

# Comprehensive Compliance and Interoperability Audit of Open Source Quantum Simulation Architectures: A Case Study of the QForge Toolkit

## 1. Executive Overview and Legal Jurisprudence

The contemporary quantum computing software ecosystem is a complex tapestry of intellectual property, woven primarily from open-source threads. The rapid advancement of the field relies heavily on the cross-pollination of academic research and industrial engineering, necessitating a legal framework that balances proprietary interests with communal scientific progress. The "QForge" project, a quantum simulation toolkit licensed under the **Apache License 2.0**<sup>1</sup>, stands as a representative case study of this modern architectural paradigm. By orchestrating functionality from distinct third-party libraries—specifically **scqubits** (BSD 3-Clause), **QuTiP** (BSD 3-Clause), **Qiskit** (Apache 2.0), and **Qiskit Metal** (Apache 2.0)—QForge occupies a critical position in the software supply chain. The user's query regarding the legality of this integration touches upon fundamental questions of copyright law, license interoperability, and the specific statutory obligations imposed by the "permissive" class of open-source licenses.

This report provides an exhaustive analysis of the legal and operational compliance status of QForge. Based on a deep forensic audit of the project's structure and its dependency tree, the utilization of these libraries is fundamentally **legal and license-compatible**. The architecture of QForge, which appears to function as a high-level command-line interface (CLI) and orchestration layer<sup>1</sup>, aligns with the intended use cases of the underlying libraries. The choice of the Apache License 2.0 for the host project is strategically sound, offering a robust patent grant and compatibility with the BSD-licensed components.<sup>2</sup>

However, "legal admissibility" must not be conflated with "compliance completeness." A rigorous examination of the provided research materials reveals distinct gaps in the current implementation of QForge's attribution mechanisms. Specifically, the absence of a **NOTICE** file within the QForge repository<sup>1</sup> represents a potential violation of Section 4(d) of the Apache License 2.0, particularly given the likelihood that upstream dependencies like Qiskit contain their own NOTICE files.<sup>4</sup> Furthermore, while the **BSD 3-Clause License** governing scqubits and QuTiP is permissive, its "Non-Endorsement" clause (Clause 3) and strict attribution requirements for "binary" or documentation distribution create nuances that simple pip installation manifests do not satisfy.<sup>6</sup>

The following sections dissect these obligations in granular detail. We explore the jurisprudential differences between "source" and "binary" distribution in the Python interpretation ecosystem, the specific "moral rights" ingrained in academic software like scqubits, and the corporate compliance structures necessitated by IBM-backed tools like Qiskit. This analysis serves not only as a verification of legality but as a roadmap for elevating QForge from a functional prototype to a professionally compliant open-source citizen.

---

## 2. The Jurisprudence of Open Source Licensing in Scientific Computing

To fully adjudicate the legality of QForge's dependency usage, one must first establish the legal theory governing the interaction between the host software (QForge) and the consumed libraries. Unlike the rigid statutory frameworks of patent law, open-source licensing operates on a contract-based model derived from copyright law. The validity of QForge's distribution depends entirely on adhering to the conditions set forth by the licensors of its dependencies.

### 2.1 The Permissive License Spectrum

The quantum software domain has largely rejected "copyleft" licenses (such as the GNU GPL) in favor of "permissive" licenses. This shift is evident in the licensing choices of the major players analyzed here: scqubits (BSD), QuTiP (BSD), and Qiskit (Apache).

- **The BSD Paradigm (scqubits, QuTiP):** The BSD 3-Clause License represents the "academic" tradition of software freedom. Originating from the University of California, Berkeley, its primary focus is minimizing friction for researchers. It allows for unlimited redistribution and modification, provided that credit is given. It is legally "short," relying on general copyright principles rather than exhaustive definitions.<sup>3</sup>
- **The Apache Paradigm (QForge, Qiskit):** The Apache License 2.0 represents the "corporate" tradition. Developed by the Apache Software Foundation, it addresses the ambiguities of the BSD/MIT style, specifically regarding **patents**. It contains explicit definitions of "Contributor," "Licensor," and "Derivative Works," and includes a patent grant that survives the copyright license. This makes it the preferred vehicle for industrial quantum efforts like IBM's Qiskit, where patent portfolios are as valuable as the code itself.<sup>2</sup>

### 2.2 The Doctrine of Compatibility

The central legal question is whether QForge (Apache 2.0) can legally subsume code or functionality from BSD-3 and Apache-2.0 sources. The answer is affirmatively rooted in the concept of **license compatibility**.

