Pyne

A dependency graph creator engine from git repositories

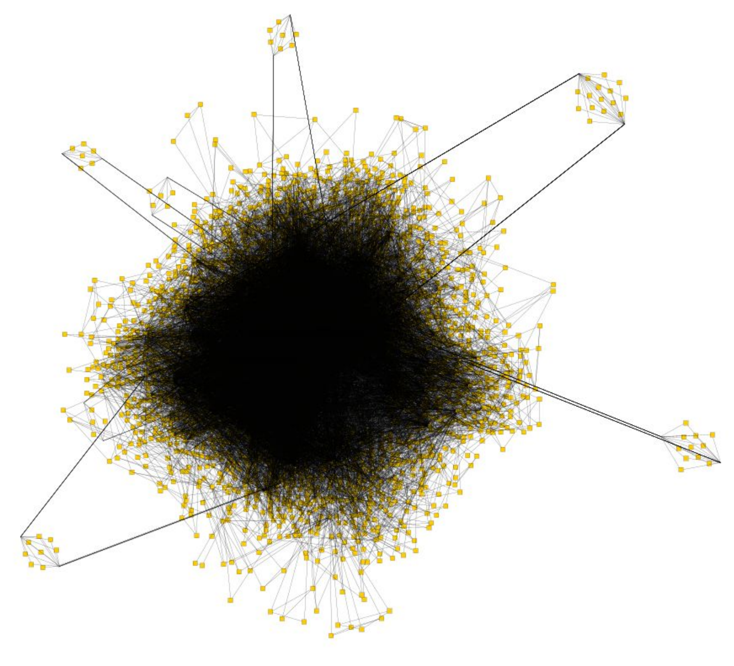
Job Heersink
Richard Westerhof

Content

- Introduction
- Design
- Spoon
- Architectural Requirements
- Additional Requirements
- Technologies
- Conclusion

Introduction

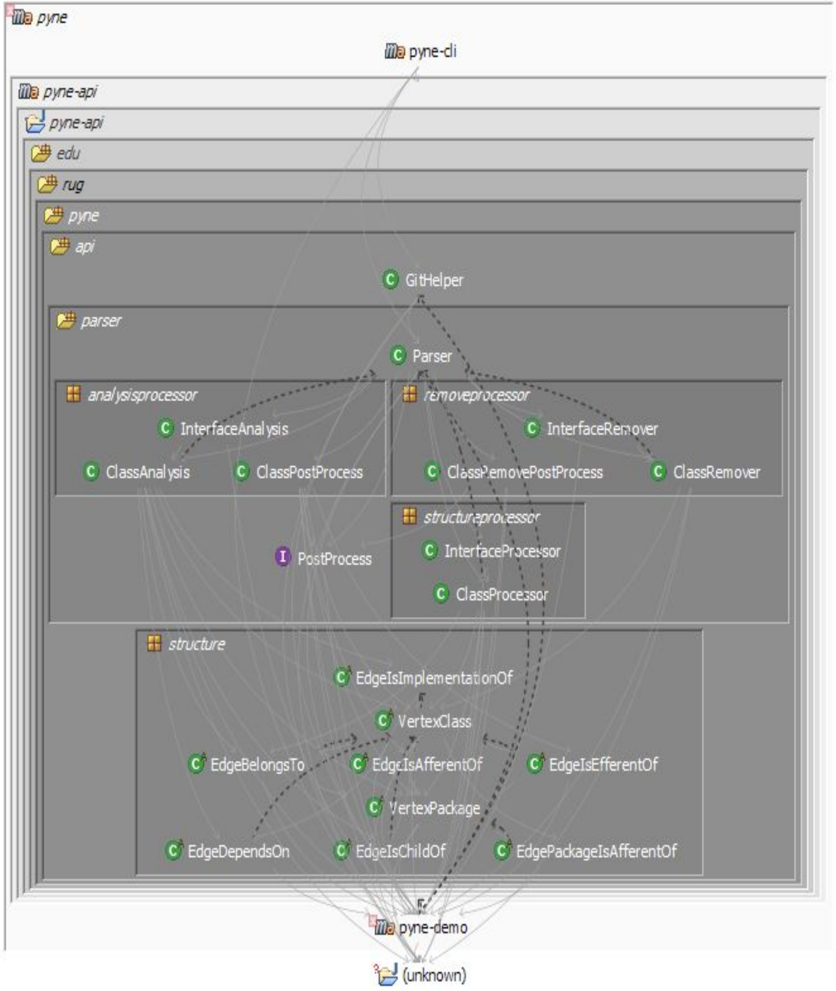
- Subject of a Bachelor Thesis in 2019
- Java
- Built in git support
- Creates graphml files



