arm
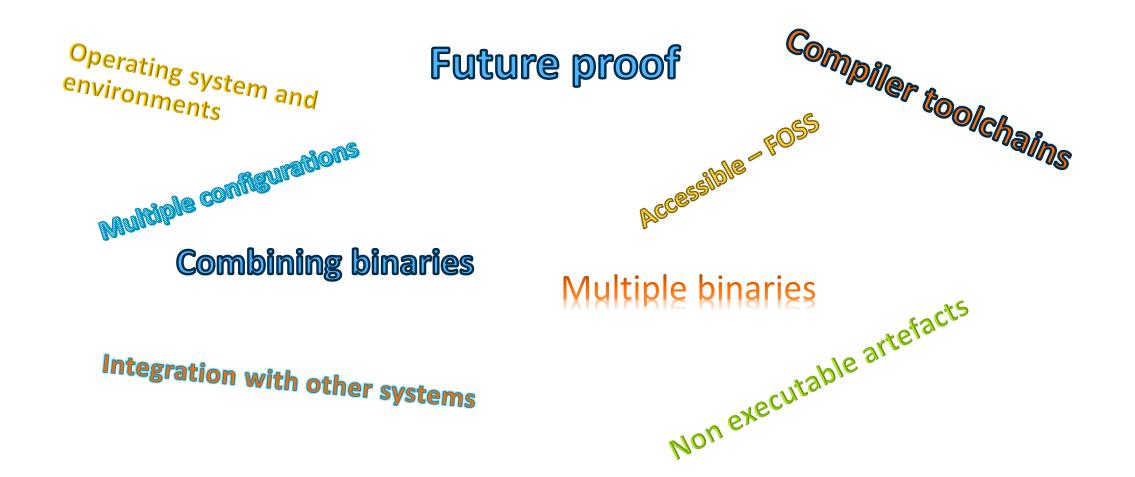
Trusted Firmware-M
Build System & Tooling

TF-M Workshop Lyon

Abhishek Pandit, David Wang
November 2019

Build system requirements

Operating system and environments

Future proof

Compiler toolchains

Multiple configurations

Accessible – FOSS

Combining binaries

Multiple binaries

Integration with other systems

Non executable artefacts

arm

Why Cmake was selected?

- Off tree build system & project generator

- Can generate configuration for many "build executors".

- Widely used in open source projects.

- Good on system integration. (Public, interface, internal.)

- Lifecycle support (build, test, release).

- Cross platform (available for all major OS –es).

- Free and open-source. No vendor lock-in, we can add features if needed.

- There is a short learning curve. What can we do to help?

arm

Build tool

Free/Open tools

- Available to everyone freely.
- What problems do you face currently?

Proprietary/Paid Tools

- Differentiating features per tool.
- Distinct requirements and customization per tool.
- Benefits a subset of users.
- There are challenges!

arm

Configuration options

Varying requirements

- System partitioning and memory layouts
  - CMSIS Zone.
  - Zephyr mechanism.
  - CMSIS Zone "like"

- TF-M feature customization
  - Include/Exclude services
  - New service inclusion

- Multiple binaries and dependencies
  - Separate build stage for S, NS, BL

- CMSIS-Pack
  - Specific TSC discussion

- Test case granularity / customization in build

arm

Debugging

Many preferences..

- DS-5 & Keil MDK are in use for debugging

- No free tools being used currently
  - GDB support for debugging secure side.

arm

Thoughts?

arm

arm

Thank You
Danke
Merci
谢谢
ありがとう
Gracias
Kiitos
감사합니다
धन्यवाद
شكرًا
תודה