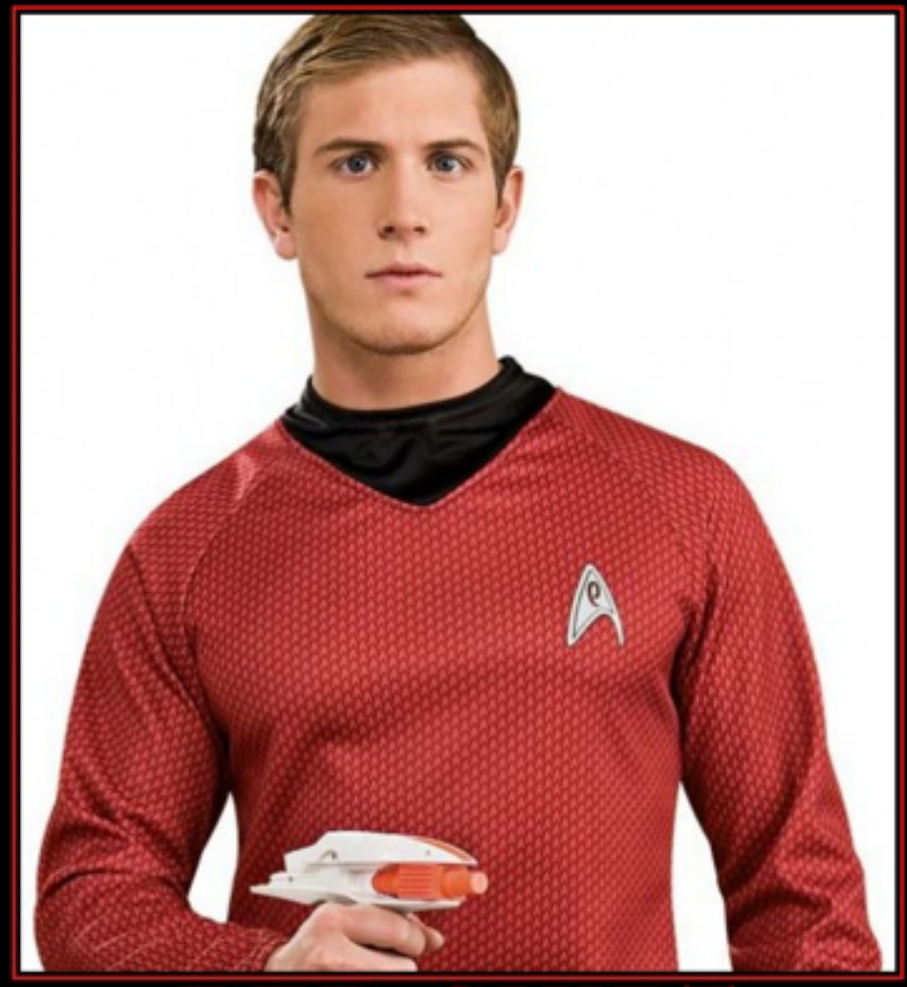
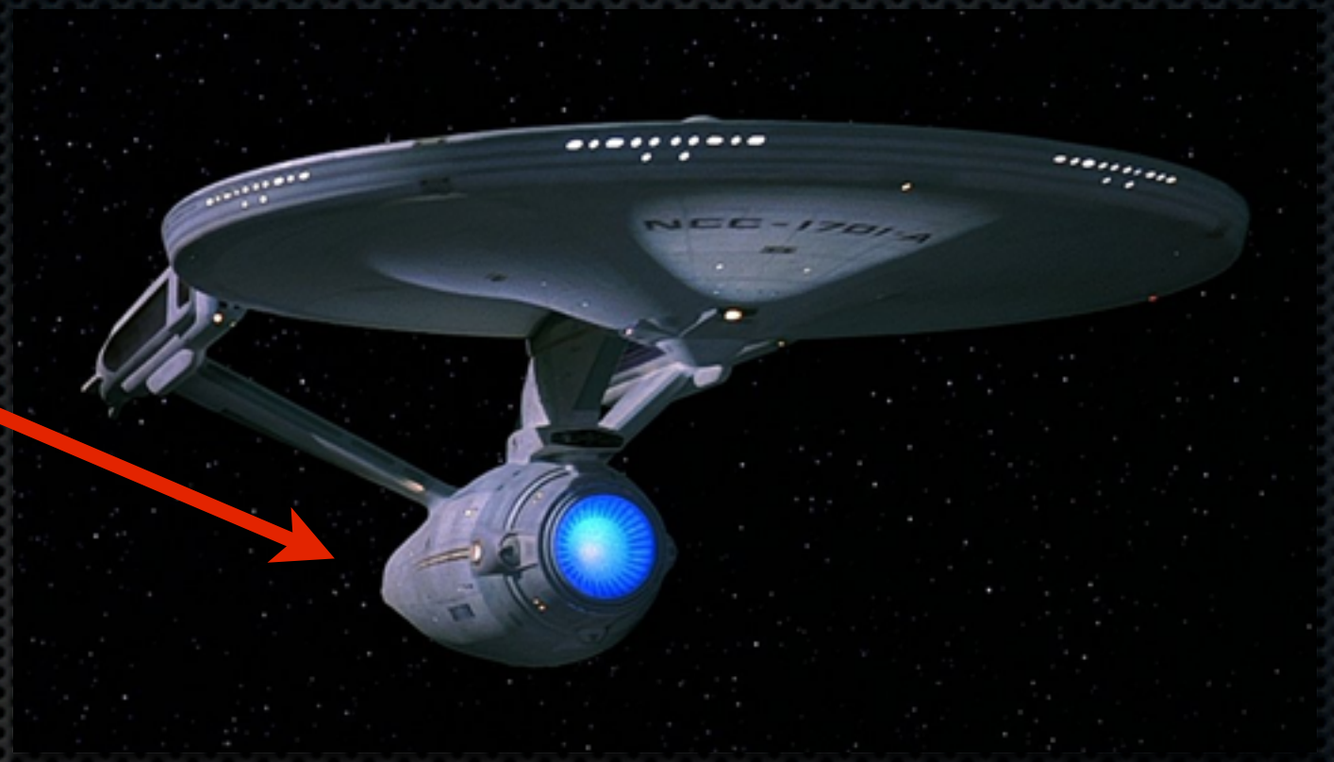
Python Packaging on the Enterprise

Hi, I’m Ed



Enterprise?



- Usually means ‘expensive and bloated’
- In this case, I mean ‘big teams’ and ‘lots of packages’
- Restricted environments

Why `pkglib`?

this look right here

Project Goals

- Create a consistent toolset for managing Python development environments
- Easy for new Python developers - minimal, intuitive command-set
- Support for development workflows: source checkouts, development packages, release packages
- Understand in-house vs 3rd party
- ‘Batteries included’ - testing tools, templates, docs

`pkglib` overview

- Library has 3 main components:
  - **`pkglib`:** tools to manage virtualenvs, source checkouts, run tests, build and upload code to PyPI
  - **`pkglib.testing`:** a suite of testing utilities for working with services, databases, webdrivers, `py.test`
  - **`pkglib.project_template`**: PasteScript template for generating pkglib-enabled projects

Shortcuts and Opinions

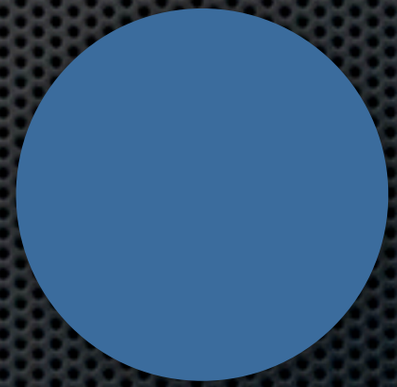
- Closed environment means we could take shortcuts:
  - Linux
  - Python 2.6 - 2.7
  - Subversion
- We also made some opinionated decisions:
  - Python eggs, no source packages
  - Py.test
- Don’t worry! OSS version is working to fill in the gaps

Packaging History

- In the beginning, there was `distutils`
  - PKG-INFO
  - `setup.py`
  - PyPI Server

Packaging History

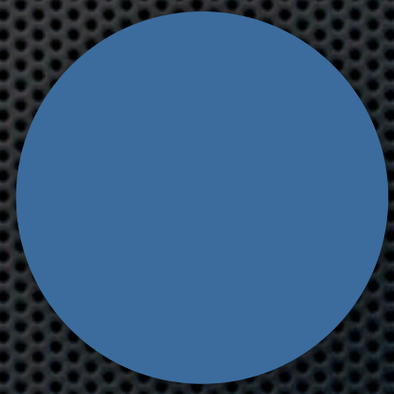
- In the beginning, there was `distutils`
  - PKG-INFO
  - `setup.py`
  - PyPI Server



distutils

Packaging History

- Along came `setuptools`
  - `easy_install`
  - egg file format
  - dependency management
  - .. and many more



distutils

Packaging History

- Along came `setuptools`
  - `easy_install`
  - egg file format
  - dependency management
  - .. and many more

setuptools distutils

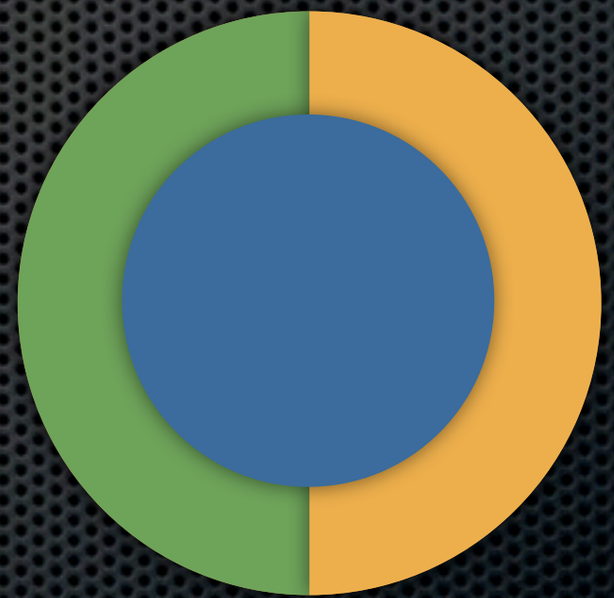
Packaging History

- `setuptools` was forked and so `distribute` was born
- more active development
- less bugs

setuptools distutils

Packaging History

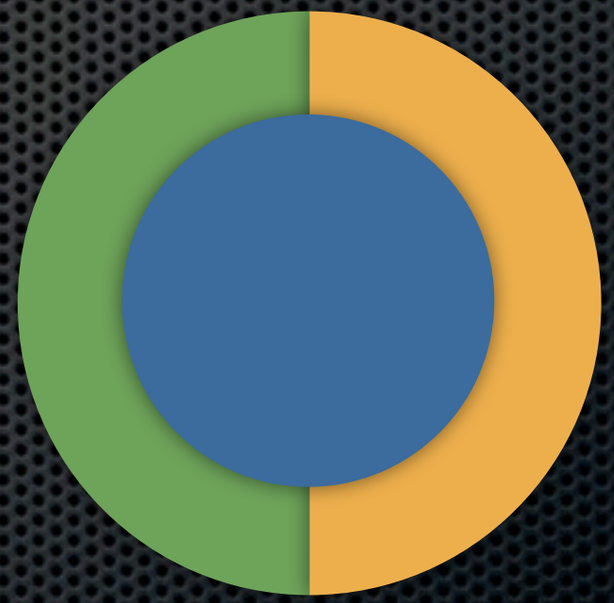
- `setuptools` was forked and so `distribute` was born
- more active development
- less bugs



setuptools
distribute
distutils

Packaging History

- Hello, `pip`!
  - new hotness
  - nicer API
  - `pip uninstall`
  - `requirements.txt`



setuptools
distribute

distutils

Packaging History

- Hello, `pip`!
  - new hotness
  - nicer API
  - `pip uninstall`
  - `requirements.txt`