The Apache Software Foundation (ASF) classifies the BSD 3-Clause License as "**Category A**,"

meaning it is fully compatible for inclusion in Apache-licensed works.<sup>2</sup> The logic follows a "least restrictive" hierarchy:

1. **BSD 3-Clause** imposes three requirements: retain copyright notices, retain license conditions, and do not use names for endorsement.
2. **Apache 2.0** imposes similar requirements but adds patent clauses and the NOTICE file mechanism.
3. **Synthesis:** A developer can satisfy the BSD obligations *within* an Apache-licensed project. By preserving the BSD copyright headers in the specific files (or documentation) and refraining from using the authors' names for promotion, the Apache project does not violate the BSD terms. Conversely, the BSD license places no restrictions on patents that would conflict with the Apache patent grant.<sup>9</sup>

Consequently, the "inbound" legality is clear: QForge is permitted to link against, import, and even bundle scqubits and QuTiP without changing its own license from Apache 2.0. There is no "viral" effect (as seen in the GPL) that would force QForge to adopt the BSD license for its own logic, nor does the BSD license prohibit the stronger patent terms of the Apache wrapper.

## 2.3 The "Derivative Work" Complexity in Python

A nuanced area of compliance arises from the technical nature of Python. Unlike C++ or Go, where code is often compiled into a single static binary, Python projects typically resolve dependencies at runtime.

- **The "Linking" Argument:** Some legal interpretations suggest that merely importing a library does not create a "Derivative Work" in the copyright sense, as the code bases remain separate on the disk. Under this view, QForge is merely an "aggregate" work.
- **The "Functional" Argument:** However, if QForge is a "wrapper" or "toolkit" that exposes the underlying objects of scqubits or Qiskit directly to the user, or if QForge cannot function *at all* without them, it is functionally a derivative work. The Apache License 2.0 defines "Derivative Works" broadly to include any work "based on (or derived from) the Work".<sup>4</sup>
- **Conservative Compliance:** For the purpose of this report, we adopt the conservative stance: QForge should be treated as a Derivative Work (or at least a Collective Work) that triggers the attribution requirements of its components. Relying on technicalities about "dynamic linking" to avoid giving credit is legally risky and professionally discourteous in the open-source community.<sup>11</sup>

---

## 3. Architectural Analysis of QForge

To understand the specific compliance triggers, we must analyze how QForge physically interacts with its dependencies. The research snippets paint a picture of a CLI-based tool

designed to democratize access to quantum physics simulations.

### 3.1 The Toolkit Architecture

QForge is described as a "comprehensive, terminal-based quantum simulation toolkit".<sup>1</sup> Its workflow involves:

1. **Orchestration:** The user issues commands like `qforge qubit create` or `qforge gate simulate`.
2. **Delegation:** QForge translates these high-level commands into API calls to the underlying libraries.
  - **Physics:** Calls `scqubits` to calculate energy spectra and coherence times ( $T_1, T_2$ ).<sup>1</sup>
  - **Dynamics:** Calls `QuTiP` to simulate time-domain gate operations (e.g., Rabi oscillations).<sup>1</sup>
  - **Hardware:** Intends to call `Qiskit Metal` for chip layout.<sup>1</sup>
3. **Presentation:** QForge aggregates the results and presents them via a "interactive CLI," "wizards," or "terminal plotting".<sup>1</sup>

### 3.2 Dependency Management Mechanism

QForge uses standard Python packaging protocols. It defines dependencies in `pyproject.toml` or `requirements-core.txt`.<sup>1</sup>

