ANACONDA.

PyData Vermont 2026



From Chaos to Confidence

Solving Python's Environment Reproducibility Crisis

www.dawnwages.info/pydata-vt-2025

ANACONDA.

PyData Vermont 2026

>>WHOAMI?

Take the Python Packaging Survey



- Outgoing Chair of the Python Software Foundation Board
- Organization for Ethical Source Advocate
- Python Sofware Engineer
- Python event organizer
- WagtailCMS Core Team Member
- Director of Community & Developer Relations @ Anaconda

19k and counting...

www.dawnwages.info/pydata-vt-2025

ANACONDA.

PyData Vermont 2026

What makes a reproducible environment?

Non-deterministic builds, missing documentation, limitations of the tool chain, hidden state, platform and architecture considerations, sensitive versions.

“it works on my machine”



www.dawnwages.info/pydata-vt-2025

ANACONDA.

PyData Vermont 2026

First Problems

Domain problem and scope

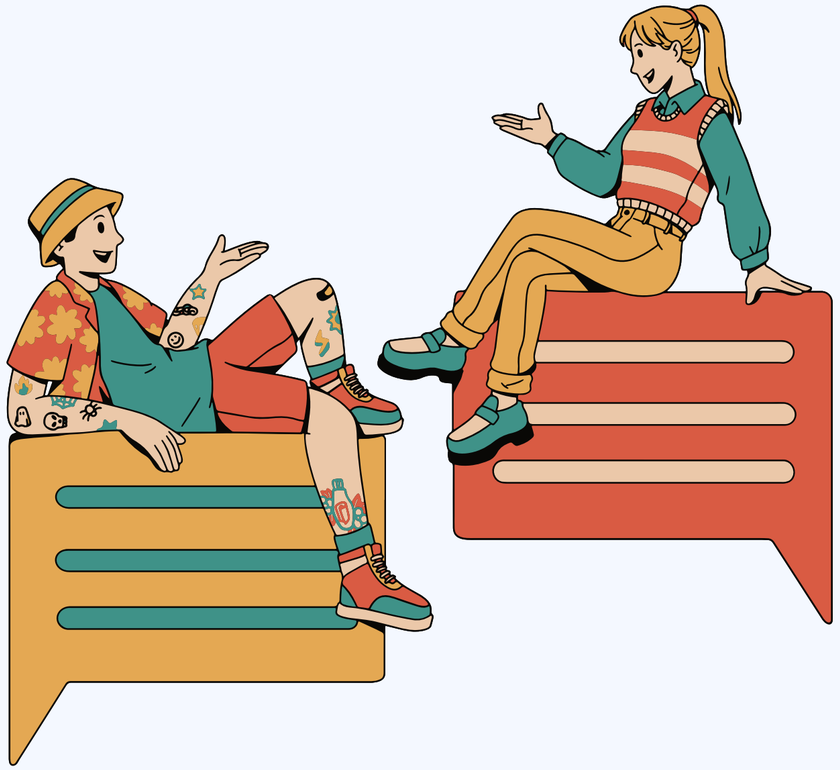
The first problems are communication driven. Gather your team together to brainstorm and keep communication lines open.

01 — Understanding the task

A top down view of what your end goal is and what it looks like is not easy. Talk with your peers and write down where you think the scope can creep in. **Start simple!**

02 — Active Collaboration

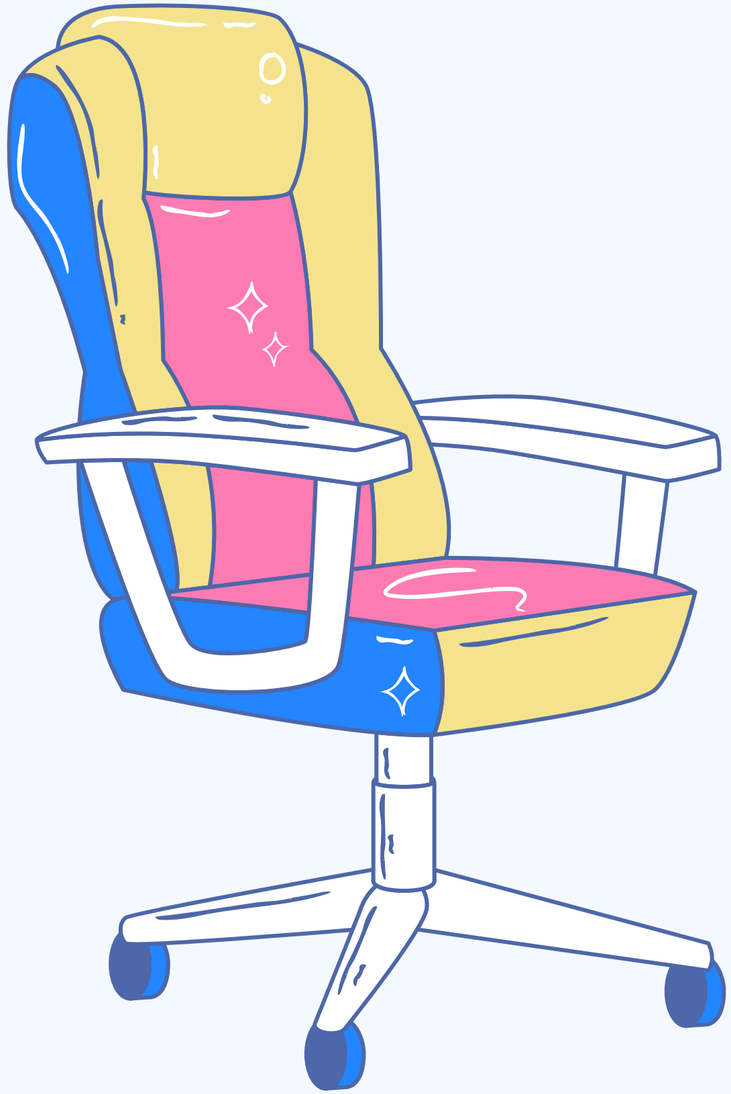
From that first spark of code, share actively. Prioritize creating pathways for potential collaborators can get your project up and running. **Early and often!**



www.dawnwages.info/pydata-vt-2025

ANACONDA.

PyData Vermont 2026



Making Decisions

Non-deterministic builds, missing documentation, limitations of the tool chain, hidden state, platform and architecture considerations, sensitive versions.

