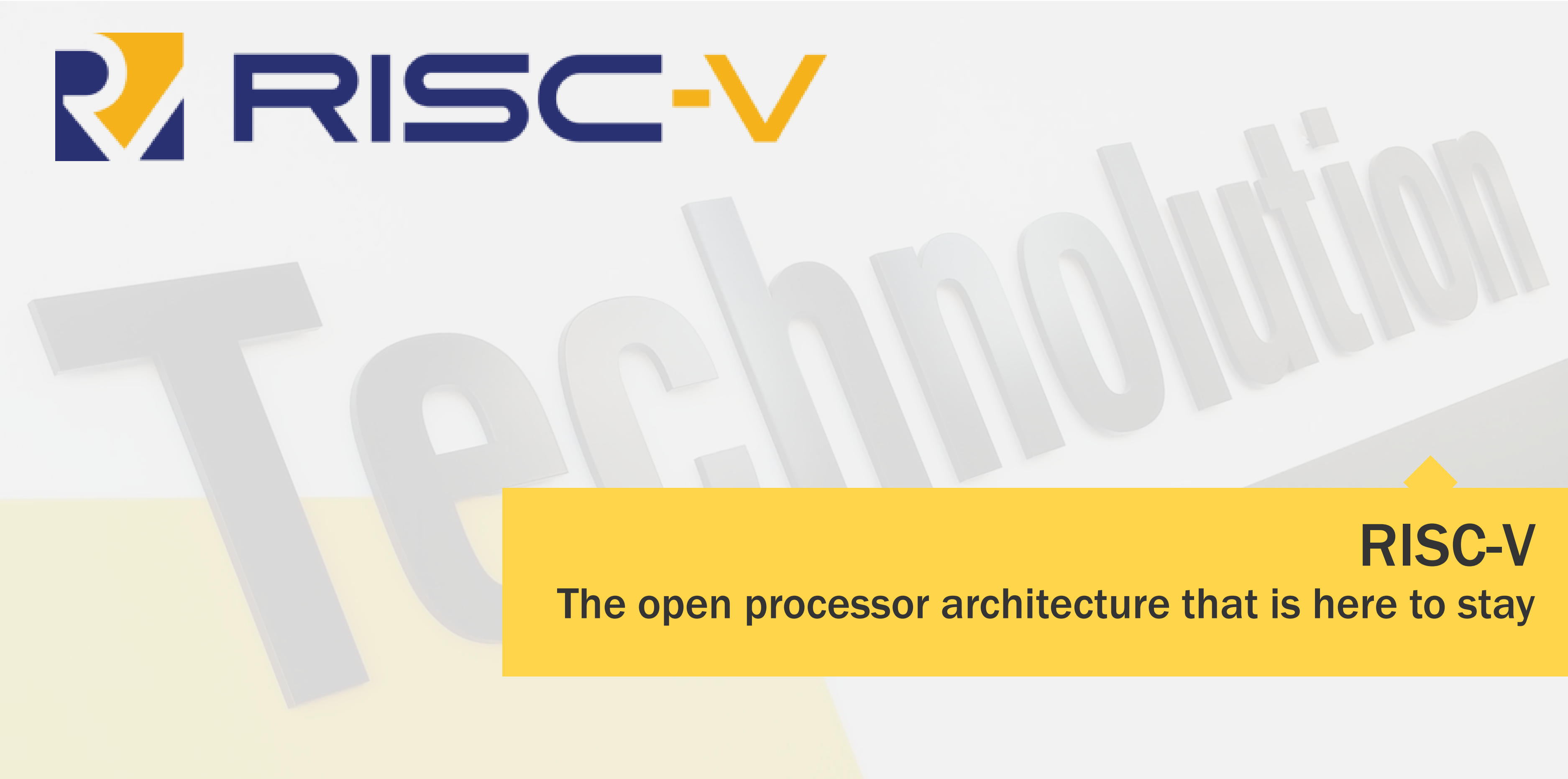


RISC-V

The open processor architecture that is here to stay

About us

- Jonathan Hofman
  - 2003 – Master computer engineering
  - 2005 – VHDL programmer
  - 2009 – Project management
  - 2011 – Technology Manager Programmable Logic
  - 2016 – Domain architect Defense, Safety & Security

- Technolution
  - We are a **technology integrator**, developing electronics and software solutions for technical information systems and **embedded systems** for amongst others **government**, **defense** and the **high-tech industry.**

- My interests
  - Mixed criticality systems
  - High assurance security systems
  - Computer architectures (RISC-V)
  - High performance real-time processing
  - Programmable logic, FPGAs, etc.

public

Technolution

What is RISC-V?

