The PlanetaryPy Project. K. Michael Aye[1], Ross A. Beyer[2,3], Andrew M. Annex[3], and Chase Million[4]. [1]University of Colorado, Boulder (michael.aye@colorado.edu), [2]SETI Institute, (rbeyer@seti.org), [3]NASA Ames Research Center, [3]Johns Hopkins University, [4]Million Concepts

What is it: The PlanetaryPy Project is a community effort to develop a core package for planetary science in Python and foster interoperability between Python planetary science packages.

The PlanetaryPy Technical Committee (TC) is the governing body for the PlanetaryPy Project. All of their work is done in the open via the TC GitHub repo. To learn more about what the TC is, how it operates, or to get involved, you can read its Charter (see paragraph below).

The PlanetaryPy TC has regular meetings to which all are welcome to attend. The meeting agendas for upcoming meetings, and notes from previous meetings can be found in the TC's GitHub repo.

History and Outlook: The concept of PlanetaryPy has existed since at least 2015. Austin Godber, and other developers created the planetarypy GitHub organization in 2015, and began maintaining several repositories there. There was no governance model, and no central *planetarypy* package. Many of these original developers moved on from planetary sciences, and active maintenance of these repositories waned.

In 2019, there was renewed interest by a new group of individuals (now the members of the PlanetaryPy TC) to create something for planetary sciences that was akin to the Astropy Project for astronomy.

At the Planetary Data Workshop in June, 2019, we found that there was a real interest in the Planetary Sciences technical community for an active planetary Python resource. Many people pointed to the Astropy Project as a possible analog or goal. The code within the planetarypy GitHub organization was a great starting point, and the naming was right.

The goal is to take that existing codebase, plus a solid governance model taken from the Planetary Software Organization, and eventually work to build something that is as robust as the Astropy Project. We are just beginning this process, and the intent is to slowly and methodically build towards a healthy Python ecosystem for planetary sciences.

Participation in PlanetaryPy activities is completely voluntary.

In the following we summarize content of the TC's main guiding documents, the Charter and Affiliated Packages.

Charter: The planetarypy TC charter is the foundational document for the community and the starting point to learn more about the standards and conduct guidelines that govern the community and enable productive and cordial collaboration. Below are some excerpts from our Charter:

The TC's responsibilities include:

- Technical direction
- Setting release dates
- Release quality standards
- Project governance and process
- Contribution policy
- GitHub repository hosting
- Conduct guidelines
- Maintaining the list of additional Collaborators

Membership in the TC can be applied for during any of the open monthly meetings, but collaboration on software packages does not require this formality.

The TC follows a consensus-seeking decision making model and follows a code of conduct that has been adapted from the Contributor Covenant (a code of conduct for open source communities). The PlanetaryPy Project is also affiliated with the Planetary Software Organization as a top-level project.

Affiliated packages: PlanetaryPy affiliated packages are an important aspect of the PlanetaryPy community. An affiliated package is a planetary science related Python package that is not part of the *planetarypy* core package, and is not managed by the project but is a part of the PlanetaryPy Project community. These packages demonstrate a commitment to PlanetaryPy's goals of improving reuse, interoperability, and interface standards for Python planetary science packages. In many (but not all) cases, affiliated packages also follow similar development processes and package templates as the core package.

PlanetaryPy affiliated packages are autonomous projects created and managed by their own authors or Technical Committees (TC), not the PlanetaryPy TC. These are primarily software projects of various kinds which already exist separately. They benefit from being PlanetaryPy affiliated packages in the following ways:

The PlanetaryPy TC helps them adopt or formalize governance and community standards to foster their existence as an open source community. They gain the ability to easily reach out to the PlanetaryPy TC for consultation or mentorship on technical or governance issues. Being recognized as a PlanetaryPy affiliated package lets others know that their project conforms to community best