Picking the right tools for the growing environment complexity is ot trivial.

www.dawnwages.info/pydata-vt-2025

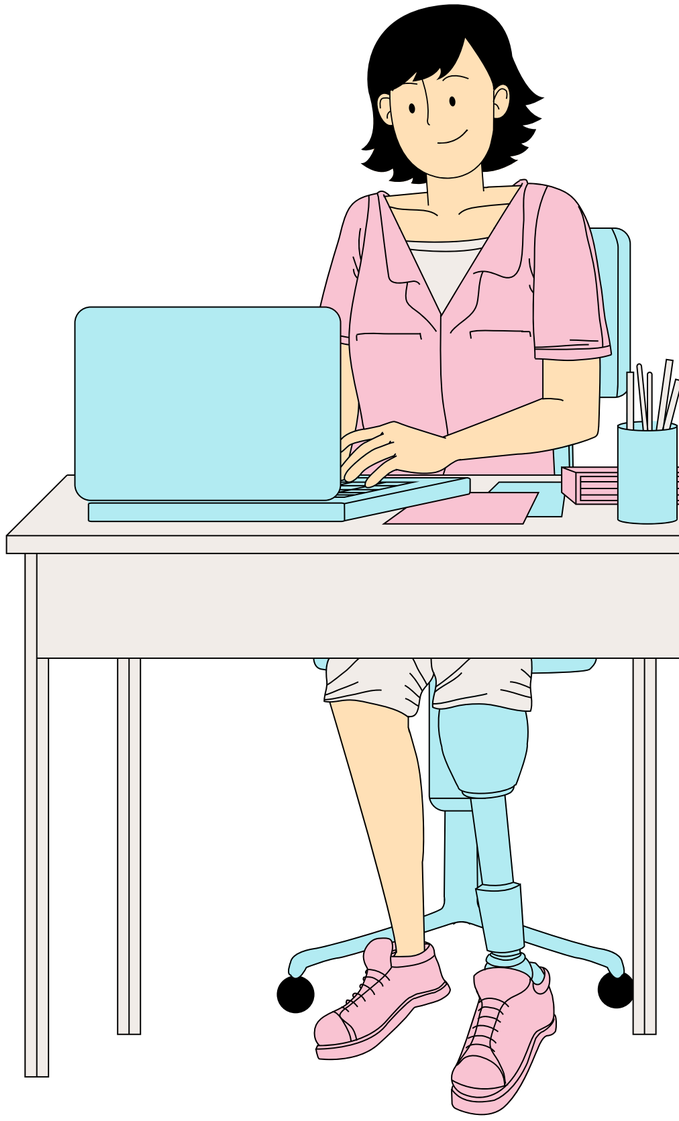
ANACONDA.

Environment Considerations

PyData Vermont 2026

Advanced **Environment Management** considerations

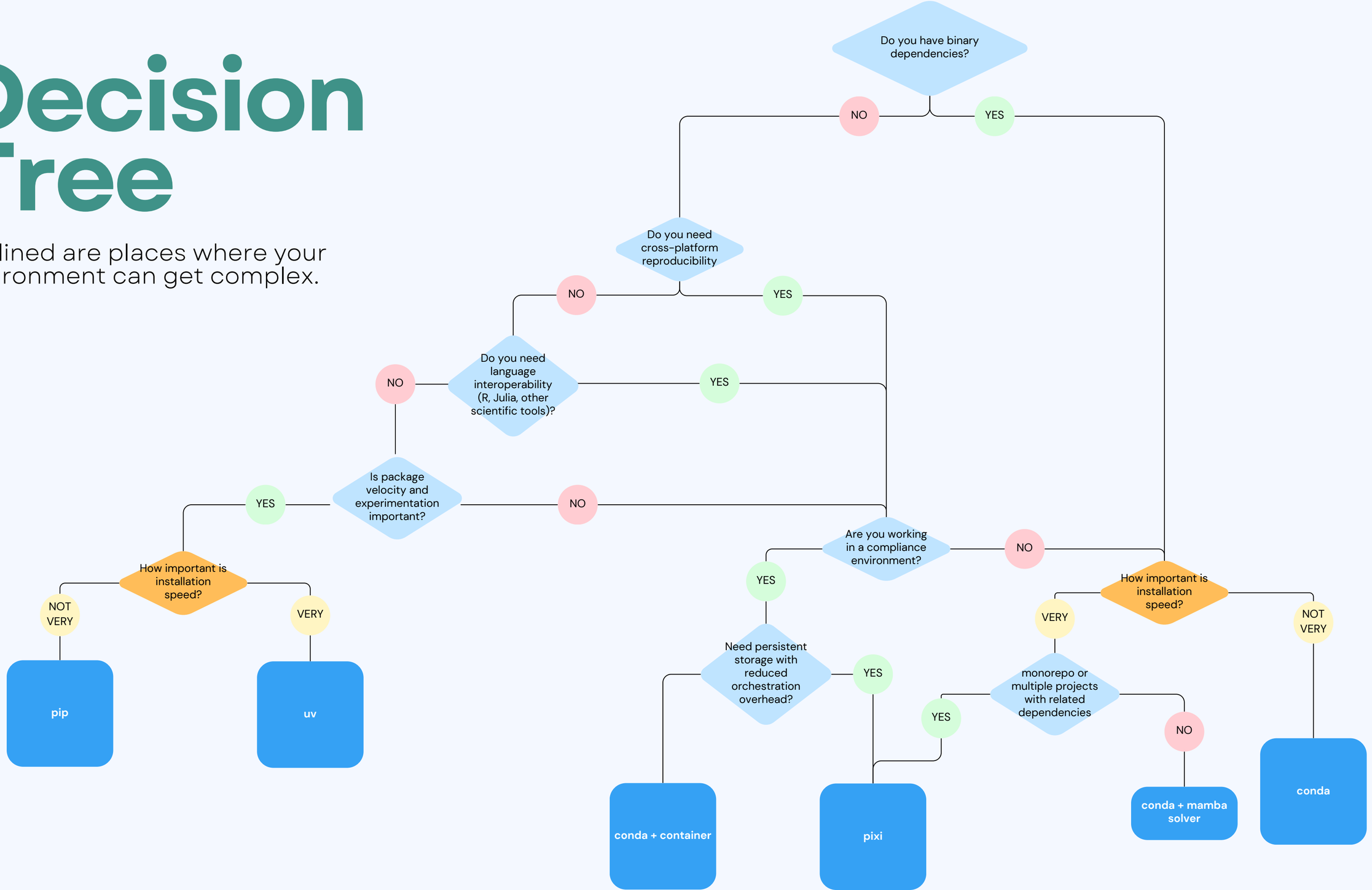
1. **Task complexity**
   building with complicated dependencies
2. **Hardware considerations**
   building with performance optimization and publishing dev tools
3. **Experimentation & collaboration**
   building with quick feedback cycles to share with peers
4. **Developer Tools**
   building your own developer tools, sharing with teams to leverage your work.
5. **Application Development**
   building resilient software systems



