.

<(NAME) Project: (Workstream NAME)>

<Title: 3 Layer (Base, Design and Product) **Software** Specification Template>

<Version>.<Revision>.<Errata>

Author: <Primary>

Author: <Secondary, etc. Delete if unnecessary>

| DELETE THIS BLOCK |
| --- |
| ***INSTRUCTIONS FOR ALL FOLLOWING SECTIONS:***   * + *This section is only required if there is a directly linked preceding document which means this is a revision to the original document. If this is a completely new version (original) you may delete this page.*   + *The revisions in the populated table below apply to the template, please start from a blank table for your contribution.*   + *See the OCP Release Nomenclature Guidelines for additional information.* |

# Version History

Note, refer to the OCP Contribution Versions, Revisions and Errata best practices documentation. Generally speaking, Versions and revisions are made to this document and logged here. Errata is a separate document such that the contribution specification document it refers to was not revised. Ex: Version 1, Errata E1

| **Date** | **Version #** | **Author** | **Description** |
| --- | --- | --- | --- |
| 01 JAN 24 | 1.0.0 | Bijan Nowroozi/Michael Shill/Rob Coyle | Initial Release |
| 17 JUN 24 | 1.0.1 | Bijan Nowroozi | Modified title page to include project information and added some context on p.10 openness tenet, Appendix C updates |
| 30 SEP 24 | 1.1.0 | Bijan Nowroozi | Added usage note and compliance sections |
| 07 NOV 24 | 1.2.0 | Bijan Nowroozi | Reflected New License, cleaned up instructional text boxes, tenets |
| 11 NOV 24 | 1.3.0 | Bijan Nowroozi | Numerous formatting changes |
|  |  |  |  |
|  |  |  |  |

# 

# Current Template Version:

3 Layer (Base, Design and Product) Software Specification Template V1.3.0

**Table of Contents**

[**Version History 3**](#_5hqof536mr77)

[**Current Template Version: 3**](#_4x8ntvex5qai)

[1.1. Open Web Foundation (OWF) CLA 8](#_oeo8w428bs5y)

[**1. Acknowledgements 9**](#_3yfbtftpm0pp)

[**2. Compliance with OCP Tenets 10**](#_zi3i5v28cisd)

[2.1. Openness 10](#_j6kl13o94r5u)

[2.2. Efficiency 10](#_fj7vgvyr9nfc)

[2.3. Impact 11](#_ulz3hvb2ue1c)

[2.4. Scale 11](#_b5zwsm4bvnuo)

[2.5. Sustainability 11](#_rrf2lq0xcws)

[**3. Scope 12**](#_1270qnky2lr6)

[3.1. Usage 12](#_v4evsm4gzbtk)

[3.2. Layered Contribution Model 13](#_ipy6h41u13x8)

[3.3. Base Specification Layer 14](#_lbi1p4r0aq5j)

[3.4. Software Design Specification Layer 15](#_yt6qhwwnizn8)

[3.5. Software Product Specification Layer 16](#_7mpyp541xqk5)

[**4. Overview 18**](#_4gwkk5dvs444)

[**5. Base Specifications 19**](#_8war03ej6o32)

[5.1. Description 19](#_6z5emikflibt)

[**5.2. Goals and Vision 19**](#_bxd6873vcdcx)

[5.3. User Requirements/User Stories 19](#_nyqydh83w9f)

[**5.4. Solution Architecture 19**](#_enlfoe72z9vl)

[5.5. Functional Requirements 19](#_hwvzzfqama4t)

[5.6. Non-Functional Requirements 19](#_rtibi0f8dcyg)

[5.7. Features 19](#_ep324l2zbd55)

[5.8. Interface Requirements 20](#_qellu0vp1z9d)

[5.9. Performance Requirements 20](#_cvp9k11m7nfr)

[5.10. Security Requirements 20](#_fl1jwtxiiztk)

[**5.11. Constraints 21**](#_ug7necm1i98r)

[5.12. Test and Validation Requirements 21](#_eqrp8za7umqi)

[**5.13. Compliance 21**](#_d34ppqie4dsq)

[**5.14. Software Support (Strongly recommended) 22**](#_vqftl4n2kqkf)

[5.15. Repository Location 22](#_z1bfnswi7yqd)

[**5.16. Prescribed Materials 22**](#_u54ppg2adls)

[**5.17. References (recommended) 23**](#_fy8siocl3sx)

[**6. Design Specifications 24**](#_v6a73aq1zqtb)

[6.1. Description 24](#_9g6tu9lnd7f5)

[**6.2. Solution Architecture 24**](#_kblf1qvu8gnz)

[**6.3. Modules and Components 25**](#_8p9398wqf8zu)

[6.4. Interfaces 26](#_1ktvcvrsl4az)

[**6.5. Control and Data Flow 27**](#_xns4mc2jlxc)

[**6.6. External Dependencies 29**](#_5tegjrq7eqja)

[6.6.2. System 29](#_biz1b4shqv7r)

[**6.7. Scalability 29**](#_ld7gu2oyh69b)

[**6.9. Management 31**](#_wblq38pgacfu)

[**6.11. Compliance 31**](#_tf51mmeple2b)

[6.12. Software Support (Strongly recommended) 32](#_mox7rh49i141)

[6.13. Repository Location 32](#_k3bnkg9rla0j)

[6.14. Prescribed Materials 33](#_q1tq34t6k104)

[6.15. References (recommended) 33](#_vhr1djkgec9l)

[**7. Product Specifications 33**](#_8g16s2u7godi)

[7.1. Description 34](#_lducr121iufp)

[7.2. Solution Architecture 34](#_94xjoo9su3yn)

[7.3. Modules and Components 34](#_s47d9s3hhudv)

[7.4. Interfaces 35](#_ynq0bbkxkdtc)

[7.4.2. Signal List 36](#_d14c91z1la4j)

[7.5. Control and Data Flow 36](#_5jkl11mjwdcl)

[7.6. External Dependencies 38](#_1uz1ytlal4ts)

[7.6.1. System 38](#_jlv5esdq19tc)

[7.7. Scalability 38](#_am5l7ijyrup3)

[7.9. Management 40](#_x5460yfuqgon)

[7.11. Compliance 40](#_rb9mjprwhmux)

[7.12. Software Support (Strongly recommended) 41](#_46hlim3urubu)

[7.13. Repository Location 41](#_yjt8ldo0gvd5)

[7.14. Prescribed Materials 42](#_69ktokcfy4pr)

[7.15. References (recommended) 42](#_lbo634tdnvyt)

[**Appendix A - Checklist for Steering Committee (SC) approval of this Specification (to be completed by contributor(s) of this Spec) 43**](#_fu98xdxru854)

[**Appendix B-\_\_ <supplier name> - OCP Supplier Information and Hardware Product Recognition Checklist 44**](#_y1ll55eabeef)

[**Appendix C - Contribution Process FAQs 46**](#_4eb5pwm0ic6z)

# 

THE UPDATED DEFAULT CONTRIBUTOR LICENSE AGREEMENT (CLA) IS [**OWFa 0.9**](https://146a55aca6f00848c565-a7635525d40ac1c70300198708936b4e.ssl.cf1.rackcdn.com/images/ed0befaf86bee2568ad720ff4a9a554d1f4260f7.pdf). PLEASE VERIFY THE CORRECT CLA/FSA IS USED AND EXECUTED FOR THIS CONTRIBUTION.

## Open Web Foundation (OWF) CLA

Contributions to this Specification are made under the terms and conditions set forth in **Modified Open Web Foundation Agreement 0.9 (OWFa 0.9)**. (As of October 16, 2024) (“Contribution License”) by:

**[Contributor Name(s) or Company name(s)]**

Usage of this Specification is governed by the terms and conditions set forth in **Modified OWFa 0.9 Final Specification Agreement (FSA)** (As of October 16, 2024) **(“Specification License”).**

You can review the applicable Specification License(s) referenced above by the contributors to this Specification on the OCP website at <https://www.opencompute.org/contributions/templates-agreements>.

​​For actual executed copies of either agreement, please contact OCP directly.

**Notes**:

The above license does not apply to the Appendix or Appendices. The information in the Appendix or Appendices is for reference only and non-normative in nature.

