

Dynamic and Static Analysis of Python Software with Kieker

Daphné Larrivain, Shinhyung Yang, Wilhelm Hasselbring
4. November 2025



1. Introduction

2. General Pipeline

a. Overview

b. Static Analysis

c. Dynamic Analysis

3. Toolbox

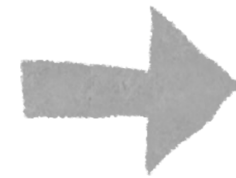
4. Visualization

5. Evaluation

What is the Kieker Observability Framework ?



kieker

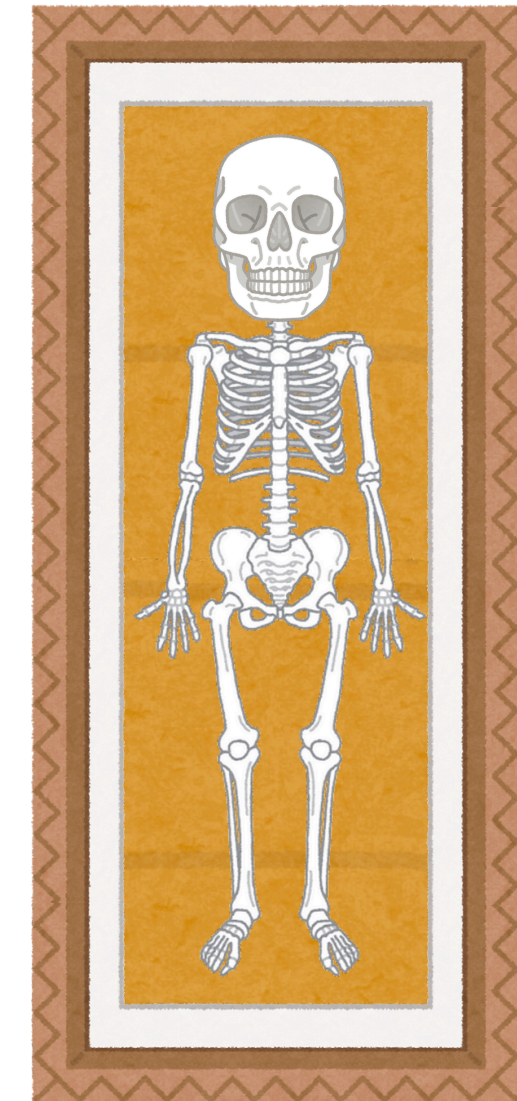


Target Software



Visualization

Static Analysis



Source code

Global Visualization