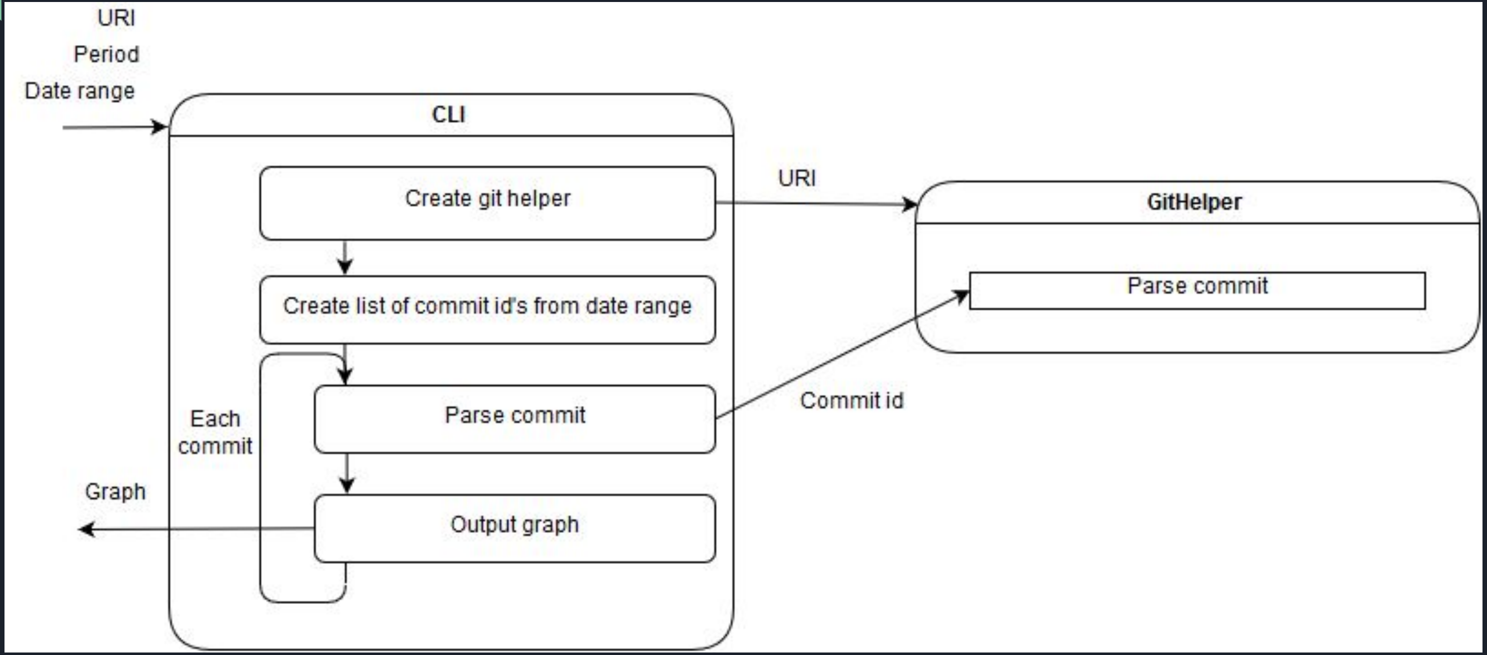
The output of Pyne on Apache Tajo

Design

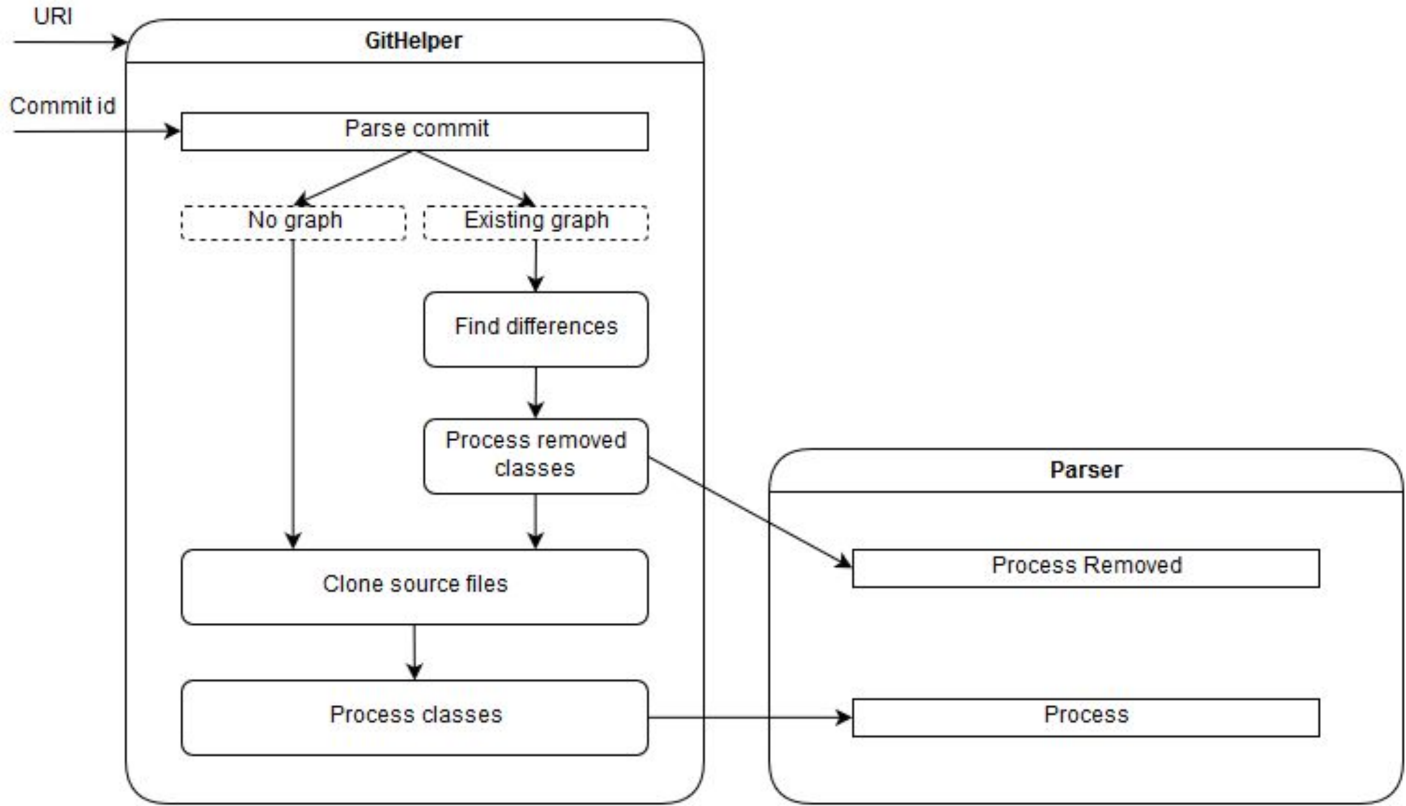
- Consists of 3 packages: pyne-api, pyne-cli, pyne-demo
- pyne-api is the internal functionality package
- Main components:
  - GitHelper (optional)
