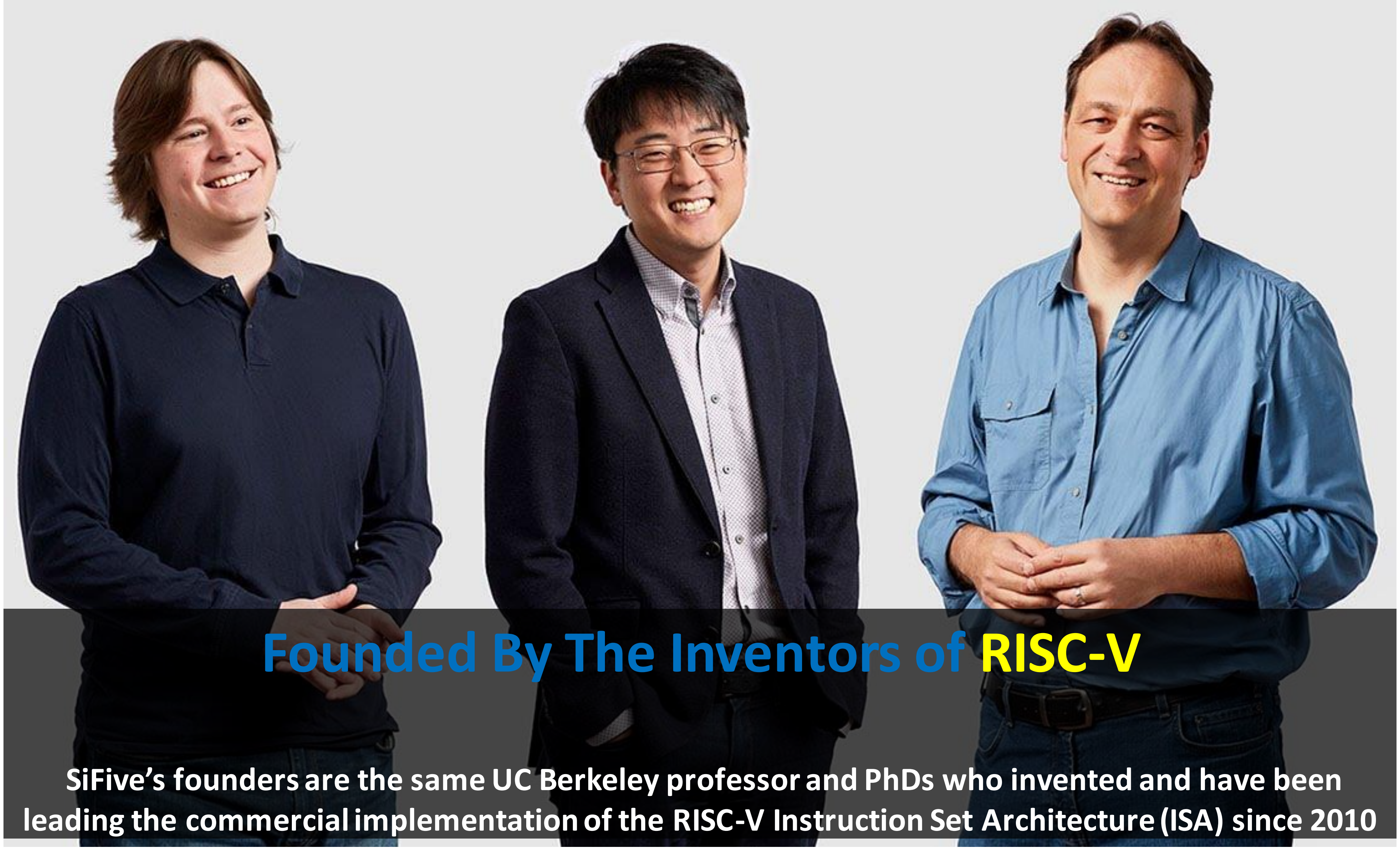
SiFive

RISC-V Security

Yann LOISEL, Security Architect, SiFive

January 2020

Silicon At The Speed Of Software



Founded By The Inventors of RISC-V

SiFive’s founders are the same UC Berkeley professor and PhDs who invented and have been leading the commercial implementation of the RISC-V Instruction Set Architecture (ISA) since 2010

RISC-V security: motivations

- RISC–V ISA designed to address existing ISAs issues about security/secrecy/lack of rationale

Avoid secrecy in design

- RISC–V ISA design built on many years of experiences, mistakes, lack of anticipation