NOTWITHSTANDING THE FOREGOING LICENSES, THIS SPECIFICATION IS PROVIDED BY OCP "AS IS" AND OCP EXPRESSLY DISCLAIMS ANY WARRANTIES (EXPRESS, IMPLIED, OR OTHERWISE), INCLUDING IMPLIED WARRANTIES OF MERCHANTABILITY, NON-INFRINGEMENT, FITNESS FOR A PARTICULAR PURPOSE, OR TITLE, RELATED TO THE SPECIFICATION. NOTICE IS HEREBY GIVEN, THAT OTHER RIGHTS NOT GRANTED AS SET FORTH ABOVE, INCLUDING WITHOUT LIMITATION, RIGHTS OF THIRD PARTIES WHO DID NOT EXECUTE THE ABOVE LICENSES, MAY BE IMPLICATED BY THE IMPLEMENTATION OF OR COMPLIANCE WITH THIS SPECIFICATION. OCP IS NOT RESPONSIBLE FOR IDENTIFYING RIGHTS FOR WHICH A LICENSE MAY BE REQUIRED IN ORDER TO IMPLEMENT THIS SPECIFICATION. THE ENTIRE RISK AS TO IMPLEMENTING OR OTHERWISE USING THE SPECIFICATION IS ASSUMED BY YOU. IN NO EVENT WILL OCP BE LIABLE TO YOU FOR ANY MONETARY DAMAGES WITH RESPECT TO ANY CLAIMS RELATED TO, OR ARISING OUT OF YOUR USE OF THIS SPECIFICATION, INCLUDING BUT NOT LIMITED TO ANY LIABILITY FOR LOST PROFITS OR ANY CONSEQUENTIAL, INCIDENTAL, INDIRECT, SPECIAL OR PUNITIVE DAMAGES OF ANY CHARACTER FROM ANY CAUSES OF ACTION OF ANY KIND WITH RESPECT TO THIS SPECIFICATION, WHETHER BASED ON BREACH OF CONTRACT, TORT (INCLUDING NEGLIGENCE), OR OTHERWISE, AND EVEN IF OCP HAS BEEN ADVISED OF THE POSSIBILITY OF SUCH DAMAGE.

# Acknowledgements

The Contributors of this Specification would like to acknowledge the following for their feedback:

List all companies or individuals who may have assisted you with the specification by providing feedback and suggestions but did not provide any IP.

| DELETE THIS BLOCK BEFORE SUBMITTING |
| --- |
| INSTRUCTIONS FOR THE FOLLOWING SECTIONS:   * The Tenets section is required for contribution. * Replace the section text (keep titles) * Please describe how this Specification complies to the following OCP tenets. * Compliance is required for at least four of the five tenets (Sustainability is a required tenet). * The ideals behind open sourcing stipulate that everyone benefits when we share and work together. Any open source project is designed to promote sharing of design elements with peers and to help them understand and adopt those contributions. There is no purpose in sharing if all parties aren't aligned with that philosophy. * The OCP Steering Committee will look beyond the contribution for evidence that the contributor is aligned with this philosophy. The contributor actions, past and present, are evidence of alignment and conviction to all the tenets. * Scope section contains information for the contributor, all of it is expected to be replaced. |

# Compliance with OCP Tenets

Please describe how this Specification complies with the OCP tenets.

A full explanation of the OCP core tenets can be seen [here](https://146a55aca6f00848c565-a7635525d40ac1c70300198708936b4e.ssl.cf1.rackcdn.com/images/bf648bb75091907147e76846cad590f402660d2e.pdf).

## Openness

Openness is measured by the ability of third parties to build, modify, or personalize your contributed device, platform, or software. The OCP aims for completely open platforms that include all programmable devices, firmware, software, mechanical and electrical design elements, and any necessary external components or tools like software utilities. Contributors are highly encouraged to collaborate with other OCP Projects that may have complementary knowledge and expertise. Actively remove barriers to openness and demonstrate collaboration by sharing, seeking feedback, and accepting changes to designs and specifications. Ensure your contribution can be extended and enhanced by others.

## Efficiency

Your contribution should be more efficient than existing or prior generations. Efficiency can be demonstrated through reduced operational and capital expenses, improved performance, modularity, increased capacity, lower power or water consumption, better utilization, reduced size, or minimized code weight and latency in software. Clearly express efficiency gains with metrics valued by end-users when proposing your contribution.

## Impact

Your contribution should have a transformative impact on the industry by introducing new technology, accelerating time-to-market, or enabling technology through global supply chains. Impact is amplified when new technologies are made accessible to many customers worldwide. Examples include widely adopted specifications or more specifically, open security features that establish and verify product trust. Ensure your contribution creates meaningful positive impact within the OCP ecosystem.

## Scale

Design your contribution for easy implementation and deployment at any scale, with minimal intervention. Aim to create additive solutions where minimal usage or instances can be deployed and incrementally scaled as needed to effectively address the entire problem. Provide all necessary tools and supporting documentation, such as installation guides, initialization processes, configuration information, and details on obtaining service support. Include features like simple manual and automated maintenance, remote management, upgradability, and error reporting. Management tools should be open-sourced and/or made available to adopters.

## Sustainability

Your contribution must be sustainable, maximizing transparency of environmental impacts with the goal of continuous improvement. Focus on the responsible use of natural resources, fostering positive societal impacts, and minimizing environmental harm. This can be achieved through design decisions that promote circularity, efficient use of materials, power-saving features, and sustainability labeling. For software, consider optimizing code to reduce resource consumption and incorporating features that enable energy efficiency.

## 

# Scope

The purpose of this template is to define a software specification that includes the Base, Design and Product as layers, in a single document. This organization allows contributors to make a single contribution while also allowing others to reuse and extend parts of this specification. For example after the initial release of your contribution in this document, a subsequent modification can be made and a derivative specification can be created by referring to this document.

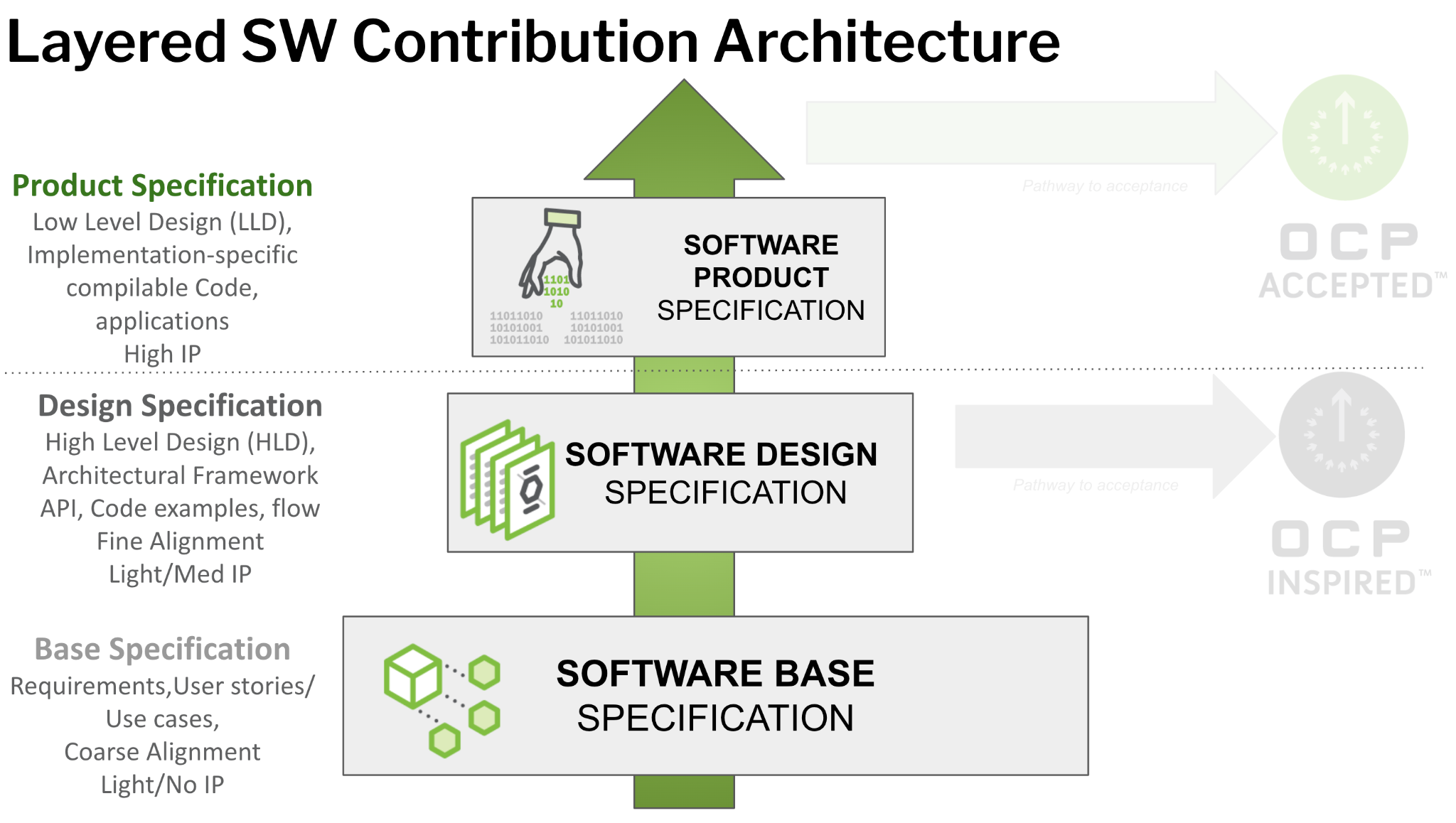


Figure 1: Specification Layers

## Usage

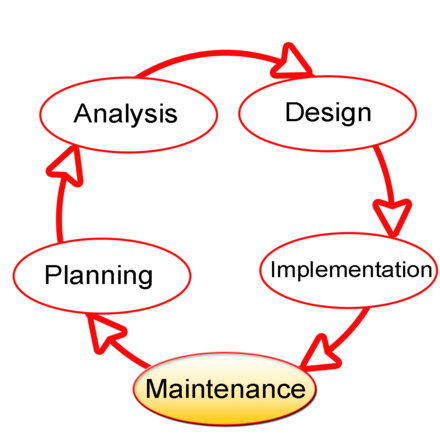
The sections displayed within this document contain examples. Any contributions that do not have any section may delete it. For example if the particular contribution is for API only- not many of the example sections herein apply, therefore contributors may delete.