pip

setuptools

distribute

distutils

Packaging History

- But also - `zc.buildout`
  - clever dependency resolver
  - build recipes

pip

setuptools

distribute

distutils

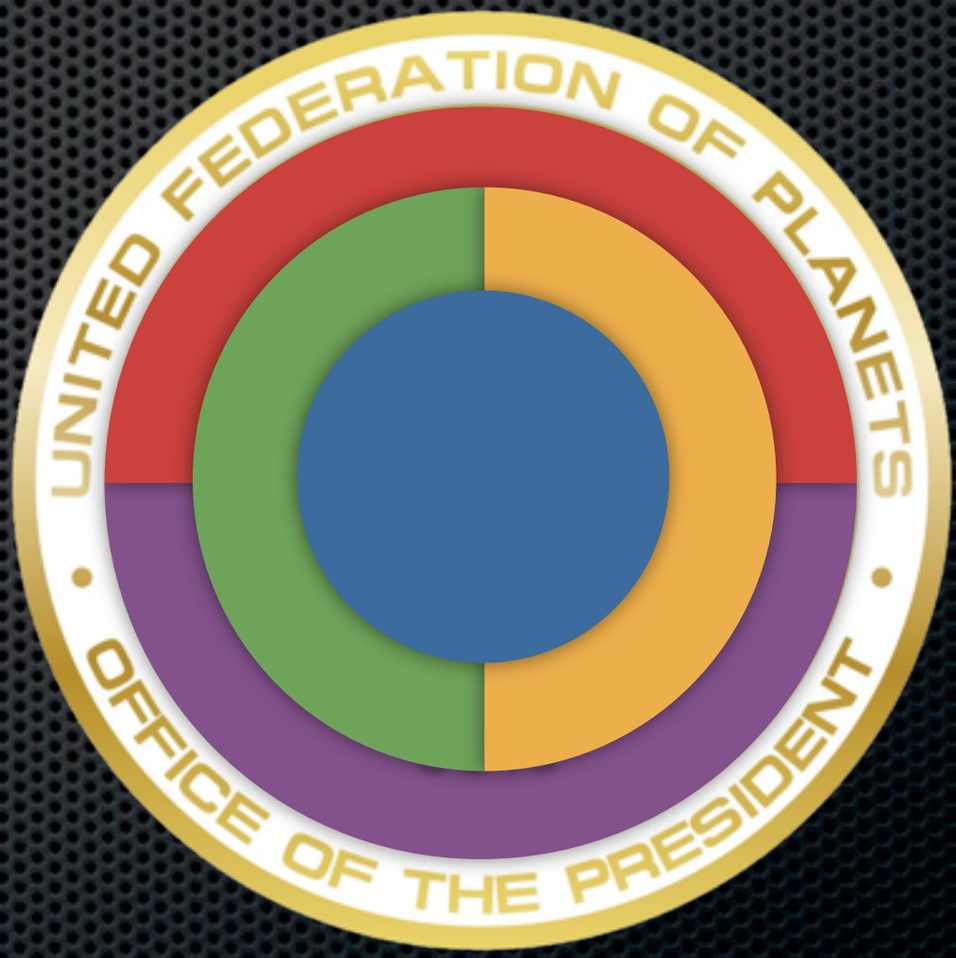
Packaging History

- But also - `zc.buildout`
  - clever dependency resolver
  - build recipes

pip
zc.buildout
setuptools
distribute
distutils

`pkglib` Architecture

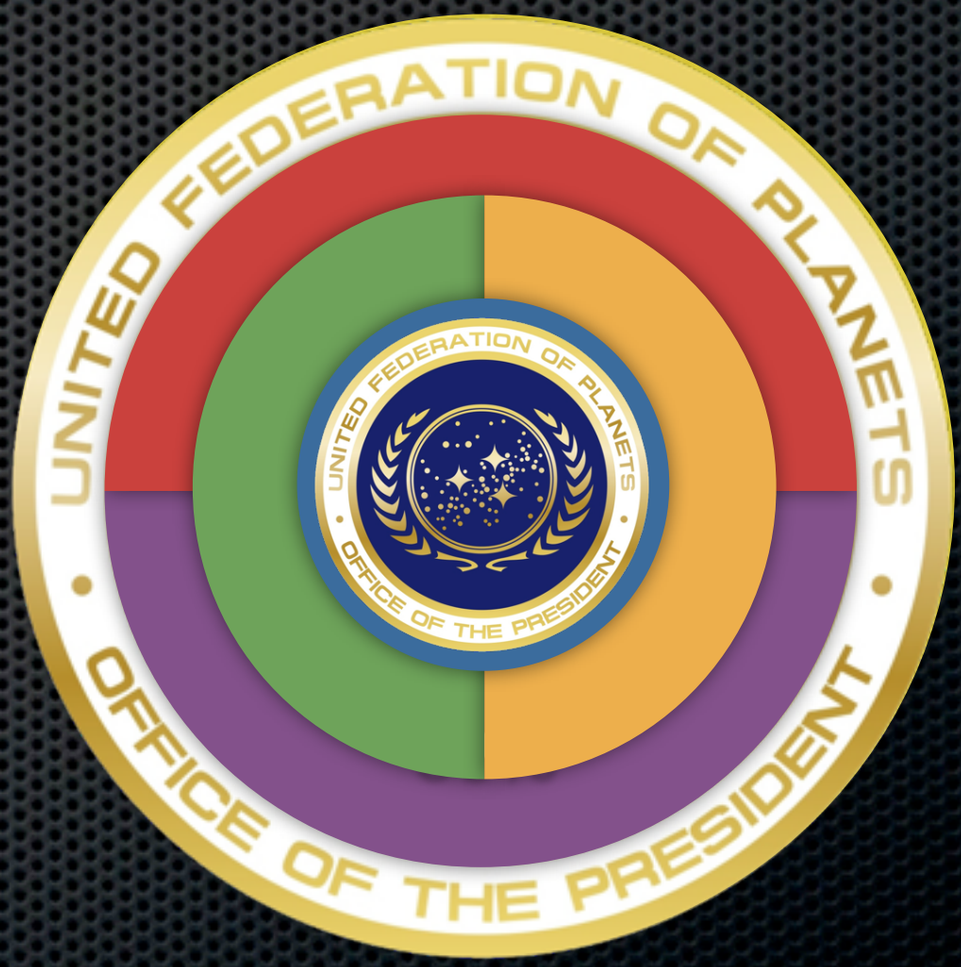
- `pkglib`
  - picked aspects of each we wanted to use
  - packaging code primarily uses `distribute's` API
  - why not `pip`? no binaries :(



pip
zc.buildout
setuptools
distribute
distutils

`pkglib` Architecture

- `pkglib`
  - picked aspects of each we wanted to use
  - packaging code primarily uses `distribute's` API
  - why not `pip`? no binaries :(

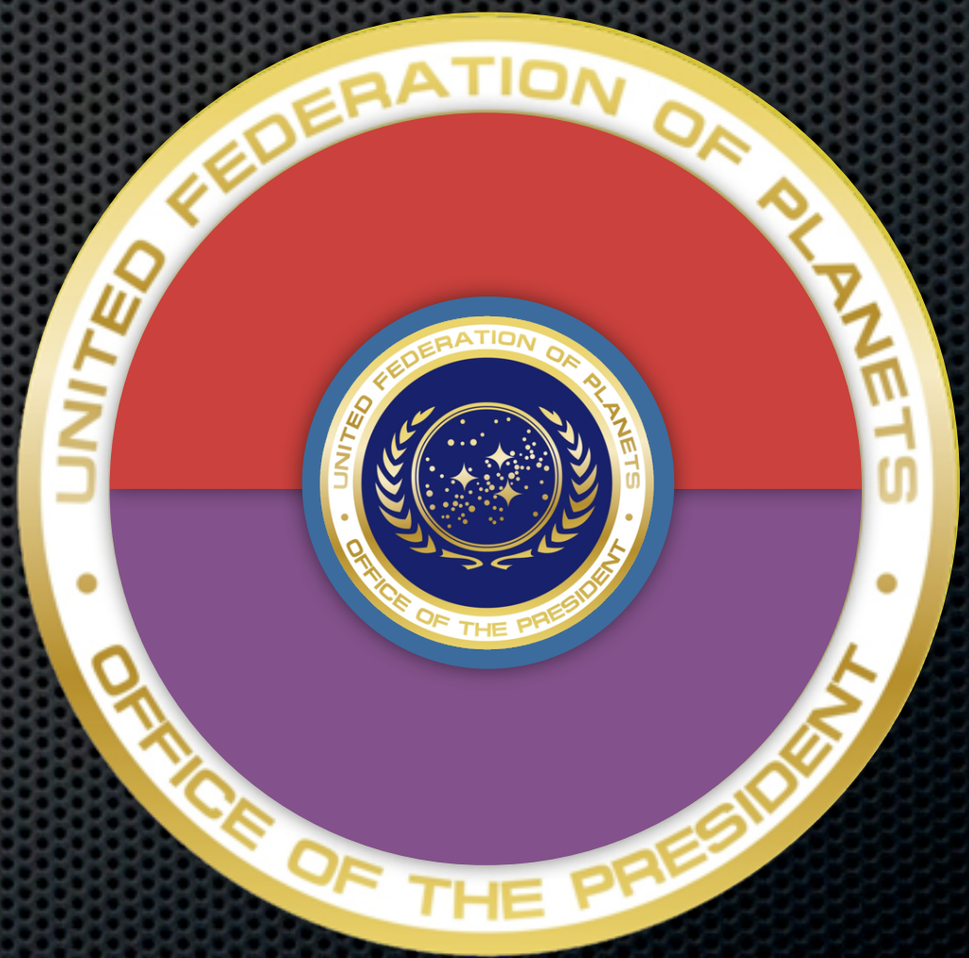


pkglib

pip
zc.buildout
setuptools
distribute
distutils

`pkglib` Architecture

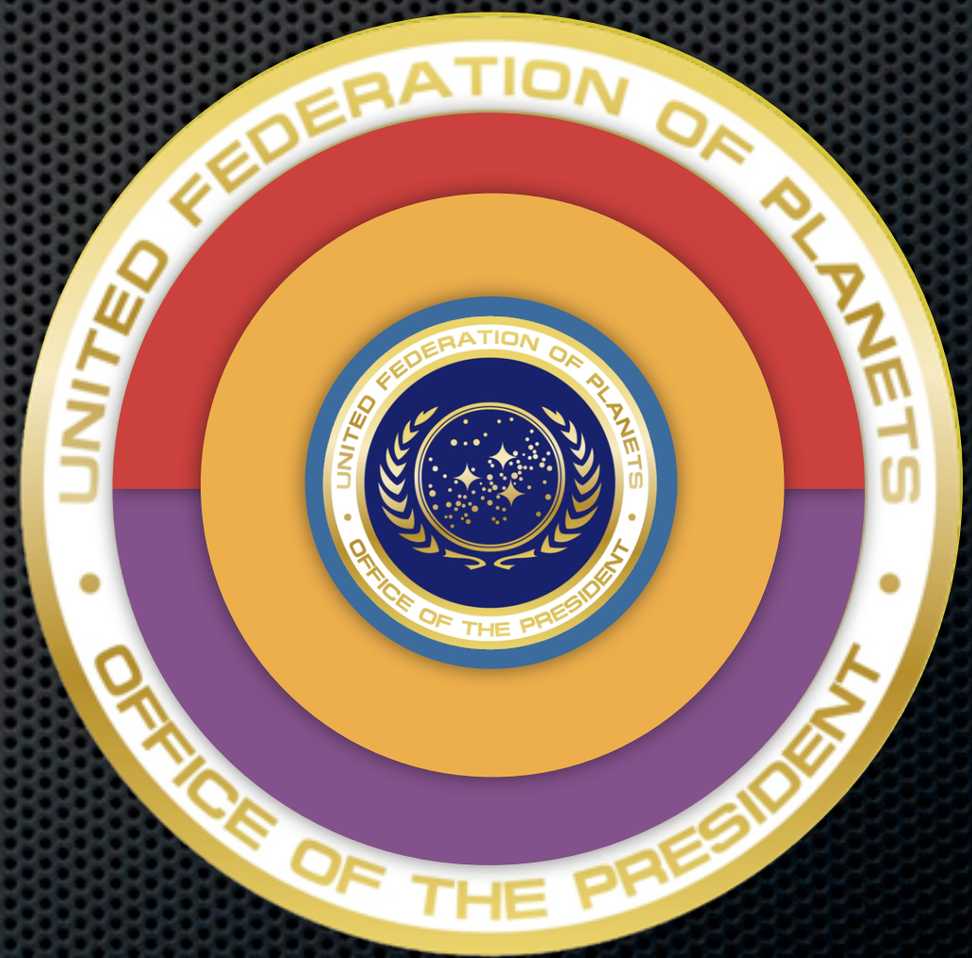
- `setuptools` and `distribute` have now merged