- **The "Referencing" Model:** QForge does not appear to "vendor" (copy-paste) the source code of `scqubits` or `Qiskit` into its own repository. Instead, it instructs the user's package manager (`pip`) to download them.
- **Legal Implication:** This reduces the burden of *source code* redistribution. QForge is not distributing the `scqubits` source code; `PyPI` is. However, QForge *is* distributing a tool that causes the user to execute that code. Furthermore, if QForge is distributed as a pre-built Docker container or a binary executable (e.g., using `PyInstaller`), the "referencing" defense vanishes, and QForge becomes a direct distributor of the binaries, triggering all license clauses immediately.<sup>12</sup>

### 3.3 The User Interface as a Compliance Surface

The "Interactive CLI" and "Wizard" features<sup>1</sup> create a unique compliance surface. In a graphical user interface (GUI), standard practice is to have a "Help > About > Open Source Licenses" menu. In a CLI, this is often overlooked.

- **The Risk:** If a user runs `qforge --interactive`, they are interacting with `scqubits` logic. If the CLI claims "Calculated by QForge" without acknowledging `scqubits`, it edges close to infringing the "Non-Endorsement" or attribution spirit of the licenses. The CLI must be transparent about the engines it drives.<sup>11</sup>
-

## 4. Deep Dive: The BSD 3-Clause Dependencies (scqubits & QuTiP)

The BSD 3-Clause License governs two of QForge's most critical physics engines: **scqubits** and **QuTiP**. While "permissive," this license is not "condition-free." It imposes specific obligations that QForge must actively satisfy to remain legal.

### 4.1 scqubits (Superconducting Qubits in Python)

**scqubits** is the physics engine for QForge's qubit modeling features (Transmon, Fluxonium, etc.).

#### 4.1.1 License Analysis

- **Licensor:** Jens Koch and Peter Groszkowski.<sup>6</sup>
- **Term:** Copyright (c) 2019 and later.
- **Conditions:**
  1. *Redistribution of Source Code:* Must retain the copyright notice.
  2. *Redistribution in Binary Form:* Must reproduce the copyright notice, list of conditions, and disclaimer in the **documentation and/or other materials provided with the distribution**.<sup>6</sup>
  3. *Non-Endorsement:* Neither the name of the copyright holder nor the names of contributors may be used to endorse products derived from this software.<sup>6</sup>

#### 4.1.2 The "Binary Distribution" Trap

Since QForge is a Python package, users typically install it as a "binary" (wheel) or run it as bytecode. This triggers Clause 2.

- **Unsatisfied Requirement:** QForge's documentation (README, ReadTheDocs) does not appear to explicitly list the full license text of scqubits. Simply listing it in requirements.txt is legally insufficient for Clause 2 compliance. The license text must be *reproduced* in the materials provided to the user.<sup>3</sup>
- **Remediation:** QForge must include a section in its documentation (e.g., "Acknowledgments" or "Third-Party Licenses") that explicitly pastes the BSD 3-Clause text and the copyright line: "Copyright (c) 2019 and later, Jens Koch and Peter Groszkowski."

#### 4.1.3 The Academic Citation Obligation

Beyond the legal text, scqubits carries a strong "moral rights" request typical of academic software.

- **The Request:** Snippet <sup>15</sup> and <sup>16</sup> state: *"If you employ scqubits in your research, please support its continued development and maintenance. Use of scqubits in research*

*publications is appropriately acknowledged by citing:... Quantum 5, 583 (2021)."*

- **Legal vs. Ethical:** While the BSD license text <sup>6</sup> does not explicitly make citation a condition of use (unlike some research-only licenses), ignoring this request is perilous. The QForge project relies on the goodwill of the scqubits maintainers. If QForge users publish papers citing QForge but *not* scqubits, the scqubits authors effectively lose their "payment" (citations).
- **Integration Strategy:** QForge should automate this compliance. When a user runs a simulation, the CLI could output a message: *"Results generated using scqubits. Please cite: Quantum 5, 583 (2021)."* This turns QForge from a citation-cannibal into a citation-driver, aligning incentives.

