La cybersecurity per i progetti open source hardware

Definition of the Trusted Life Cycle (TLC) in the ORSHIN project

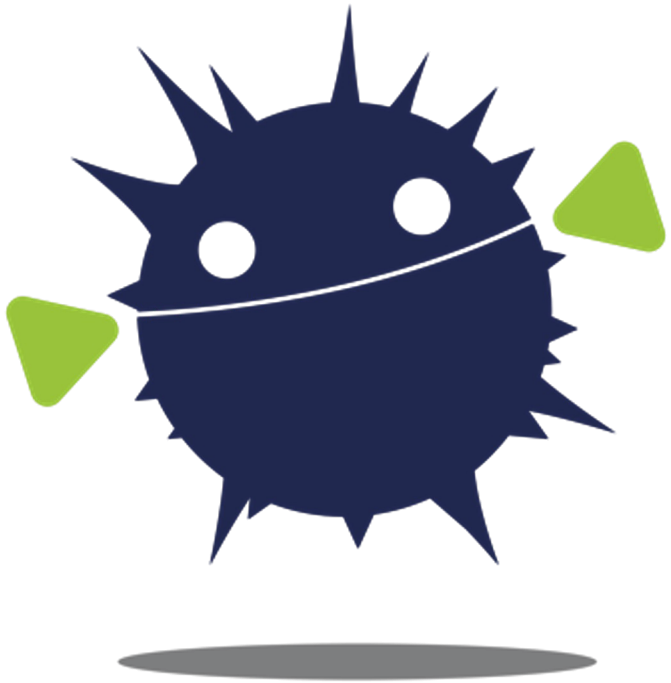
Guido Bertoni, Security Pattern

25/05/2023

Outline

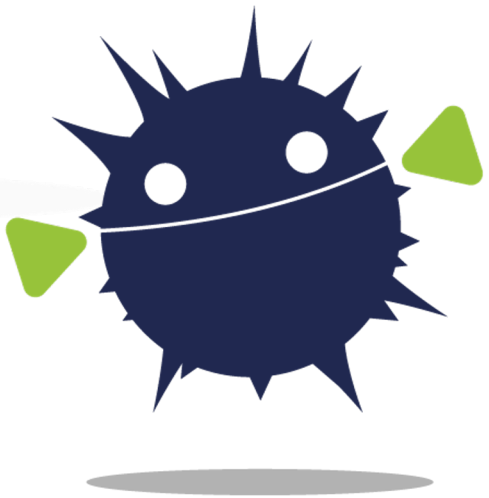
- The ORSHIN project
- Definition of open source HW
- Definition of the Trusted Life Cycle

