Index Priority @ WheelNext summit

Michael Sarahan, NVIDIA
Andrey Talman, Meta

March 21, 2025

PEP 766: Explicit Priority Choices Among Multiple Indexes

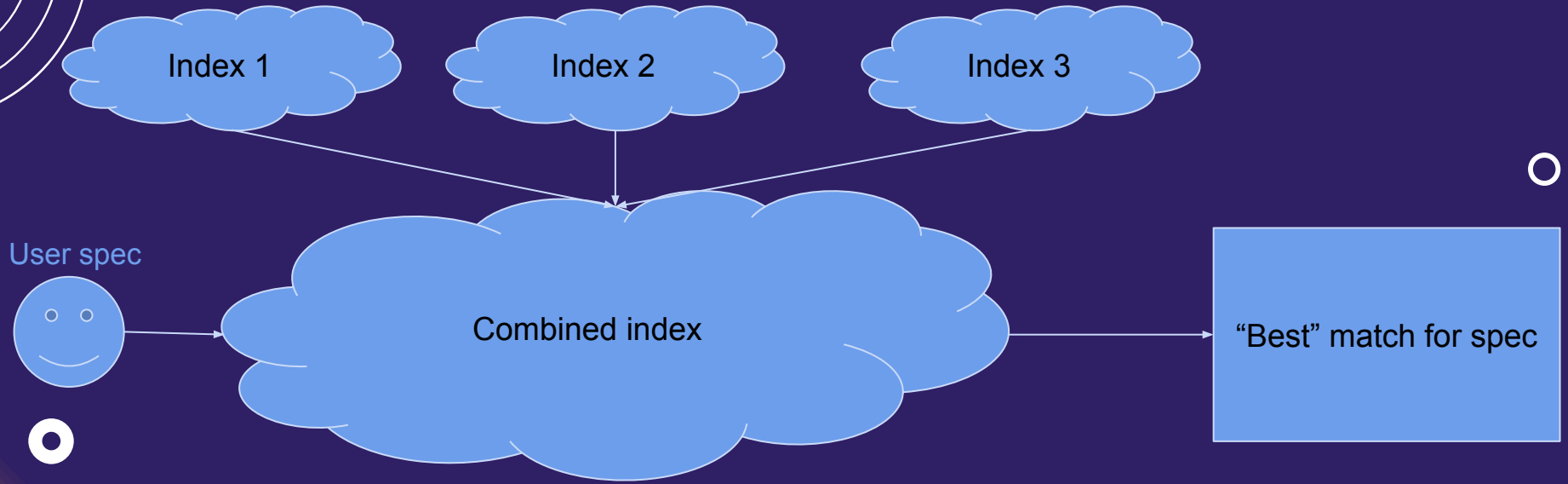
Trust in a source

Latest and greatest package

PEP 766: Explicit Priority Choices Among Multiple Indexes

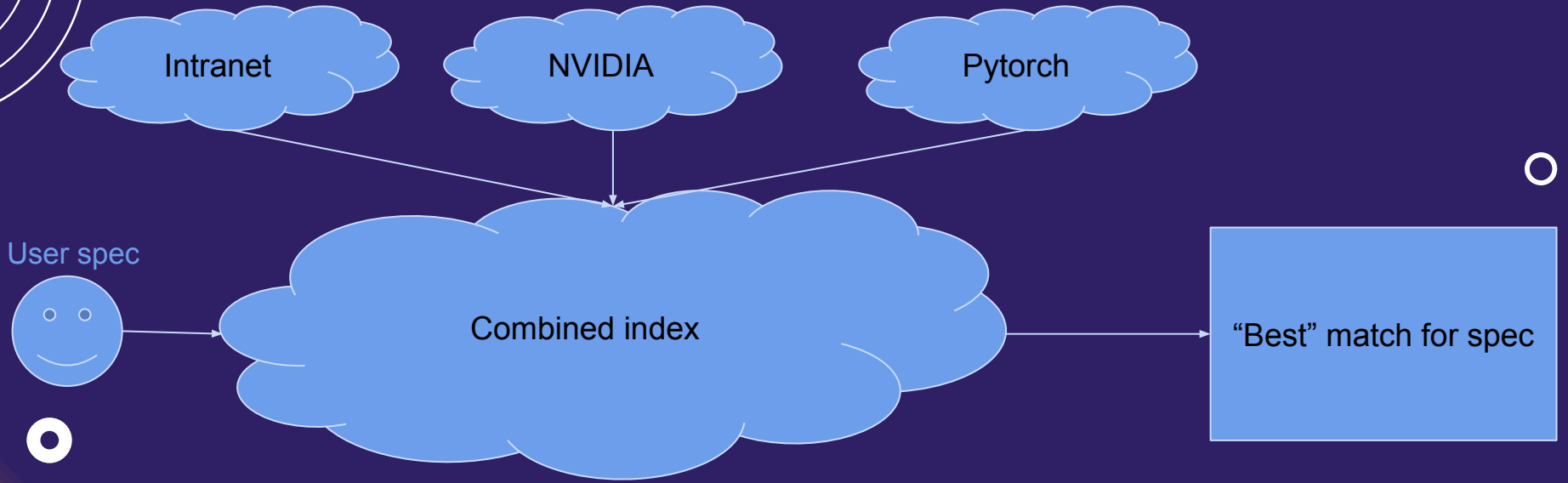
- No shared notion of allowing users to prioritize package source over package version
- Pip's lack of support for source priority is by design
- Leads to unpredictable results at best, and dependency confusion attacks at worst
- PEP merged, in draft status. Pip prototype in progress.
  - Must allow specifying source priority without disrupting current Pip behavior.

