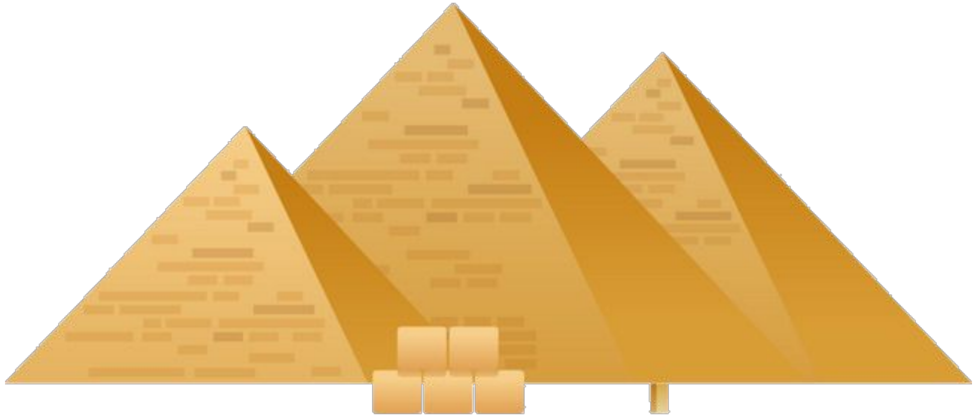


Thoth’s open database for Python developers

Fridolin Pokorny <fridolin@redhat.com>

Agenda

1. $ whoweare
2. Introducing the Python cloud resolver
3. Using Python cloud resolver
4. Declarative interface for the resolver - prescriptions
5. Security - AIDevSecOps
6. References

$ whoarewe

- Thoth - AIDevSecOps
  - Started (2018) as a research project in Red Hat AICoE team, Office of the CTO
  - thoth-station.ninja

- See our linked YouTube channel for more information
- Follow us on Twitter - @ThothStation

Our mission

- Help Python developers and data scientists create healthy applications

- Project has multiple parts:
  - AICoE-CI - a CI that builds container images
  - Thoth resolver - a recommendation engine for Python applications
    - AIDevSecOps
  - Dependency Monkey - a service that can validate software in a cluster
  - jupyterlab-requirements extension for managing dependencies
  - Bots maintaining GitHub repositories
  - A self hosted Python package index using Pulp available to all Red Hatters
  - Container image analysis and containerized Python applications

Introducing the Python cloud resolver

Python resolvers

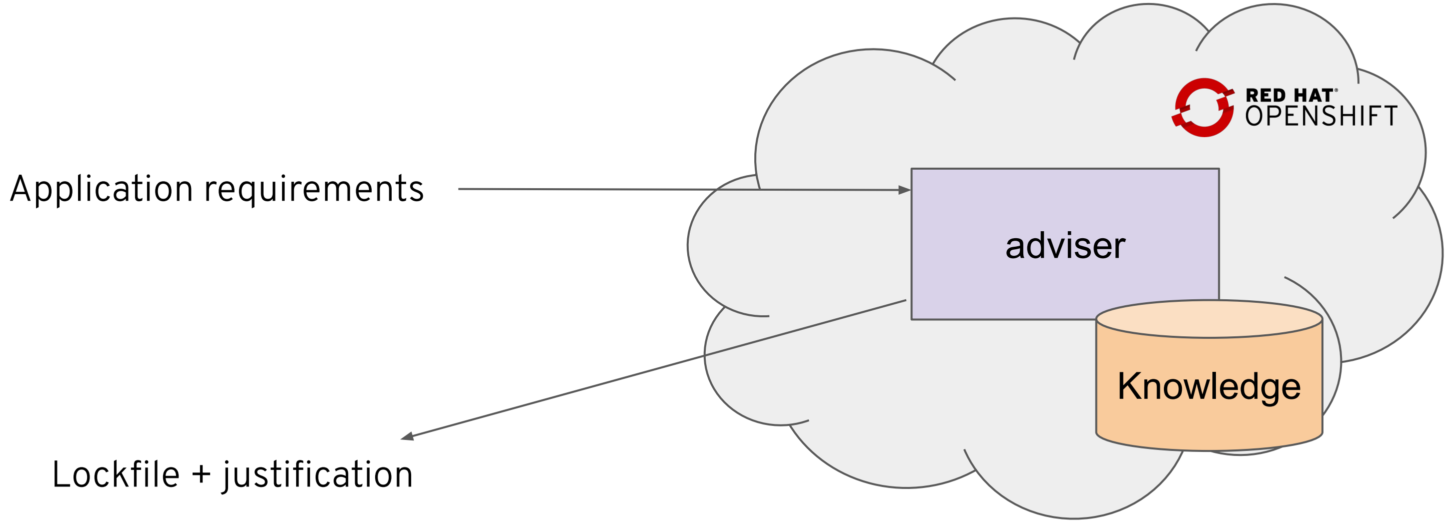
- pip
  - the package installer for Python
- Pipenv
  - Python development workflow for humans
- Poetry
  - Python dependency management and packaging made easy
- Thoth
  - Resurrected ancient deities helping humans with software development