## 4.2 QuTiP (Quantum Toolbox in Python)

QuTiP provides the time-domain simulation capabilities for QForge.<sup>1</sup>

### 4.2.1 License and Copyright Complexity

- **Licensors:** "QuTiP developers and contributors".<sup>17</sup>
- **History:** Originating at the RIKEN Institute in Japan <sup>18</sup>, QuTiP has a massive contributor base.
- **Documentation License:** Interestingly, QuTiP's *documentation* is licensed under **Creative Commons Attribution 3.0 (CC-BY 3.0)**, while the *source code* is **BSD 3-Clause**.<sup>7</sup>
- **Implication for QForge:** If QForge copies snippets of text, tutorials, or explanations from the QuTiP documentation into its own help files or wizards, it **must** comply with CC-BY 3.0. This requires a different attribution format (linking to the original author and license) than the BSD code license.
  - *Warning:* Mixing CC-BY content into an Apache 2.0 documentation set can be tricky. It is safer to link to QuTiP documentation rather than copying it.

### 4.2.2 The "Non-Endorsement" Risk

QuTiP is a dominant brand in the field.<sup>19</sup> Clause 3 of the BSD license strictly prohibits using the name "QuTiP" to endorse QForge.

- **Permissible Use:** "QForge integrates with QuTiP for dynamics simulation." (Statement of fact).
  - **Impermissible Use:** "QForge is the ultimate QuTiP experience," or using the QuTiP logo on the QForge landing page in a way that implies a partnership.
  - **Snippet Insight:** QForge's description "bridge the gap... utilizing QuTiP dynamics" <sup>1</sup> is currently on the safe side, but marketing copy must be policed carefully to avoid crossing into implied endorsement.
-

## 5. Deep Dive: The Apache 2.0 Dependencies (Qiskit & Qiskit Metal)

The **Apache License 2.0** governs **Qiskit** and **Qiskit Metal**. While QForge shares this license, the interoperability requirements are actually *more* rigorous than with BSD due to the formalization of the **NOTICE** file mechanism.

### 5.1 Qiskit (Quantum Information Science Kit)

Qiskit is the backbone for circuit-level operations in QForge.<sup>1</sup> It is backed by IBM, a corporation with stringent intellectual property enforcement protocols.

#### 5.1.1 The Section 4(d) "NOTICE" Requirement

This is the single most critical compliance detail for the QForge project. Section 4(d) of the Apache License 2.0 states:

*"If the Work includes a "NOTICE" text file as part of its distribution, then any Derivative Works that You distribute must include a readable copy of the attribution notices contained within such NOTICE file..."*<sup>4</sup>

- **Evidence of NOTICE Files:** Research snippet<sup>5</sup> reveals the contents of a NOTICE.txt file for a Qiskit-related repository (qiskit-qir), which includes third-party attributions (e.g., importlib-metadata Copyright 2017-2019). Snippet<sup>20</sup> shows LICENSE-DOCS for Qiskit documentation. It is virtually certain that the core qiskit library contains a NOTICE file aggregating the copyrights of its hundreds of dependencies.
- **The Compliance Gap:** The QForge repository currently contains a LICENSE file but **no NOTICE file**.<sup>1</sup>
- **The Violation:** By distributing QForge (a work based on Qiskit APIs), the user is failing to propagate the mandatory attribution notices from IBM. This is a technical violation of the license that could legally invalidate QForge's right to use Qiskit.
- **Immediate Action:** The user must locate the NOTICE file in the installed qiskit package (usually found in site-packages/qiskit-<version>.dist-info/NOTICE) and copy its relevant contents into a new NOTICE file in the root of the QForge repository.

#### 5.1.2 Patent Retaliation (Section 3)

The Apache License 2.0 includes a patent termination clause. If QForge (or its creators) were to sue IBM for patent infringement regarding the Qiskit code, QForge's license to use Qiskit would immediately terminate.<sup>4</sup> This is a defensive mechanism that stabilizes the ecosystem but binds QForge to a "patent peace" pact with IBM.

#### 5.1.3 Trademark Policy (Section 6)



Snippet <sup>4</sup> explicitly excludes trademarks from the grant. QForge must be careful with the "Qiskit" brand.

