Zephyr Project:

Open Source Project Best Practices Over Time

Kate Stewart
Feb 1, 2025

Who am I?

Embedded Open Source:

- Zephyr Project: 2016 →
- Real Time Linux: 2016 → 2024
- ELISA Project: 2018 →
- Space Grade Linux: 2024 →

Volunteer:

- SPDX: 2009 →
- SBOM: 2018 →

Hobbies:

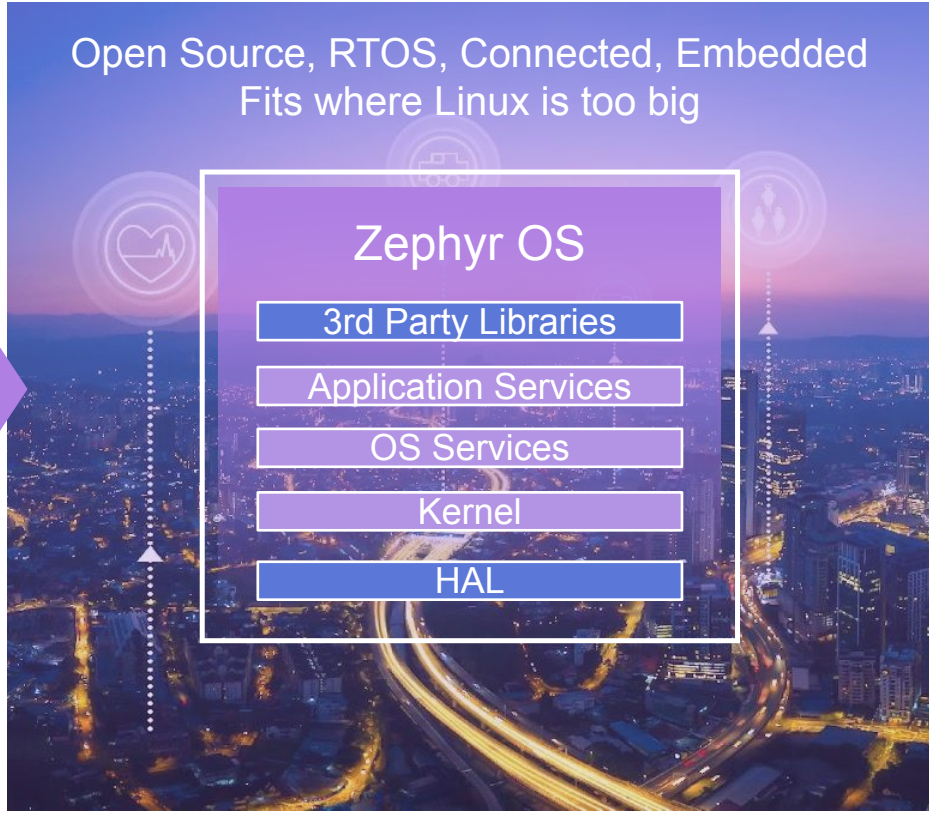
- Photography
- Travel to places with penguins

Contact:

kstewart@linuxfoundation.org
https://www.linkedin.com/in/katestewartaustin/

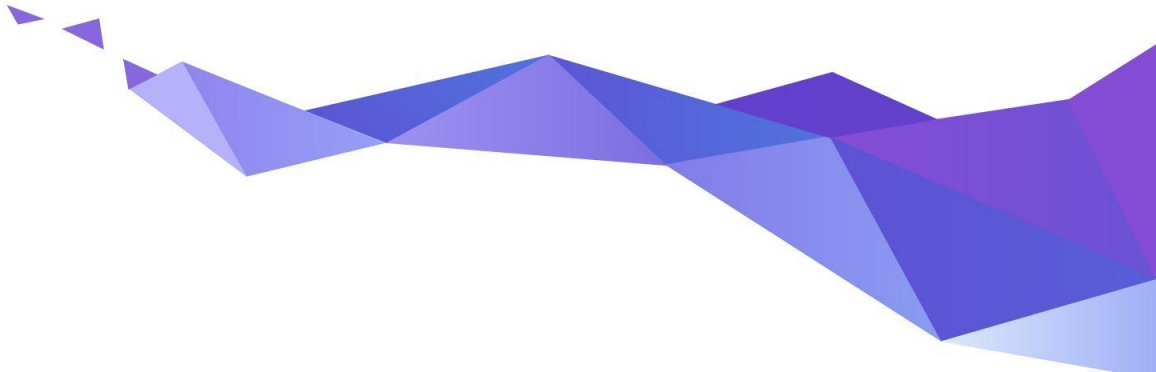
Zephyr Project

- **Open source** real time operating system
- **Developer friendly** with vibrant community participation
- Built with **safety and security** in mind
- **Broad SoC, board and sensor support**.
- **Vendor Neutral** governance
- **Permissively licensed** - Apache 2.0
- **Complete**, fully integrated, highly configurable, **modular** for **flexibility**
- **Product** development ready using LTS includes **security updates**
- **Certification** ready with Zephyr Auditable



THE LINUX FOUNDATION PROJECTS

Zephyr in 2024?

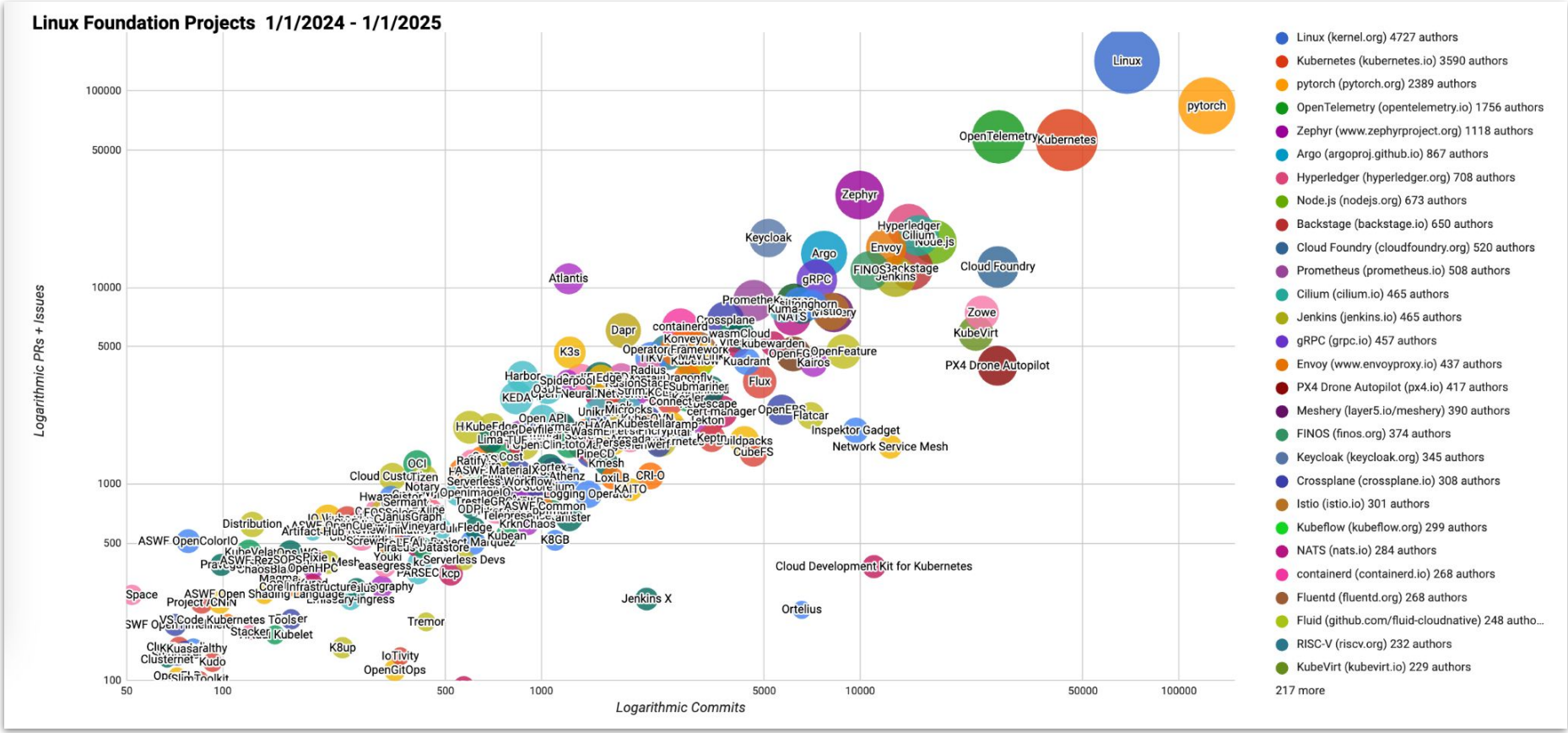


Source: https://zephyrproject.org/zephyr-rtos-2024-wrap-up-a-year-of-growth-innovation-and-community-impact/

Source: https://zephyrproject.org/zephyr-rtos-2024-wrap-up-a-year-of-growth-innovation-and-community-impact/

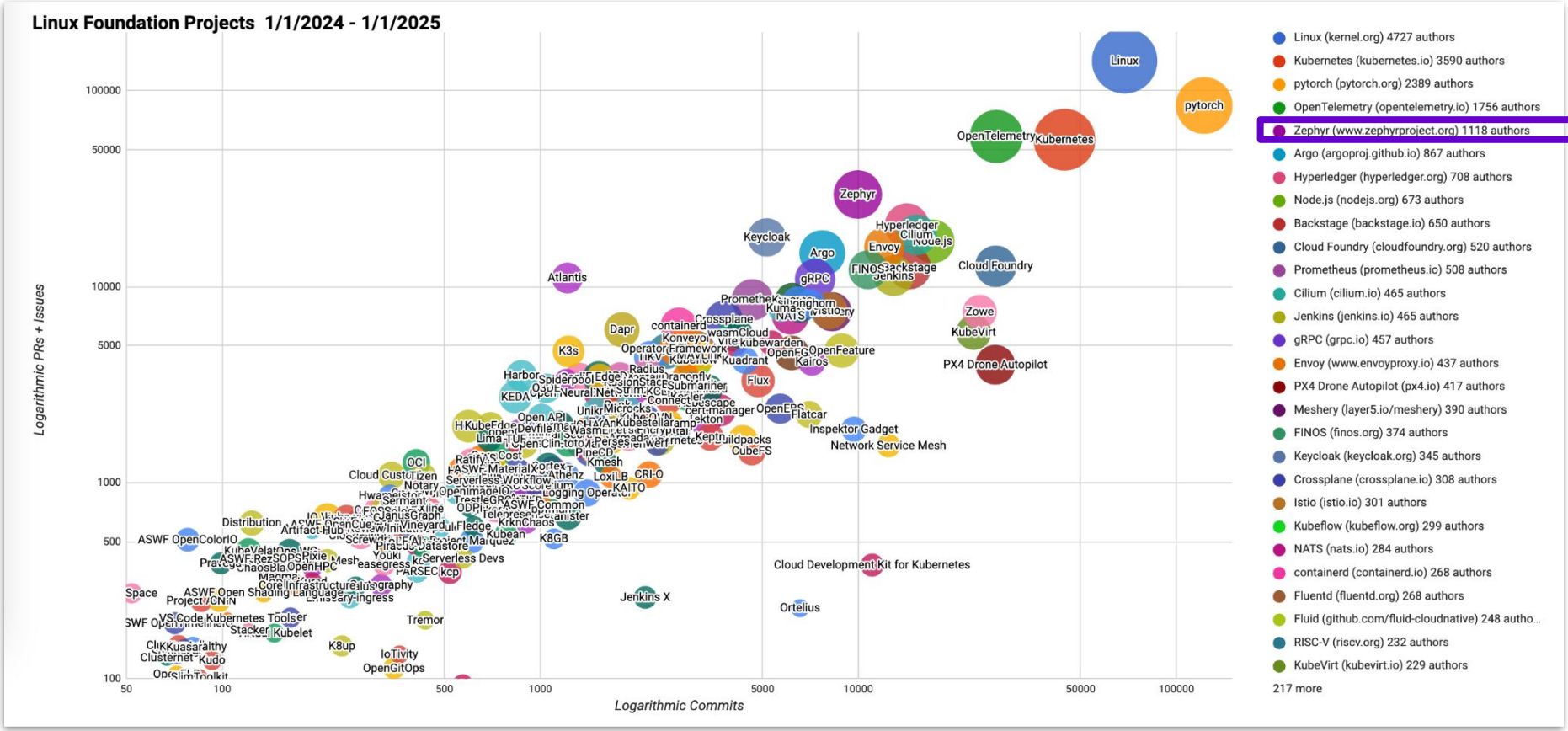
Source: https://zephyrproject.org/zephyr-rtos-2024-wrap-up-a-year-of-growth-innovation-and-community-impact/

Linux Foundation Projects Velocity



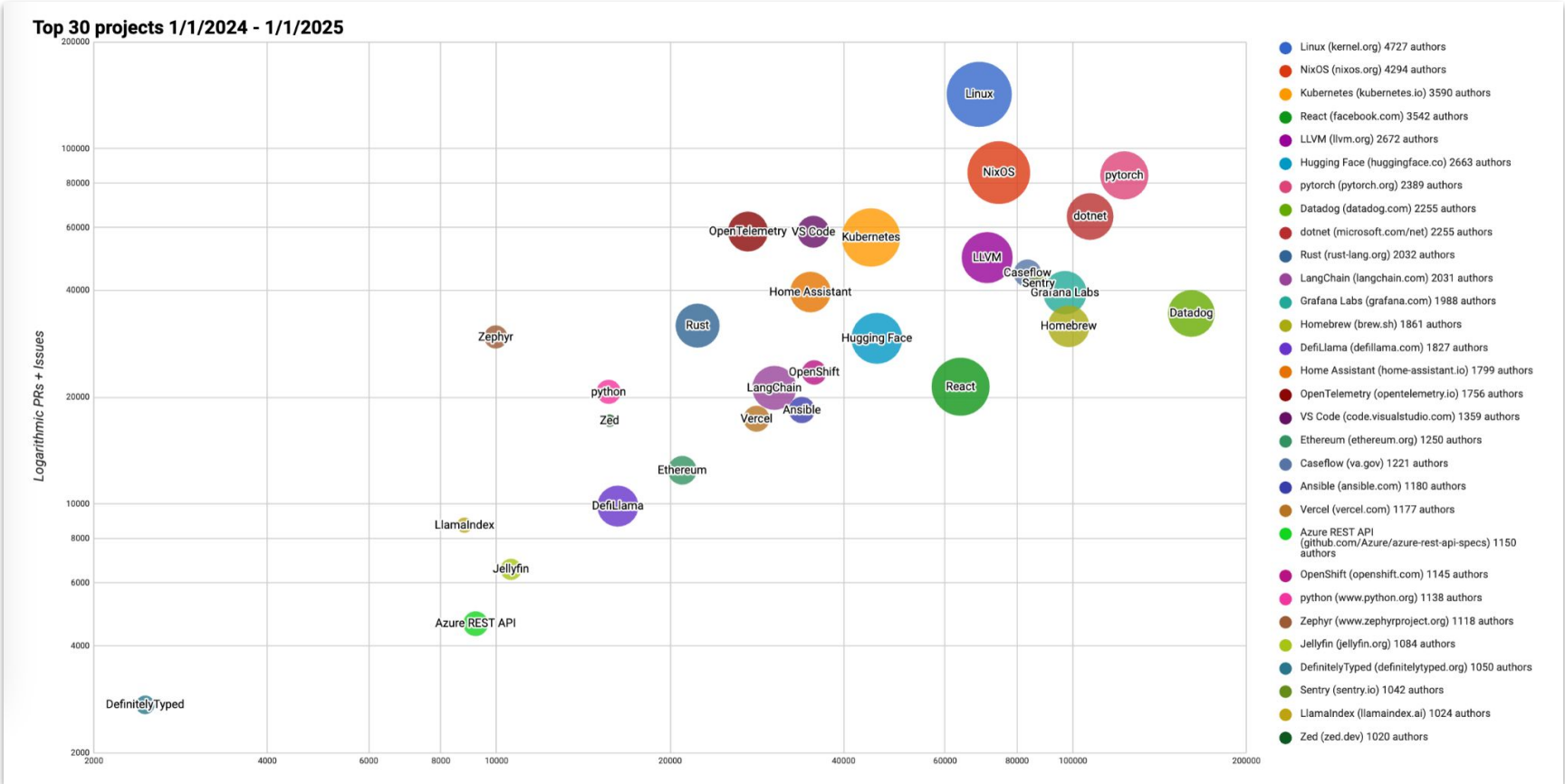
Source: https://github.com/cncf/velocity

Linux Foundation Projects Velocity



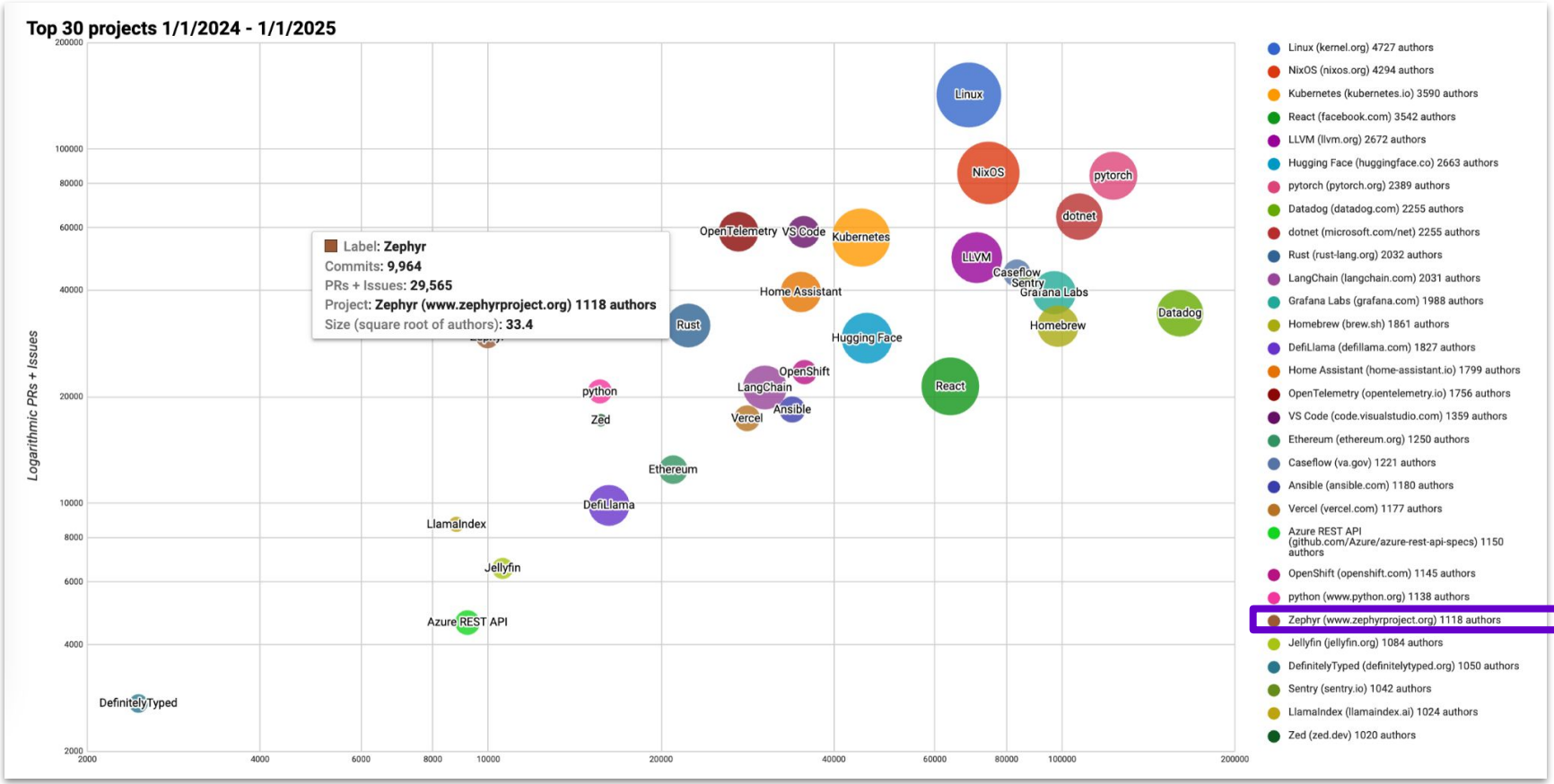
Source: https://github.com/cncf/velocity