Python resolvers

- pip
  - the package installer for Python
- Pipenv
  - Python development workflow for humans
- Poetry
  - Python dependency management and packaging made easy
- Thoth
  - Resurrected ancient deities helping humans with software development

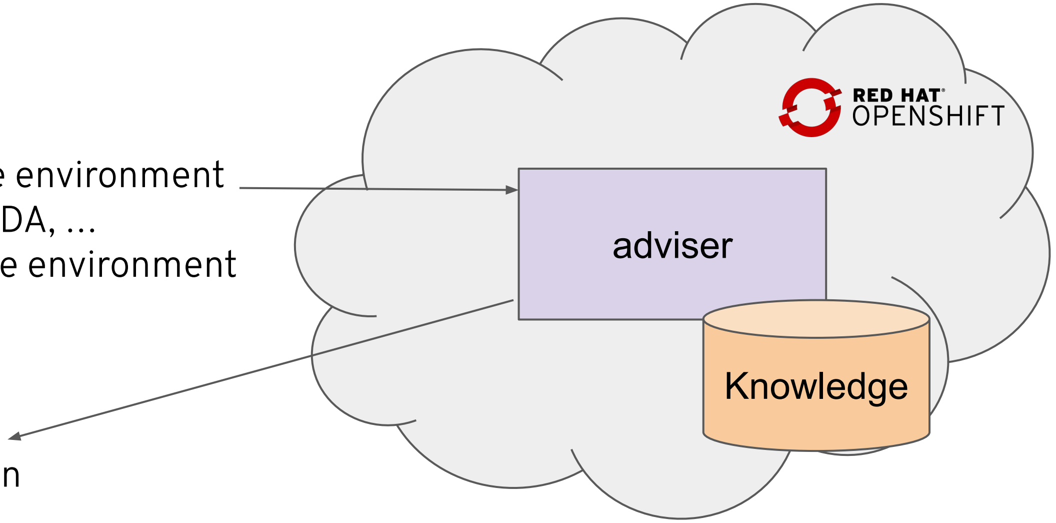
Latest software is not always the greatest choice.

Python cloud resolver



Python cloud resolver

Requirements
Constraints
Information about software in runtime environment
 - OS, Python version, base image, CUDA, ...
Information about hardware in runtime environment
 - CPU, GPU, ...
Static source-code analysis
Recommendation type



Lockfile + justification

Using Python cloud resolver

📄 Managing security in Python applications with the Thoth cloud Python resolver

Thamos Command Line Interface

- One of the Thoth client tools, other tools:
  - jupyterlab-requirements
  - Kebechet bot

- Talks to Thoth's backend and helps with managing your environment

- Available on PyPI:

```
$ pip install thamos
$ thamos --help
$ thamos config
```

