Linaro Connect
MADRID 2024 | MAY 14-17 2024

arm

Introducing the next generation of Arm Compiler

Will Lovett
Director of Technology Management - Compilers | Arm

Part 1

Why do we need another compiler?

Automakers Face Unprecedented Complexity & Opportunity

A New Approach to Automotive Development is Needed

More Autonomy

Advanced
User Experiences

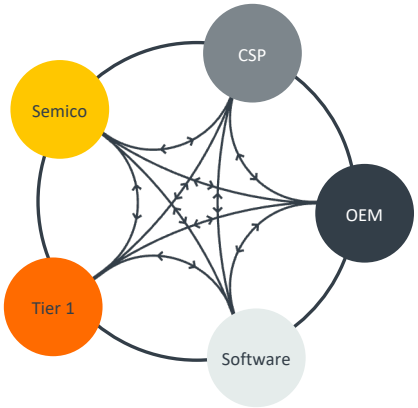
Electrification

=

Growing Software
Complexity

Increased Demand for AI
Capable Compute

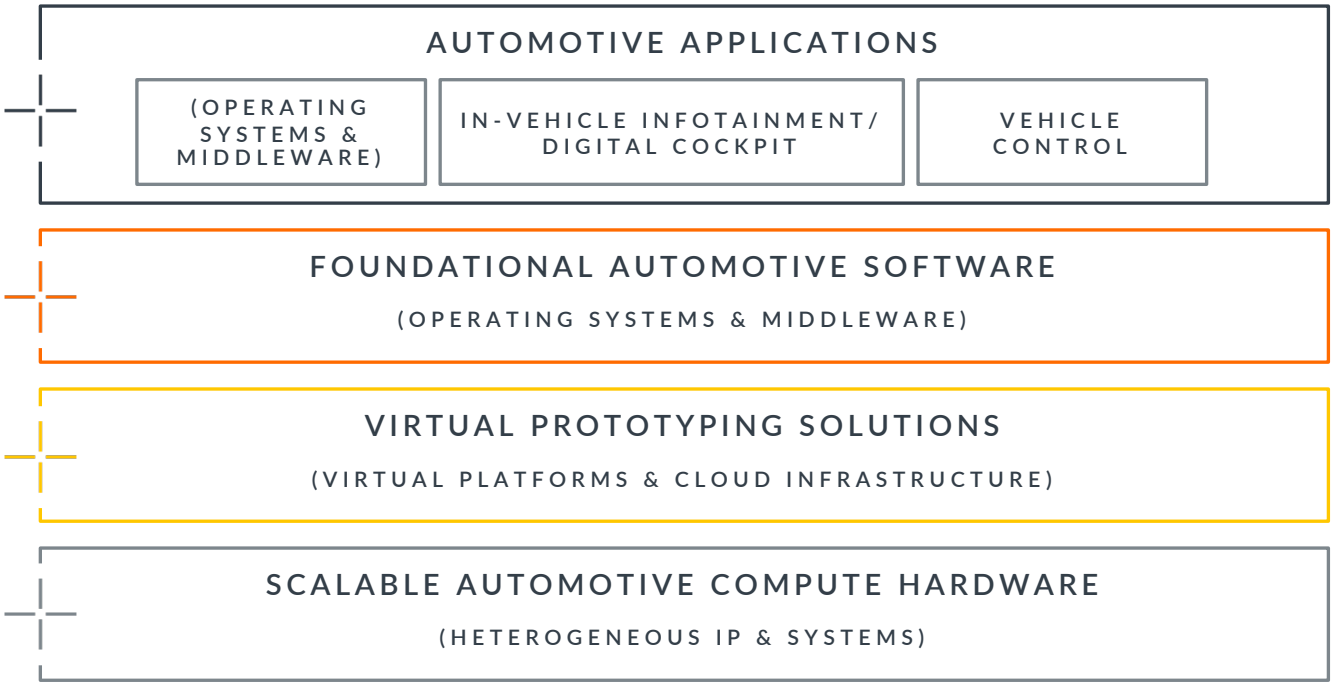
=



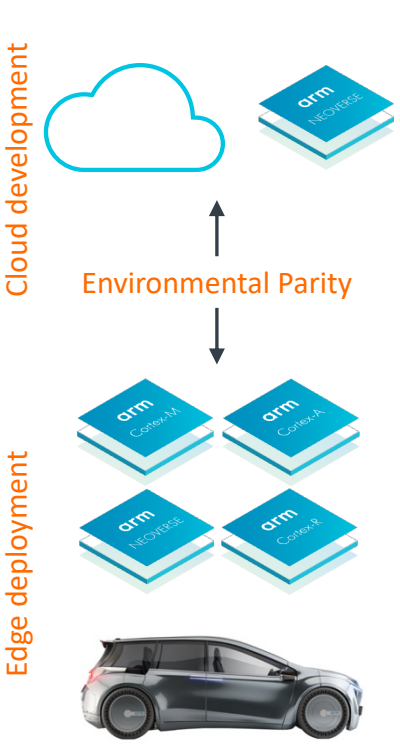
New Challenges and
Opportunities for the Industry
Demand New Solutions

The Automotive Technology Stack

A complex landscape that requires collaboration



Paradigm Shift in Development Methodologies



CI/CD (Continuous Integration / Continuous Delivery)

= container

Infotainment App | ADAS Service

Orchestrator

OS / Container

Virtual Compute Environment

SOAFEE

Containerized App deployment

Infotainment App | ADAS Service

Orchestrator