The ORSHIN project



ORSHIN

- Open-source ReSilient Hardware and software for Internet of thiNgs



ORSHIN

SECURITY PATTERN

TECHNIKON

KU LEUVEN

tropicsquare

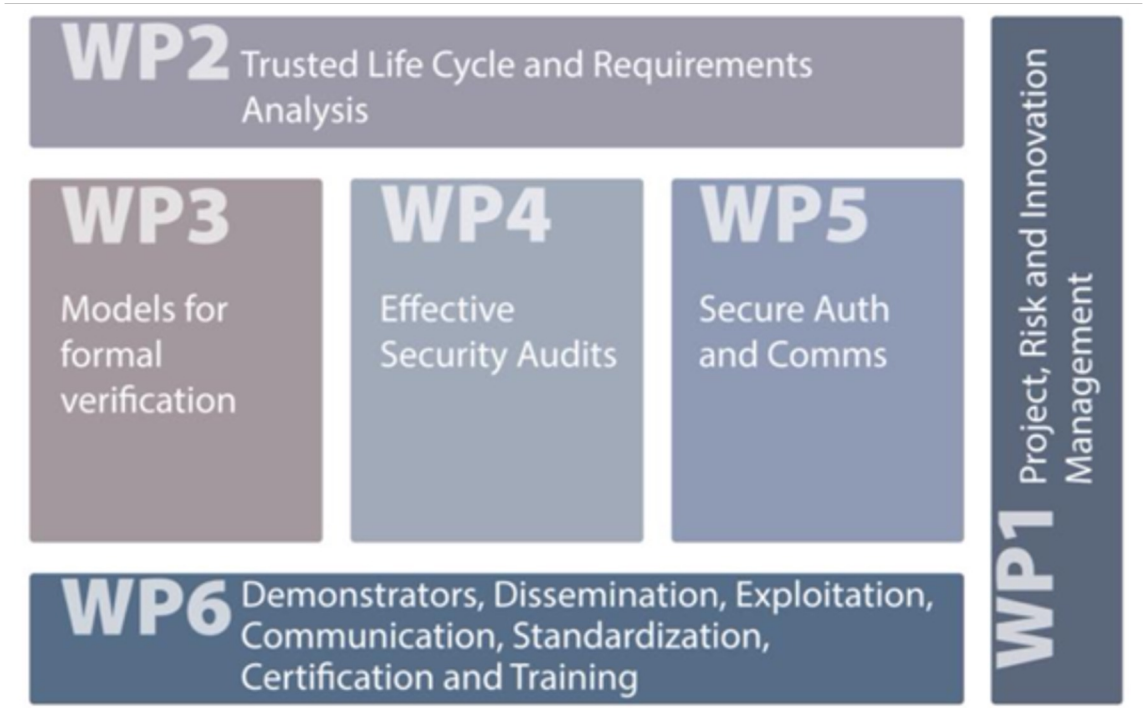
EURECOM
Sophia Antipolis

Texplained
HARDWARE SECURITY INSIGHT

NXP

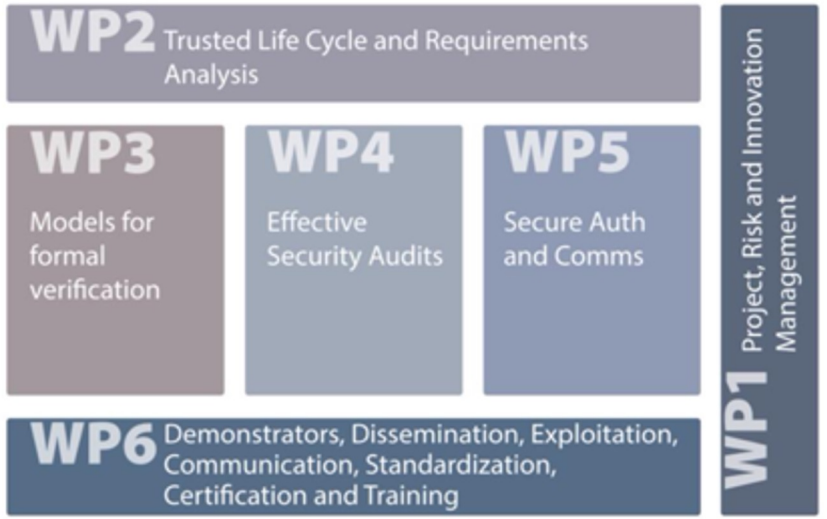
ORSHIN goal

- Define a methodology to develop secure and privacy-preserving (I)IoT devices taking advantage of open-source hardware (and software)

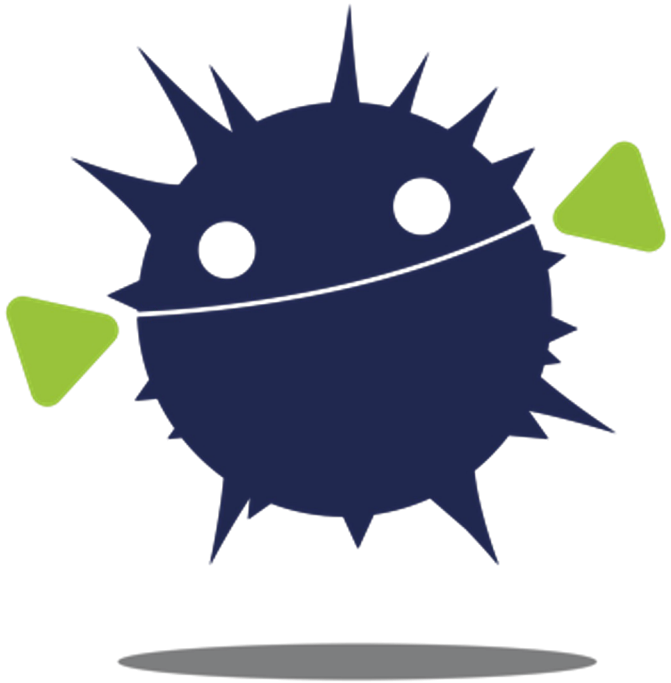


Technical WPs

- WP2 – Trusted Life Cycle and Requirements Analysis
  - Definition of our innovative design methodology trusted life cycle for the development of secure embedded, connected devices that integrate ORSHIN components.
- WP3 – Models for formal verification
  - New models are proposed to support and improve the formal verification of practically relevant security properties.
- WP4 – Effective Security Audits
  - Effective and novel methods are provided for achieving system-wide security auditing of firmware programs for ORSHIN.
- WP5 – Security Auth and Comms
  - Original and practical methods providing security and privacy guarantees for connected embedded devices are developed.



