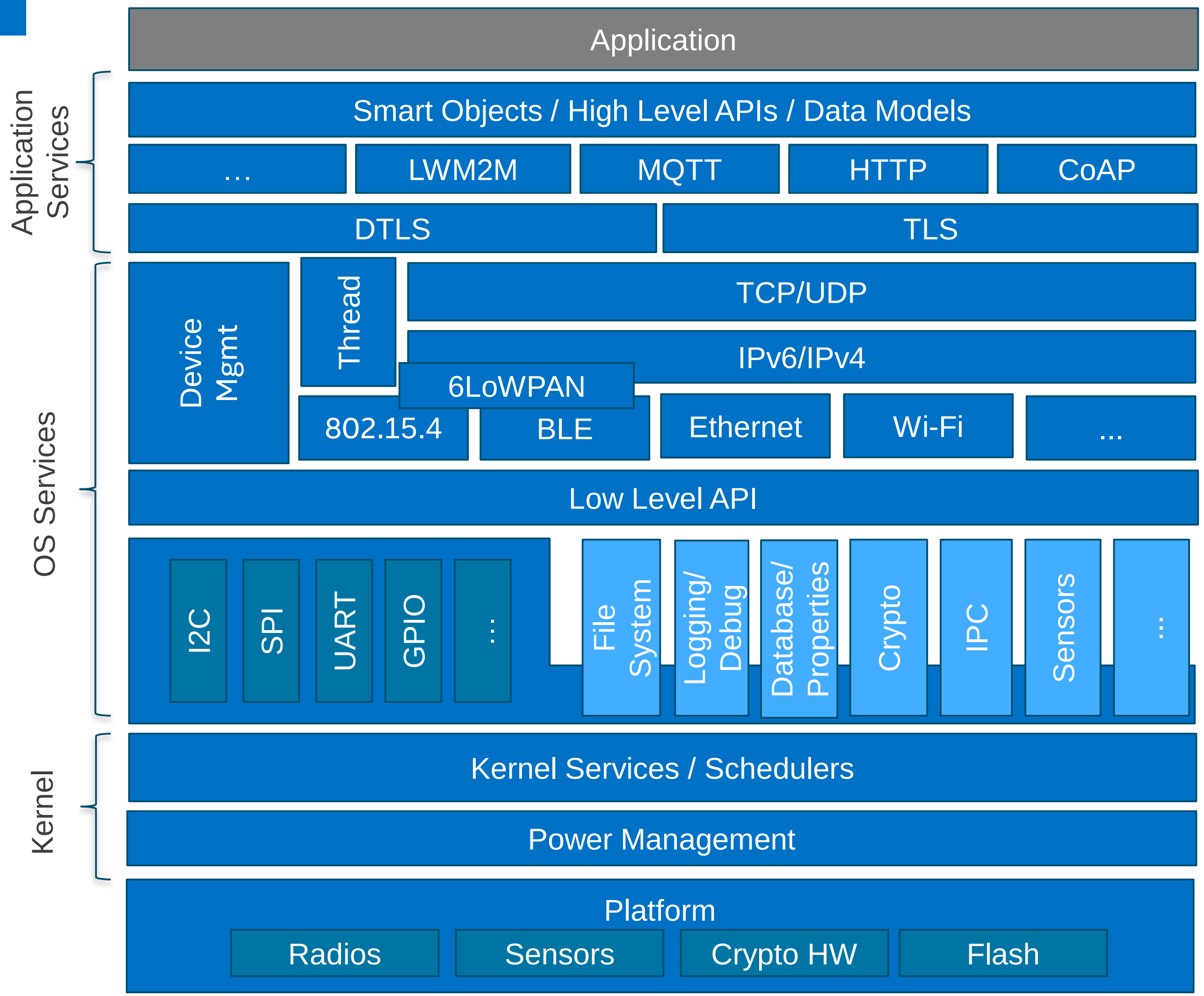
Zephyr OS: Towards Functionally Safe Open Source RTOS

Andrei Laperie

Zephyr: Overview



Application

Application Services

Smart Objects / High Level APIs / Data Models

… | LWM2M | MQTT | HTTP | CoAP

DTLS | TLS

OS Services

Device Mgmt | Thread | TCP/UDP

IPv6/IPv4

6LoWPAN

802.15.4 | BLE | Ethernet | Wi-Fi | ...

Low Level API

I2C | SPI | UART | GPIO | …

File System | Logging/ Debug | Database/ Properties | Crypto | IPC | Sensors | …

Kernel

Kernel Services / Schedulers

Power Management

Platform

Radios | Sensors | Crypto HW | Flash

First release Feb 2016, Apache 2.0

All-in-one solution, not just kernel

Highly modular, using Kconfig

Minimal footprint 8K

Bluetooth LE controller/host, LE Mesh

Native IPv4/6, Thread, 802.15.4,

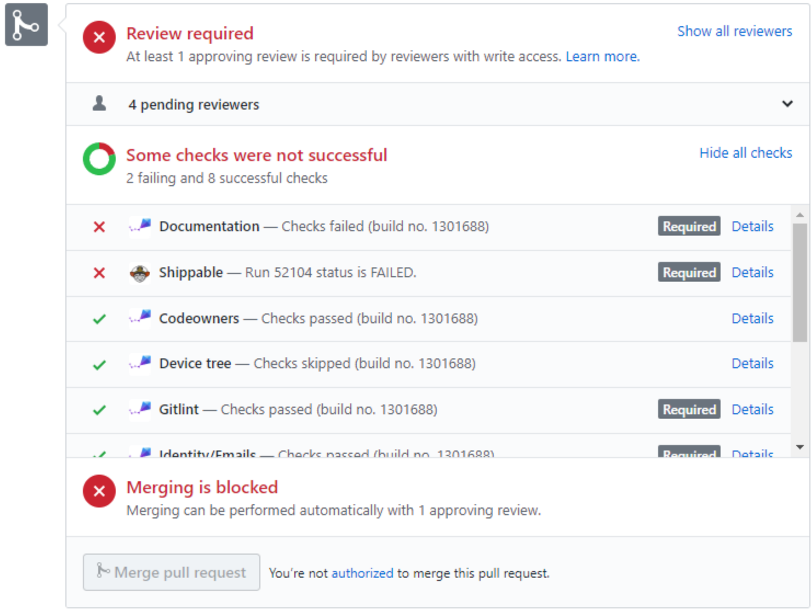
Ecosystem Support

intel | ARC Synopsys | Nios II Processor

ARM | tensilica | RISC-V

Zephyr: Development Process

- https://github.com/zephyrproject-rtos
- Comprehensive CI (~5x1h jobs)
- Area owners/maintainers
- Fixed release cadence
- First Long-Term Service release done maintained
- Test Case management @ Testrail.io
- Automated test suites for Networking and Bluetooth



✕ **Review required** — Show all reviewers

At least 1 approving review is required by reviewers with write access. Learn more.

4 pending reviewers

Some checks were not successful — Hide all checks

2 failing and 8 successful checks

✕	**Documentation** — Checks failed (build no. 1301688)	Required	Details
✕	**Shippable** — Run 52104 status is FAILED.	Required	Details
✓	**Codeowners** — Checks passed (build no. 1301688)		Details
✓	**Device tree** — Checks skipped (build no. 1301688)		Details
✓	**Gitlint** — Checks passed (build no. 1301688)	Required	Details
✓	**Identity/Emails** — Checks passed (build no. 1301688)	Required	Details

✕ **Merging is blocked**

Merging can be performed automatically with 1 approving review.