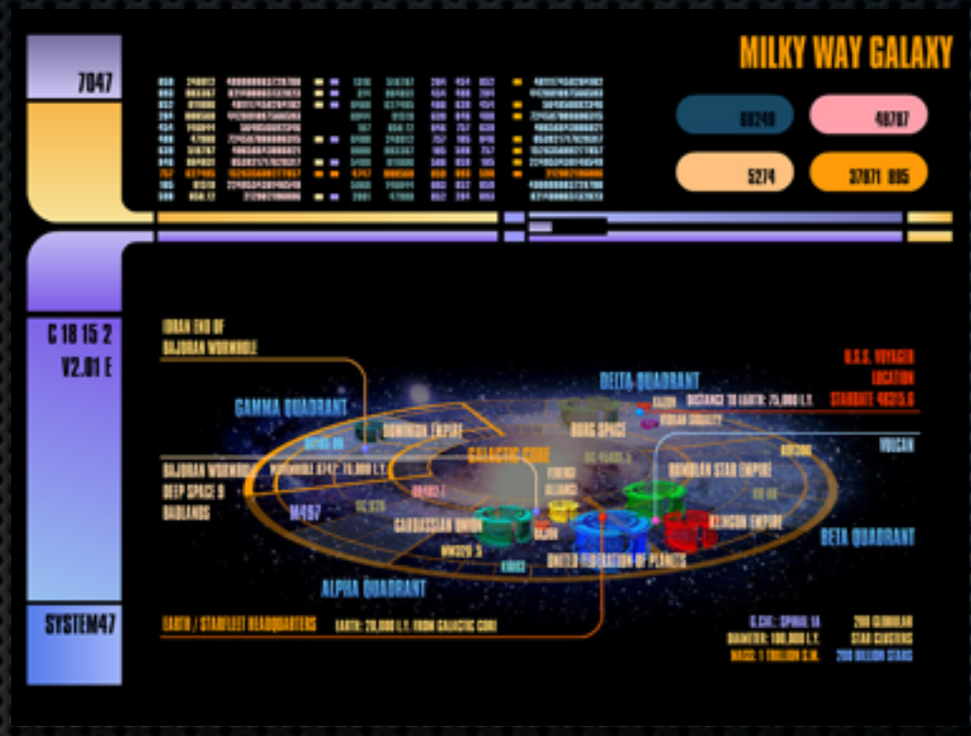
- pkglib
- pip
- zc.buildout
- setuptools
- distutils

`pkglib` Architecture

- `setuptools` and `distribute` have now merged



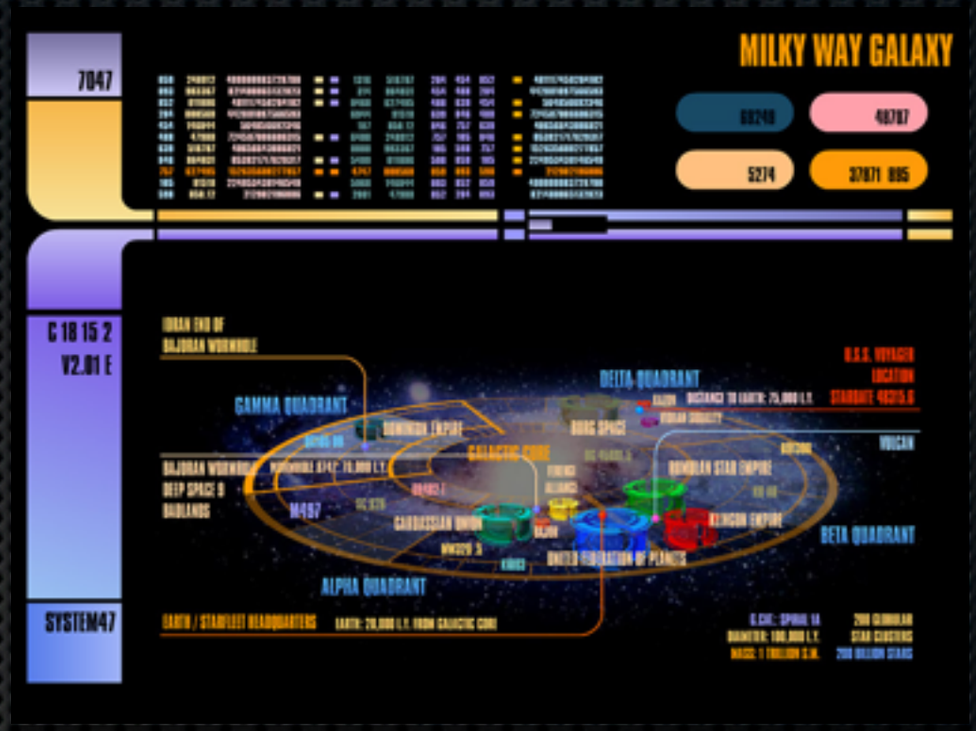
Configuration



- Pkglib need to know a few things about your company
  - PyPI server address
  - In-house package namespaces
  - Jenkins/Hudson server address
  - Test directory layout
  - Library search paths
  - .. and much more

Configuration

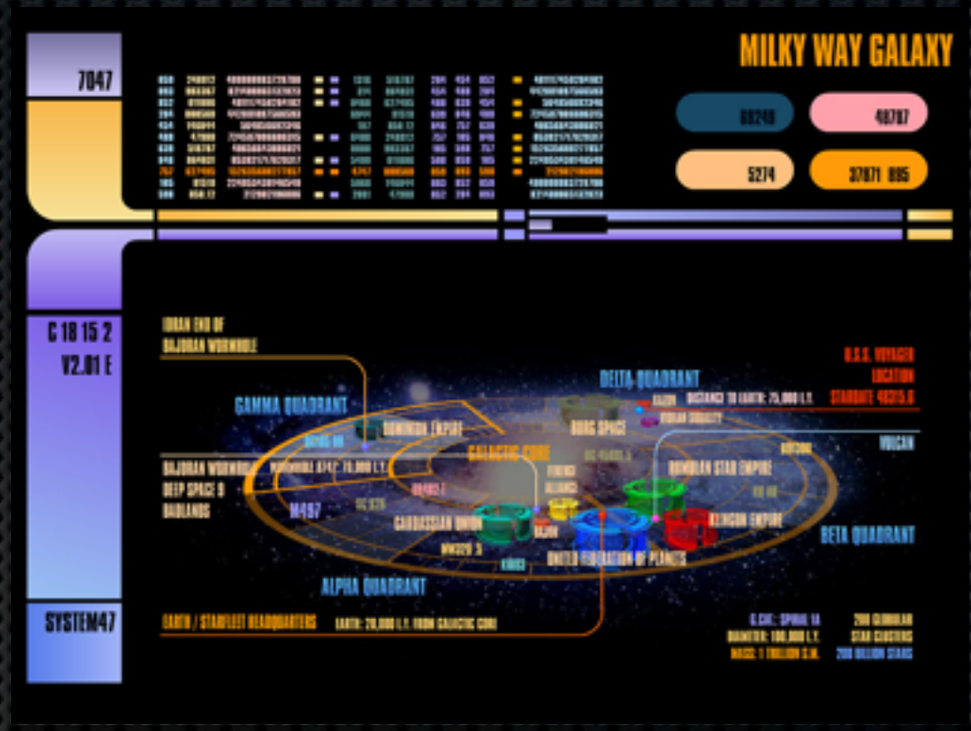
- Global config file, set by `${PKGLIB_CONFIG}`




```
# Acme Co. pkglib configuration file
[pkglib]
pypi_url = http://pypi.acme.example
jenkins_url = http://jenkins.acme.example
vcs = svn
namespaces =
    acme
virtualenv_executable = ${VIRTUALENV_DIR}/bin/virtualenv
mongo_bin = /usr/sbin
redis_executable = /usr/sbin/redis-server
...
```

Configuration

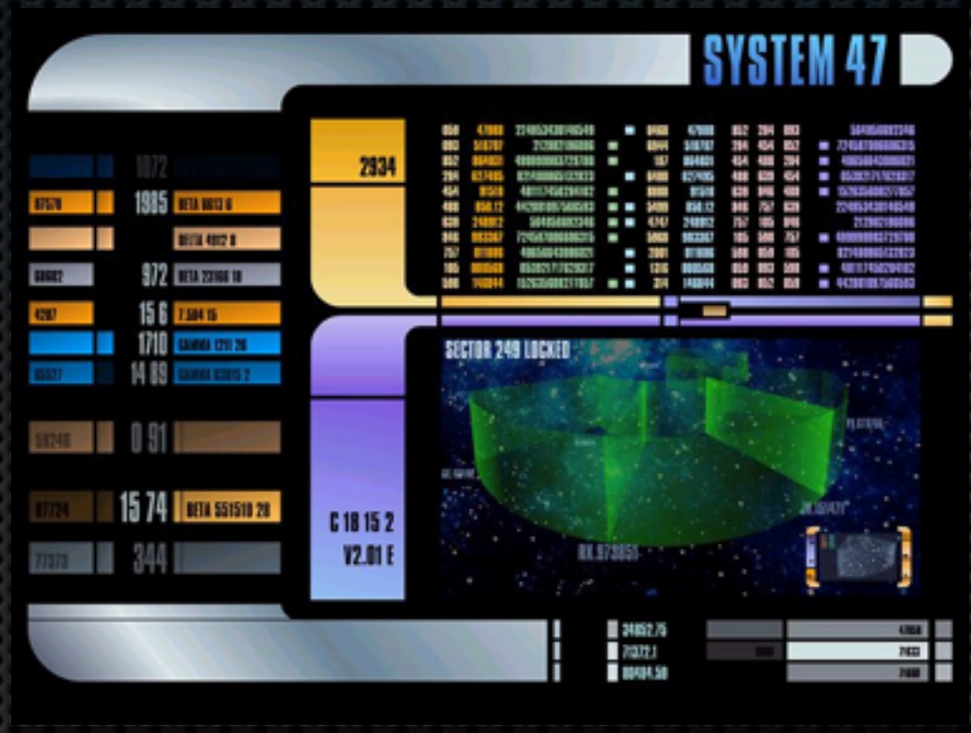
- Use `setup.py config` to show the current settings




```
> python setup.py config
running config
Organisation Config
    pypi_url: http://pypi.acme.example
    namespaces: ['acme']
    namespace_separator: .
    email_suffix: acme.example
    dev_build_number: 0.0
    platform_packages: []
    installer_search_path: []
    default_platform_package: None
    deploy_path: /home/eeaston/packages
 ...
```

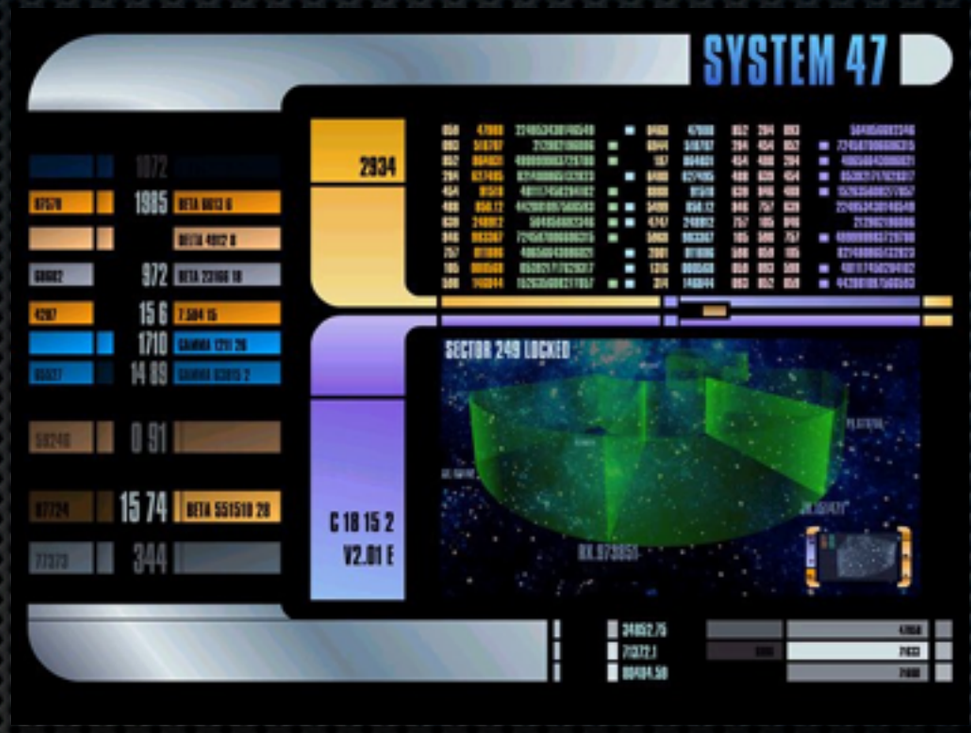
Configuration

- All package metadata is stored in `setup.cfg`




```
[metadata]
name = acme.utils
version = 1.0.0
author = Edward Easton
author_email = ed@acme.com
description = Acme Co's famous utilities
install_requires =
    numpy
console_scripts =
    acmectl=acme.utils.scripts.acmectl:main
```

