r/programming • 8 mo. ago

henk53

Python dependency management is a dumpster fire

nielscautaerts.xyz

Open

419 241 Share

KemingHe/pyohio-2025

Beyond Benchmarks

A Python Dependency Management Story

Quick Survey

KemingHe/pyohio-2025

My Journey

lincolnloop/python-package-manager-shootout

Benchmarks

lincoln
loop

KemingHe/pyohio-2025

Developers vs. Data Scientists

NOTE: different needs and focus

TODO: bench conda

Adoption Rates

KemingHe/pyohio-2025

methodologies

Adoption Rates

Tool	Repository Count	Market Share	Relative Adoption
Pip	563,200	2.565%	**Dominant** (14x Poetry)
Poetry	40,320	0.184%	**Leading Modern** (2.8x UV)
Conda	18,856	0.086%	**Niche** (Data Science)
UV	14,384	0.066%	**Emerging** (Minimal)

- **Total Analyzed**: 21,955,124 Python repositories
- **Data Source**: GitHub public repositories (2025-07-26)

Sidetrack

Why most are without tooling?

3 Distinct Ecosystems

Educational/Experimental (Majority)

Tutorial projects, Jupyter notebooks, student assignments using standard library to avoid complexity barriers.

Standard Library Maximalists (Minority)

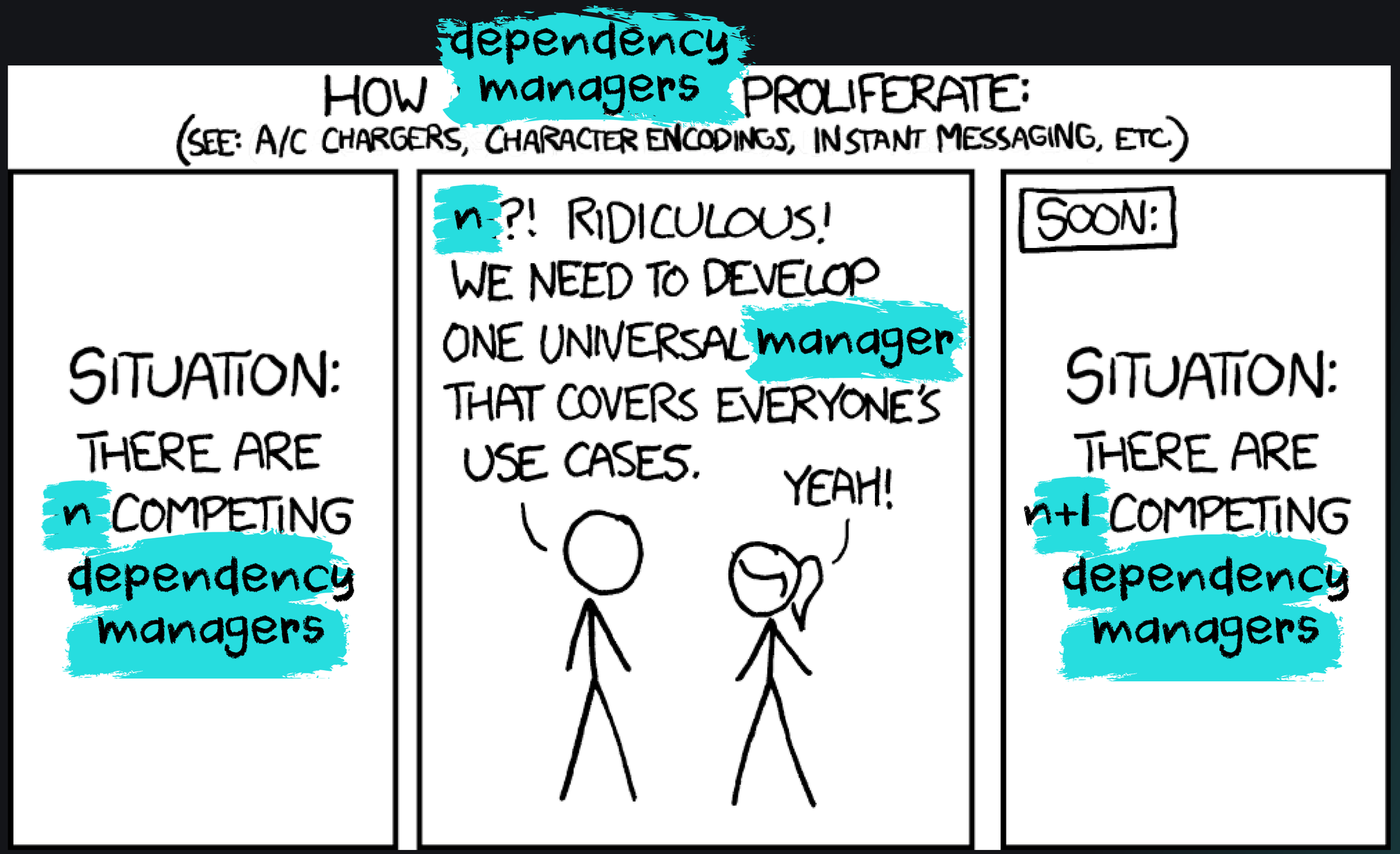
Leveraging Python's 238 built-in modules for self-contained solutions without external dependencies.