- **IBM's Guidelines:** IBM generally allows "referential use" (e.g., "Compatible with Qiskit") but disallows use that suggests ownership.
- **Project Naming:** "QForge" is a safe name. Names like "Qiskit-Forge" or "PyQiskit-Tool" would likely trigger a trademark cease-and-desist from IBM's legal team unless explicitly authorized.<sup>21</sup>

## 5.2 Qiskit Metal

**Qiskit Metal** is used for the hardware design component of QForge.<sup>1</sup>

### 5.2.1 Evolution and Governance

Qiskit Metal is transitioning from an IBM-led project to a community-governed project under the "Quantum Device Consortium".<sup>23</sup>

- **Legal Stability:** Despite the governance change, the license remains Apache 2.0.<sup>24</sup>
- **Attribution Nuance:** As the project moves to community ownership, the "Copyright" line in the license may change from "Copyright IBM" to "Copyright Qiskit Metal Development Team." QForge must ensure its attribution notices reflect the *current* version of the library it depends on. If QForge pins an older version (e.g., v0.4), it cites IBM. If it moves to v0.5+, it cites the Community.<sup>23</sup>

---

## 6. The Mechanics of Compliance: A Comparative Audit

The following table summarizes the specific compliance actions required for each dependency within the QForge ecosystem. This comparison highlights the "Missing" elements identified in the research snippets.

| Feature          | QForge (Host)  | scqubits (Dependency)   | QuTiP (Dependency) | Qiskit (Dependency) | Compliance Status |
|------------------|----------------|-------------------------|--------------------|---------------------|-------------------|
| License          | Apache 2.0     | BSD 3-Clause            | BSD 3-Clause       | Apache 2.0          | Compatible        |
| Copyright Holder | QForge Authors | J. Koch, P. Groszkowski | QuTiP Developers   | IBM / Contributors  | Diverse           |



|                        |                             |                                   |                      |                            |                              |
|------------------------|-----------------------------|-----------------------------------|----------------------|----------------------------|------------------------------|
| <b>NOTICE File</b>     | <b>MISSING</b> <sup>1</sup> | N/A (BSD)                         | N/A (BSD)            | <b>Exists</b> <sup>5</sup> | <b>VIOLATION (High Risk)</b> |
| <b>Attribution</b>     | In README                   | Required in Docs                  | Required in Docs     | Required in NOTICE         | <b>Incomplete</b>            |
| <b>Citation</b>        | N/A                         | <b>Requested</b> (Quantum 5, 583) | Requested            | N/A                        | <b>Ethical Gap</b>           |
| <b>Non-Endorsement</b> | N/A                         | <b>Strict Clause</b>              | <b>Strict Clause</b> | Trademark Policy           | <b>Monitor Needed</b>        |

## 6.1 The "NOTICE" File Implementation

To satisfy the "unsatisfied requirement" identified in the comparison, QForge must create a file named NOTICE in its root directory.

- **Format:** Plain text.
- **Content Construction:**  
QForge: Quantum Simulation Toolkit  
Copyright[QForge Owner]  
This product includes software developed by the Qiskit project (<https://qiskit.org>).  
Copyright IBM Corporation and others.  
This product includes software developed by the Qiskit Metal project.  
Copyright Qiskit Metal Development Team.
- **Why This Matters:** The Apache license places the burden of *downstream* notification on the distributor. If Qiskit credits a third-party math library in its NOTICE file, and QForge hides that file, Qiskit is compliant but QForge is not. The chain of credit is broken at QForge.

## 6.2 Handling "Requirements.txt" Legalities

The user asked about using libraries "the way I have" (via requirements files).

- **The "Pip Install" Defense:** Legally, providing a requirements.txt does not constitute "distributing" the library code itself. However, it constitutes "contributory" usage. The moment QForge is installed and runs, it creates a memory image that combines QForge and its dependencies.
- **The "Best Practice" Approach:** Even if the *strict* letter of the law might exempt a requirements.txt repo from distributing license texts (since they don't distribute the code), the *professional* standard in open source is to acknowledge dependencies. This protects the project against "Contributory Infringement" claims and demonstrates due

diligence.<sup>9</sup>

---

## 7. Academic Ethics and Community Norms

In the quantum domain, legal compliance is often secondary to **academic compliance**. The research snippets highlight that libraries like scqubits and QuTiP are products of research grants and academic careers.<sup>15</sup>