www.dawnwages.info/pydata-vt-2025

Decision Tree

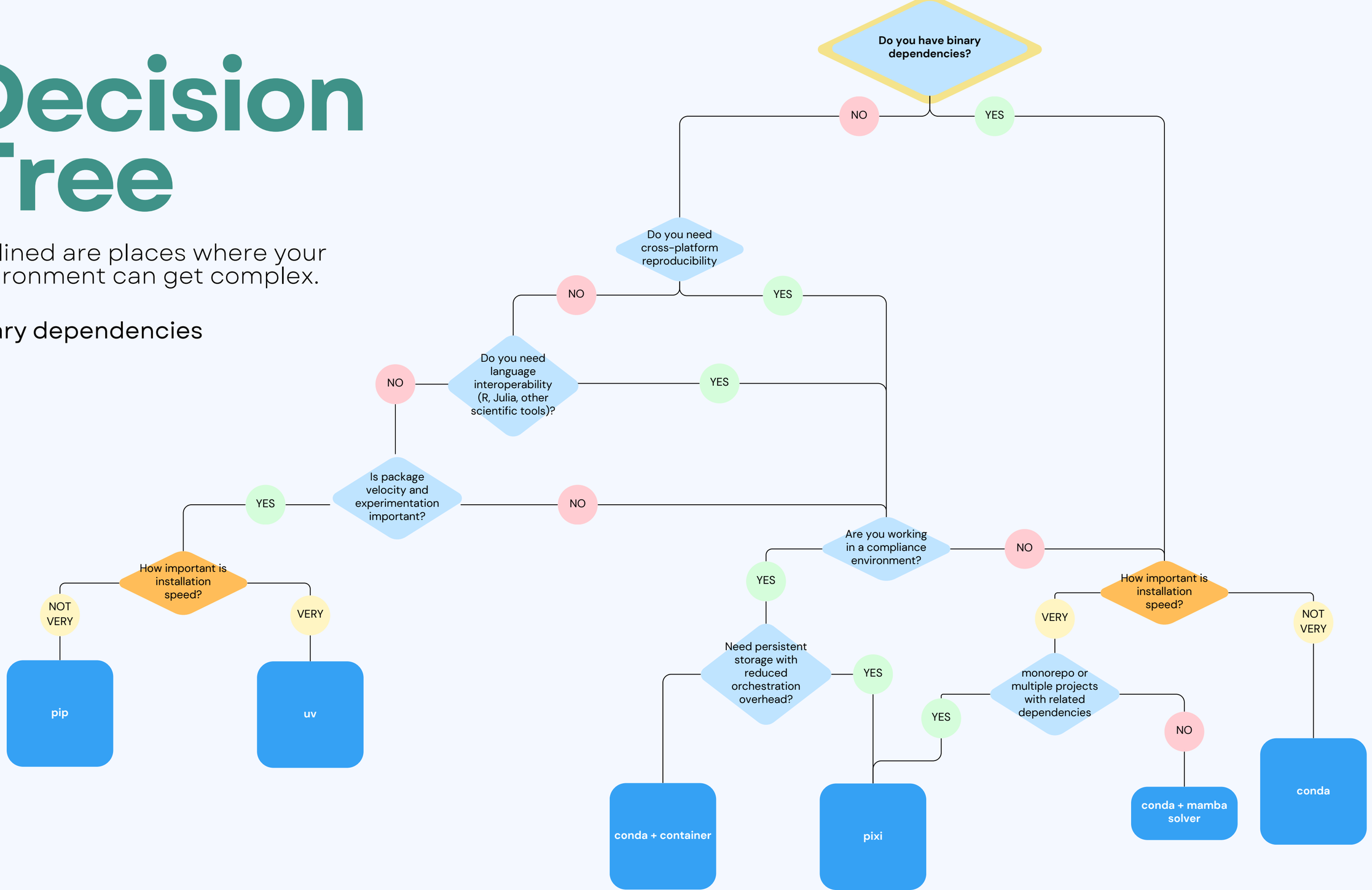
Outlined are places where your environment can get complex.



Decision Tree

Outlined are places where your environment can get complex.

Binary dependencies



ANACONDA.

PyData Vermont 2026

Binary Dependencies

What are they?

Python extensions with compiled (non-Python) code.

Python is an excellent glue language. This means that other languages will slip into your toolchain. A way around that while still having a pure python environment are binary dependencies with your packages.

www.dawnwages.info/pydata-vt-2025



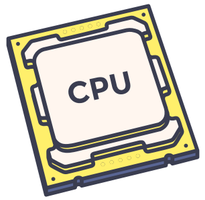
Why do they make environments difficult?

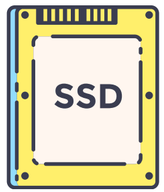
PyPI / pip / uv handle wheels compiled with C/C++ Rust code well, common packages like numpy, pandas and Pillow or self-contained binaries

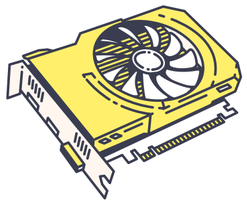
ANACONDA.

PyData Vermont 2026

Let’s talk about hardware

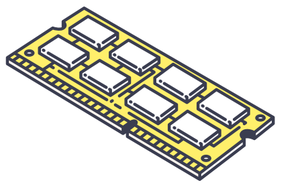


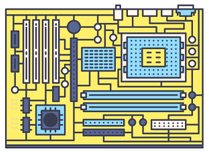










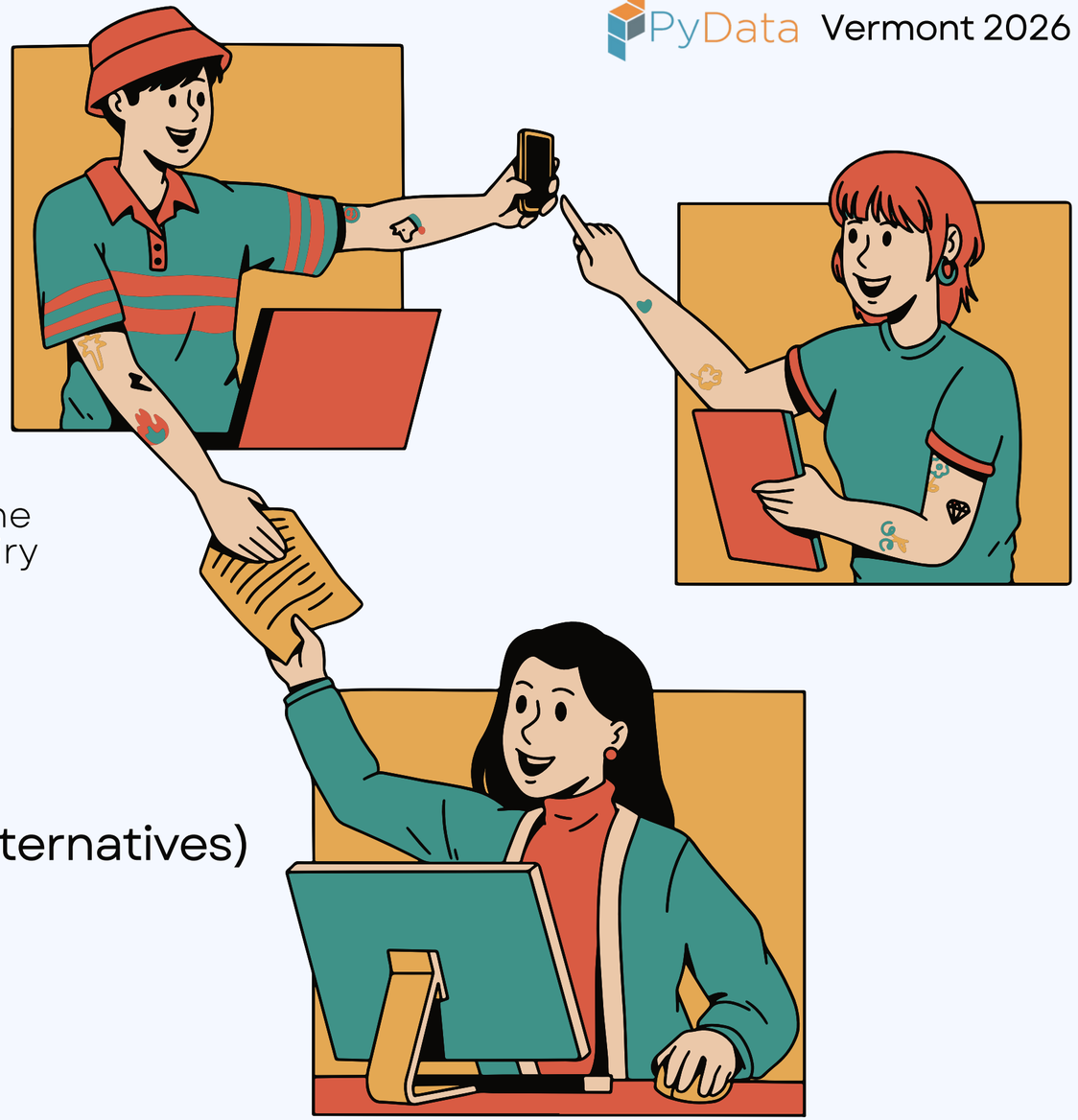


www.dawnwages.info/pydata-vt-2025

ANACONDA.