Merge pull request — You're not authorized to merge this pull request.

Zephyr: Wide Industry Support

Platinum Members

intel

oticon PEOPLE FIRST

NORDIC SEMICONDUCTOR Smarter Things

NXP

Silver Members

SYNOPSYS

FOUNDRIES.IO

TEXAS INSTRUMENTS

Linaro

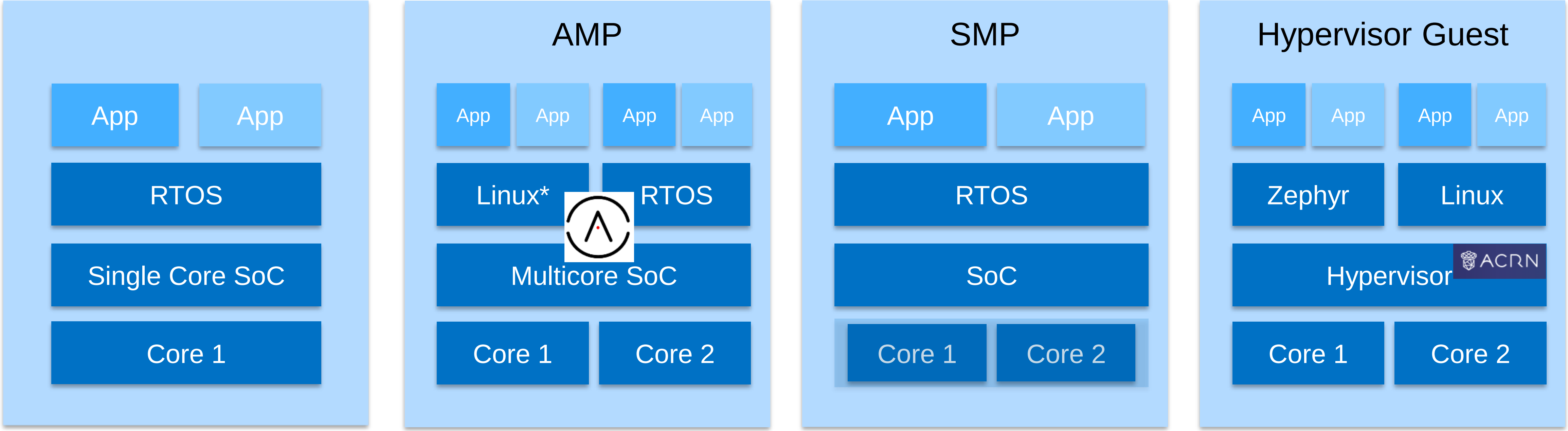
BOSE

SiFive

antmicro

Linino.org

Zephyr*: Use Cases



Single Core MCU

Supported with OpenAMP

Supported on Xtensa* and x86_64

Supported with ACRN

*Other names and brands may be claimed as the property of others.

Zephyr: More Than 170 Supported Boards

FRDM K64F

Arduino* Due

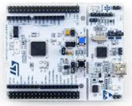
Nucleo 103RB

NRF51

Nucleo64 L476RG

Nucleo F411RE

NRF52 pca10040


Nucleo F334R8

Arduino 101*

Minnowboard

MAX® 10 FPGA

Nucleo 401RE

Hexiwear*

ARM* V2M MPS2

STM32* 10c

Atmel* SAM E70

Intel® Galileo

Synopsys* ARC EMSK

NRF52

Seeed* Carbon

TI* Launchpad Wi-Fi

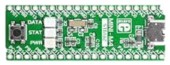
BBC Micro:bit*

STM32* 373c

Redbear BLE Nano

Intel® Quark™ Processor

STM32* Olimexino

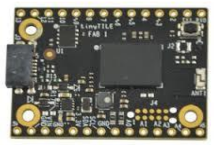

STM* Mini A15

Seeed* Nitrogen

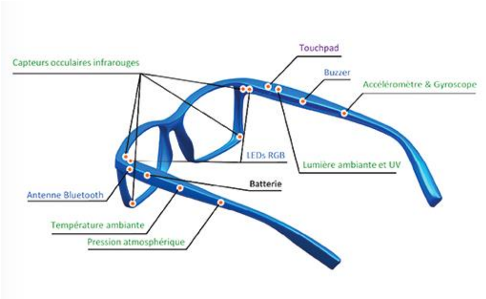
ARM* V2M Beetle

Zedboard Pulpino

FRDM-KW41Z


tinyTILE

Products Running Zephyr Today



Ellcie-Healthy Smart Connected Eyewear

Rigado IoTGateway

ProGlove Scanning Gloves

Adero tracking devices

RUUVI node

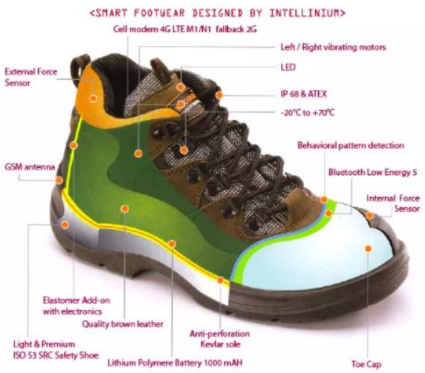
Point Home Alarm

GNARBOX 2.0 SSD

GEPS

Grush Gaming Toothbrush

hereO Smartwatch


<SMART FOOTWEAR DESIGNED BY INTELLINIUM>
Cell modem 4G LTE M1/N1 fallback 2G
Left / Right vibrating motors
LED
External Force Sensor
IP 68 & ATEX
-20°C to +70°C
Behavioral pattern detection
GSM antenna
Bluetooth Low Energy 5
Internal Force Sensor
Elastomer Add-on with electronics
Quality brown leather
Anti-perforation Kevlar sole
Light & Premium ISO S3 SRC Safety Shoe
Lithium Polymere Battery 1000 mAH
Toe Cap

Intellinium Safety Shoes

Anicare reindeer tracker

HereO Core Box

Zephyr In Open Source RTOS Landscape*

#2

Total Contributors

Rank	RTOS	#
1	mbed OS	532
2	**Zephyr**	**509**
3	nuttX	315

#2

Total Commits

Rank	RTOS	#
1	nuttX	39,013
2	**Zephyr**	**32,206**
3	mbed OS	25,574

#1

Commits to Master (last 30 days)

