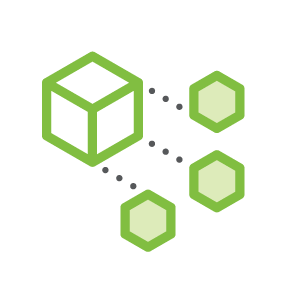
.



<Title: 3 Layer Architectural (Base, Design and Profile) Specification Template>

<Version>.<Revision>.<Errata>

**Modular Architectural Specification**

Effective XXXX, 2024

Author: <Primary>

Author: <Secondary, etc. Delete if unnecessary>

# Version History

| 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.* |

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 NOV 24 | 1.0-1.3 | Bijan Nowroozi | Created new template, multiple edits |
| 07 NOV 24 | 1.4 | Bijan Nowroozi | Reflected New License, cleaned up instructional text boxes, tenets |
| 11 NOV 24 | 1.5 | Bijan Nowroozi | Reorganized sections, various Edits |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |

# 

# Current Template Version:

**3 Layer (Base, Design and Product) Architectural Specification Template V1.5.0**

Effective November 2024

**Table of Contents**

[**Version History 2**](#_5hqof536mr77)

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

[**1. License 7**](#_grkgzhtimizo)

[1.1. Open Web Foundation (OWF) CLA 7](#_oeo8w428bs5y)

[**1.2. Acknowledgements 8**](#_3yfbtftpm0pp)

[**2. Compliance with OCP Tenets 9**](#_zi3i5v28cisd)

[1.1. Openness 9](#_j6kl13o94r5u)

[1.2. Efficiency 9](#_fj7vgvyr9nfc)

[1.3. Impact 10](#_ulz3hvb2ue1c)

[1.4. Scale 10](#_b5zwsm4bvnuo)

[1.5. Sustainability 10](#_rrf2lq0xcws)

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

[3.1. Introduction to System Architecture 12](#_nrko2uohk4bx)

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

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

[3.5. Profile Specification Layer 16](#_7mpyp541xqk5)

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

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

[5.1. Description 19](#_6z5emikflibt)

[**5.2. Goals, Requirements, Use Cases, Applications, Technical Objectives, Promising Techniques, Design Principles 19**](#_bxd6873vcdcx)

[5.2.1. Goals and Vision 20](#_6aq9dn4wusf6)

[5.2.2. User Requirements/User Stories 20](#_nyqydh83w9f)

[5.2.3. Solution Architecture 20](#_bq5onvu0y20)

[**5.3. Environmental, Regulatory and Compliance and Requirements 21**](#_bpku9xs36228)

[**5.4. Physical/Logical Requirements 21**](#_396jyrpjrclp)

[**5.4.1. Interfaces 21**](#_ipp57zral3zv)

[**5.4.2. Special Mechanical/Thermal Design Requirements 22**](#_nn07wlrr91b)

[**5.5. Scaling: Horizontal and Vertical 22**](#_r5lf8dv68zoq)

[**5.6. Resilience/Redundancy 22**](#_vwzhronp77z2)

[**5.7. Systems Management 23**](#_ug7necm1i98r)

[5.8. Constraints 23](#_3zbegscg774j)

[5.9. Test and Validation Requirements 24](#_eqrp8za7umqi)

[**5.10. Compliance 24**](#_d34ppqie4dsq)

[**5.11. Software Support (Strongly recommended) 25**](#_vqftl4n2kqkf)

[5.12. Repository Location 25](#_z1bfnswi7yqd)

[**5.13. Prescribed Materials 25**](#_u54ppg2adls)

[**5.14. References (recommended) 25**](#_fy8siocl3sx)

[**6. Architectural Design Specifications 26**](#_v6a73aq1zqtb)

[6.1. Description 27](#_9g6tu9lnd7f5)

[**6.2. Requirements, Use Cases, Applications, Technical Objectives, Promising Techniques, Design Principles 27**](#_cb0qclfofqxu)

[6.3. Solution Architecture 27](#_kblf1qvu8gnz)

[6.4. Modules and Components 29](#_8p9398wqf8zu)

[6.5. Interfaces 29](#_1ktvcvrsl4az)

[6.6. Control and Data Flow 31](#_xns4mc2jlxc)

[6.7. External Dependencies 32](#_5tegjrq7eqja)

[6.7.2. System 33](#_biz1b4shqv7r)

[6.8. Scalability 33](#_ld7gu2oyh69b)

[6.10. Management 34](#_wblq38pgacfu)

[6.12. Compliance 35](#_tf51mmeple2b)

[6.13. Software Support (Strongly recommended) 36](#_mox7rh49i141)

[6.14. Repository Location 36](#_k3bnkg9rla0j)

[6.15. Prescribed Materials 36](#_q1tq34t6k104)

[6.16. References (recommended) 37](#_vhr1djkgec9l)

[**7. Profile Specifications 37**](#_8g16s2u7godi)

[7.1. Description 38](#_jxfln14nd47f)

[**7.2. Requirements, Use Cases, Applications, Technical Objectives, Promising Techniques, Design Principles 38**](#_o85lxe3p7e6d)

[7.3. Solution Architecture 39](#_sv2erlflr4l)

[7.4. Modules and Components 41](#_chs9s7fw7mhu)

[7.5. Interfaces 41](#_vi8c5ltbu7o2)

[7.6. Control and Data Flow 43](#_godc7ewk34up)

[7.7. External Dependencies 44](#_sdk2l8nno78c)

[7.7.2. System 45](#_l8xu4drh8tt1)

[7.8. Scalability 45](#_1c9xjofwtfot)

[7.10. Management 46](#_57kk64azx0co)

[7.12. Compliance 47](#_46tkxil8dlug)

[7.13. Software Support (Strongly recommended) 48](#_7rbu2axllxzg)

[7.14. Repository Location 48](#_j5j9qlslwsyu)

[7.15. Prescribed Materials 48](#_l9wdy7k23kd3)

[7.16. References (recommended) 49](#_c150kdvr6trm)

[**Appendix A - Checklist for Steering Committee (SC) approval of this Specification (to be completed by contributor(s) of this Spec) 50**](#_fu98xdxru854)

[**Appendix B-\_\_ <supplier name> - OCP Supplier Information and Hardware Product Recognition Checklist 51**](#_y1ll55eabeef)

[**Appendix C - Contribution Process FAQs 53**](#_4eb5pwm0ic6z)

# 

# License

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

| DELETE THIS BLOCK BEFORE SUBMITTING |
| --- |
| INSTRUCTIONS FOR THE FOLLOWING SECTIONS:   * This scope section contents have been added for author’s information. * Replace the section text (keep titles) with the relevant scoping information intended for this specification document. * The purpose of this template is to define a 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. |

## Introduction to System Architecture

*The system architecture serves as the foundational blueprint outlining the structure, components, and interactions within a software system. It specifies the high-level design decisions and technical framework necessary for the development, deployment, and maintenance of the system. This section provides an overview of the system architecture, detailing its purpose, key components, design principles, and architectural patterns employed.*

* *Purpose*

*The system architecture articulates the overarching design strategy aimed at fulfilling the system’s functional and non-functional requirements. It delineates the structural organization of the system, facilitating a comprehensive understanding of its constituent elements and their interrelationships.*

* *Key Components*

*This section identifies the primary building blocks of the system architecture, encompassing hardware components, software modules, data storage mechanisms, communication protocols, and external interfaces. Each component’s role and contribution to the system’s functionality are explained, establishing a clear understanding of the system’s composition.*

* *Design Principles*

*The design principles underlying the system architecture are outlined to effectively guide the development process. These principles encompass modularity, scalability, flexibility, maintainability, and performance optimization, ensuring that the architecture aligns with the system’s objectives and can adapt to future evolution.*

* *Architectural Patterns*

*The architectural patterns employed within the system architecture—such as client-server, microservices, event-driven, or layered architecture—are described. Each pattern’s suitability to the system’s requirements and its implications on system behavior and performance are discussed, providing insights into the rationale behind architectural choices.*

* *Interactions and Dependencies*

*The interactions among system components and their dependencies are explained to illustrate the flow of data, control, and communication within the system. This includes identifying critical interfaces, protocols, and data exchange formats, ensuring seamless interoperability among system elements.*

* *Constraints and Assumptions*

*Any constraints or assumptions governing the system architecture—such as hardware limitations, regulatory compliance requirements, or technological dependencies—are explicitly stated. Understanding these constraints is essential for mitigating risks and ensuring the feasibility and viability of the proposed architecture.*

* *Evolution and Maintenance*

*Considerations regarding the evolution and maintenance of the system architecture are discussed, addressing aspects such as extensibility, adaptability, and future scalability. Strategies for accommodating changes, incorporating new features, and addressing emerging technological trends are outlined to ensure the longevity and relevance of the architecture.*

*By providing a comprehensive overview of the system architecture, this document serves as a cornerstone for stakeholders, enabling effective collaboration, informed decision-making, and successful system implementation.*

* 1. **Modular Architecture Contribution Pattern**

The general modular contribution hierarchy is illustrated in Figure 1: Modular Contribution Specification Layers.