Dynamic Analysis



Instrumented Software



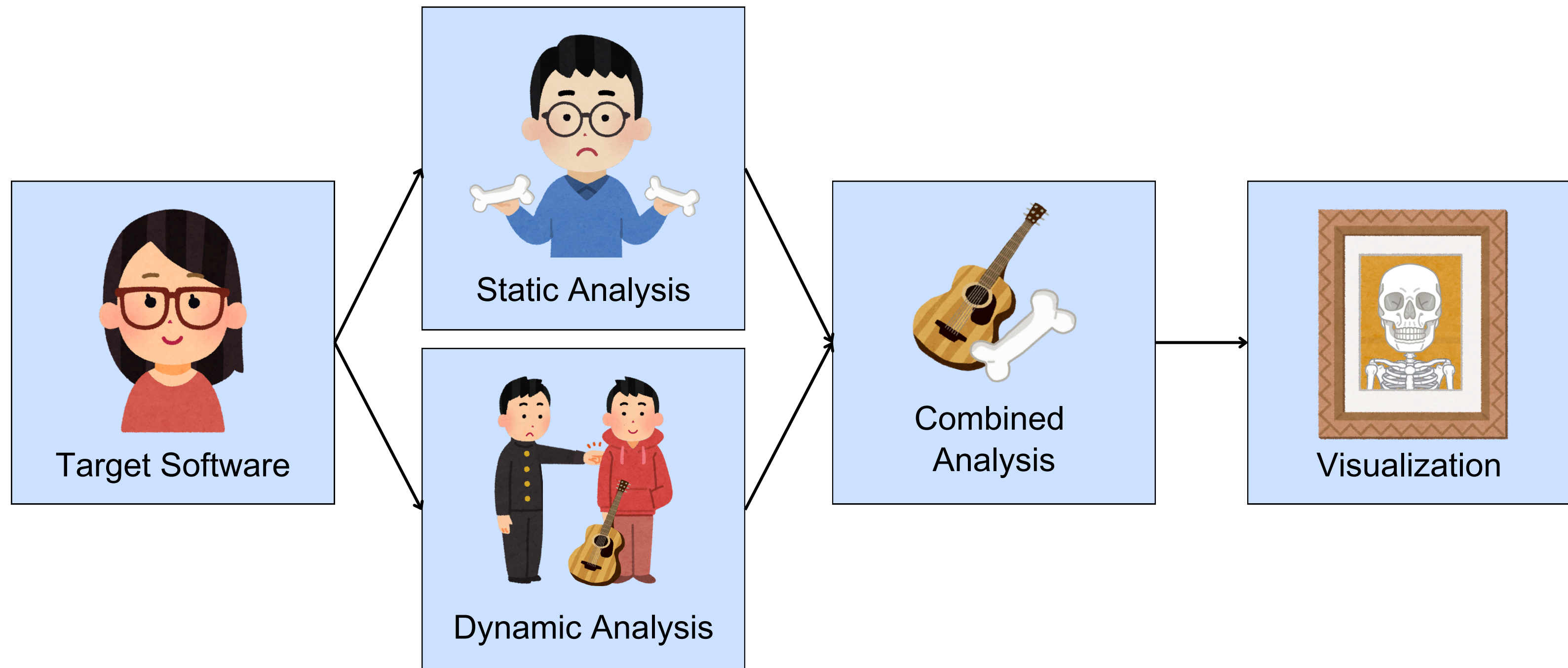
Focused
Visualization

Goals

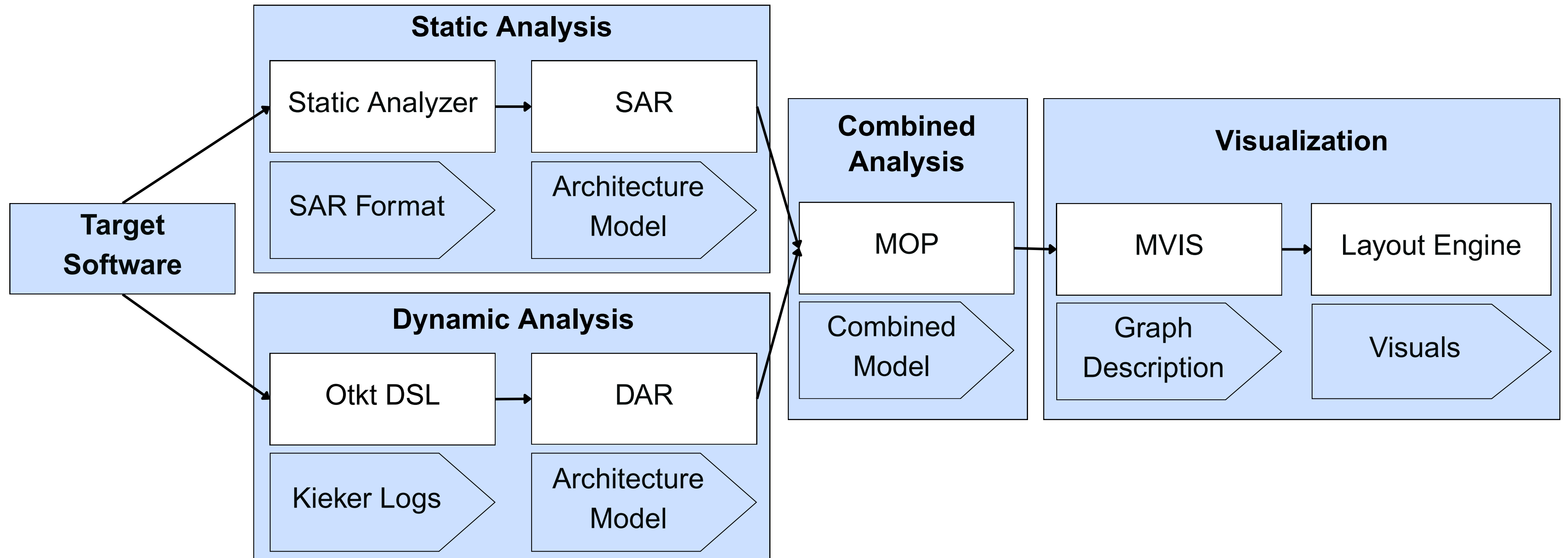
Expand the current toolbox to support
Python software analysis



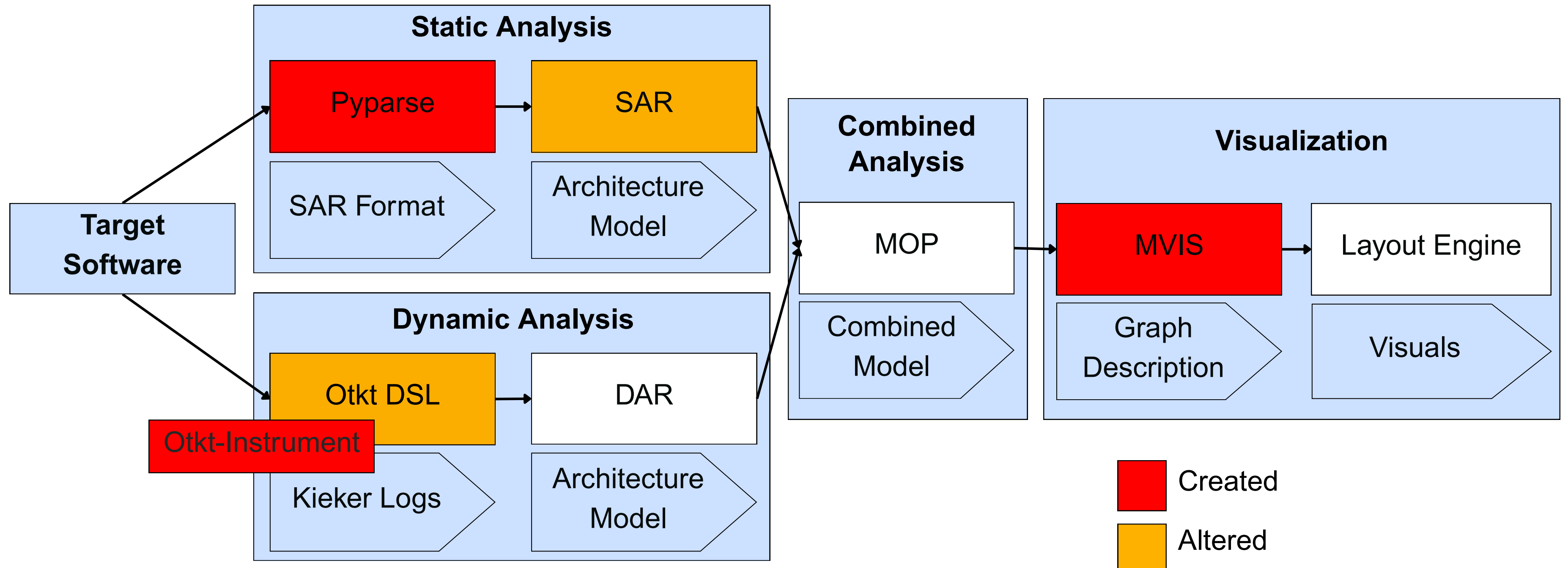
Simplified Overview



Tool Breakdown



Tool Breakdown



Otkt-Instrument

Applies Otkt DSL to instrument Python code with heuristics:

- Build files: ✗
- Break internal file structure: ✗



```
from otkt.tools.instrument import instrument
```