Rank	RTOS	#
1	**Zephyr**	**900**
2	mbed OS	269
3	RIOT	165

* as of 2019/7/25

Functional Safety And Zephyr OS

Functional Safety: Introduction

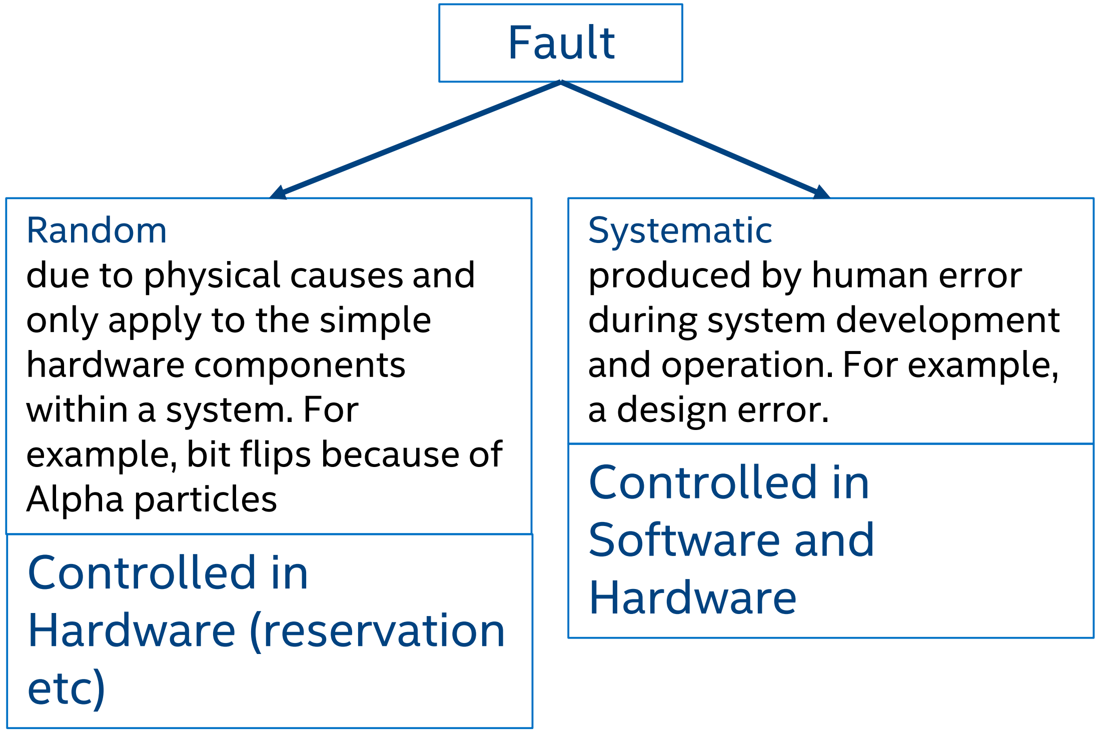
Applicable to active physical systems

Functional Safety

Ability of the system to react on potentially dangerous condition by using safety function and reduce the risk.

Example

The detection of smoke by sensors and the ensuing intelligent activation of a fire suppression system



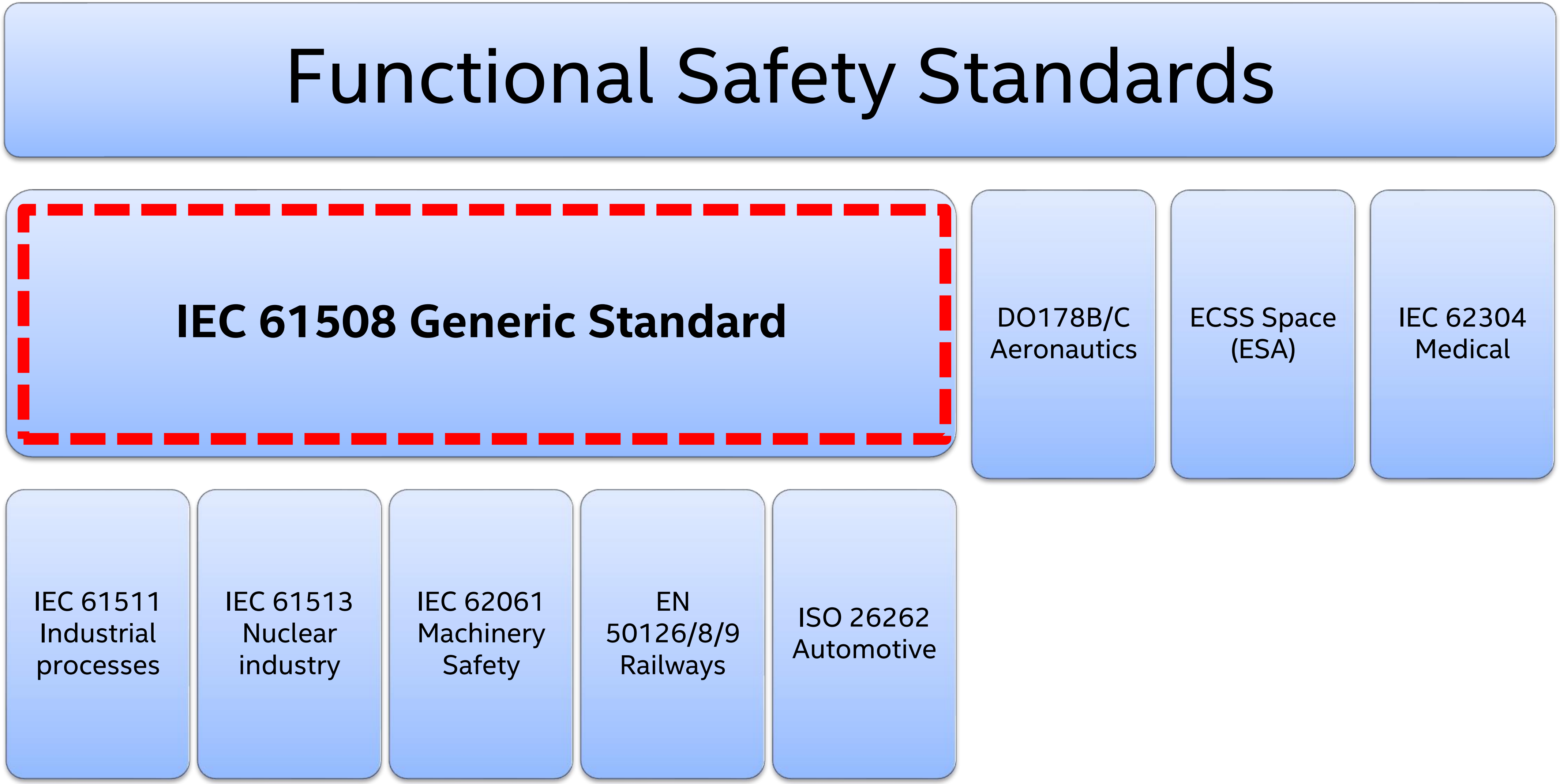
Fault

Random
due to physical causes and only apply to the simple hardware components within a system. For example, bit flips because of Alpha particles

Controlled in Hardware (reservation etc)

Systematic
produced by human error during system development and operation. For example, a design error.

Controlled in Software and Hardware



Functional Safety Standards

IEC 61508 Generic Standard

DO178B/C Aeronautics

ECSS Space (ESA)