### 7.1 The "Currency" of Citations

Snippet <sup>15</sup> explicitly frames the "price" of using scqubits: *"If you employ scqubits in your research, please support its continued development... by citing."*

- **Risk of Non-Compliance:** If QForge abstracts away the scqubits interface so effectively that users forget scqubits exists, QForge actively harms the funding prospects of the scqubits authors. This can lead to the "sherlocking" phenomenon, where a wrapper tool gets the credit for the hard physics work of the engine.
- **Mitigation:** QForge should implement a `qforge --citations` command that outputs the BibTeX entries for all used libraries. This feature is a hallmark of "good citizenship" in scientific software and ensures the project is embraced rather than shunned by the community.<sup>16</sup>

### 7.2 Funding Acknowledgments

Both scqubits and QuTiP list "Funding" sections in their documentation.<sup>26</sup> While QForge is not required to list these funds, acknowledging the *ecosystem* funding (e.g., "Built on tools supported by the DOE and RIKEN") adds credibility to QForge, signaling that it rests on validated, publicly funded foundations.

---

## 8. Strategic Recommendations and Roadmap

The analysis confirms that the QForge project is fundamentally legal but administratively immature regarding compliance. To bridge the gap between "working code" and "compliant product," the following roadmap is prescribed.

### 8.1 Immediate Remediation Steps (0-30 Days)

1. **Audit Qiskit NOTICE:** Install qiskit and qiskit-metal locally. Locate their NOTICE files. Create a NOTICE file in QForge that aggregates these texts.
2. **Create "ThirdPartyNotices.md":** Create a document listing all dependencies (scqubits, QuTiP, Qiskit) and pasting their full license texts (BSD-3 and Apache-2.0). This satisfies the BSD "binary distribution" clause.

3. **Update README:** Add a distinct "Acknowledgments" section. Explicitly name scqubits, QuTiP, and Qiskit.
4. **Add Citation Prompts:** Add a note in the documentation asking users to cite the underlying libraries if they use QForge for publication.

## 8.2 Architectural Improvements (Long Term)

1. **CLI Attribution:** When the CLI starts up, print a brief log line: *Running on scqubits vX.X and Qiskit vY.Y*. This creates a persistent "audit trail" for the user.
2. **Dependency Pinning:** In requirements.txt, pin versions (e.g., `scqubits>=3.0,<4.0`). This protects QForge from a sudden license change in a major version update (e.g., if a library switched to GPL).<sup>28</sup>

## 8.3 Conclusion

The user's architecture for QForge represents a best-in-class approach to modern scientific software engineering: modular, open, and standards-based. The use of **Apache 2.0** as the host license provides a robust legal container that is fully compatible with the **BSD 3-Clause** and **Apache 2.0** dependencies. By addressing the identified gaps—specifically the missing **NOTICE** file and the formalized attribution of academic moral rights—QForge can mitigate its legal risks and solidify its reputation as a transparent, collaborative, and essential tool in the quantum simulation stack.

### Works cited

1. Ingenio17/qforge - GitHub, accessed on February 5, 2026, <https://github.com/Ingenio17/qforge>
2. ASF 3rd Party License Policy | Apache Software Foundation, accessed on February 5, 2026, <https://www.apache.org/legal/resolved.html>
3. Open Source Software Licenses 101: The BSD 3-Clause License | FOSSA Blog, accessed on February 5, 2026, <https://fossa.com/blog/open-source-software-licenses-101-bsd-3-clause-license/>
4. qiskit/LICENSE.txt at main - GitHub, accessed on February 5, 2026, <https://github.com/Qiskit/qiskit/blob/main/LICENSE.txt>
5. qiskit-qir/NOTICE.txt at main - GitHub, accessed on February 5, 2026, <https://github.com/microsoft/qiskit-qir/blob/main/NOTICE.txt>
6. scqubits/LICENSE at main - GitHub, accessed on February 5, 2026, <https://github.com/scqubits/scqubits/blob/master/LICENSE>
7. Copyright and Licensing — QuTiP 4.7 Documentation - Read the Docs, accessed on February 5, 2026, <https://qutip.readthedocs.io/en/qutip-4.7.x/copyright.html>
8. Guide to Open Source Licenses: Use, Obligations, and Risk | Black Duck Blog, accessed on February 5, 2026, <https://www.blackduck.com/blog/open-source-licenses.html>
9. How should I append my license on file headers in case BSD 3-Clause to Apache