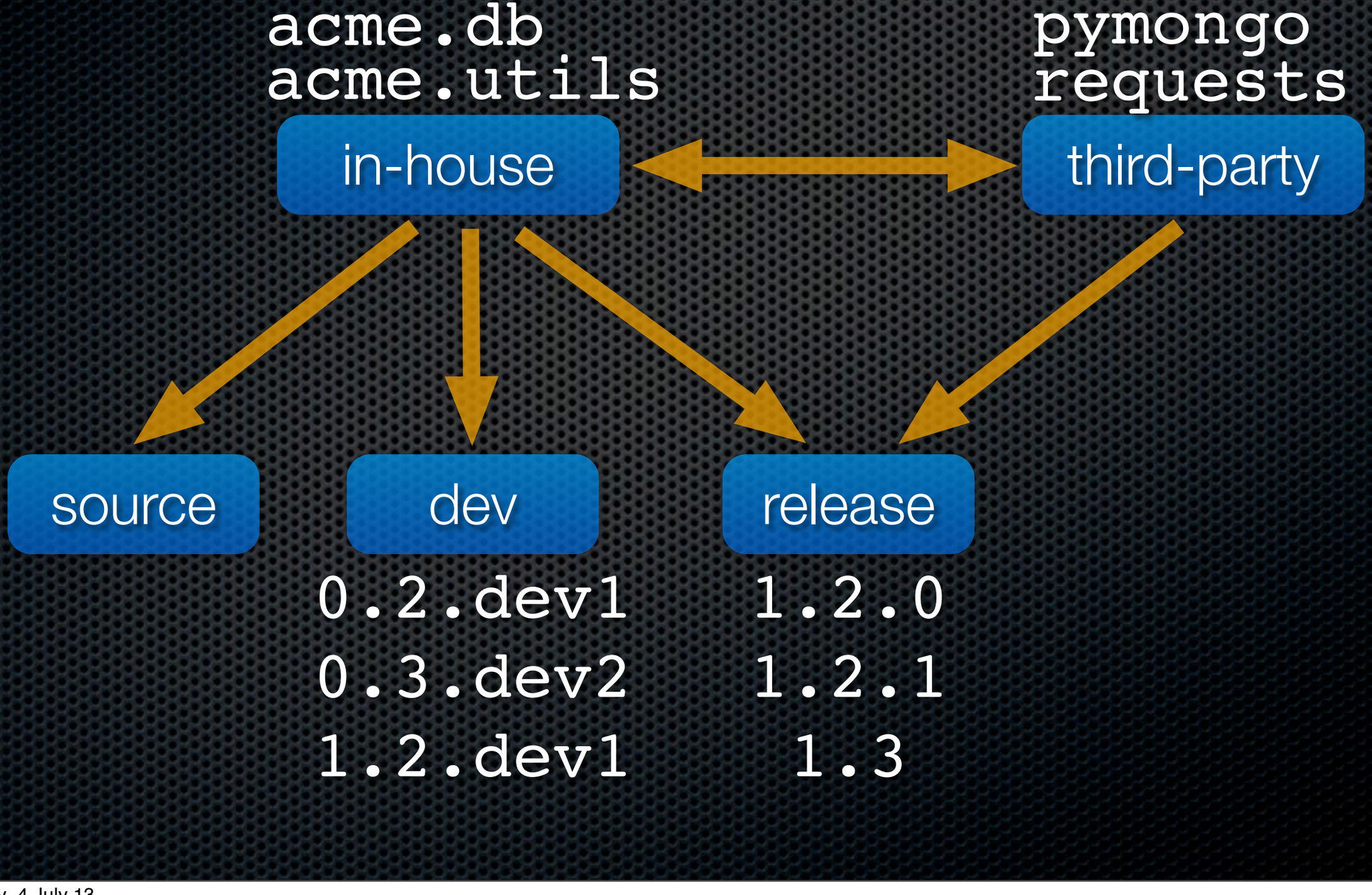
Configuration



- No more hacking on `setup.py` required (unless you have C-extensions)

```
> cat setup.py
#!/usr/bin/env python
from pkglib import setup
setup()
```

Package names and versions



Package names and versions

acme.db
acme.utils

pymongo
requests

in-house

third-party

source

dev

0.2.d

0.3.d

1.2.dev1

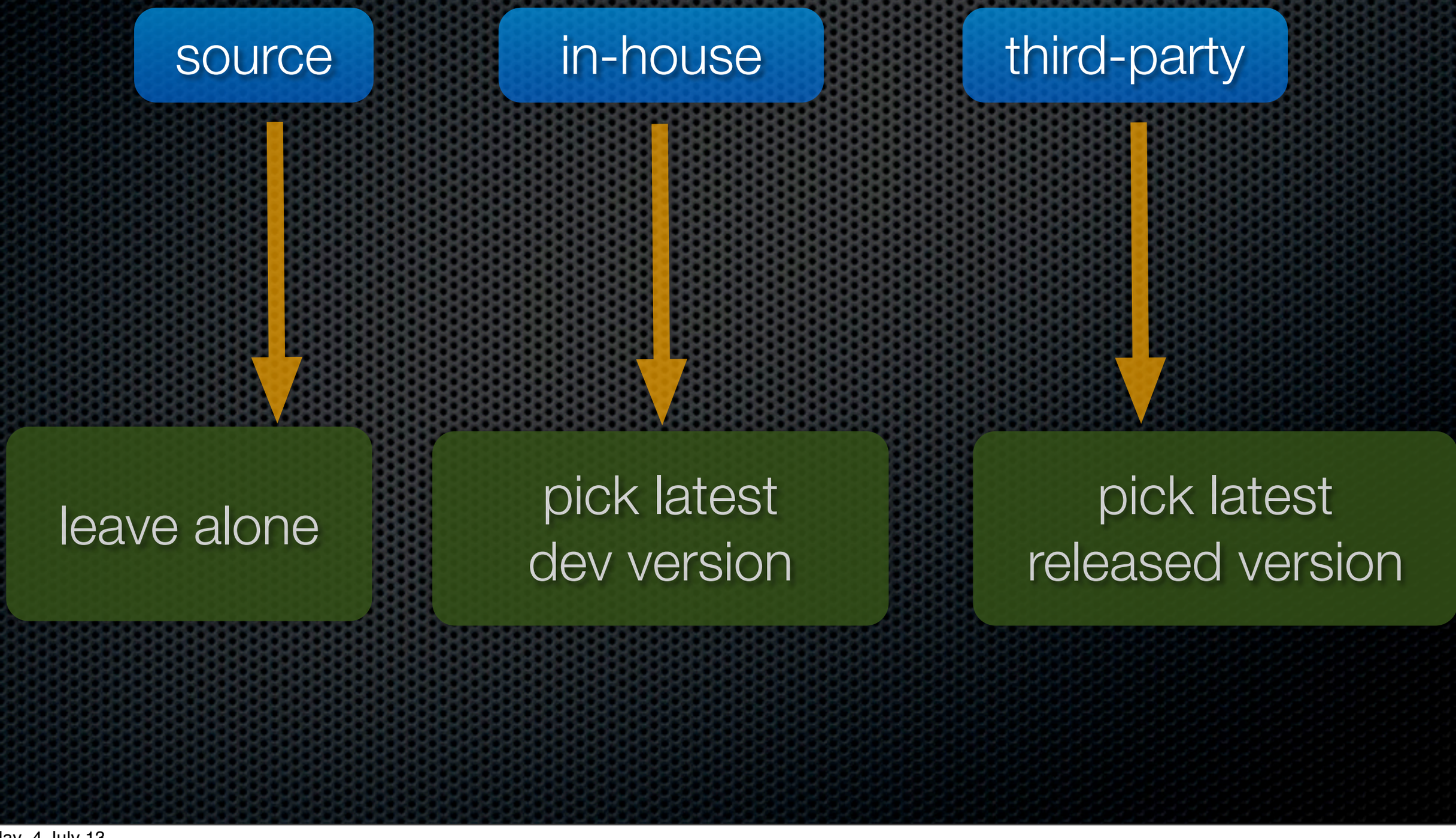
1.3

```
# Acme Co. pkglib configuration file
[pkglib]
namespaces =
    acme
    blackmesa
    aperture
namespace_separator = .
...
```

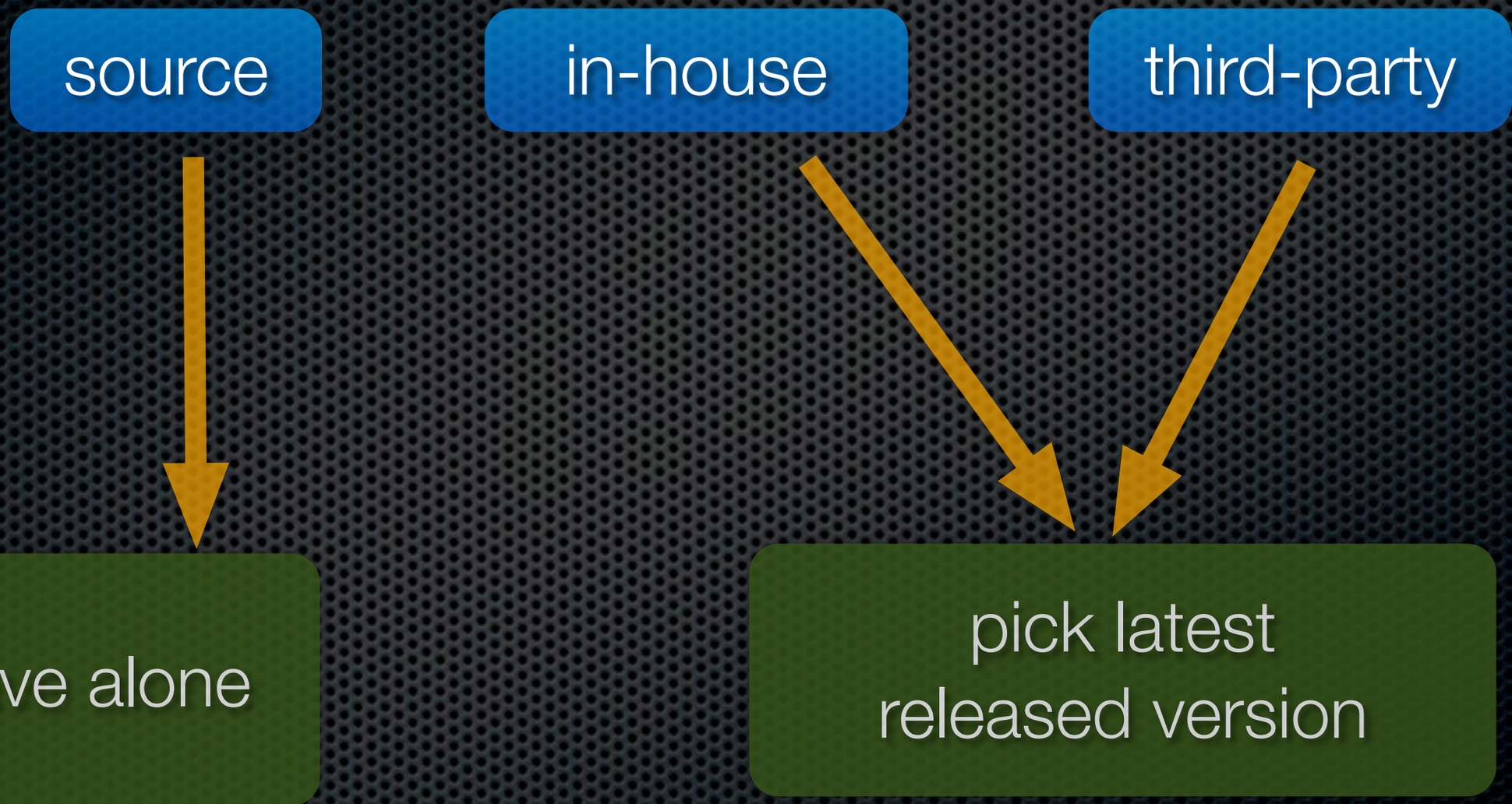
Dependency Resolution

- Ostensibly like `pip/easy_install`, but:
  - We can operate in ‘dev’ or ‘release’ mode:
    - `pyinstall <pkg>`: ‘release’ mode
    - `pyinstall --dev <pkg>`: ‘dev’ mode
    - `python setup.py develop`: ‘dev’ mode
  - Backtracking resolver to solve the ‘diamond problem’
  - Defers download of build targets requirements