  - Parser
  - Analysis
  - Build Internal Graph
  - Output



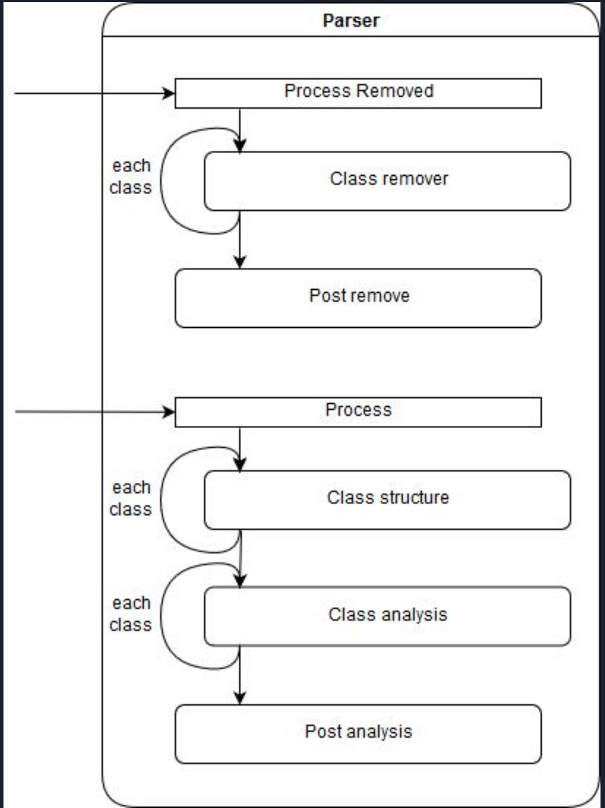
Design


URI
Period
Date range
CLI
Create git helper
Create list of commit id's from date range
Parse commit
Output graph
Each commit
Graph
URI
GitHelper
Parse commit
Commit id