## Layered Contribution Model

The Open Compute Project (OCP) has developed a layered contribution model to facilitate rapid development by contributors and collaborators, enable quick iteration and improvement by the entire OCP community, and support swift implementation by the broader ecosystem. This approach fosters the growth of an ecosystem around a solution, similar to building with Lego bricks. Ideas can iterate and improve over time while being easily modified for specific or vertical uses.

A key aspect of this approach for software development is aligning specification documents with the Software Development Life Cycle (SDLC). The intent is to unify the community on project goals and provide documentation that makes it easy for new collaborators to reuse the project, contribute to it, or understand the scope of subsequent efforts.

This diagram[[1]](#footnote-0) shows the SDLC phases:



The SDLC phases and their corresponding OCP specifications are as follows:

| **SDLC** | **OCP** |
| --- | --- |
| Planning | Base Software Specification |
| Analysis | Base Software Specification |
| Design | Software Design Specification |
| Implementation | Software Product Specification |
| Maintenance | Development considerations in all; operationally “Other” |

Note for additional information on the OCP layered contribution model, please see the [presentation](https://146a55aca6f00848c565-a7635525d40ac1c70300198708936b4e.ssl.cf1.rackcdn.com/images/4d14a1280e13c7dcf17123678eedc7e56053de96.pdf) and [recording](https://www.youtube.com/watch?v=IKqGFyE0V9E) on the Modular Contribution Process from the OCP Global Summit 2022 for more information.

## Base Specification Layer

*The Base Software Specification serves as a conceptual framework for coarse alignment. It includes requirements descriptions and potentially use cases or user stories for software modules or layers. Market requirements drive Base Specifications, and without defining details of a specific design, the Base Specification may have light IP content. This structure simplifies the process for multiple parties, including potential competitors, to engage in this phase.*

*This corresponds to the Planning and Analysis phases of the SDLC.*

*This layer defines the technical details for one of the following types of specifications:*

* *Conceptual framework for an extensible technology platform or layer, representing technical community-wide consensus and possibly used as a de facto standard.*
* *Requirements for a specific solution.*
* *Extension or modification of an existing specification (state which existing spec is being modified):*
  + *A complete version update.*
  + *A minor revision.*

*Note: Any supplier seeking OCP recognition must be 100% compliant with the requirements and compliance statements listed in the Base requirements and subsequent design and product specifications.*

## Software Design Specification Layer

*The Software Design Specification captures customer requirements for finer alignment by building on the Base Specification. This layer contains what is commonly referred to as the* ***High-Level Design (HLD)****, which includes:*

* *Architecture.*
* *Data flow and database design.*
* *Descriptions of systems, critical subsystems, services, platforms, and relationships among modules.*

*While the Base Specification provides general requirements, user stories, and design goals, the Design Specification adds details that define the specific role of the contribution. It offers enough detailed design information—such as architectures and layouts—to enable end users to begin realizing it in the market. One or more parties may collaborate to develop detailed design specifications. Compared to the Base Specification, this effort typically contains significantly more detail, including future roadmaps and IP-related information. This group may have a multi-party NDA on their own (outside of the OCP umbrella) as part of the normal practice of developing products.*

*Design Specifications can be reused. For example, if collaboration results in a “foo” design specification, another team could reuse much of it to create a “bar” specification that is distinct yet shares common elements and is potentially compatible in various ways. Having the same Base Specification for several Design Specifications helps increase the commonality of interfaces and code, meeting a set of common infrastructure hardware, software, and firmware requirements while allowing for generation-to-generation variations or product differentiation.*

*This corresponds to the Design phase of the SDLC.*

*This document defines the technical details for one of the following types of specifications:*

* *Detailed architecture and data flow/database design for an extensible technology platform or layer, representing technical community-wide consensus and possibly used as a de facto standard.*
* *Requirements and descriptions of systems, critical subsystems, services, platforms, and relationships among modules for a specific solution or implementation.*
* *Extension or modification of an existing specification (state which existing spec is being modified):*
  + *A complete version update.*
  + *A minor revision.*

*Recommendation: Contributors of this specification should make best efforts to provide an output (e.g., a reference design) within 180 days of approval. This cadence helps the community move quickly, although it is only a recommendation.*

## Software Product Specification Layer

*The Software Product Specification captures manufacturing requirements, including all design and build files, building on the Design Specification. This layer contains what is commonly referred to as the Low-Level Design (LLD), where detailed system components and their interactions are specified. The Product Specification offers a detailed description of each module and includes actual logic for every system component, being very specific with module-level specifications. It is also known as micro-level or detailed design.*

*The HLD information is synthesized into a more detailed blueprint, addressing specific algorithms, data structures, and interfaces. The LLD serves as a guide for developers during coding, ensuring the accurate and efficient implementation of the system’s functionality.*

*Typically, fewer companies will engage to create a single Product Specification, but the goal is to increase the total number of products that meet a Design Specification (derived from a Base Specification). The resulting Product Specification should be contributed to OCP via a Final Specification Agreement (FSA). A product typically undergoes significant effort for qualification and mass-production readiness beyond what is specified in a typical design specification.*

*Product Specifications can be reused. For instance, assuming the base and design specifications allow, if one contributor creates a specific solution with a generic ISA (Instruction Set Architecture) in the design specification, the same or another team could reuse it to make a solution with a specific CPU ISA in the product specification.*

*At the productization phase, even fewer companies may be involved in developing a specific final product for contribution to OCP. Typically, for a hardware product, this layer of specification may be submitted to OCP for “OCP Accepted™” or “OCP Inspired™” designation (with different levels of collateral such as a Design Package). For software, this opportunity exists but is not prescribed.*

*This corresponds to the Implementation and partially the Maintenance phases of the SDLC.*

*This document defines the technical details for one of the following types of specifications:*

* *Product Specification with a detailed description of each module, including the actual logic for every system component, being very specific with module-level specifications.*
* *Modification of an existing product specification (state which existing spec is being modified):*
  + *A complete version update.*
  + *A minor revision.*

*Recommendation: Contributors of this specification should make best efforts to provide an output (e.g., a reference design) within 120 days of approval. This recommended cadence helps the community move quickly, although it is only a recommendation.*

*Note: Any supplier seeking OCP recognition—such as OCP Accepted™—must use a product specification that is 100% compliant with the preceding Base requirements, Design Specification, and these Product Specifications as described.*

| DELETE THIS BLOCK BEFORE SUBMITTING |
| --- |
| INSTRUCTIONS FOR THE FOLLOWING SECTIONS:   * The following (Sections 5-8 are required to document features and functions of the contributed system, subsystem, platform, card, component or other unit as appropriate- and are broken out into layers. * The layout of the sections has some boilerplate and random examples. These are illustrative only. Please rearrange, add, delete, and change as necessary to describe the contribution. * Be sure to complete only the incremental requirements for each subsequent layer. (ex: Assume Base Specification and include only changes in the Product Specifications, etc…) * Please use the OCP Terminology Guidelines for Inclusion and Openness. * No NDA (Non-disclosure Agreement) or confidential material is to be included in this document, including charts and included materials. This will be an OPEN document. |

# 

# Overview

Describe your contribution and the modularity of this spec within the framework of modular specification process (this might be the openness tenet too) Include the problems it addresses. Explain its utility within the Open Compute Project ecosystem.

# 

# Base Specifications

## Description

*Provide a high-level overview of the software, including its key features, intended users, and any assumptions made during development.*

## Goals and Vision

*List the goals of the software and any vision statements. This helps consumers of the specification understand the problem being addressed and how the solution fits into the broader context.*

## User Requirements/User Stories

*Include all end-user requirements and user stories that describe what the software is supposed to do. Note that users may include other subsystems or platforms.*

## Solution Architecture

Provide a high-level view of the software’s elements, interfaces, and other significant components. Support this section with a diagram that visually represents the structure and relationships between various parts, illustrating how they interact and function together.