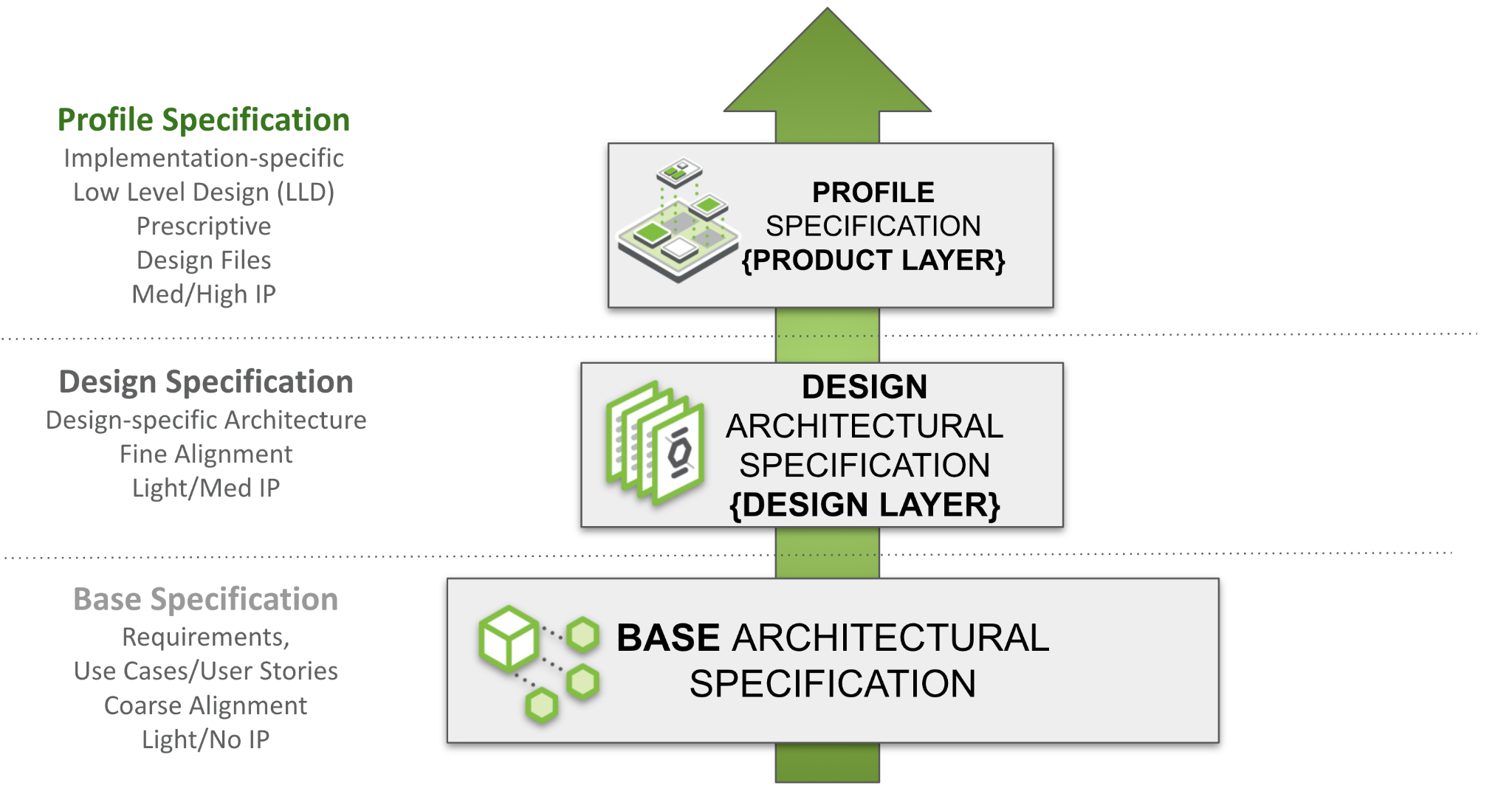


Figure 1: Modular Contribution Specification Layers

*These modular specification templates generally refer to system or subsystem-level design. The purpose is to create a framework for architectural specification, from the high-level or base layer to the very prescriptive product layer known as a Profile Specification. Figure 1 shows the Modular Contribution Specification Layers, and Figure 2 depicts the derivative Architectural Contribution Specification Layers.*

## Base Architectural Specification Layer

*The Base Specification is an architectural framework for coarse alignment—a requirements description enabling flexible hardware and software modules or layers to interoperate. Market requirements drive Base Specifications. Without defining details of a specific design, the Base Specification may have minimal intellectual property content. This structure simplifies the process for multiple parties (including potential competitors) to engage in this phase.*

*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.*

* *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 for a hardware product must be 100% compliant with the system-level Base requirements and subsequent design and product specifications as described. OCP product recognition is not extended to an architectural specification at this time.*

*The defining difference between Hardware System/Subsystem level specifications and this architectural specification is the inclusion of multiple systems into a single architecture or an aggregation of disaggregated subsystems. For example, a mobile wireless core comprises disaggregated subsystems with interfaces between them and even other cores. Together, these subsystems (media servers, databases, policy services, etc.) act as a single system according to transaction-specific rules (e.g., home call, roaming voice call). This definition is outside the scope of a Hardware Base Specification; thus, an Architectural Base Specification serves as a container for these designs. The components may be an aggregation of OCP-only or both OCP and non-OCP specified systems.*

## Architectural Design Specification Layer

The Design Specification captures customer requirements for finer alignment by building on the Base Specification. While the Base Specification provides general requirements and design goals, the Design Specification includes detailed information that further defines the specific role of the contribution and provides sufficient design details—such as high-level schematics, layouts, and flow diagrams—to enable end-users to realize it in the market. One or more parties may collaborate to develop a detailed design specification; they need not be the same entities that worked on the Base Specifications. Compared to the Base Specification, this effort typically contains significantly more detail, including future roadmaps and IP-related information. This group may choose to use an open or closed Contribution License Agreement (CLA) or establish a multi-party NDA independently (outside of the OCP umbrella) in line with their corporate practices.

Design Specifications can be reused. For example, if one contributor creates an indoor design specification, another team could reuse it to make an outdoor specification. Having the same Base Specification for several Design Specifications increases the commonality of physical and logical interfaces to meet common infrastructure hardware/software/firmware requirements while allowing generational variations or product differentiation.

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

* A new Architectural Design Specification for an Architectural Base Specification.
* Modification of an existing specification (state which existing spec is being modified), either:
  + a complete version update, or
  + a minor revision update.
  + A specification with additional detail over the Base Specification for a specific architecture type. This includes expanding an existing specification without changes to the existing specifications.

The components of the architecture may be an aggregation of OCP-only or both OCP and non-OCP specified systems.

## Profile Specification Layer

The Profile Specification captures exact requirements, including specific mandatory support, chosen or available options, all design details, and any build files, typically building on an Architectural Design Specification. Fewer companies may engage to create a single Profile Specification. The goal is to increase the total number of products that meet a Design Specification (derived from a Base Specification) and to expand targeted solutions derived from a common Base/Design level specification. A product or service profile typically undergoes extensive qualification and mass-production/service readiness efforts beyond what is specified in a typical design specification.

Profile Specifications can be reused. For instance, if one contributor creates an architecture based on 480VAC power systems specifications, another team could reuse it to make a -48VDC-based architecture specification, assuming the base and design specifications allow.

A Profile Specification can apply to a single system or subsystem, often serving as a Product Specification. For example, a product or service with a design specification that has multiple configurations or options may have a Profile Specification defining a specific implementation. In this context, the templates for Profile and Product Specification are somewhat interchangeable. The intended purpose of this template is to provide a definitive configuration of Base/Design Architectural specifications.

A product profile (based on a Product Specification) may be submitted to OCP for “OCP Accepted™” or “OCP Inspired™” designation, with different levels of collateral such as a Design Package.

Please see the presentation and recording on the Modular Contribution Process from the OCP Global Summit 2022 for more information.

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

* Product Profile Specification for an intended physical hardware product type.
* Architectural Profile Specification for a specific configuration (intended physical/logical implementation) of an Architectural Design Specification.
* Modification of an existing product specification (state which existing spec is being modified), either:
  + a complete version update, or
  + a minor revision update.
  + A detailed prescriptive specification for a product type, with a product typically available within 120 days of approval of this specification.

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

The components may be an aggregation of OCP-only or both OCP and non-OCP specified systems.

| DELETE THIS BLOCK BEFORE SUBMITTING |
| --- |
| INSTRUCTIONS FOR THE FOLLOWING SECTIONS:   * The following sections 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.