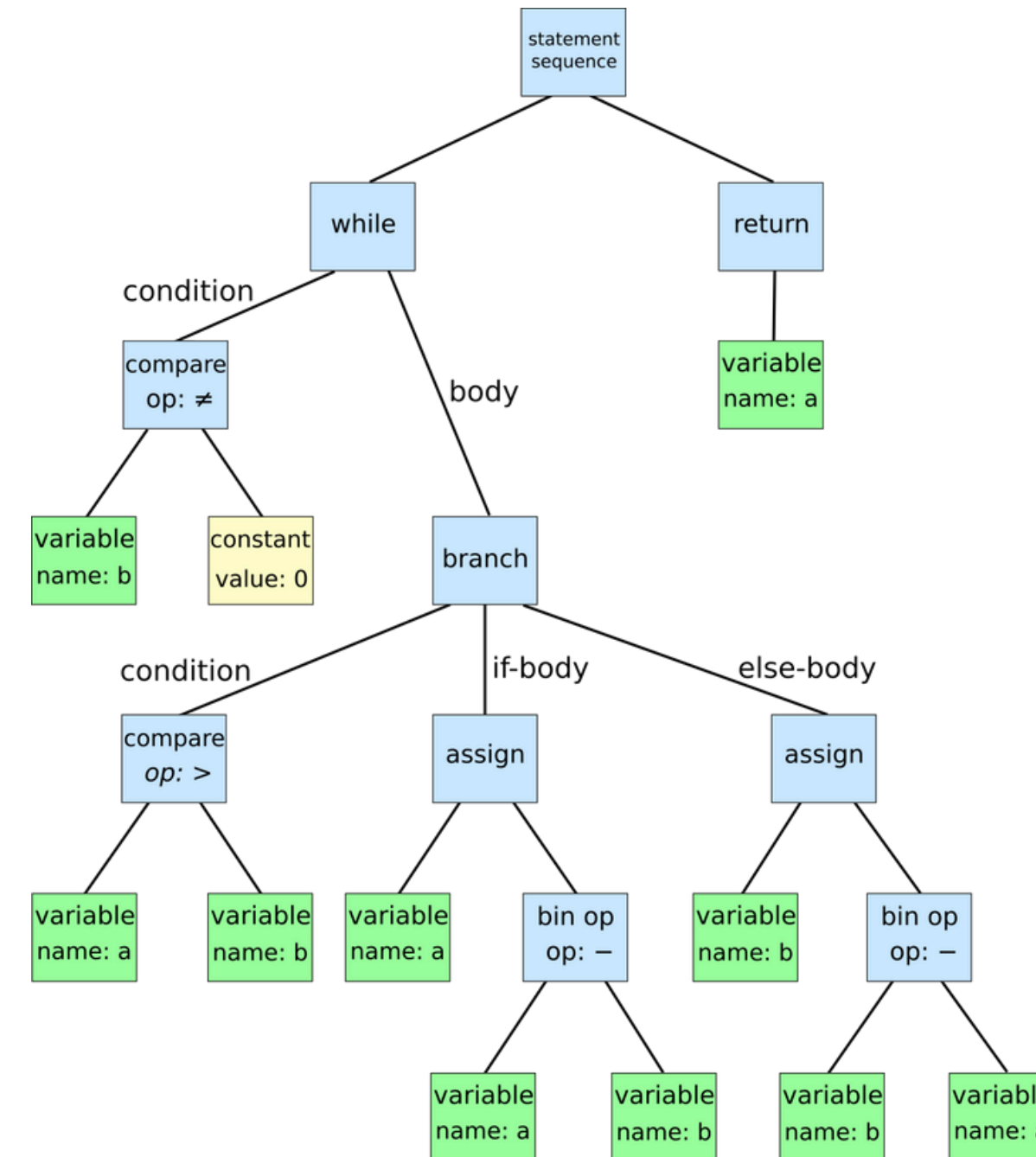
```
@instrument  
def hello_world():  
    print("Hello world!")
```


Pyparse

Static code analyzer:

- AST matcher.
- Compatible with SAR.