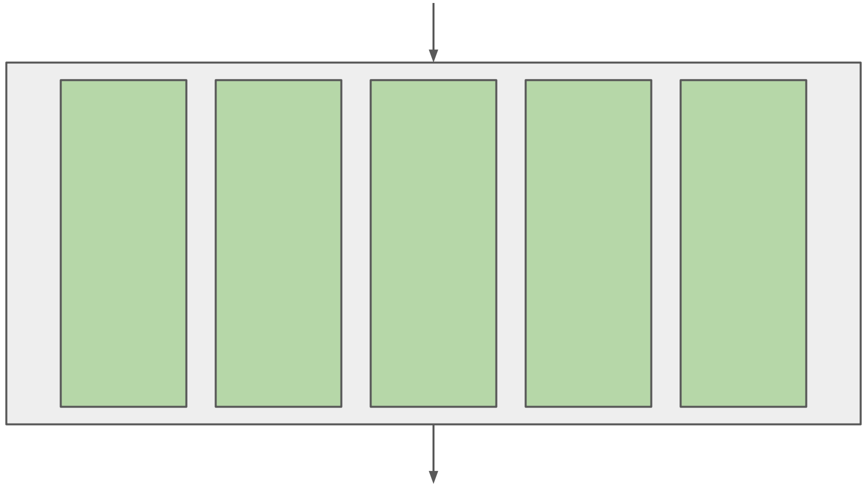
🎥 *Demo: Thamos CLI*

Declarative interface for the resolver to resolve Python packages following prescribed rules

📄 Thoth prescriptions for resolving Python dependencies

Resolution pipeline

Requirements
Constraints
Information about software in runtime environment
 - OS, Python version, base image, CUDA, ...
Information about hardware in runtime environment
 - CPU, GPU, ...
Static source-code analysis
Recommendation type



Lockfile + justification

Prescriptions - declarative interface to the cloud based resolver

- Provide a way to declarativelly state how the resolution process should look like

- Community driven open database used by the resolver to resolve high quality software
  - github.com/thoth-station/prescriptions

- A set of YAML files that are automatically consumed by the resolver in a deployment

- See documentation for more information:
  - thoth-station.ninja/docs/developers/adviser/prescription.html

Prescriptions - Example

- Pillow in version 8.3.0 does not work with NumPy

  github.com/python-pillow/Pillow/issues/5571

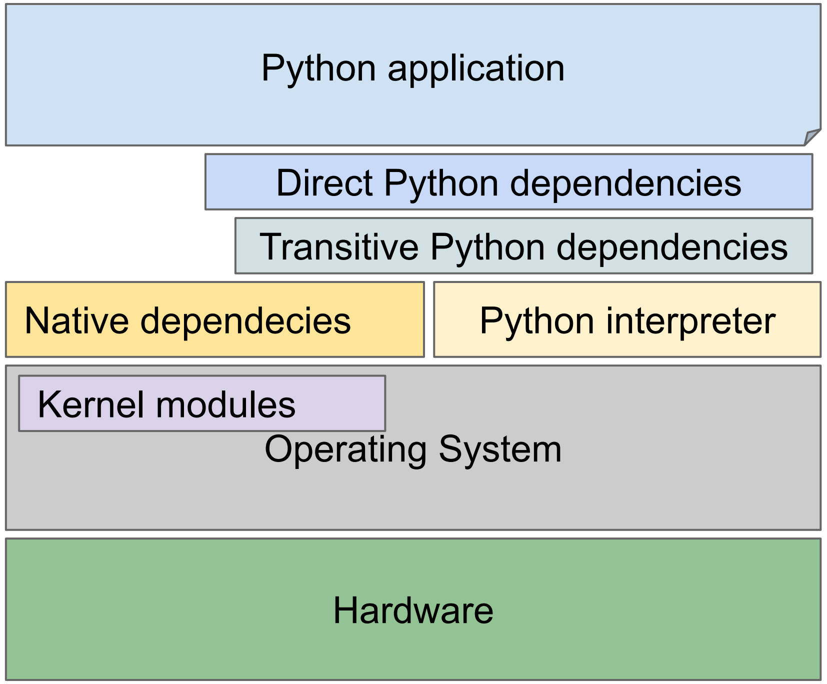
```
with PIL.Image.open(filepath) as img:
    numpy.array(img, dtype=numpy.float32)
```

