Wheel-Next: Red Hat PoV

Fabien Dupont, Jeremy Eder, Tom Gundersen, Christian Heimes, Doug Hellmann + teams

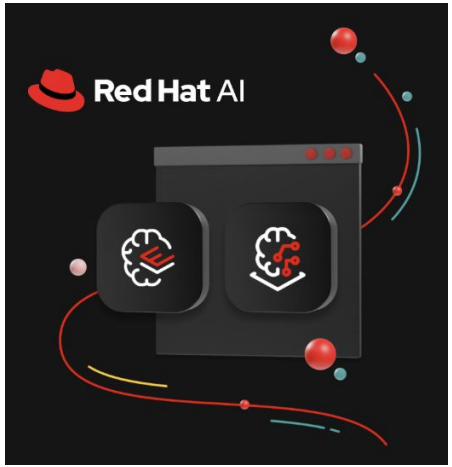
Red Hat

Do you think python is on the right path for GenAI?

If not, why not?

If so, why are we here?

Background and why Red Hat is here



- Red Hat is growing an AI portfolio.
  - Diverse-hardware strategy for accelerators
  - Enterprise lifecycle
  - Partner-centric strategy
  - Working with Quansight
- Products: RHEL AI & OpenShift AI & some IBM SaaS offerings & IBM Spyre
- Shipping Python packages as wheels in a venv, non-Python packages as RPMs

Rebuilding all packages from source

- **For technical and security reasons, we build all wheels we ship from source**
- Fromager is the tool we use to build a whole dependency tree as wheels
- Obtaining sources is sometimes challenging without sdists uploaded on PyPI
  - Andrew James from QS got us in touch with atalman about triton

Main challenges

- Python packaging has no awareness of accelerators
  - Needing multiple indexes for CUDA, ROCm, oneAPI wheels is cumbersome
  - No accelerator-aware installer
  - **We would like to be able to install certain wheels based on metadata we provide.**

- Achieving build hermeticity requires patching/workarounds in many Python packages
  - Ex.: packages unconditionally rely on CMake, which bypasses our system CMake
  - Ex.: packages invoke Make from within setup.py to download and build a shared library

Looking forward to adopting: wheel variants

- We have worked around the issues by running a **wheel index server for each hardware platform**
- The ability to store all wheels in a single index server and `pip install pkgname` doing the right thing will lower our maintenance & development costs

Note that we do not rely on PyPI, so we don't need PyPI-focused WheelNext proposals for RHEL AI, however we do recognize the value to the community and are supportive of the initiative.

Top 3 “works well”

- PEP 517 adoption & build backends: moving from custom setup.py to CMake, Meson. Yields faster & more reliable builds, easier to debug.
- Focus on static metadata & security - SPDX license expressions, trusted publishing
- Venv-based deployment

Top 3 “needs improvement”

- Making Python packages easier to consume for a distro:
  - **Allow using system libraries** ← vLLM
  - Don’t download anything during builds or test suite run
  - Don’t vendor other packages if you can avoid it
  - Tagging releases, providing sdists on PyPI or GitHub Releases
- Python packaging standards and tools got a lot better, but many (most?) packages don’t use them

WheelNext initiative & our planned contributions

- We plan to contribute by:
  - Sharing our use cases
  - Testing wheel variant prototypes
  - Supporting PEP discussions on the packaging Discourse
  - Any other asks that come our way and we may be in a position to help

Closing thoughts

- We see the WheelNext initiative as a **shared, community concern**, where a complete solution will reduce all our technical debt and maintenance burdens which multiply with each accelerator.
- It is also a way of moving the state of Python packaging forward, to where **RH can support AI stacks more seamlessly and fluidly as the ecosystems (plural) evolve.**