IEC 62304 Medical

IEC 61511 Industrial processes

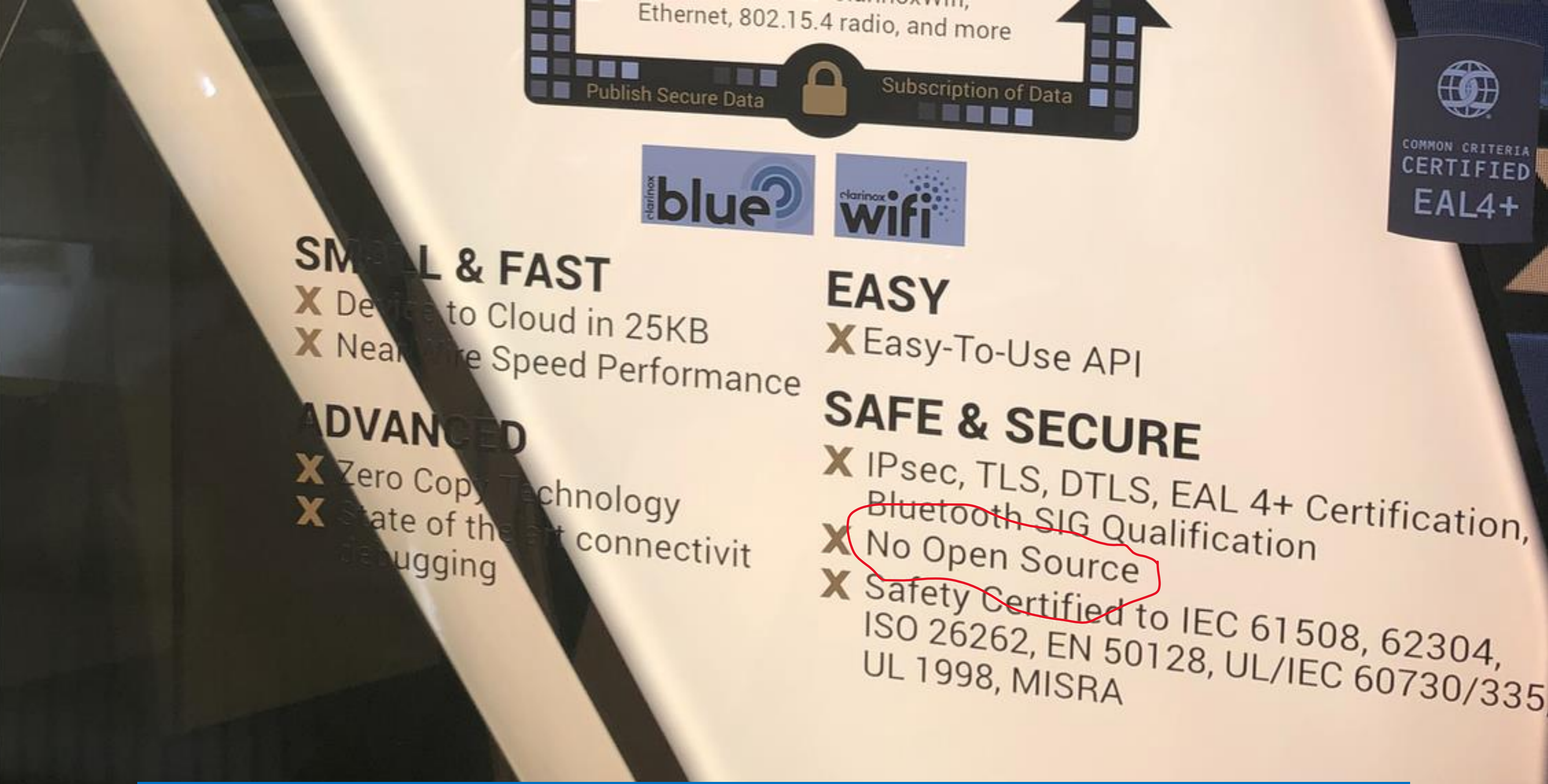
IEC 61513 Nuclear industry

IEC 62061 Machinery Safety

EN 50126/8/9 Railways

ISO 26262 Automotive

“The nice thing about standards is that there are so many of them to choose from.” [Tanenbaum]



Ethernet, 802.15.4 radio, and more

Publish Secure Data

Subscription of Data

clarinox blue

clarinox wifi

COMMON CRITERIA CERTIFIED EAL4+

SMALL & FAST

- X Device to Cloud in 25KB
- X Near Wire Speed Performance

ADVANCED

- X Zero Copy Technology
- X State of the art connectivit debugging

EASY

- X Easy-To-Use API

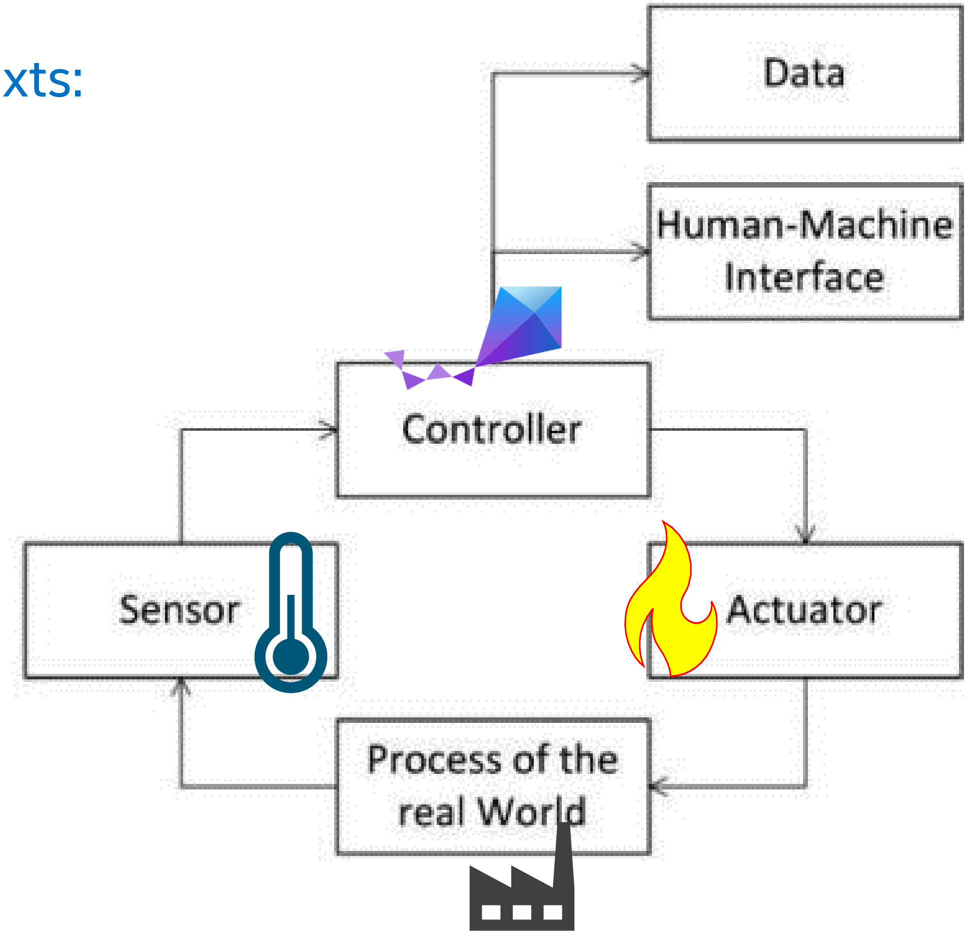
SAFE & SECURE

- X IPsec, TLS, DTLS, EAL 4+ Certification, Bluetooth SIG Qualification
- X No Open Source
- X Safety Certified to IEC 61508, 62304, ISO 26262, EN 50128, UL/IEC 60730/335, UL 1998, MISRA

No Open Source as a feature?

Why Consider Safety Standards For Zephyr?

