Collecting Use Cases Information

Method and Template for Physical Science

Workgroup Notes

Revision 1.0 (November 17, 2017)

OpenPOWER™

www.openpowerfoundation.org

Collecting Use Cases Information: Method and Template for Physical Science

OpenPOWER Foundation for Physical Science Workgroup
<opfps-chair@openpowerfoundation.org>
OpenPower Foundation

Revision 1.0 (November 17, 2017)

Abstract

The purpose of this document is is to define a method and a template to collect Physical Science use cases from scientists and research engineers working on Physical Science projects in the context and within the scope of the OpenPOWER Foundation for Physical Science Workgroup.

This document is a Non-standard Track, Work Group Note work product owned by the OpenPower Workgroup for Physical Science and handled in compliance with the requirements outlined in the *OpenPOWER Foundation Work Group (WG) Process* document. It was created using the *Document Development Guide* version 1.1.0. Comments, questions, etc. can be submitted to the public mailing list for this document at <opfps-method@mailinglist.openpowerfoundation.org>.

Table of Contents

Preface

1. Conventions

The OpenPOWER Foundation documentation uses several typesetting conventions.

Notices

Notices take these forms:

Note

A handy tip or reminder.

Important

Something you must be aware of before proceeding.

Warning

Critical information about the risk of data loss or security issues.

Changes

At certain points in the document lifecycle, knowing what changed in a document is important. In these situations, the following conventions will used.

- *New text will appear like this.* Text marked in this way is completely new.
- ~~Deleted text will appear like this.~~ Text marked in this way was removed from the previous version and will not appear in the final, published document.
- Changed text will appear like this. Text marked in this way appeared in previous versions but has been modified.

Command prompts

In general, examples use commands from the Linux operating system. Many of these are also common with Mac OS, but may differ greatly from the Windows operating system equivalents.

For the Linux-based commands referenced, the following conventions will be followed:

$ prompt Any user, including the `root` user, can run commands that are prefixed with the $ prompt.

 # prompt The `root` user must run commands that are prefixed with the # prompt. You can also prefix these commands with the **sudo** command, if available, to run them.

Document links

Document links frequently appear throughout the documents. Generally, these links include a text for the link, followed by a page number in parenthesis. For example, this link, Preface [iv], references the Preface chapter on page iv.

2. Document change history

This version of the guide replaces and obsoletes all earlier versions.

The following table describes the most recent changes:

Revision Date	Summary of Changes
November 17, 2017	• Version 1.0: WG Approved Workgroup Note
June 9, 2017	• Revision 0.4: language improvements. Moved Section 3.5 to 3.2. Added definitions.
May 30, 2017	• Revision 0.3: Discussion during the OPFPS Coordination meeting 2, May 9, 2017. Some issues around two subsections: (1) “Underlying causes of a problem”: it is not clear what the user should specify, and the risk is that the user starts to think about the solution in a “problem definition” section. Dirk and Marilena propose the title “Known challenging and risks”, to make clear that sometimes there are challenges and risks behind the problem. It could be important for finding them that may not be immediately apparent for an external stakeholder. (2) In the subsection “scope“ it must be much clear that the user should provide just a summary. Dirk proposes to rename the section in “Outline of the scope”. (3) Added section Section 2.2.1, “An iterative process” [3]
May 5, 2017	• Revision 0.2: Conversion to OpenPOWER Foundation Document Template
April 27, 2017	• Revision 0.2: Dirk's suggestions Authors: Andrea Bulgarelli (INAF), Daniele Gregori (E4), Valentina Fioretti (INAF) , Dirk Pleiter (Jülich Supercomputing Centre (JSC))
March 21, 2017	• Revision 0.1: First Draft Authors: Andrea Bulgarelli (INAF), Daniele Gregori (E4), Valentina Fioretti (INAF)

1. Introduction

1.1. Purpose

The main purpose of this Note is to define a method and a template to collect Physical Science use cases from scientists and research engineers working on Physical Science projects in the context and within the scope of the OpenPOWER Foundation for Physical Science Workgroup.