# 

# Architectural Base Specifications

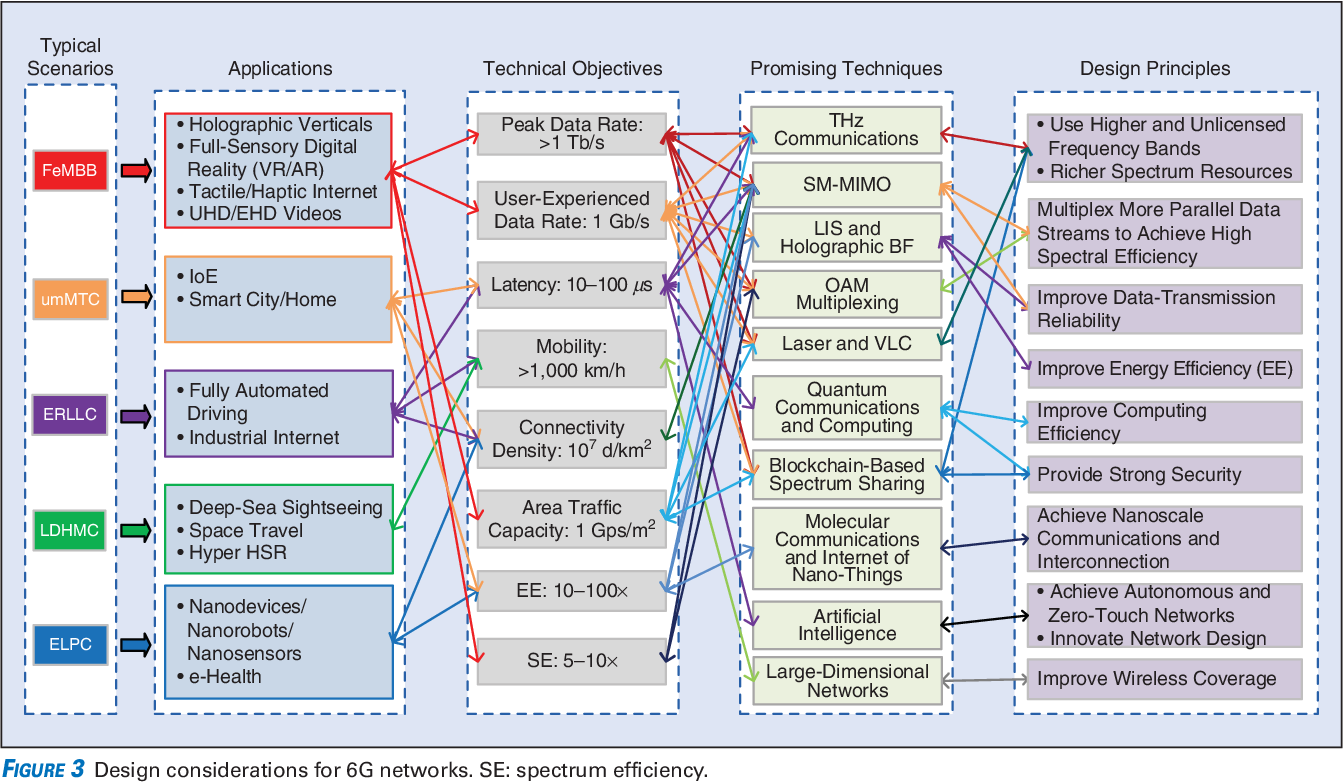
| DELETE THIS BLOCK BEFORE SUBMITTING |
| --- |
| INSTRUCTIONS FOR THE FOLLOWING SECTIONS:   * The following sections are required to document features and functions of the contributed system, subsystem, platform, card, component or other unit as appropriate- and are broken out into layers. * The layout 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. * 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. |

## Description

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

# Goals, Requirements, Use Cases, Applications, Technical Objectives, Promising Techniques, Design Principles

*Please describe the problem that is being addressed and provide supporting detail to help the reader understand the end goals, methods, requirements, etc…*

*\*Source: Zhang, Zhengquan et al. “6G Wireless Networks: Vision, Requirements, Architecture, and Key Technologies.” IEEE Vehicular Technology Magazine 14 (2019): 28-41.*

Sample Requirements Diagram

## 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.*

# Environmental, Regulatory and Compliance and Requirements

*Please describe any environmental regulations or requirements for any platform boards and full system, if applicable..*

*Note to author of this specification: This section can include the following but is not limited to the below items:*

* *UL/CE/NRTL/FCC/IEC/EN/etc Requirements*
* *RoHS/WEE directives, REACH regulations*
* *NEBS compliance requirements*
* *Operating temperature range*
* *Storage temperature range*
* *Transportation temperature range*
* *Shock and Vibration requirements*
* *Operating Altitude*

# Physical/Logical Requirements

*Please describe the physical requirements for your contribution. This may be the limitations of the physical envelope and logical considerations such as functionality, statefulness, clouds, etc…*

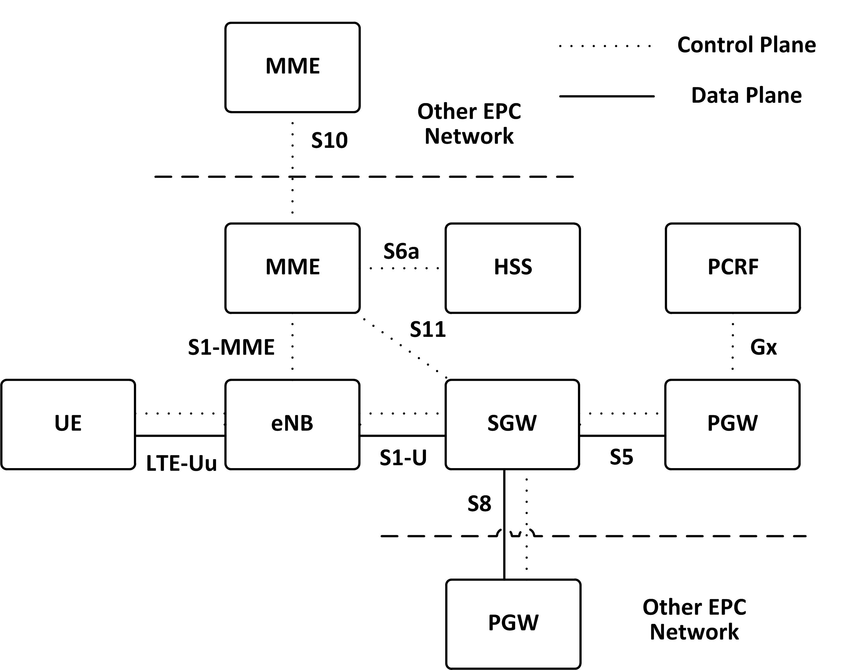
*If this specification defines a chassis type system, be sure to include the description of the chassis and associated modules, midplane, backplanes etc….*

*Note to author of this specification: This section can include the following but is not limited to the below items:*

* *Block and flow Diagrams*
* *Form Factor Requirements*
* *Figures & Illustrations*

# Interfaces

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

******

Sample Block Diagram Showing Interfaces

*\*Source: Virtualised EPC for on-demand mobile traffic offloading in 5G environments - Scientific Figure on ResearchGate. Available from:* [*https://www.researchgate.net/figure/The-Reference-Architecture-for-the-3GPP-Evolved-Packet-Core\_fig1\_292149641*](https://www.researchgate.net/figure/The-Reference-Architecture-for-the-3GPP-Evolved-Packet-Core_fig1_292149641)

# Special Mechanical/Thermal Design Requirements

*Please describe the thermal design requirements for your contribution and any CFD and/or thermal models etc...*

# Scaling: Horizontal and Vertical

*Please describe the ways the system extends, both vertically (ex: unit feature or capacity expansion) and horizontally (ex: across shelves, racks, regions) etc…*

# Resilience/Redundancy

*Please describe the ways the system survives failures and faults.*

*Note to author of this specification: Examples include:*

* *N+1 Redundancy*
* *Pooling*

# Systems Management

*Please document the hardware management implementation of your contribution. Include Firmware (BIOS) optional Board Management Controller (BMC), Data Center Secure Control Modules (DC-SCM), etc.*

*Note to author of this specification: This section* ***should*** *include the following below items:*

* *Statement on whether the contribution supports out-of-band manageability.*
* *Statement on the modularity of the manageability architecture. (i.e. is an OCP management module used?)*

*Note to author of this specification: This section* ***can*** *include the following but is not limited to the below items:*

* *Architecture of out-of-band management*
  + *Dedicated or shared NIC*
  + *In which power state is the OOB management enabled*
* *A list of on-platform manageability interfaces:*
  + *Connections: I2C/I3C, SMBus, RMII,*
  + *Transport Protocol: MCTP, IPMI (KCS, BT, etc)*
  + *Commands constructs: PLDM …, IPMI, SPDM, CPER*
* *A list of components whose firmware which can updated programmatically*
  + *Which support failover/rollback mechanisms*
* *A list of diagnostic or management LEDs supported*
* *A list of minimum telemetry/sensors*
* *A list of minimum controls*
* *Whether conformance to OCP Profiles has been tested*
* *For Arm-based Servers, whether conformance to Arm Server Base Manageability Requirements Specification has been followed. If so, please also indicate the conformance level (e.g., M2)*

## 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. |

*(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 adjacent systems or 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.