Top Open Source Projects Velocity



Source: https://github.com/cncf/velocity

Top Open Source Projects Velocity



Source: https://github.com/cncf/velocity

Open Source RTOS Ecosystem

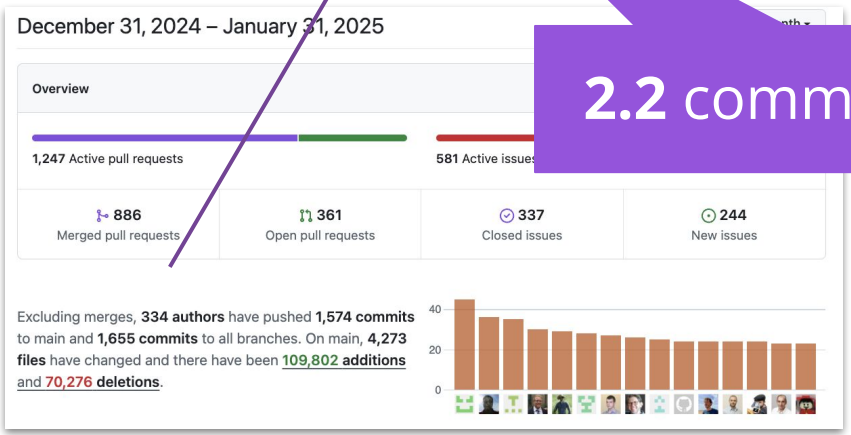
Operating System	First Commit	Controls Commits	Declared License	Total Contributors	Contributors in last month	Total Commits	Commits in last month
Zephyr	2014/11	**community**	**Apache-2.0**	**2476**	**334**	**109,429**	**1,655**
nuttX	2007/?	community	BSD-variant → Apache-2.0	593	62	57,664	357
RT-Thread	2009/06	community	GPL-2.0 → Apache-2.0	729	28	16,855	78
Tizen RT	2015/04	Samsung	BSD-variant → Apache-2.0	203	23	11,557	79
RIOT	2010/09	community	LGPL-2.1	368	19	47,062	82
FreeRTOS	2004/07	Richard Barry	GPL-2.0 w/ FreeRTOS → MIT	207	9	3,565	14
Contiki-NG	2017/10	community	BSD-3-Clause	219	3	17,975	8
SeL4	2014/07	community	GPLv2 AND BSD-2-Clause	113	2	4,615	3
myNewt	2015/06	community	Apache-2.0	135	2	11,143	3
mbed OS	2013/02	ARM	Apache-2.0 or BSD-3-Clause	692	0	34,621	0
ThreadX	2020/05	MSFT → community	MSL → MIT	21	0	208	0

Data extracted on 2025-01-31 from github

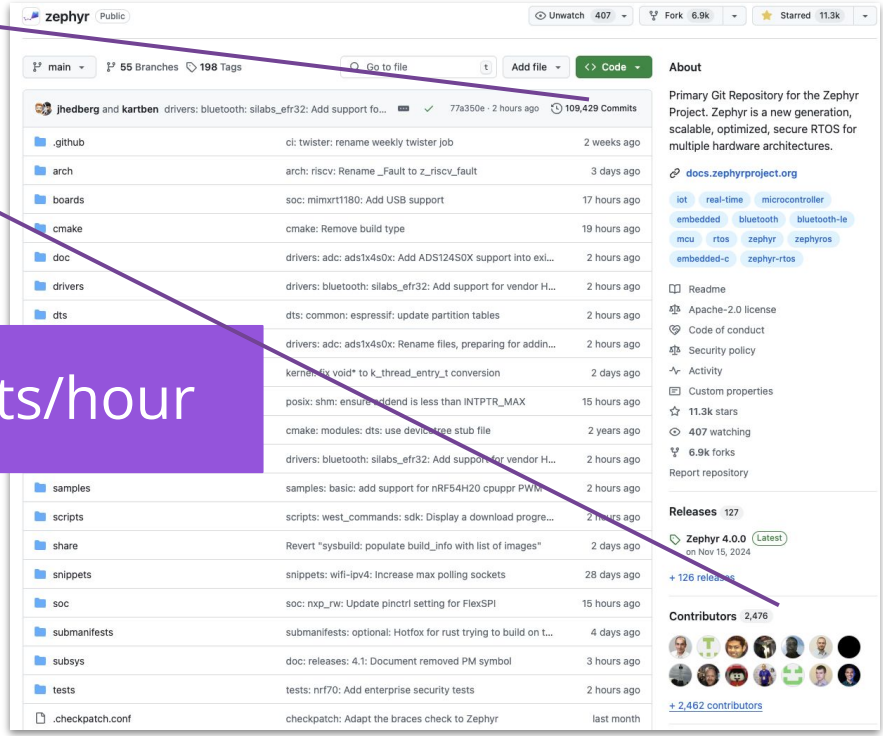
Methodology: Sample from Github

https://github.com/zephyrproject-rtos/zephyr
- Total commits: 109,429
- Total contributors: 2,476

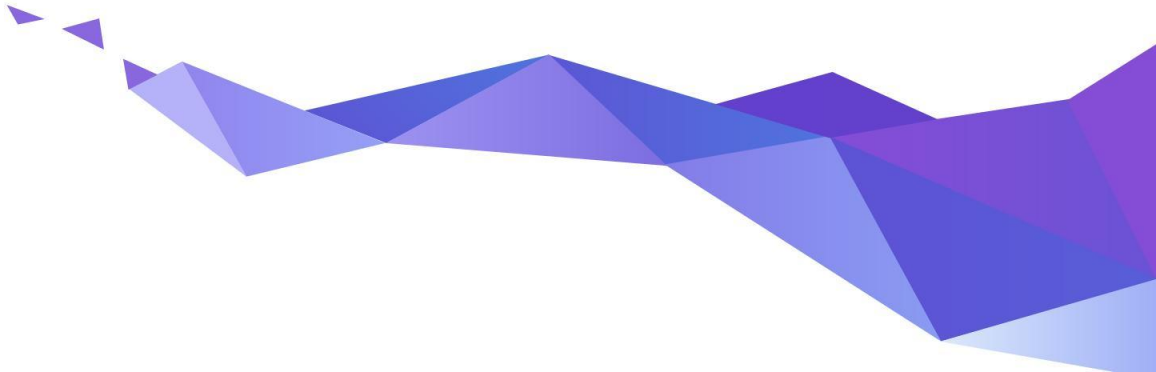
https://github.com/zephyrproject-rtos/zephyr/pulse/monthly
- Monthly contributors: 334
- Monthly commits: 1,655



2.2 commits/hour



How does this compare to the Linux Kernel?



How does this compare to Linux?

6.8 Linux Kernel Statistics*

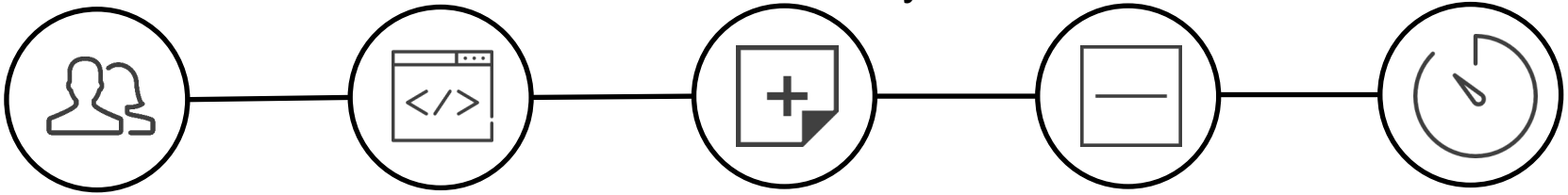
1,938*
Contributors From
219 Organizations

7,064
Lines of Code
Added Daily

1,945
Lines of Code
Modified Daily

1,281
Lines of Code
Removed Daily

9
Changes Per
Hour



* Source: https://lwn.net/Articles/964106/ Time period for 6.8: 2024/1/8-2024/3/10=63 days
Also data from: Source: https://github.com/gregkh/kernel-history/blob/master/kernel_stats.ods from 6.5

So what was it like when Linux started?

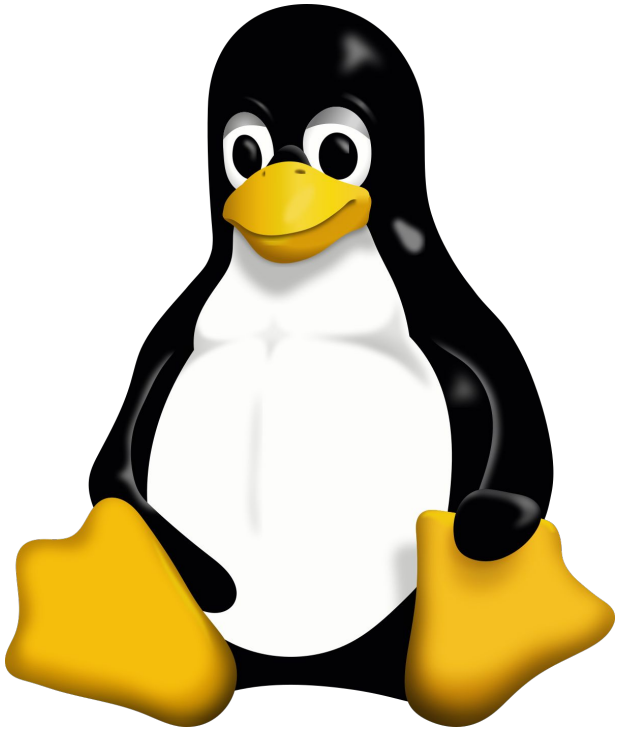
When Linux Started in 1991…

UNIX Source Available:

SVR4, MINIX 1.5, 4.3BSD

Commercial Distributions:

A/UX, IBM AIX, Dec Ultrix,
HP-UX, IRIX, SunOS, MIPS
RISC/os, Xenix …



What is Linux like Today?

100% Supercomputer Market

82% Smartphone Market Share

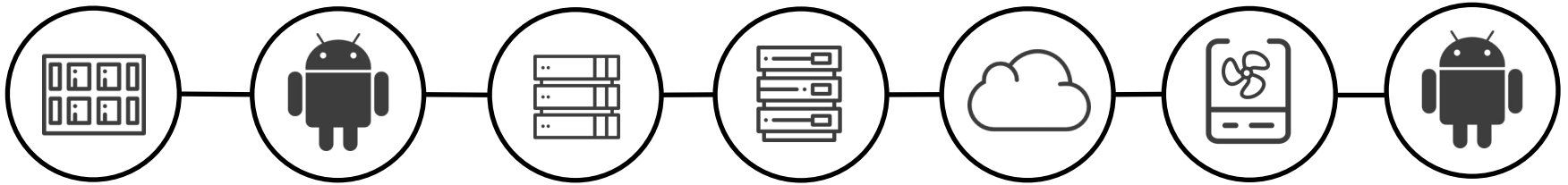
2nd To Windows in Enterprise

90% Mainframe Customers

90% Public Cloud Workload

69% Embedded Systems Market

#1 Internet Client



Every market Linux has entered it eventually dominated

Lessons Learned by Linux Community circa 2016/2017

Source:
https://www.linuxfoundation.org/resources/publications/linux-kernel-report-2017

More recent stats can be found at:
https://www.linuxfoundation.org/tools/linux-kernel-history-report-2020/

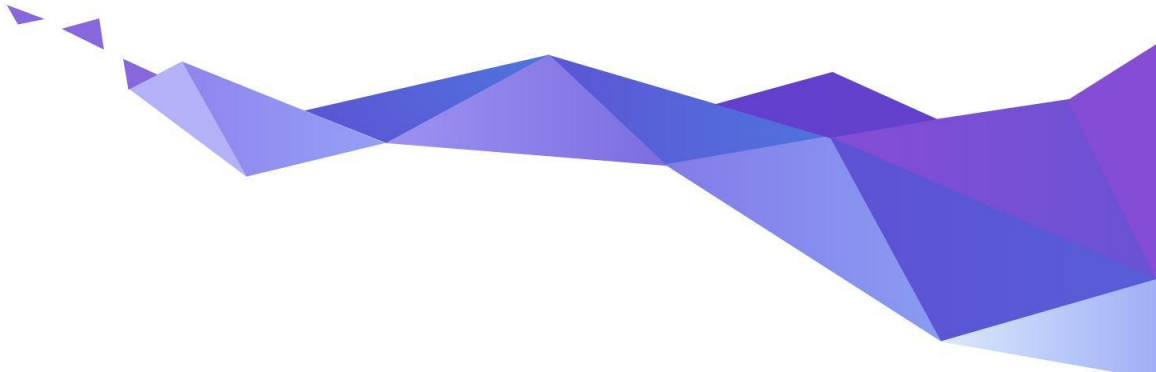
- Short release cycles are important.
- Process scalability requires a distributed, hierarchical development model.
- Tools matter.
- The kernel's strongly consensus-oriented model is important.
- A related factor is the kernel's strong "no regressions" rule.
- Corporate participation in the process is crucial.
- There should be no internal boundaries within the project

++ Lessons Learned

- **Vendor-neutral environment for technical decision making**
- Mix of companies and individuals participating – “scratching their itches”
- **Streamline upstreaming process** – DCO - “signed-off-by:”
- **Public code reviews** – “reviewed-by:”
- Consensus-oriented decision model – email, in-person summits
- Hierarchical development model (**maintainer model**) – “signed-off-by”
- No internal boundaries – developer can contribute anywhere
- **Tools matter** - git enabled distributed version control - push/pull
- Short predictable release cycles and **with fixed merge windows**
- **Stable & LTS:** stable and long term support releases support product development

KEY: Developer frustration with status quo inspires creative solutions.

So what lessons did Zephyr apply from the Linux Kernel Best Practices?



Zephyr's Vision

The Zephyr Project strives to deliver the **best-in-class RTOS** for **connected resource-constrained** devices, built to be **secure and safe**.

Developers Decide Directions

Zephyr

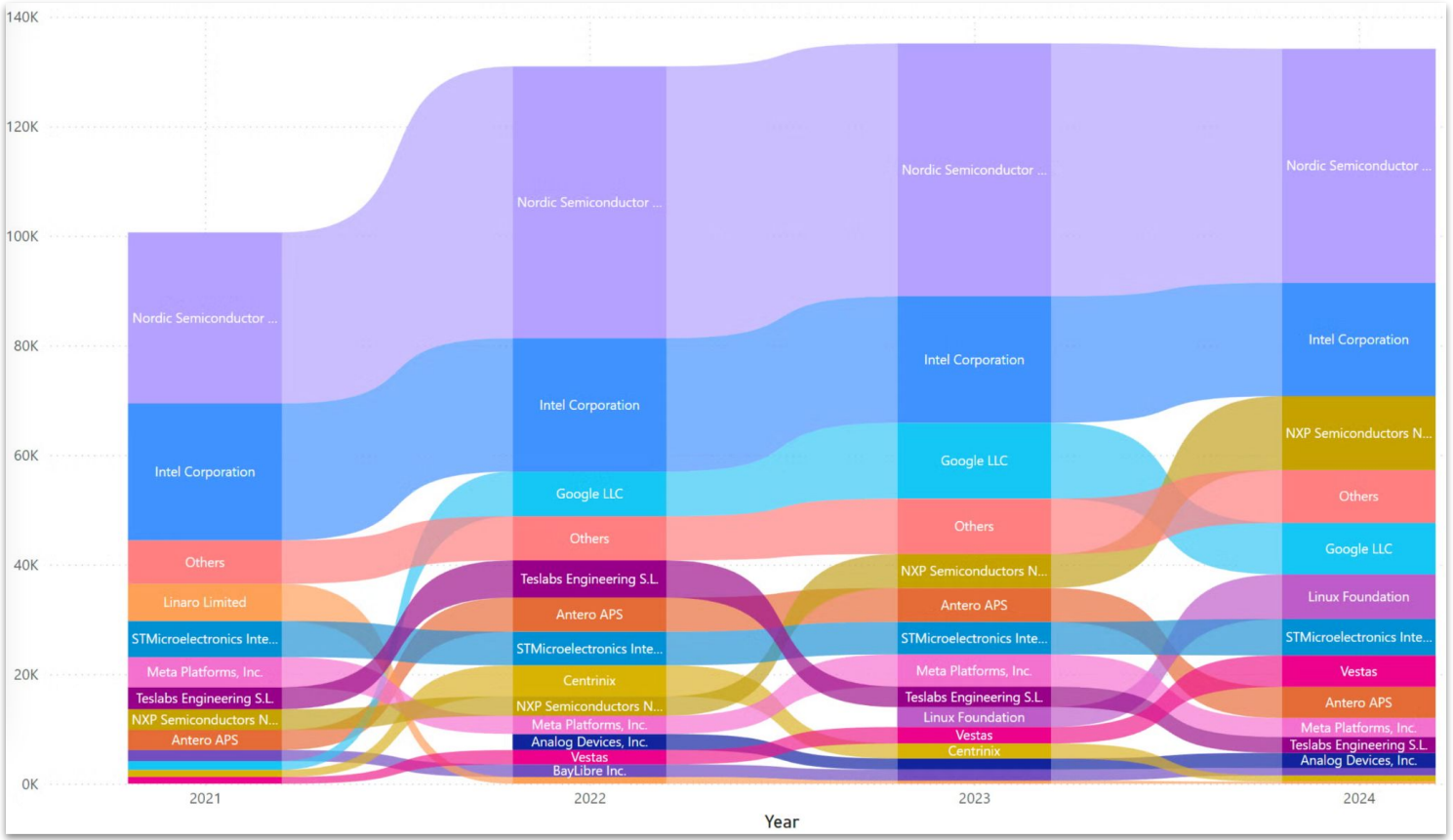
- **Configuration:** kconfig & kbuild added in 2015 prior to launch
- **Unified kernel:** nano + microkernels → unified kernel in 2016
- **Infrastructure:** Gerrit/JIRA → GitHub/Issues in 2017
- **Build system:** kbuild → cmake in 2018
- **Code of Conduct:** adopted in 2018
- **Other areas:**
  - APIs & HALs - reworked
  - Modularization & Device Tree support
  - Release & LTS processes refined

Applying ++ Lessons Learned

Linux Best Practice	Zephyr Adoption
Vendor Neutral Decision Making	
Companies and Individuals Participate	
Streamline upstreaming process	
Public code reviews?	
Consensus Oriented Decision Models	
Hierarchical development (Maintainers)	
No Internal Boundaries	
Distributed version control	
Short Release Cycle (w/ Merge Window)	
Long Term Support Releases	

Applying ++ Lessons Learned

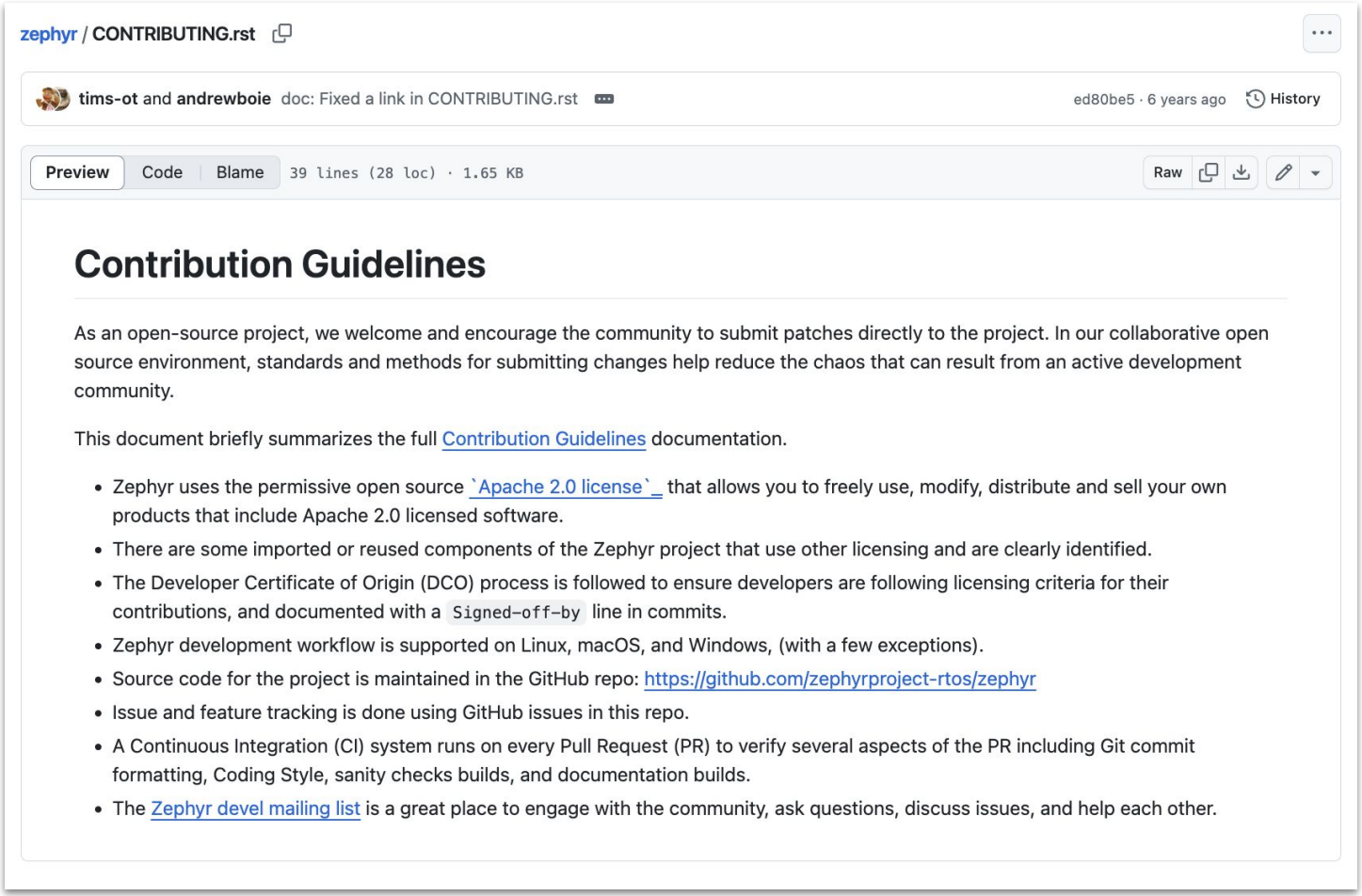
Linux Best Practice	Zephyr Adoption
Vendor Neutral Decision Making	Yes, Project support from multiple companies.
Companies and Individuals Participate	
Streamline upstreaming process	
Public code reviews?	
Consensus Oriented Decision Models	
Hierarchical development (Maintainers)	
No Internal Boundaries	
Distributed version control	
Short Release Cycle (w/ Merge Window)	
Long Term Support Releases	



Source: https://zephyrproject.org/zephyr-rtos-2024-wrap-up-a-year-of-growth-innovation-and-community-impact/

Applying ++ Lessons Learned

Linux Best Practice	Zephyr Adoption
Vendor Neutral Decision Making	Yes, Project support from multiple companies.
Companies and Individuals Participate	Yes, TSC has companies & community participation.
Streamline upstreaming process	
Public code reviews?	
Consensus Oriented Decision Models	
Hierarchical development (Maintainers)	
No Internal Boundaries	
Distributed version control	
Short Release Cycle (w/ Merge Window)	
Long Term Support Releases	



zephyr / CONTRIBUTING.rst

tims-ot and andrewboie doc: Fixed a link in CONTRIBUTING.rst

ed80be5 · 6 years ago History

Preview Code Blame 39 lines (28 loc) · 1.65 KB Raw

Contribution Guidelines

As an open-source project, we welcome and encourage the community to submit patches directly to the project. In our collaborative open source environment, standards and methods for submitting changes help reduce the chaos that can result from an active development community.