Get rid of legacy security

- RISC–V ISA future backed by the RISC-V Foundation, managing the standards and the assets

Shared rationale, sustainable design

- RISC–V ISA open to security audits and academic reviews

Improve auditability

RISC-V security: give trust

- Ability to do formal verifications, detecting inconsistencies with the standard, detecting additions
- Foster the verification industry
- Shared and sustainable effort because of an open standard
- Results can be public
- RISC-V Foundation Security Standing Committee created in 2018
- Best security practices
- Collaboration with other groups

- TEE subgroup
- Crypto extensions subgroup
- Bit manip subgroup
- Academic and industry together

RISC-V security: give trust

- With the open community, the risks are shared, the countermeasures are shared

- Open ISA allows a fully open-source hardware implementation
  - easy access to deep details, easy modification, testing, prototyping
- Open ISA could help for micro-architecture better security
- An action for the future, not a reaction to the past

- In line with industry concerns for more security assurance
  - IPSA

RISC-V security: add security, ease the security

Adding, extending is in the DNA of RISC-V

- Instructions extensions: add crypto instructions at micro architecture level
  - AES, SHA, TRNG
  - Bit manip
- Vector extensions: ease the use of cryptography for an easier use of the security
- Software architecture: secure monitor, secure boot, TEE APIs, attestation, ...
- Large scope, scalability, better consistency, longer sustainability
  - 32-bit, 64-bit, 128-bit, … : from small, single-core to large, multi-core systems
- Lot of initiatives: DARPA, Thalés-Microchip contest, ...

RISC-V security: privileges management

- Driven by the principles of the smallest attack surface in M mode and the least needed privilege
  - Delegate as much as possible
  - Even in M-mode, you couldn't do what you want
    - Even in S-mode, you couldn't run U code

- Native definitions of multiple privileges levels: M, S, U

- Privileged instruction set

- **RISC-V Privileged Specification defines 4 levels of privilege, called Modes**
- **Machine mode is the highest privileged mode and the only required mode**
  - Flexibility allows for a range of targeted implementations from simple MCUs to high-performance Application Processors
- **Machine, Hypervisor, Supervisor modes each have Control and Status Registers (CSRs)**

RISC-V Modes		
Level	Name	Abbr.
0	User/Application	U
1	Supervisor	S
2	(Hypervisor)	H
3	Machine	M

Supported Combinations of Modes	
Supported Levels	Modes
1	M
2	M, U
3	M, S, U
4	M, H, S, U

RISC-V security: privileges management

- Configuration depending on system complexity
  - M, or M/U or M/S/U
  - M/S/U initially for large systems, running big OSes (linux)
  - M/S/U tends to become the standard even for "small" devices (w/o satp register)

- Traps (interrupts, exceptions) management delegation
  - By default, any interrupt goes into M-mode, but it can automatically be delegated to S-mode (for S and U interrupts) or U-mode (for U interrupts)
  - Designed for performances, but also good for security