An effective method (shared between all stakeholders) could contribute to

- understand the workflow, starting with user expectations;
- help Physical Science projects maintain costs within a chosen envelope;
- map the functionalities to the scientific requirements;
- remove possible misunderstanding between the scientific community and ICT stakeholders.

Chapter 2, “The context” [3] describes the context and the main OPFPS workflow. The proposed method is a template reported in Chapter 3, “Requirement inception method” [4] and justified in Chapter 4, “Justifications” [9] .

1.2. Scope

This OPFPS WG should contribute, with the foreseen deliverables (WP2, WP3, and WP4) (see [1]), in a more effective process of managing the increasing complexity of a Physical Science project, because this could help to take the ICT system under control or to find new solutions that are at the corner of the ICT industry.

1.3. Stakeholders identification

This section describes the intended audience of this Note and of the OPFPS WG.

Stakeholders are individuals, groups, organizations who are part or are interested to a Physical Science project. All stakeholders could be OPFPS members, but the participation is open also to not-OPF members. The stakeholders are:

- physical Science community: stakeholders that are actively involved in the scientific project, are affected by its outcome or can influence its outcome, and that should provide requirements:
  - scientists;
  - research engineers.
- ICT community: stakeholders that could provide a solution;
- WG participants: members of both communities that should provide support for collecting, analyzing and categorizing requirements.

1.4. Acronyms

OPF = OpenPOWER Foundation

OPFPS = OpenPOWER Foundation for Physical Science

WG = Workgroup

1.5. Definitions

- **Physical Science** is any science concerned with nonliving matter, energy, and the physical properties of the Universe, such as physics, chemistry, astronomy, and geology.
- **Problem analysis:** Problem analysis involves identifying the overriding problem and establishing the causes and effects related to that problem.
- **Scientific objectives:** The scientific objectives are the goals of a scientific project. The scientific requirements are needed to reach the scientific objectives.
- **Scientific requirements** A) are high-level statements that are needed to reach the scientific objectives/goals of the scientific project (and of the scientific organisation that manages the project). They describe the results that the project will achieve, and the metrics which will be used to measure its success. Scientific requirements must be achieved to provide value for the scientific project. B) A requirement of a physical science experiment described in terms of the parameters to perform science (e.g. minimum sensitivity of a telescope, minimum angular resolution).
  - Example: we need to observe N celestial objects in T time.
  - Example: we need to repoint the telescope in 30 seconds.
- **System requirements:** requirements of the HW/SW solution, i.e. the functional and non-functional requirements of the proposed system. Functional requirements define a function or a behaviour of a system (a list of the major features or user capabilities unique to the product). Non-functional requirements impose qualities and constraints on the design or implementation (such as performance, security, or reliability).
- **System:** the solution that should be proposed/developed.
- **Big Science Projects:** large international experiments, generally systematically organized and managed.

1.6. Reference

[1] OPFPS charter, 2016, https://members.openpowerfoundation.org/wg/OPFPS/document/1080

[2] SEI, “A Framework for Software Product Line Practice, Version 5.0”, 2012, http://www.sei.cmu.edu/productlines/frame_report/req_eng.htm

[3] Alistair Cockburn, “Writing Effective Use Cases”, Addison Wesley, 2000

[4] M. Christel, K. Chang, “Issues in Requirements Elicitation”, CMU/SEI-92-TR-012, 1992

[5] Macaulay, Linda, Fowler, Chris, Kirby, Mark, and Hutt, Andrew. USTM: A New Approach to Requirements Specification. Interacting with Computers 2(1):92-118, April 1990.