Definition of open source HW



Introduction

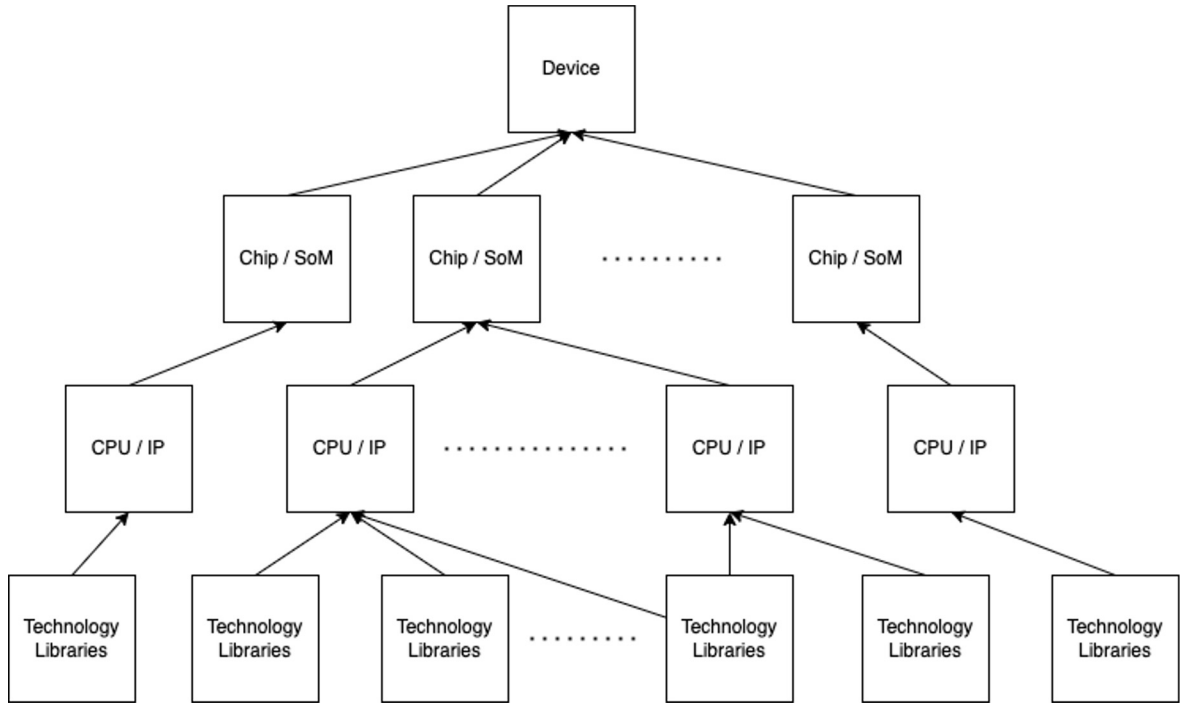
- There are nuances to the concept of "open source"
- Categorization is not always immediate
- Examples
  - Open source product using a closed-design microcontroller
  - Open source HDL distributed with proprietary toolchain
- Existing definitions/guidelines from OSHWA
  - Good starting point, but too generic
  - https://www.oshwa.org/definition/
- Is it possible to agree on a general definition, but with fine detail?

Definition of open source hardware

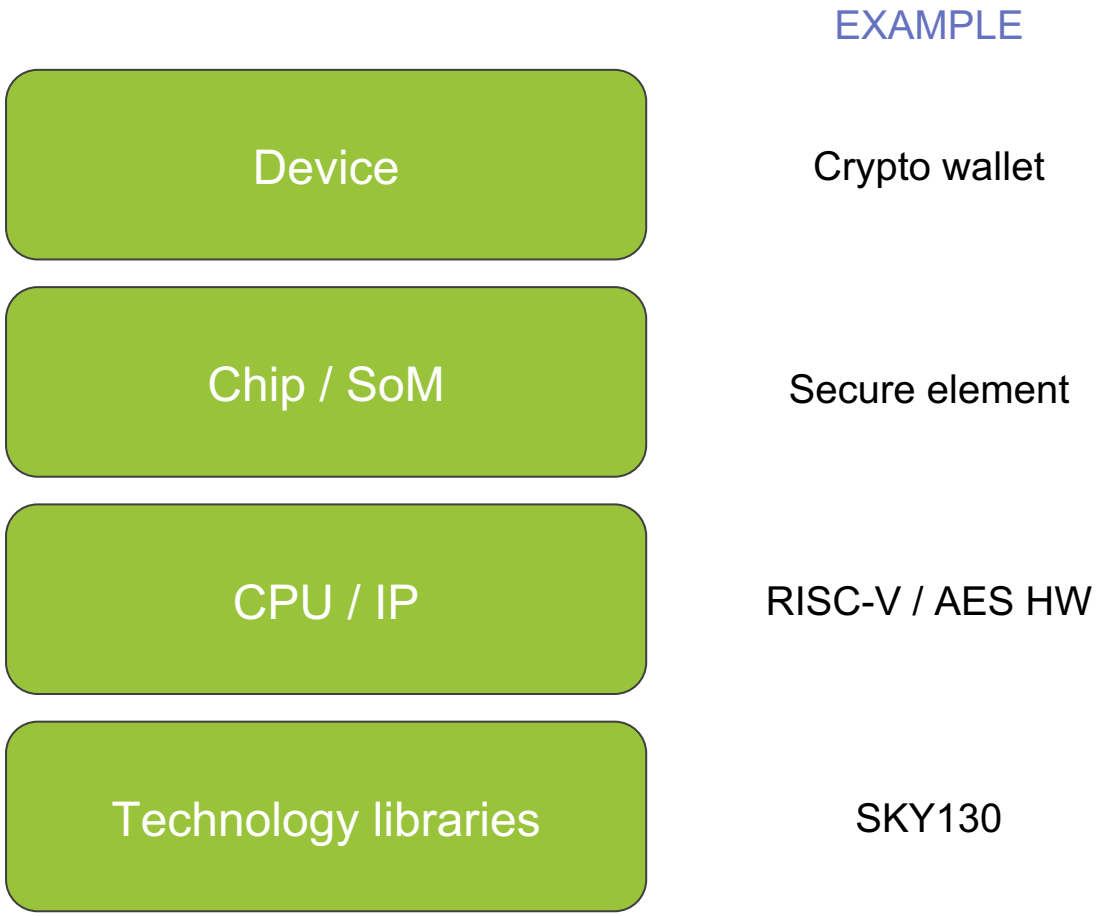
- Is it possible to agree on a general definition, but with fine detail?
  - Tentative answer: YES