PyData Vermont 2026

Passing of responsibilities



The division of responsibility is not consistent on handling the gaps in compatibility across toolchain and has led to the hairy system.

01 — System Libraries (System)

compiler, system libraries and build tools

02 — Wheel / Conda format Builders (many alternatives)

Python extensions, optional bundled libraries

02 — Requirements (User)

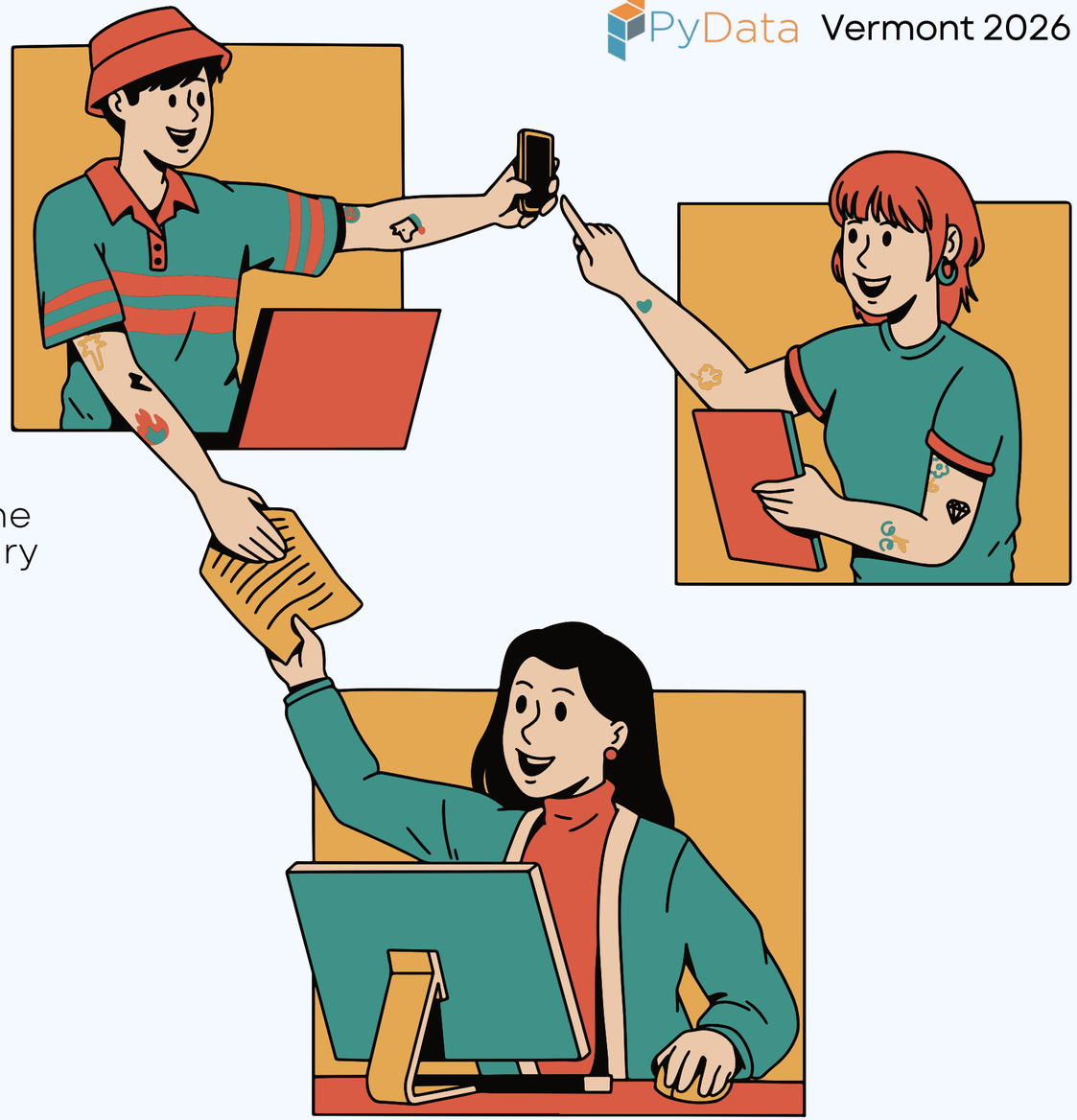
installing system pre-requisites

www.dawnwages.info/pydata-vt-2025

ANACONDA.

PyData Vermont 2026

Passing of responsibilities



The division of responsibility is not consistent on handling the gaps in compatibility across toolchain and has led to the hairy system.

01 — System Libraries (Conda)

compiler, system libraries and build tools

02 — Wheel / Conda format Builders (Conda)

Python extensions, optional bundled libraries

02 — Requirements (Conda)