‘Dev’ mode

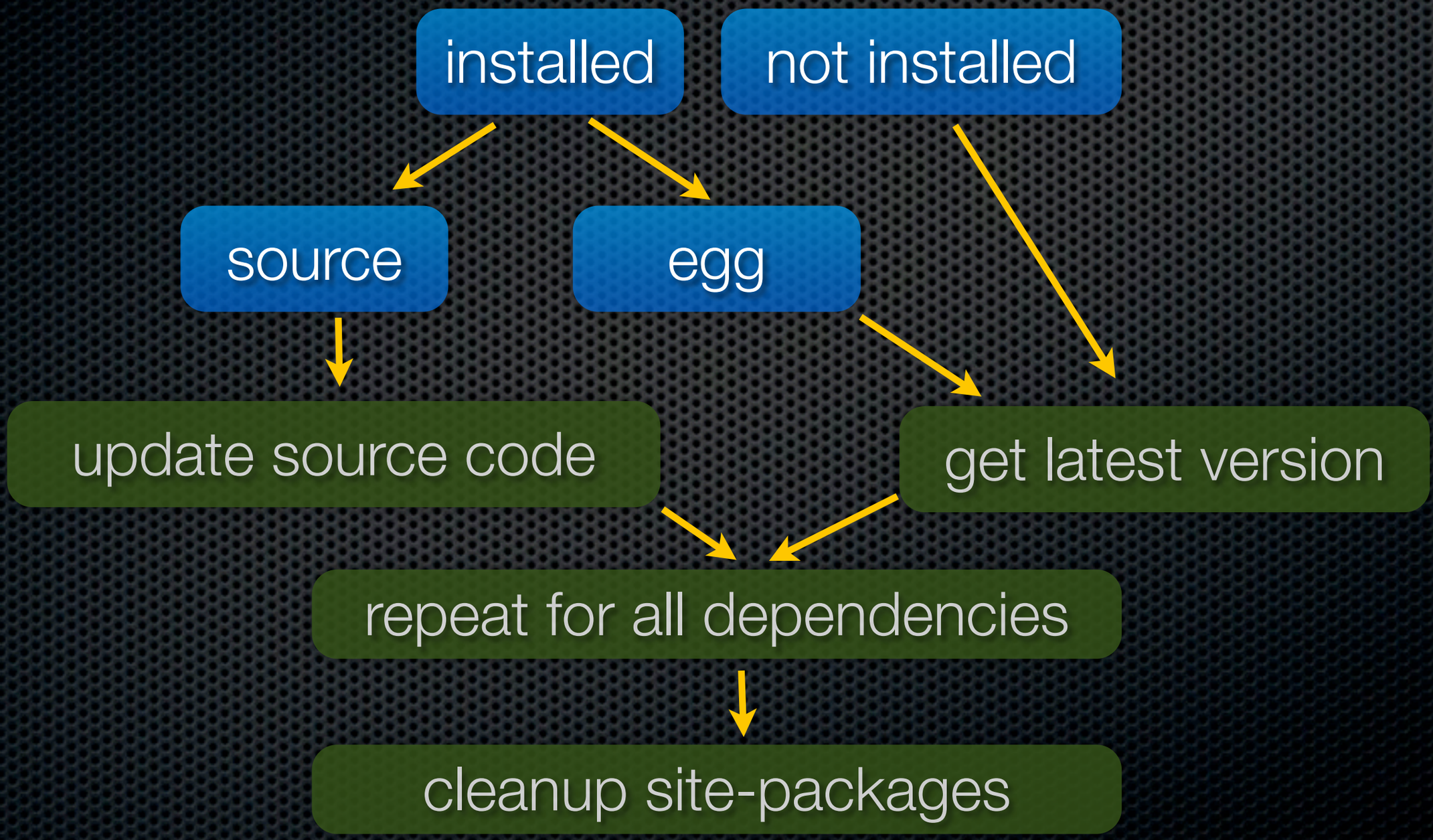

source
in-house
third-party
leave alone
pick latest
dev version
pick latest
released version

‘Release’ mode

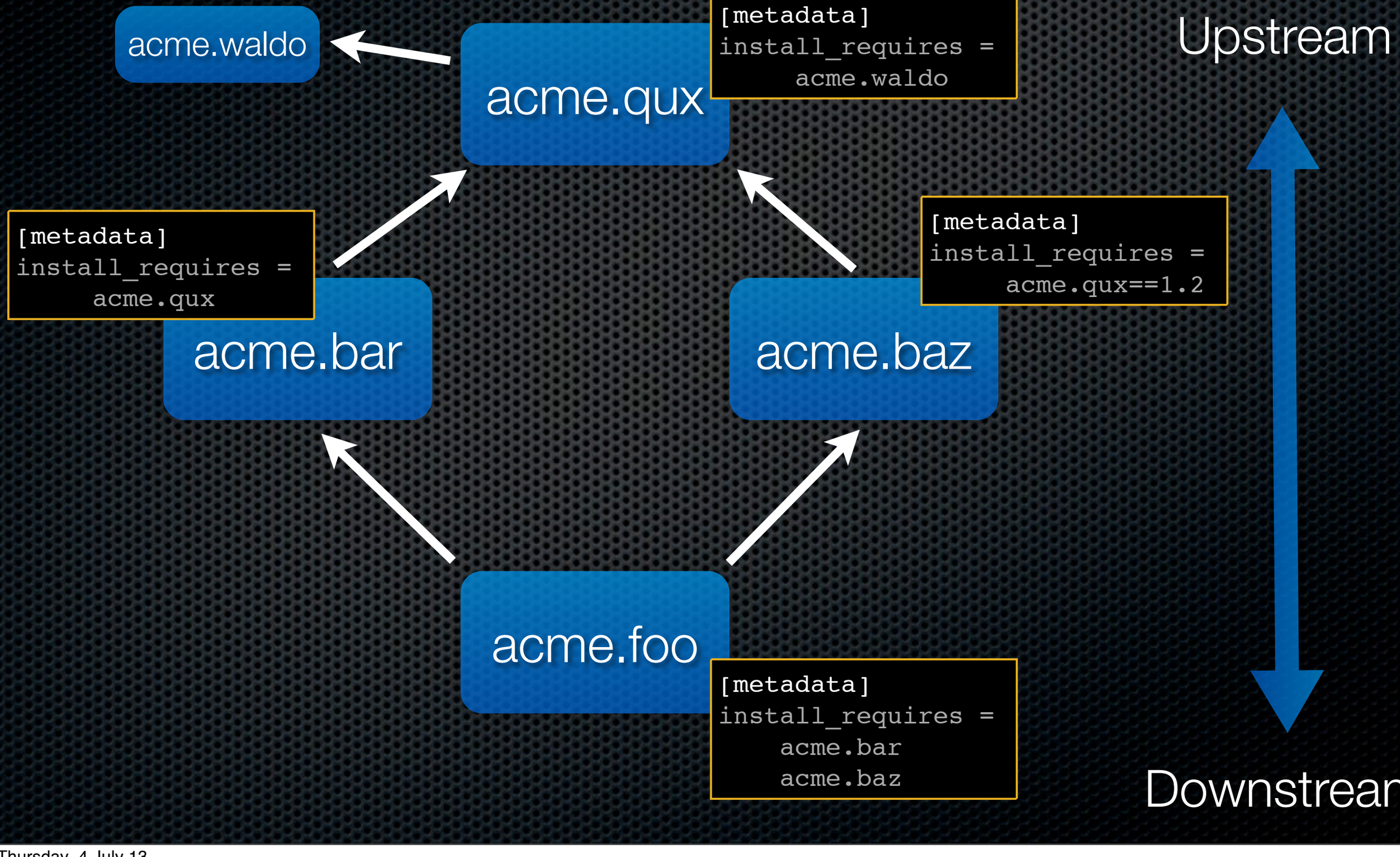


`pyinstall` workflow

```
$ pyinstall acme.utils
```



A sample dependency graph



The ‘diamond’ problem

acme.waldo-3.0

acme.qux-1.3

acme.waldo-2.0

Upstream

acme.qux-1.2

```
[metadata]
install_requires =
    acme.qux
```

```
[metadata]
install_requires =
    acme.qux==1.2
```

acme.bar

acme.baz

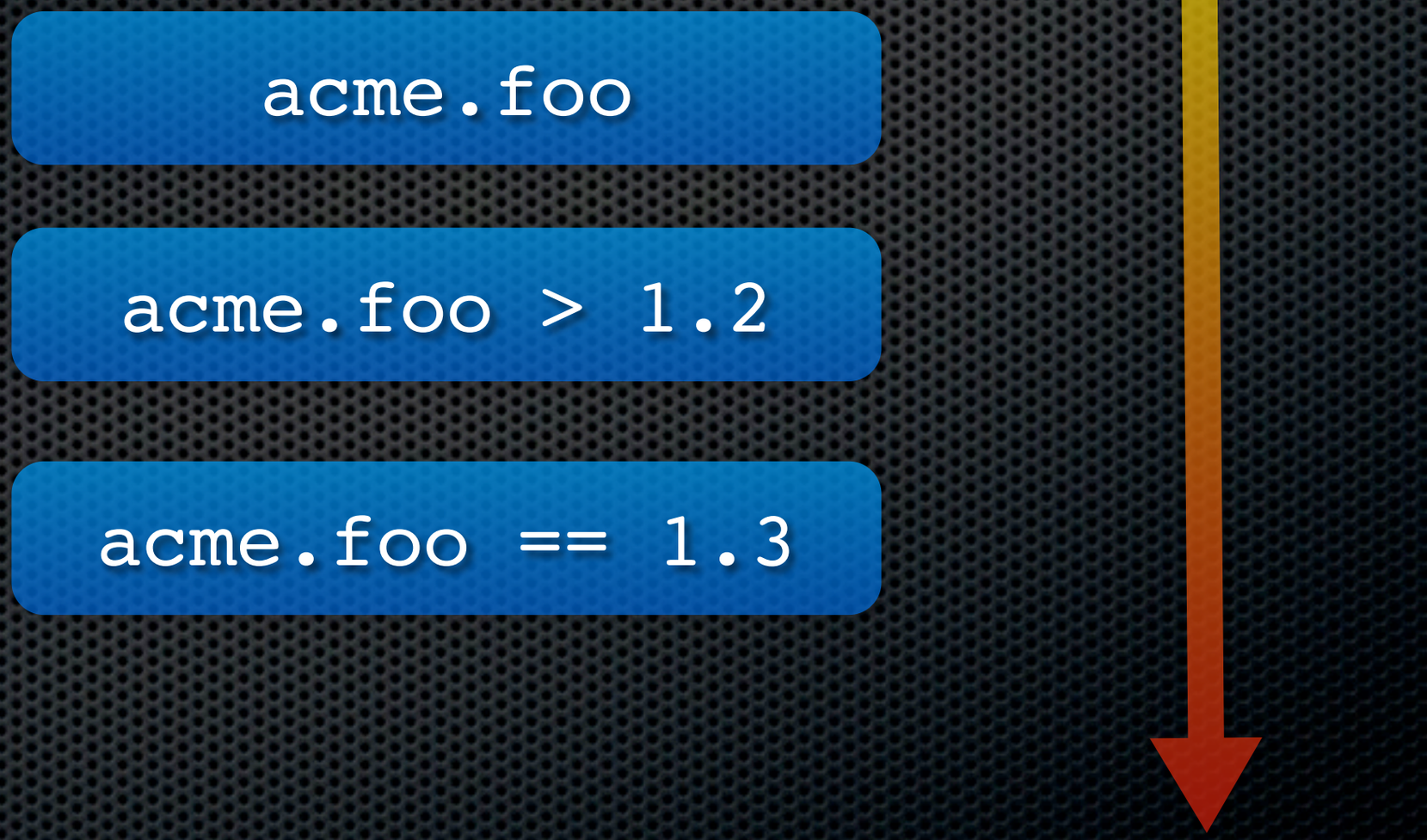
acme.foo

```
[metadata]
install_requires =
    acme.bar
    acme.baz
```