## Functional Requirements

*List the key functions the software must perform, detailing how users interact with it and how the system responds. Include considerations for manageability, resilience, and availability.*

## Non-Functional Requirements

*Describe qualities such as performance, usability, security, and maintainability that the software must exhibit.*

## Features

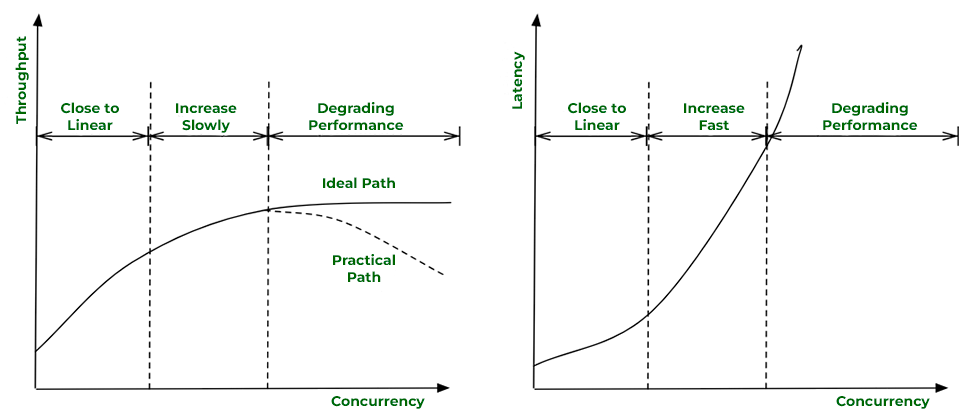
*Break down the software’s specific features, explaining the purpose and functionality of each one.*

## Interface Requirements

*Explain how the software will interact with other systems or users, including hardware interfaces, software APIs, or communication protocols.*

## Performance Requirements

*Set expectations for the system’s speed and efficiency, including resource consumption targets and scalability considerations.*



Performance Example

## Security Requirements

*Detail the measures needed to keep the software and end users secure from errors, cyber-attacks, or data breaches. Briefly describe the security functionalities that your specification requires and recommends. Include a “required by” date for any recommendations.*

*Note: Omit any items that do not apply and add any that are missing. Remember, the more detailed this specification is, the less flexibility is allowed during the design and product specification phases.*

## Constraints

*List any limitations the software must account for, such as hardware constraints, legal regulations, or specific tools that the OCP community must consider.*

*Document any hardware management implementations either dependent on or to your contribution, including firmware (BIOS), optional Board Management Controller (BMC), Data Center Secure Control Modules (DC-SCM), etc.*

## Test and Validation Requirements

*Provide the concepts for testing and validation of this solution.*

## Compliance

| DELETE THIS BLOCK BEFORE SUBMITTING |
| --- |
| INSTRUCTIONS FOR THE FOLLOWING SECTIONS:   * This mandatory section serves as a checklist for consumers of the specification to ensure compliance with the requirements. It applies to any implementations—whether listed on the OCP Marketplace or not—that declare compliance with this specification. * For a Base Specification, this section can be concise. Consider the minimum requirements needed for compliance and leave specific details to subsequent specifications. Avoid introducing requirements that could be easily overridden in higher-level specifications to prevent confusion or lack of purpose. * While it is possible to mandate full compliance with all items in this specification, that is not the intent of the Base Specification. |

*This* ***mandatory*** *section is also applicable to create the checklist for the consumers of the specification to adhere to in order for them to declare it complies to the requirements. This applies to any implementations (OCP Marketplace or not) that declare compliance with this specification.*

*(Example Compliance Table Follows)*

| ***ITEM*** | ***REQUIREMENT*** | ***REFERENCE*** | ***MANDATORY*** |
| --- | --- | --- | --- |
| *1* | *Complies to Special Publication Secure Software Development Framework (SSDF) Version 1.1: Recommendations for Mitigating the Risk of Software Vulnerabilities.* | *(SP) 800-218* | *Y* |
| *2* | *Optional: The device shall facilitate clearing the Timestamp Origin field in the Timestamp (Feature*  *Identifier 0Eh)* | *NVMe-OPT-4* | *N* |

## 

## Software Support (*Strongly* recommended)

*Identify any external software required to support the contribution, including notional architecture and necessary features. Provide information about the OCP GitHub repository where the project is hosted.*

## Repository Location

*It is highly recommended that OCP projects participate in a collaborative development process. OCP provides GitHub resources—with access control if needed—for this purpose. Please request a repository for your contribution from your project leads.*

*Include the OCP GitHub repository information here.*

## Prescribed Materials

*List any prescribed materials included in your contribution, such as specific components that are referenced but not contributed.*

*This section may include, but is not limited to:*

* *Disallowed components.*
* *Specifically required components with no substitution allowed.*

## References (recommended)

[1] “Title”, publication year, OCP specification, version, link to publication if available.

[2] “Title”, publication year, publication journal/conference/standard, volume, pages, link to publication if available.

# 

# Design Specifications

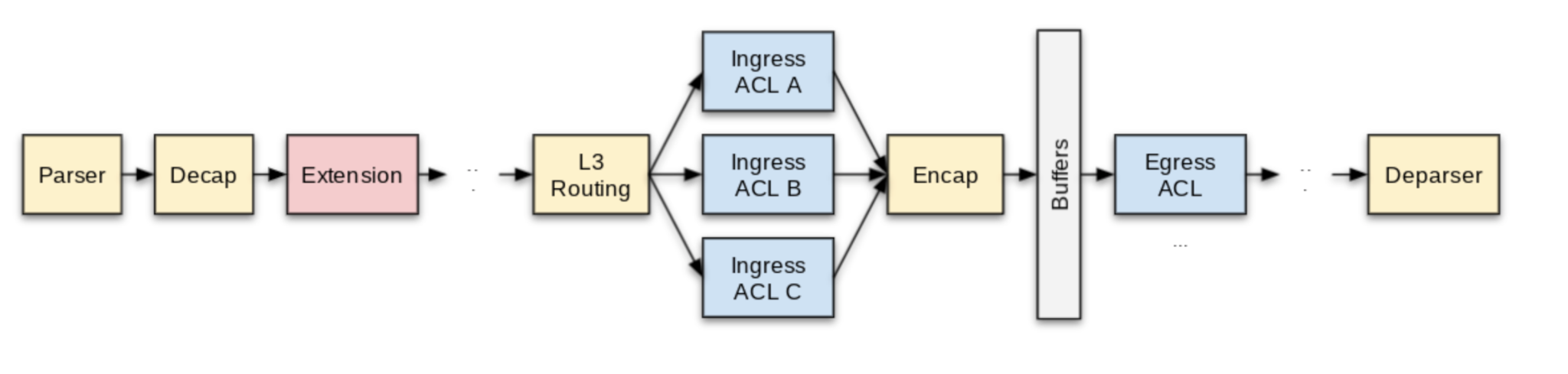
*Note to authors: This section refines the previous specification. It’s unnecessary to repeat earlier sections verbatim; only include additions or modifications that refine the previous content. Treat this section as a Software High-Level Design (HLD) document.*

## Description

*Provide a high-level overview of the problem and the solution your software offers, including key features, intended users, and any assumptions made during development.*

## Solution Architecture

* + 1. *Overview: Describe the structure and relationships between various components of the solution. Include a visual diagram to illustrate how different parts interact and function together.*
    2. *Technology Stack: Detail the programming languages, frameworks, databases, and other technologies required for development.*
    3. *Deployment Architecture: Include any deployment-related architecture, such as fleet management and external resources.*

**

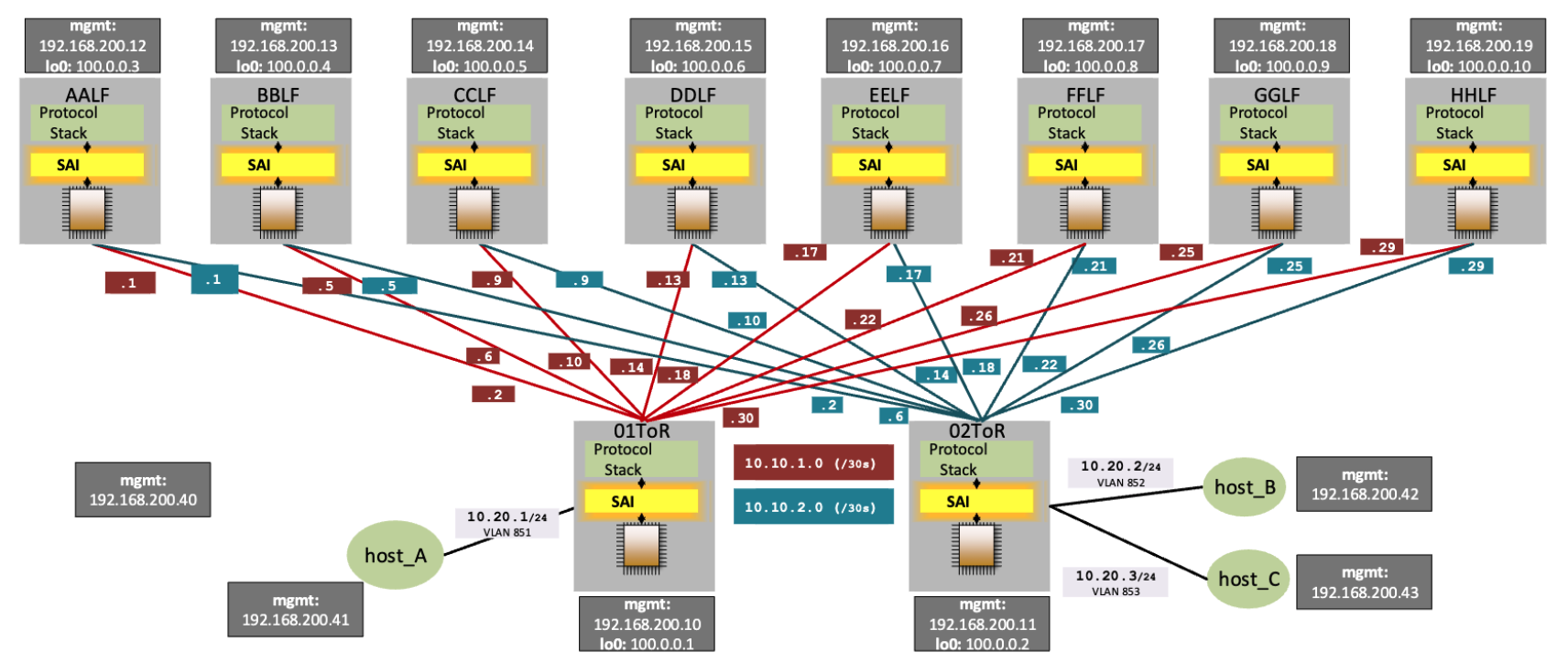
Sample Block Diagram

* + 1. *Further Considerations:*
  + *Use of specific products (programming languages, databases, libraries).*
  + *Reuse of existing software components for implementing features.*
  + *Future plans for extending or enhancing the software.*
  + *User interface paradigms or system input/output models.*
  + *Hardware and software interface paradigms.*
  + *Error detection and recovery mechanisms.*
  + *Memory management policies.*
  + *External databases or data storage management and persistence.*
  + *Distributed data or control over a network.*
  + *General approaches to control.*
  + *Concurrency and synchronization.*
  + *Communication mechanisms.*
  + *Resource management.*