This document briefly summarizes the full Contribution Guidelines documentation.

- Zephyr uses the permissive open source `Apache 2.0 license`_ that allows you to freely use, modify, distribute and sell your own products that include Apache 2.0 licensed software.
- There are some imported or reused components of the Zephyr project that use other licensing and are clearly identified.
- The Developer Certificate of Origin (DCO) process is followed to ensure developers are following licensing criteria for their contributions, and documented with a `Signed-off-by` line in commits.
- Zephyr development workflow is supported on Linux, macOS, and Windows, (with a few exceptions).
- Source code for the project is maintained in the GitHub repo: https://github.com/zephyrproject-rtos/zephyr
- Issue and feature tracking is done using GitHub issues in this repo.
- A Continuous Integration (CI) system runs on every Pull Request (PR) to verify several aspects of the PR including Git commit formatting, Coding Style, sanity checks builds, and documentation builds.
- The Zephyr devel mailing list is a great place to engage with the community, ask questions, discuss issues, and help each other.

Source: https://github.com/zephyrproject-rtos/zephyr/blob/main/CONTRIBUTING.rst

Applying ++ Lessons Learned

Linux Best Practice	Zephyr Adoption
Vendor Neutral Decision Making	Yes, Project support from multiple companies.
Companies and Individuals Participate	Yes, TSC has companies & community participation.
Streamline upstreaming process	Yes, see /CONTRIBUTING.rst, **DCO** used
Public code reviews?	
Consensus Oriented Decision Models	
Hierarchical development (Maintainers)	
No Internal Boundaries	
Distributed version control	
Short Release Cycle (w/ Merge Window)	
Long Term Support Releases	

Applying ++ Lessons Learned

Linux Best Practice	Zephyr Adoption
Vendor Neutral Decision Making	Yes, Project support from multiple companies.
Companies and Individuals Participate	Yes, TSC has companies & community participation.
Streamline upstreaming process	Yes, see /CONTRIBUTING.rst, DCO used
Public code reviews?	Yes, issues & pull requests reviewed on https://github.com/zephyrproject-rtos/zephyr
Consensus Oriented Decision Models	
Hierarchical development (Maintainers)	
No Internal Boundaries	
Distributed version control	
Short Release Cycle (w/ Merge Window)	
Long Term Support Releases	

Applying ++ Lessons Learned

Linux Best Practice	Zephyr Adoption
Vendor Neutral Decision Making	Yes, Project support from multiple companies.
Companies and Individuals Participate	Yes, TSC has companies & community participation.
Streamline upstreaming process	Yes, see /CONTRIBUTING.rst, DCO used
Public code reviews?	Yes, issues & pull requests reviewed on https://github.com/zephyrproject-rtos/zephyr
Consensus Oriented Decision Models	Yes, TSC votes on features & release readiness.
Hierarchical development (Maintainers)	
No Internal Boundaries	
Distributed version control	
Short Release Cycle (w/ Merge Window)	
Long Term Support Releases	

Applying ++ Lessons Learned

Linux Best Practice	Zephyr Adoption
Vendor Neutral Decision Making	Yes, Project support from multiple companies.
Companies and Individuals Participate	Yes, TSC has companies & community participation.
Streamline upstreaming process	Yes, see /CONTRIBUTING.rst, DCO used
Public code reviews?	Yes, issues & pull requests reviewed on https://github.com/zephyrproject-rtos/zephyr
Consensus Oriented Decision Models	Yes, TSC votes on features & release readiness.
Hierarchical development (Maintainers)	Yes, see /MAINTAINERS.yml
No Internal Boundaries	
Distributed version control	
Short Release Cycle (w/ Merge Window)	
Long Term Support Releases	

Applying ++ Lessons Learned

Zephyr

Linux Best Practice	Zephyr Adoption
Vendor Neutral Decision Making	Yes, Project support from multiple companies.
Companies and Individuals Participate	Yes, TSC has companies & community participation.
Streamline upstreaming process	Yes, see /CONTRIBUTING.rst, DCO used
Public code reviews?	Yes, issues & pull requests reviewed on https://github.com/zephyrproject-rtos/zephyr
Consensus Oriented Decision Models	Yes, TSC votes on features & release readiness.
Hierarchical development (Maintainers)	Yes, see /MAINTAINERS.yml
No Internal Boundaries	Yes, anyone can make pull request for any area
Distributed version control	
Short Release Cycle (w/ Merge Window)	
Long Term Support Releases	

Applying ++ Lessons Learned

Linux Best Practice	Zephyr Adoption
Vendor Neutral Decision Making	Yes, Project support from multiple companies.
Companies and Individuals Participate	Yes, TSC has companies & community participation.
Streamline upstreaming process	Yes, see /CONTRIBUTING.rst, DCO used
Public code reviews?	Yes, issues & pull requests reviewed on https://github.com/zephyrproject-rtos/zephyr
Consensus Oriented Decision Models	Yes, TSC votes on features & release readiness.
Hierarchical development (Maintainers)	Yes, see /MAINTAINERS.yml
No Internal Boundaries	Yes, anyone can make pull request for any area
Distributed version control	Yes, see /CONTRIBUTING.rst
Short Release Cycle (w/ Merge Window)	
Long Term Support Releases	

Zephyr

Release Life Cycle and Maintenance

Periodic Releases

The Zephyr project provides periodic releases every 4 months leading to the long term support releases approximately every 2 years. Periodic and non-LTS releases are maintained with updates, bug fixes and security related updates for at least two cycles, meaning that the project supports the most recent two releases in addition to the most recent LTS.

Long Term Support and Maintenance

A Zephyr Long Term Support (LTS) release is published every 2 years and is branched and maintained independently from the main tree for at least 2.5 years after it was released.

Support and maintenance for an LTS release stops at least half a year after the following LTS release is published.

Source: https://docs.zephyrproject.org/latest/releases/index.html#release-life-cycle-and-maintenance

Applying ++ Lessons Learned

Zephyr

Linux Best Practice	Zephyr Adoption
Vendor Neutral Decision Making	Yes, Project support from multiple companies.
Companies and Individuals Participate	Yes, TSC has companies & community participation.
Streamline upstreaming process	Yes, see /CONTRIBUTING.rst, DCO used
Public code reviews?	Yes, issues & pull requests reviewed on https://github.com/zephyrproject-rtos/zephyr
Consensus Oriented Decision Models	Yes, TSC votes on features & release readiness.
Hierarchical development (Maintainers)	Yes, see /MAINTAINERS.yml
No Internal Boundaries	Yes, anyone can make pull request for any area
Distributed version control	Yes, see /CONTRIBUTING.rst
Short Release Cycle (w/ Merge Window)	Yes, 10 week merge, 2-4 week stabilize
Long Term Support Releases	

Zephyr

Supported Releases

Release	Release date	EOL
Zephyr 2.7.6	2024-03-01	2025-01-26
Zephyr 3.7.0	2024-07-26	2027-01-26
Zephyr 4.0.0	2024-11-15	2025-07-18

As of 2022-01-01, LTS1 (1.14.x) is not supported and has reached end of life (EOL).

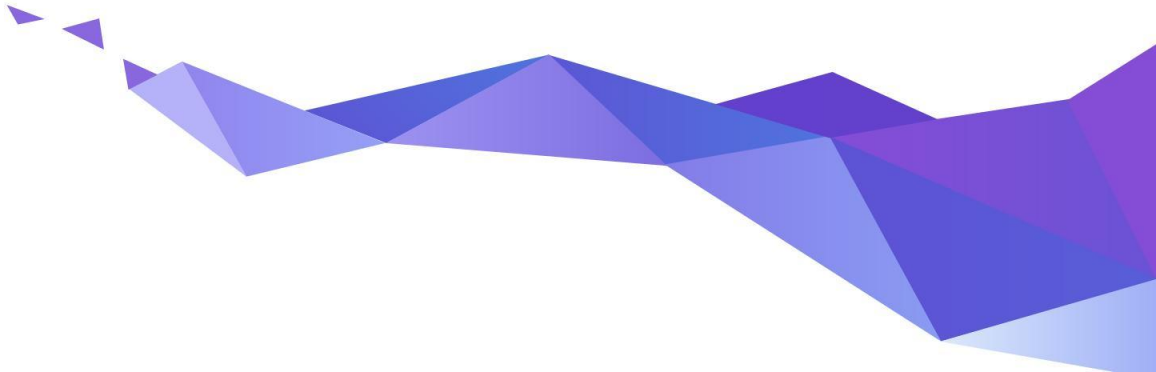
Source: https://docs.zephyrproject.org/latest/releases/index.html#supported-releases

Applying ++ Lessons Learned

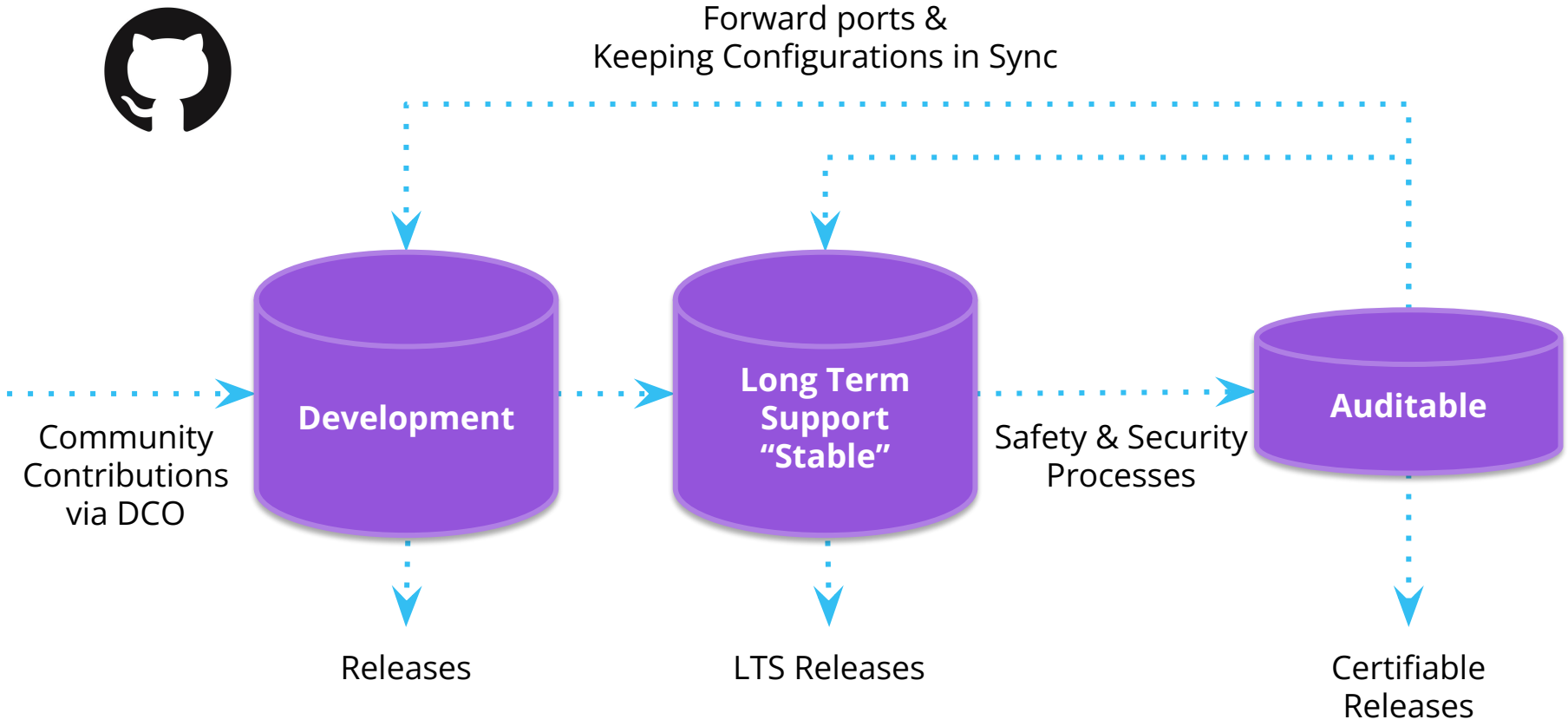
Zephyr

Linux Best Practice	Zephyr Adoption
Vendor Neutral Decision Making	Yes, Project support from multiple companies.
Companies and Individuals Participate	Yes, TSC has companies & community participation.
Streamline upstreaming process	Yes, see /CONTRIBUTING.rst, DCO used
Public code reviews?	Yes, issues & pull requests reviewed on https://github.com/zephyrproject-rtos/zephyr
Consensus Oriented Decision Models	Yes, TSC votes on features & release readiness.
Hierarchical development (Maintainers)	Yes, see /MAINTAINERS.yml
No Internal Boundaries	Yes, anyone can make pull request for any area
Distributed version control	Yes, see /CONTRIBUTING.rst
Short Release Cycle (w/ Merge Window)	Yes, 10 week merge, 2-4 week stabilize
Long Term Support Releases	Yes, LTS 2 had 6 update release, LTS 3 active maintain

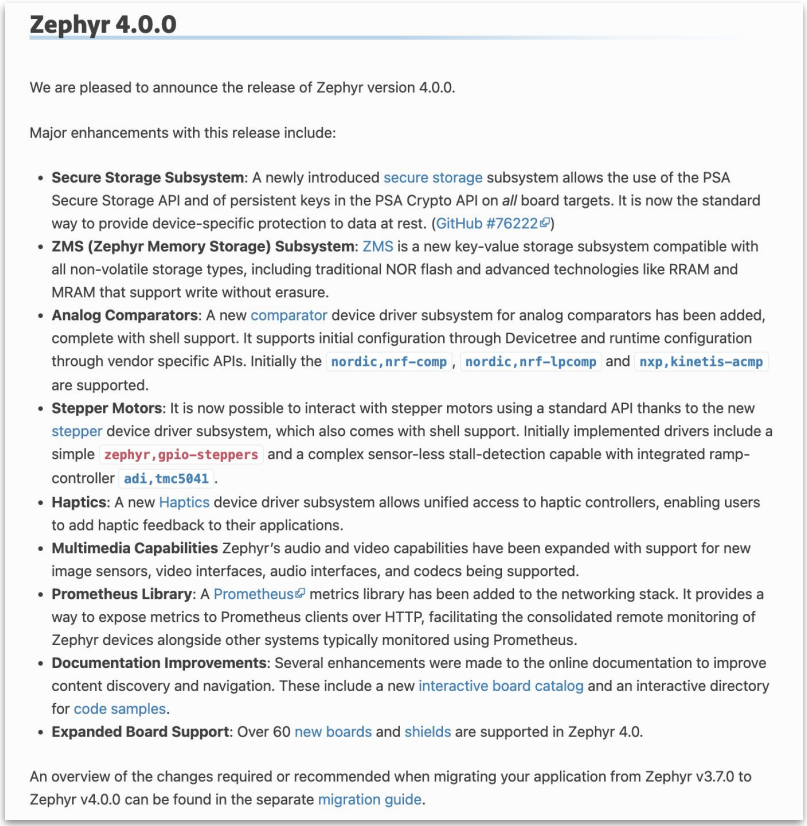
What about Zephyr security best practices?



Code Repositories



Zephyr 4.0 (November 2024)



Zephyr 4.0.0

We are pleased to announce the release of Zephyr version 4.0.0.

Major enhancements with this release include:

- **Secure Storage Subsystem**: A newly introduced secure storage subsystem allows the use of the PSA Secure Storage API and of persistent keys in the PSA Crypto API on *all* board targets. It is now the standard way to provide device-specific protection to data at rest. (GitHub #76222)
- **ZMS (Zephyr Memory Storage) Subsystem**: ZMS is a new key-value storage subsystem compatible with all non-volatile storage types, including traditional NOR flash and advanced technologies like RRAM and MRAM that support write without erasure.
- **Analog Comparators**: A new comparator device driver subsystem for analog comparators has been added, complete with shell support. It supports initial configuration through Devicetree and runtime configuration through vendor specific APIs. Initially the `nordic,nrf-comp`, `nordic,nrf-lpcomp` and `nxp,kinetis-acmp` are supported.
- **Stepper Motors**: It is now possible to interact with stepper motors using a standard API thanks to the new stepper device driver subsystem, which also comes with shell support. Initially implemented drivers include a simple `zephyr,gpio-steppers` and a complex sensor-less stall-detection capable with integrated ramp-controller `adi,tmc5041`.
- **Haptics**: A new Haptics device driver subsystem allows unified access to haptic controllers, enabling users to add haptic feedback to their applications.
- **Multimedia Capabilities** Zephyr's audio and video capabilities have been expanded with support for new image sensors, video interfaces, audio interfaces, and codecs being supported.
- **Prometheus Library**: A Prometheus metrics library has been added to the networking stack. It provides a way to expose metrics to Prometheus clients over HTTP, facilitating the consolidated remote monitoring of Zephyr devices alongside other systems typically monitored using Prometheus.
- **Documentation Improvements**: Several enhancements were made to the online documentation to improve content discovery and navigation. These include a new interactive board catalog and an interactive directory for code samples.
- **Expanded Board Support**: Over 60 new boards and shields are supported in Zephyr 4.0.

An overview of the changes required or recommended when migrating your application from Zephyr v3.7.0 to Zephyr v4.0.0 can be found in the separate migration guide.

To Learn More:

Zephyr 4.0 &

Release notes 4.0

→ **Next release**: Zephyr 4.1

Zephyr 3.7 LTS (July 2024)

🛰️ **New Hardware Model**

🔐 Integration of **TF-M PSA Crypto API**

⏱️ Support for **Precision Time Protocol** (PTP)

📄 **SBOM generation** supports **SPDX 2.3** & **PURL/CPE**

To Learn More: 3.7 Blog Post & Release notes 3.7

LTS Support Windows

Supported Releases

Release	Release date	EOL
Zephyr 2.7.6	2024-03-01	2025-01-26
Zephyr 3.7.0	2024-07-26	2027-01-26
Zephyr 4.0.0	2024-11-15	2025-07-18

As of 2022-01-01, LTS1 (1.14.x) is not supported and has reached end of life (EOL).

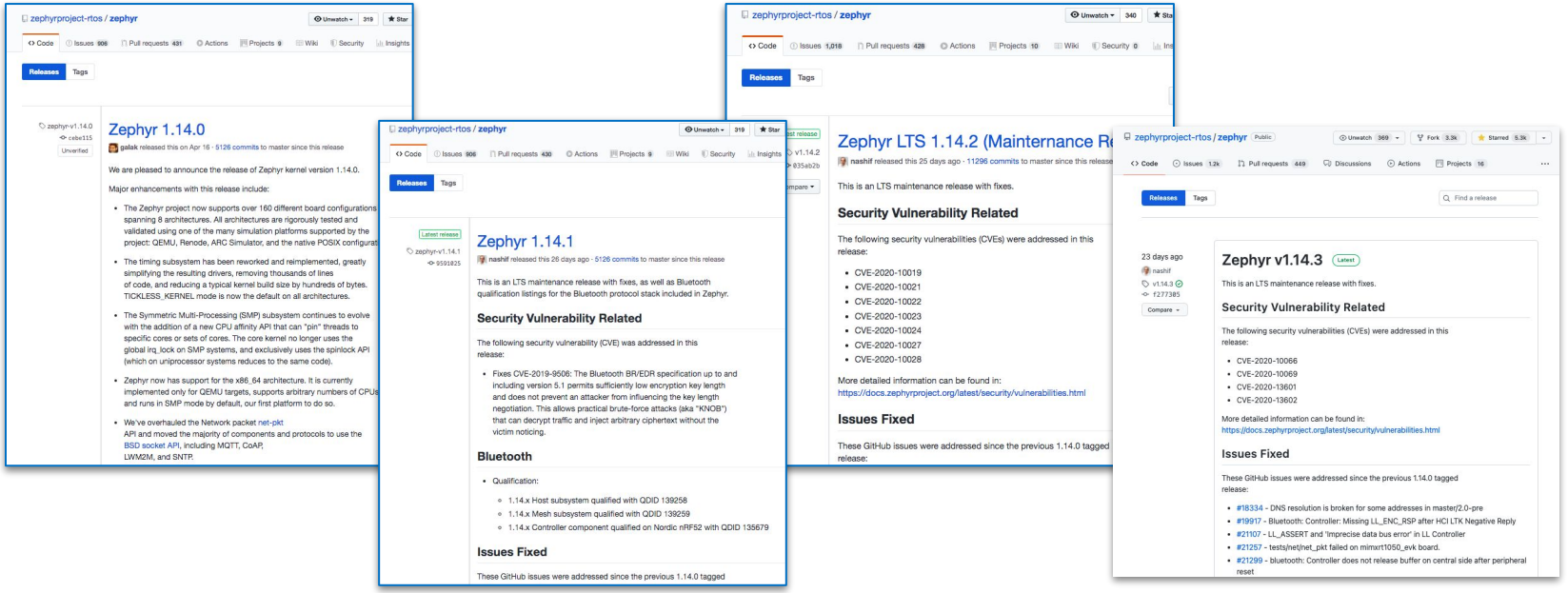
Source: https://docs.zephyrproject.org/latest/releases/index.html#supported-releases

Long Term Support (Zephyr 3.7.x)

- **Product Focused**
- Current with latest **Security Updates**
- Compatible with new hardware
  - Functional support for new hardware is regularly backported
- **Tested**: Shorten the development window and extend the Beta cycle to allow for more testing and bug fixing
- **Supported for 2+ years**
- ⚠️ **Doesn't include cutting-edge functionality**

https://docs.zephyrproject.org/3.7.0/

Long Term Support



Delivered bug fixes and latest security updates for 2 years!