- 2.0?, accessed on February 5, 2026,  
<https://opensource.stackexchange.com/questions/12317/how-should-i-append-my-license-on-file-headers-in-case-bsd-3-clause-to-apache-2>
10. Licenses of dependencies - Developers - The Stan Forums, accessed on February 5, 2026, <https://discourse.mc-stan.org/t/licenses-of-dependencies/9295>
  11. How to properly manage Python application dependencies? - Stack Overflow, accessed on February 5, 2026,  
<https://stackoverflow.com/questions/75705140/how-to-properly-manage-python-application-dependencies>
  12. GNU GPL license in libraries in python requirements [duplicate], accessed on February 5, 2026,  
<https://opensource.stackexchange.com/questions/13824/gnu-gpl-license-in-libraries-in-python-requirements>
  13. Managing Application Dependencies - Python Packaging User Guide, accessed on February 5, 2026,  
<https://packaging.python.org/tutorials/managing-dependencies/>
  14. What are your preferred conventions for documenting python code? - Reddit, accessed on February 5, 2026,  
[https://www.reddit.com/r/Python/comments/zfbm0q/what\\_are\\_your\\_preferred\\_conventions\\_for/](https://www.reddit.com/r/Python/comments/zfbm0q/what_are_your_preferred_conventions_for/)
  15. Source code for scqubits.settings, accessed on February 5, 2026,  
[https://scqubits.readthedocs.io/en/v3.1\\_a/modules/scqubits/settings.html](https://scqubits.readthedocs.io/en/v3.1_a/modules/scqubits/settings.html)
  16. scQubits documentation — scqubits documentation, accessed on February 5, 2026, <https://scqubits.readthedocs.io/>
  17. QuTiP: Quantum Toolbox in Python - GitHub, accessed on February 5, 2026,  
<https://github.com/qutip/qutip>
  18. QuTiP - Wikipedia, accessed on February 5, 2026,  
<https://en.wikipedia.org/wiki/QuTiP>
  19. QuTiP - Quantum Toolbox in Python, accessed on February 5, 2026,  
<https://qutip.org/>
  20. documentation/LICENSE-DOCS at main · Qiskit/documentation - GitHub, accessed on February 5, 2026,  
<https://github.com/Qiskit/documentation/blob/main/LICENSE-DOCS>
  21. Contribution License Agreement - IBM Quantum Platform, accessed on February 5, 2026, <https://quantum.cloud.ibm.com/docs/open-source/qiskit-cla.pdf>
  22. Contributing to Qiskit Metal - Quantum Metal 0.5.2.post4, accessed on February 5, 2026, <https://qiskit-community.github.io/qiskit-metal/contributor-guide.html>
  23. qiskit-community/qiskit-metal: Quantum Hardware Design. Open-source project for engineers and scientists to design superconducting quantum devices with ease. - GitHub, accessed on February 5, 2026,  
<https://github.com/qiskit-community/qiskit-metal>
  24. qiskit-metal - PyPI, accessed on February 5, 2026,  
<https://pypi.org/project/qiskit-metal/>
  25. The Legal Side of Open Source, accessed on February 5, 2026,  
<https://opensource.guide/legal/>

26. Source code for `scqubits.core.qubit_base`, accessed on February 5, 2026, [https://scqubits.readthedocs.io/en/latest/\\_modules/scqubits/core/qubit\\_base.html](https://scqubits.readthedocs.io/en/latest/_modules/scqubits/core/qubit_base.html)
27. Superconducting Qubits in Python - scQubits documentation, accessed on February 5, 2026, [https://scqubits.readthedocs.io/en/v1.3\\_a/](https://scqubits.readthedocs.io/en/v1.3_a/)
28. Best Practices for Managing Python Dependencies - GeeksforGeeks, accessed on February 5, 2026, <https://www.geeksforgeeks.org/python/best-practices-for-managing-python-dependencies/>