Downstream

Requirement specificity

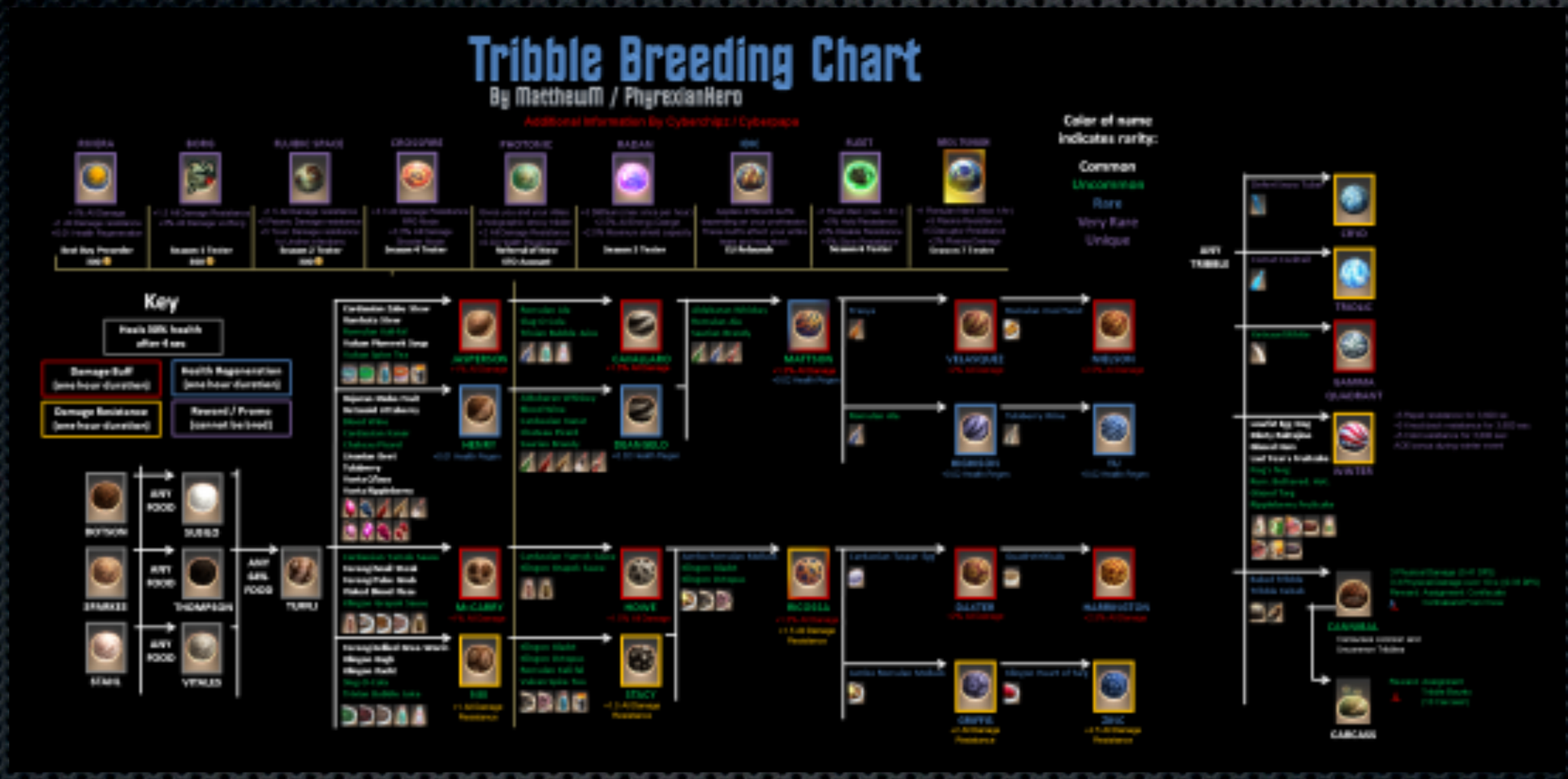
Less specific



More specific

Why all the bother?

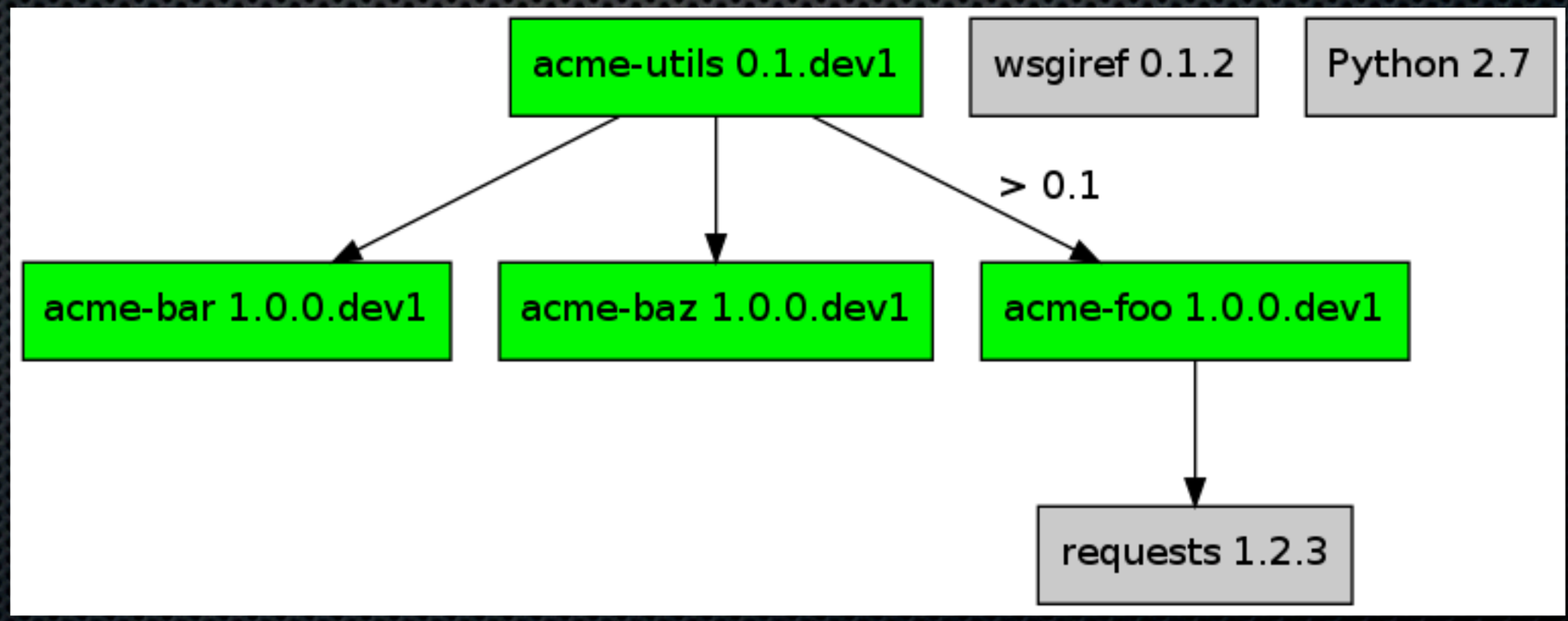
Why all the bother?



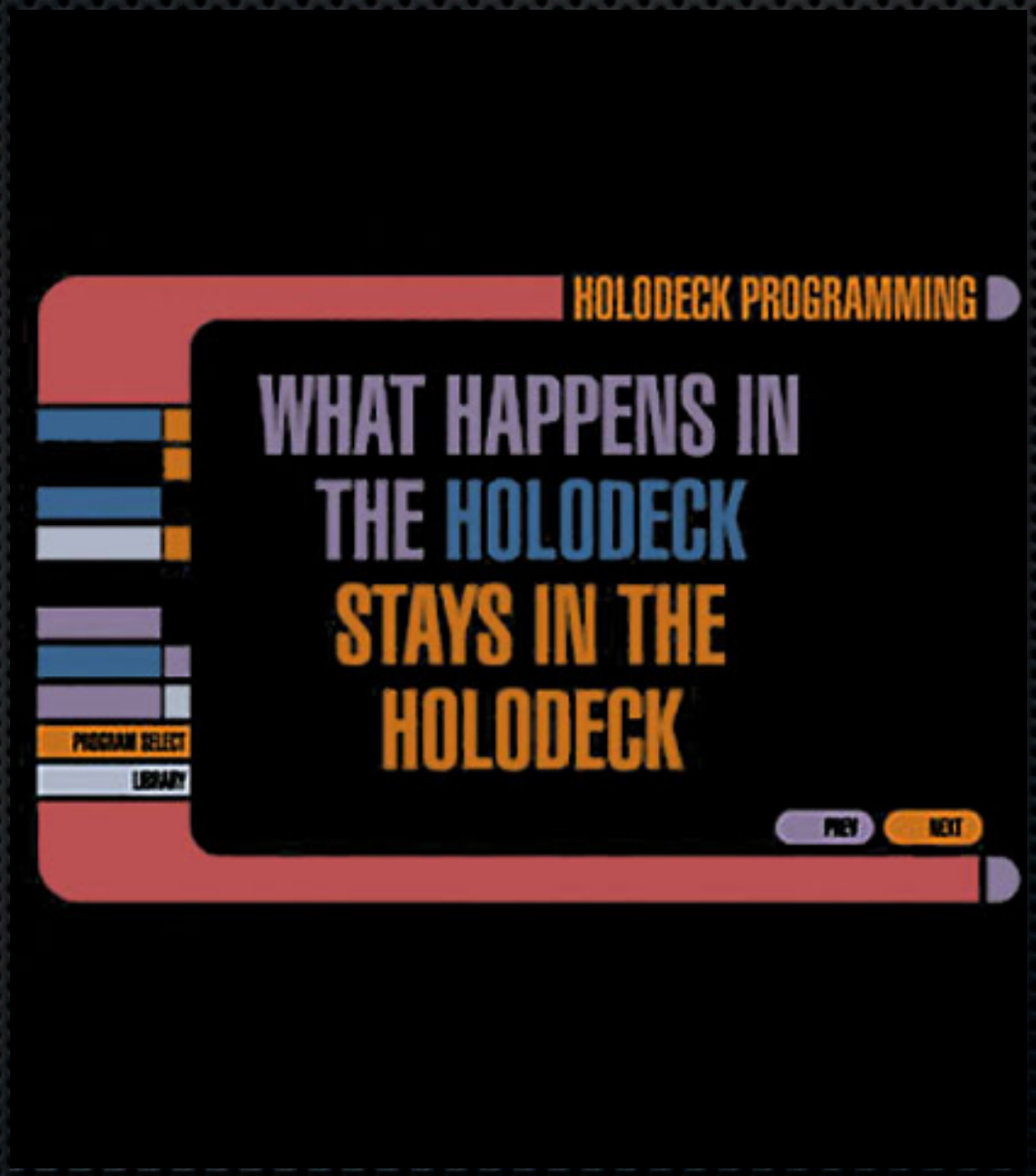
Why all the bother?

pydegraph

```
> pydepgraph -a
#-- Dependency Graph: note - includes only installed
packages ----#
#-- Rendering using pydot ------------------------------#
Reading http://pypi.python.org/simple/pydot/
Rendering graph to /tmp/tmpTQ543H/graph.png
```



Installer search path



Installer search path

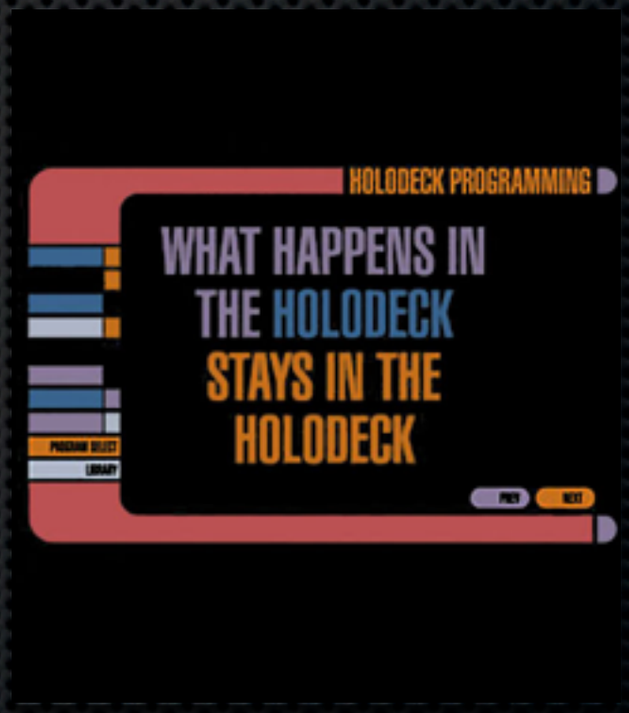
```
# Acme Co. pkglib configuration file
[pkglib]
installer_search_path =
    /opt/eggs
    ${HOME}/eggs
    ...
```

```
lib/python27/site-packages
    numpy-1.6.egg-link
    acme.utils-1.2.egg-link
```

```
/opt/eggs/
    n/
        numpy-1.6.egg
        numpy-1.6.1.egg
    a/
        acme.utils-1.2.egg
        acme.utils-1.3.egg
```