Container runtime

OS / Hypervisor

SOAFEE

Firmware

Common Silicon Hardware

Case Study | Elektrobit Corbos Linux for Safety Applications

April 23, 2024

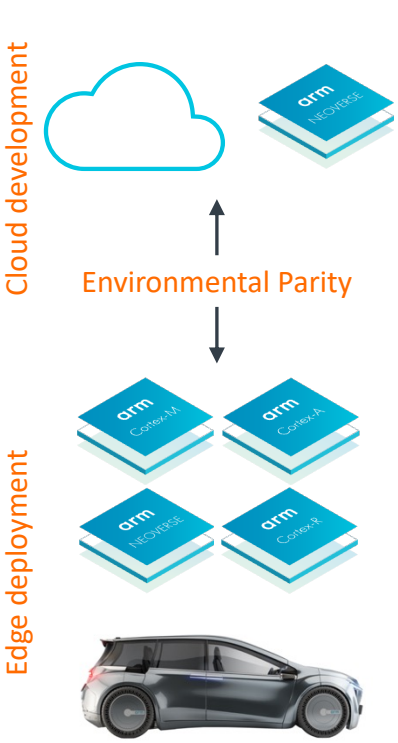
EB corbos Linux for Safety Applications provides the industry with the first open source-based automotive OS solution for safety applications, including full life cycle maintenance.

- Elektrobit are collaborating with Ubuntu to build Corbos Linux for Safety Applications
- This is a functional-safety qualified Linux distribution
  - Hardware
  - Firmware
  - Kernel
  - Libraries
  - Applications

To build this stack, Elektrobit need two toolchains:

- A Functional-safety platform toolchain for building the OS itself
- A functional-safety C++ application toolchain for building Applications

Paradigm Shift in Development Methodologies



CI/CD (Continuous Integration / Continuous Delivery)

= container

Infotainment App | ADAS Service

Orchestrator | OS / Container | Application Toolchain | Virtual Compute Environment | SOAFEE