Example of a
generic AST



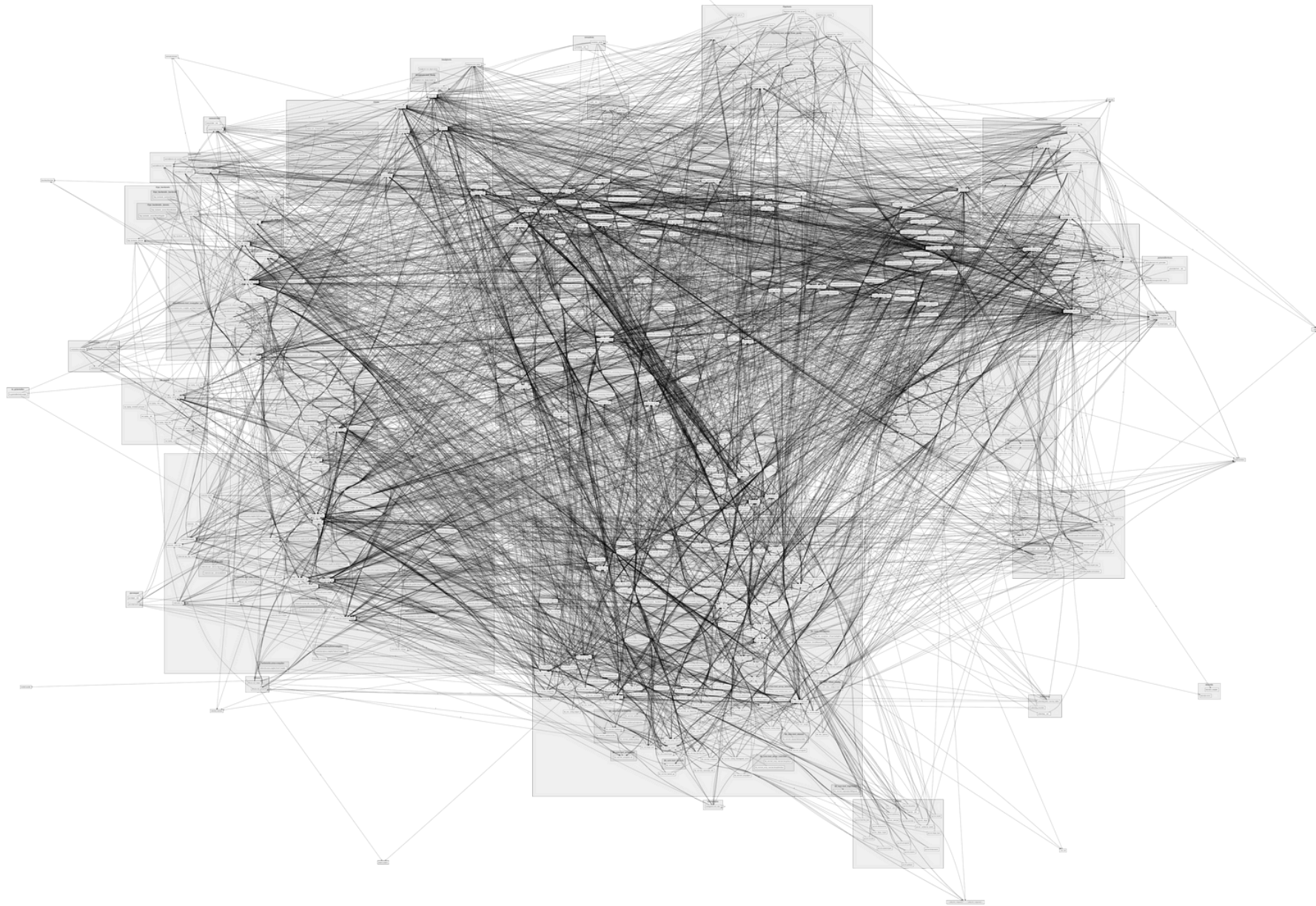
GGVIS

Lay out a graph in a clear and timely manner:

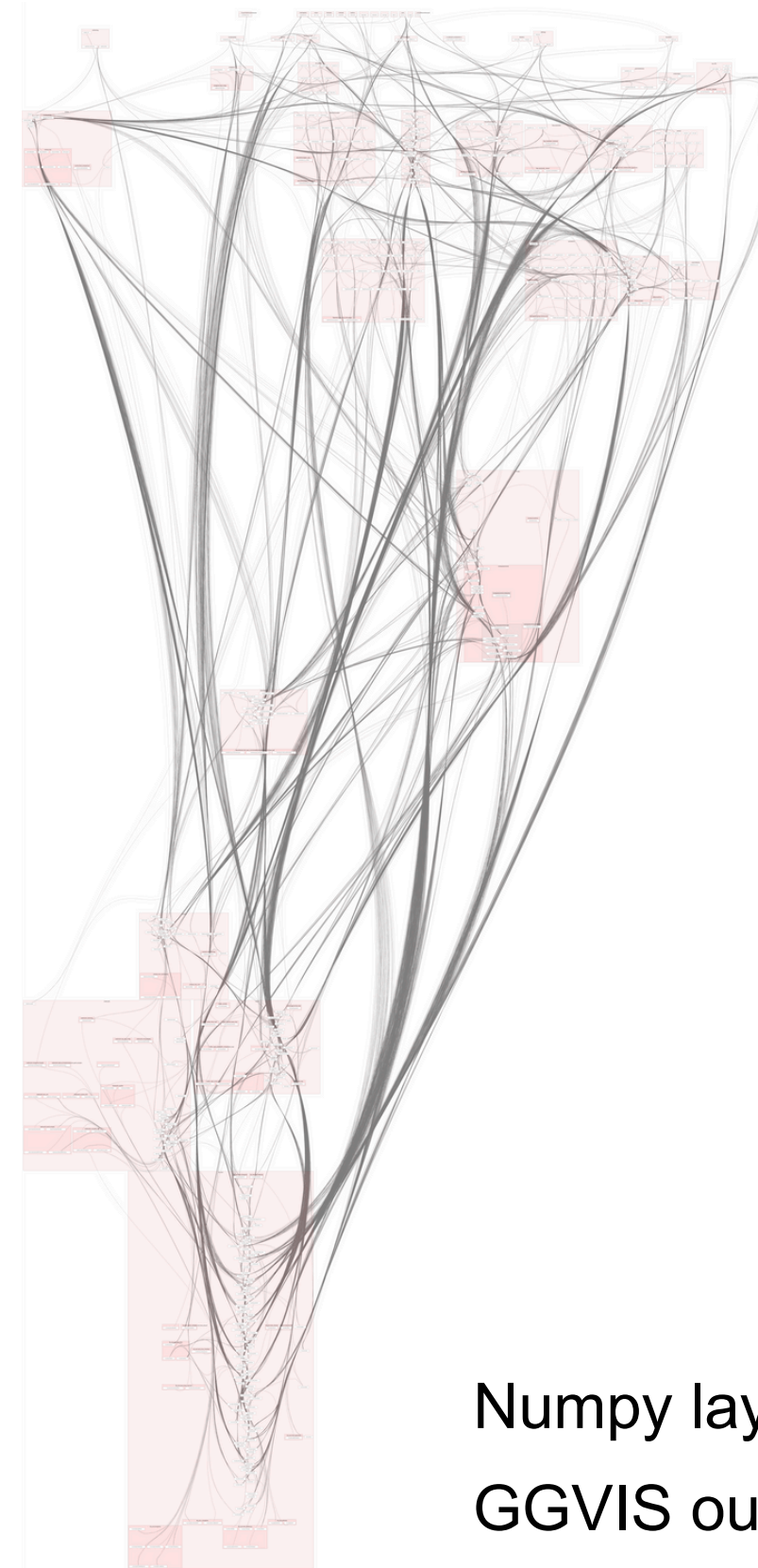
- Goal: nested output.
- Uses the Tulip Visualization Framework.
- Much faster.