# Architectural Design Specifications

| DELETE THIS BLOCK BEFORE SUBMITTING |
| --- |
| INSTRUCTIONS FOR THE FOLLOWING SECTIONS:   * The following sections 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. |

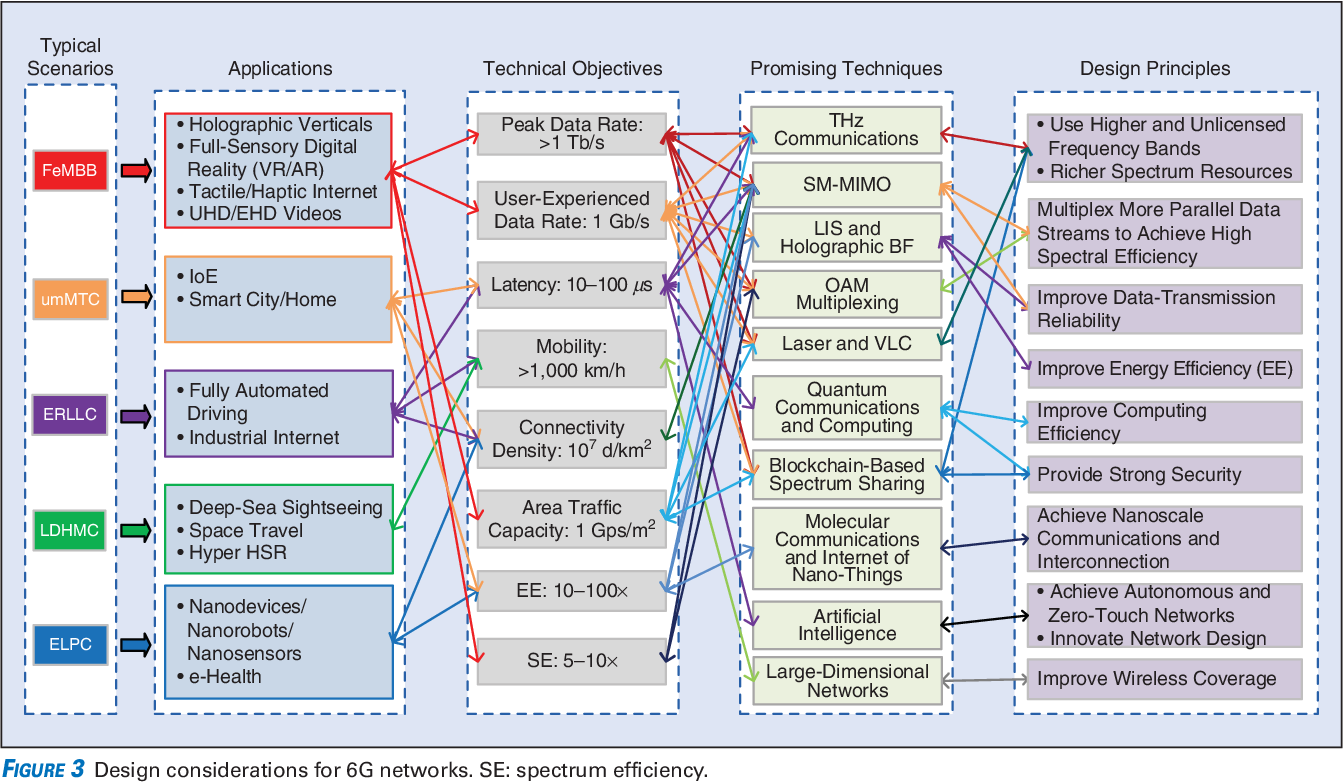
*Reminder to authors, this section refines the previous section, so it’s unnecessary to strictly repeat the previous sections, only add what modifies and refines the previous. This section is for the Design Specifications. If a Base Specification provides general requirements and design goals, the Design Specifications has detail that further defines what specific role this contribution plays, and enough detailed design information such as high level board layouts, enumerations, etc that enables end users to utilize this part of the specifications to begin the journey to realize this design.*

## 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.*

# Requirements, Use Cases, Applications, Technical Objectives, Promising Techniques, Design Principles

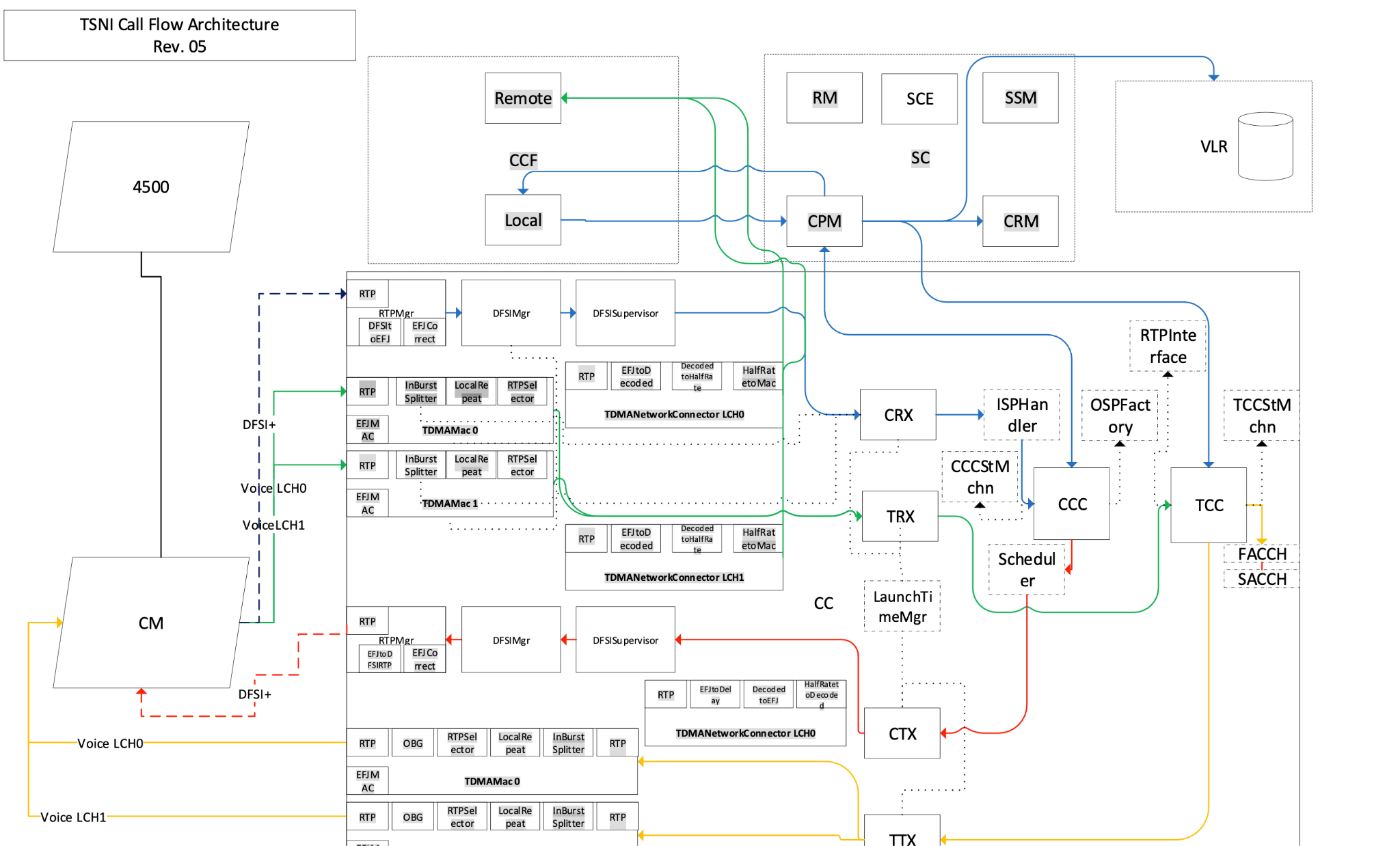
*Please restate/describe the problem, and provide a highly targeted scoping for the solution, both refining the high level information provided in the base specifications and any new requirements.*

*\*Source: Zhang, Zhengquan et al. “6G Wireless Networks: Vision, Requirements, Architecture, and Key Technologies.” IEEE Vehicular Technology Magazine 14 (2019): 28-41.*