Platform Toolchain

Containerized App deployment

Infotainment App | ADAS Service

Orchestrator | Container runtime | OS / Hypervisor | SOAFEE

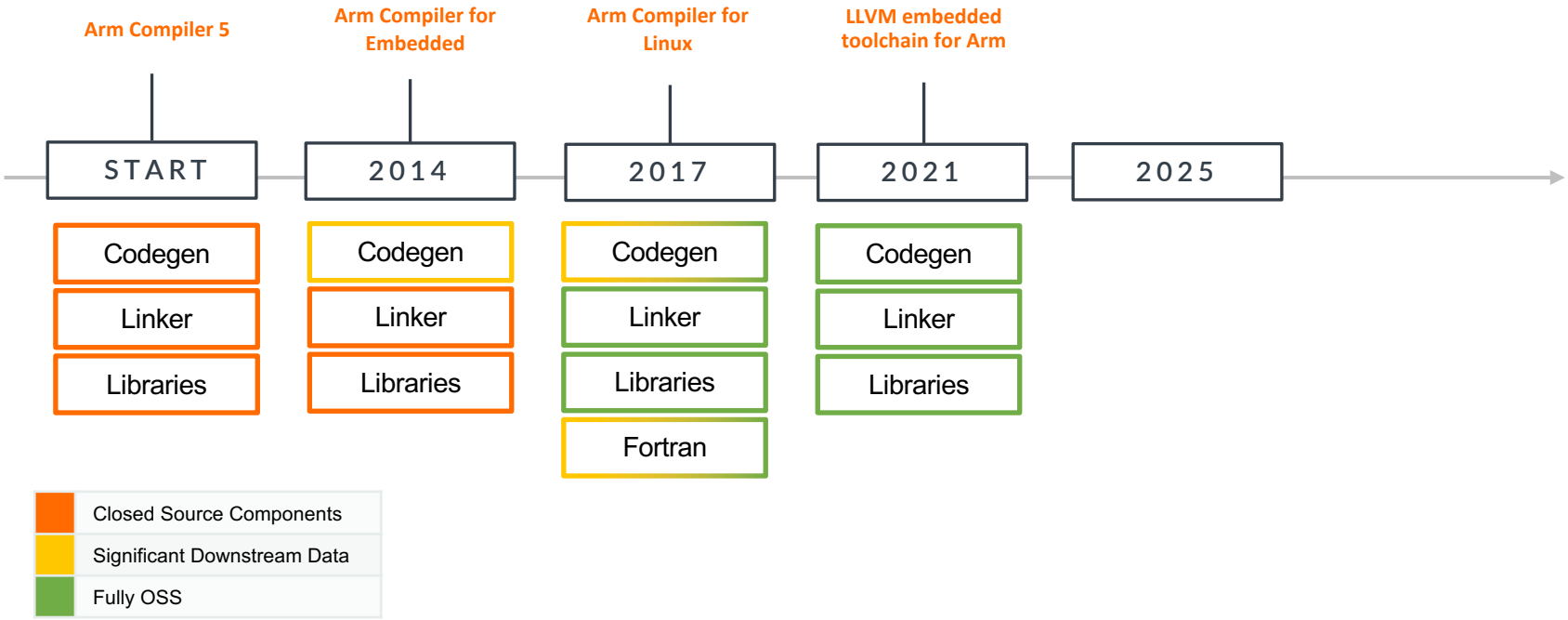
Firmware

Common Silicon Hardware

Part 2

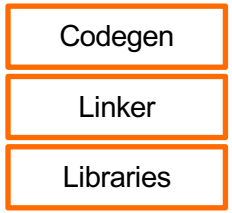
How we’re using LLVM to deliver compilers

A Brief History | LLVM @ Arm



A Brief History | LLVM @ Arm | Arm Compiler 5

Arm Compiler 5

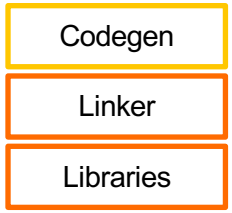


- Designed for baremetal embedded development
- All development is closed-source

A Brief History | LLVM @ Arm | Arm Compiler for Embedded

START | 2014 | 2017 | 2021 | 2025

Arm Compiler for Embedded



- First LLVM-based toolchain from Arm
- Flagship compiler for Architecture, Licensed by all Arm IP licensees
- Designed for embedded development
- Significant downstream delta for codegen
- Linker and libraries are closed source
  - Inherits advance code-placement linker features from Arm Compiler 5
  - Proprietary linker configuration - migration from GNU has a learning curve
- Long-term supported Functional safety (FuSa) releases
  - Safety qualified compiler
  - Safety qualified C libraries for embedded development
  - Long-term support of >10 years

A Brief History | LLVM @ Arm | Arm Compiler for Linux

START | 2014 | **2017** | 2021 | 2025

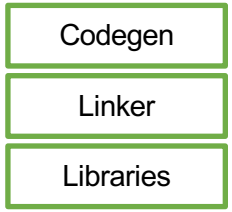
Arm Compiler for Linux

- Designed for HPC and server application development on Linux
- Codegen now fully upstream
- Uses Linux system linker
- Most libraries now upstream
  - Arm Optimized Routines (link)
  - glibc
- LLVM Fortran project progressing well
  - Beta release this year