The idea

- Clarify the different components of open source hardware
- Assign level of "open-sourceness" separately for each component
- Compute overall vector / score

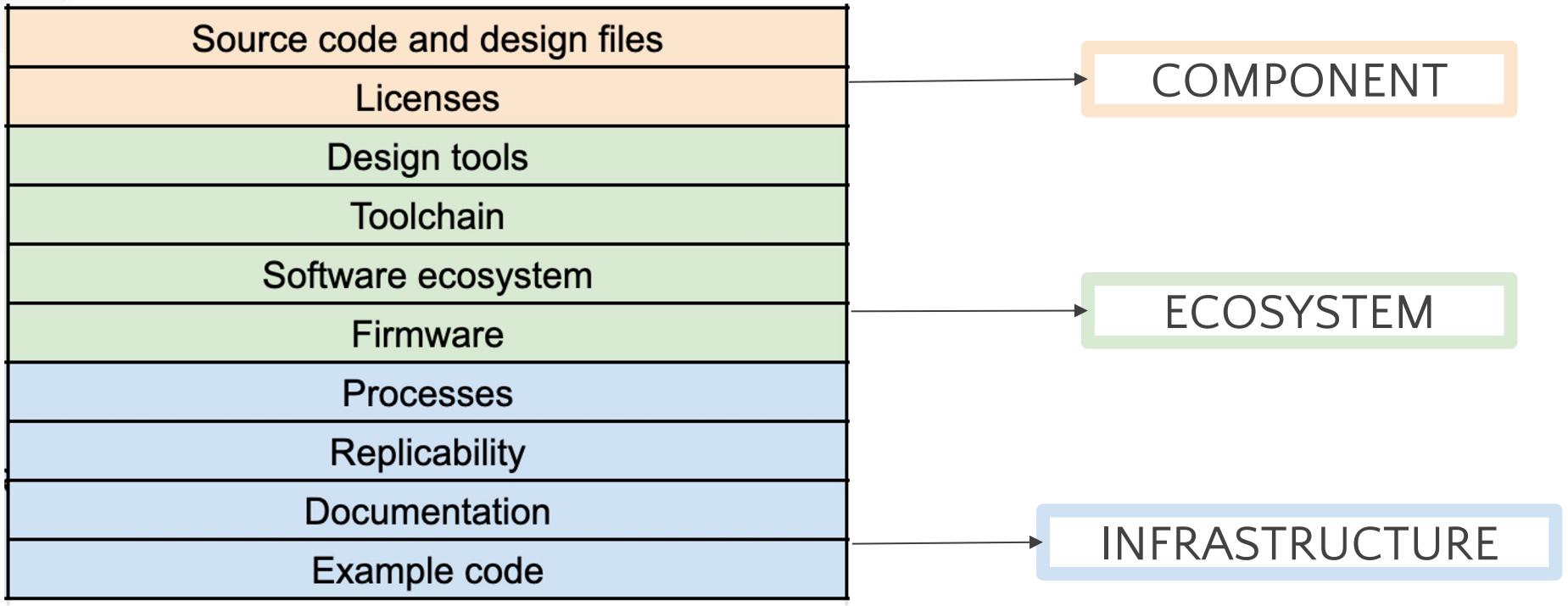


Four views


EXAMPLE
Device
Crypto wallet
Chip / SoM
Secure element
CPU / IP
RISC-V / AES HW
Technology libraries
SKY130