[6] Robin F. Goldsmith, JD, “Conventional Requirements Model Flaw Misses REAL Business Requirements”, 2007, available at *https://www.google.com/url?q=http://www.gopromanagement.com&sa=D&ust=1493996251691000&usg=AFQjCNFeYz4qDqEpGVfWdbi5gyI1Xp3X-A*

[7] Robin F. Goldsmith, JD, “BAs Will Falter Until They Learn to Discover REAL, Business Requirements”, IEEE 17th Requirements Engineering Conference (RE’09), 2009

[8] Dean Leffingwell, Don Widrig, “Managing Software Requirements: A Use Case Approach”, 2nd Edition, 2003

2. The context

2.1. Physical Science

Physical Science is any science concerned with nonliving matter, energy, and the physical properties of the Universe, such as physics, chemistry, astronomy, and geology.

In this context, the OPFPS WG cover the following aspects:

- needs of a scientific experiment from Big Science Projects to experiments at laboratory level;
- software frameworks and libraries used in Physical Science projects;
- challenging hardware requirements.

2.2. OPFPS workflow

A basic **OPFPS workflow** could be summarized as the following:

- *requirement inception (eliciting requirements)*, to collect functional requirements (e.g. as use cases) from the scientific community (from scientific stakeholders);
- *requirement analysis*, determining and refining the system requirements, to provide a coherent view of the use cases, and to resolve misunderstanding (performed by OPFPS members);
- *system modeling*: deriving a logical workflow/data flow model of the system, to derive commonalities between different use cases (performed by OPFPS members);
- *enable ICT stakeholders to provide a suitable solution.*

This Note covers the requirement inception phase.

2.2.1. An iterative process

The proposed workflow is iterative so that reference solutions can be reworked in the light of increased knowledge and to take under control the problems listed in Section 4.1, "Requirement inception for a Physical Science project" [9]. Iteration means that the scientific stakeholders should provide use cases and refine them based on feedbacks received by other stakeholders and by the OPFPS members. With an iterative process that will involve all stakeholders, mistaken assumptions can be detected and corrected.

The OPFPS members are available to update the common workflow for each new use case that is added or updated.

Iterations will also be made on the proposed method:

- defines a method (see Chapter 3, "Requirement inception method" [4]);
- apply the method to concrete use cases;
- review approach and update the method;
- apply updated method to the next use cases.

3. Requirement inception method

3.1. Requirement inception

Requirement inception is the collection of use cases from the scientific community. The main purpose is to identify **system requirements** (the requirements of a solution) that provide real value for the scientific project, i.e. that fulfill **scientific requirements**.

Two main steps are foreseen:

- **Description of the scientific problem and related scientific requirements.** In this context, the WG asks scientific stakeholders to provide an overview/summary of their problem (with a short description of the scientific objectives), and to provide scientific requirements that could be measurable; this helps to identify the scientific requirements for which a possible hardware/ software solution has the greatest impact. In this way the users will understand the real value of the proposed solution;
- **Definition of use cases and identification of system requirements** (functional and non-functional) that are used as input to define a solution.

System requirement can be divided in *system functional requirements*, to define a function or a behaviour of a system, and *system non-functional requirements*, to impose qualities and constraints on the design or implementation of the system (such as performance, security, or reliability).

In Section 3.3, “The template” [4] we report the template, that reflects the two steps depicted above. Section 3.4, “Problem analysis and scientific requirements” [6] , Section 3.5, “System requirements” [7] and Section 3.2, “The system (the solution)” [4] report a detailed description of the template. Section 3.6, “In a nutshell” [8] is a summary.

3.2. The system (the solution)

The system, for which we collect system requirements could implement a workflow, or a data flow of a solution for a scientific project, a software library, or a software framework that could be used in a workflow or a data flow.

3.3. The template

Scientists, research engineers, hardware and software developers and vendors do not have a common language and terms; all WG participants need details and guidance to organize the information uniformly. For the purpose of this work, the use cases must be readable by all interested stakeholders.