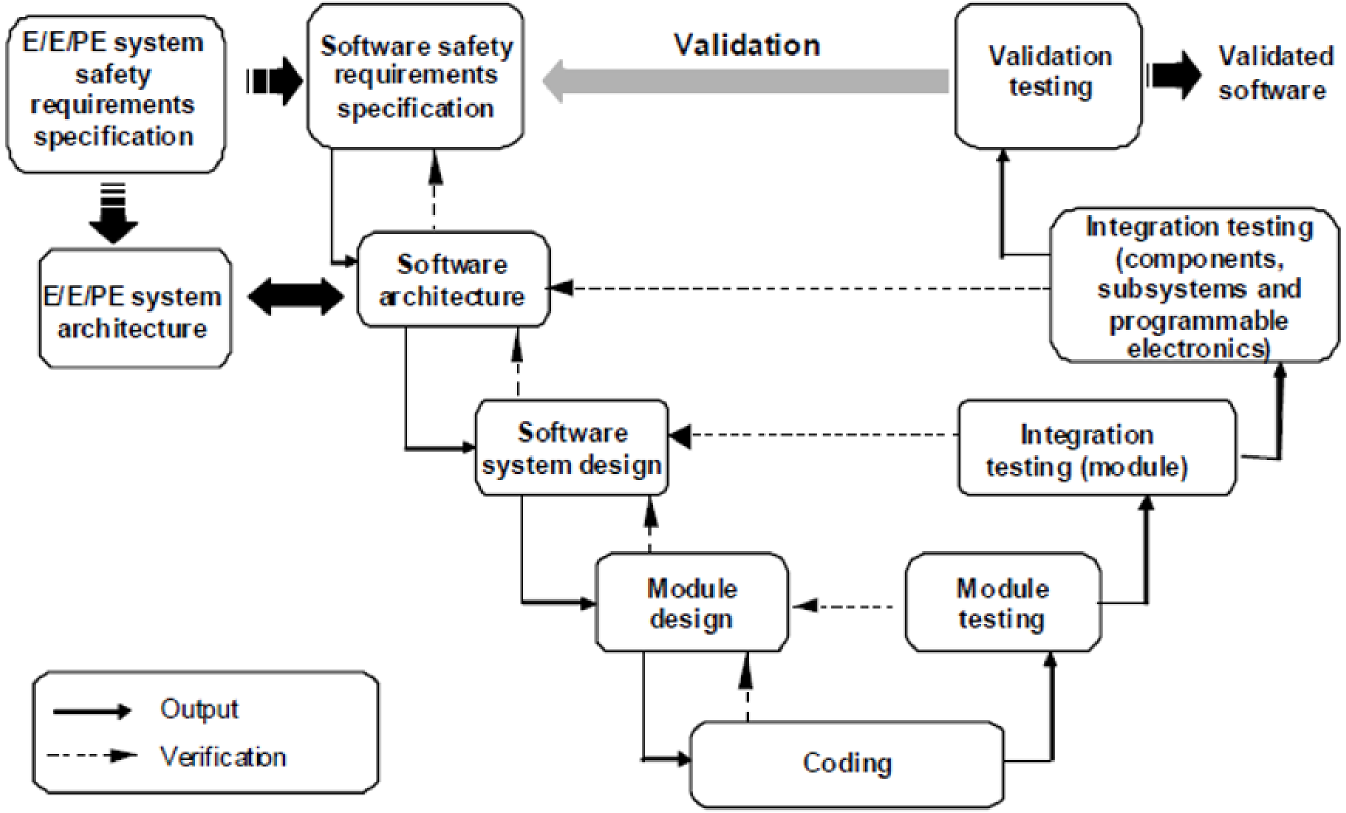
- We want to see Zephyr used in safety-critical contexts:
  - Medical
  - Industrial/manufacturing
  - Transportation/automotive
  - Power generation
  - Aerospace
- No Open Source OS is safety certified



Functional Safety And Software

Phase-oriented Lifecycle, per IEC 61508

- To control systematic failures
- Every phase has specific requirements



E/E/PE system safety requirements specification
Software safety requirements specification
Validation
Validation testing
Validated software
E/E/PE system architecture
Software architecture
Integration testing (components, subsystems and programmable electronics)
Software system design
Integration testing (module)
Module design
Module testing
Coding
Output
Verification

There is no known way to prove the absence of failures in reasonable complex software

Certifying Existing Software

For 61508 certification of as pre-existing software (IEC 61508-3, 7.4.2.12, “Route 3S”) , assessment needed for:

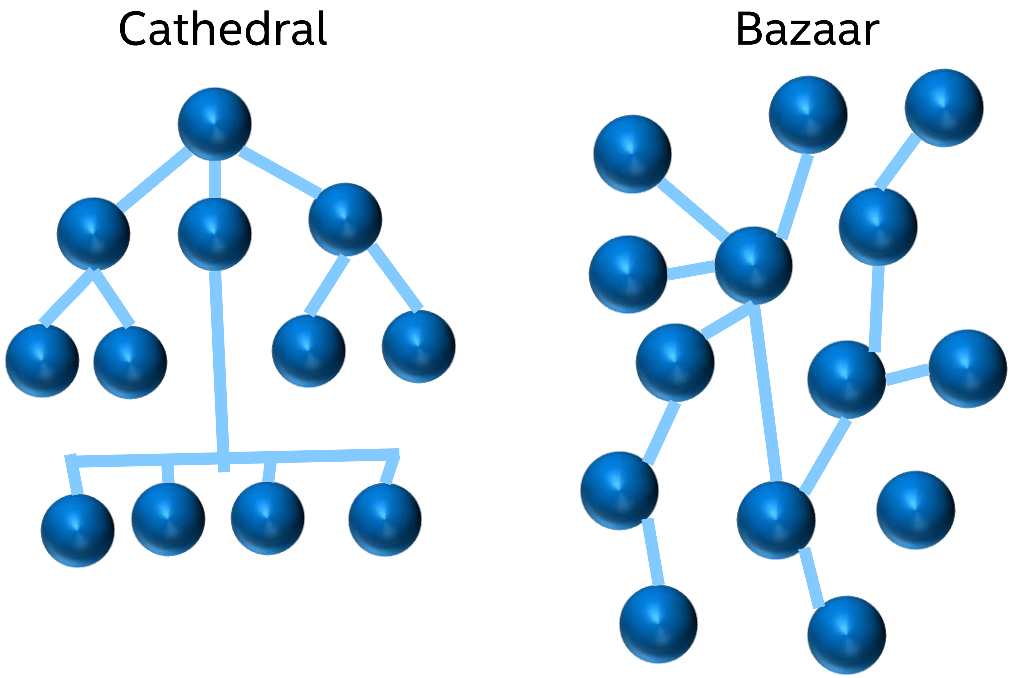
- Requirements specification and traceability
- Documentation on architecture, design and modules, coding standard
- Testing on module and integration level
- Validation of requirements
- Tools, reference hardware configuration

Photo https://pixabay.com/photos/evaluation-exam-passed-list-1516644/

Open Source And Safety Certification

Open source *software* is not a problem in itself

Photo: https://pixabay.com/photos/florence-market-outdoor-church-1964638/



The *process* of creating the software is:

- Functional Safety requires V model/phases
- Detailed specification of features
- Comprehensive documentation
- Traceability from requirements to source code
- Number of committers and information known about them
- Certification authority not familiar with open source development