Installer search path

```
# Acme Co. pkglib configuration file
[pkglib]
installer_search_path =
    /opt/eggs
    ${HOME}/eggs
    ...
```




```
lib/python27/site-packages
    numpy-1.6.egg-link
    acme.utils-1.2.egg-link
```

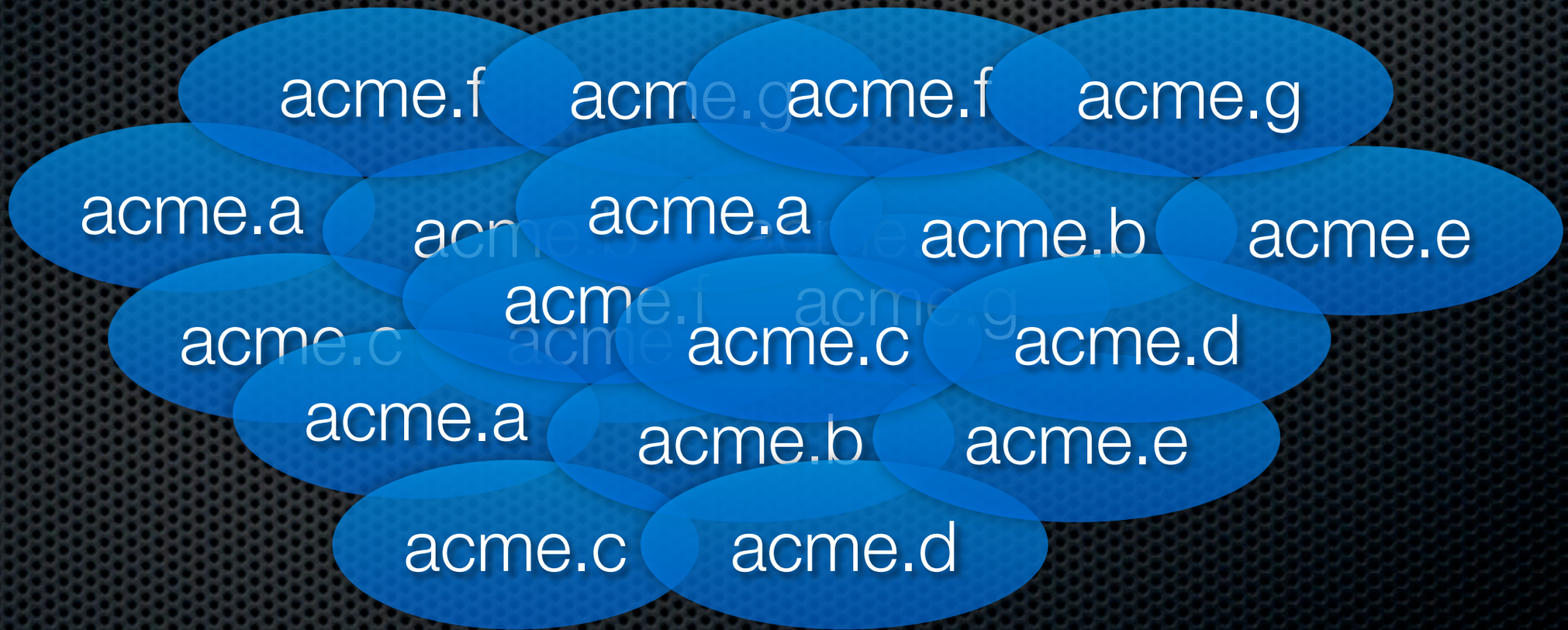
```
/opt/eggs/
    n/
        numpy-1.6.egg
        numpy-1.6.1.egg
    a/
        acme.utils-1.2.egg
        acme.utils-1.3.egg
```

pycheckout

```
$ pycheckout acme.utils
Checking out acme.utils from http://acme-svn/acme.utils/trunk
Setting up package in ./acme.utils
Done!
```

```
$ cat acme.utils/acme_utils.egg-info/PKG-INFO
Name: acme-utils
Version: 1.0.0.dev1
Summary: Acme Co. Utilities
Home-page: http://acme-svn/acme.utils
Author: Edward Easton
Author-email: ed@acme.com
License: Proprietary
Description: UNKNOWN
Platform: UNKNOWN
Classifier: Topic :: Software Development ::
Libraries
```

plat


acme.f
acme.g
acme.f
acme.g
acme.a
acme.a
acme.b
acme.e
acme.c
acme.c
acme.d
acme.a
acme.b
acme.e
acme.c
acme.d

plat

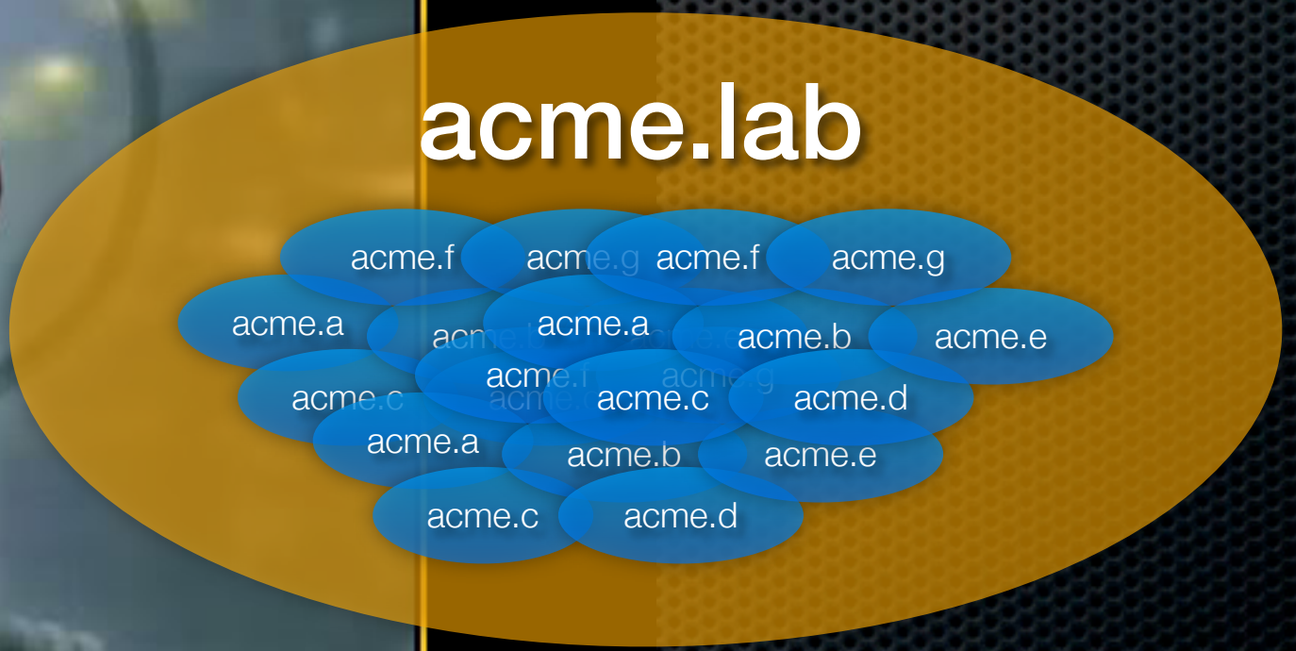
acme.lab

BARACKUS OF BORG

YOU AND YOUR PAYCHECK WILL BE ASSIMILATED.
RESISTANCE IS FUTILE.

plat


BARACKUS OF BORG
YOU AND YOUR PAYCHECK WILL BE ASSIMILATED.
RESISTANCE IS FUTILE.


acme.lab
acme.f
acme.g
acme.f
acme.g
acme.a
acme.a
acme.b
acme.e
acme.c
acme.c
acme.d
acme.a
acme.b
acme.e
acme.c
acme.d

plat

```
# Acme Co. pkglib configuration file
[pkglib]
platform_packages =
    acme.lab
    blackmesa.lab
```

```
$ plat info
acme.lab: 2.3.0
blackmesa.lab: not installed
Other source checkouts:
    acme-bar: 1.0.0.dev1
    acme-utils: 1.0.0.dev1
```

plat

```
$ plat --help up
usage: plat [-h] [-q] [-d]
            {use,up,develop,undevelop,info,list,versions,components} ...

Manages the installation of platform packages and their components

positional arguments:
  {use,up,develop,undevelop,info,list,versions,components}
    use                 Makes a deployed platform package available in the
                        currently active virtualenv
    up                  Updates all dev packages that are installed in the
                        current virtualenv
    develop             Makes the source of a package available in the current
                        virtualenv.
    undevelop           Disables the source of a package from the current
                        virtualenv.
    info                Shows details of the platform packages that are active
                        in the current virtualenv.
    list                Shows the available platform packages.
    versions            Shows the versions of a platform package.
    components          Shows the packages that are components of a platform
                        package.
```

Running tests

```
$ python setup.py test
Pytest args: --verbose --cov=acme.utils --
cov=acme.utils.scripts --cov-report=term /home/vagrant/pycon/
acme-utils/tests
============= test session starts ==============================
tests/unit/test_sample.py:5: test_import PASSED
tests/unit/test_sample.py:11: test_app PASSED
-- coverage: platform linux2, python 2.7.3-final-0 -----------
Name                                Stmts   Miss  Cover
-------------------------------------------------------
acme/utils/__init__                     1      0   100%
acme/utils/example                      1      1     0%
acme/utils/scripts/__init__             1      0   100%
acme/utils/scripts/app                  9      1    89%
-------------------------------------------------------
TOTAL                                  12      2    83%

============ 2 passed in 0.03 seconds ==========================
```

Test discovery

```
# Acme Co. pkglib configuration file
[pkglib]
test_dirname = tests
...
```

```
$ cd acme-utils; find .
./setup.py
./acme/
    utils/
        foo.py
        bar.py
./tests/
    unit/
        test_foo.py
        test_bar.py
    integration/
        test_baz.py
```

```
$ cd acme-utils; find .
./setup.py
./acme/utils/
    foo.py
    bar.py
    tests/
        unit/
            test_foo.py
            test_bar.py
        integration/
            test_baz.py
```

Running tests

```
$ python setup.py test --unit
```

```
$ python setup.py test --integration
```

```
$ python setup.py test --regression
```

```
$ python setup.py test --doctest
```

```
$ python setup.py test
```

coverage.xml

junit.xml

pylint.xml

Testing C/C++ with Gcov

This is awesome

```
$ python setup.py ext_gcov_test

============== test session starts ==============================
tests/unit/test_ext.py:5: test_gcov PASSED
------------------------------------------------------------------
File                                Lines    Exec Cover
------------------------------------------------------------------
src/ext/ext.c                          28      28  100%
------------------------------------------------------------------
TOTAL                                  28      28  100%

============ 1 passed in 0.04 seconds ============================
```