Sample Requirements Diagram

## 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.*

**

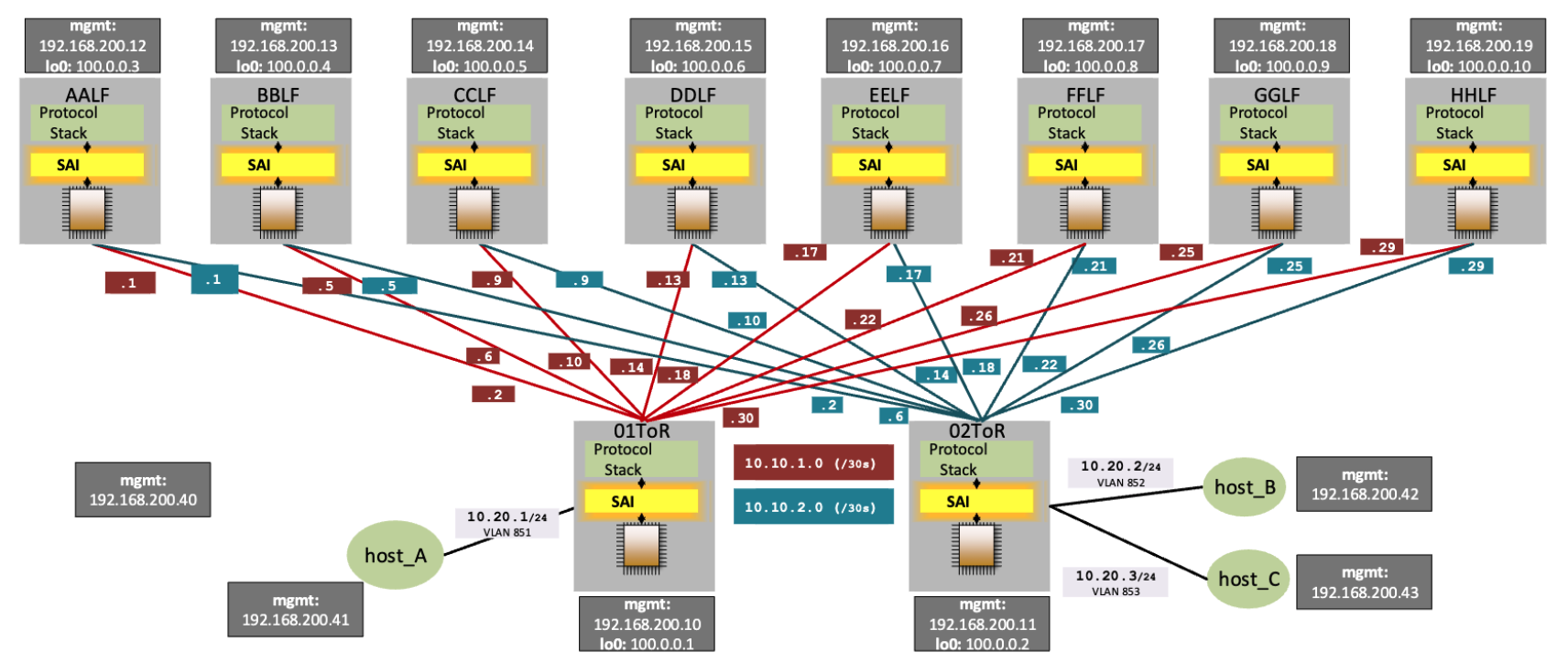
Sample Architectural Block Diagram

* + 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.*

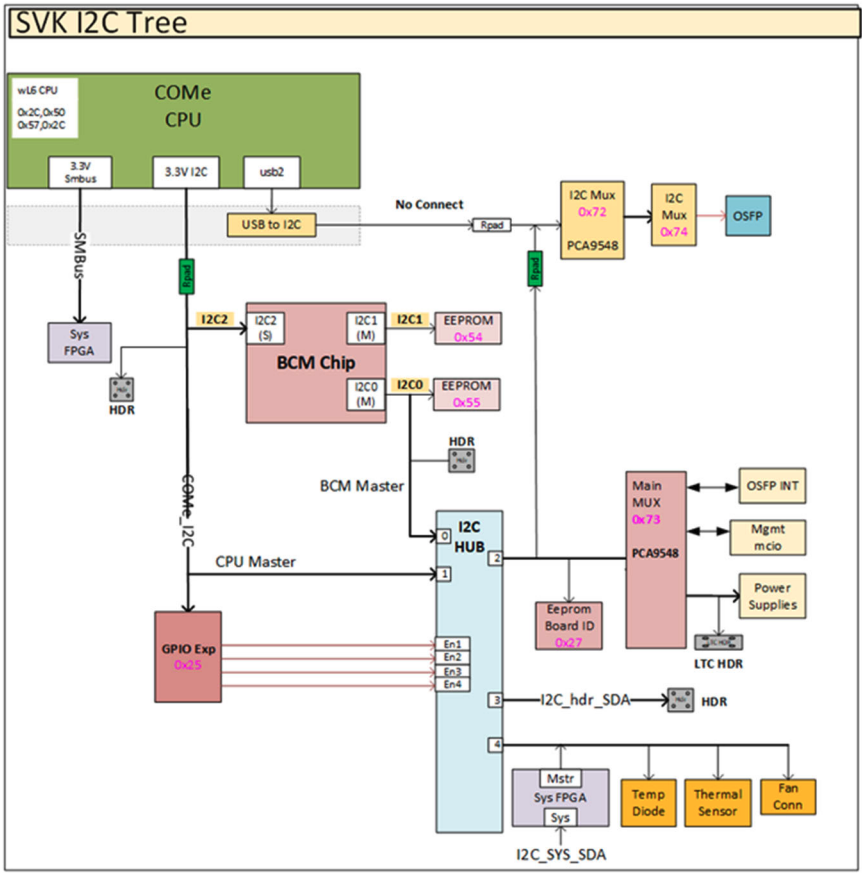
**

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.*

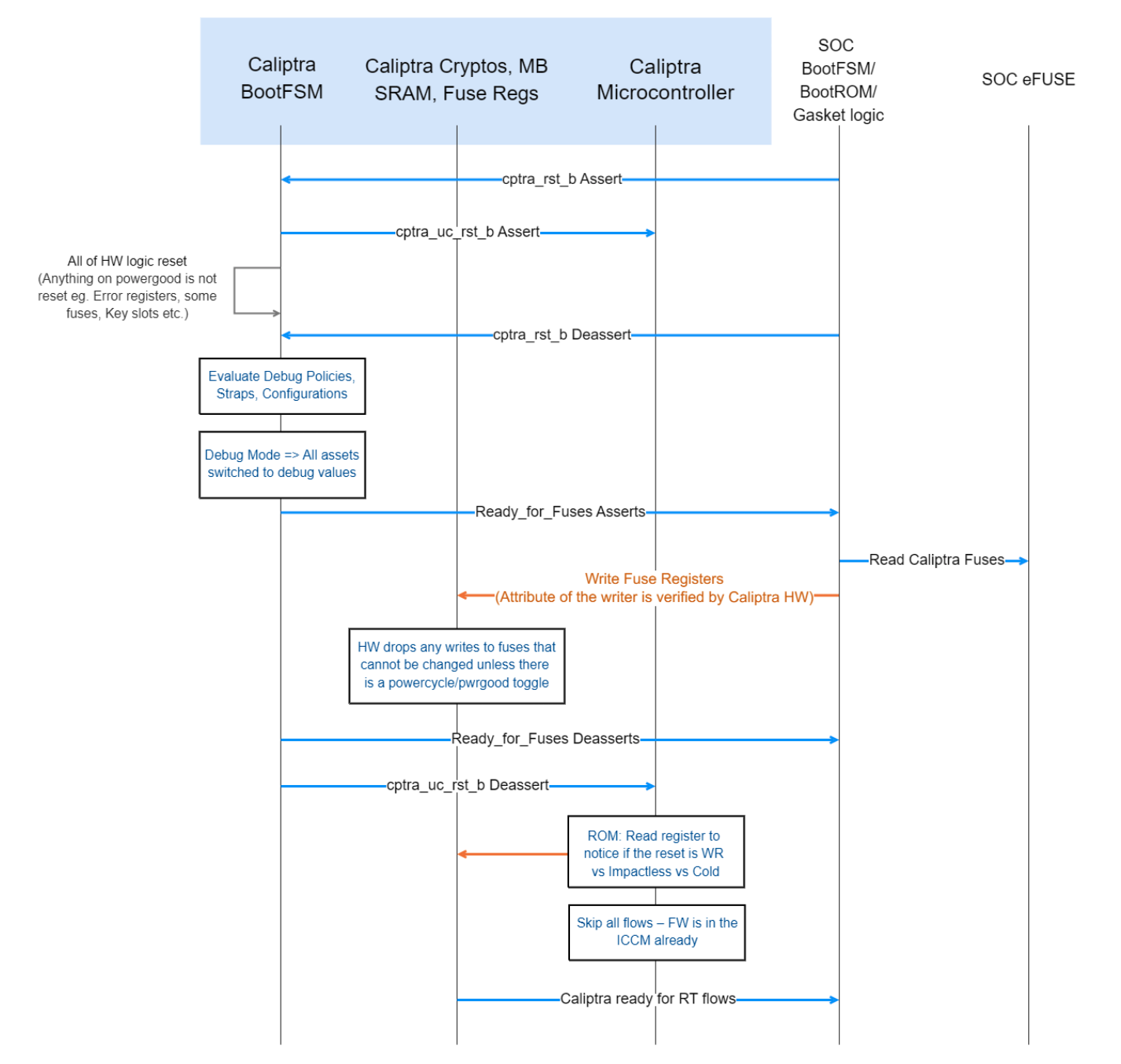
**

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.*

**

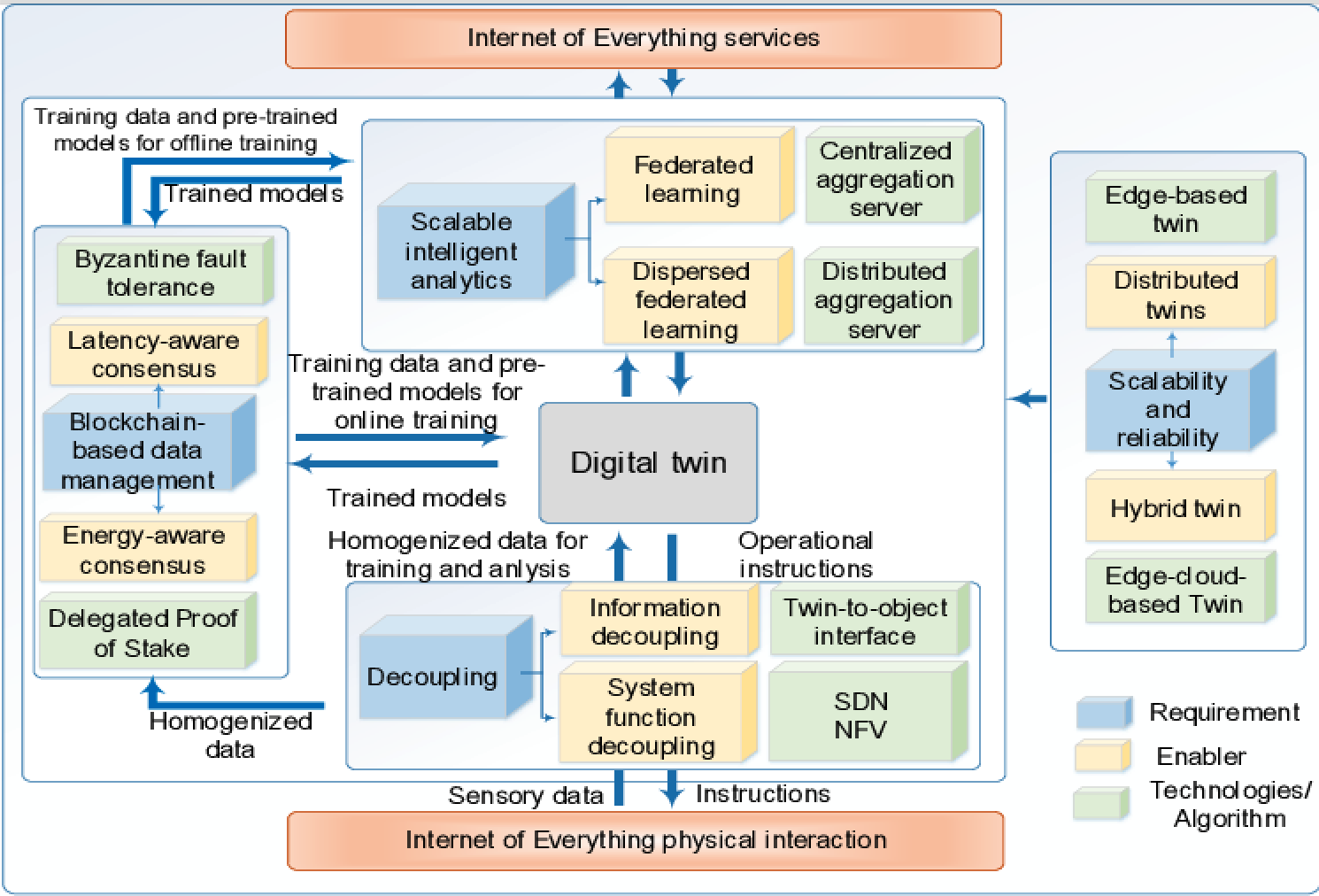
Sample Control Flow Diagram

* + 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 Profile Specification, this section can be concise if simply inheriting requirements from lower layers, however some distinctions should apply. * While it is possible to mandate full compliance with all items in this specification, it’s best to consider the consumer of this specification and only require the minimum, marking others as optional. |