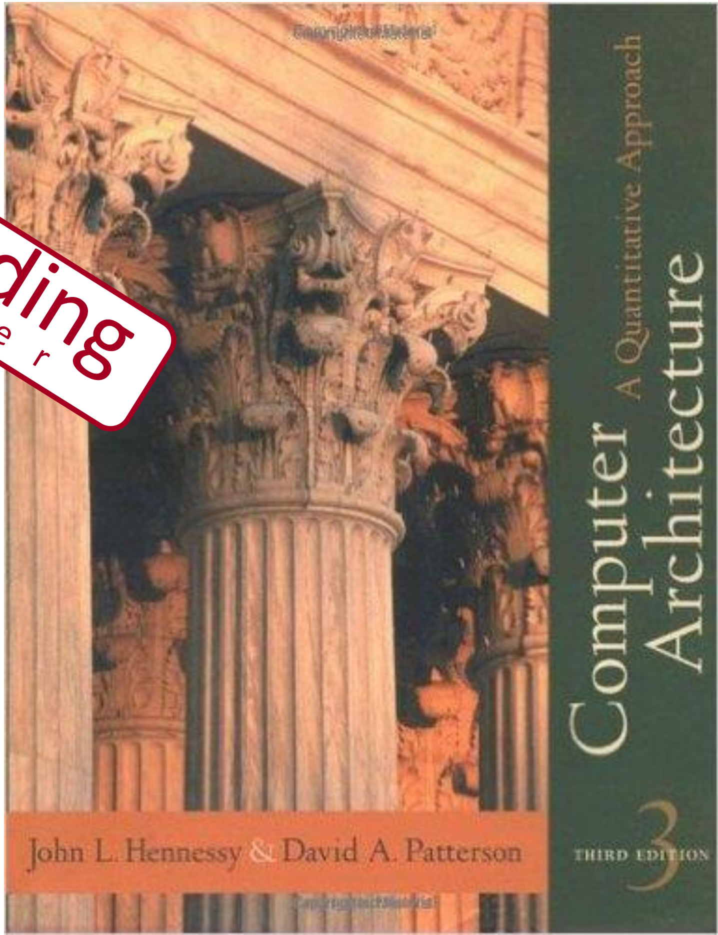
RISC-V

- **an open and free ISA** (Instruction Set Architecture)
  - from Berkeley (contributed: Patterson)
  - result of 30 years of architecture research
  - from IoT to server based
  - flexible and extendable
  - provide the alternative for ARM and Intel

founding member

- **we believe that what Linux did for embedded devices, RISC-V can do for SoCs and FPGAs**
  - boost development by providing core technology
  - grow an eco-system
  - stimulate a community to create, share and reuse new technology

→ enable academia and industry to create new and innovative solutions



Computer Architecture

A Quantitative Approach

John L. Hennessy & David A. Patterson

THIRD EDITION 3

What is RISC-V?

make RISC reduced again!

47

basic instruction

(38 minimal)