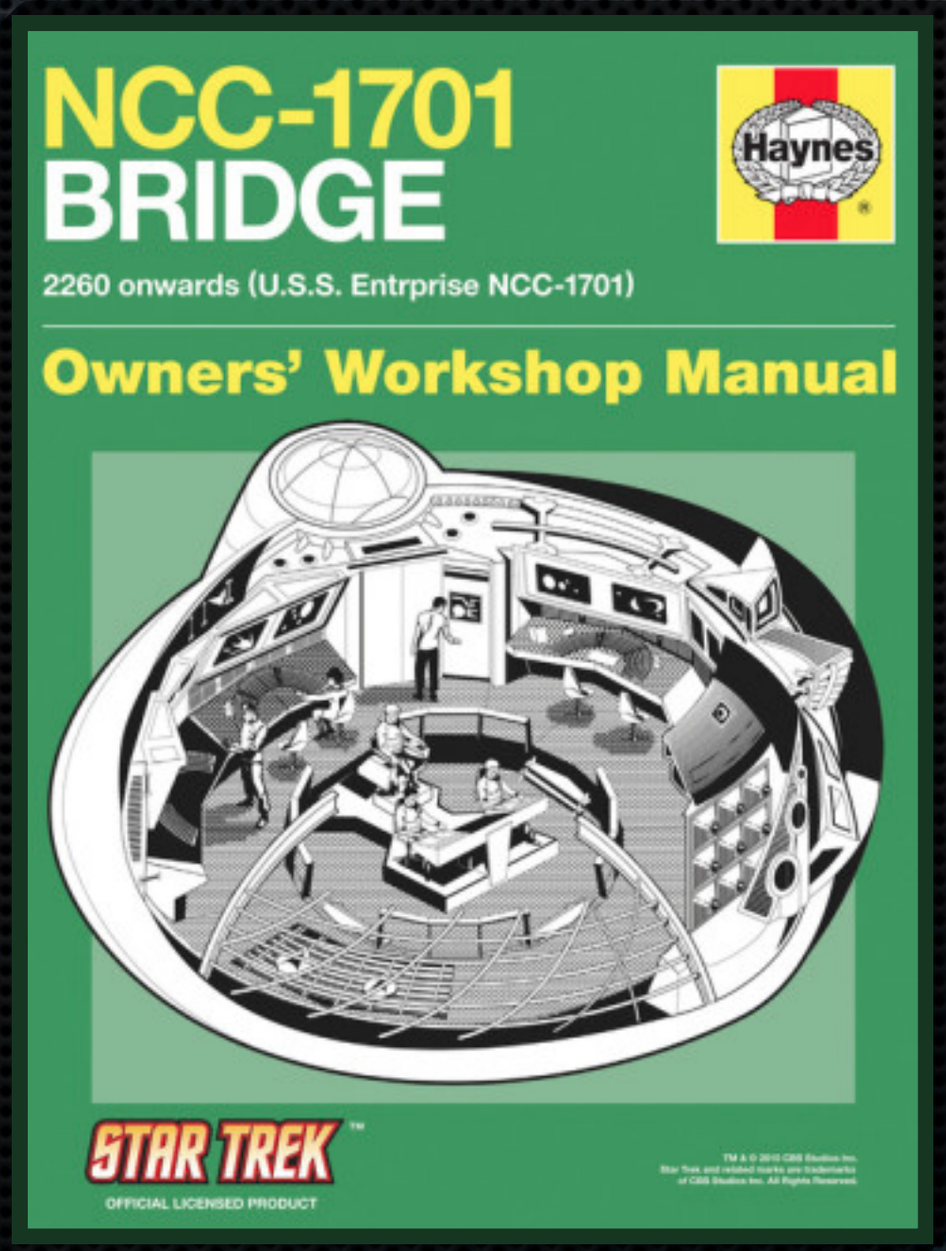
Project Templates

```
$ pymkproject acme-utils
Enter version (Version (like 1.0)) ['1.0.0']: 0.1
Enter description (One-line description of the package) ['']:
Acme Co. Utilities
Enter author (Author Name) ['Your Name']: Edward Easton
Enter author_email (Author Email) ['ed@acme.com']: ed@acme.com
Enter namespace_separator (Namespace Separator) ['-']:
Creating template pkglib_project
Creating directory ./acme-utils
......
Successfully created ./acme-utils
```

Project Templates

```
$ cd acme-utils; find .
./setup.cfg
./setup.py
./acme/
    __init__.py
    utils/
        example.py
    scripts/
        app.py
        __init__.py
./docs/
    conf.py
    index.rst
./tests/
    unit/
       test_sample.py
    integration/
```

Project Documentation



Project Documentation

```
$ python setup.py build_sphinx
running build_sphinx
Reading http://pypi.python.org/pypi/numpydoc/
Auto-generating documentation from docstrings.
...
Running Sphinx v1.2b1
...
running tests...

Document: autodoc/acme.utils
----------------------------
1 items passed all tests:
   1 tests in default
1 tests in 1 items.
1 passed and 0 failed.
Test passed.

Doctest summary
===============
    1 test
    0 failures in tests
    0 failures in setup code
    0 failures in cleanup code
build succeeded.
```

Project Documentation

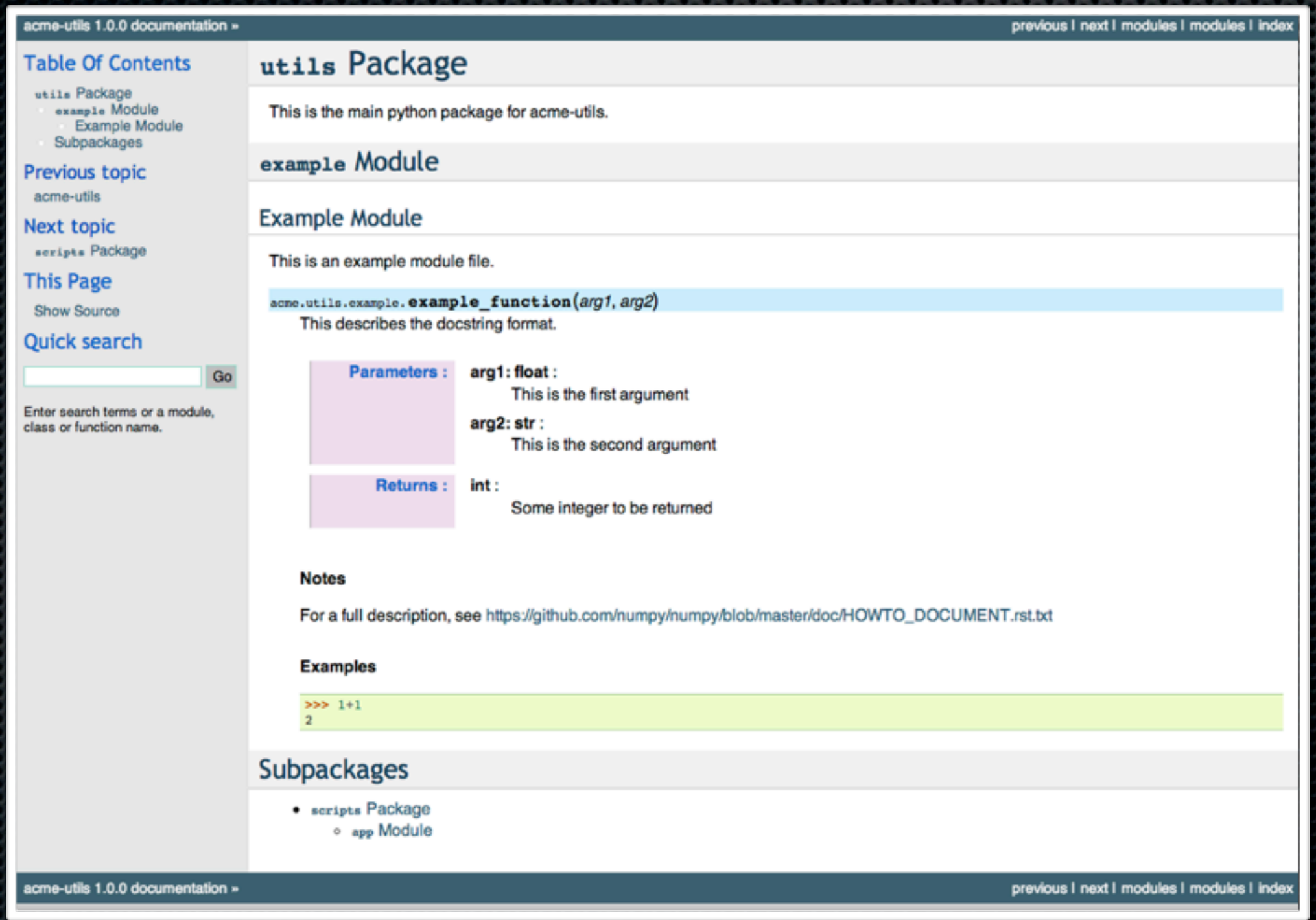
```
$ cd acme-utils; find .
./acme/
    utils/
        scripts/
            __init__.py
        example.py
        _private.py
./docs
    autodoc/
        acme.utils.rst
        acme.utils.scripts.rst

    index.rst
    recipes.rst
    howto.rst
```

autodoc

skipped

standalone

Project Documentation



acme-utils 1.0.0 documentation »

previous | next | modules | modules | index

Table Of Contents

- utils Package
  - example Module
    - Example Module
  - Subpackages

Previous topic

acme-utils

Next topic

scripts Package

This Page

Show Source

Quick search

Go

Enter search terms or a module, class or function name.

utils Package

This is the main python package for acme-utils.

example Module

Example Module

This is an example module file.

acme.utils.example.**example_function**(*arg1*, *arg2*)

This describes the docstring format.

Parameters : **arg1: float :**
This is the first argument
arg2: str :
This is the second argument

Returns : **int :**
Some integer to be returned

Notes

For a full description, see https://github.com/numpy/numpy/blob/master/doc/HOWTO_DOCUMENT.rst.txt

Examples

```
>>> 1+1
2
```

Subpackages

- scripts Package
  - app Module

acme-utils 1.0.0 documentation »

previous | next | modules | modules | index

Project Documentation

```
def example_function(arg1, arg2):
    """ This describes the docstring format.

    Parameters
    ----------
    arg1:  float
      This is the first argument
    arg2:  str
      This is the second argument

    Returns
    -------
    int
         Some integer to be returned

    Examples
    --------
    >>> 1+1
    2

    Notes
    -----
    For a full description, see
    https://github.com/numpy/numpy/blob/master/doc/HOWTO_DOCUMENT.rst.txt

    """
```

Build and deploy tools

```
# Acme Co. pkglib configuration file
[pkglib]
jenkins_url = http://jenkins.acme.com
```

- **`python setup.py jenkins`**
- Looks up the VCS address just like **`pycheckout`**
- Creates or updates a build on jenkins that runs through the standard set of develop, test, build_sphinx and upload steps.
- Build is setup with config for coverage (Cobertura), test reporting (Xunit) and code quality (Violations)

Build and deploy tools

```
# Acme Co. pkglib configuration file
[pkglib]
deploy_path = /opt/python/packages
deploy_bin = /opt/python/bin
```

- **`python setup.py deploy`**
- Creates a versioned virtualenvs under **`$CONFIG.deploy_path`**
- Maintains the '**`current`**' symlink to the latest version
- Symlinks selected console scripts into **`$CONFIG.deploy_bin`**

Build and deploy tools

```
# Acme Co. pkglib configuration file
[pkglib]
test_egg_namespace = acmetest
```

- Somewhat experimental!
- **`python setup.py test_egg`**
- Generates an egg that contains **only** tests and any test runner config from **`setup.cfg`**
- Installing this egg provides the **`runtests`** script to do the equivalent of '**`python setup.py test`**' from your source checkout
- Useful for testing hot-fixes to released packages

Build and deploy tools

```
# Acme Co. pkglib configuration file
[pkglib]
dev_build_number = 0.0
```

- Also somewhat experimental!
- **`python setup.py egg_info --new-build`**
  - Generates package metadata that is the equivalent of '`pip freeze`'
  - Queries PyPI server to generate a new version number for this package that is unrelated to the current version number stream.
  - Eg: `0.0.dev1, 0.0.dev2` etc.

pkglib.testing

- Managing temporary directories
- Creating virtualenvs and pkglib-enabled packages
- Running up servers instances in a port-safe manner, with save, restore and teardown. Mongo, redis, jenkins and pyramid implementations.
- Selenium Webdriver, integrated with the Pyramid server runner fixture.
- Page Objects pattern implementation for better structured Selenium tests
- Mocking utils for databases and other common types

`pkglib` Feature Summary

- `pyinstall <pkg>` : install latest released version
- `pyinstall --dev <pkg>` : install latest dev version
- `pyuninstall <pkg>` : remove a package
- `pymkproject acme.utils.db` : create projects
- `plat info|use|up|list` : manage platform packages
- `pycheckout <pkg>` : checkout and setup package
- `pydepgraph [<pkg>]` : show dependency graphs
- `pycleanup` : clean up your virtualenv

`pkglib` Feature Summary

- Numerous `setup.py` targets:
  - `test` : run tests under `py.test` with coverage reports, unit/integration/regression modes
  - `update` : synchronise this package and dependencies with VCS repository and PyPI server
  - `build_sphinx` : generate standalone plus autodoc formatted html.
  - `jenkins` : create or update Jenkins/Hudson builds
  - `deploy` : deploy this package into a new virtualenv
  - `test_egg` : build a tests-only egg package

Where to from here?

- Depends on you guys!
- Complete the OSS export?
- Full support for git and mercurial
- OSX and Windows support
- Python 2.4 -> 3.x support
- Bring the project in-line with recent developments in the Python packaging space:
  - wheel binary distribution format
  - integrate with `distlib` library?

Thanks for listening!

@syspython

http://github.com/eeaston/pkglib

http://readthedocs.org/projects/pkglib/

https://groups.google.com/forum/#!forum/pkglib