Properties

- Ten properties, grouped in three sets

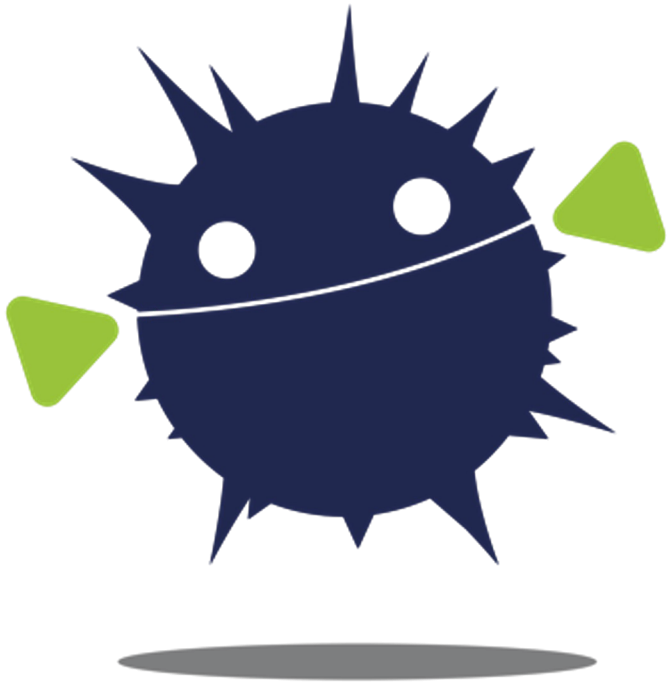


Scoring of the “OpenSourceness”

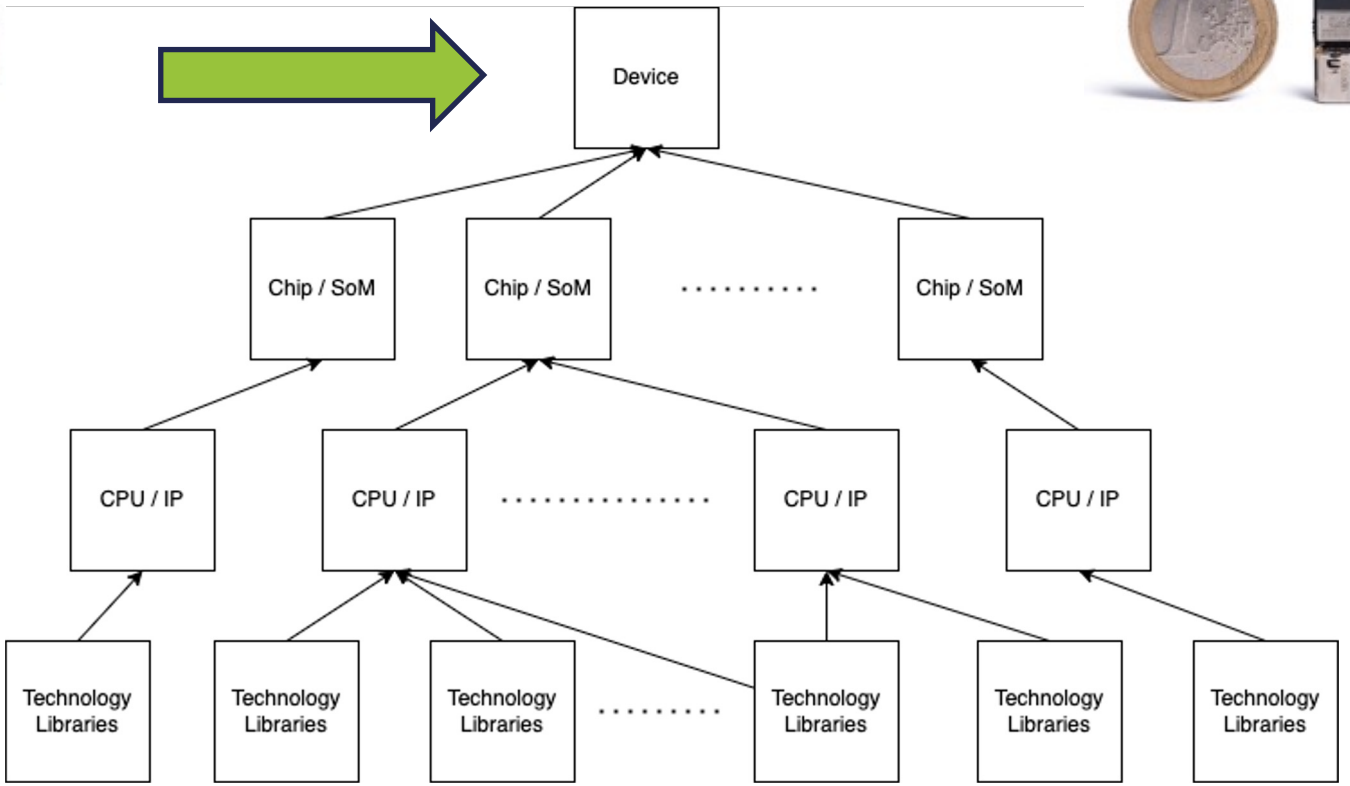
- To score how much an hardware is open:
  - Set the hardware into a **view**
  - Compute of the **vector of scores**: for each property that can be considered for that view, give for that a score, from 0 to 3
  - The **overall score** is computed through a formula, starting from the vector of scores

level	description
0	completely closed
1	more closed than open
2	more open than closed
3	completely open