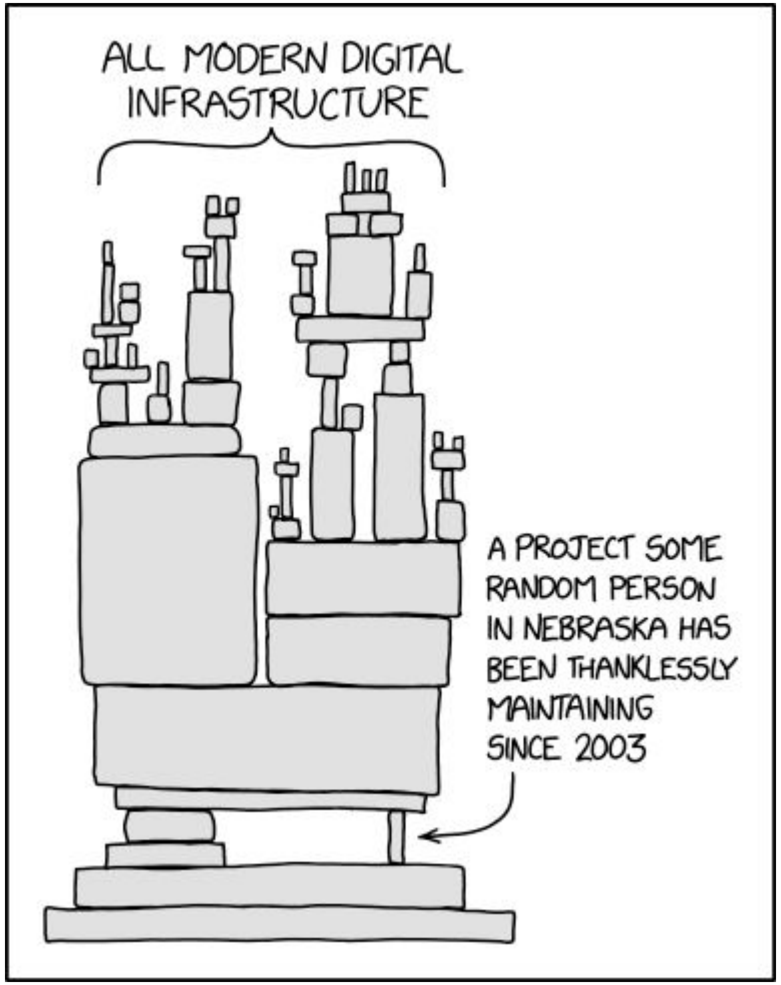
```
>         frame_paletted = np.array(im, np.uint8)
E         TypeError: __array__() takes 1 positional argument but 2 were given

  /lib/python3.9/site-packages/imageio/plugins/pillow.py:745: TypeError
```