Graphviz vs Tulip

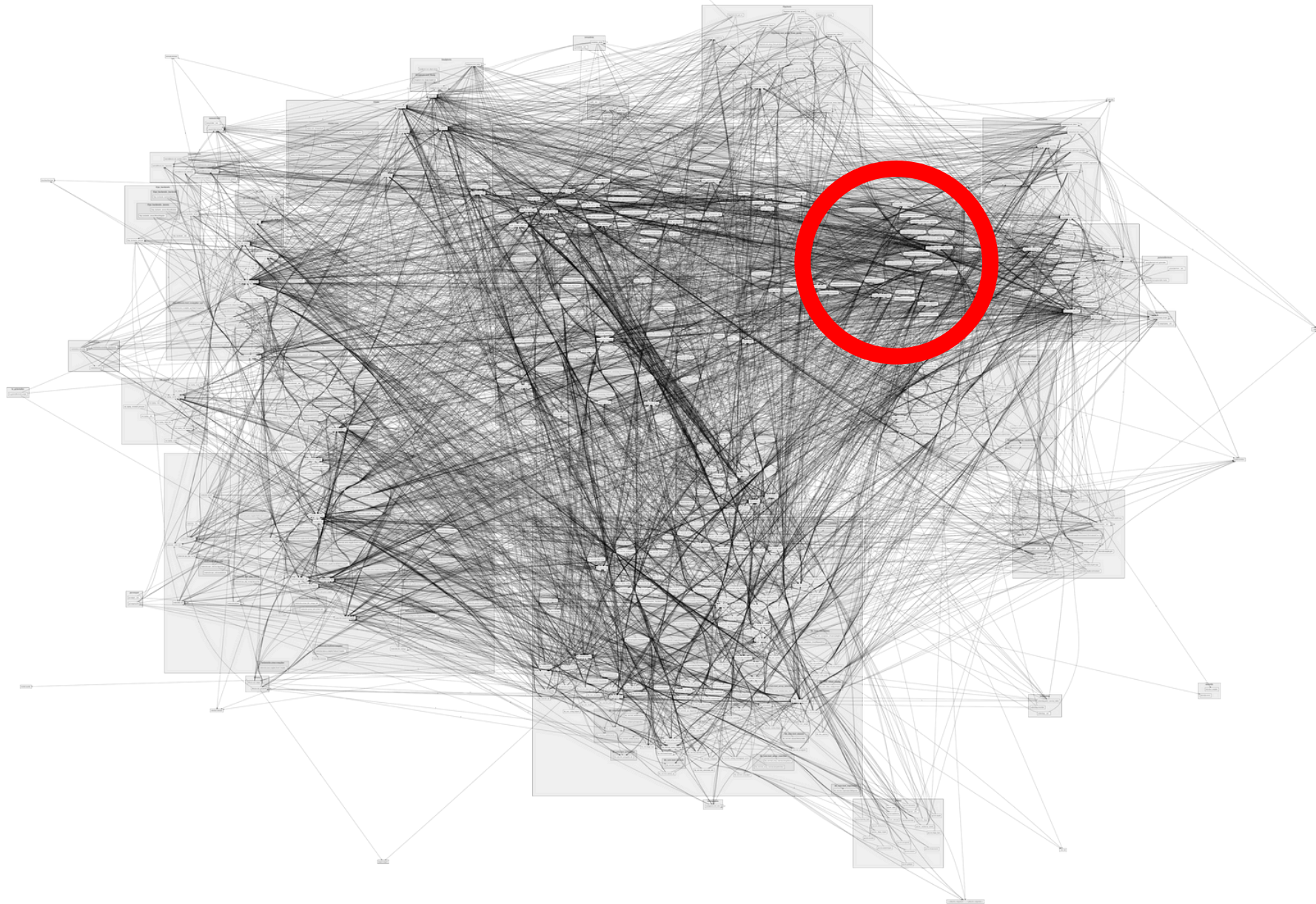


Numpy layout, Graphviz output

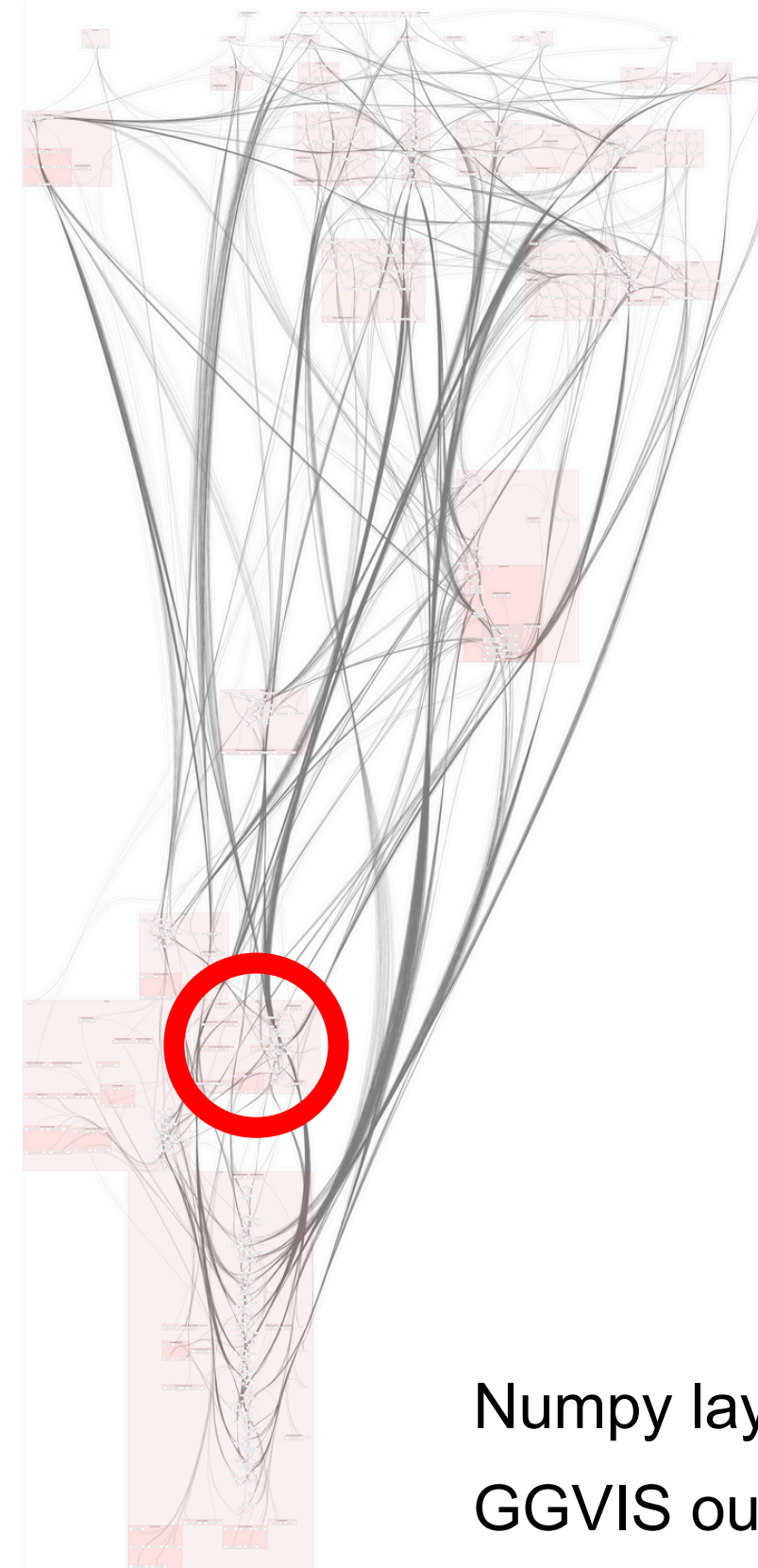


Numpy layout,
GGVIS output

Graphviz vs Tulip



Numpy layout, grapviz output



Numpy layout,
GGVIS output

Example: the
omlib package

Numpy layout,
Graphviz output

Omlib

Parent package name



omlib._iotools