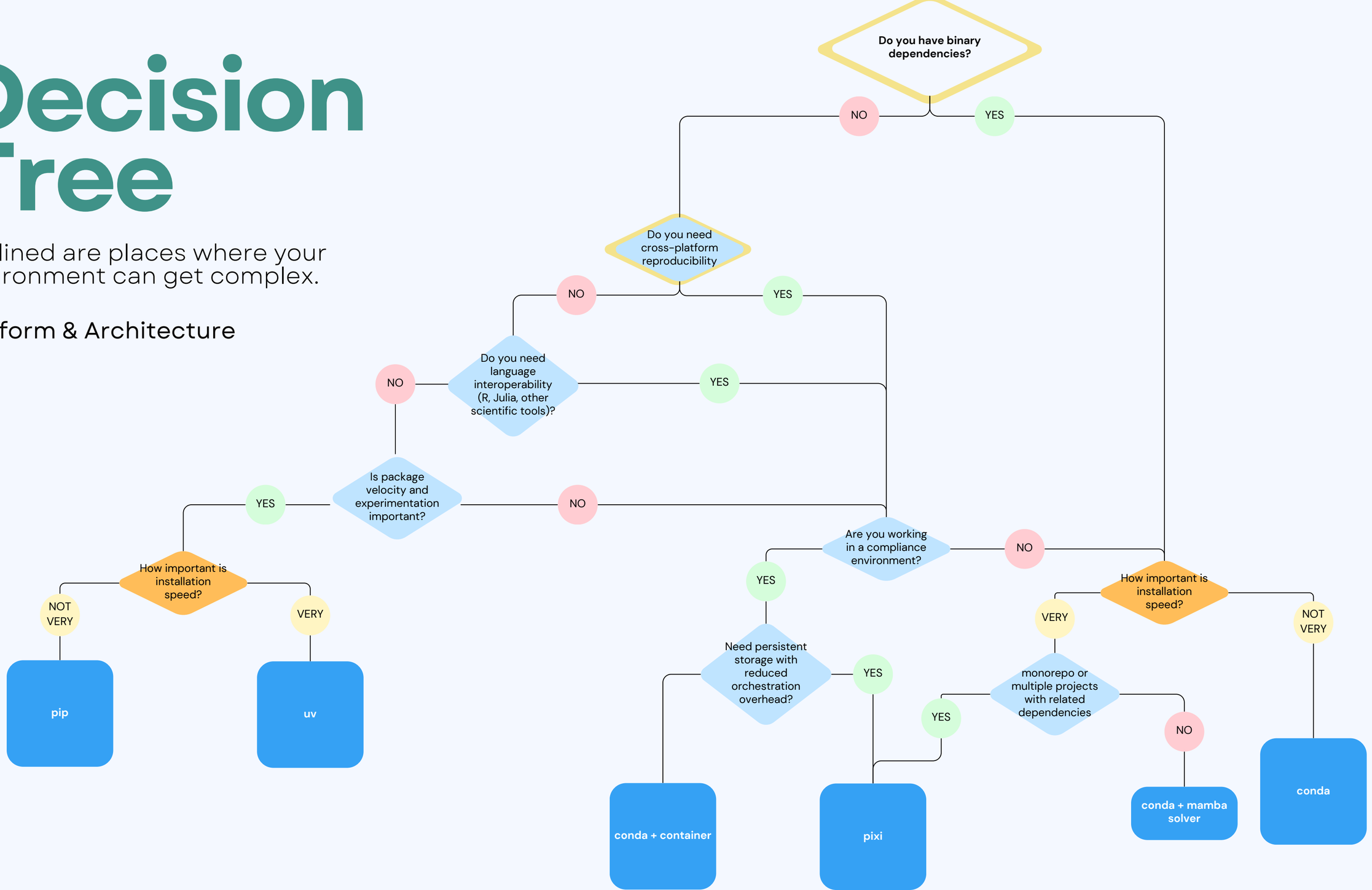
installing system pre-requisites

www.dawnwages.info/pydata-vt-2025

Decision Tree

Outlined are places where your environment can get complex.

Platform & Architecture

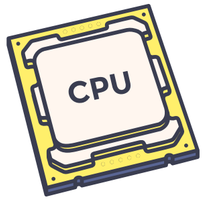


ANACONDA.

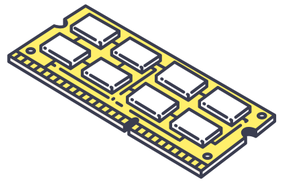
PyData Vermont 2026

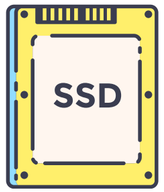
Architecture

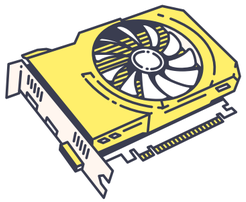
When solving packaging considerations, especially in compute intensive scenarios like scientific comptuing we have to consider what architecture you're using.

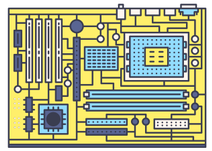












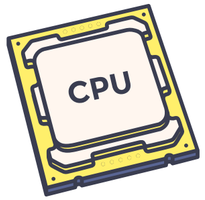
www.dawnwages.info/pydata-vt-2025

ANACONDA.

PyData Vermont 2026

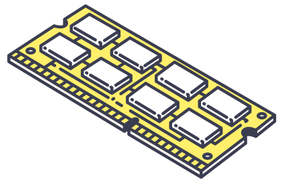
Architecture

When solving packaging considerations, especially in compute intensive scenarios like scientific comptuing we have to consider what architecture you're using.



Operating System

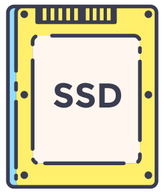


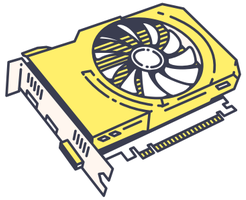


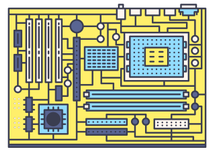
00 — Operating System

Mac, Linux, Window

Requires different commands to interact with the machine.







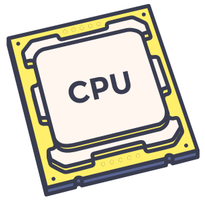
www.dawnwages.info/pydata-vt-2025

ANACONDA.

PyData Vermont 2026

Architecture

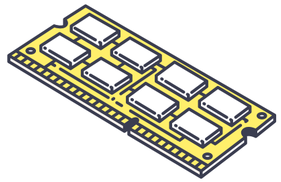
When solving packaging considerations, especially in compute intensive scenarios like scientific comptuing we have to consider what architecture you're using.



Central Processing Unit

Operating System

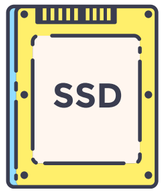


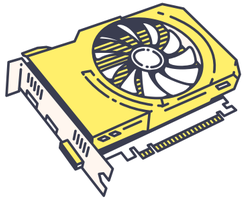


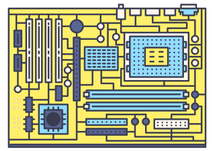
00 — CPU

Responsible for processing instructions as the "brain" of the computer.

Execute commands from the operating system.







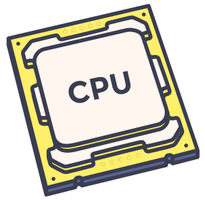
www.dawnwages.info/pydata-vt-2025

ANACONDA.

PyData Vermont 2026

Architecture

When solving packaging considerations, especially in compute intensive scenarios like scientific comptuing we have to consider what architecture you're using.

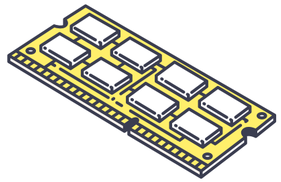


Central Processing Unit

Operating System



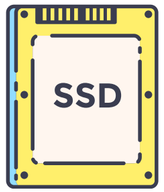
Random Access Memory

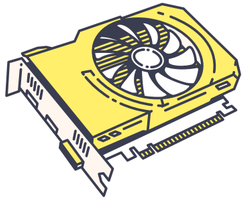


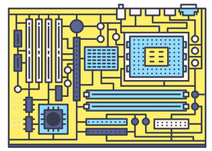
00 — RAM

Temporary memory that stores data while machine is in use.

The more the ram, the faster the in-memory solving.







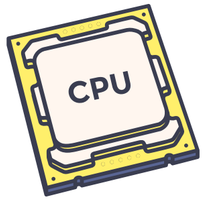
www.dawnwages.info/pydata-vt-2025

ANACONDA.

PyData Vermont 2026

Architecture

When solving packaging considerations, especially in compute intensive scenarios like scientific comptuing we have to consider what architecture you're using.

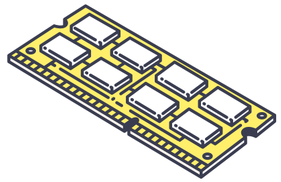


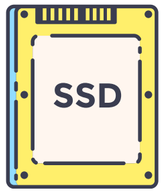
Central Processing Unit

Operating System



Random Access Memory



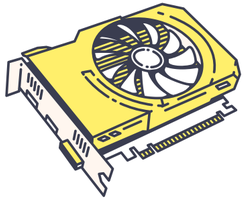


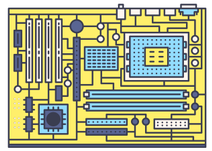
Hard Drive

00 — Hard Drive

Computer's main storage.