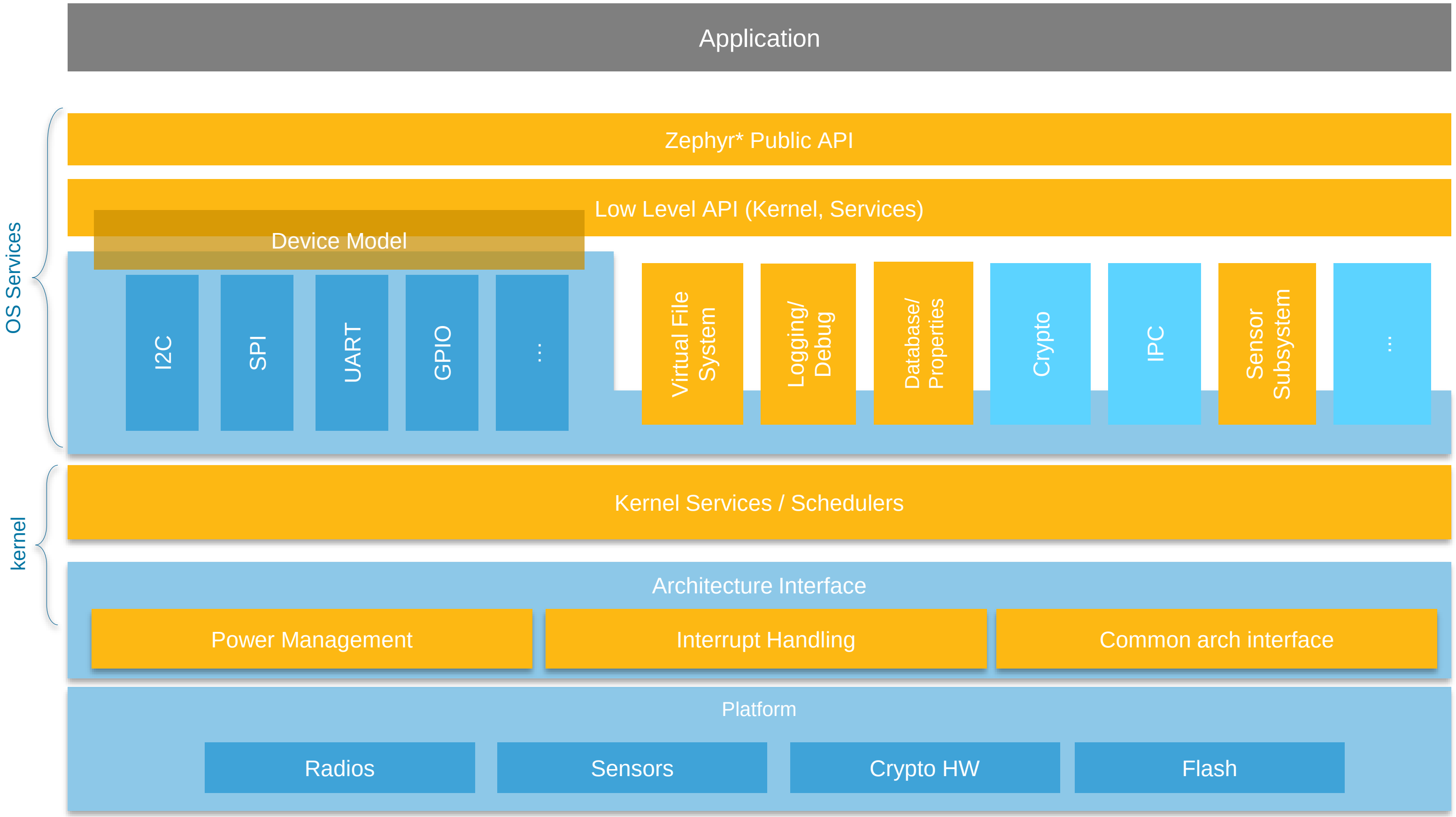
Our Approach

- Snapshotting a Source Tree (branch), validating it then controlling updates is a viable approach to software qualification
  - Build a cathedral on top of (or beside) the bazaar
- Getting supported feature set right is most important up front decision
  - The more you support, the more documentation and testing you are going to provide
- Automate as much of the information tracking as you can
- Auto-generate documents from test and issue tracking systems
- Get proof of concept approval from a certification authority as early as possible

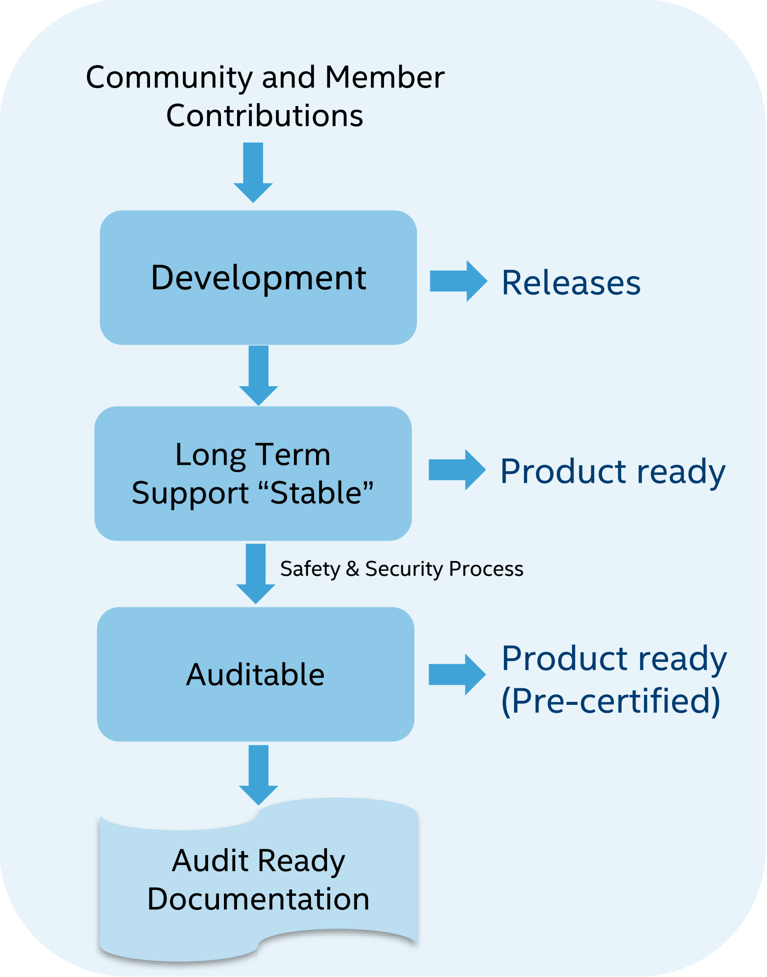
Photo: https://pixabay.com/photos/florence-market-outdoor-church-1964638/

Scope Of Certification: Zephyr Kernel + Services

- Initial scope
  - Kernel
  - Logging
  - VFS
  - Properties/database
  - Device model
- Only using well-defined and stable APIs



Zephyr* Approach: Auditable Code Base

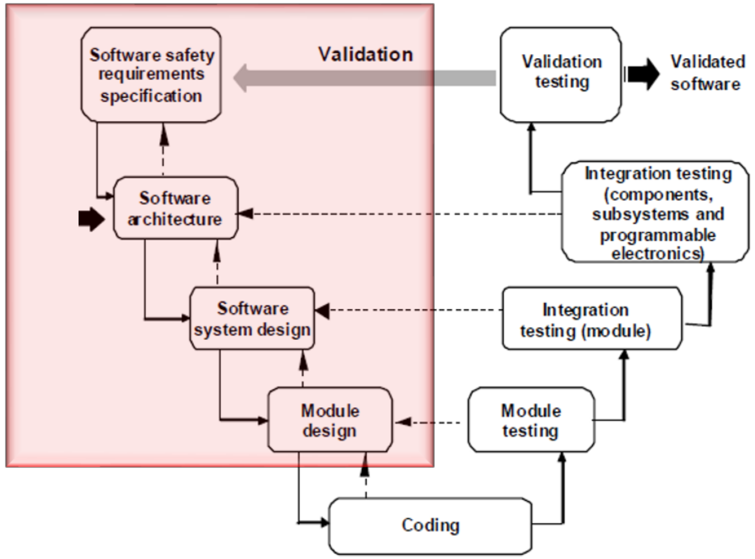

Community and Member Contributions
Development
Releases
Long Term Support “Stable”
Product ready
Safety & Security Process
Auditable
Product ready (Pre-certified)
Audit Ready Documentation

- An auditable code base shall be established from a subset of Zephyr* OS features.
- Both code bases shall be kept in sync from that point forward.
- More rigorous processes (necessary for certification) will be applied before new features move into the auditable code base.

