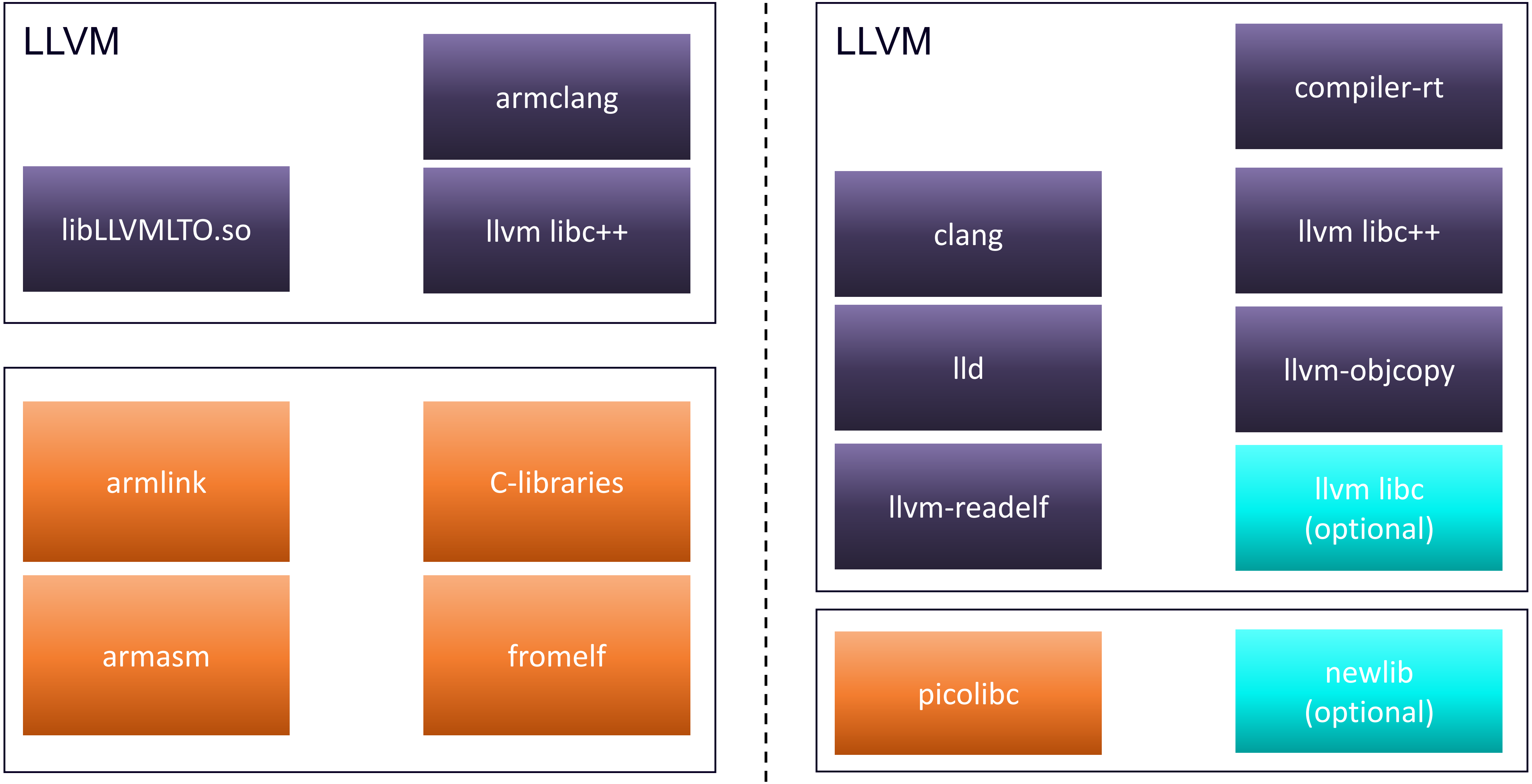
Presentation and discussions at FOSDEM and EMBO 2023

Going fully open-source, replacing the proprietary toolchain

- First class support for Arm architecture needed in both LLVM and GNU ecosystems.
  - Must support the platform compilers for the dominant software platforms.
- Yet again, Arm does not want to support 3 toolchains.
- Arm-toolchain replaces LLVM embedded toolchain for Arm
  - Nightly builds with upstream CI.
  - Full releases to coincide with numbered LLVM releases.
- Aiming for an embedded toolchain built entirely from llvm-project
  - Adopt llvm-libc as the default C-library.



From Arm Compiler 6 to Arm Toolchain for Embedded.



LLVM

armclang

libLLVMLTO.so

llvm libc++

armlink

C-libraries

armasm

fromelf

LLVM

compiler-rt

clang

llvm libc++

lld

llvm-objcopy

llvm-readelf

llvm libc (optional)

picolibc

newlib (optional)

Lessons learned over 15 years of LLVM Contributions

Not just technology community is important too.

Aligned open-source communities go faster

Find and develop your community

Share goals with adjacent communities

Context and Community changes over time

arm

Merci
Danke
Gracias
Grazie
谢谢
ありがとう
Asante
Thank You
감사합니다
धन्यवाद
Kiitos
شكرًا
ধন্যবাদ
תודה
ధన్యవాదములు
Köszönöm

References

- LLVM Project Blog
  - https://blog.llvm.org/
- LLVM Binutils Euro LLVM BOF
  - https://llvm.org/devmtg/2019-04/slides/BoF-HendersonRupprecht-LLVM_binutils.pdf
- Living Downstream Without Drowning
  - https://llvm.org/devmtg/2015-10/slides/RobinsonEdwards-LivingDownstreamWithoutDrowning.pdf
- Norcroft/armcc history
  - https://www.youtube.com/watch?v=hwUoVU_XCis
- Arm history
  - https://newsroom.arm.com/blog/arm-official-history
  - https://arstechnica.com/gadgets/2022/09/a-history-of-arm-part-1-building-the-first-chip/
  - https://arstechnica.com/gadgets/2022/11/a-history-of-arm-part-2-everything-starts-to-come-together/
  - https://arstechnica.com/gadgets/2023/01/a-history-of-arm-part-3-coming-full-circle/
- More secure firmware
  - https://security.googleblog.com/2023/02/hardening-firmware-across-android.html
- Open-Source Arm Toolchain, includes AArch64 hosted (Arm Toolchain for Linux) and Embedded
  - https://github.com/arm/arm-toolchain
- ABI and ACLE
  - https://github.com/ARM-software/abi-aa/ and https://github.com/ARM-software/acle
- 16-bit char support in LLVM
  - https://archive.fosdem.org/2017/schedule/event/llvm_16_bit/

Wikipedia references for images

- https://en.wikipedia.org/wiki/Acorn_C/C%2B%2B
- https://en.wikipedia.org/wiki/Symbian
- https://en.wikipedia.org/wiki/Nokia_5110
- https://en.wikipedia.org/wiki/Acorn_Archimedes
- https://en.wikipedia.org/wiki/HTC_Dream
- https://en.wikipedia.org/wiki/Zune
- https://en.wikipedia.org/wiki/PlayStation_Vita
- https://en.wikipedia.org/wiki/ARM_Cortex-M
- https://commons.wikimedia.org/wiki/File:Ships_that_pass_in_the_night,_Atlantic_City,_New_Jersey.png