Design


URI
GitHelper
Commit id
Parse commit
No graph
Existing graph
Find differences
Process removed classes
Clone source files
Process classes
Parser
Process Removed
Process

Design


Parser
Process Removed
each class
Class remover
Post remove
Process
each class
Class structure
each class
Class analysis
Post analysis

Spoon

- Source code parsing library
- AST optimized for code analysis
- Split in 3 parts
  - Structure
  - Code
  - Reference

http://spoon.gforge.inria.fr/

Spoon (Structure)

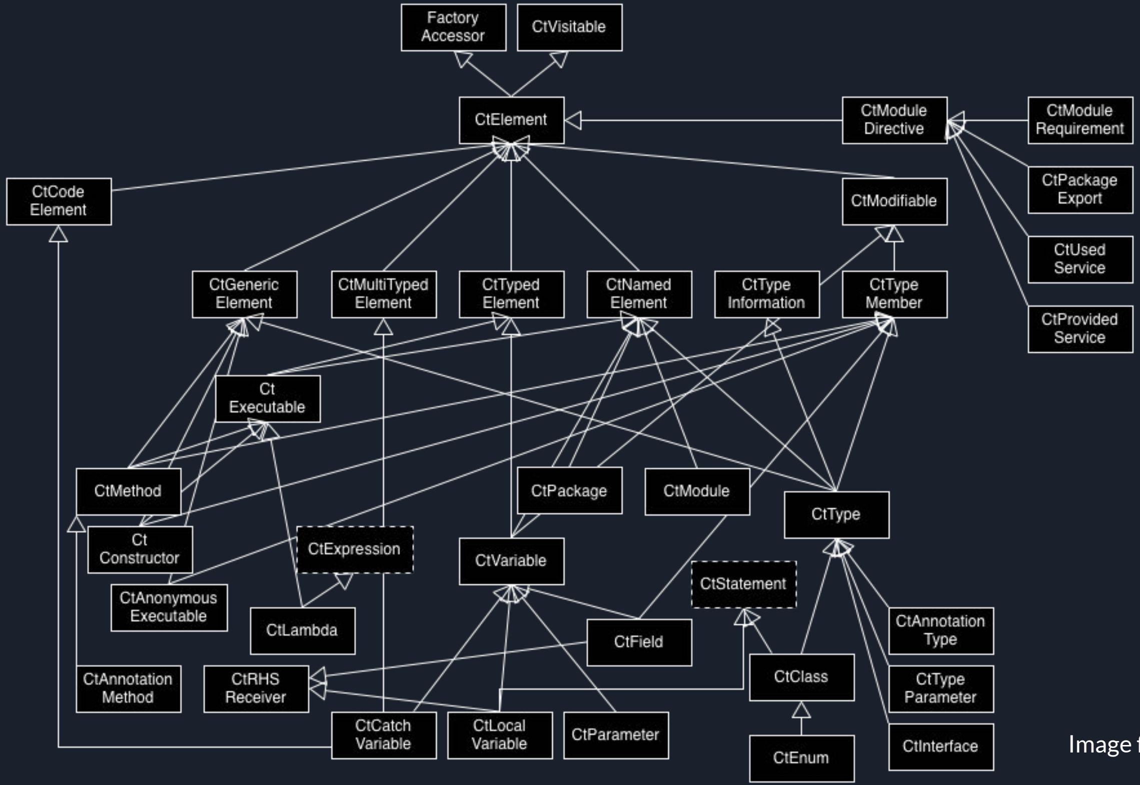


Image from: http://spoon.gforge.inria.fr/

Spoon (Code)