Production (Hidden)

Private repositories, containerized deployments, and enterprise tooling invisible in public GitHub statistics.

Let’s Circle Back

Everyone should just use uv, right?


HOW dependency managers PROLIFERATE:
(SEE: A/C CHARGERS, CHARACTER ENCODINGS, INSTANT MESSAGING, ETC.)
SITUATION: THERE ARE n COMPETING dependency managers
n?! RIDICULOUS! WE NEED TO DEVELOP ONE UNIVERSAL manager THAT COVERS EVERYONE'S USE CASES.
YEAH!
SOON:
SITUATION: THERE ARE n+1 COMPETING dependency managers

Switching Cost

Beyond "here's an migration guide to uv"

Switching Cost

1. 🎓 Learners/Students

Switching Cost: 🟢 Low

Cost Factors:

- Local setup: 🟢 Tutorial refresh
- Project migration: 🟢 Fresh starts expected
- Knowledge update: 🟢 Part of learning process
- Team coordination: 🟢 None required

Switching Cost

2. 🔬 Data Scientists

Switching Cost: 🟡 Medium

Cost Factors:

- Environment recreation: 🟡 Complex scientific stack dependencies
- Research continuity: 🔴 Experiment reproducibility concerns
- Package ecosystem: 🔴 Domain-specific availability varies by tool
- Collaboration: 🟡 Team environment synchronization

Switching Cost

3. 💻 Working Developers

Switching Cost: 🟡 Medium-High

Cost Factors:

- Active projects: 🔴 Multiple concurrent migrations required
- Development velocity: 🔴 Temporary productivity loss during learning
- **Business reality**: 🔴 Work planned 6-12 months ahead with feature backlogs
- Sprint capacity: 🔴 Infrastructure migration competes with client-requested features
- Local toolchain: 🟡 IDE/editor integration updates

Switching Cost

4. 👔 Engineering Managers

Switching Cost: 🔴 High

Cost Factors:

- Team coordination: 🔴 Team or department-wide migration across 10+ developers
- Infrastructure: 🔴 CI/CD pipeline reconstruction, production updates
- Documentation: 🔴 Internal docs, onboarding materials, team training
- **ROI challenge**: 🔴 10x faster installs generate zero revenue vs new features
- **Planning reality**: 🔴 Competes with business-critical initiatives planned years ahead
- Emergency disruption: 🔴 Security patches and pivots consistently override tooling

Switching Cost

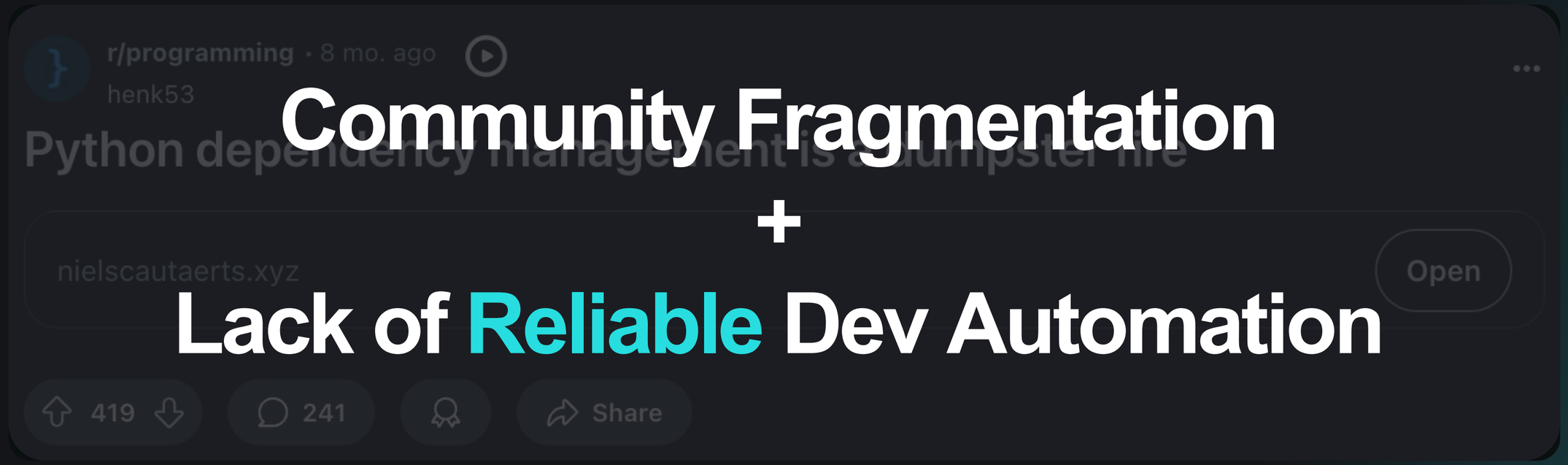
5. 🌍 Open Source Maintainers

Switching Cost: 🔴 High

Cost Factors:

- Community barriers: 🔴 Contributor adoption and learning curves
- Compatibility: 🔴 Backward workflow maintenance required
- Documentation: 🔴 README, contributing guides, issue templates
- Ecosystem risk: 🔴 Fragmentation outweighs personal productivity gains
- Downstream impact: 🔴 Responsibility for dependent projects

Root Problem



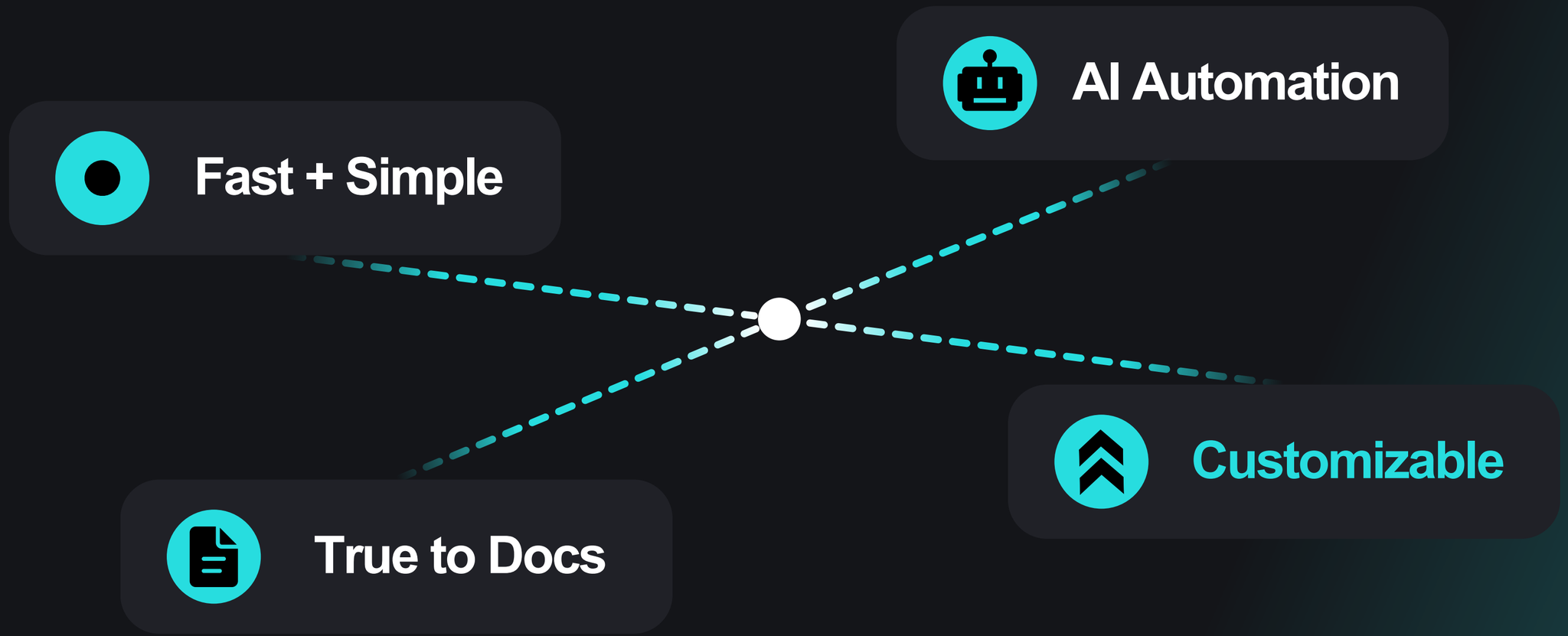
Community Fragmentation

+

Lack of Reliable Dev Automation

KemingHe/pyohio-2025

A Uniting Solution



KemingHe/pyohio-2025

KemingHe/python-dependency-manager-companion-mcp-server

DEMO

docker pull keminghe/py-dep-man-companion:latest

KemingHe/pyohio-2025

KemingHe/python-dependency-manager-companion-mcp-server

DEMO+

a generic pattern for ANY self-updating docs MCP server

KemingHe/pyohio-2025 ⭐

KemingHe/python-dependency-manager-companion-mcp-server ⭐

thank you

+ questions?

KemingHe

in/keminghe