In this section, we propose a **template to collect Physical Science use cases** from **scientific stakeholders**. This template could help to collect problem statements and solution requirements. The template could by filled by scientific stakeholders, or OPF members could help scientific stakeholders to fill the template. With this template we will help stakeholders to extract useful information:

1. to better understand which scientific requirement the proposed solution should fulfill;
2. to better identify use cases/workflows/functional requirements and the context;

3. to reduce problem of scope, understanding, and presumptions (check Section Chapter 4, "Justifications" [9]);

4. to avoid to omit information that is mandatory to extract workflows;

5. to enable solutions with a real benefit for scientists.

The template is organized in sections and where some key questions or suggestions are presents.

Scientific problem analysis	
Name of the **scientific project**	
Scientific **objectives**	Write some short statements about the objectives of the scientific project for which the scientific stakeholders want to resolve a problem. This section is useful to put the scientific project under analysis in the general context. Define objectives, not requirements.
Scientific requirements and problem definition	Write a **scientific problem statement**. The statement should include: • What is the problem for the scientific project? • Who is affected by it? • What is the impact of this problem on the scientific project? Identify here the **scientific requirements** that the solution can contribute to fulfill. The scientific requirements should be **measurable.** *The problem of …* *affects …* *and results in … (list scientific requirements that are not fulfilled/ that is too expensive to fulfill/ that must be fulfilled)*
Known **challenging** and **risks**	Sometimes there are challenges and risks behind the problem. It could be important to find factors that may not be immediately apparent for an external stakeholder.
System requirements of the solution (a product/hardware-software solution)	
Outline of the **scope**	Define solution system boundaries, e.g. with a context diagram or in a textual form. Details are in the following sections of the table.
Benefits of a **solution**	The resolution of these problems is also the resolution of scientific requirements, i.e. the increasing of real value for the scientific community. This contributes to define new measurable to compare different solutions. Some questions: • Is there a proposed or a current (in-operation) solution? • Why the current (if exists)/proposed solution is it not profitable? Poor design? Too expensive? • What are key benefits to find a new solution for this problem? Example. *The current solution is... (if present). Benefits of a solution/new solution that create a new system to address the problem include: ... The solution could have a direct impact on the following scientific requirements: ...*
Available technologies of the context	List currently available technologies used in the context of the considered scientific problem.
Assumptions	Assumptions made
Description of the **workflow**/main functional and non-functional **requirements**	It is possible to describe the solution in term of a workflow, a list of functional requirement (list the major features or user capabilities unique to the product) or a use case and a list of non-functional requirements. This Note does not specify "how to" write requirements. **Guidelines:** it is important to define problems parametric as some performance targets, to help to understand how different performance requirements scale. The final solution (hardware and software) depends on storage needs, computing power, and programming techniques. Some questions are provided, but feel free to add any information that you think is important:

	1. For an experiment, the data acquisition is one of the most important aspects and may require technologies capable of transmitting data between sensors and acquisition equipment in very short times and with low latencies. Could you describe the main system requirements of your data acquisition system? (e.g. data rate, latency, need of high precision synchronization protocols (PTP)). *Example: for a scientist who needs to acquire data at a rate of 3 GB/s from a distributed network of sensors the solution is a system able to receive and store data at the same rate but with a scalability of ...* 2. In some cases, the data must be transferred from the acquisition site to another place for storage and data analysis. Could you describe the workflow? Do you need appropriate protocols to ensure the integrity of the data during the transfer? Do you need compression techniques? Do you need public/private interfaces to access the data? Could you provide the main system requirements of your data transfer system? 3. If you perform simulations, could you describe the workflow? Could you provide the data rate and the storage size that your simulation generates? 4. Must the storage be accessible by more computing nodes simultaneously without any special requirements regarding performance? Must the storage be accessible by more computing nodes simultaneously, allowing high throughput values and the possibility of the available space to growth? Could you provide the main system requirements of your storage system (size, latency, rate, the number of read/write accesses, and so on)? 5. Does the storage require separate areas for data and metadata (e.g. to improve performance and security)? 6. Do you need an archive system? Could you provide the main system requirements of your archive system? Do you need in-memory databases? Do you need No-SQL databases? 7. Do you need a data preservation system? Could you provide the size and the number of years that the archive should last? 8. The code used for the analysis or simulation is implemented with a distributed memory programming paradigm (e.g. MPI)? Or the code make use of shared memory (e.g. OpenMP)? 9. Do you need distributed computing frameworks (e.g. Hadoop)? 10. Do you need stream computing frameworks (e.g. Spark)? 11. Is your code single thread/multithread/multiprocessing/vectorizable? 12. The code links numerical libraries? Which libraries? 13. The code performs the same operation on a large amount of data (SIMD programming paradigm)? 14. Do you require machine learning techniques for data mining? 15. Do you have specific software libraries (that maybe could require porting to the OpenPOWER architecture)? 16. Do you need some special equipment that should be connected to the data acquisition or data analysis system (e.g. FPGAs)
Constraints	Restrictions on the solution space and major dependencies to external factors outside of the project's control: • Economics (e.g. costs, licensing issues) • Politics (e.g. internal or external, interdepartmental issues) • Technology (e.g. choice of technology, interface with existing technologies/platforms) • Systems (e.g. existing system, compatibility issues) • Environment (e.g. legal/environmental/security/standards) • Schedule and resources (e.g. fixed schedule, team)

3.4. Problem analysis and scientific requirements

This section reports a justification of the template.

The section **"Scientific problem analysis"** is a summary of the problem that the scientific project covers/want to resolve, and helps OPFPS members and ICT stakeholders to focus on useful information from the problem domain of the scientific project. This section is a summary of the problem analysis performed by scientific stakeholders that help OPF members and ICT stakeholders to identify the real value for scientists in the resolution of their problems.

The first sub-section *"Scientific objectives"* contains the identification of scientific objectives, i.e. the goals of the scientific project.

The scientific requirements are needed to reach the scientific objectives. The sub-section *"Scientific requirements and problem definition"* helps scientific stakeholders to identify the **scientific requirements** of the project (in analogy with the "business requirements" for an enterprise), and to provide a **metric** that will be used to estimate the effectiveness of the proposed solution.

The primary reason products/systems don't help to improve/satisfy the right scientific requirements is because the challenging and risks of the scientific requirements have not been identified adequately or they have not been properly understood by the ICT stakeholders. The sub-section *"Known challenging and risks"* helps to focus on this aspect.

For a justification of the problem analysis, and the impact on the requirement inception see Section 4.2, "Problem Analysis" [10].

3.5. System requirements

With the scientific requirements in mind, the scientific stakeholders should define the system requirements for the hardware/software solution, i.e. the functional and non-functional requirements of the proposed system.

These are the primary information about the product/solution:

- Project Scope: identifies solution system boundaries. A context diagram could be provided that shows a top-level view of a system and the system's boundaries (section "Outline of the scope").
- The benefit of a solution: to connect the solution with scientific requirements that must be affected by the solution.
- Available technologies of the context: to let non-scientific stakeholders to better understand the technological context of the scientific projects.
- Provide clear statements of assumptions (section "Assumptions").
- Functional and non-functional requirements. It is possible to describe the solution in term of a workflow, a list of functional requirement (i.e. list the major features or user capabilities unique to the product) (Section 3.5.1, "Use cases and system functional requirements" [7]) and provide a list of non-functional requirements (Section 3.5.2, "System non-functional requirements" [8]).
- Identify the constraints: put limitations on the ability to deliver a solution as envisioned restricting the solution space (section "Constraints").