Security Focus From the Start

Exhibit B

Zephyr Project Charter (the "Charter")
The Linux Foundation
Updated August 21, 2023

1. Mission of the Zephyr Project ("Zephyr," or, alternatively, the "Project").

The mission of the Project is to:

a. deliver the best-in-class RTOS for connected resource-constrained devices, built to be secure and safe.

b. maintain an auditable code base, while taking advantage of community participation; this auditable code base is open source;

c. include participation of leading members of this ecosystem, including micro-controller manufacturers, hardware developers, software developers and other members of the ecosystem; and

d. host the infrastructure for the open source Project and sub-projects, establishing a neutral home for community meetings, events and collaborative discussions and providing structure around the business and technical governance of the Project.

Security Focus From the Start

Exhibit B

Zephyr Project Charter (the "Charter")
The Linux Foundation

1. Mission of the Zephyr

The mission of the Project is

a. deliver the best-i
to be secure and

b. maintain an audi
participation; th

c. include participat
controller manuf
members of the e

d. host the infrastruc
neutral home for
providing structu

6. Security Committee

a. Composition – the Security Committee members shall consist of:

i. one appointed voting representative from each Platinum Member, plus

ii. non-voting Silver Member representatives who shall not count towards quorum.

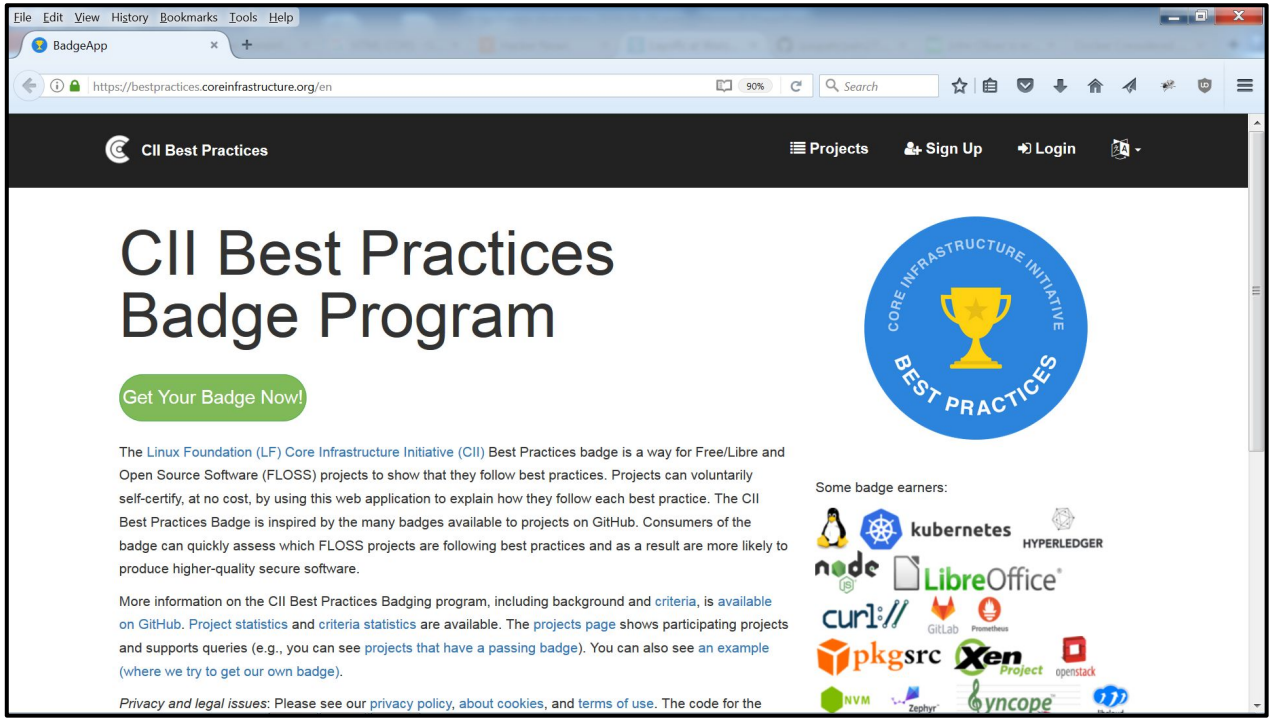
b. Responsibilities – the Security Committee shall be responsible for:

i. the definition of the processes to ensure an auditable code base, as well as any associated certification artifacts ("Security Artifacts");

ii. annually elect a Representative on the Security Committee to serve as chair of the Security Committee; and

iii. annually elect a security architect (the "Security Architect"), who may be different from the chair of the Security Committee.

Starting Point: Adopt Known Best Practices



https://bestpractices.coreinfrastructure.org

Best Practices Badge

Identified best practices for OSS projects

- For *production* of OSS
- Based on practices of well-run OSS projects
- Increase likelihood of better quality & security
- Criteria designed for *any* OSS project

Web application: OSS projects self-certify

- If OSS project meets criteria, it gets a badge
- No cost
- Self-certification mitigated by automation, public display of answers (for criticism), spot-checks, and can be overridden if false

⇒ moved under Open SSF in 2021

Source: https://www.bestpractices.dev

Criteria

Three badge levels (passing, silver, gold)

- Any level is an achievement
- For higher levels, must meet previous level
- Based on real projects
  - Not “people should do X, but no one does that”
- Gold requires multiple developers
  - bus factor > 1*, 2-person review

More info at: https://github.com/coreinfrastructure/best-practices-badge

* A “bus factor” is how many people would have to be hit by a bus before a project stalls (e.g., due to lack) knowledge)

Statistics about Criteria & Levels

Criteria Statistics

Level	Total active	MUST	SHOULD	SUGGESTED	Allow N/A	Met justification required	Require URL	Met justification or URL required	Includes details	New at this level	Future
Passing	67	43	10	14	27	1	8	9	52	67	0
Silver	55	44	10	1	41	38	17	54	39	48	0
Gold	23	21	2	0	9	13	9	22	16	14	0

The "active" criteria are criteria that are included in the percentage calculations (as opposed to "future" criteria). The next columns identify the number of active criteria in each level that are MUST, SHOULD, SUGGESTED, allow a "N/A" as an answer, require justification when "met" is the answer, require a URL, require justification when "met" is the answer or a URL, include details, or are new at this level. "Future" criteria are shown on the form, and are expected to be added as active criteria to some level in the future, but are not included in completion calculations.

You can see statistics about projects over time at the project stats page.

You may also see the actual criteria.

- There are not a lot of gold criteria, but they are challenging.
- Source: https://www.bestpractices.dev/en/criteria

Zephyr's Path - Initial Passing Badge

Zephyr Launched 2016/2

- Initial security team was composed of device security experts or either open source embedded experts from our members, but limited knowledge domain overlap and understanding of issues in either space.

CII badge program launched 2016/5

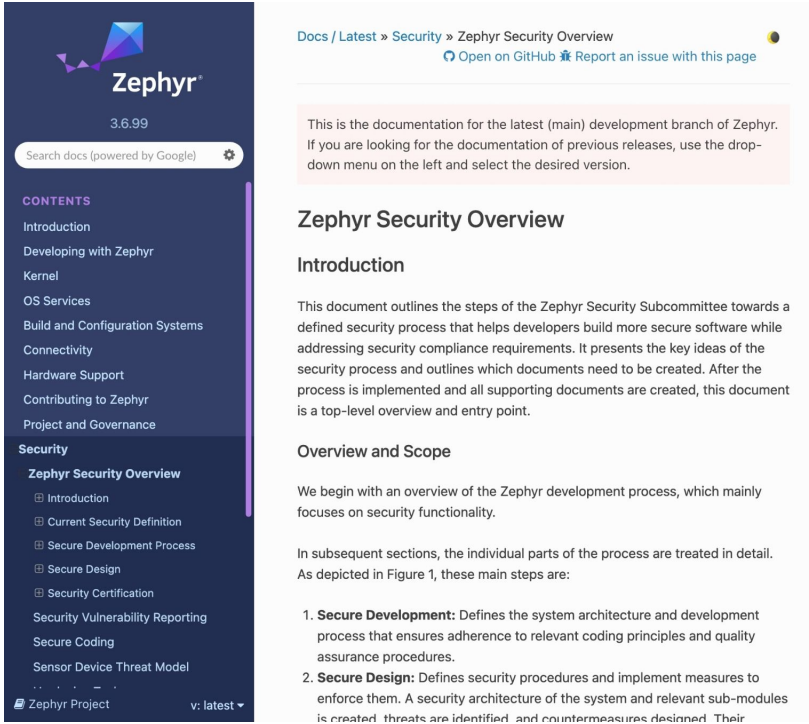
- Looked through the criteria and decided to aim for passing badge.
- 75% was straight forward to fill out and was done within first week.
- Security and Analysis sections served as a focus to start organizing knowledge from diverse participants in the security team.

Zephyr achieved “Passing” badge 2016/11

- Some criteria we met fairly easily, other criteria caused significant discussion, and took a while to create the documentation (which we needed to do!)

Project Security Documentation

- **Project Security Overview**
- Started with documents from other projects
- Built around Secure Development, Secure Design, and Security Certification
- Ongoing process, rather than something to just be accomplished



Zephyr's Path - Oops... Passing Regained

Zephyr stopped “Passing” 2017/2

- Zephyr project infrastructure underwent significant transition in 2017 (JIRA →Issues, Gerrit → github)
- Prior data was inaccurate, and we had forgotten to update it.
- Badge app notified us we were not longer “passing”

Zephyr regains passing 2017/8

- After all transitions done, updated documentation to reflect the infrastructure and we were passing again.
- **Decided to try for Silver** – but there were some big lifts for the project: key roles and responsibilities documented, longer roadmap than we’d been keeping, TLS certificate verification

Zephyr's Path - Become a CNA?

A CNA allows Zephyr Project to manage vulnerabilities, assign them CVE IDs, and handle the disclosure of information pertaining to those vulnerabilities.

- Zephyr Project CNA determines the validity of issues/vulnerabilities,
- whether or not they will be publicly disclosed,
- the amount of information that will be disclosed,
- the timing for that disclosure.

Changes made by the Zephyr Project to become a CNA:

- Zephyr Project security **documentation was be reviewed and modified** to handle the new requirements levied by the CNA process.
- **New email lists** were created to be used as points of contact for external entities (provided to MITRE to be used for contact and also will be added to Zephyr Project websites.
    - vulnerabilities@zephyrproject.org (used as primary contact for external entities)
    - zephyr-psirt-request@lists.zephyrproject.org (internal project list for CNA communications)

Zephyr's Path - Become a CNA? Yes!

Four things required* for getting a CNA in place:

1. Definition of scope:
   All Zephyr project components and vulnerabilities discovered by Zephyr project participants that are not covered by another CNA.
2. Public point of contact:
   vulnerabilities@zephyrproject.org was listed on websites (both Zephyr project and MITRE).
3. Direct point of contact for backdoor communications from MITRE:
   zephyr-psirt-request@lists.zephyrproject.org
4. A list of email addresses that will be added to the MITRE announcement:
   zephyr-psirt-request@lists.zephyrproject.org

Sent email with above in August 2017, and MITRE announced Zephyr as CNA

*per phone discussion with MITRE, July 2017

Zephyr Listed as CNA in NVD in 2017

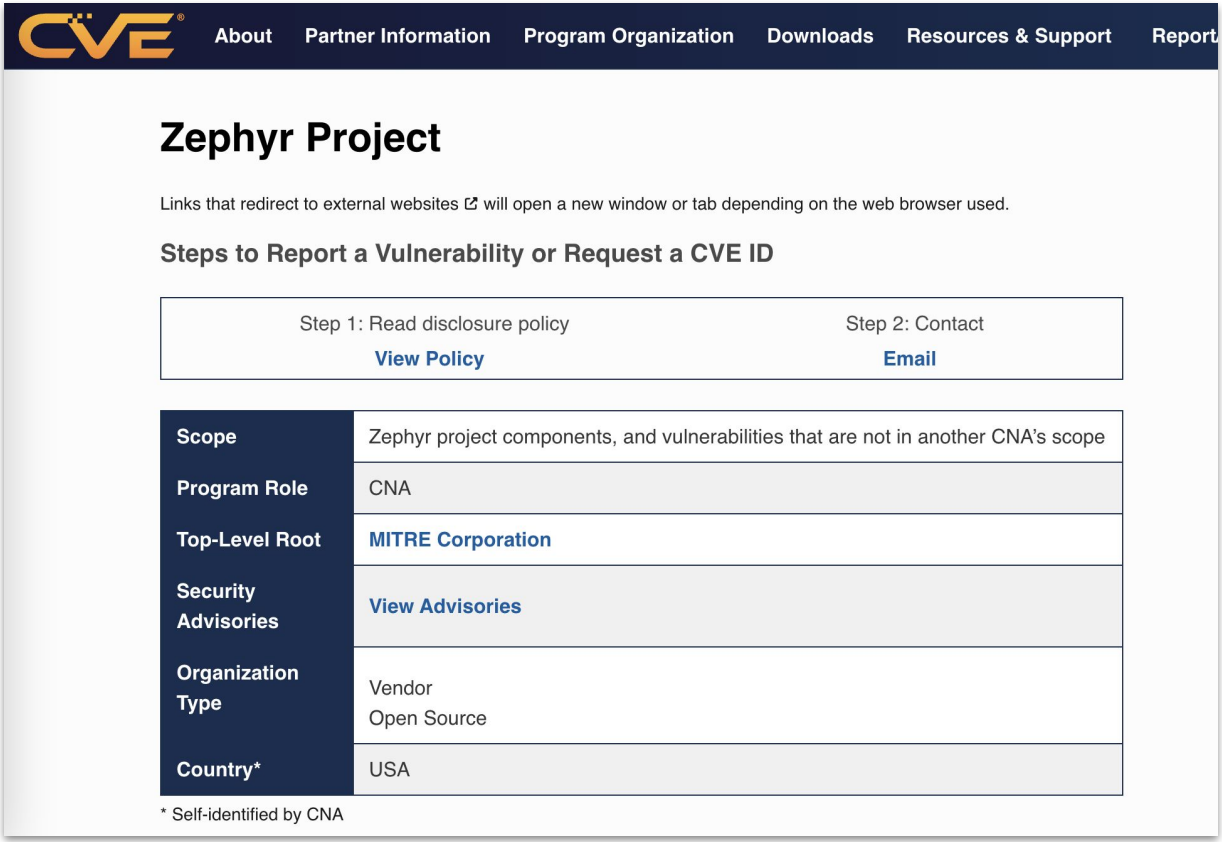
Product, Vendor, or Product Category Name	Scope	CNA Contact Email and/or Webpage (if applicable)	CNA Type*
MITRE Corporation	All vulnerabilities not already covered by a CNA listed on this page	MITRE CVE Request web form	Primary CNA
Zephyr Project	Zephyr project components and vulnerabilities that are not covered by another CNA	vulnerabilities@zephyrproject.org	Vendors and Projects
Zero Day Initiative	Products and projects covered by its bug bounty programs not already covered by another CNA	zdi-disclosures@trendmicro.com ZDI contact page	Bug Bounty Programs
ZTE Corporation	ZTE products only	psirt@zte.com.cn	Vendors and Projects

*** Key for CNA Types:**

Bug Bounty Programs - assigns CVE IDs to products and projects that utilize the Bug Bounty service's product offerings.
National and Industry CERTs - performs incident response and vulnerability disclosure services for nations or industries. They may assign CVE IDs as part of their role and scope.
Primary CNA - oversees the CNA program.
Root CNA - manages a group of sub-CNAs within a given domain or community.
Vendors and Projects - assigns CVE IDs for vulnerabilities found in their own products and projects.
Vulnerability Researchers - assigns CVE IDs to products and projects upon which they perform vulnerability analysis.

* https://cve.mitre.org/cve/request_id.html#cna_participants

Zephyr CNA Entry Today



CVE® About Partner Information Program Organization Downloads Resources & Support Report

Zephyr Project

Links that redirect to external websites will open a new window or tab depending on the web browser used.

Steps to Report a Vulnerability or Request a CVE ID

Step 1: Read disclosure policy	Step 2: Contact
View Policy	Email

Scope	Zephyr project components, and vulnerabilities that are not in another CNA's scope
Program Role	CNA
Top-Level Root	MITRE Corporation
Security Advisories	View Advisories
Organization Type	Vendor Open Source
Country*	USA

* Self-identified by CNA

Source: https://www.cve.org/PartnerInformation/ListofPartners/partner/zephyr

Zephyr PSIRT Today

Project Security Incident Response Team

- Led by Zephyr Security Architect (elected annually from peers)
- Volunteers from Security Committee (Zephyr Project Members) do initial triage
- Manage embargo windows and interaction with maintainers for fixes into upstream and then backports to LTS
- Responsible for satisfying evolving CVE Program & CNA Process Requirements.

Zephyr's Badge Path Continues...

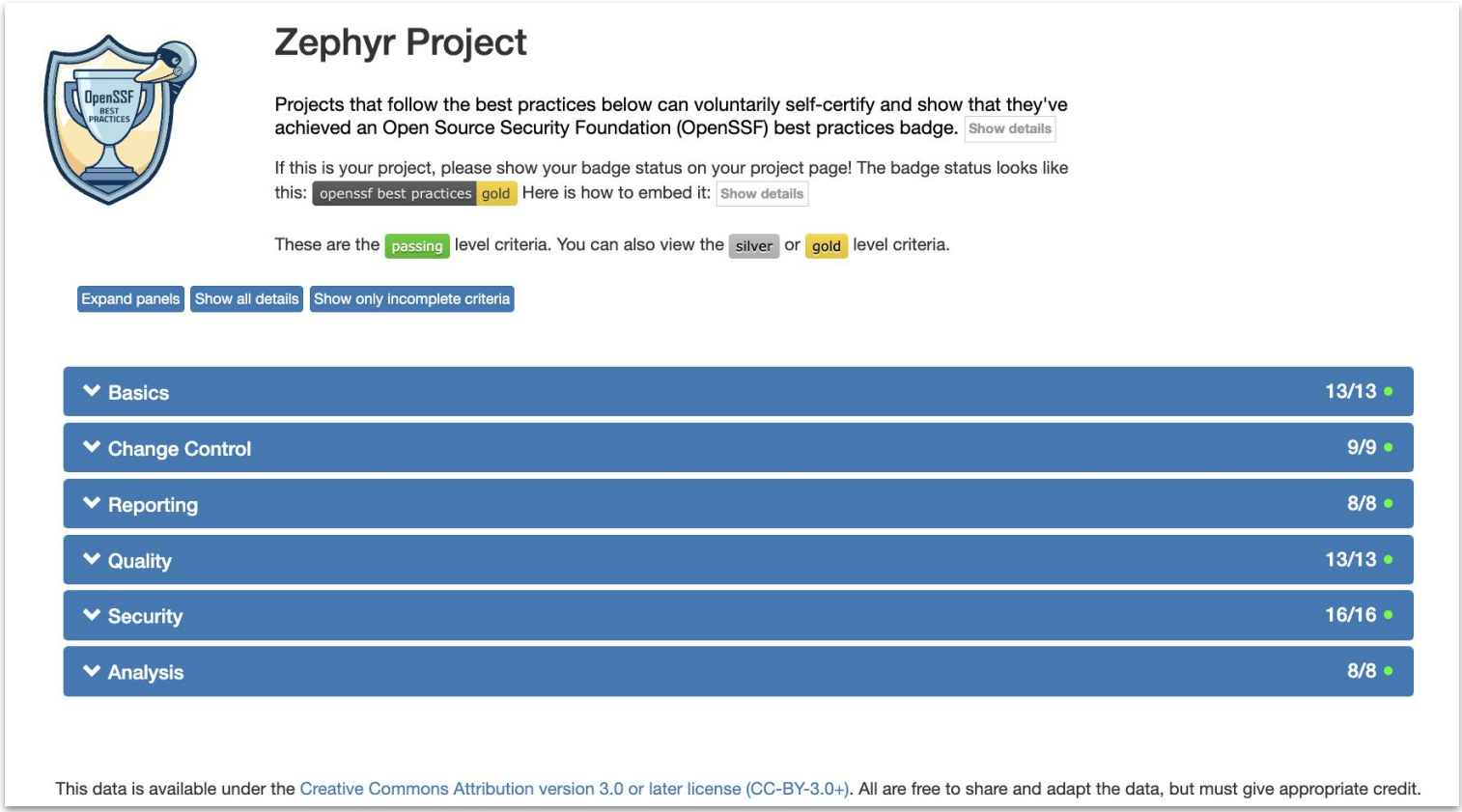
Zephyr almost at “Silver” 2018/4

- Zephyr addressed all issues except “TLS certificate verification”, we had a TLS library, but Zephyr is an OS, not an App.
- Threat model and justification documents that security requirements are met had to be created, again issue not an App.

Zephyr gets Silver 2018/9

- After implementing a separate application as a sample for TLS issue

Zephyr's Gold Badge - Feb 2019!