Where parallization and general speed benefits happen.





www.dawnwages.info/pydata-vt-2025

ANACONDA.

PyData Vermont 2026

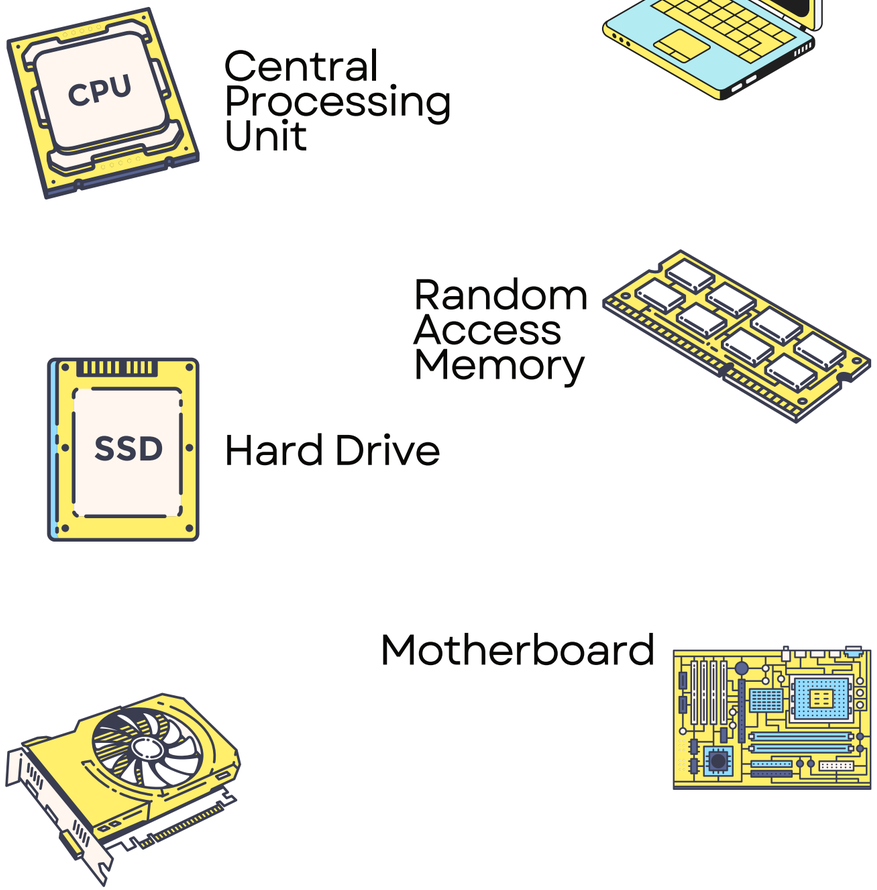
Architecture

When solving packaging considerations, especially in compute intensive scenarios like scientific comptuing we have to consider what architecture you're using.

00 — Motherboard

Main circuit board where all other components are connected

Completely different instruction sets x86_64 vs ARM64 and binaries for one will not compile for the other.



www.dawnwages.info/pydata-vt-2025

ANACONDA.

PyData Vermont 2026

Architecture

When solving packaging considerations, especially in compute intensive scenarios like scientific comptuing we have to consider what architecture you're using.

00 — GPU

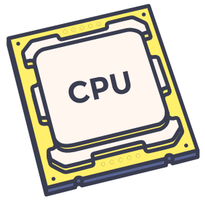
Used to process graphics and images.

These processors are built for specialized compute scenarios now called "GP GPU" General Purpose GPU

www.dawnwages.info/pydata-vt-2025

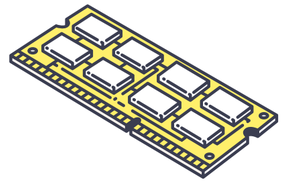
Operating System

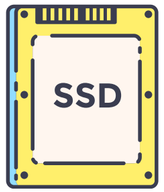




Central Processing Unit

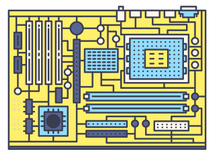
Random Access Memory

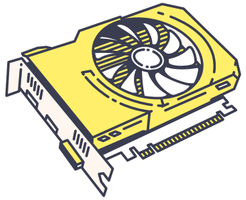




Hard Drive

Motherboard



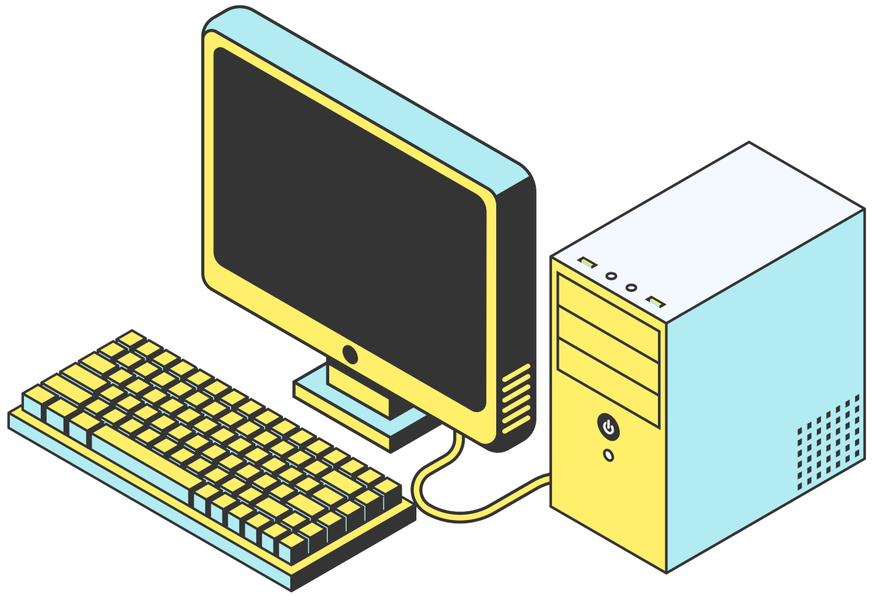


GPU

ANACONDA.

PyData Vermont 2026

Architecture



01 — HDD v SSD

02 — Memory

03 — CPU

04 — GPU & specialized processing

www.dawnwages.info/pydata-vt-2025

ANACONDA.