*(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.

# Profile Specifications

| DELETE THIS BLOCK BEFORE SUBMITTING |
| --- |
| INSTRUCTIONS FOR THE FOLLOWING SECTIONS:   * The following sections 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. |

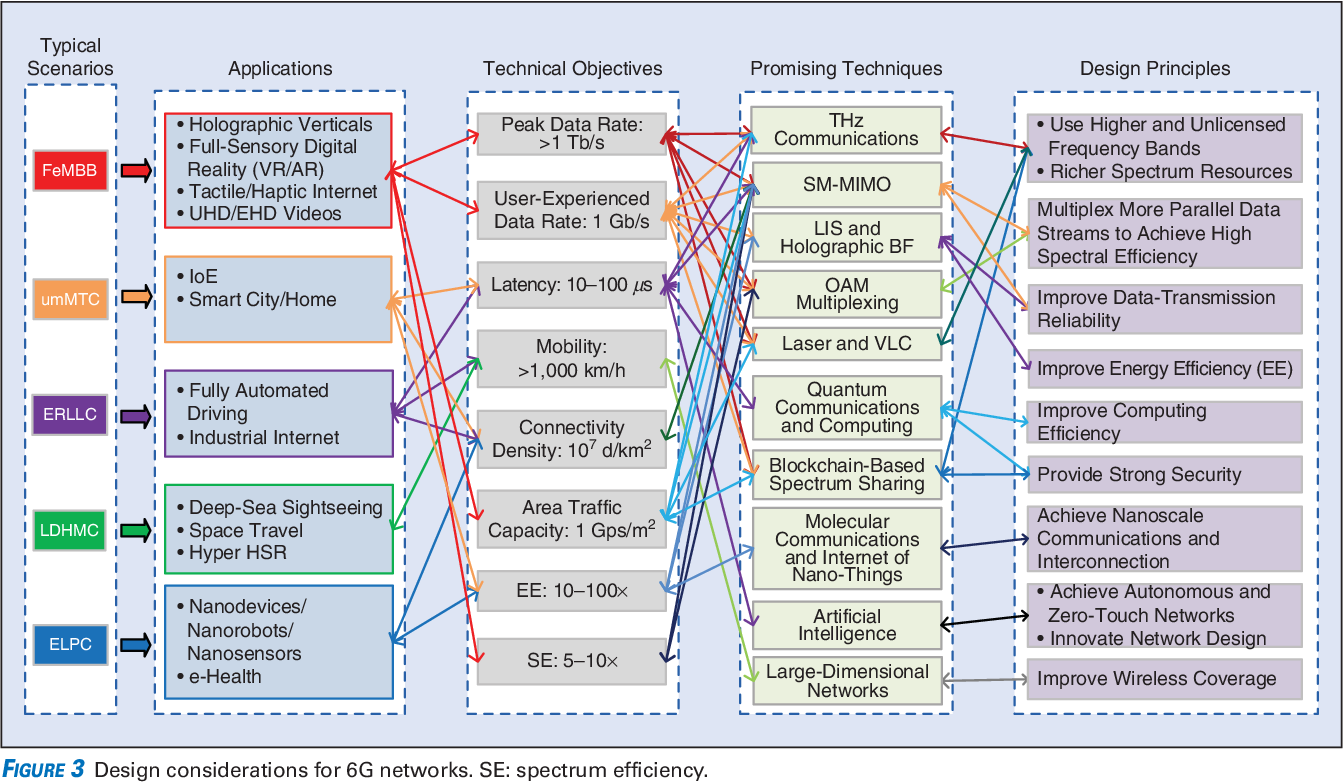
*Reminder to authors, the intended use of a profile is to be prescriptive in definition of options, configurations and so on. To the extent this section refines any previous sections, it’s unnecessary to strictly repeat the previous sections, only add what modifies the previous. This section, the Profile Specifications, requires further detail such as but not limited to bills of materials, any component part numbers, supporting gerber/design, software, tools and any other files required to be able to produce the contribution.*