*Other names and brands may be claimed as the property of others.

Requirement Traceability

- Needs formal requirements
- Multiple levels, satisfaction links from decomposed requirements
- Verification links from related tests
- Implementation links from user stories



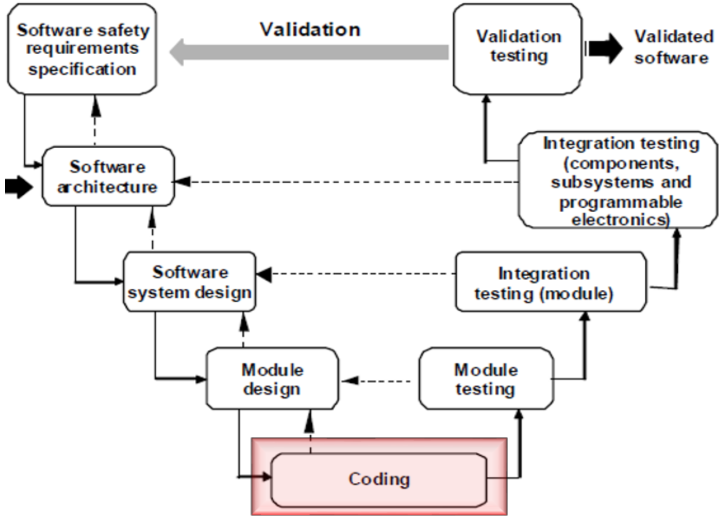
Software safety requirements specification
Validation
Validation testing
Validated software
Software architecture
Integration testing (components, subsystems and programmable electronics)
Software system design
Integration testing (module)
Module design
Module testing
Coding

Zephyr status

- WIP
- High-level requirements are being created (post-factum, manually)
- Decomposition is being created
- Connection between requirements and code will be partially automated based on the test coverage analysis

Code reviews

- Code is available publicly and can be scrutinized by anyone.
- Code reviews and direct user feedback help improve quality


Software safety requirements specification
Validation
Validation testing
Validated software
Software architecture
Integration testing (components, subsystems and programmable electronics)
Software system design
Integration testing (module)
Module design
Module testing
Coding

However…

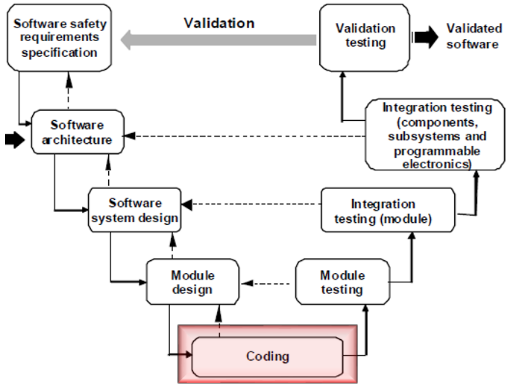
- Do we have the right set of reviewers?
- Who gets to have the final say?
- How do we guarantee that the reviewer is aware of safety implications?
- For how long should changes be reviewed?

Zephyr status

- Process WIP
- Current assumptions:
  - committers to Auditable to be trained for FuSa
  - Well-defined list of module responsible

Coding Standard: MISRA-C

- Certification does not mandate MISRA-C compliance
- … but it is a de-facto standard for embedded safety, last release 2012
- ~180 guidelines. Some are mandatory, some are required unless a deviation report duly filed, some are advisory
- Commercial! (15 GBP per copy)
- Some rules are controversial
- Require right tooling to validate

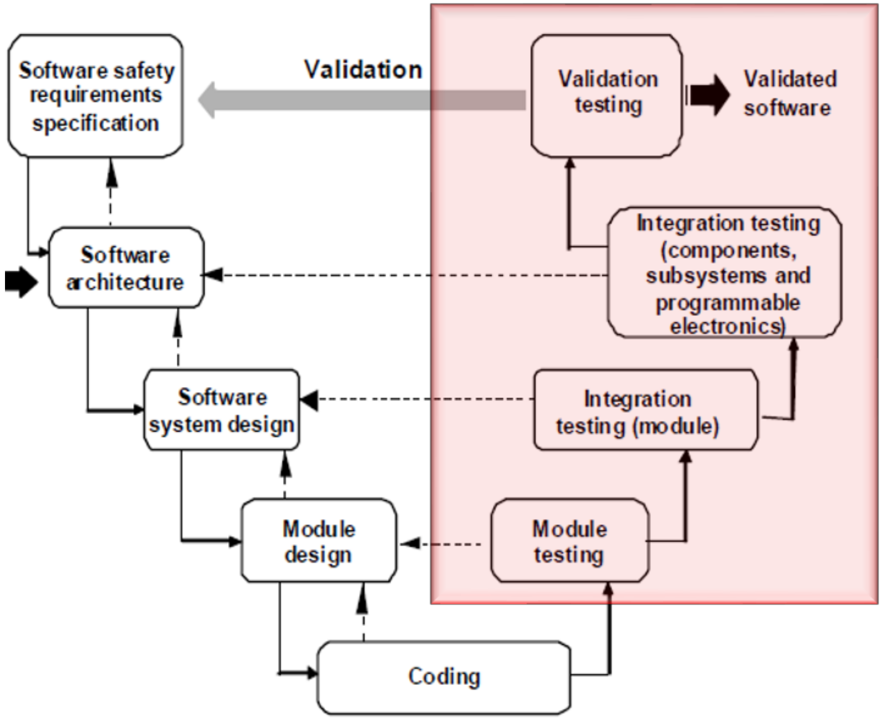

Software safety requirements specification
Validation
Validation testing
Validated software
Software architecture
Integration testing (components, subsystems and programmable electronics)
Software system design
Integration testing (module)
Module design
Module testing
Coding

Zephyr status

- Standard is WIP, based on Misra-C, Cert-C and JPL standards
- Deviations are key
- Rigorous standard compliance will have limited scope
- Will be part of Zephyr contribution guidelines and CI
- Some MISRA-C rules already applied to Zephyr kernel using Coccinelle

Quality Management

- Quality Management System is a **mandatory** expectation for software across the industry.
- Software QMS **is not** an additional requirement caused by functional safety standards.
- Functional safety considers QMS as an **existing pre-condition**.
- Quality Managed (QM) status should be the aspiration of any open source project, regardless of functional safety or certification goals.