## Modules and Components

*Break down the system into modules or components, outlining each one’s specific roles and responsibilities. This helps stakeholders understand the overall structure and functionality.*

*This helps stakeholders understand the overall structure and functionality of the system.*

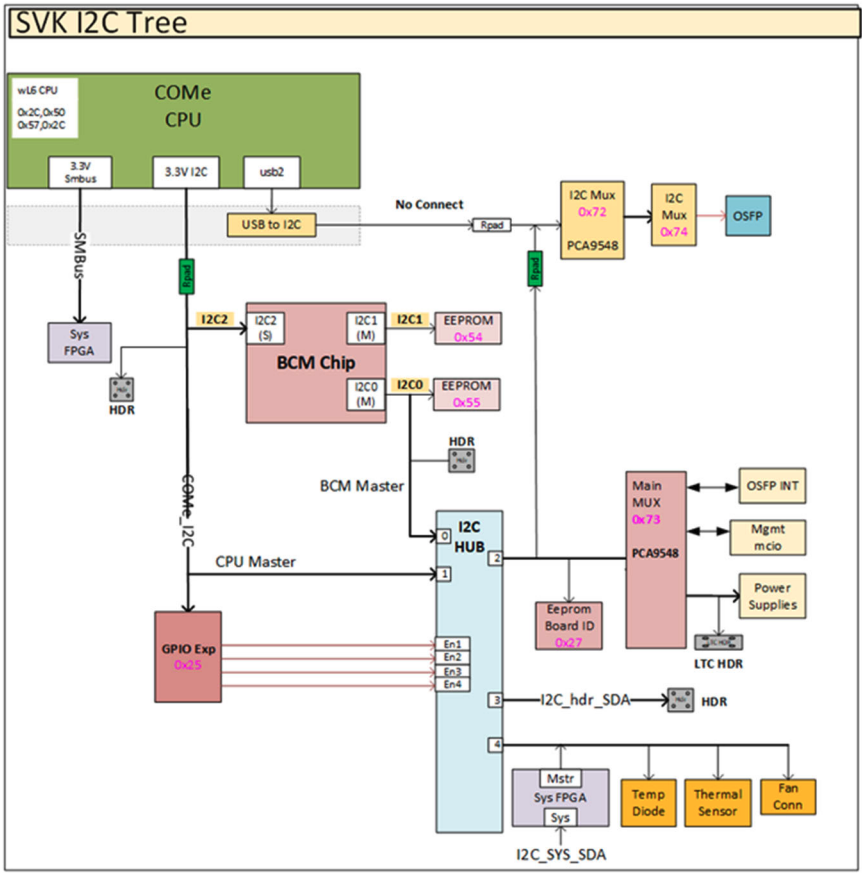
**

Sample Modules and Components Diagram

## Interfaces

*Focus on how different modules communicate with one another. Detail the Application Programming Interfaces (APIs) and any external or user interfaces necessary for seamless interaction between components.*

* + 1. *I/O System: Describe the input/output system of the contribution, clearly delineating the control and data planes.*



Sample Block Diagram Showing Interfaces

*Please describe the I/O System of the contribution, be sure to delineate the control and data planes. Block diagrams here.*

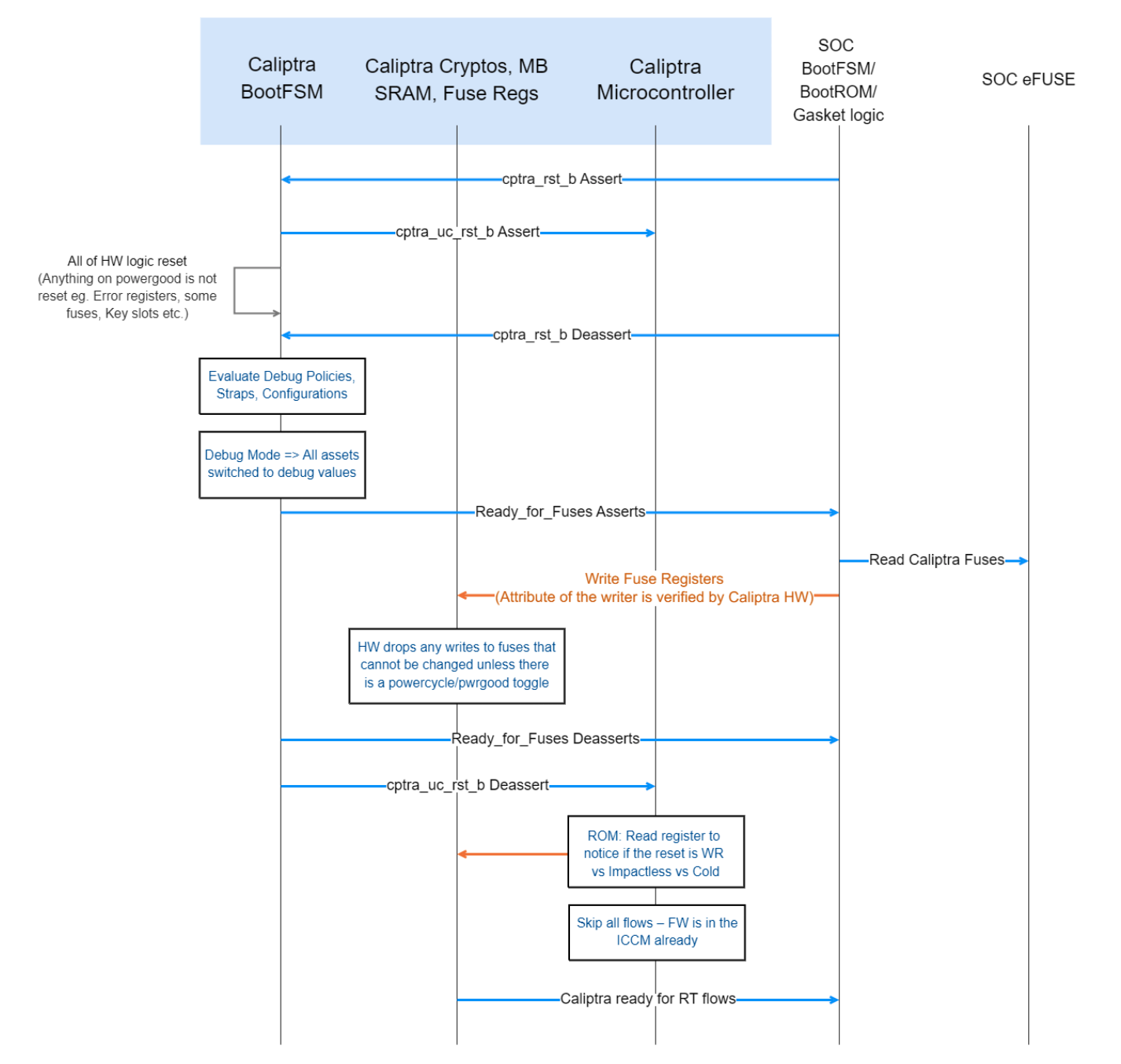
**

Example: Dual-Band RRU External Interface Requirements

## Control and Data Flow

* + 1. Control Flow

*Provide an overview of how control is managed and transferred between different components and modules. Include diagrams or descriptions of pipelines, message queues, event-driven architectures, or other processing mechanisms.*

**

Sample Control Flow Diagram

* + 1. Data Flow

*Present Data Flow Diagrams (DFDs) and descriptions that illustrate how data moves within the solution. This helps consumers understand how information is processed and handled.*

## External Dependencies

*Identify external dependencies, including third-party libraries, frameworks, APIs, and services the system relies on. Discuss licensing, compatibility, and versioning considerations.*

* + 1. *Specific Requirements: For example, processor ISA, bus speed, bandwidth.*

**

Example: Dual-Band RRU External Requirements

## System

*Document external dependencies such as operating systems, firmware functions, necessary features, licensing and distribution rights, ownership rights, system build utilities, test regimes, standards compliance, options for changing firmware configurations, and firmware upgrade processes.*

## Scalability

*Discuss scalability and performance requirements, constraints, and considerations. Include topics like horizontal and vertical scaling, load balancing, caching, and optimization techniques.*