## 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.*

# Requirements, Use Cases, Applications, Technical Objectives, Promising Techniques, Design Principles

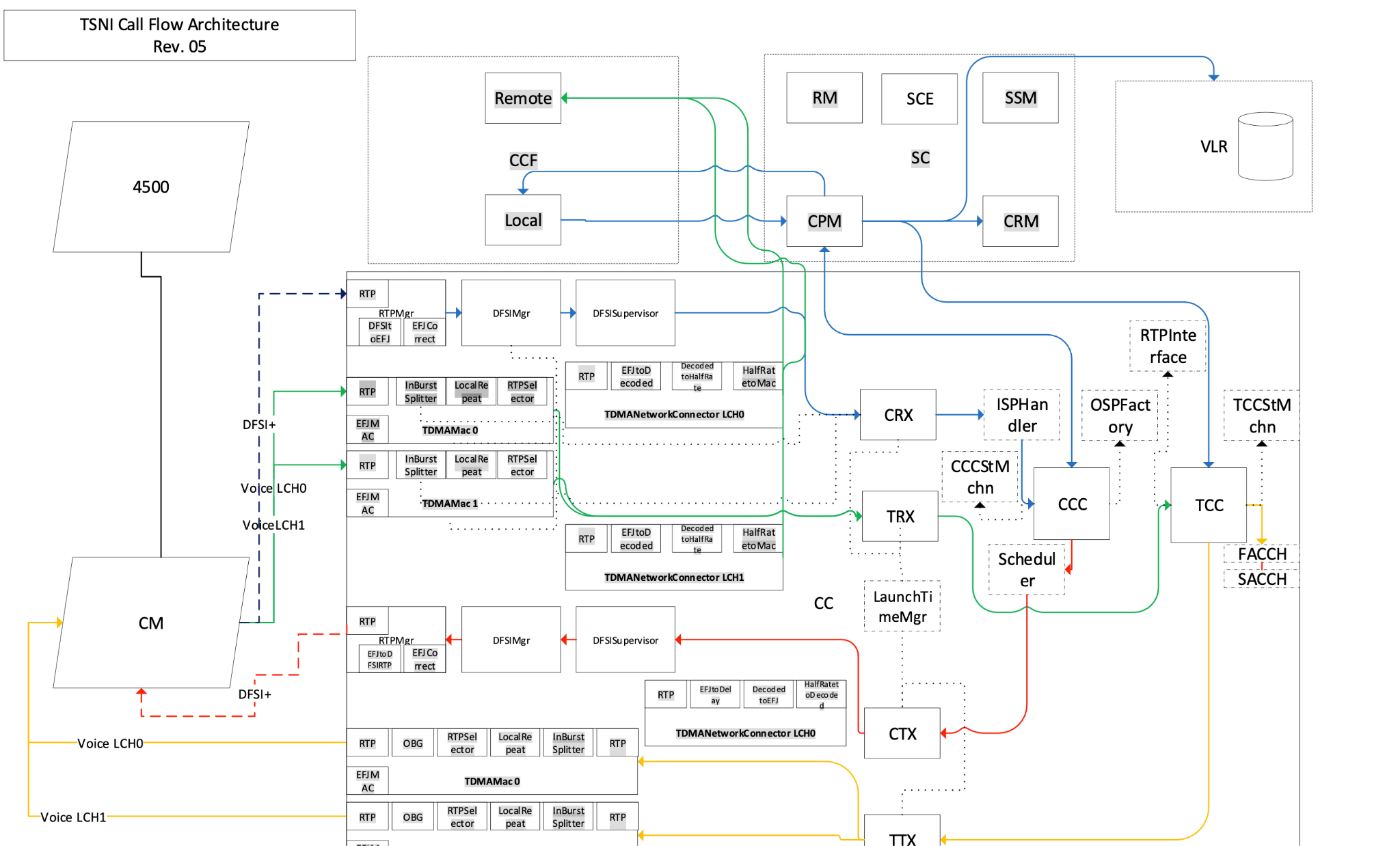
*Please restate/describe the problem, and provide a highly targeted scoping for the solution, both refining the high level information provided in the base specifications and any new requirements.*

*\*Source: Zhang, Zhengquan et al. “6G Wireless Networks: Vision, Requirements, Architecture, and Key Technologies.” IEEE Vehicular Technology Magazine 14 (2019): 28-41.*