extensions to scale performance/code size

IoT ↔ server

Success of an ISA

completeness
efficiency
simplicity
flexibility

eco-system

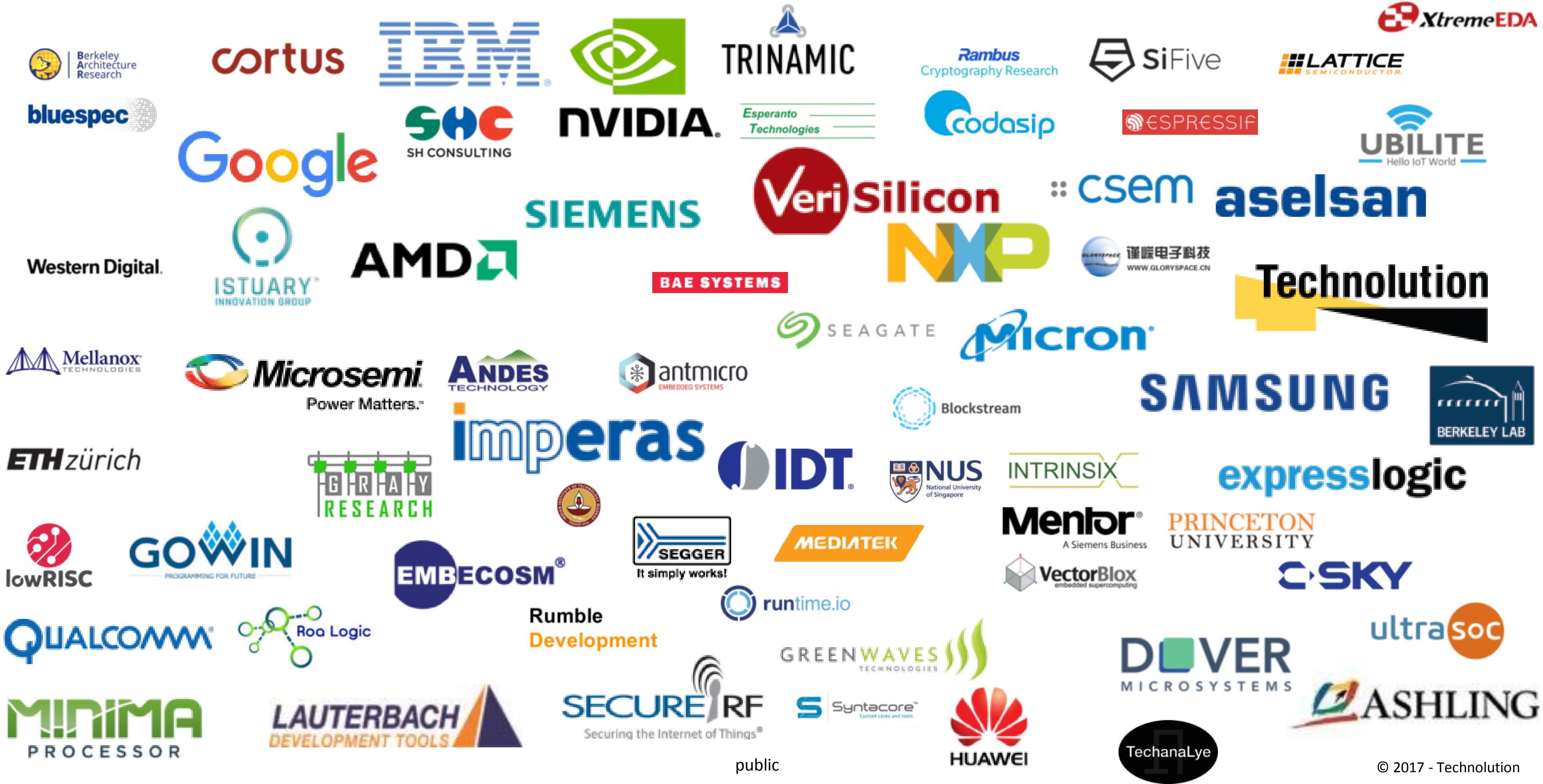
by

community traction

by

need & believe

RISC-V foundation 2015 → 2017



Community status – Software eco-system

- Compilers, tools & core libs
  - Binutils → upstream (as of 2.28)
  - GCC → upstream (as of 7.1)
  - Newlib → upstream (as of 2.5.0)
  - LLVM → upstream in progress
  - Gdb → upstream patches submitted
  - Glibc → upstream patches submitted

    Coreboot → upstream submitted
  - OpenOCD → fork available
  - …
- Simulators
  - Golden simulator (Spike)
  - QEMU
- Other
  - Verification suite

- Kernels/OS
  - Linux → upstreaming in progress
  - RTEMS → upstreaming in progress
  - seL4 → upstreaming in progress
  - Genode → upstream (64 bit)
  - FreeRTOS → port available
  - ThreadX → port in progress
  - Zephyr → port available
  - …
- Distributions
  - FreeBSD → upstream (as of 11.0)
  - Fedora → fork available
  - Debian → fork available
  - Gentoo → fork available
  - buildroot → fork available
  - Yocto → fork available