* 1. **Security**

*Briefly describe the security functionalities that this specification requires and recommends. Include a “required by” milestone or date for recommendations.*

*Note to authors: This section can include, but is not limited to:*

*Cryptography:*

* *Required algorithms, modes, strengths, and usage.*
* *Compliance with national or international standards.*
* *Acceptable sources of entropy.*
* *Certifications for algorithm implementations.*
* *Recommended certifications for cryptographic modules.*
* *Safeguards against cryptanalysis by quantum computers.*

*Secure Boot:*

* *Required flow starting from hardware root(s) of trust.*
* *Required measurements from hardware reset through firmware.*
* *Required attestation protocols.*

*Key Management:*

* *Environments and processes for provisioning keys and device secrets.*
* *Processes for identifying CVEs and distributing field updates.*
* *Secure Boot and Attestation key lifecycle management (from generation through revocation).*

*Standards and Recovery:*

* *Recommended standards for Software Bills of Materials (SBOM).*
* *Recommended firmware recovery mechanisms.*

*Definitions:*

* *Required: Mandatory now.*
* *Acceptable: Mandatory now, chosen from acceptable alternatives.*
* *Recommended: Suggested now, but mandatory by a specified future date.*

*Please find guidance and examples in the OCP Security Project documents on* [*Secure Boot*](https://www.opencompute.org/documents/secure-boot-2-pdf)*,* [*Attestation of System Components*](https://www.opencompute.org/documents/attestation-v1-0-20201104-pdf)*,* [*Common Security Threats*](https://www.opencompute.org/documents/common-security-threats-notes-1-pdf)*, and the* [*CSIS document on Secure Firmware Development Best Practices*](https://www.opencompute.org/documents/csis-firmware-security-best-practices-position-paper-version-1-0-pdf)*.*

*All products seeking OCP Inspired™ or OCP Accepted™ Product Recognition shall have a completed Security Profile in the latest Supplier Requirements Checklist. Whether the answer is a yes or no, the profile must be completed. For the base specification, a statement about the intention for the contribution to comply with OCP Inspired™ or OCP Accepted™ Product Recognition requirements.*

## Management

*Document the manageability implementation of your contribution.*

* 1. **Validation**

*Provide high-level concepts on how to test and validate this solution.*

## Compliance

| DELETE THIS BLOCK BEFORE SUBMITTING |
| --- |
| INSTRUCTIONS FOR THE FOLLOWING SECTIONS:   * This mandatory section serves as a checklist for consumers of the specification to ensure compliance with the requirements. It applies to any implementations—whether listed on the OCP Marketplace or not—that declare compliance with this specification. * For a Base Specification, this section can be concise. Consider the minimum requirements needed for compliance and leave specific details to subsequent specifications. Avoid introducing requirements that could be easily overridden in higher-level specifications to prevent confusion or lack of purpose. * While it is possible to mandate full compliance with all items in this specification, that is not the intent of the Base Specification. |

*(Example Compliance Table Follows)*

| ***ITEM*** | ***REQUIREMENT*** | ***REFERENCE*** | ***MANDATORY*** |
| --- | --- | --- | --- |
| *1* | *Complies to Special Publication (SP) 800-218, Secure Software Development Framework (SSDF) Version 1.1: Recommendations for Mitigating the Risk of Software Vulnerabilities.* | *10.3.2 (multiple items)* | *Y* |
| *2* | *Optional: The device shall only clear the Timestamp Origin field to 000b in the Timestamp (Feature*  *Identifier 0Eh) on a main power cycle or NVM Subsystem Reset (e.g., NSSR). The device*  *shall not clear the Timestamp Origin field on a power cycle of only AUX power.* | *NVMe-OPT-4* | *N* |
| *3* | *Any supplier seeking OCP recognition for a hardware product dependent on this specification*  *shall be 100% compliant with the requirements X, Y, Z* | *4* | *Y* |

## Software Support (*Strongly* recommended)

*Identify any external software required to support the contribution, including notional architecture and necessary features. Provide information about the OCP GitHub repository where the project is hosted.*

## Repository Location

*It’s highly recommended that OCP projects participate in a collaborative development process. OCP provides GitHub resources—with access control if needed—for this purpose. Please request a repository for your contribution from your project leads.*

*Include the OCP GitHub repository information here.*

## Prescribed Materials

*List any prescribed materials included in your contribution, such as specific components that are referenced but not contributed.*

*This section may include, but is not limited to:*

* *Disallowed components.*
* *Specifically required components with no substitution allowed.*

## References (recommended)

[1] “Title”, publication year, OCP specification, version, link to publication if available.

[2] “Title”, publication year, publication journal/conference/standard, volume, pages, link to publication if available.

# Product Specifications

*Note to authors: This section refines previous sections. It’s unnecessary to repeat earlier content; only include additions or modifications. This section requires further detail, such as bills of materials with component part numbers, supporting design files, software, tools, and any other files required to produce the contribution.*

*Treat this section as a Software Low-Level Design (LLD) document.*

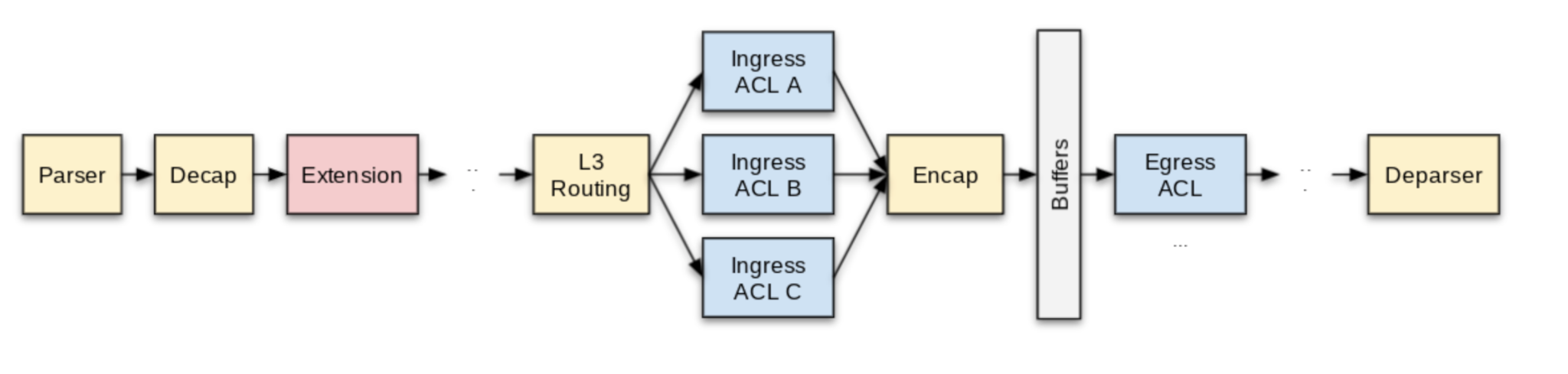
## Description

*Provide a detailed description of the problem and the solution your software offers, including key features, intended users, and any relevant assumptions.*

## Solution Architecture

* + 1. *Detailed Overview: Offer exact details of the solution, including the structure and relationships between components. Describe the required technology stack, specifying programming languages, frameworks, and databases.*