omlib._polynomial_impl

omlib._index_tricks_impl

omlib._arrayterator_impl

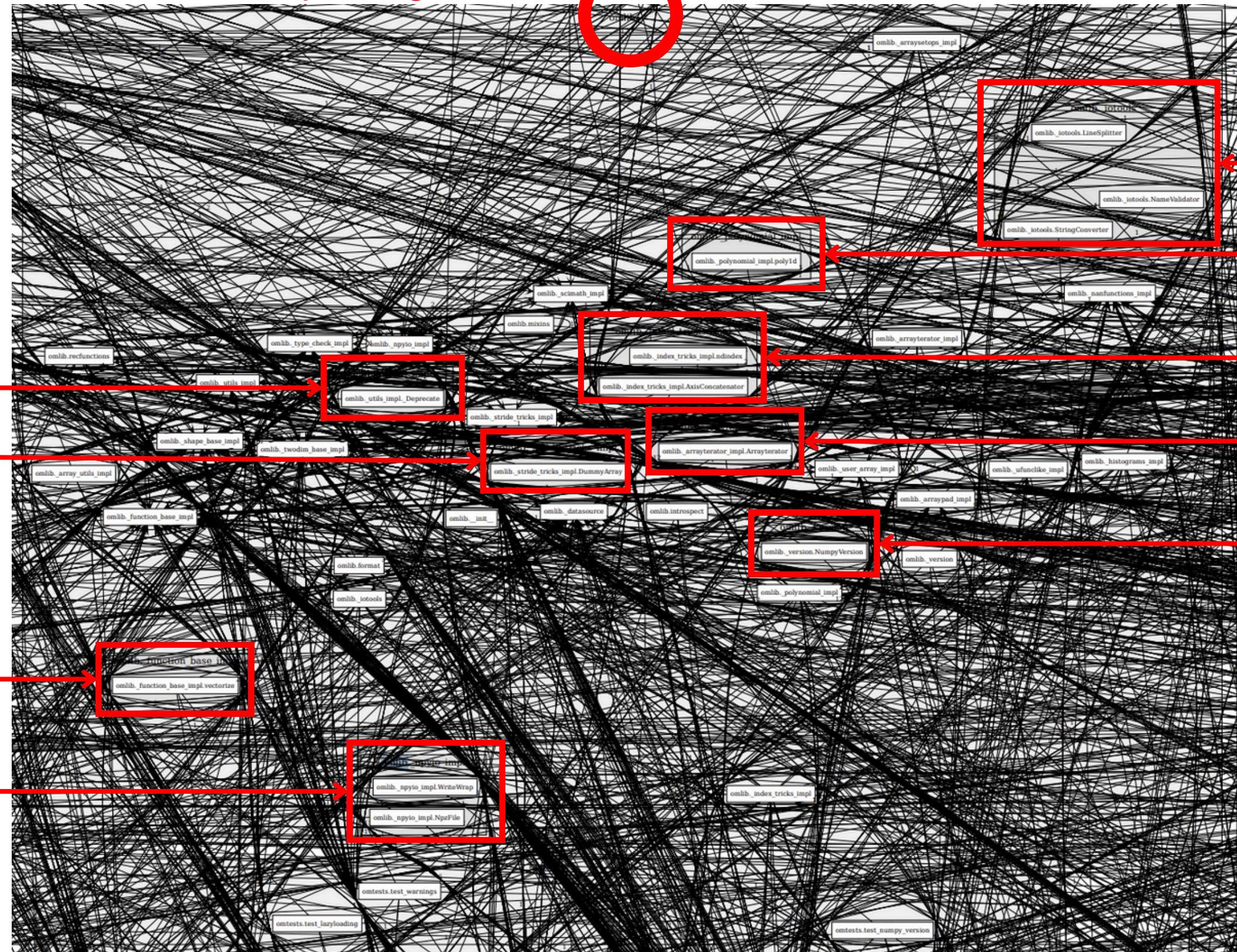
omlib._version

omlib._utils_impl

omlib._stride_tricks_impl

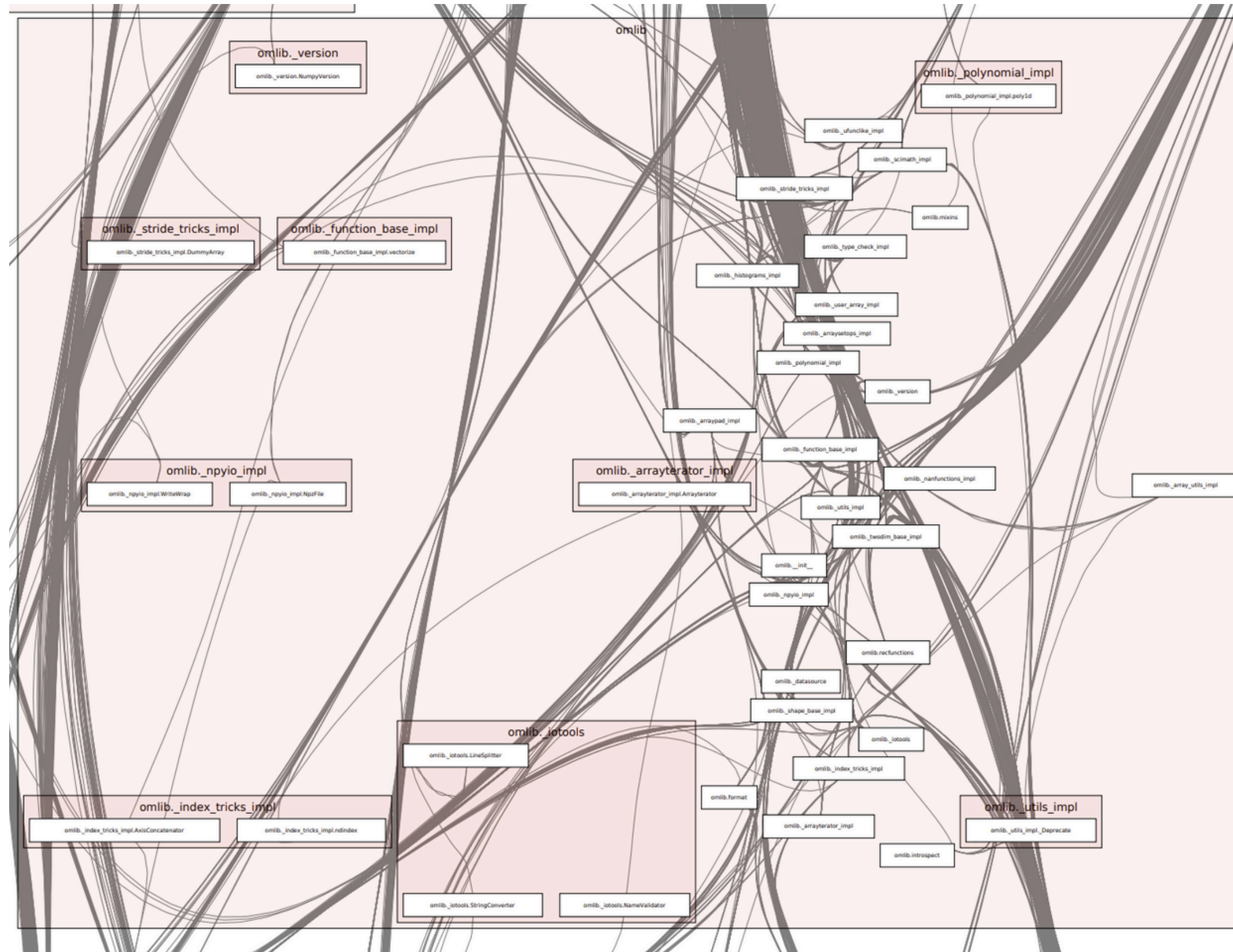
omlib._function_base_impl

omlib._npysio_impl



Numpy layout,
Graphviz output

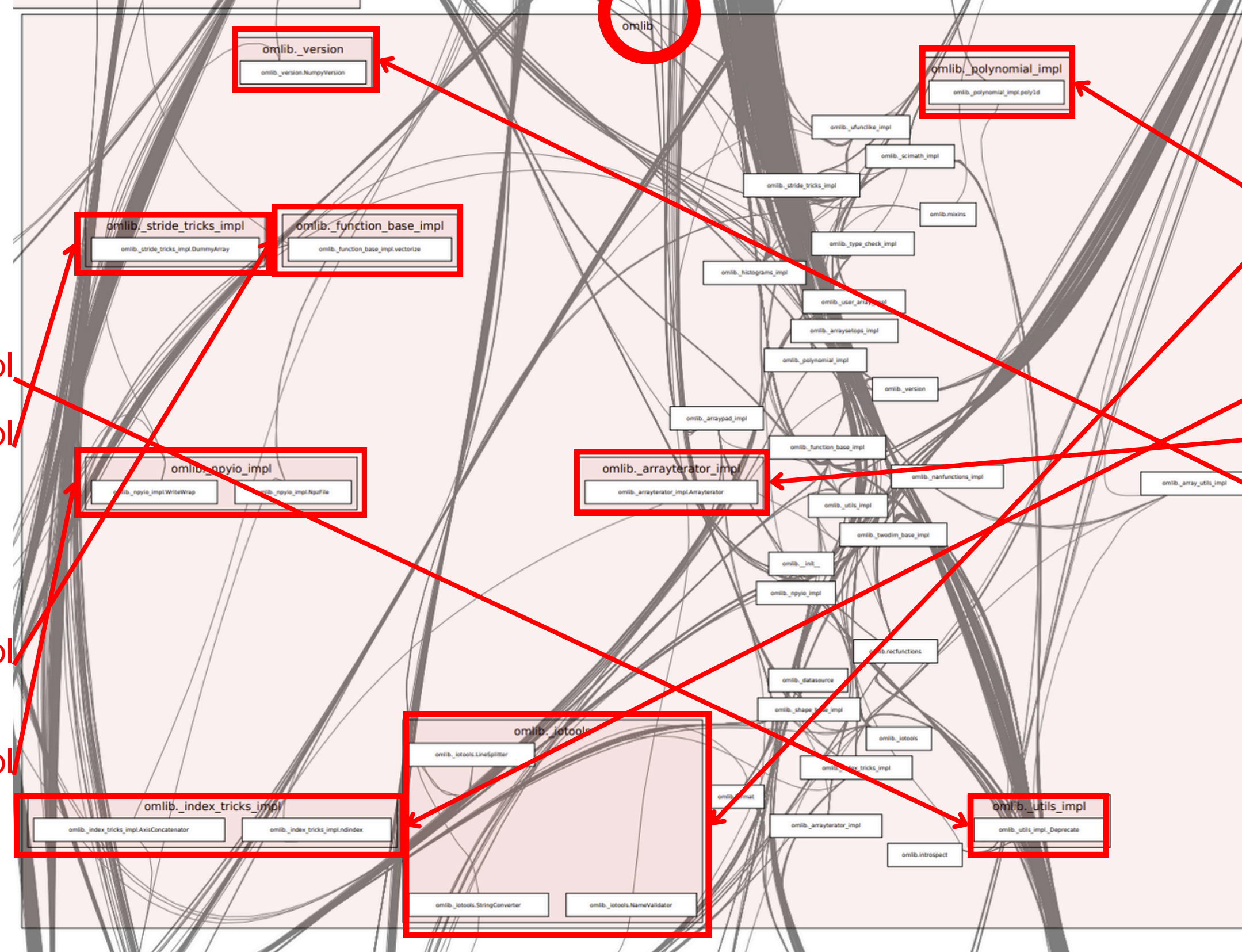
Omlib



Numpy layout,