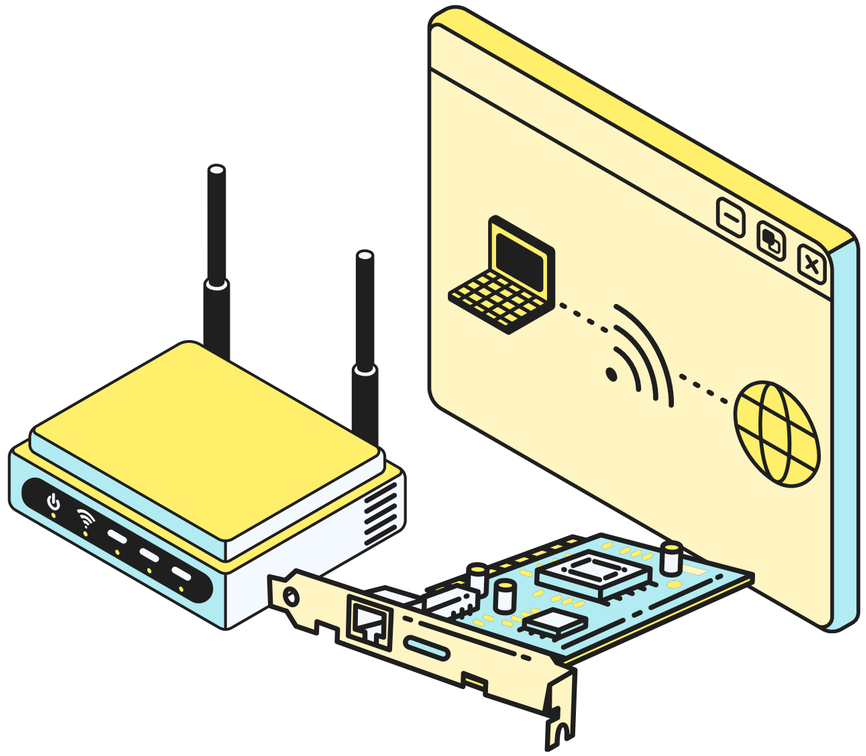
PyData Vermont 2026

Network

04 — Network

05 — HPC

06 — Containers / Cloud



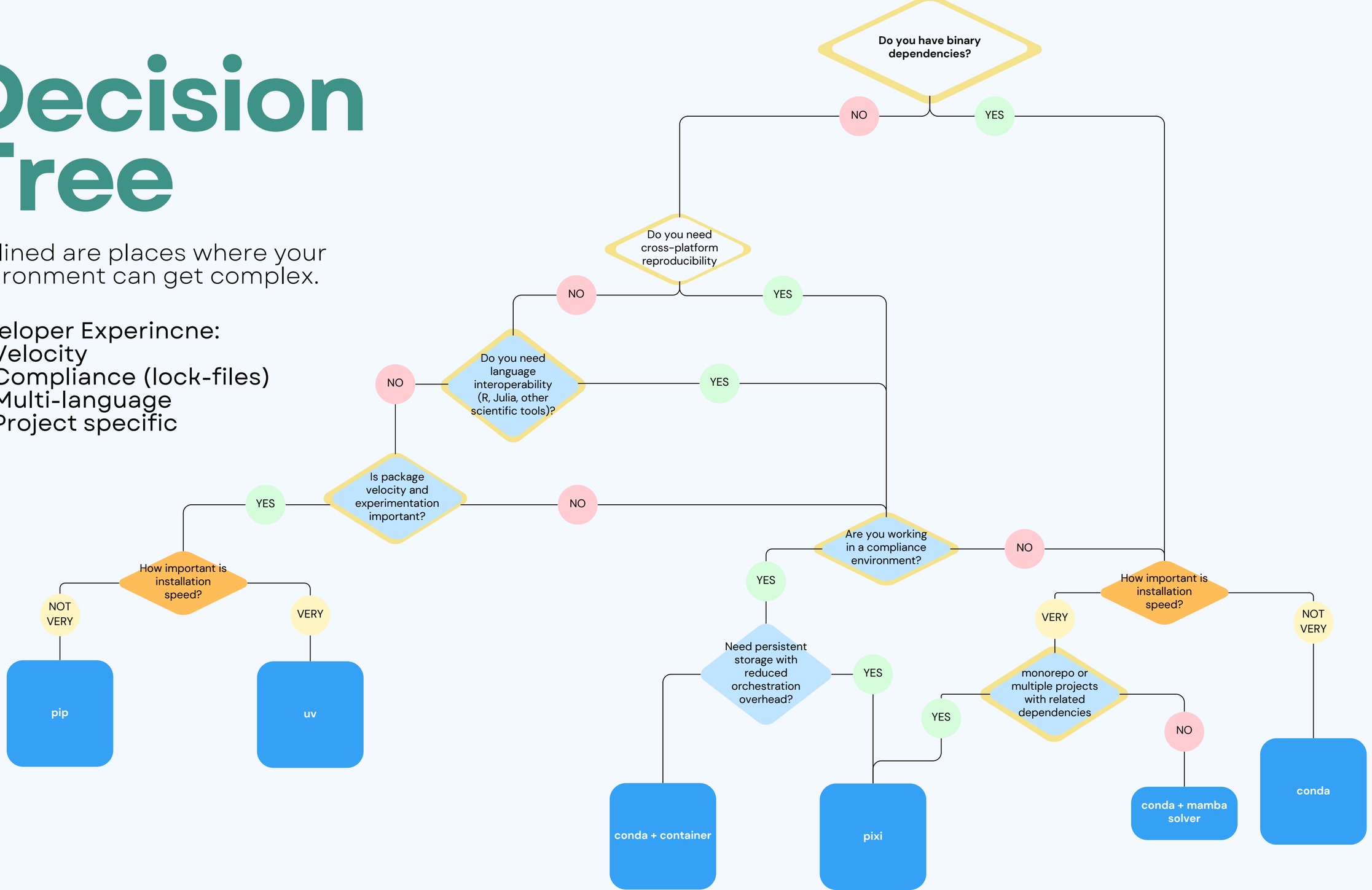
www.dawnwages.info/pydata-vt-2025

Decision Tree

Outlined are places where your environment can get complex.

Developer Experincne:

- Velocity
- Compliance (lock-files)
- Multi-language
- Project specific



ANACONDA.

PyData Vermont 2026

Developer Experience

	Pixi	Conda	Pip	Poetry	uv
Installs Python	✅	✅			✅
Multi Language	✅	✅			
Lockfiles	✅			✅	✅
Task Runner	✅				
Workspace Management	✅			✅	✅



www.dawnwages.info/pydata-vt-2025

ANACONDA.

PyData Vermont 2026

Developer Experience

Python Distribution



	Pixi	Conda	Pip	Poetry	uv
Installs Python	✅	✅			✅
Multi Language	✅	✅			
Lockfiles	✅			✅	✅
Task Runner	✅				
Workspace Management	✅			✅	✅

www.dawnwages.info/pydata-vt-2025

ANACONDA.

PyData Vermont 2026

Developer Experience

Multi Language



	Pixi	Conda	Pip	Poetry	uv
Installs Python	✅	✅			✅
Multi Language	✅	✅			
Lockfiles	✅			✅	✅
Task Runner	✅				
Workspace Management	✅			✅	✅

www.dawnwages.info/pydata-vt-2025

ANACONDA.