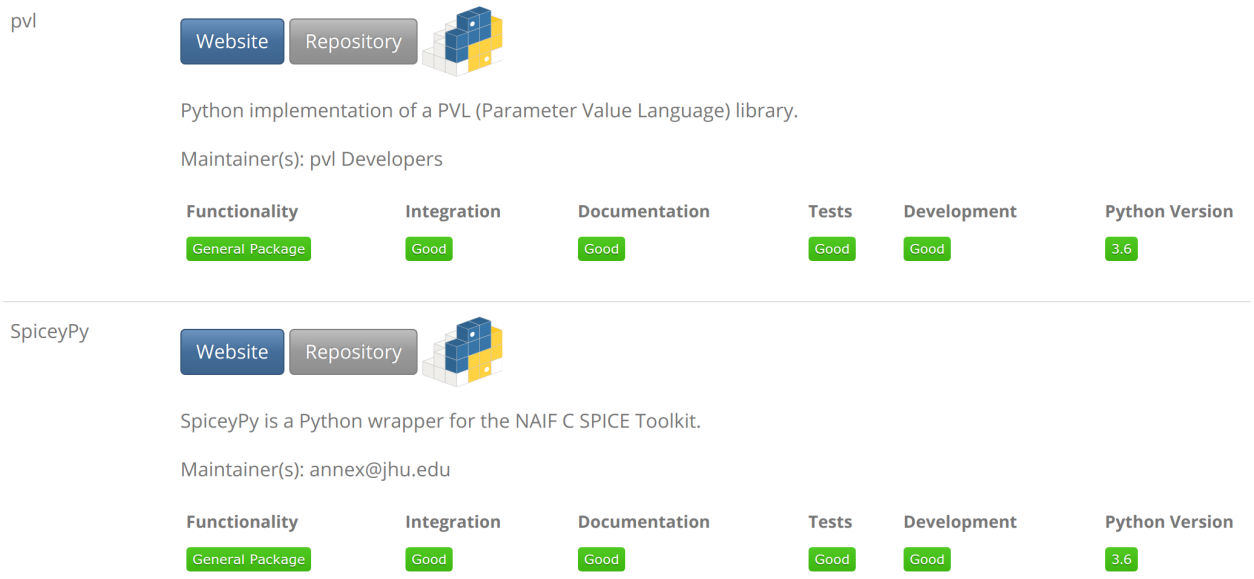

Figure 1: Current affiliated packages of the PlanetaryPy project. Our page listing indicates the status of aspects of the affiliated project evaluation.

practices which are likely to result in long-term success for their project.

Current affiliated packages of the project are the widely used packages *pvl* (for reading and writing PDS3 labels) and *SpiceyPy* (a widely used Python-based wrapper for SPICE) and their evaluation status are shown in Fig. 1.

Core package: One of our main goals for the core package (which is still in development) is to provide modules which enable access to the very diverse landscape of planetary data, and tools to work with it. We hope to soon release a beta version of our core modules for PDS data search via downloadable index files converted to pandas dataframes, and scripted download and local management of PDS data using patterns similar to the upcoming PDS API of the engineering node. Once the PDS API is itself out of the 'alpha' stage, we plan on making use of it in the core package as well. PDS3 data is very heterogeneous, and can be difficult to read with Python tools without specific knowledge of the data set. A recently-funded project, the Planetary Data Reader (*pdr*, Million et al., this conference) should address this shortcoming, and we expect it will eventually be embedded into the core *planetarypy* package as well.

How to get involved: If you're a Python user, we have a listing of PlanetaryPy Affiliated Packages that show good coding practices and good community practices. We're also working on developing a core *planetarypy* library. The OpenPlanetary Python Slack channel is a great place to engage with a community of other planetary Python users.

If you are a Python developer looking for ways to contribute to the planetary sciences, please consider contributing to our Core Library and Affiliated Packages. There are many ways to contribute to an open source project outside of just code, a good resource that lists many forms of contribution is available at opensource.guide/how-to-contribute.

If you are a Python developer with an existing package that you think others would benefit from, and would benefit the planetary science Python ecosystem, please consider applying to be an Affiliated Package.