Definition of open source HW: Raspberry Pi4 VS USB Armory



Raspberry Pi4 vs USB Armory

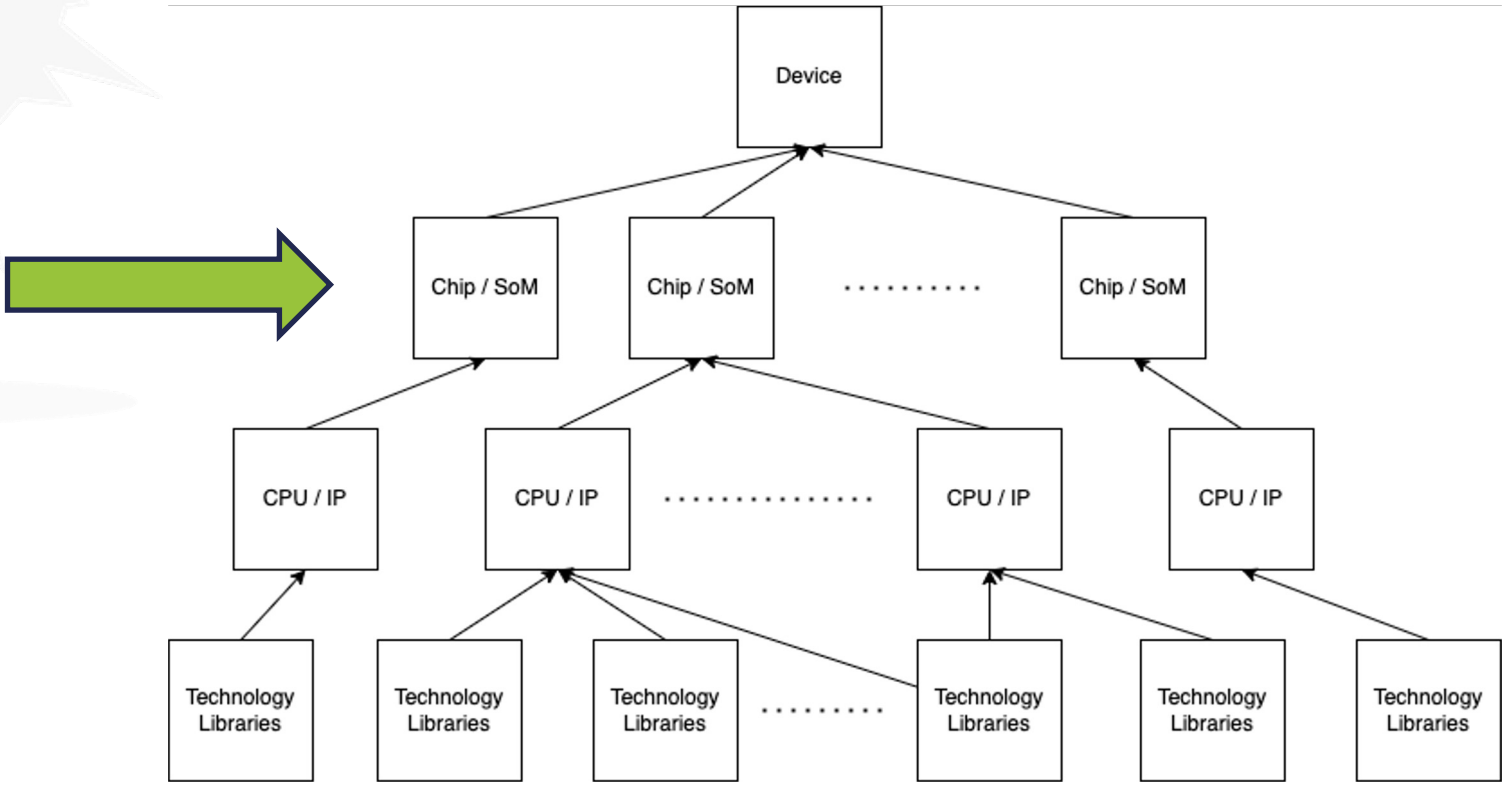


Raspberry Pi4 VS USB Armory Scores

RASPEBERRY Pi4	Properties	Score			Final score
COMPONENT	Source code and design files	2	2	2,416666667	2
	Licenses	2			
ECOSYSTEM	Design tools	3	3		
	Toolchain	3			
	Software ecosystem	3			
	Firmware	3			
INFRASTRUCTURE	Processes	1	2,25		
	Replicability	2			
	Documentation	3			
	Example code	3			

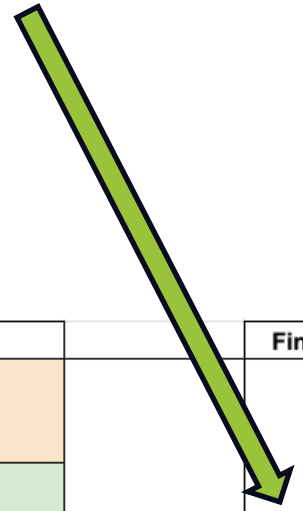
USB Armory	Properties	Score			Final score
COMPONENT	Source code and design files	3	3	3	3
	Licenses	3			
ECOSYSTEM	Design tools	3	3		
	Toolchain	3			
	Software ecosystem	3			
	Firmware	3			
INFRASTRUCTURE	Processes	3	3		
	Replicability	3			
	Documentation	3			
	Example code	3			