Zephyr Project

Projects that follow the best practices below can voluntarily self-certify and show that they've achieved an Open Source Security Foundation (OpenSSF) best practices badge. Show details

If this is your project, please show your badge status on your project page! The badge status looks like this: openssf best practices gold Here is how to embed it: Show details

These are the passing level criteria. You can also view the silver or gold level criteria.

Expand panels | Show all details | Show only incomplete criteria

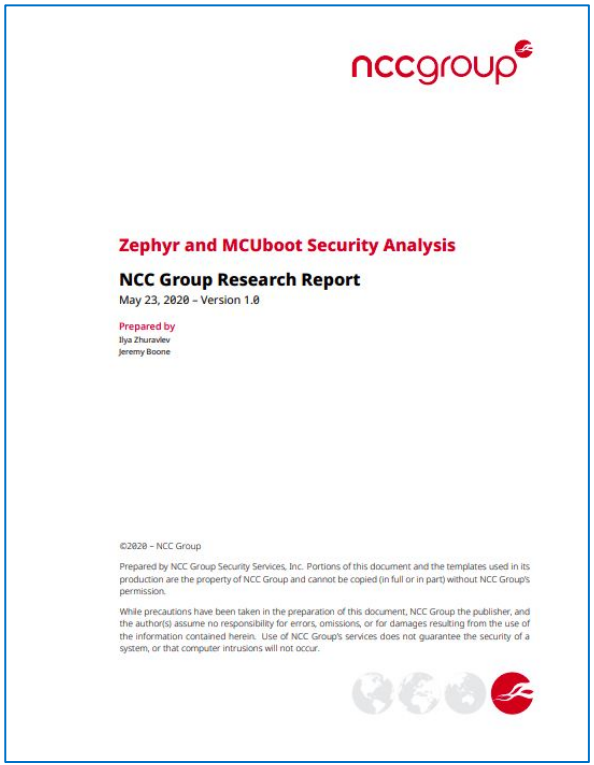
- Basics 13/13
- Change Control 9/9
- Reporting 8/8
- Quality 13/13
- Security 16/16
- Analysis 8/8

This data is available under the Creative Commons Attribution version 3.0 or later license (CC-BY-3.0+). All are free to share and adapt the data, but must give appropriate credit.

Source: https://www.bestpractices.dev/en/projects/74

First Bulk Security Report (2019)

- NCC Group reported ~26 issues
- Critical, High and Medium made into JIRA tickets (we used JIRA before transitioning private github we use today)
- All were addressed
- After embargo, everything updated in the vulnerability report page
- Most resulted in 1 or more CVEs being reported


nccgroup

Zephyr and MCUboot Security Analysis

NCC Group Research Report

May 23, 2020 – Version 1.0

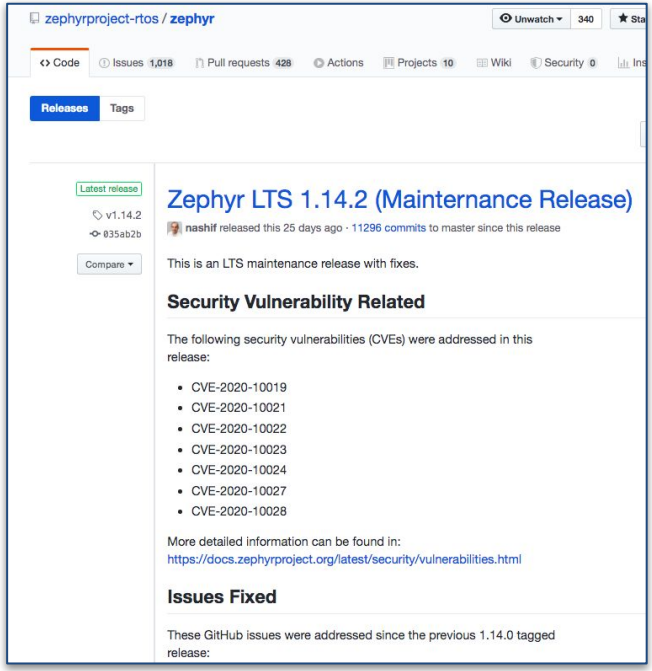
Prepared by

Ilya Zhuravlev

Jeremy Boone

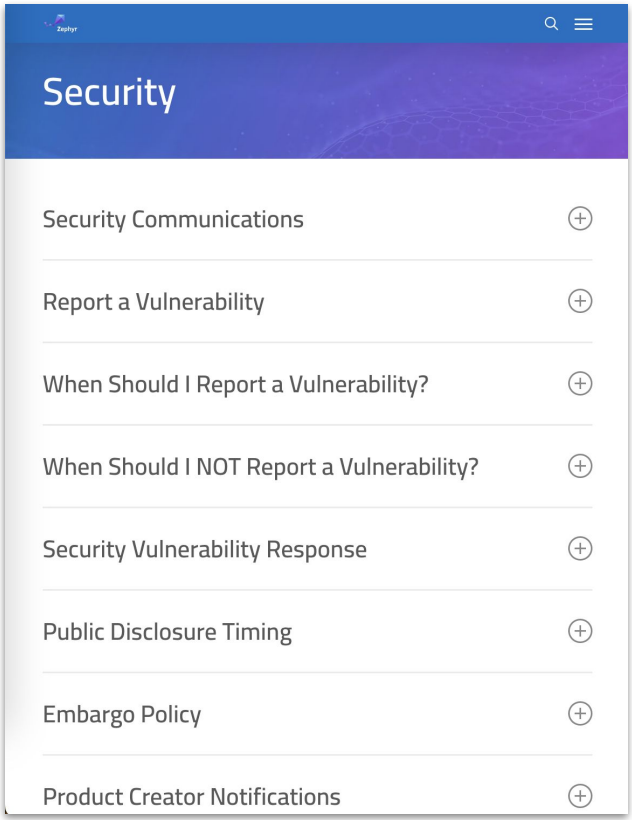
Results from the 2019 NCC Report

- Most issues were fixed in reasonable time and included in releases
- One issue, recommendation is to disable
- Increased embargo from 60 to 90 days
  - Zephyr isn't an end product, vendors need time to incorporate fixes into products
  - Zephyr needs alert system to notify vendors
- Continue to improve processes



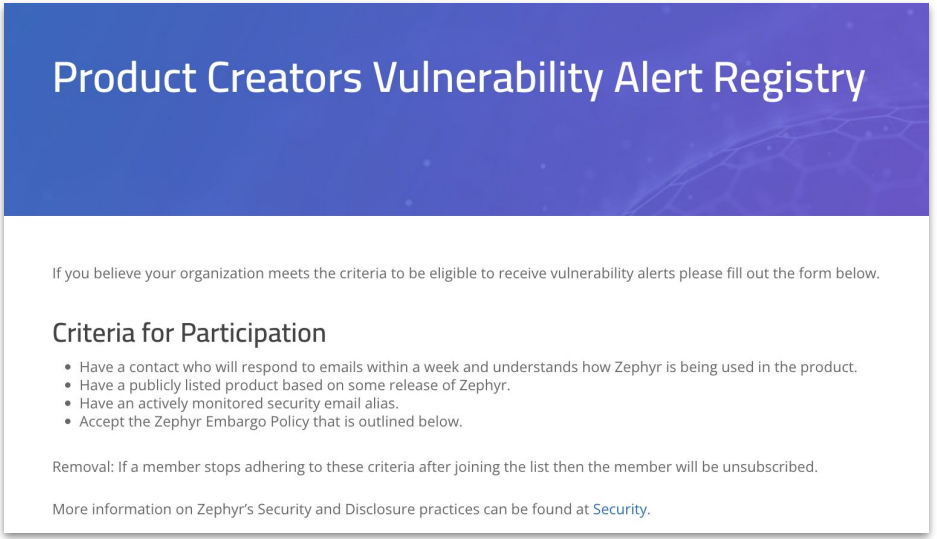
Improving Processes...

- Highlighted need to better document process
- Added vulnerability reporting to project docs
- Added security section to main project page
- Process:
  - Embargo period extended
  - Stages issue goes through
  - Working with maintainers to see issues fixed
  - Public disclosure at end



Better Support for Product Makers

- For an embargo to work, product makers need to be notified early so they can remediate.
- Created Vulnerability Registry for **vendors to register** to receive these alerts for **free**
- **Goal:** Zephyr to **fix issues within 30 days** to give **vendors 60 days** before publication of vulnerability



Source: https://www.zephyrproject.org/vulnerability-registry/

What we had to do before VEX...

Advisory Issued by project on 20201208:

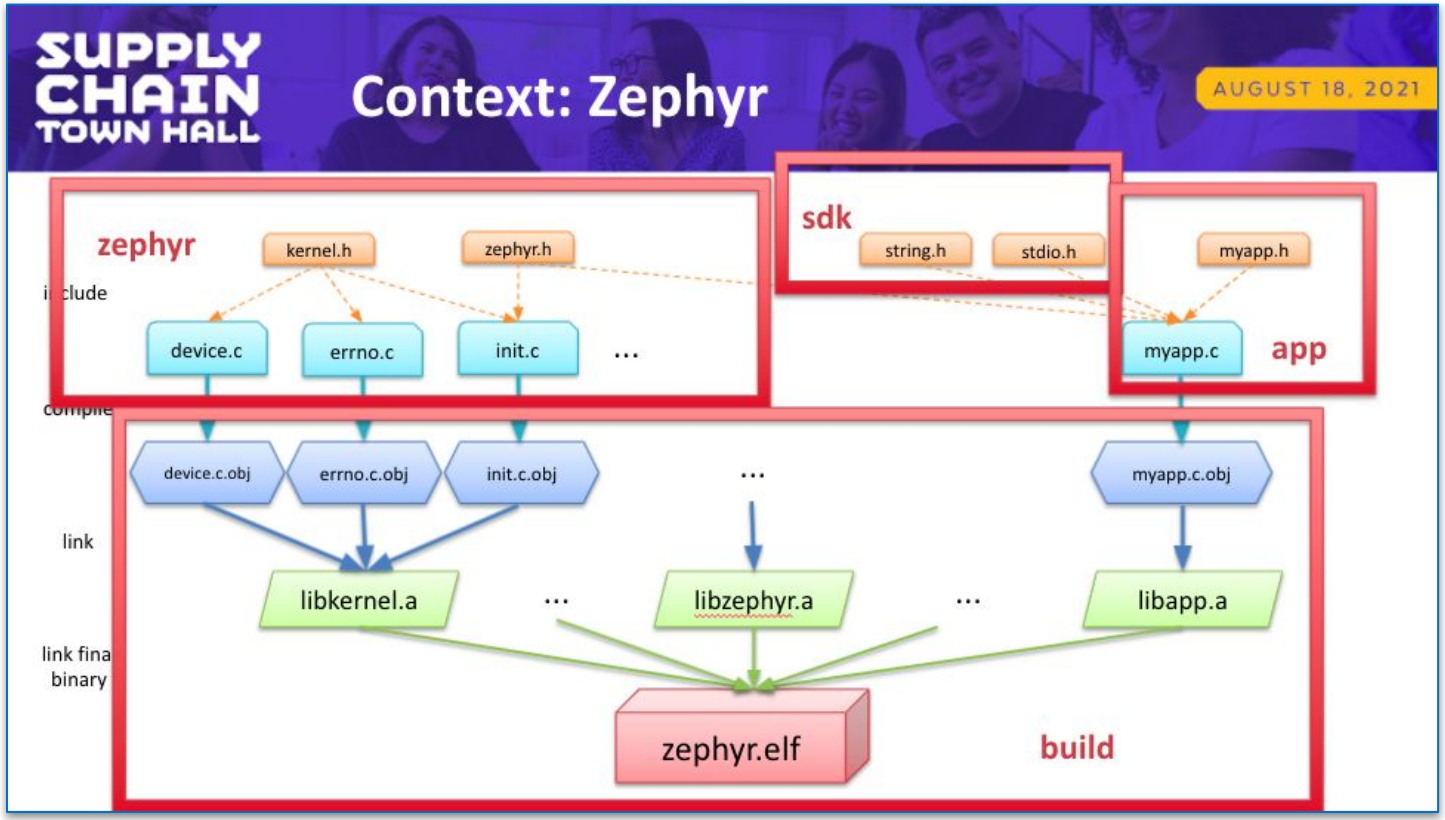
Zephyr current release (2.4) does **not use** Fnet or other stacks.

The Zephyr LTS release 1.14 contains an implementation of the TCP stack from Fnet.

- Of the vulnerabilities reported in Fnet, 2, CVE-2020-17468, and CVE-2020-17469, are in the IPv6 Fnet code, one, CVE-2020-17467, affects Link-local Multicast Name Resolution LLMNR), and 2, CVE-2020-24383, and CVE-2020-17470 affect DNS functionality.
- **None of the affected code has been used** in the Zephyr project, while 1.14 does use the Fnet TCP, it does not use the affected IPv6, DNS or LLMNR code.

https://www.zephyrproject.org/zephyr-security-update-on-amnesia33/

SBOM generation added in 2021



Learn more at: https://www.youtube.com/watch?v=KYC3YpSu9zs

Automated SBOM Generation During Build!

1. Create a build directory with CMake file API enabled
2. Build project with “build metadata” enabled
3. Compute SBOM(s)

```
west spdx --init -d BUILD_DIR
west build -d BUILD_DIR -- -DCONFIG_BUILD_OUTPUT_META=y
west spdx -d BUILD_DIR
```

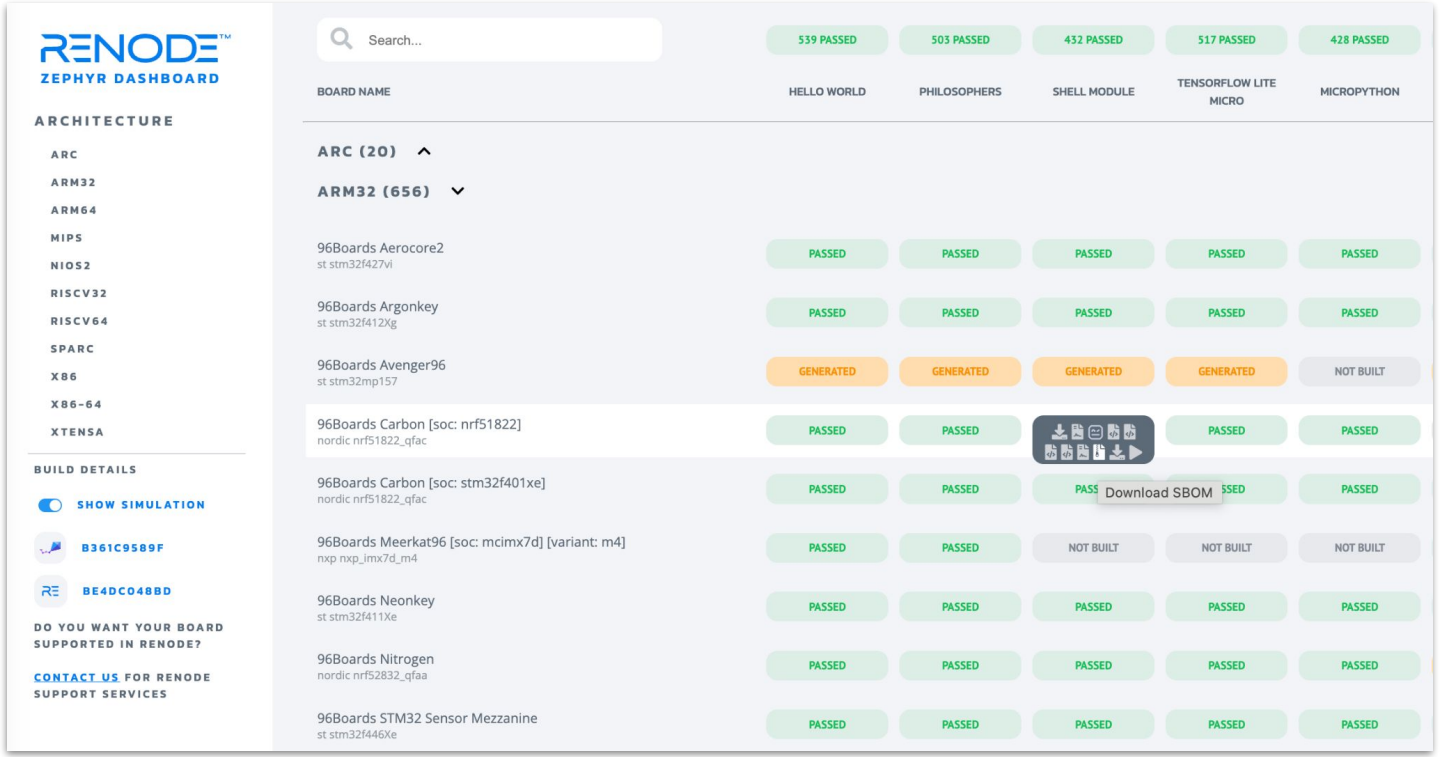
`zephyr.spdx`	SBOM for the **Zephyr source files** actually used by your application
`app.spdx`	SBOM for the source files of your **application**
`build.spdx`	SBOM for **all the build objects**, inc. of course your final image

SBOM's at Scale...Automatically

875 boards

13 apps

All BUILT, PASSED, GENERATED have **3 SBOM**s available to download & inspect



Source: https://zephyr-dashboard.renode.io/

Dashboard SBOM

blinky-app.spdx

```
SPDXVersion: SPDX-2.3
DataLicense: CC0-1.0
SPDXID: SPDXRef-DOCUMENT
DocumentName: app-sources
DocumentNamespace: http://spdx.org/spdxdocs/zephyr-ab992a5d-47b4-44ee-8357-1b68719b389b/app
Creator: Tool: Zephyr SPDX builder
Created: 2024-06-07T01:12:51Z

Relationship: SPDXRef-DOCUMENT DESCRIBES SPDXRef-app-sources

##### Package: app-sources

PackageName: app-sources
SPDXID: SPDXRef-app-sources
PackageDownloadLocation: NOASSERTION
PackageLicenseConcluded: Apache-2.0
PackageLicenseDeclared: NOASSERTION
PackageCopyrightText: NOASSERTION
PrimaryPackagePurpose: SOURCE
PackageLicenseInfoFromFiles: Apache-2.0
FilesAnalyzed: true
PackageVerificationCode: a5993032fe245294fb73f4ed2f53be33566662f6

FileName: ./src/main.c
SPDXID: SPDXRef-File-main.c
FileChecksum: SHA1: d71a9d7b80f5eac4b749b84c57297614ef8e3899
FileChecksum: SHA256: cdc42b14891c38dfc131eb3dea80906698289496a18c7e76e9945f2e3dd17152
LicenseConcluded: Apache-2.0
LicenseInfoInFile: Apache-2.0
FileCopyrightText: NOASSERTION
```

blinky-zephyr.spdx

```
SPDXVersion: SPDX-2.3
DataLicense: CC0-1.0
SPDXID: SPDXRef-DOCUMENT
DocumentName: zephyr-sources
DocumentNamespace: http://spdx.org/spdxdocs/zephyr-ab992a5d-47b4-44ee-8357-1b68719b389b/zephyr
Creator: Tool: Zephyr SPDX builder
Created: 2024-06-07T01:12:51Z

Relationship: SPDXRef-DOCUMENT DESCRIBES SPDXRef-zephyr-sources

##### Package: zephyr-sources

PackageName: zephyr-sources
SPDXID: SPDXRef-zephyr-sources
PackageDownloadLocation: NOASSERTION
PackageLicenseConcluded: Apache-2.0
PackageLicenseDeclared: NOASSERTION
PackageCopyrightText: NOASSERTION
PackageLicenseInfoFromFiles: Apache-2.0
FilesAnalyzed: true
PackageVerificationCode: f10da9dec03dd29bb556c72963bf33ae9f840643

FileName: ./zephyr/arch/arm/core/cortex_m/__aeabi_read_tp.S
SPDXID: SPDXRef-File---aeabi-read-tp.S
FileChecksum: SHA1: 62d0921844d538be8c28eae5bc4c0b9f87692bd3
FileChecksum: SHA256: 1ba5712dbc2a5d48a57fde5070b2cdc0f6b2bb86a740ae2f55811aaf1bea0aa1
LicenseConcluded: Apache-2.0
LicenseInfoInFile: Apache-2.0
FileCopyrightText: NOASSERTION
```