Key design assumption in ecosystem



Key design assumption in ecosystem

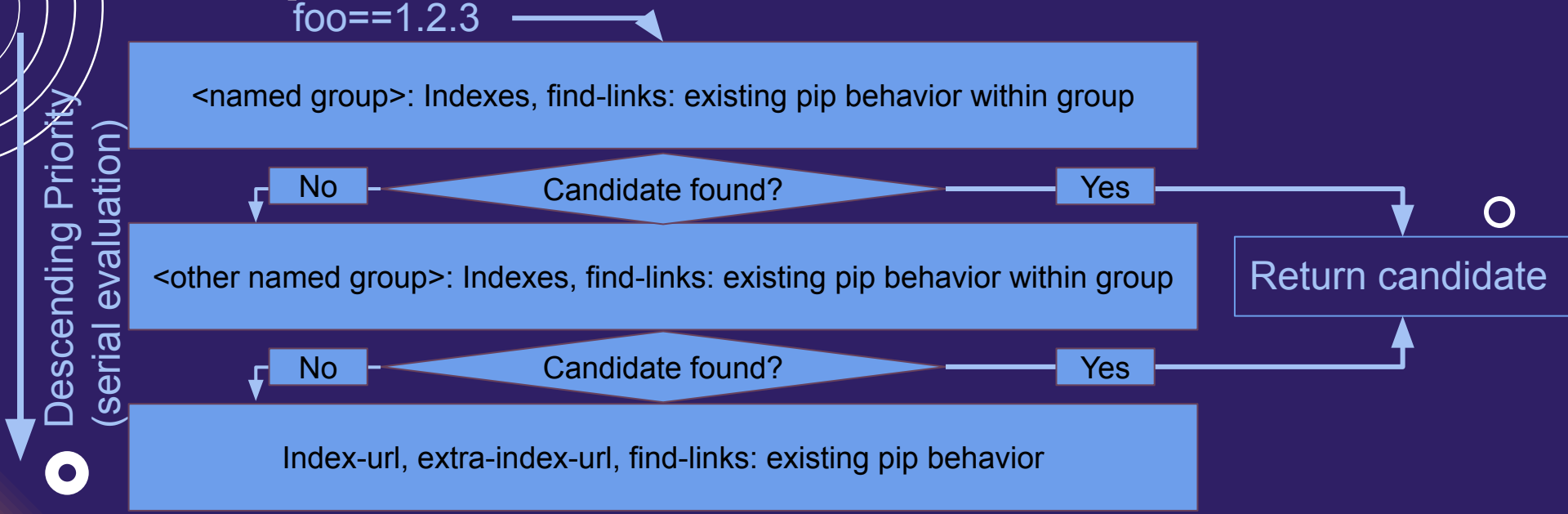
All indexes are equal! No priority



Current Issues with multiple indexes and workarounds

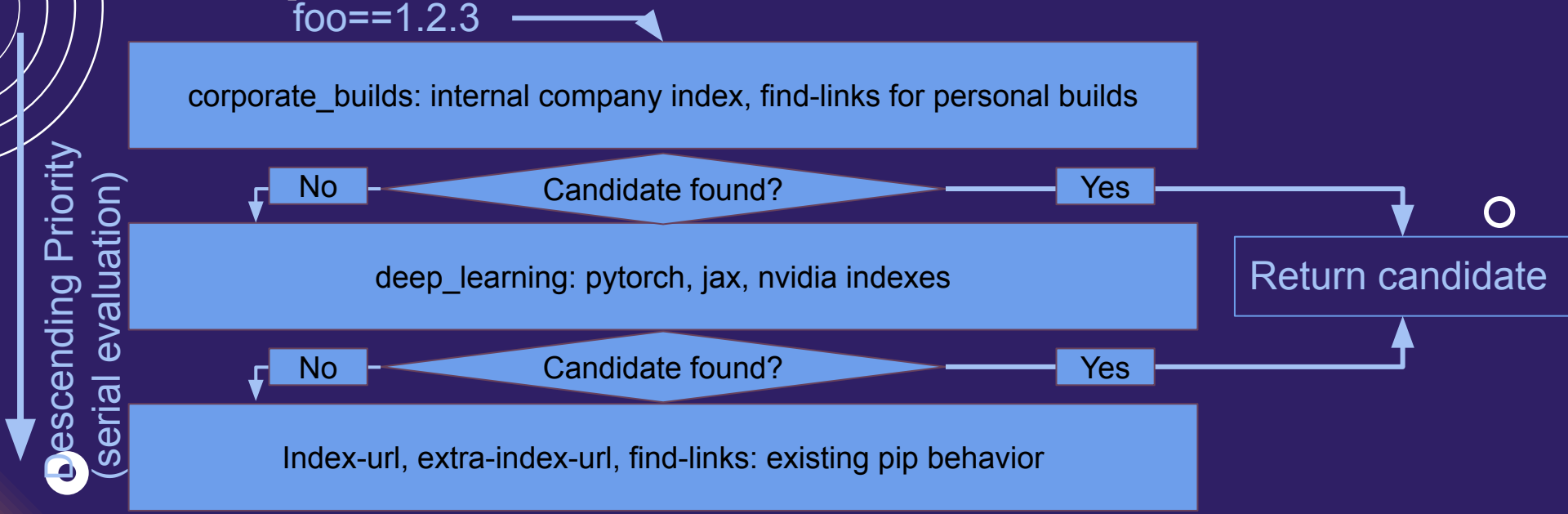
- extra-index-url is not secure: leads to dependency confusion attack
  - Workaround is possible by publishing all required dependencies packages to pypi before publishing to custom index. This is not always possible or easy to achieve.
- index-url solves security issue with dependency confusion attack however:
  - Requires hosting all dependencies of a package on a customized index:
    - This is very maintenance intensive, each custom index need to implement an update mechanism on top of the custom index.
    - Rather than servicing dependencies from their official channels, this produces even more duplication of dependencies on each of the custom indices
  - Not flexible enough solution allowing only 1 index at a time
  - Does not prevent user issuing extra-index-url command instead of index-url

PEP 766: Explicit Priority Choices Among Multiple Indexes



This is an extension of Poetry's repository priority concept, with uv's `first-index` behavior applying "vertically": new behavior without disruption to pip legacy!

PEP 766: Explicit Priority Choices Among Multiple Indexes



Groups are arbitrary. New complexity in configuration, but it also makes more sense to configure sources semi-permanently in config files instead of CLI flags.

Workshop plans

- UX development
  - What should CLI syntax look like?
  - What should config files look like?
  - How hard should we work on trying to get other projects to adopt this?
  - Should we open Pandora's box in terms of saying that a given filename does not need to be similar content everywhere? That location + filename is the identifier?
- PEP refinement
  - PEP was written under assumption that we could not get pip to make a breaking change in adopting index priority. Are things different now that we have a non-breaking way to alter pip?
  - Incorporate interoperability with PEP 708
- Brainstorming potential shortfalls