Subcomponent: BCM2711 VS IMX6



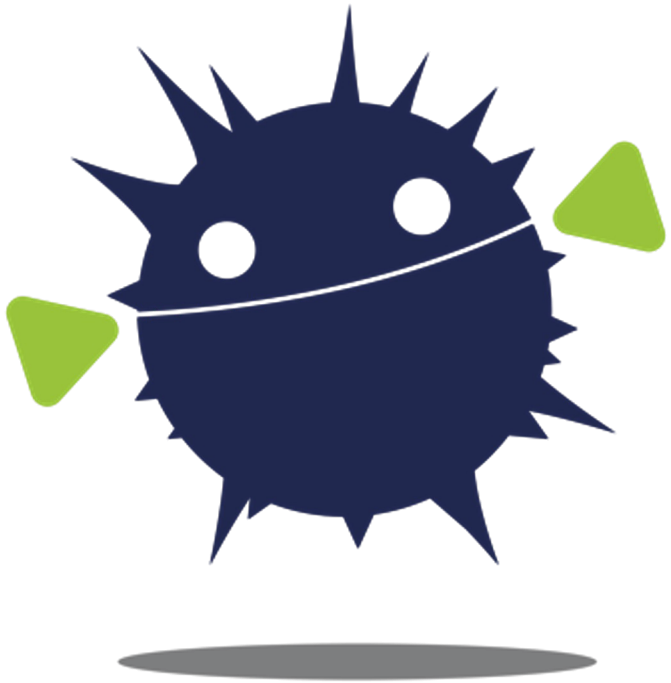
BCM2711 VS IMX6 Scores

BROADCOM BCM2711	Properties	Score			Final score
COMPONENT	Source code and design files	0	0	0,25	0
	Licenses	0			
ECOSYSTEM	Design tools	0	0		
	Toolchain	0			
	Software ecosystem	0			
	Firmware	0			
INFRASTRUCTURE	Processes	0	0,75		
	Replicability	0			
	Documentation	3			
	Example code	0			



IMX6	Properties	Score			Final score
COMPONENT	Source code and design files	2	1	1,583333333	2
	Licenses	0			
ECOSYSTEM	Design tools	2	2,25		
	Toolchain	3			
	Software ecosystem	2			
	Firmware	2			
INFRASTRUCTU	Processes	1	1,5		
	Replicability	1			
	Documentation	1			
	Example code	3			