3.5.1. Use cases and system functional requirements

A Use Case is a description of system behaviour in response to a request from one actor. We are collecting Use Cases (and related scenarios) starting from scientific requirements to

- describe how an actor interacts with the system in terms of functions and workflows of the Physical Science Projects/software library/software framework;
- aim for a concise, specific and comprehensive definition of the requirements anticipating that trade-offs may need to be made if technology does not meet all requirements;
- encourage a common understanding of requirements on system behaviour for all the involved stakeholders;

- improve communication among WG participants, OPF members and Physical Science community at large;
- discover and document issues related to requirements that are not covered by current hardware/software solutions/current technology;
- give context to non-functional requirements.

3.5.2. System non-functional requirements

This document does not list all the non-functional requirements. Between them we could remember:

- **performance, availability, efficiency, reliability, scalability, security** requirements
- we should also take into account the **reproducibility** of the results as requirements (in particular in parallel computing environment);
- think about: testable, and or (at least) traceable

3.6. In a nutshell

With **use cases/workflow/list of system requirements**, we describe a possible workflow/data flow of a system and/or software library of a scientific project, using the template.

The **workflow** is a view of the functional requirements of the system, with attached system non-functional requirements.

A **system** could be a data analysis system or a software library/framework that covers some requirements.

The **proposed solution** is a possible solution, using OPF hardware and software, that implement a system of a scientific project, or that optimize an existing solution, or that optimize the use of software libraries/framework used/developed by the scientific community

The proposed solution contributes to achieve the **scientific requirements** of a scientific project.

4. Justifications

In this section are provided the reasons for using the proposed method.

4.1. Requirement inception for a Physical Science project

Before system requirements can be analyzed, modeled, or specified, they must be gathered through an elicitation process.

An effective process of requirement inception (shared between all stakeholders) could contribute and therefore to

- understand the workflow, starting with user expectations;
- help Physical Science projects maintain costs within a chosen envelope, helping to decide what/ which ICT system to build, what the system must do, how it must behave (in the context of a general Physical Science project workflow), the properties it must exhibit, the qualities it must possess, and the constraints that the system and its development must satisfy;
- map the functionalities to the scientific requirements.

Physical Science projects are at the corner of the Physical Science (to open new windows in the scientific domain) and for this reason are at the corner of the current technology, or (more common) new technology should be developed to fulfill the requirements. With this in mind, a requirement inception process is a challenge because there are many different problems that the OPFPS members should manage to **remove misunderstanding between scientific and ICT stakeholders**. Between them:

- *The problem of scope:*
  - the user (scientist/research engineer) specifies too many technical details with the risk to lose the functional aspect of their problem, and this implies a poor understanding of the functional requirements, i.e. of the workflow;
  - and/or the boundary of the system is not well defined;
  - to mix functional and non-functional requirements.
- *The problem of understanding:* some of them could come from
  - the users (scientist/research engineer) is not able to translate the problem domain (because the problem domain covers too many areas of knowledge) in a comprehensible way;
  - the involved stakeholder has a full understanding of his specific problem but do not have a full understanding of the general problem domain;
  - have trouble communicating needs to the hardware/software developers/vendors;
  - omit information that is believed to be “obvious”;
  - specify requirements that conflict.

- *The problem of presumption:* Presumptions: an assumption, an idea that is taken to be true, and often used as the basis for other ideas, although it is not known for certain. In our context, this presumption could be made by the scientific stakeholders or by who is collecting requirements, usually when some aspect of the problem domain are not well known or understood.
- *Requirements volatility:* the requirements change over time (see Section Section 4.1.1, “Volatility” [10]).

A Physical Science project has no similar examples that could be followed, due to its uniqueness. The current experiments help us in collecting experience, but sometimes new Physical Science projects are one order of magnitude bigger than current scientific projects (Big Science Projects).

4.1.1. Volatility

This section is for ICT stakeholders, to understand some aspects of the organization of the scientific projects.

