



# WheelNext Community Event

Stan Seibert

March 21, 2025



## Disclaimer

Anaconda is the creator of both conda and Numba, *but those OSS projects have their own governance that is larger than Anaconda.*

I will try to communicate Anaconda's interests and my understanding of issues and history that involve conda and Numba, but I'm not offering official perspective or making decisions for those OSS projects.

(Prod me if it isn't clear what hat I'm wearing when I say something. 😁)

# What is Anaconda trying to build?

- **What we care about:**
  - Packaging and distributing open source software for data science, machine learning, AI, modeling, and scientific computing users. (“People who use numbers and stuff”)
  - The relationships between *packages*, *repositories*, *environments* and the different personas who build them.
- **Areas of focus for us:**
  - Consistent support across major desktop operating systems (Windows, macOS, Linux), and future support for mobile OSes.
  - Numerical performance: Requires specialization for specific hardware at a more granular level (“Linux x86-64” isn’t always good enough). This includes both GPU acceleration hardware and driver versions, as well as CPU instruction extensions (AVX and friends, or different levels of ARM).
  - Interop between conda and other Python packaging tools and formats.
  - *Non-Python packages*: We believe it is important to be able to treat not-Python libraries and the Python interpreter itself on co-equal footing with Python packages.

## Our Top 3 Areas of Interest for WheelNext

- **Variant handling:**

- This is something the community has to solve for in order to stay relevant for numerically intense computing. There are too many kinds of hardware now to ignore this.
- *Biggest concern:* Solving for the “best” variant when user intent is unclear. There might not be a total ordering of variants, and we don’t have a way for the user to signal what to prioritize.

- **External channels/indexes and priorities:**

- This feels like a necessary feature to go with variants (since official package maintainers may not be able or interested in building all variants) and a place to harmonize / learn from the conda community experience.
- *Biggest concern:* We’ll need to think about (and socialize) the security implications of priorities. The potential for spoofing one index from another needs to be thought through.

- **Specifying non-Python dependencies (build or runtime):**

- Should we pull PEP725 (Specifying external dependencies in pyproject.toml) into this effort? Seems very relevant as many numerical packages have external non-Python dependencies that need to be checked.

## Ways we could contribute

- Sharing our experience from trying to solve these issues with conda for the past 10+ years:
  - “Don’t reinvent the wheel” is going to be a frequent, unintentional pun at these meetings, but we did in fact reinvent “the wheel” (for good reasons), so we want to make sure our experience doesn’t have to be rediscovered.
- Help figure out ways to bring wheels and conda packages into better alignment. Conda’s scope will always be bigger than Python packaging, but the more we can use the same terms and definitions, the better.
- Self-service hosting of new format wheels on Anaconda.org:
  - Anaconda.org has had the ability to host wheels for a very long time.
  - Scientific python folks already host nightly wheels there.
  - What changes would we need to make to give everyone the ability to upload WheelNext wheels to Anaconda.org for easier testing and experimentation?