RISC-V security: privileges management

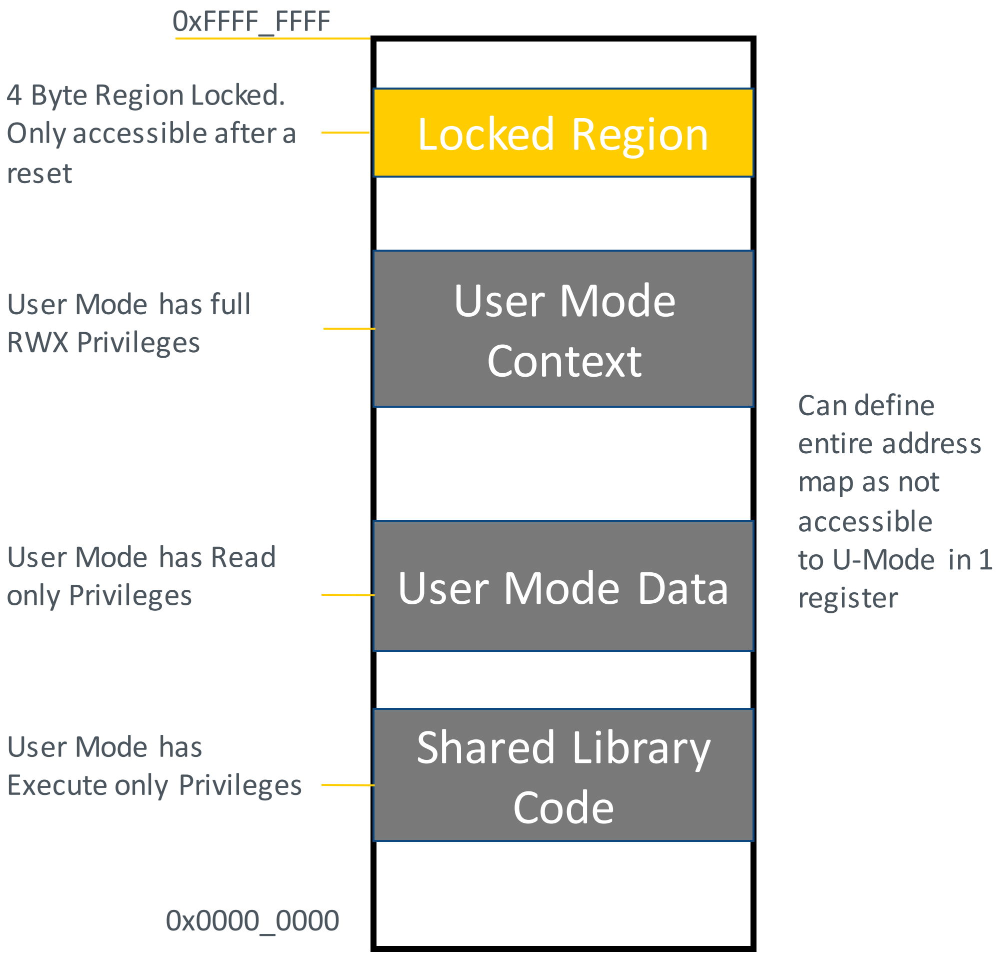
- PMP: physical memory protection
- Defines memories areas access rights and conditions

Physical Memory Protection (PMP)

- **Can be used to enforce access restrictions on less privileged modes**
  - Prevent Supervisor and User Mode software from accessing unwanted memory

- **Up to 16 regions with a minimum region size of 4 bytes**

- **Ability to Lock a region**
  - A locked region enforces permissions on all accesses, including M-Mode
  - Only way to unlock a region is a Reset



0xFFFF_FFFF
4 Byte Region Locked. Only accessible after a reset
Locked Region
User Mode has full RWX Privileges
User Mode Context
Can define entire address map as not accessible to U-Mode in 1 register
User Mode has Read only Privileges
User Mode Data
User Mode has Execute only Privileges
Shared Library Code
0x0000_0000

Example PMP Memory Map

RISC-V security: privileges management

- sPMP: similar to PMP but in S-mode
  - Proposed by the TEE WG

What are Control and Status Registers (CSRs)

- **CSRs are Registers which contain the working state of a RISC-V machine**

- **CSRs are specific to a Mode**
  - Machine Mode has ~17 CSRs (not including performance monitor CSRs)
  - Supervisor Mode has a similar number, though most are subsets of their equivalent Machine Mode CSRs
    - Machine Mode can also access Supervisor CSRs