Sample Requirements Diagram

## 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.*

**

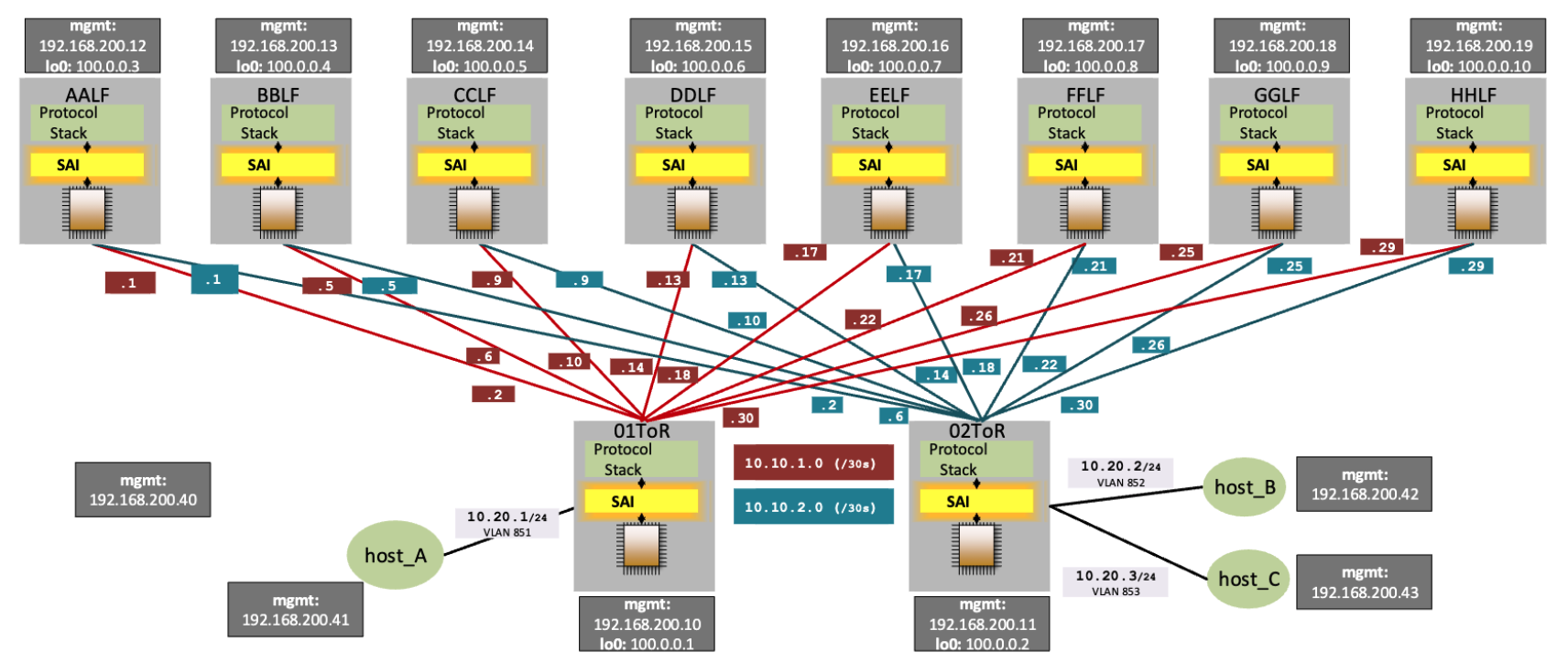
Sample Architectural Block Diagram

* + 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.*

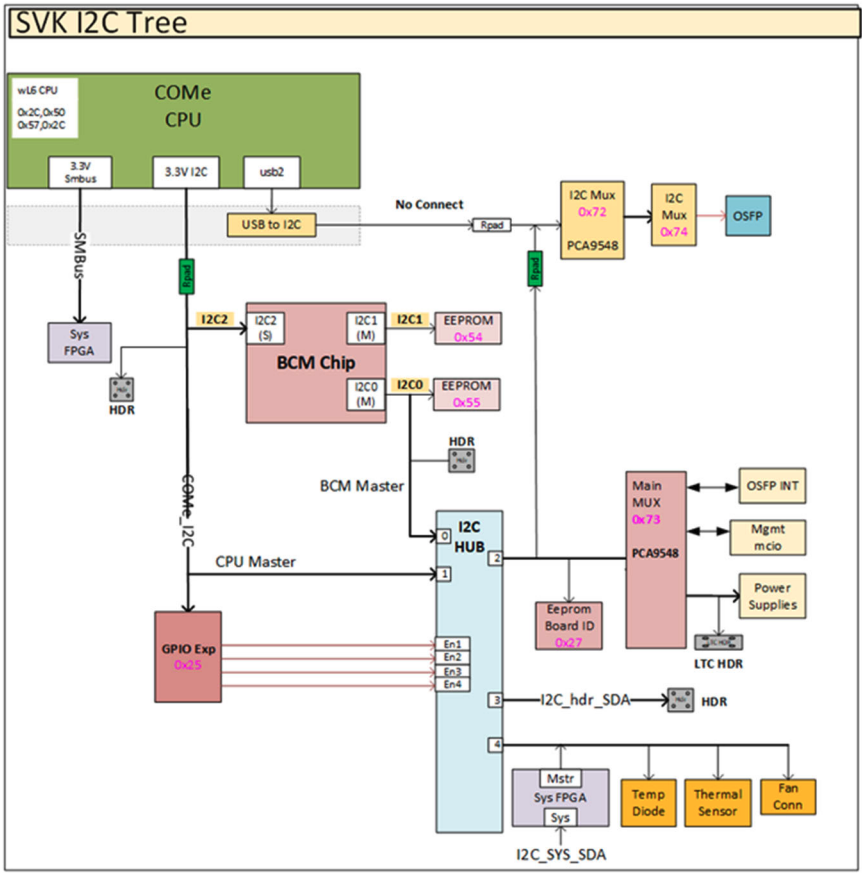
**

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.*

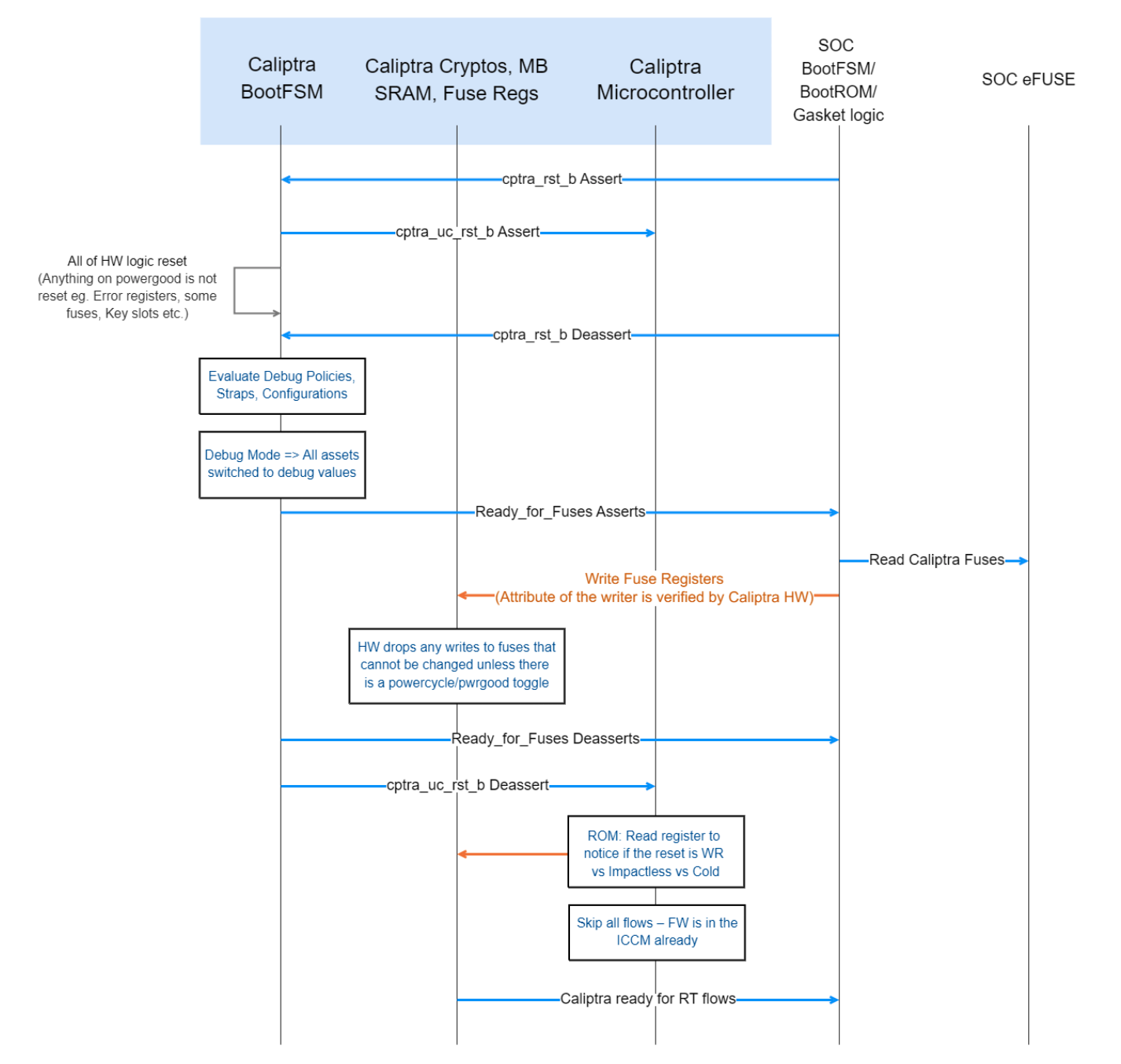
**

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.*

**

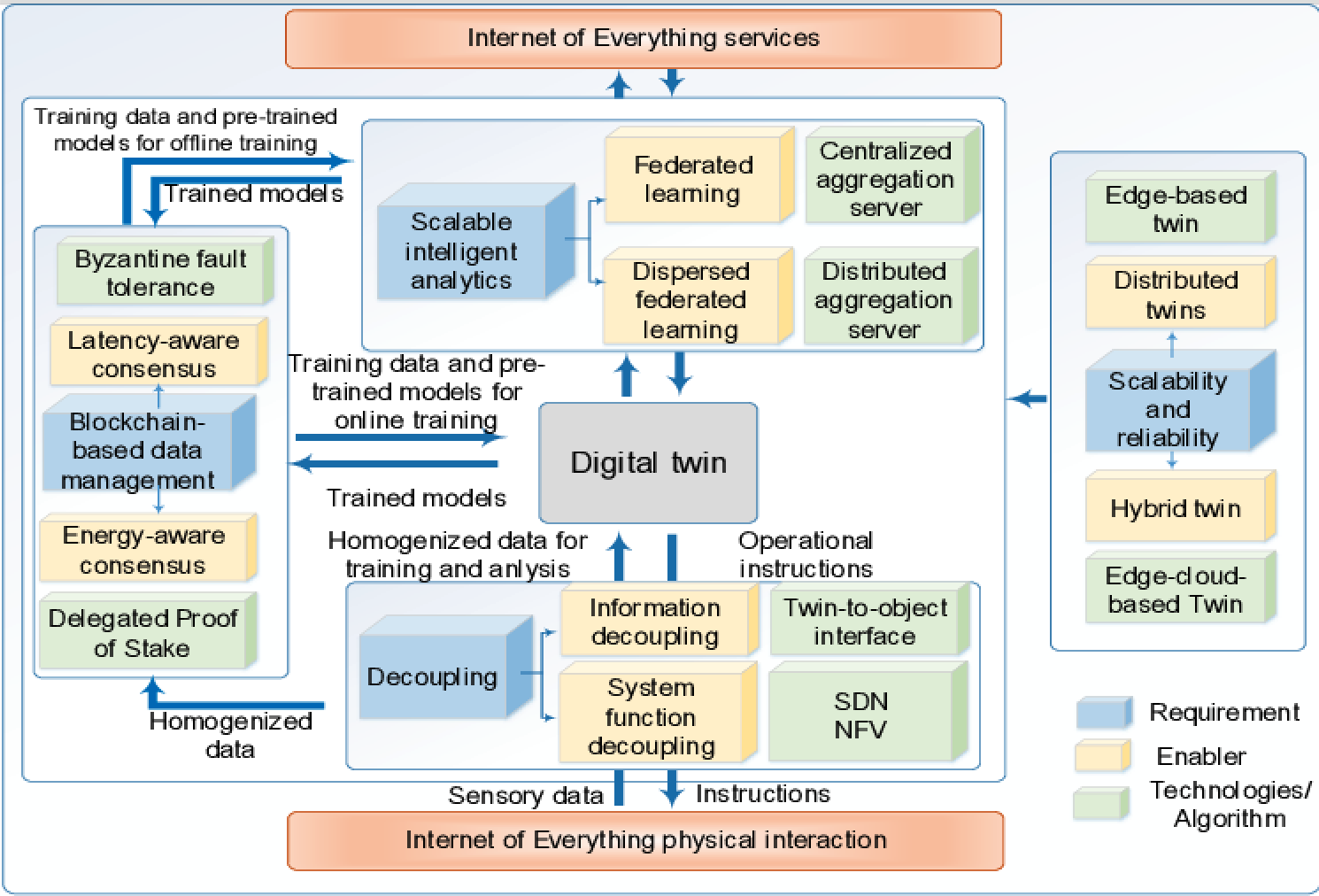
Sample Control Flow Diagram

* + 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 Profile Specification, this section can be concise if simply inheriting requirements from lower layers, however some distinctions should apply. * While it is possible to mandate full compliance with all items in this specification, it’s best to consider the consumer of this specification and only require the minimum, marking others as optional. |

*(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.

# 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 type of hardware specification am I contributing to OCP? Is it any of the below?
   1. base, design and product specification for a de-facto standard (new standard with no hardware product on the horizon)
   2. base, design and product specification for an intended physical <hardware product type> (product may be coming but within the next 1-2 years)
   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
   4. design spec (based on an existing base specification) with more refined design details (product coming in 12-15 months)
   5. a detailed product specification for a <hardware product type> for a very specific product being available in 3-6 months of approval of this Spec
   6. A Base, Design and Product Specification in a single document.
   7. If none of the above, please contact OCP Staff for better direction.
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. They will help you with your contribution and 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 [contributors@opencompute.org](mailto:contributors@opencompute.org) 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.