blinky-build.spdx

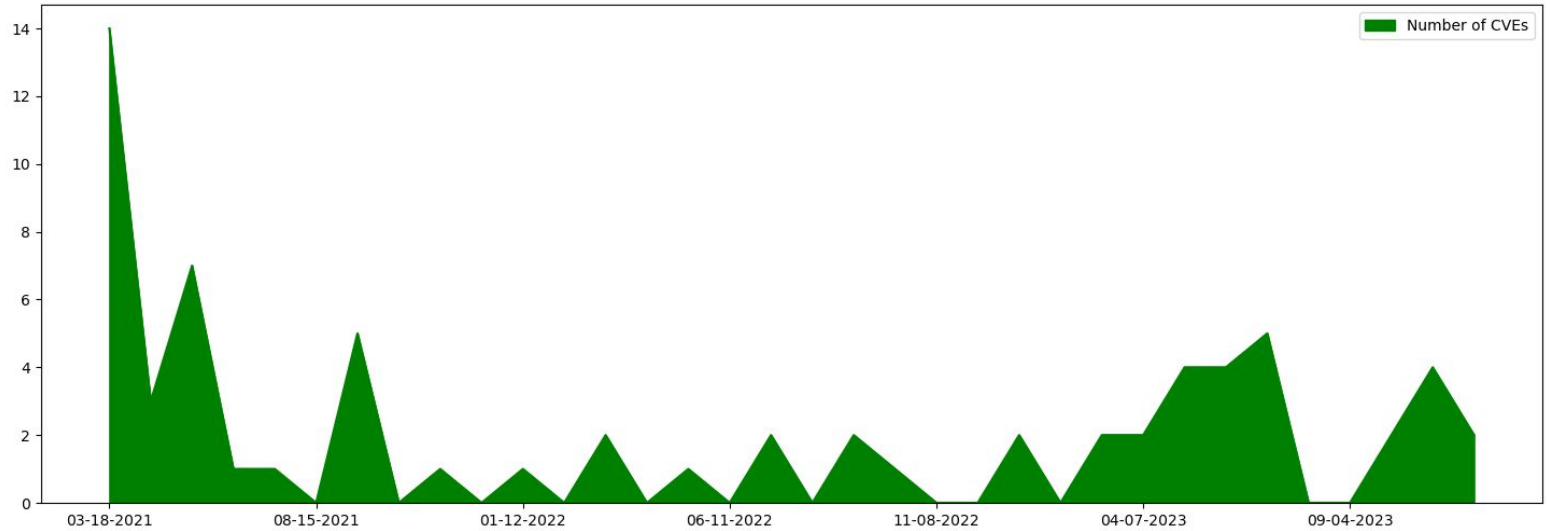
```
SPDXVersion: SPDX-2.3
DataLicense: CC0-1.0
SPDXID: SPDXRef-DOCUMENT
DocumentName: build
DocumentNamespace: http://spdx.org/spdxdocs/zephyr-ab992a5d-47b4-44ee-8357-1b68719b389b/build
Creator: Tool: Zephyr SPDX builder
Created: 2024-06-07T01:12:51Z

ExternalDocumentRef: DocumentRef-app http://spdx.org/spdxdocs/zephyr-
ab992a5d-47b4-44ee-8357-1b68719b389b/app SHA1: 594de9d45188c55bdb059a2b0045987bb87e79be
ExternalDocumentRef: DocumentRef-zephyr http://spdx.org/spdxdocs/zephyr-
ab992a5d-47b4-44ee-8357-1b68719b389b/zephyr SHA1: 4ae97af97a0e9fbc050f72ea71ad3bf2f9caffa7

Relationship: SPDXRef-DOCUMENT DESCRIBES SPDXRef-zephyr-final

...

FileName: ./zephyr/arch/arch/arm/core/cortex_m/libarch__arm__core__cortex_m.a
SPDXID: SPDXRef-File-libarch--arm--core--cortex-m.a
FileChecksum: SHA1: 310c7abd765821c8e8df8ceb1ac8bae330f371b1
FileChecksum: SHA256: 5efe0a524dd3a48e7cf6d637966a4f6fffa60119f4ab2b2b2f3ec4d924f5ea2a
LicenseConcluded: NOASSERTION
LicenseInfoInFile: NONE
FileCopyrightText: NOASSERTION

Relationship: SPDXRef-File-libarch--arm--core--cortex-m.a GENERATED_FROM DocumentRef-zephyr:SPDXRef-File-exc-exit.c
Relationship: SPDXRef-File-libarch--arm--core--cortex-m.a GENERATED_FROM DocumentRef-zephyr:SPDXRef-File-fault.c
Relationship: SPDXRef-File-libarch--arm--core--cortex-m.a GENERATED_FROM DocumentRef-zephyr:SPDXRef-File-fault-s.S
Relationship: SPDXRef-File-libarch--arm--core--cortex-m.a GENERATED_FROM DocumentRef-zephyr:SPDXRef-File-fpu.c
Relationship: SPDXRef-File-libarch--arm--core--cortex-m.a GENERATED_FROM DocumentRef-zephyr:SPDXRef-File-reset.S
Relationship: SPDXRef-File-libarch--arm--core--cortex-m.a GENERATED_FROM DocumentRef-zephyr:SPDXRef-File-scb.c
Relationship: SPDXRef-File-libarch--arm--core--cortex-m.a GENERATED_FROM DocumentRef-zephyr:SPDXRef-File-thread-abort.c
Relationship: SPDXRef-File-libarch--arm--core--cortex-m.a GENERATED_FROM DocumentRef-zephyr:SPDXRef-File-vector-table.S
Relationship: SPDXRef-File-libarch--arm--core--cortex-m.a GENERATED_FROM DocumentRef-zephyr:SPDXRef-File-swap.c
Relationship: SPDXRef-File-libarch--arm--core--cortex-m.a GENERATED_FROM DocumentRef-zephyr:SPDXRef-File-swap-helper.S
Relationship: SPDXRef-File-libarch--arm--core--cortex-m.a GENERATED_FROM DocumentRef-zephyr:SPDXRef-File-irq-manage.c
Relationship: SPDXRef-File-libarch--arm--core--cortex-m.a GENERATED_FROM DocumentRef-zephyr:SPDXRef-File-prep-c.c
Relationship: SPDXRef-File-libarch--arm--core--cortex-m.a GENERATED_FROM DocumentRef-zephyr:SPDXRef-File-thread.c
Relationship: SPDXRef-File-libarch--arm--core--cortex-m.a GENERATED_FROM DocumentRef-zephyr:SPDXRef-File-cpu-idle.c
Relationship: SPDXRef-File-libarch--arm--core--cortex-m.a GENERATED_FROM DocumentRef-zephyr:SPDXRef-File-irq-init.c

...

FileName: ./zephyr/zephyr.elf
SPDXID: SPDXRef-File-zephyr.elf
FileChecksum: SHA1: 2e80741d3c373bd7626bc49625783ea8fd1bcacb
FileChecksum: SHA256: 7a838128652e85835f9167be429d41559701533fbd0d09b6bab9176a289fdc5e
LicenseConcluded: NOASSERTION
LicenseInfoInFile: NONE
FileCopyrightText: NOASSERTION

Relationship: SPDXRef-File-zephyr.elf GENERATED_FROM DocumentRef-zephyr:SPDXRef-File-empty-file.c
Relationship: SPDXRef-File-zephyr.elf GENERATED_FROM SPDXRef-File-isr-tables.c
Relationship: SPDXRef-File-zephyr.elf STATIC_LINK SPDXRef-File-libapp.a
Relationship: SPDXRef-File-zephyr.elf STATIC_LINK SPDXRef-File-libzephyr.a

...
```

Vulnerability Infrastructure → Github 2021

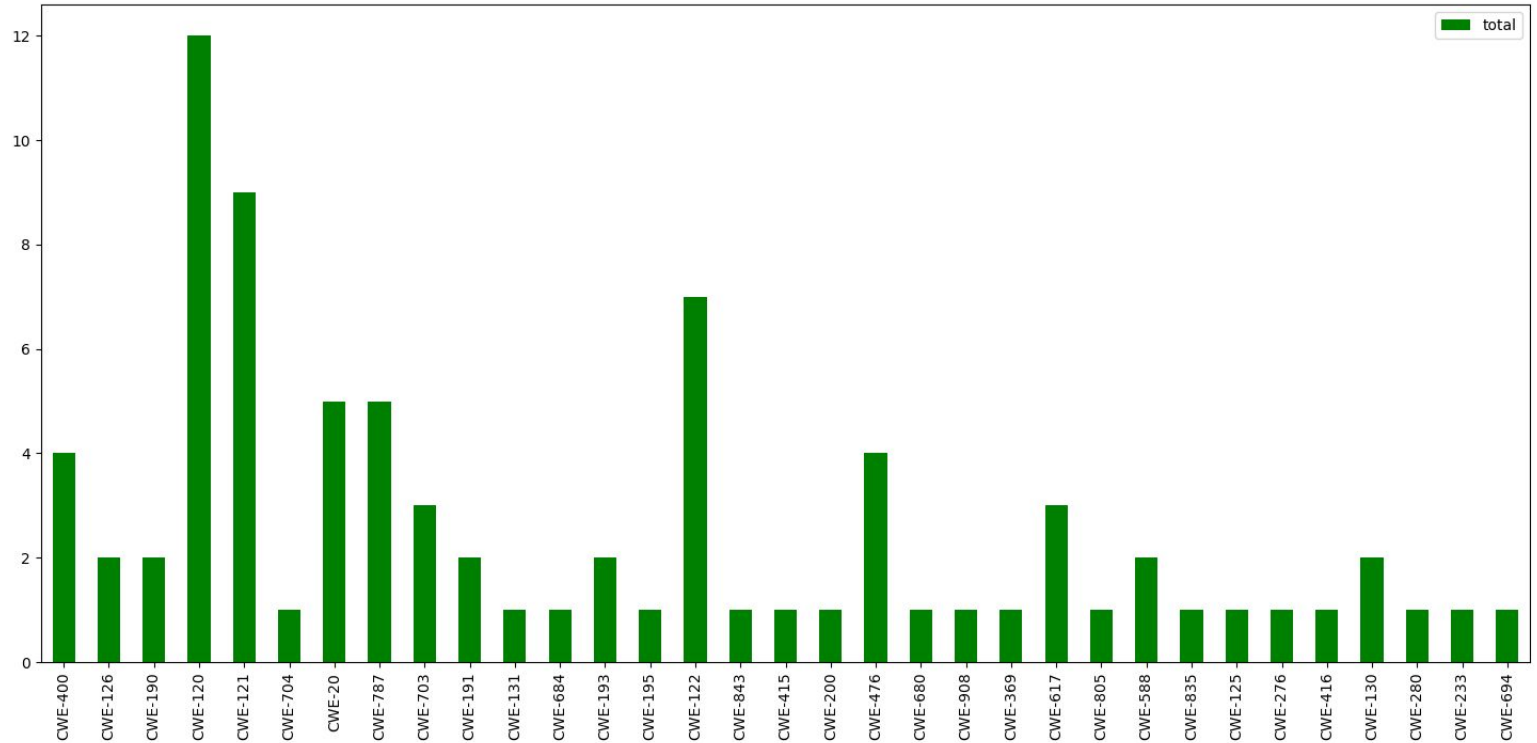
Why Transition?

Private repos became available. Better integration with rest of code.
No additional ids to manage. Improved analysis capabilities

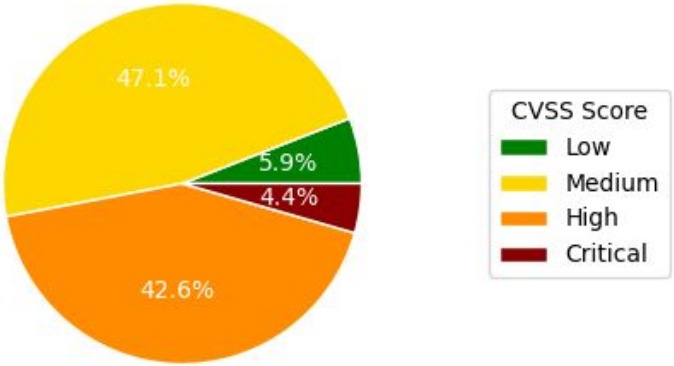
Total of CVEs published : 68 (since we started using github)

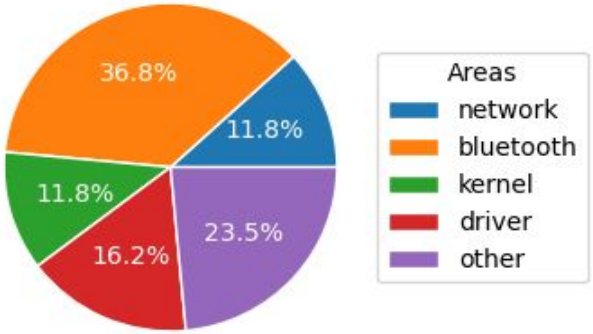


CWE Breakdown



Scoring & Code Area Breakdown





Security Working Group added March 2022

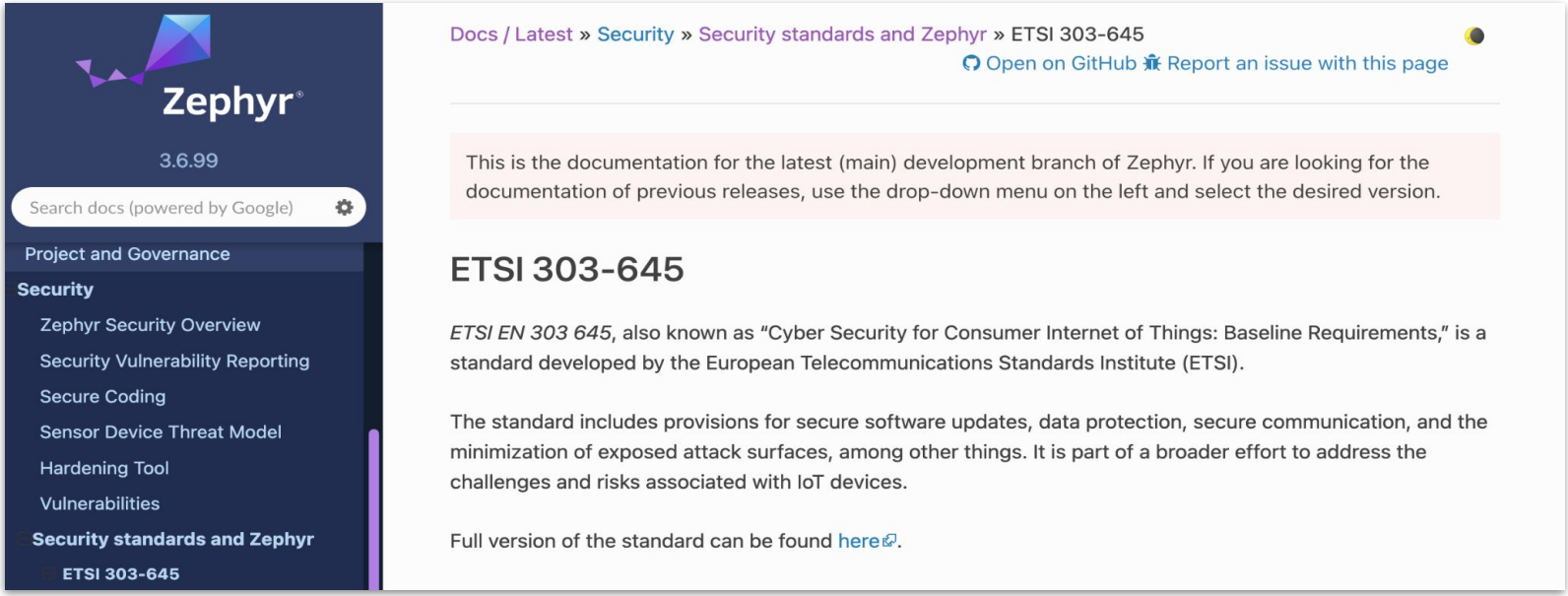
Security Committee

- **Restricted** to one representative from each platinum member, an architect (Flavio Ceolin), and a chair (David Brown)
- Meeting: Every 2 weeks
- Topics:
  - Vulnerabilities
  - PSIRT processes
  - Financial/contracts
  - Other sensitive information

Security Working Group

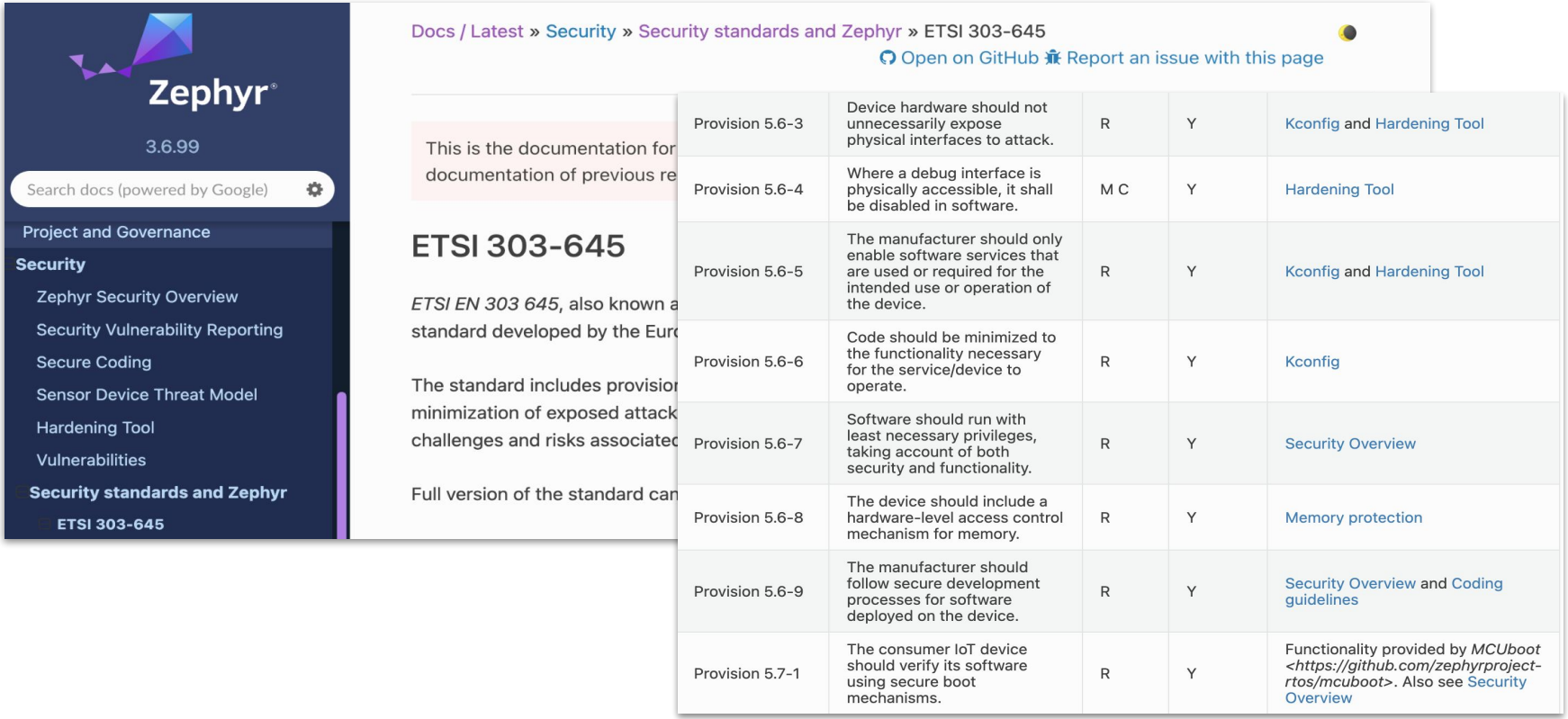
- **Open** to any participant
- Meeting: Every 2 weeks
- Topics:
  - Security Standards
    - ETSI EN 303-645
    - FIPS 140-3
    - SP 800-128
    - Annex K (C11 standard)
  - Evolving Security Processes
  - Code Analysis Tools
  - Documentation

Work on ETSI EN 303-645 in 2023



Source: https://docs.zephyrproject.org/latest/security/standards/etsi-303645.html

Work on ETSI EN 303-645 in 2023



Provision 5.6-3	Device hardware should not unnecessarily expose physical interfaces to attack.	R	Y	Kconfig and Hardening Tool
Provision 5.6-4	Where a debug interface is physically accessible, it shall be disabled in software.	M C	Y	Hardening Tool
Provision 5.6-5	The manufacturer should only enable software services that are used or required for the intended use or operation of the device.	R	Y	Kconfig and Hardening Tool
Provision 5.6-6	Code should be minimized to the functionality necessary for the service/device to operate.	R	Y	Kconfig
Provision 5.6-7	Software should run with least necessary privileges, taking account of both security and functionality.	R	Y	Security Overview
Provision 5.6-8	The device should include a hardware-level access control mechanism for memory.	R	Y	Memory protection
Provision 5.6-9	The manufacturer should follow secure development processes for software deployed on the device.	R	Y	Security Overview and Coding guidelines
Provision 5.7-1	The consumer IoT device should verify its software using secure boot mechanisms.	R	Y	Functionality provided by *MCUboot <https://github.com/zephyrproject-rtos/mcuboot>*. Also see Security Overview

Source: https://docs.zephyrproject.org/latest/security/standards/etsi-303645.html#provisions-assessment

2024 Security Audit with NCC Group

Why External Audit?

- Identifying Vulnerabilities
- Independent Assessment
- Best Practices
- Community Trust
- Reputation

Scope Definition

- Security Objectives
- Components
  - Narrow to something doable and that benefits most users
- Depth of Analysis
- Threat Model

Results from NCCGroup

- Target Zephyr 3.6 / 3.7
  - 02/2024 ~ 03/2024
- Three issues found
  - Two low severity caused by integer overflow and TOCTOU
  - One informational caused by integer overflow

Lessons Learned from the Audit

Defining the scope is hard

- Resource Constraints
- Depth and Breadth
- Future-Proofing
- Stakeholder Agreement

Threat model is useful

- Guiding the Audit Process
- Validating Security Controls
- Facilitating Communication

Comprehensive testing importance

- The audit make it clear the importance of comprehensive testing

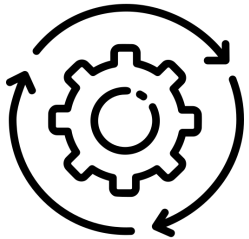
Outcomes:

- Enhanced Security
  - The identification and subsequent remediation of even low-severity issues contribute to a more secure system
- Increased Confidence
  - Third-party auditor validated the security and quality of the code base increasing confidence among developers, stakeholders, and users
- Recommendations aligned with Zephyr plans
  - Guided Fuzzing of Libraries and Subsystems

More Details Available...