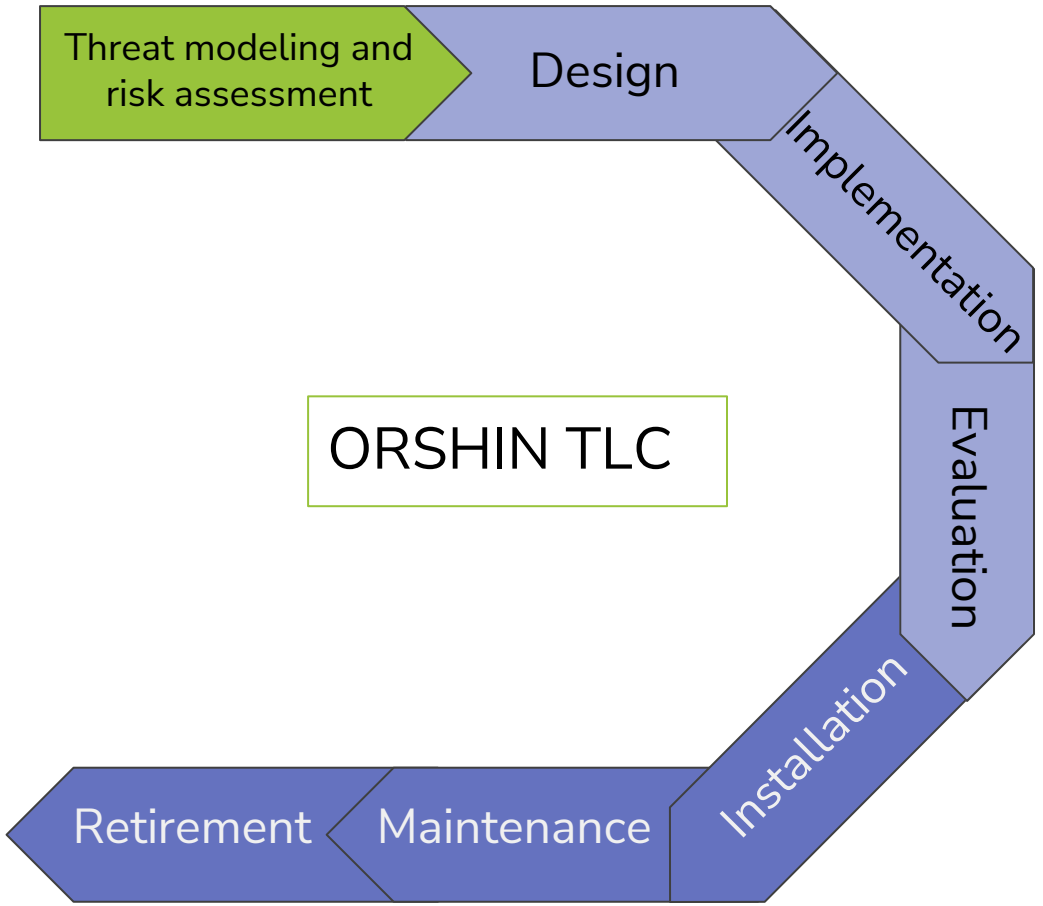
Definition of the Trusted Life Cycle



TLC definition

- TLC = **T**rusted **L**ife **C**ycle of a device
- It is a methodology for the development of secure embedded devices that integrate OSH components

TLC Phases


Threat modeling and risk assessment
Design
Implementation
Evaluation
Installation
Maintenance
Retirement
ORSHIN TLC

TLC definition: steps

- We started from existing process requirements, from some well-known cybersecurity standards
- We adapted these pre-existing requirements to the ORSHIN context
- Starting from those, we are drafting requirements that are specific to **hardware design** and **open source**

Example of requirements

- New requirements

Selection of third-party components in Open Source (Process category)

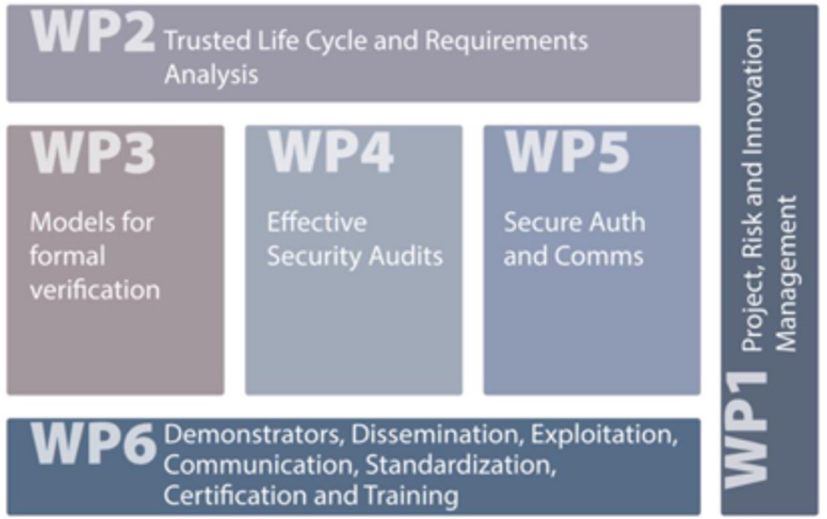
To make it easier for others to replicated and modify the hardware, when possible it is better to prefer the use of free and open-source third-party components, as opposed to proprietary technology.

Apply hierarchical and modular design approach in Hardware design (Technology category)

Apply a hierarchical modular approach to design, by recursively divide systems into modules, reuse regular modules when possible, define well-formed interfaces between modules and sub-systems.

Current focus of Security Pattern

- WP2 – Trusted Life Cycle and Requirements Analysis
  - Definition of our innovative design methodology trusted life cycle for the development of secure embedded, connected devices that integrate ORSHIN components.
- WP3 – Models for formal verification
  - New models are proposed to support and improve the formal verification of practically relevant security properties.
- WP4 – Effective Security Audits
  - Effective and novel methods are provided for achieving system-wide security auditing of firmware programs for ORSHIN.
- WP5 – Security Auth and Comms
  - Original and practical methods providing security and privacy guarantees for connected embedded devices are developed.



Conclusion

- **ORSHIN**: European project
  - *Goal*: define a methodology to develop secure and privacy-preserving (I)IoT devices taking advantage of open-source hardware (and software)
- Definition of **open source HW and Trusted Life Cycle**
- Next steps:
  - Research in the field of the trade off between side channel protections and formal methods
  - Secure and efficient chip 2 chip communications

ORSHIN Grant Agreement No. 101070008

If you need further information, please contact the coordinator:

TECHNIKON Forschungs- und Planungsgesellschaft mbH

Burgplatz 3a, 9500 Villach, AUSTRIA
Tel: +43 4242 233 55 Fax: +43 4242 233 55 77
E-Mail: coordination@horizon-orshin.eu

Funded by the European Union

Funded by the European Union under grant agreement no. 101070008. Views and opinions expressed are however those of the author(s) only and do not necessarily reflect those of the European Union. Neither the European Union nor the granting authority can be held responsible for them.