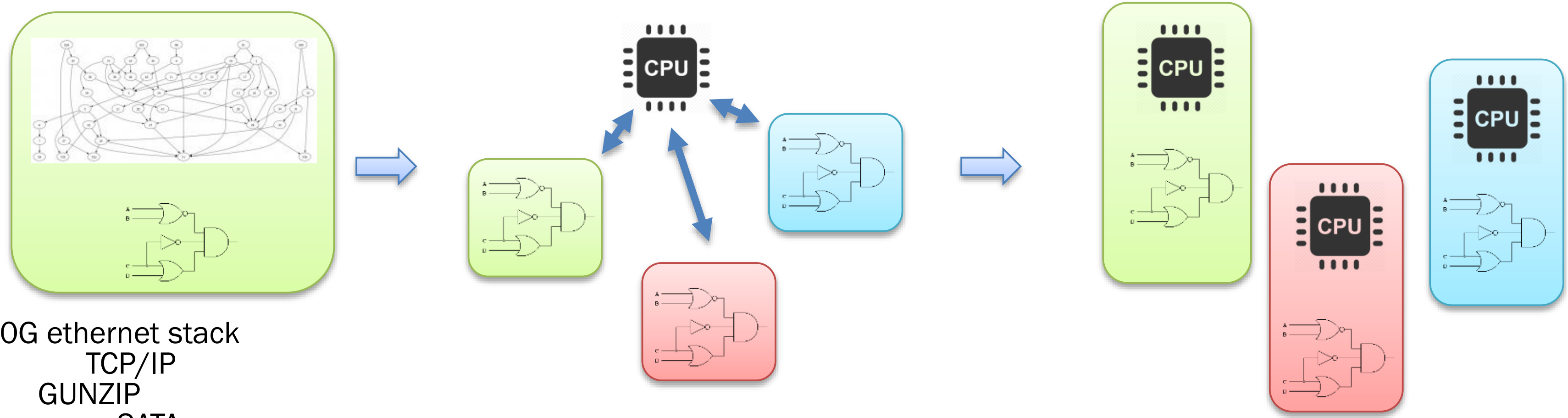
Community status

- Specifications
  - User level
    - 32/64 integer → frozen
    - Multiply/div → frozen
    - atomic → frozen
    - float → frozen
    - Compressed → frozen
    - 128 integer → in progress
    - SIMD → in progress
    - Vector → in progress
    - user interrupts → in progress
  - Privileged
    - Machine → in progress
    - Supervisor → in progress
    - Hypervisor → open
  - Other
    - Debug → in progress

- Cores
  - Numerous open-source
  - Numerous proprietary

Technolution's motivation for RISC-V

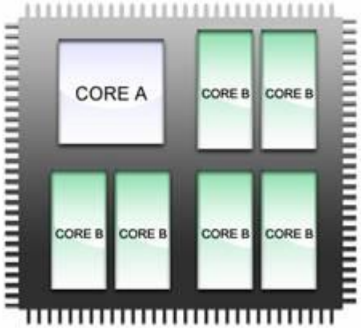
- IP blocks must go hybrid
  - **control flow** becomes more and more complex
  - IP needs to be **flexible**
  - fast development cycles, reuse and **agility** desired
  - better detection, handling and reporting of **exceptional conditions**



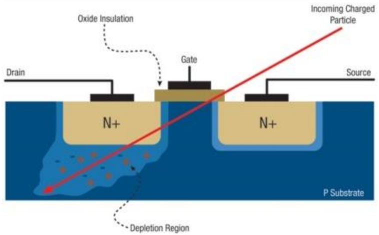
10G ethernet stack
TCP/IP
GUNZIP
SATA core
.....

Technolution's motivation for RISC-V

- Security, reliability and safety
  - **separation** of concerns
  - multiple software **components**
  - multiple **isolated cores** in **mixed criticality** systems



- add reliability, safety and security oriented features
  - evaluation
  - memory labeling
  - fault tolerance
  - secure & reliable NoC



government & defense

medical

aviation & space

Why RISC-V for us?

Hybrid IP blocks

- Easy to integrate into IP block
  - no distribution restrictions
  - no complex license fees
  - small
  - easy to integrate

Security, reliability & safety

- source code available for evaluation
- extendable with domain specifics
  - memory labeling
  - fault tolerance
- both small and high performance cores

- Large and growing eco-system
- technology independent
  - FPGA vendor
  - migrate towards ASIC
- easy simulation

public

What do we have

FreNox-E

Embedded processor

- hardware
  - RV32I(M)
  - 32bits, mul/div
  - 5 stages - Harvard arch
  - cache or internal RAM
  - IO space
- software
  - Bare metal
  - FreeRTOS
  - ThreadX

FreNox-S

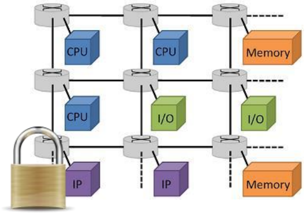
Application processor

- hardware
  - RV32IMAS
  - 32bits, mul/div, atomic, supervisor
  - 5 stages - Harvard arch
  - iMMU, dMMU (1 - 128 entries)
  - 8 way associative cache (4 - 32k)
  - cache coherency (DMA)
  - IO space
- software
  - Linux
  - Buildroot

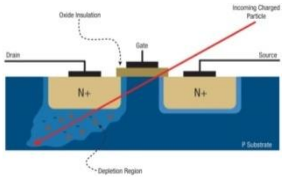
What do we plan

- Genode (component based security OS)
    - 32 bit port
    - microkernel hardware acceleration
    - security hardware features