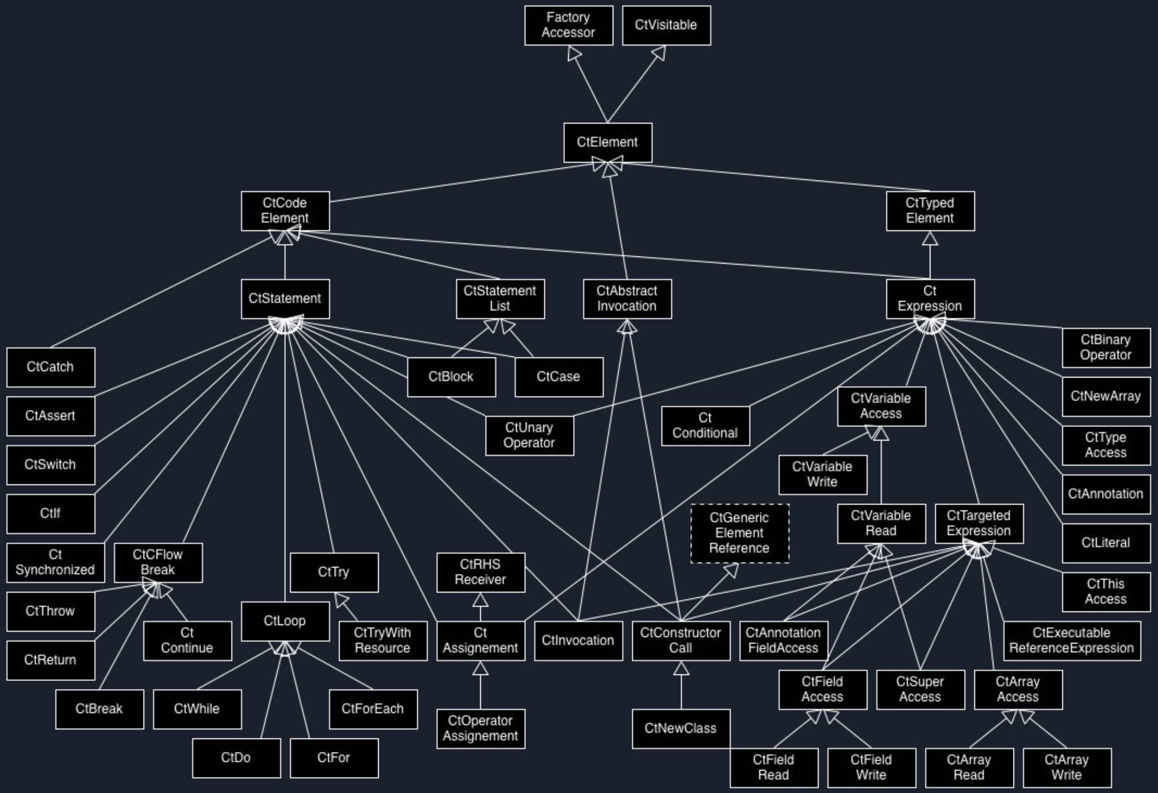


Image from:
http://spoon.gforge.inria.fr/

Spoon (Reference)