About the **volatility**, in a Physical Science project the requirements could change over time for many different reasons:

- the **organizational complexity** (and its evolution) is a cause of requirement volatility. Organizational goals, policies, and roles of intended end users all may change during the development of the system; to keep this problem under control it is important to identify the right **stakeholders** within the organization;
- the requirements are the product of the contributions of many individuals, and these individuals often have conflicting needs and goals (different scientific objects within the same project); to keep this under control we have proposed a template in which we try to extract basic information from the scientific projects (using some basic steps of **problem analysis**, see Section Section 4.2, “Problem Analysis” [10]);
- increasing knowledge during the development: (i) people are learning/will learn new things during the development of the experiment, during prototyping activities or testing and commissioning phases, and will learn during the operations; (ii) one of the most important causes of requirements volatility is that user needs evolve over time, because the science evolves over time thanks to new scientific discoveries (the advance of science) that could change the scientific objectives and/or the workflow of the project itself during the development. We have foreseen an **iterative process** (see Section Section 2.2.1, “An iterative process” [3]).

4.2. Problem Analysis

The OPFPS members should analyze the information provided by the scientific stakeholders to understand if this is useful to define an effective solution. A key element of this analysis should ensure that “root causes,” not just the symptoms of the problem, are identified and subsequently addressed by the system.

Due to the uniqueness of the scientific projects and the complexity of the problem domain, the problem analysis helps to manage the problems of scope and understanding, because they could strongly affect the requirement inception. In addition, the problem analysis could avoid or reduce presumptions (e.g. from OPFPS members that will analyze use cases).

Some aspects of the problem analysis of a scientific project that could be useful in the OPFPS context are:

- Problem definition: ask scientific community to write a problem statement in an agreed format, to provide the context and to identify the real value of a solution;
- Try to find underlying causes of a problem to define measurable of a new solution. There is often a problem behind the problem or behind the current solution. It could be important to consists of finding underlying causes that may not be immediately apparent. Some key questions help in this process: why is it not profitable? Poor design? Too expensive? It is important to define which factors contribute to the problem and a way to measure the improvements.

Appendix A. OpenPOWER Foundation overview

The OpenPOWER Foundation was founded in 2013 as an open technical membership organization that will enable data centers to rethink their approach to technology. Member companies are enabled to customize POWER CPU processors and system platforms for optimization and innovation for their business needs. These innovations include custom systems for large or warehouse scale data centers, workload acceleration through GPU, FPGA or advanced I/O, platform optimization for SW appliances, or advanced hardware technology exploitation. OpenPOWER members are actively pursing all of these innovations and more and welcome all parties to join in moving the state of the art of OpenPOWER systems design forward.

To learn more about the OpenPOWER Foundation, visit the organization website at openpowerfoundation.org.

A.1. Foundation documentation

Key foundation documents include:

- *Bylaws of OpenPOWER Foundation*
- *OpenPOWER Foundation Intellectual Property Rights (IPR) Policy*
- *OpenPOWER Foundation Membership Agreement*
- *OpenPOWER Anti-Trust Guidelines*

More information about the foundation governance can be found at openpowerfoundation.org/about-us/governance.

A.2. Technical resources

Development resouces fall into the following general categories:

- Foundation work groups
- Remote development environments (VMs)
- Development systems
- Technical specifications
- Software
- Developer tools

The complete list of technical resources are maintained on the foundation Technical Resources web page.

A.3. Contact the foundation

To learn more about the OpenPOWER Foundation, please use the following contact points:

- General information -- <info@openpowerfoundation.org>
- Membership -- <membership@openpowerfoundation.org>
- Technical Work Groups and projects -- <tsc-chair@openpowerfoundation.org>
- Events and other activities -- <admin@openpowerfoundation.org>
- Press/Analysts -- <press@openpowerfoundation.org>

More contact information can be found at openpowerfoundation.org/get-involved/contact-us.