- Secure NoC
    - Share memory but keep strong separation



- Fault-tolerant implementation
    - Processor & caches
    - Build on previous research



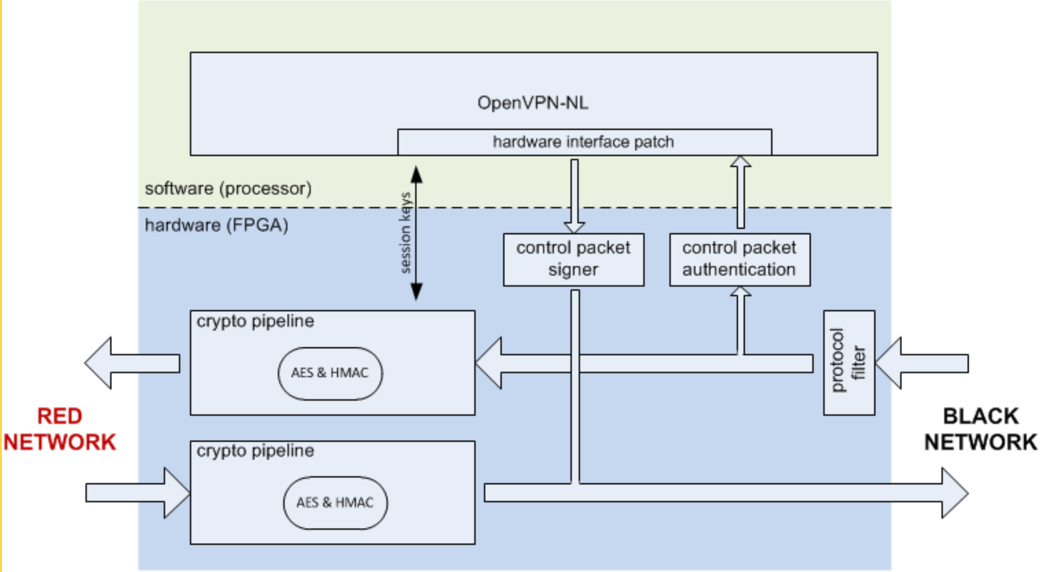
- RUST on RISC-V
    - safer embedded programming

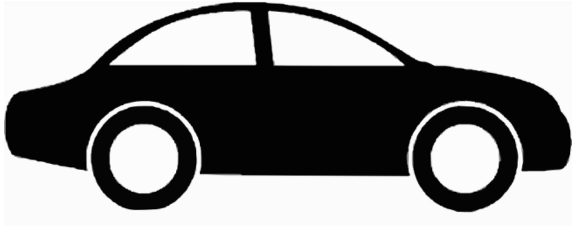
public

Technolution

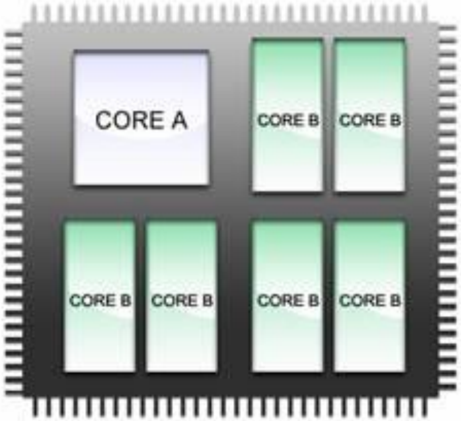
What do we build?

High-assurance crypto device





Next generation car platform
(mixed criticality)



public

RISC-V

serious community traction

great adaptations in the

software eco system

provides a common architecture that scales from

small to large

fill the need of

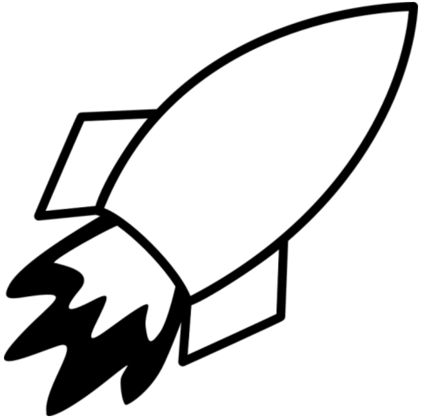
future applications

How could we bring

RISC-V

to

space



Technolution

/ the right development

Jonathan Hofman
jonathan.hofman@technolution.nl
+31 6 10 782 782
linkedin.com/in/jonathanhofman