- **CSRs are defined in the RISC-V privileged specification**

Supervisor CSRs

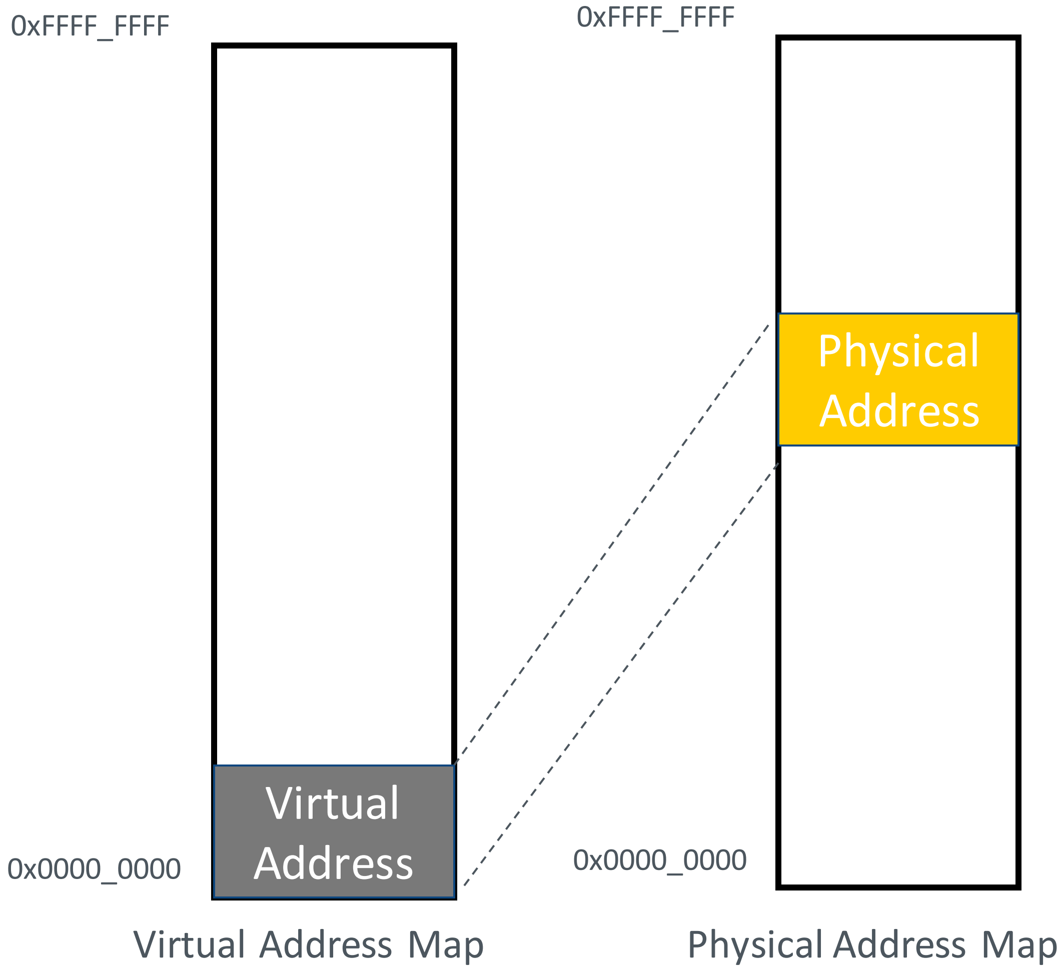
- **Most of the Machine mode CSRs have Supervisor mode equivalents**
  - Supervisor mode CSRs can be used to control the state of Supervisor and User Modes.
  - Most equivalent Supervisor CSRs have the same mapping as Machine mode without Machine mode control bits
  - *sstatus, stvec, sip, sie, sepc, scause, satp*, and more

- ***satp* - Supervisor Address Translation and Protection Register**
  - Used to control Supervisor mode address translation and protection

Virtual Memory

- RISC-V has support for Virtual Memory allowing for sophisticated memory management and OS support (Linux)

- Requires an S-Mode implementation
- Sv32
  - 32bit Virtual Address
  - 4KiB, 4MiB page tables (2 Levels)
- Sv39 (requires an RV64 implementation)
  - 39bit Virtual Address
  - 4KiB, 2MiB, 1GiB page tables (3 Levels)
- Sv48 (requires an RV64 implementation)
  - 48bit Virtual Address
  - 4KiB, 2MiB, 1 GiB, 512GB page tables (4 Levels)
- Page Tables also contain access permission attributes



RISC-V security: debug with security

- RISC-V debug specification standardizes the debug module

- And considers an authentication module, open and flexible

RISC-V SoC needs more security

Scalable architecture

Enhanced isolation

Finer grained controls

System level security

SiFive

SiFive Security

SiFive WorldGuard

- Cryptographic blocks (application, memories, …)
- Secure boot, secure update
- Secure key provisioning
- Secure debug
- System-level isolation

SiFive

SiFive WorldGuard

SiFive WorldGuard

- proposes a resources isolation solution at system level

- splits the system into distinct worlds, each world made of resources

- resources can be masters (cores, DMA channels, …), slaves (portions of memories, peripherals)