PyData Vermont 2026

Developer Experience

Lockfiles



	Pixi	Conda	Pip	Poetry	uv
Installs Python	✅	✅			✅
Multi Language	✅	✅			
Lockfiles	✅			✅	✅
Task Runner	✅				
Workspace Management	✅			✅	✅

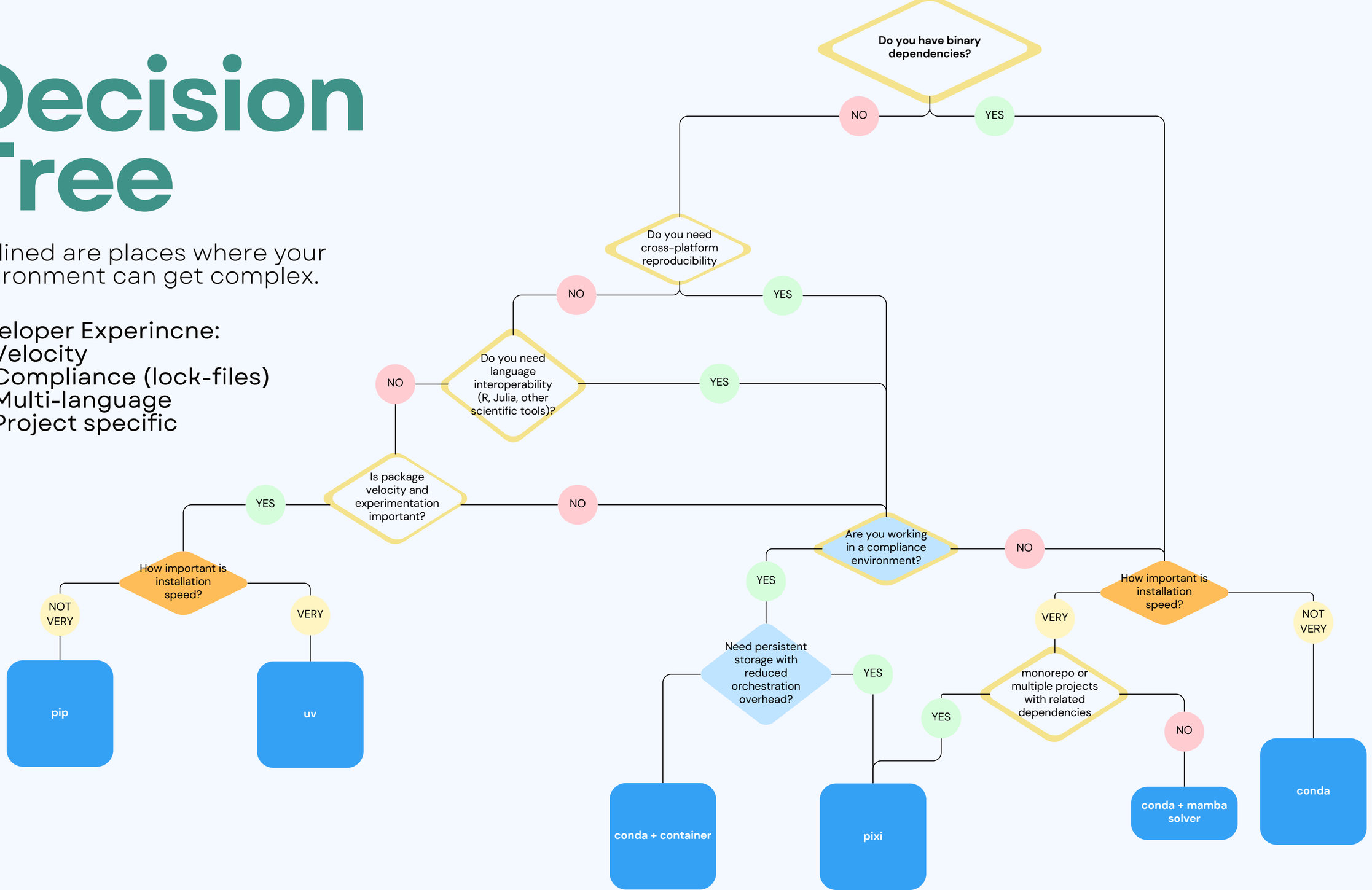
www.dawnwages.info/pydata-vt-2025

Decision Tree

Outlined are places where your environment can get complex.

Developer Experincne:

- Velocity
- Compliance (lock-files)
- Multi-language
- Project specific



ANACONDA.

PyData Vermont 2026

Developer Experience

Task Runner



	Pixi	Conda	Pip	Poetry	uv
Installs Python	✅	✅			✅
Multi Language	✅	✅			
Lockfiles	✅			✅	✅
Task Runner	✅				
Workspace Management	✅			✅	✅

www.dawnwages.info/pydata-vt-2025

ANACONDA.

PyData Vermont 2026

Developer Experience

Workspace management



	Pixi	Conda	Pip	Poetry	uv
Installs Python	✅	✅			✅
Multi Language	✅	✅			
Lockfiles	✅			✅	✅
Task Runner	✅				
Workspace Management	✅			✅	✅

www.dawnwages.info/pydata-vt-2025

ANACONDA.

uv-workon

pixi global

conda

virtualenvwrapper

PyData Vermont 2026

Developer Experience

Workspace management



	Pixi	Conda	Pip	Poetry	uv
Installs Python	✅	✅			✅
Multi Language	✅	✅			
Lockfiles	✅			✅	✅
Task Runner	✅				
Workspace Management	✅			✅	✅

www.dawnwages.info/pydata-vt-2025

ANACONDA.

PyData Vermont 2026

Make it reliable.

Trends in scientific computing tools:

“FIRST GENERATION”

- NumPy
- pandas
- SciPy
- Matplotlib
- Seaborn
- Plotly
- conda
- spyder
- jupyter

www.dawnwages.info/pydata-vt-2025



NEXT GENERATION

ANACONDA.

PyData Vermont 2026

Make it reliable.

Trends in scientific computing tools:

“FIRST GENERATION”

- NumPy
- pandas
- SciPy
- Matplotlib
- Seaborn
- Plotly
- conda
- spyder
- jupyter

www.dawnwages.info/pydata-vt-2025



NEXT GENERATION

- polars
- narwhals
- RAPIDS
- ray
- PySpark & PyArrow
- pixi
- marimo
- PyTorch

ANACONDA.

PyData Vermont 2026

Make it fast.



- Compiled Python implementations
- JIT (Just-In-Time) compilation
- PyPy (alternative Python implementation)
- Rust in the toolchain:
  - Pyo3 (Rust-Python bindings)
  - Packages using Rust
    - Polars - DataFrame library (faster than pandas)
    - Pydantic v2 - Data validation (Rust core)
    - Ruff - Linter (replaces flake8, 10-100x faster)
    - uv - Package installer (replaces pip, much faster)
    - tokenizers - Hugging Face (NLP preprocessing)

ANACONDA.

PyData Vermont 2026

Production-ready

If you're building production ready, most Python toolchains you're "shipping the whole machine"

- Must recreate exact environment 6 months later
- Security patches require controlled updates
- Rollback needs known-good state
- Audit trail for compliance



www.dawnwages.info/pydata-vt-2025

ANACONDA.

PyData Vermont 2026

Advanced Environment Management considerations

1 Task complexity

building with complicated dependencies

- **Multi-language – compiling extensions C/ C++ / Fortran**
- **System library paths**
- **Image processing or geospatial**
- **Parallel, distributed computing, multinode clusters**

2 Hardware considerations

building with performance optimization and publishing dev tools

- **Works on multiple architectures and OS (Windows / Linux / MacOS)**
- **Parallelization**
- **Binary dependencies**
- **GPU / CPU and specialized compute**

3 Experimentation & collaboration

building with quick feedback cycles to share with peers

- **Untested packages**
- **Dependency resolution**
- **Switching between experiments**
- **Locking package versions**
- **Single file scripts**

4 Developer Tools

building your own developer tools, sharing with teams to leverage your work.

- **Analytics Tools**
- **Feedback Mechanisms**

5 Application Development

building resilient software systems

- **Production pipelines**
- **Persistent storage**
- **API**
- **Deploy to multiple environments**

ANACONDA.

PyData Vermont 2026



Dawn Wages

Director of Community & Developer Relations, Anaconda

Questions?

Reach out community@anaconda.com

Take the Python Packaging Survey

19k and counting...