Details at:
https://www.youtube.com/watch?v=vEG-Owv9TEs&list=PLzRQULb6-ipHnRUuy2UJpqZjTM9FPWtWx&index=22

Zephyr Security Summary



Weekly Coverity scans

MISRA scans

Automated Code checks per pull request

Documented secure coding practices

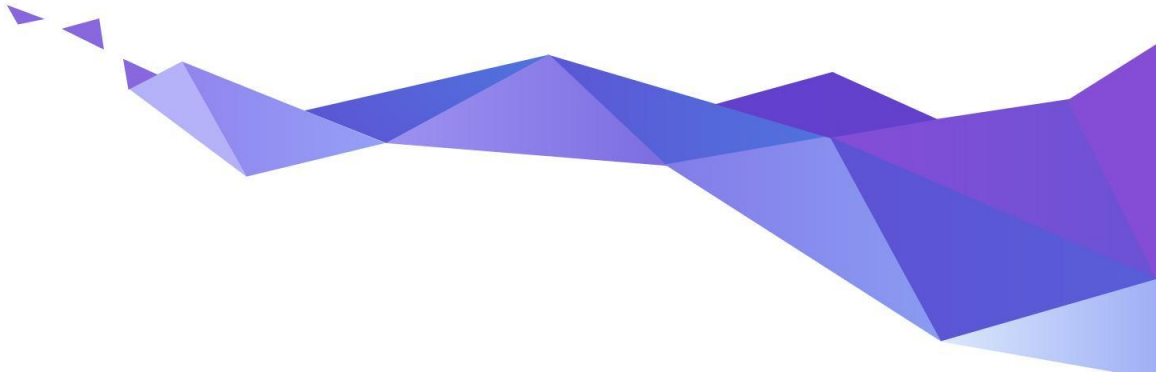
Vulnerability response criteria publicly documented

SBOM generation

per

ISO/IEC 5962:2021

What about Zephyr and safety?



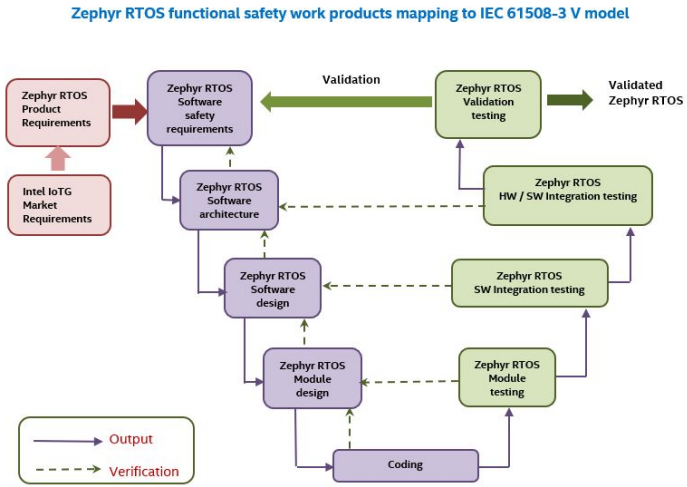
Auditable

- An **auditable code base** will be established from a **subset of the Zephyr OS LTS**
- Code bases will be kept in sync
- More rigorous processes (necessary for certification) will be applied to the auditable code base.
- Processes to achieve selected certification to be:
  - Determined by Safety Committee and Security Committee
  - Coordinated with Technical Steering Committee

Compliant Development: V-model

It is difficult to map a stereotypical open-source development to the V-model

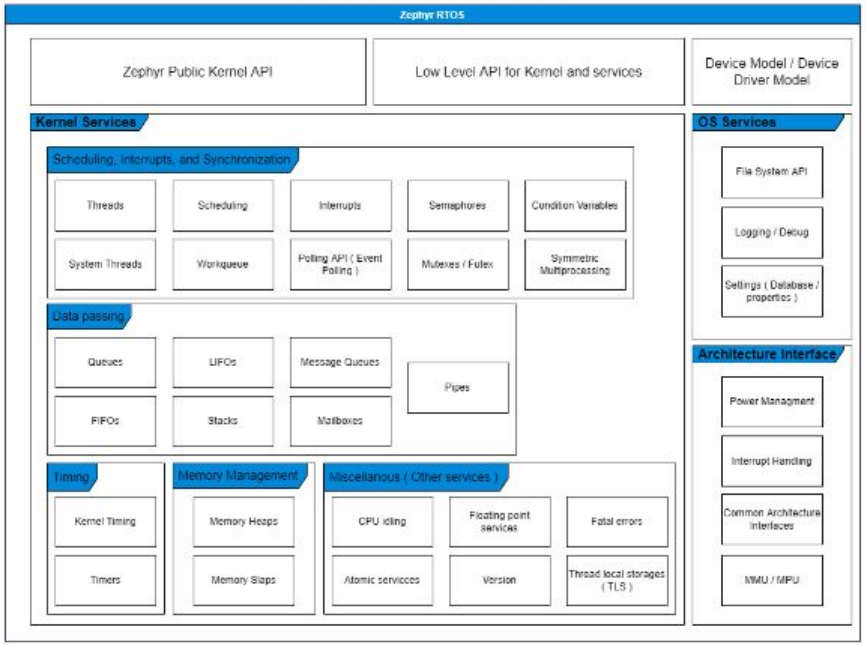
- Specification of features
- Comprehensive documentation
- Traceability from requirements to source code
- Number of committers and information known about them



⇒ Provide the evidences that open source developers can map to compliance and meet all requirements

Initial certification focus

- Start with a limited scope of kernel and interfaces
- Initial target is IEC 61508 SIL 3 / SC 3 (IEC 61508-3, 7.4.2.12, Route 3s)
- Option for 26262 ASIL D certification has been included in contract with certification authority should there be sufficient member interest



Scope can be **extended** to include **additional components** with associated **requirements** and **traceability** as determined by the safety committee

Safety Collateral Proposal

Zephyr®

Draft (Pending Approval by Certification Authority)			
Artifacts	**Type of Doc**	**Owner**	**Work in progress Visibility**
Plans	Category		
Safety Development Plan	Plan/Process	Safety Committee	Public - Project Docs
Safety Assesment Plan	Plan/Process	FSM	Safety Committee Github
Verification / Validation / Integration Test Plan	Plan/Process	Testing WG	Public - Project Docs
Software Development Plan	Plan/Process	TSC	Public - Project Docs
Configuration and Change Management Plan	Plan/Process	TSC	Public - Project Docs
Coding Guideline	Plan/Process	TSC	Public - Project Docs
Tools Documentation	Plan/Process	TSC	Public - Project Docs
Specifications	Category		
Safety Scope Definition	Spec.	Safety Committee	Safety Committee Github
Safety Software Requirement Specification (SRS) **	Spec.	Safety Committee	Safety Committee Github
Safety Software Architecture and Interface Specification (SAIS) **	Spec.	Safety Committee	Safety Committee Github
Safety Software Component Design Specification (SMDS) **	Spec.	Safety Committee	Safety Committee Github
Safety Software Component Test Specification (SMTS) **	Spec.	Safety Committee	Safety Committee Github
Safety Software Integration Test Specification (SITS) **	Spec.	Safety Committee	Safety Committee Github
Safety Software Test Specification (STS) **	Spec.	Safety Committee	Safety Committee Github
Sources	Category		
Source Code	Source	TSC	Public
- Coding Guideline Compliance	Source	TSC	Public
Project Documentaton	Source	TSC	Public
- Software Requirement Specifications	Spec	TSC	Public
- Software Architecture and Interface Specification	Spec	TSC	Public
- Software Component Design Specification	Spec	TSC	Public
Project Testing	Source	TSC	Public
- Software Component/Unit Test Specification	Spec	TSC	Public
- Software Integration Test Specification	Spec	TSC	Public
- Software Test Specification	Spec	TSC	Public
- Tests	Source	TSC	Public
Reports	Category		
Code Review Report (pre-merge)	Report	TSC	Public
Code Change Test Report (post-merge)	Report	Testing WG	Public
Test Coverage Report	Report	Testing WG	Public
Coding Guideline Compliance Report	Report	Safety WG & Security WG	Public
Traceability Report	Report	Safety WG	Public
Tools Classification	Report	Safety Committee	Public
Tools Validation	Report	Safety Committee	TBD (based on specific tools)
Fault Injection Test Report	Report	Safety Committee	Safety Committee
Safety Traceability Report (for Safety Scope) **	Report	Safety Committee/FSM	Safety Committee
Safety Test Coverage Report (for Safety Scope) **	Report	Safety Committee/FSM	Safety Committee
Safety Analysis (e.g., FMEA)	Report	FSM	Safety Committee
Manuals	Category		
Software User Manual	Manual	TSC	Public
Safety Manual	Manual	FSM	Safety Committee
Certificates			
All safety certificates	Certificate	Safety Committee	N/A

- Requirement definition, Source Code & Test linkage are **public**; and developed in open using strictdoc
- The set of requirements (and associated traceability) are applicable to safety scope is managed by the safety committee.
- Other project artifacts have owners designated.

What’s happening now..

Safety Committee

- Safety Certification Strategy decisions
  - Scope of certification
  - Certification standards
  - Certification timeline
- Assessment and audit specific tasks
- Owner of certification artefacts and managing contract with certification authority
- Participation limited to the project’s members, the safety architect and the functional safety manager

Safety Working Group

- Enabling safety qualifications/ certifications in the project
- Working on creating the required documentation and evidence in open
  - creating/deriving and documenting requirements
  - Linking requirements to code and tests
- Open to everyone to participate, join today: https://lists.zephyrproject.org/g/safety-wg

Doulos, Honda, Hubble Network, IAR, inovex and Microchip Technology join the Zephyr Project as it gets Closer to Safety Certification

January 30, 2025

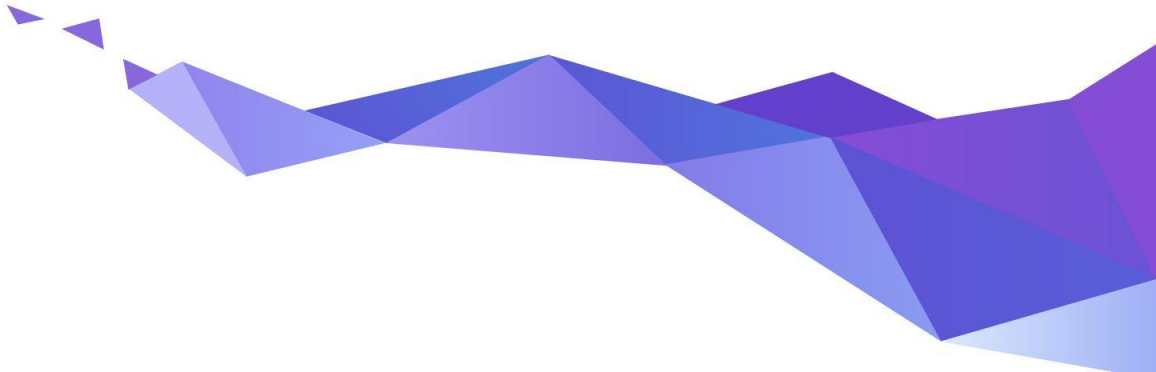
See Zephyr RTOS at FOSDEM on February 1-2

SAN FRANCISCO, January 30, 2025 – Today, the Zephyr® Project announced that Doulos, Honda, Hubble Network, IAR, inovex and Microchip Technology have joined as Silver members. Zephyr, an open source project at the Linux Foundation that builds a secure, connected and flexible RTOS for future-proof and resource-constrained devices, is easy to deploy and manage. It is a proven RTOS ecosystem created by developers for developers.

Last year, the project achieved several milestones including obtaining written concept approval for IEC 61508 certification of the Zephyr kernel. The Zephyr Project will continue to advance the functional safety and quality management processes for a safety element out of context (SEooC) that meets the requirements of the IEC 61508 standard, which is a globally recognized benchmark for ensuring the functional safety of systems, and a foundation for other safety standards. Compliance with IEC 61508 ensures that a system is developed and maintained with a rigorous approach to minimizing risks and increasing operational reliability. By integrating these processes into the development lifecycle, Zephyr aims to ensure traceability, transparency and accountability at every stage, from initial design to deployment and maintenance.

Source: https://zephyrproject.org/doulos-honda-hubble-network-iar-inovex-and-microchip-technology-join-the-zephyr-project-as-it-gets-closer-to-safety-certification/

Results from applying best practices?



New Products based on Zephyr

Oticon More Hearing Aid

Lildog & Lilcat Pet Tracker

Livestock Tracker

Moto Watch 100

Samsung Galaxy Ring

Proglove

Adhoc Smart Waste

Google Chromebook

Framework laptop

Keeb.io BDN9

Hati-ACE

Safety Pod

BLiXT solid state circuit breaker

Aethero Deimos Satellite

PHYTEC Distancer

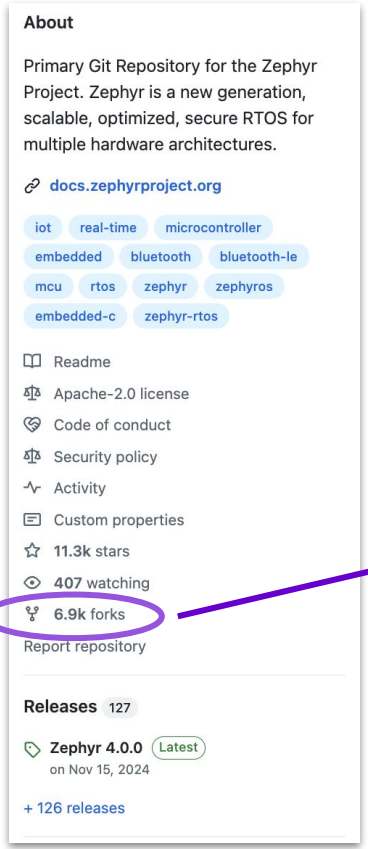
Laird Connectivity sensors & gateways

BeST pump monitoring

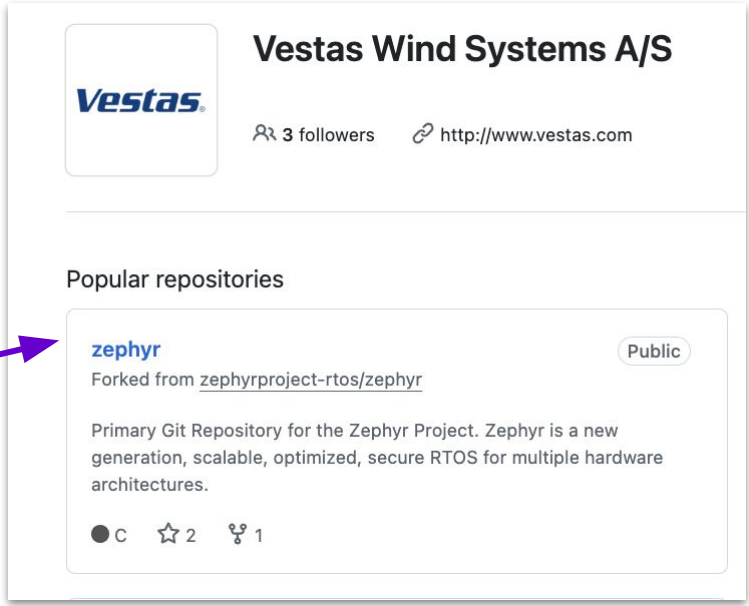
Vestas Wind Turbines

zephyrproject.org/products-running-zephyr

Zephyr in the wild… 6.9K Forks!



Source: https://github.com/zephyrproject-rtos/zephyr



Source: https://github.com/vestas-wind-systems

Supported Hardware Architectures

Cortex-M, Cortex-R
& Cortex-A

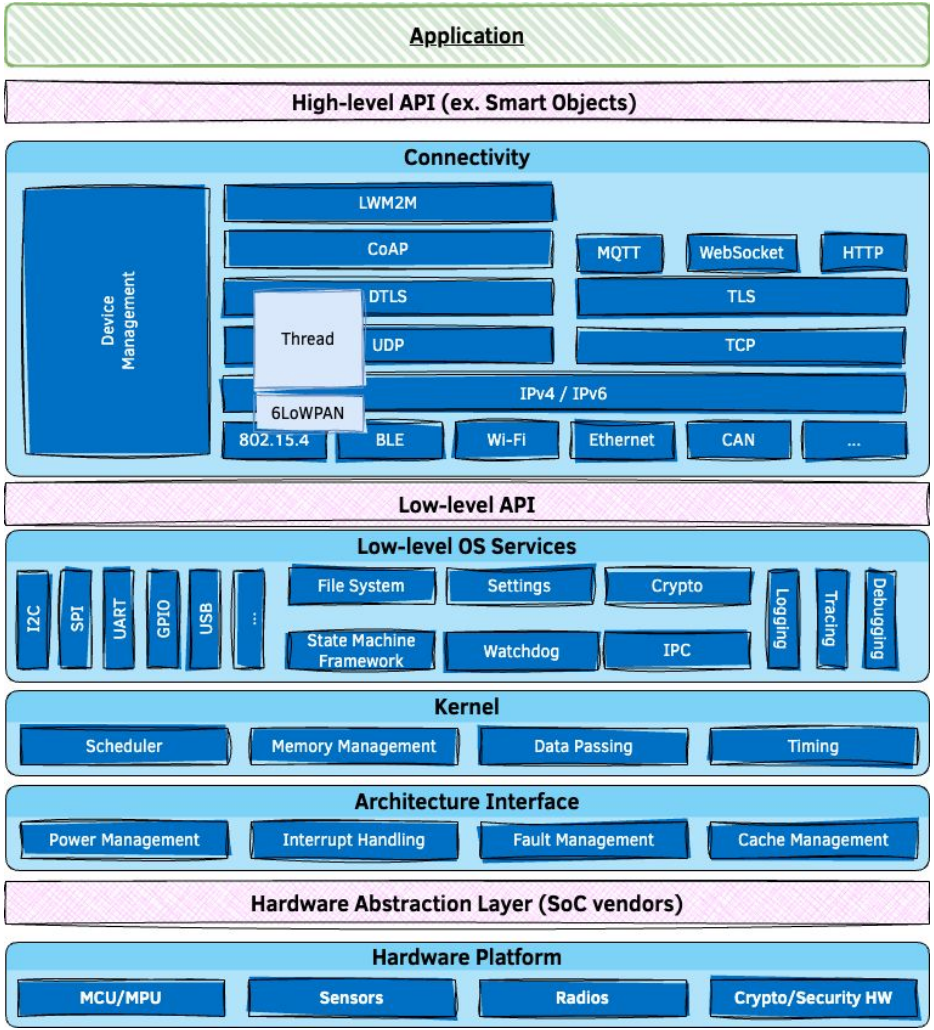
x86 & x86_64

32 & 64 bit

Xtensa

docs.zephyrproject.org/latest/hardware/index.html#hardware-support

Software Architecture



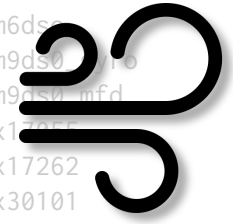
220+ Sensors Already Integrated

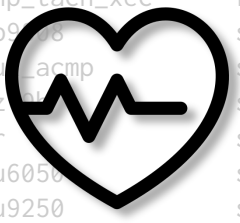
adt7420
adxl345
adxl362
adxl372
ak8975
amg88xx
ams_as5600
ams_iAQcore
apds9960
bma280
bmc150_magn
bme280
bme680
bmg160
bmi160
bmi270
bmm150
bmp388
bq274xx
ccs811

dht
dps310
ds1[illegible]
ens[illegible]
esp[illegible]
fdc[illegible]
fx[illegible]
fxo[illegible]
grove
grow_r502a
hmc5883l
hp206c
ht[illegible]221
i[illegible]250d
i[illegible]2605
i[illegible]2670
i[illegible]
icp[illegible]125
iis2dh
iis2dlpc

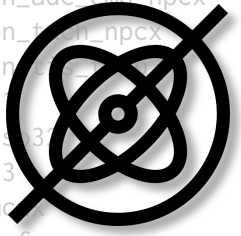
iis2iclx
iis2mdc
iis3dhhc
ina219
ina23[illegible]
isl2[illegible]35
ism3[illegible]
ite_ta[illegible]x2
ite_vcmp_it8xxx2
lis2dh
lis2ds12
lis2dw12
lis2[illegible]
lis3[illegible]
lm75[illegible]
lm77[illegible]
lps22[illegible]
lps22hh
lps25hb
lsm303dlhc_magn

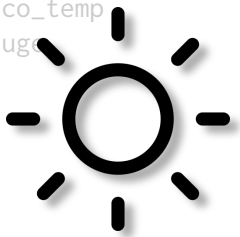
lsm6ds0
lsm6dsl
lsm6ds[illegible]
lsm9ds0[illegible]
lsm9[illegible]mfd
max1[illegible]
max17262
max30101
max31875
max44009
max6675
mchp_tach_xec
mcp9[illegible]
mcu[illegible]acmp
mhz[illegible]
mpr
mpu6050
mpu9250
ms5607
ms5837





nrf5
nuvoton_adc_cmp_npcx
nuvoton_t[illegible]_npcx
nxp_kin[illegible]
opt300[illegible]
pcnt_es[illegible]
pms7003
qdec_mc[illegible]
qdec_nrfx
qdec_sam
qdec_stm32
rpi_pico_temp
sbs_gaug[illegible]
sgp40
sht3xd
sht4x
shtcx
si7006
si7055
si7060





si7210
sm351lt
stm32_temp
stm32_vbat
stmemsc
stts751
sx9500
th02
ti_hdc
ti_hdc20xx
tmp007
tmp108
tmp112
tmp116
vcnl4040
vl53l0x
wsen_hids
wsen_itds

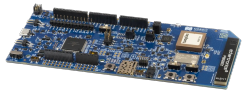
github.com/zephyrproject-rtos/zephyr/tree/main/drivers/sensor