*It helps to visually represent how different parts interact and function.*

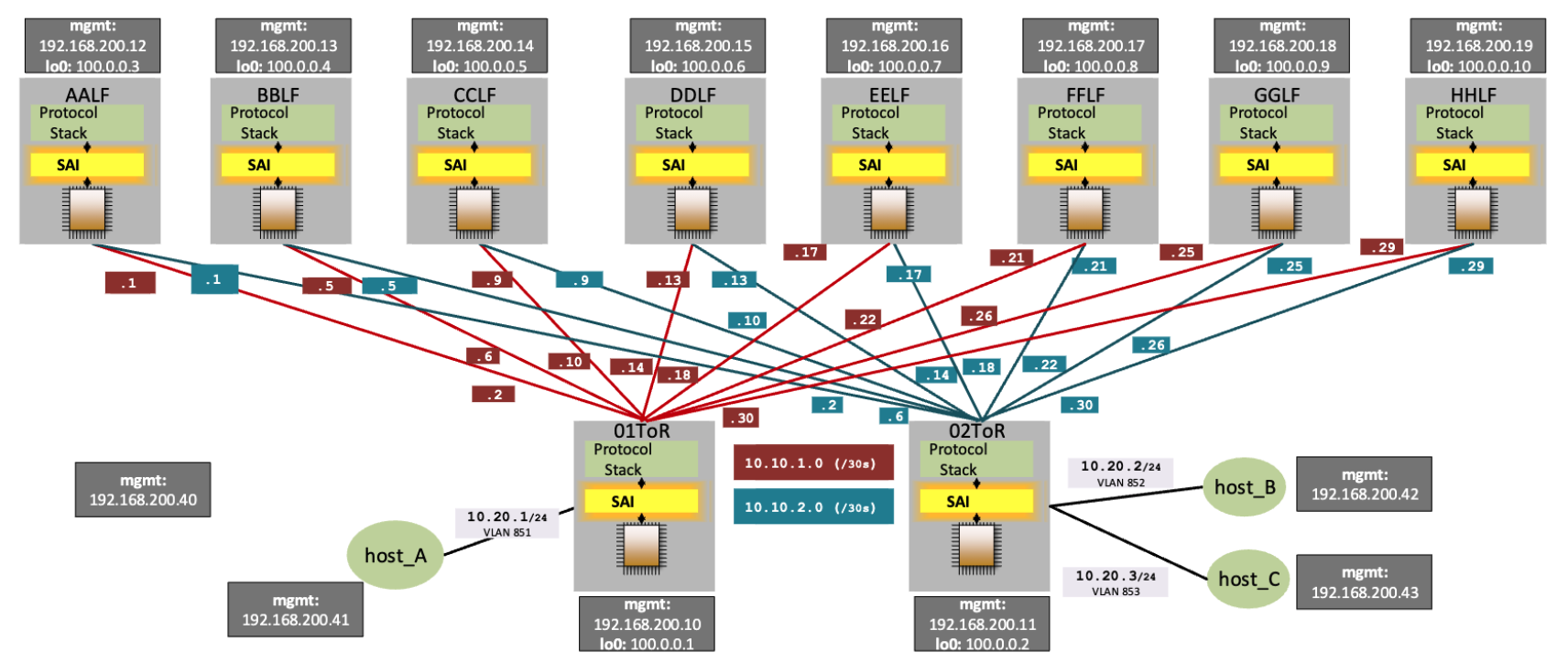
**

Sample Block Diagram

* + 1. *Deployment Overview:Provide detailed deployment architecture, including fleet management and external resources.*

## Modules and Components

*Break down the system into atomic levels of modules or components, outlining each one’s specific roles and responsibilities. This level of detail helps stakeholders understand the intricate workings of the system.*

**

Sample Modules and Components Diagram

## Interfaces

*Detail how different modules communicate with one another, including APIs and any external or user interfaces necessary for seamless interaction.*

**

Example: Dual-Band RRU External Interface Requirements

* + 1. *I/O System: Provide detailed descriptions of the input/output system, clearly delineating the control and data planes.*

**

Sample Block Diagram Showing Interfaces

## Signal List

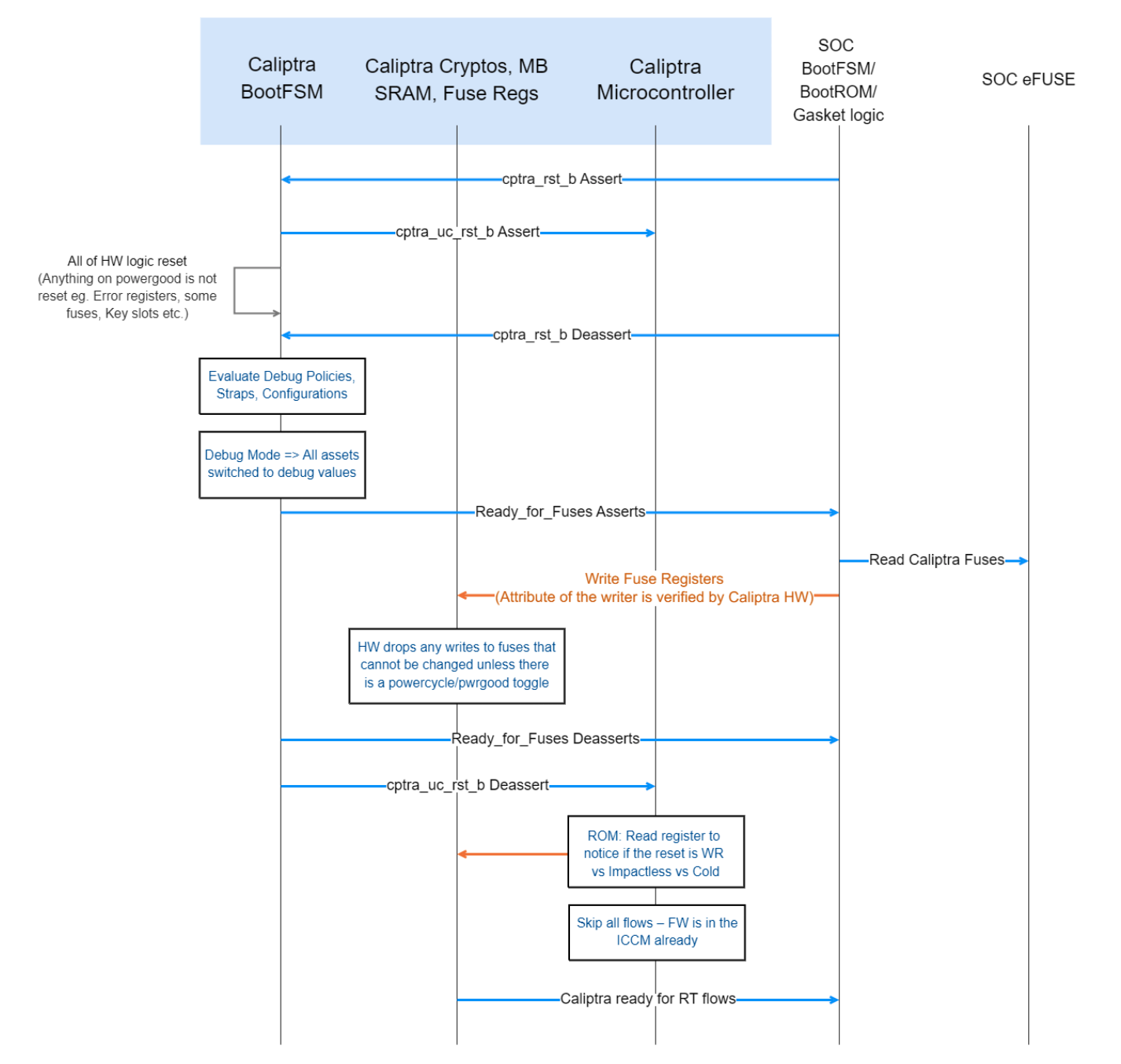
*Note to author of this specification: Examples only*

* *IEEE 802.3 Ethernet*
* *Synchronization/Clocks*
* *IEEE PTM*
* *PCIe*
* *i2C/i3C*
* *GPIO*
* *USB 3.0*

## Control and Data Flow

* + 1. Control Flow

*Offer a comprehensive overview of how control is managed and transferred between components and modules. Include detailed diagrams or descriptions of pipelines, message queues, event-driven architectures, or other data processing mechanisms.*

**

Sample Control Flow Diagram

* + 1. Data Flow

*Present detailed Data Flow Diagrams (DFDs) and descriptions that demonstrate data movement within the solution, aiding consumers in understanding data processing and handling.*

## External Dependencies

*Identify all external dependencies, including third-party libraries, frameworks, APIs, and services. Discuss licensing, compatibility, and versioning, and provide specific requirements (e.g., processor ISA, bus speed, bandwidth).*

*Please describe specific requirements. Example: processor ISA, bus speed, bandwidth etc…*

**

Example: Dual-Band RRU External Requirements

## System

*Document all external dependencies such as operating systems, firmware functions, necessary features, licensing and distribution rights, ownership rights, system build utilities, test regimes, standards compliance, options for changing firmware configurations, and firmware upgrade processes.*

## Scalability

*Discuss detailed scalability and performance requirements, constraints, and considerations. Include in-depth discussions on horizontal and vertical scaling, load balancing, caching, and optimization techniques.*

* 1. **Security**

*Provide a comprehensive description of the security functionalities that this specification requires and recommends. Include “required by” milestones or dates for any recommendations.*

*Note to authors: This section can include, but is not limited to:*

*Cryptography:*

* *Required algorithms, modes, strengths, and usage.*
* *Compliance with national or international standards.*
* *Acceptable sources of entropy.*
* *Certifications for algorithm implementations.*
* *Recommended certifications for cryptographic modules.*
* *Safeguards against cryptanalysis by quantum computers.*

*Secure Boot:*

* *Required flow starting from hardware root(s) of trust.*
* *Required measurements from hardware reset through firmware.*
* *Required attestation protocols.*

*Key Management:*

* *Environments and processes for provisioning keys and device secrets.*
* *Processes for identifying CVEs and distributing field updates.*
* *Secure Boot and Attestation key lifecycle management (from generation through revocation).*

*Standards and Recovery:*

* *Recommended standards for Software Bills of Materials (SBOM).*
* *Recommended firmware recovery mechanisms.*

*Definitions:*

* *Required: Mandatory now.*
* *Acceptable: Mandatory now, chosen from acceptable alternatives.*
* *Recommended: Suggested now, but mandatory by a specified future date*

*Please find guidance and examples in the OCP Security Project documents on* [*Secure Boot*](https://www.opencompute.org/documents/secure-boot-2-pdf)*,* [*Attestation of System Components*](https://www.opencompute.org/documents/attestation-v1-0-20201104-pdf)*,* [*Common Security Threats*](https://www.opencompute.org/documents/common-security-threats-notes-1-pdf)*, and the* [*CSIS document on Secure Firmware Development Best Practices*](https://www.opencompute.org/documents/csis-firmware-security-best-practices-position-paper-version-1-0-pdf)*.*