- complementary to RISC-V security and virtualization

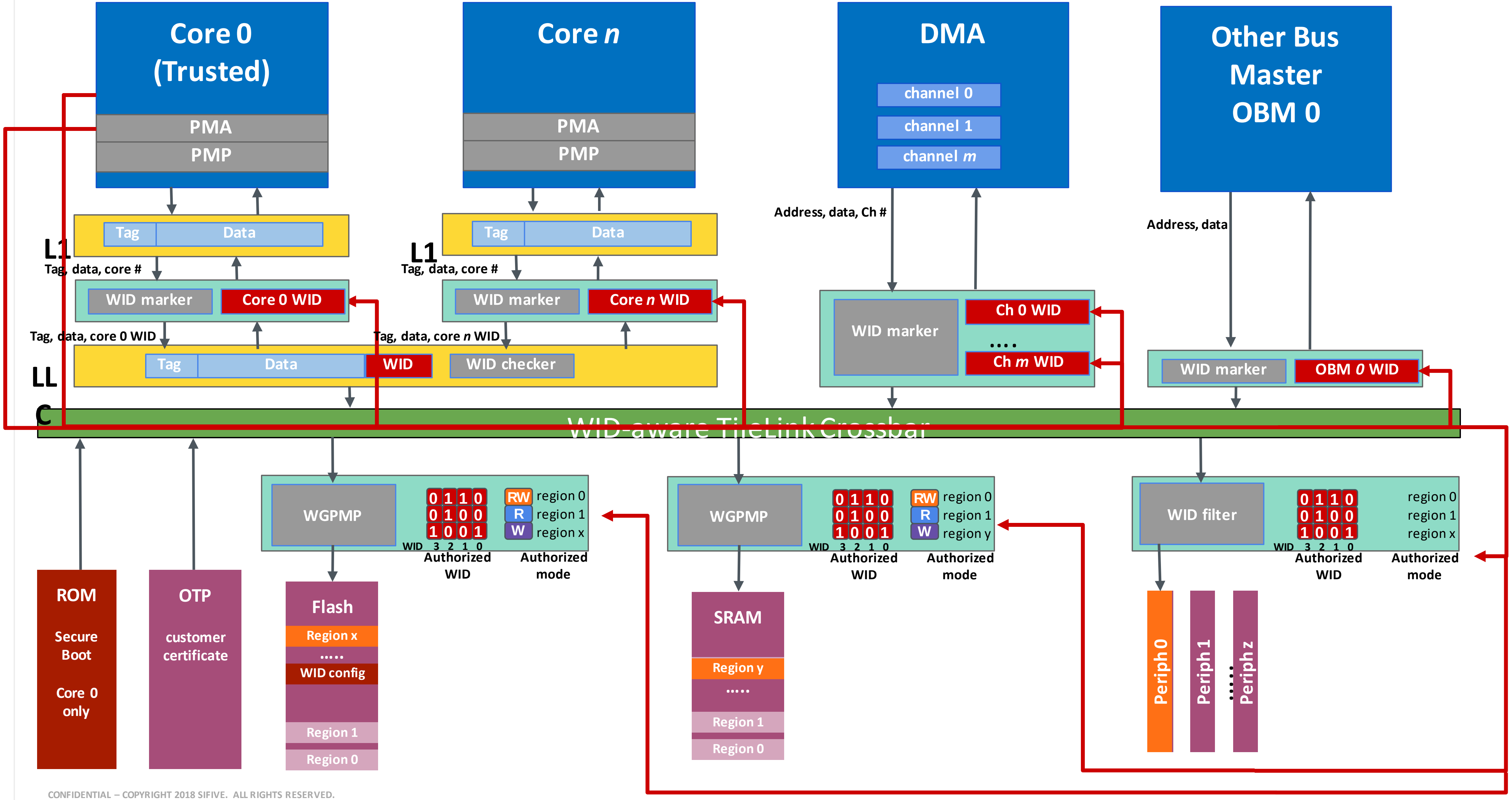
SiFive WorldGuard Security architecture benefits

- **Multi-level trust model for enhanced security and flexibility**
  - Multiple worlds are hardware-controlled and protect memories and peripherals from illegal access
  - Supports multiple cores, multiple bus masters (ie. DMA controllers, caches, eFPGA…)
  - Complementary to what PMP offers for software protection

- **Low system overhead**
  - core agnostic
  - RISC-V ISA remains untouched
  - very low overhead on control logic for peripherals,memories and bus masters
  - very low impact on performances

- **Fine grain control**
  - Up to *n* individual worlds, identified by a WID (World ID)
  - Up to *m* memory regions per memory can be shared between different worlds.
  - Each peripheral has its own access control list per world.

SiFive WorldGuard



RISC-V security: security rationale

- limited TCB: the *trusted core* and its firmware

- do not trust M-mode in other cores

- WG PMPs and WG filters are gate keepers, whatever happens on master side.

- goes beyond the single core security (PMP)

RISC-V security: conclusion

- **secure, simple, scalable**

- **very limited impact on the firmware**

- **the system-level security solution RISC-V community needs**

- **demo at *embedded world 2020* in Nuremberg (feb 2020)**

- **specifications released in march 2020**

RISC-V security

QUESTIONS ?