A Brief History | LLVM @ Arm | LLVM Embedded Toolchain

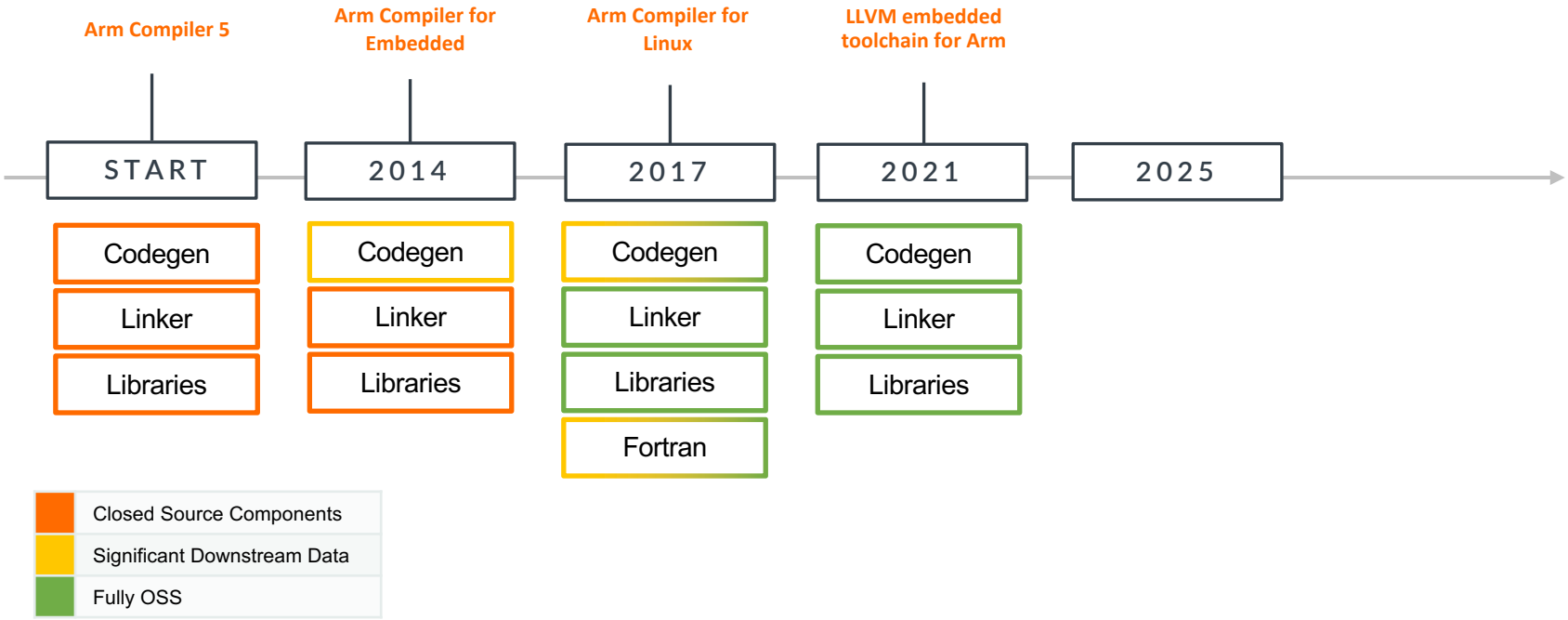
START — 2014 — 2017 — **2021** — 2025

LLVM Embedded Toolchain for Arm

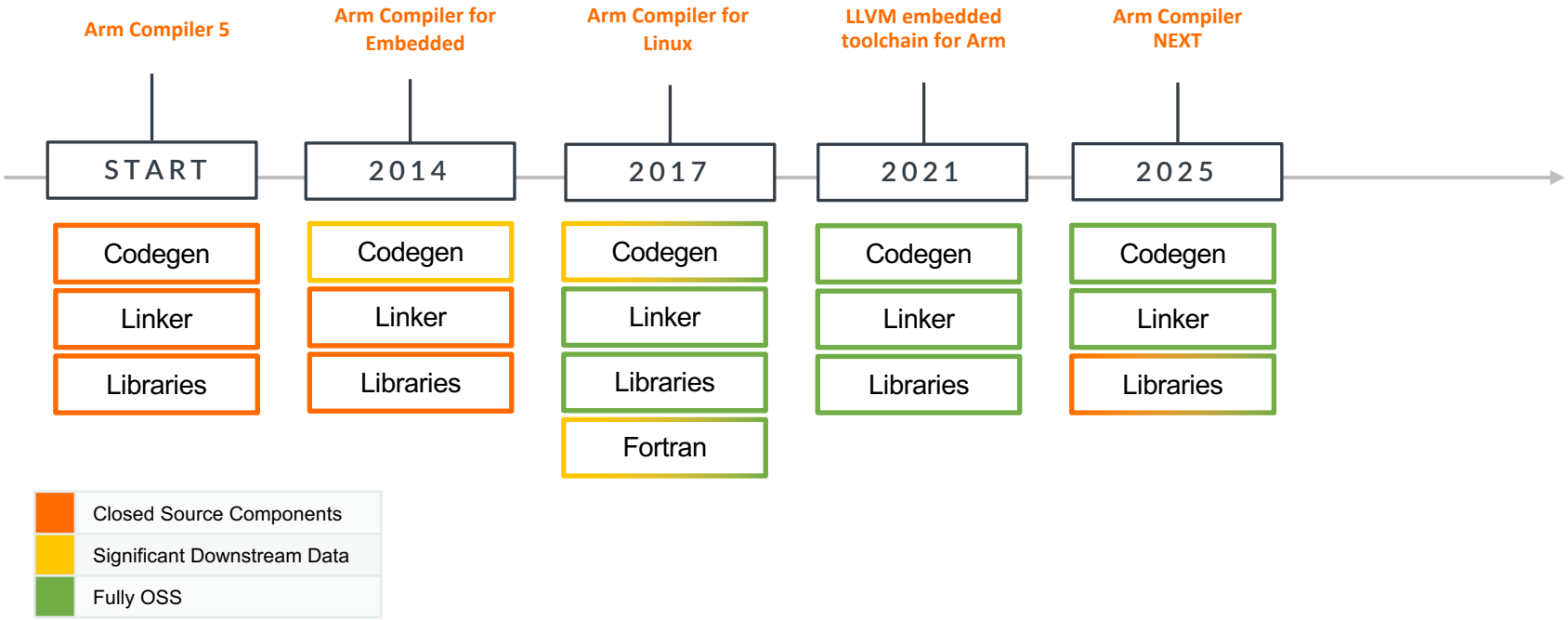


- Open-source Github project, Apache 2.0 license
- Complete Embedded toolchain using OSS LLVM releases
- Contributions to the lld Linker so far include
  - Added improved support for advance code placement features
  - Added support for multilib (automatic selection of library variants)
  - Improved compatibility with Arm GNU Toolchain
  - Added runtime library support for security features (UBSan, code profiling)
- Pre-built binary packages available on GitHub
  - For all major LLVM releases since LLVM 13
  - For all major host platforms (Linux, Windows, macOS)
- Supports Armv6-M and above (ie. all Arm Cortex cores)
  - R and A profile support is in progress

A Brief History | LLVM @ Arm



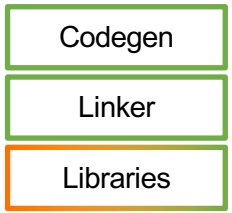
A Brief History | LLVM @ Arm | Arm Compiler NEXT



Introducing Arm Compiler NEXT

START — 2014 — 2017 — 2021 — **2025**

Arm Compiler NEXT



- New flagship compiler from Arm, launching in 2025
- Upstream-first development model
  - Fully upstream LLVM codegen
  - Fully upstream LLVM linker
- Support for embedded development
  - Arm Compiler for Embedded C libraries ported to lld
  - Baremetal C++ library
- Now also supports Linux/RTOS hosted development
  - Work with Elektrobit to develop a Fully qualified C and C++ toolchain for EB Corbos application development
- First functional safety release planned later in 2025

Summary

Close collaboration with partners to develop Functional-safety toolchains for Automotive

10+ year shift from downstream compilers to upstream-first everywhere

Only possible due to the exceptionally-high quality of Upstream LLVM

Linaro Connect
MADRID 2024 | MAY 12-17 2024

Thank you

arm