*All products seeking OCP Inspired™ or OCP Accepted™ Product Recognition shall have a completed Security Profile in the latest Supplier Requirements Checklist. Whether the answer is a yes or no, the profile must be completed. For the base specification, a statement about the intention for the contribution to comply with OCP Inspired™ or OCP Accepted™ Product Recognition requirements.*

## Management

*Document the manageability implementation of your contribution in detail.*

* 1. **Validation**

*List the tests and provide detailed procedures on how to test and validate this solution.*

## Compliance

| DELETE THIS BLOCK BEFORE SUBMITTING |
| --- |
| INSTRUCTIONS FOR THE FOLLOWING SECTIONS:   * This mandatory section serves as a checklist for consumers of the specification to ensure compliance with the requirements. It applies to any implementations—whether listed on the OCP Marketplace or not—that declare compliance with this specification. * For a Base Specification, this section can be concise. Consider the minimum requirements needed for compliance and leave specific details to subsequent specifications. Avoid introducing requirements that could be easily overridden in higher-level specifications to prevent confusion or lack of purpose. * While it is possible to mandate full compliance with all items in this specification, that is not the intent of the Base Specification. |

*(Example Compliance Table Follows)*

| ***ITEM*** | ***REQUIREMENT*** | ***REFERENCE*** | ***MANDATORY*** |
| --- | --- | --- | --- |
| *1* | *Complies to Special Publication (SP) 800-218, Secure Software Development Framework (SSDF) Version 1.1: Recommendations for Mitigating the Risk of Software Vulnerabilities.* | *10.3.2 (multiple items)* | *Y* |
| *2* | *Optional: The device shall only clear the Timestamp Origin field to 000b in the Timestamp (Feature*  *Identifier 0Eh) on a main power cycle or NVM Subsystem Reset (e.g., NSSR). The device*  *shall not clear the Timestamp Origin field on a power cycle of only AUX power.* | *NVMe-OPT-4* | *N* |
| *3* | *Any supplier seeking OCP recognition for a hardware product dependent on this specification*  *shall be 100% compliant with the requirements X, Y, Z* | *4* | *Y* |

## Software Support (*Strongly* recommended)

*Identify any external software required to support the contribution, including detailed architecture and necessary features. Provide information about the OCP GitHub repository where the project is hosted.*

## Repository Location

*It is highly recommended that OCP projects participate in a collaborative development process. OCP provides GitHub resources—with access control if needed—for this purpose. Please request a repository for your contribution from your project leads.*

*Include the OCP GitHub repository information here.*

## Prescribed Materials

*List any prescribed materials included in your contribution, such as specific components that are referenced but not contributed.*

*This section may include, but is not limited to:*

* *Disallowed components.*
* *Specifically required components with no substitution allowed.*

## References (recommended)

[1] “Title”, publication year, OCP specification, version, link to publication if available.

[2] “Title”, publication year, publication journal/conference/standard, volume, pages, link to publication if available.

# Appendix A - Checklist for Steering Committee (SC) approval of this Specification (to be completed by contributor(s) of this Spec)

Complete all the checklist items in the table with links to the section where it is described in this spec or an external document .

| **Item** | **Status or Details** | **Link to detailed explanation** |
| --- | --- | --- |
| Is this contribution entered into the OCP Contribution Portal? | Yes or No | If no, please state the reason. |
| Was it approved in the OCP Contribution Portal? | Yes or No | If no, please state the reason. |
| Is there a Supplier(s) that is building a product based on this Spec? (Supplier must be an OCP Solution Provider) | Yes or No | List Supplier Name(s) |
| Will Supplier(s) have the product available for GENERAL AVAILABILITY within 120 days? | Yes or No | If more time is required, please state the timeline and reason for extension request.  Please have each Supplier fill out Appendix B. |

# 

# Appendix B-\_\_ <supplier name> - OCP Supplier Information and Hardware Product Recognition Checklist

(to be provided by each supplier seeking OCP recognition for a Hardware Product based on this specification)

Company:

Contact Info:

Product Name:

Product SKU#:

Link to Product Landing Page:

The following is needed for OCP hardware product recognition:

**For OCP Inspired™**

* All Suppliers must be an OCP Member. All corporate membership levels are eligible.
* Declare product is 100% compliant with specification
* Complete the [OCP Inspired™ Product Recognition Checklist](https://docs.google.com/spreadsheets/d/1p7g_bPWzgXDDTkxbOEOkLrbvfKmqVWspKOi7J20yJcE/copy?resourcekey=0-UWRTqqnBa3i6BcSNTDJfmA#gid=963873675), which includes hardware management conformance checks and security profile.

**For OCP Accepted™**

* All Suppliers must be an OCP Member. All corporate membership levels are eligible.
* Complete the [OCP Accepted™ Product Recognition Checklist](https://docs.google.com/spreadsheets/d/1SNqQYCta4CVsZsZcRRVR5A779YyCHxA2gLSINlFtnTs/copy#gid=963873675), which includes hardware management conformance checks, security profile and open system firmware conformance checks.
* Submit a design package meeting [OCP Hardware Design Guideline Contribution Checklist](https://docs.google.com/document/u/0/d/1SdLlXxn_jz__t8I33ATraYvHDYX3go3w_rR4LJ1PNTE/edit) (if not already submitted by the contributor). If already submitted, declare the product is 100% compliant with the design package.
* Submit a firmware package including a firmware image, build scripts, documentation, test results and a tool that verifies modifications
* Submit the BMC source code, if applicable to product type

Please complete the OCP Inspired™ Product Recognition Submission Checklist or OCP Accepted™ Product Recognition Checklist and the following table.

| **Item** | **Details** | **Links** |
| --- | --- | --- |
| Which product recognition? | OCP Accepted™ or OCP Inspired™ | Provide link for the appropriate Product Checklist |
| If OCP Accepted™, who provided the Design Package? |  | Link to OCP Contribution Database |
| Where can a potential adopter purchase the product? |  | Link to OCP Marketplace |

# 

# Appendix C - Contribution Process FAQs

As a contributor to a hardware specification, here are some questions that often come up.

1. What are acceptable types of hardware specification am I able to contribute to OCP? Is it any of the below?
   1. base, design or product specification for a de-facto standard (new standard with no hardware product on the horizon)
   2. base, design or product specification for an intended physical <hardware product type> (product may be coming but within the next 12-15 months for base and design specification, with a product specification with design files resulting in a product in 3-6 months)
   3. modification of an existing specification (state which existing spec is being modified)
      1. either a complete revision update or
      2. a minor version update
      3. Note: errata does not require a specification update. Only an errata statement.
   4. an architectural specification for <product types> that may incorporate several/multiple specifications, types, standards and other components into a logical system for given purposes.
      1. this may include a/any specification(s) that strictly conforms to an architectural specification
      2. this may include a/any specification(s) that is/are standalone or conforms to an architectural specification
   5. A Base, Design and Product Specification in a single document.
   6. *If none of the above, please contact OCP Staff for consultation.*
2. How do I know if what I am contributing will be accepted by OCP?
   1. Before contributing any specifications, please contact either OCP Staff (Michael Schill, Rob Coyle or Bijan Nowroozi) or the Project Lead for the Project that best represents your contribution. For example, if you are contributing a Server Specification, please contact one of the Server Project Leads. You can see all the Projects [here](https://www.opencompute.org/projects).
   2. The OCP Foundation, Project Leadership and OCP Community are resources to help you navigate the process.
3. What is the contribution process for my hardware spec?
   1. Follow the flow for your spec type [here](https://docs.google.com/presentation/d/1PlXGLhCdgVEGWQ0hLYdAQEH5qCScwYij/edit#slide=id.g10e20dc1292_0_101).
   2. This flow is subject to change so please check with the OCP Staff for more information or any questions.
4. What if my spec is not developed yet and I want to collaborate with other companies?
   1. Please contact either OCP Staff (Michael Schill, Rob Coyle or Bijan Nowroozi) or the Project Lead for the Project that best represents your contribution.
   2. They will help you find other collaborators and help you with the contribution process for a multi-party contribution.
5. I have a question on the Contribution License Agreement.
   1. Please contact OCP Staff and we can help you with questions.
6. Do I need to have a product in order to contribute a spec?
   1. Please see Q1. Some types of specs do not require an immediate product. Some do. Please work with the OCP Staff on better direction on your specification type.
7. How do I know if someone else is already working on this idea?
   1. Please discuss with the Project Leadership and/or the OCP Foundation staff.
8. What other considerations are there?
   1. The OCP Foundation encourages contributors to consult with the Project Leadership and/or the OCP Foundation staff as early as possible in the development cycle. There may be others in the community who are willing and able to help share some of the development effort.
   2. It may be desirable for speed to gather key collaborators and work together outside of public meetings, with a closed CLA. Just be aware of the tradeoffs such as potential collaboration is missed and the idea may not be as strong as it could be with additional eyes on it. Some recommendations are:
      1. even if the idea is worked on in a closed CLA group, find ways to hold regular public briefings
      2. consider opening the CLA group as soon as possible and
      3. be open to the possibility of duplication of effort by others whom were not aware and chance this approach will end up fracturing of the potential total market by having subsequent exclusive groups working in parallel

1. https://en.wikipedia.org/wiki/Systems\_development\_life\_cycle [↑](#footnote-ref-0)