Prescriptions - Example

- Pillow in version 8.3.0 does not work with NumPy
  - Fix using prescriptions 👷

Fitting the runtime environment




ALL MODERN DIGITAL INFRASTRUCTURE
A PROJECT SOME RANDOM PERSON IN NEBRASKA HAS BEEN THANKLESSLY MAINTAINING SINCE 2003

Source: https://xkcd.com/2347/

Security - AIDevSecOps

📄 Secure your Python applications with Thoth recommendations

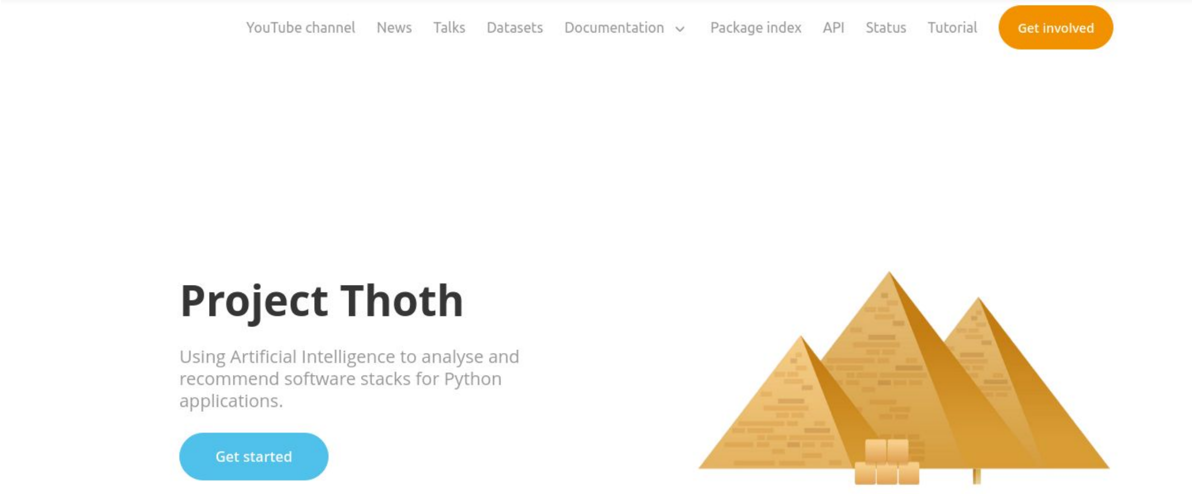
Security - AIDevSecOps

- Docs: Thoth security advises
- Recommendations based on static source code analysis
  - See recommendations from the Python standard library (example)
- PyPA - advisory-db
  - A database of known vulnerabilities in Python ecosystem
  - github.com/pypa/advisory-db
- Security Scorecards by Open Source Security Foundation
  - openssf.org/blog/2020/11/06/security-scorecards-for-open-source-projects
  - Example: see scorecards_ prefixed prescriptions for TensorFlow
- Container image analyses - vulnerabilities in the base container images used

… *additional information about Python packages not strictly related to security*

References

🖥️ References



Project Thoth

Using Artificial Intelligence to analyse and recommend software stacks for Python applications.

Red Hat | IBM

thoth-station.ninja

📄 References

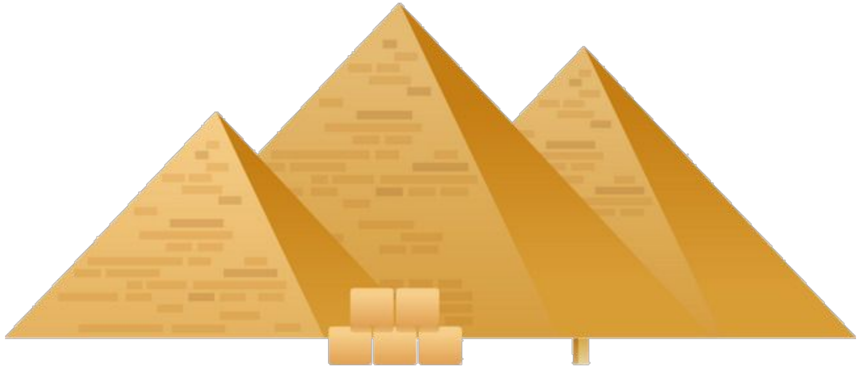
- Introspecting containerized Python applications in a cluster with Thoth Amun
- How to self-host a Python package index using Pulp
- Extracting dependencies from Python packages
- Extracting information from Python source code
- Prevent Python dependency confusion attacks with Thoth
- Build and extend containerized applications with Project Thoth
- Customize Python dependency resolution with machine learning
- Generating pseudorandom numbers in Python
- Secure your Python applications with Thoth recommendations
- Find and compare Python libraries with project2vec

📄 References

- Thoth prescriptions for resolving Python dependencies
- Resolve Python dependencies with Thoth Dependency Monkey
- micropipenv: Installing Python dependencies in containerized applications
- Continuous learning in Project Thoth using Kafka and Argo
- Can we consider --editable a bad practice?
- Managing Python dependencies with the Thoth JupyterLab extension
- Use Kebechet machine learning to perform source code operations
- AI software stack inspection with Thoth and TensorFlow
- Microbenchmarks for AI applications using Red Hat OpenShift on PSI in project Thoth

🐦 References

- Thoth’s website
  - ○ thoth-station.ninja
- Source code:
  - ○ github.com/thoth-station
- @ThothStation Twitter handle
- Thoth Station YouTube channel
- Talks and presentations



Thank you

Red Hat is the world's leading provider of enterprise open source software solutions. Award-winning support, training, and consulting services make Red Hat a trusted adviser to the Fortune 500.

in linkedin.com/company/red-hat

▶ youtube.com/user/RedHatVideos

f facebook.com/redhatinc

twitter.com/RedHat