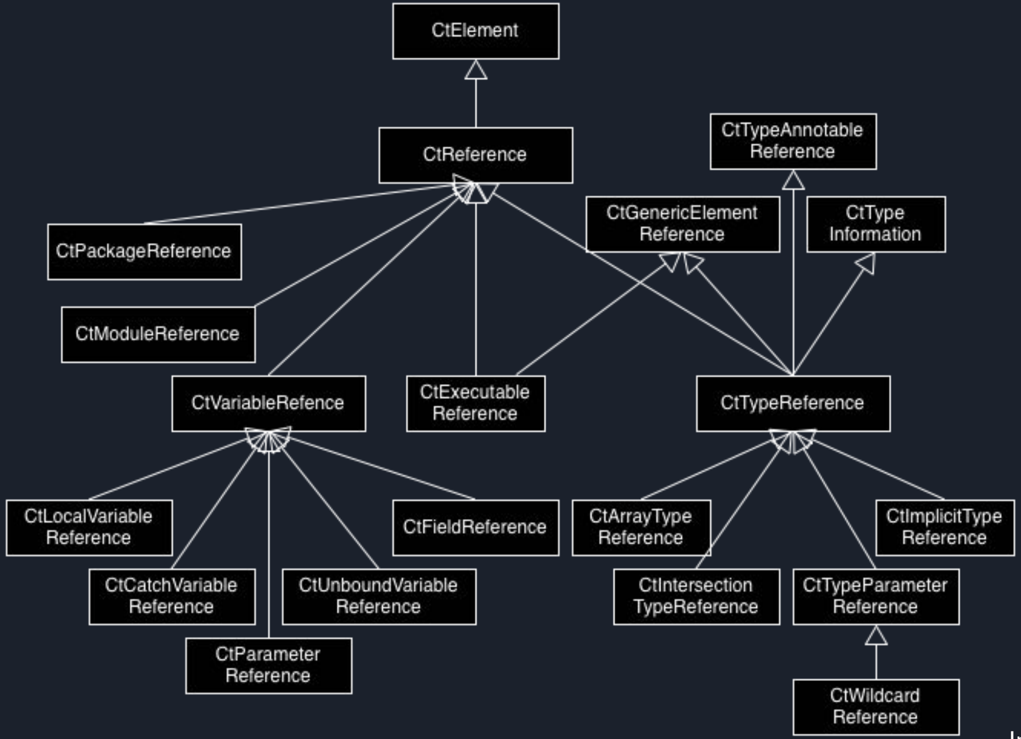


Image from: http://spoon.gforge.inria.fr/

Architectural Requirements

The program should be able to create a dependency graph using program files.

- The program needs to retrieve the files
- The program needs to parse the files
- The program needs to analyse the structure
- The program needs to produce a graph

Architectural Requirements

The program should be able to create a graph from Java source files using git.

More detailed requirements according to the documentation:

- The output graph should be the same as that of Arcan
- The program should parse Java source files in such a way so it can find the different Java classes and packages.
- The program needs to be able to use Git.
- The program should provide a command line interface.

Architectural Requirements

The program should be able to create a graph from Java source files using git.

More detailed requirements according to the documentation:

- The output graph should be the same as that of Arcan[1]
  - ✔ Apache Tinkerpop[2]
- The program should parse Java source files in such a way so it can find the different Java classes and packages.
  - ✔ Spoon[3]
- The program needs to be able to use Git.
  - ✔ Jgit[4]
- The program should provide a Command line interface
  - ✔ Apache commons cli[5]

[1] https://essere.disco.unimib.it/wiki/arcan/

[2] https://tinkerpop.apache.org/

[3] http://spoon.gforge.inria.fr/

[4] https://www.eclipse.org/jgit/

[5] https://commons.apache.org/proper/commons-cli/

Additional Requirements

- The output graph should be in graphml format
  - ✅
- The resulting graph should be similar to Structure101 in content
  - ❌

Technologies

What does a dependency graph creator engine need to do?

- Retrieve code
- Parse the code
- Analyse the structure
- Make the graph

Technologies

According to the documentation:

- Retrieve code
  - Jgit
- Parse the code
  - Spoon
- Analyse the structure
- Make the graph
  - Apache Tinkerpop

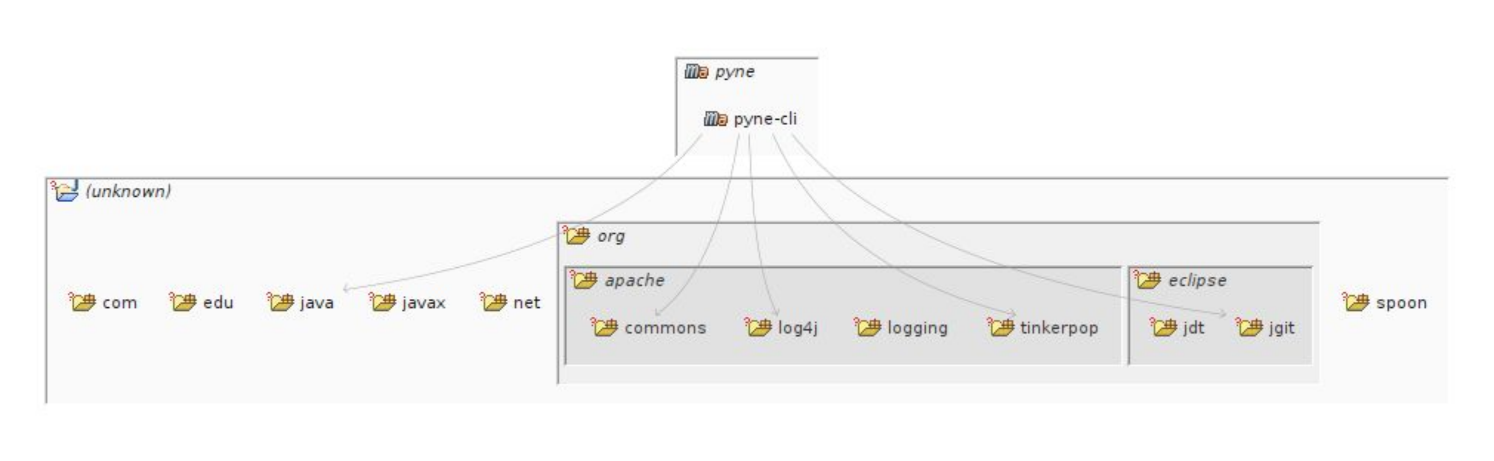
Technologies pyne-cli

According to the documentation:

- Apache Tinkerpop
- Spoon
- Jgit

According to structure101:

- Apache Commons CLI
- log4j



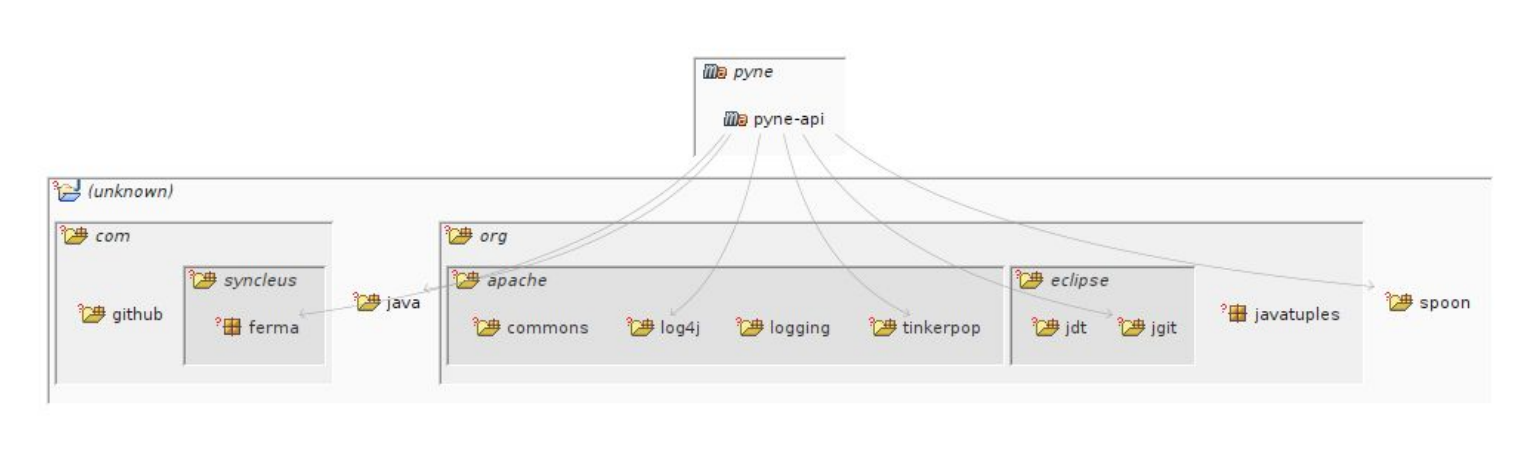
Technologies pyne-api

According to the documentation:

- Apache Tinkerpop
- Spoon
- Jgit

According to structure101:

- Log4j
- Ferma



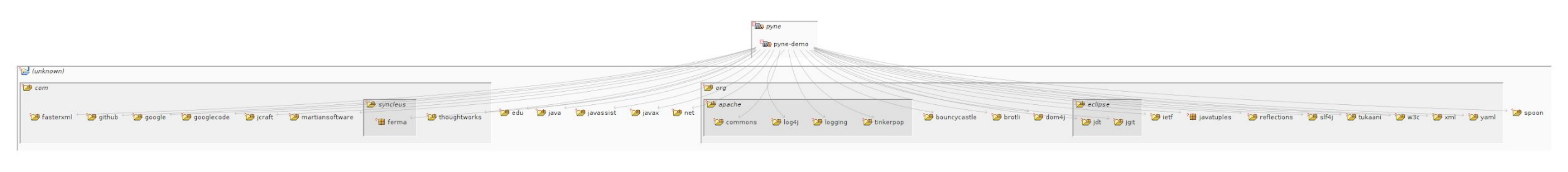
Technologies pyne-demo

According to the documentation:

- Apache Tinkerpop
- Spoon
- Jgit

According to structure101:

- A lot more



Any questions?