Software safety requirements specification
Validation
Validation testing
Validated software
Software architecture
Integration testing (components, subsystems and programmable electronics)
Software system design
Integration testing (module)
Module design
Module testing
Coding

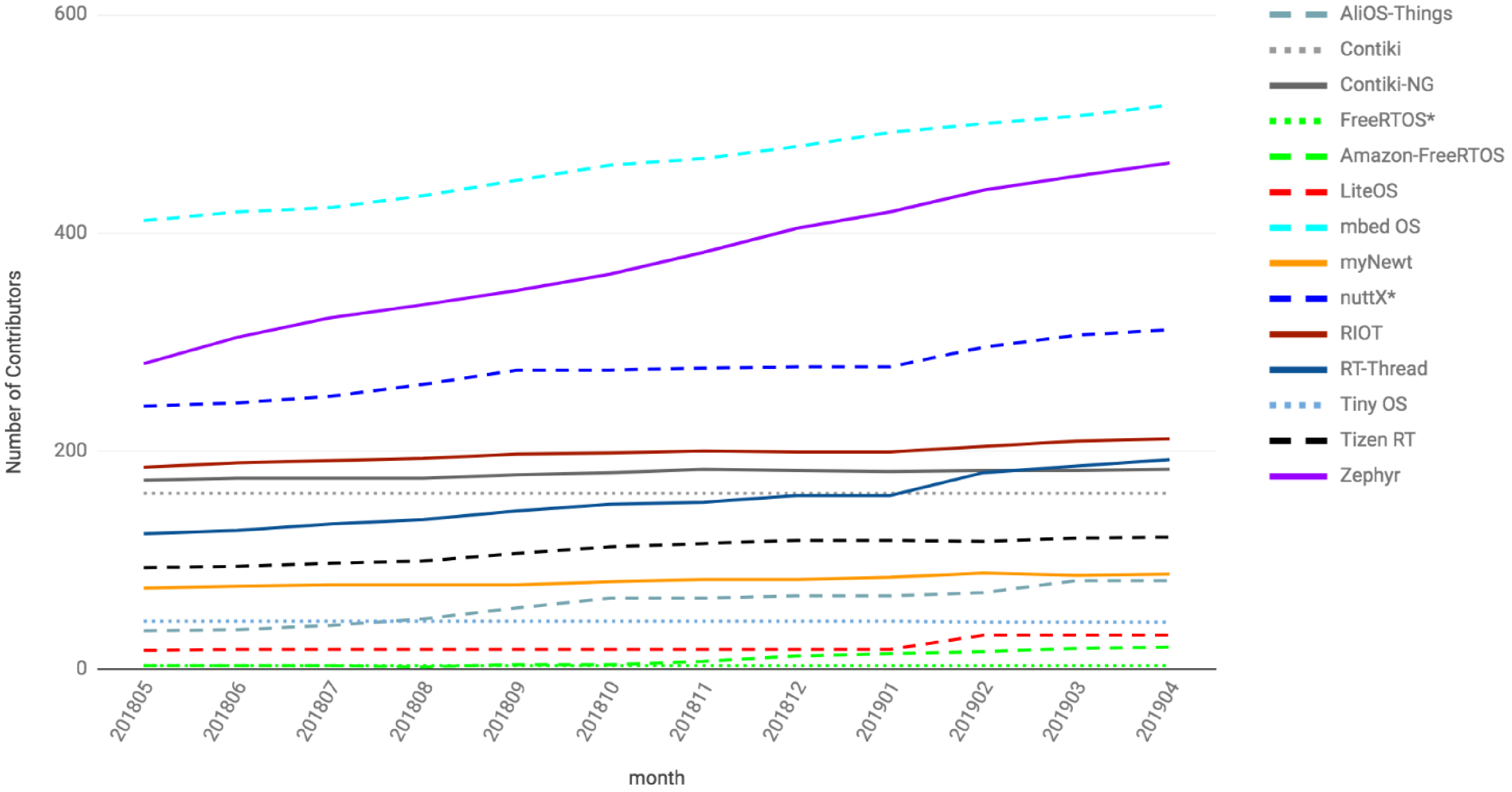
Summary

- Certifying Open Source OS for functional safety and keeping it open:
  - Challenging
  - Doable!
- Was never done before, we are paving the way
- Companies in Zephyr working *together* to make it
- Working hard to ensure project’s community buy-in

THANK YOU!

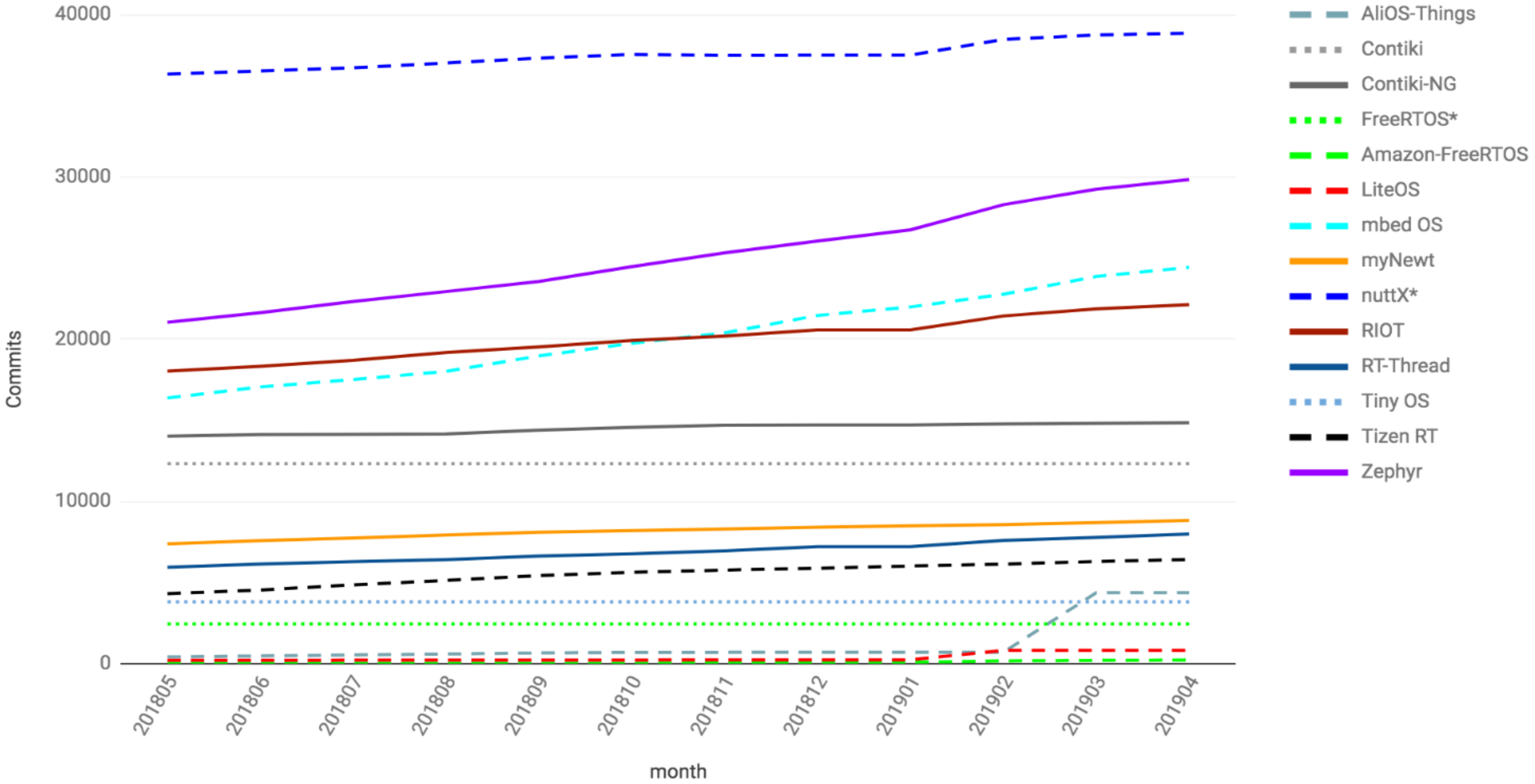
BACKUP

Operating System Contributors



Source: Data as of 2019-4-25 from github (* from openhub.net)

Total Commits by Operating System



Source: Data extracted on 2019-4-25 from github (* from openhub.net) intel®