700+ supported boards… and growing

Arduino Portenta H7

ESP32

Sipeed HiFive1



nRF9160 DK

STM32F746G Disco

M5StickC PLUS

TDK RoboKit 1

BBC micro:bit v2

Blue Wireless Swan

Arduino Nano 33 BLE

Intel UP Squared

Dragino LSN50 LoRA Sensor Node

Microchip SAM E54 Xplained Pro Evaluation Kit

Raspberry Pi Pico

Altera MAX10

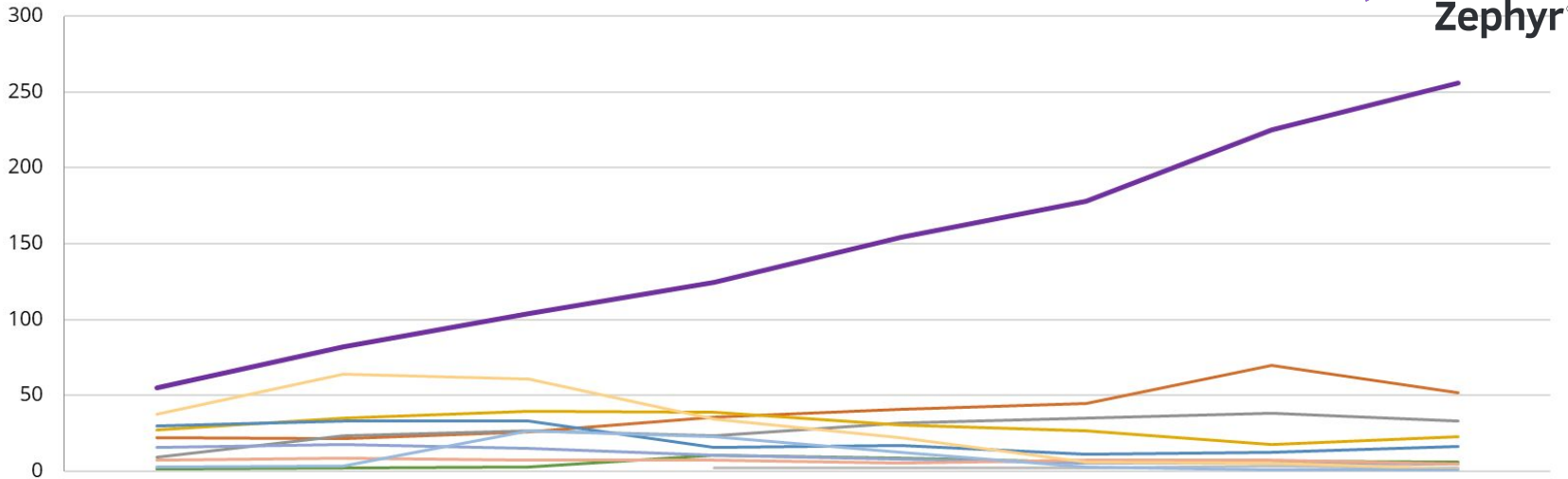
NXP i.MX8MP EVK

Adafruit Feather M0 LoRa

u-blox EVK-NINA-B3

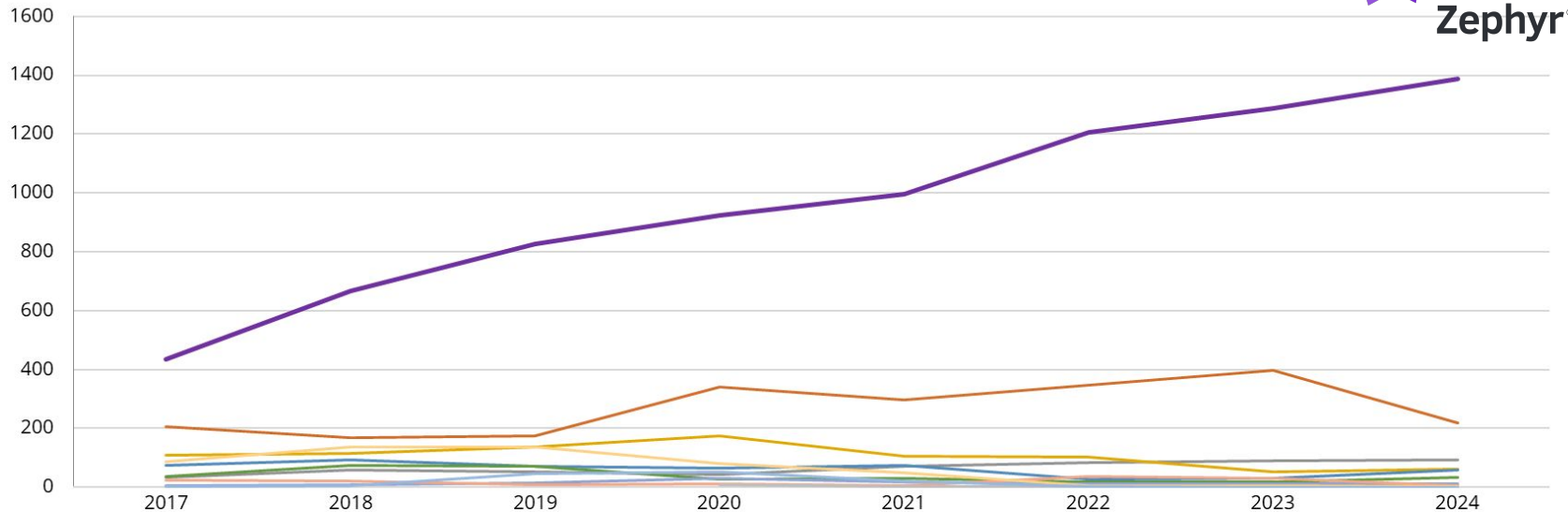
docs.zephyrproject.org/latest/boards

Average Number of Unique Contributors per Month



	2017	2018	2019	2020	2021	2022	2023	2024
Zephyr	55	82	104	125	154	178	225	256
Apache NuttX	22	22	26	36	41	45	70	52
RT-Thread	9	24	26	23	32	35	38	33
RIOT OS	27	35	39	39	30	27	18	23
TizenRT	30	33	33	15	17	11	13	16
FreeRTOS	2	2	3	11	8	6	7	6
Apache Mynewt	16	18	15	11	8	5	5	5
Contiki-NG	8	9	7	7	5	7	7	5
Eclipse ThreadX				2	2	2	3	2
Arm Mbed OS	38	64	61	35	22	6	5	2
Amazon FreeRTOS	3	3	27	23	13	3	1	1

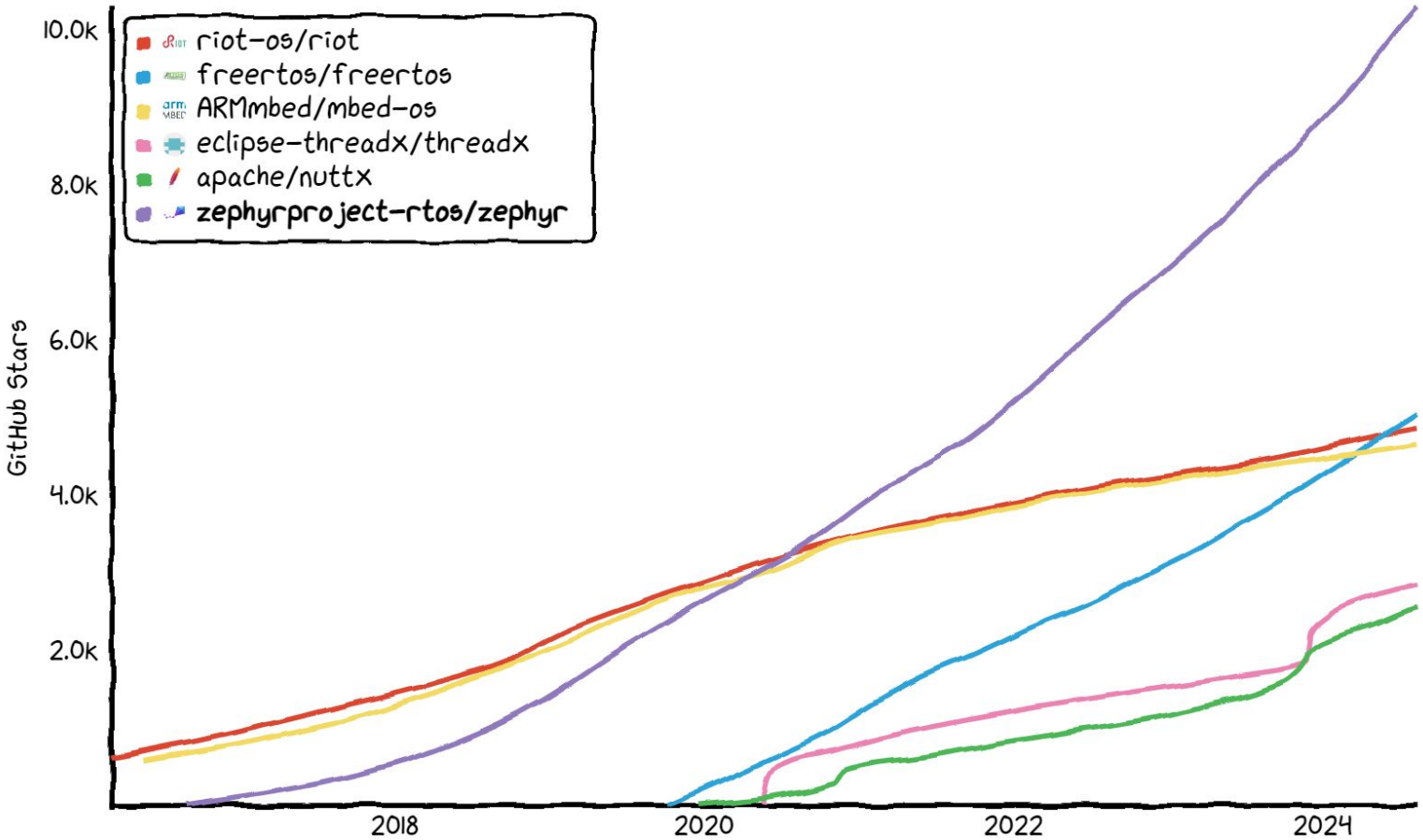
Average Number of Commits per Month



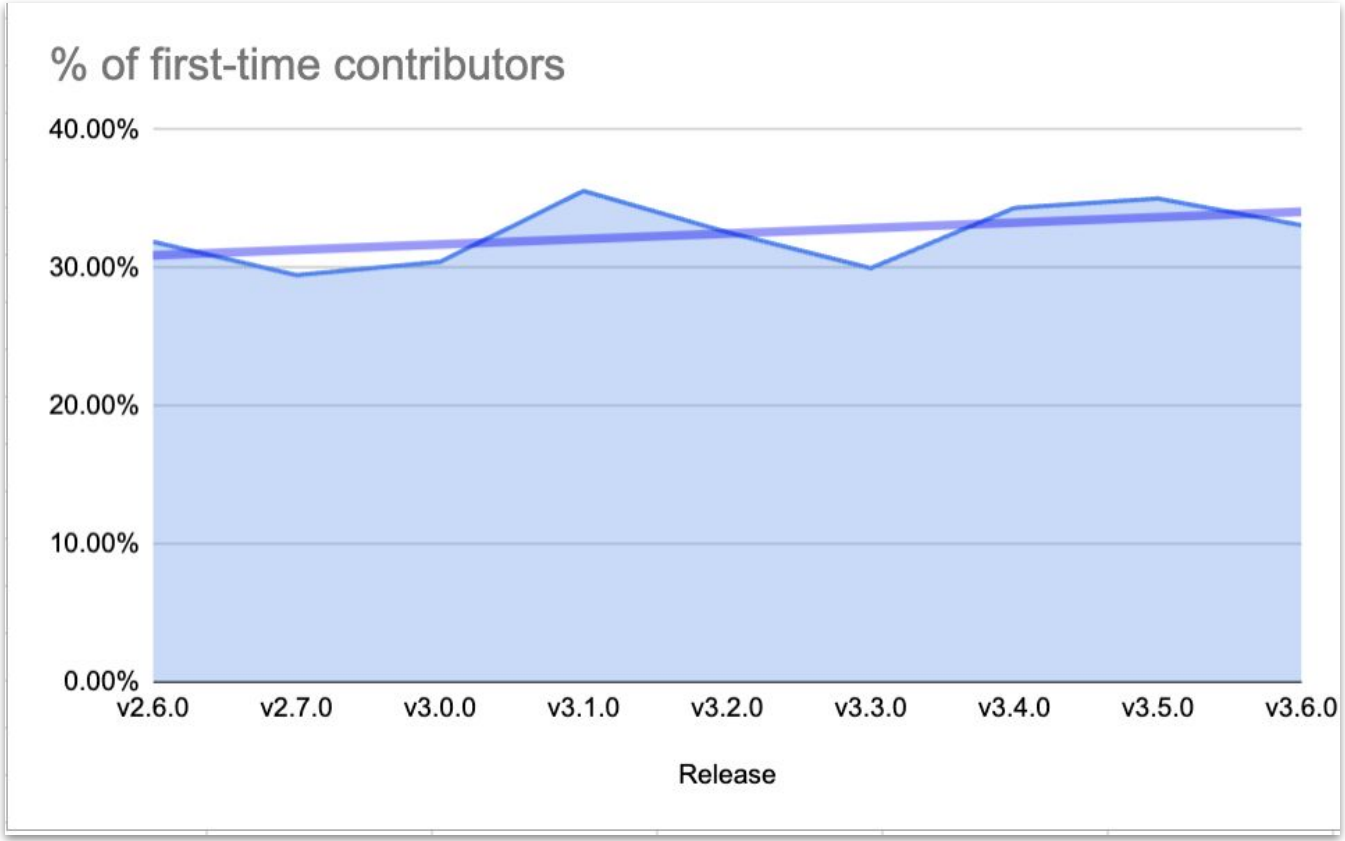
	2017	2018	2019	2020	2021	2022	2023	2024
Zephyr	434	667	825	924	995	1206	1287	1387
Apache NuttX	206	170	175	342	297	347	397	219
RT-Thread	35	59	53	43	70	84	91	92
RIOT OS	108	115	136	175	105	103	52	61
TizenRT	73	93	71	64	74	27	29	58
Apache Mynewt	38	74	70	27	31	18	19	33
FreeRTOS	4	8	13	32	17	11	12	11
Contiki-NG	23	22	9	11	7	38	30	6
Eclipse ThreadX				7	1	2	3	2
Arm Mbed OS	86	136	138	82	51	6	5	2
Amazon FreeRTOS	2	4	47	53	20	2	0	0

Zephyr®

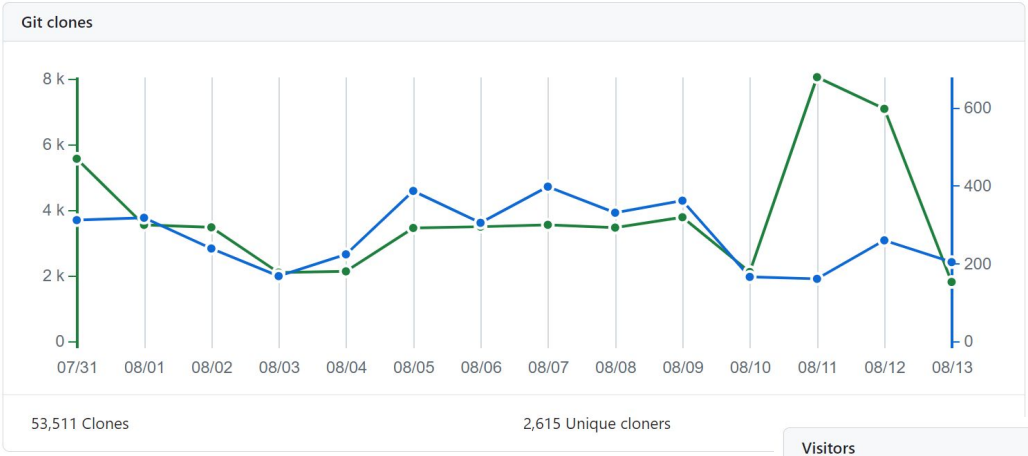
Star History

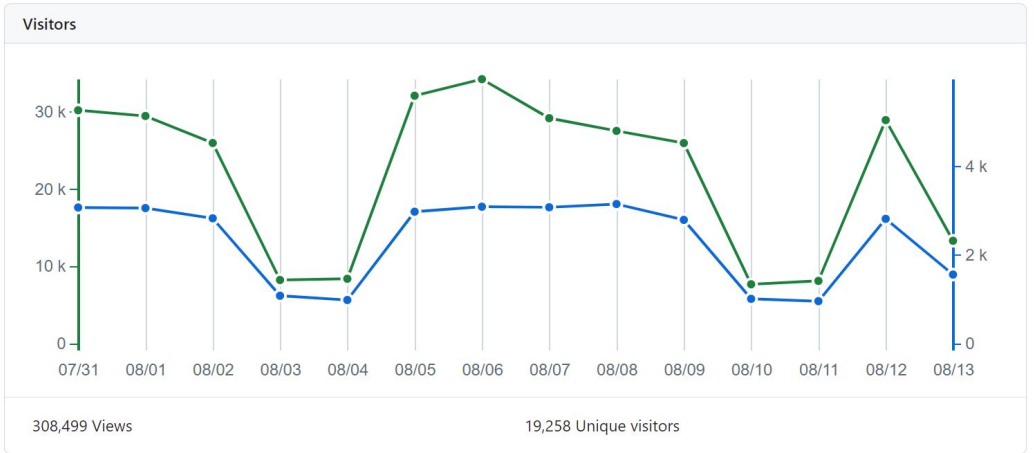


New Contributors per Release



GitHub Clones & Unique Visitors

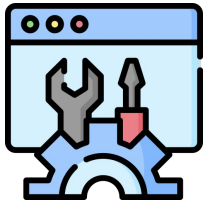




2024-07-31 → 2024-08–13

~186 unique clones per day
~1375 unique visitors per day

Vibrant Ecosystem



Development Tools

Zephyr®

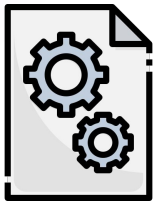
Governing Board

Technical Steering Committee

Contributors

Applications & Middlewares

Training & Consulting



Firmwares & Libraries

Ecosystem // Developer Tools

Zephyr®

Development Tools

Training & Consulting

Firmwares & Libraries

Applications & Middlewares

IDE





Compilers

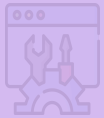


Debuggers / Tracing Tools

Emulation / Simulation

Ecosystem // Training & Consulting

Zephyr®



Development Tools

Training & Consulting

Firmwares & Libraries

Applications & Middlewares

Training

NORDIC SEMICONDUCTOR

Golioth

THE LINUX FOUNDATION

percepio SENSING SOFTWARE

ac6

DOULOS

Savoir-faire LINUX

Services & Consulting

ANALOG DEVICES

antmicro

Baylibre

AVSYSTEM

ezurio

NXP

WNDRVR

Ecosystem // Firmwares & Libraries

Zephyr®

Development Tools

Training & Consulting

Firmwares & Libraries

Applications & Middlewares

Security

wolfSSL

psacertified™

Mbed TLS

TrustedFirmware.org

TinyML

TensorFlow Lite

EDGE IMPULSE

Language runtimes

C++

JS

Rust

Lua

MicroPython

Others

SOUND OPEN FIRMWARE

zscilib

nRF Connect SDK

Memfault

Zenoh

Ecosystem // Apps & Middlewares

Zephyr®

Development Tools

Training & Consulting

Firmwares & Libraries

Applications & Middlewares

Remote Management

AVSYSTEM blues Blynk Golioth

HERALD Memfault STERNUM

Graphical Interfaces

LVGL MICROEJ Qt Group slint

Robotics

ROS

Zephyr Project: Platinum Members

ANALOG DEVICES

antmicro

CARIAD
A VOLKSWAGEN GROUP COMPANY

Google

intel.

Meta

NORDIC® SEMICONDUCTOR

NXP

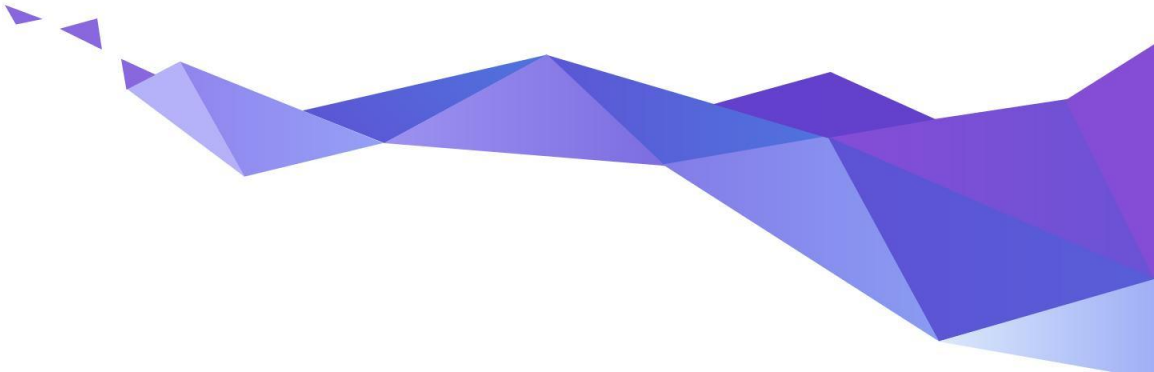
oticon
life-changing technology

Qualcomm Innovation Center

ZEISS

Zephyr Project: Silver Members

What's next?



Focus areas:

- Impact of growth on Maintainers
- Project driven benchmarking
- Test infrastructure rework
- CRA readiness
- Domain expertise for requirement formulation

Improving Contributor Diversity

Short Survey (inspired by Rust survey) at:
https://linuxfoundation.research.net/r/zephyr-diversity

Zephyr Participation Information

zephyrproject.org

github.com/zephyrproject-rtos

lists.zephyrproject.org

chat.zephyrproject.org

Zephyr®

zephyrproject.org