GGVIS output

Omlib

Parent package name



omlib._iotools

omlib._polynomial_impl

omlib._index_tricks_impl

omlib._arrayiterator_impl

omlib._version

Numpy layout,
GGVIS output

General Pipeline Evaluation

	Anytree	UXsim	Pillow	Matplotlib	Numpy	Scipy
Pyparse	1s	2s	10s	3m53s	2m57s	7m15s
DAR	2s	4s	1s	2s		0m2s
SAR	3s	6s	10s	7m26s	10m44s	21m58s
MOP	3s	6s	5s	26m13s		28s
MVIS	2s	3s	2s	1m10s	19m22s	3s
GGVIS	1s	1s	1s	4m28s	34s	1s
Total	12s	22s	29s	43m12s	33m37s	29m47s

Visualization Evaluation

	UXsim	Anytree	Numpy
Fdp	3s	2s	1m50s
Dot	1m9s**	18s	∞
Fdp grouped	6s	2s	16m36s
Dot grouped	∞	∞	∞
GGVIS	1s	1s	34s
Total	2m22s	0m23s	18m26s

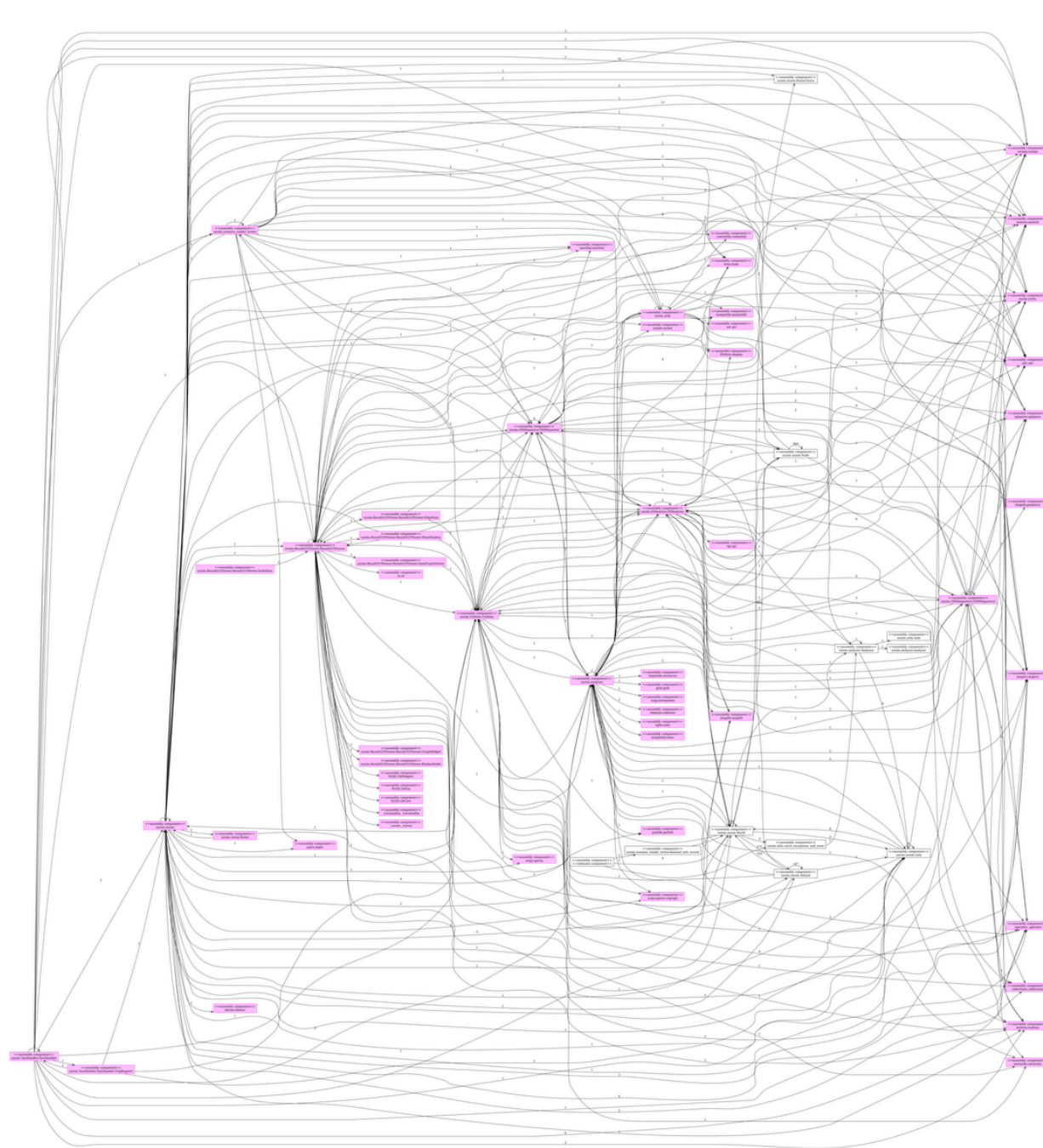
** Where dot starts to fail.

∞ Runs forever.

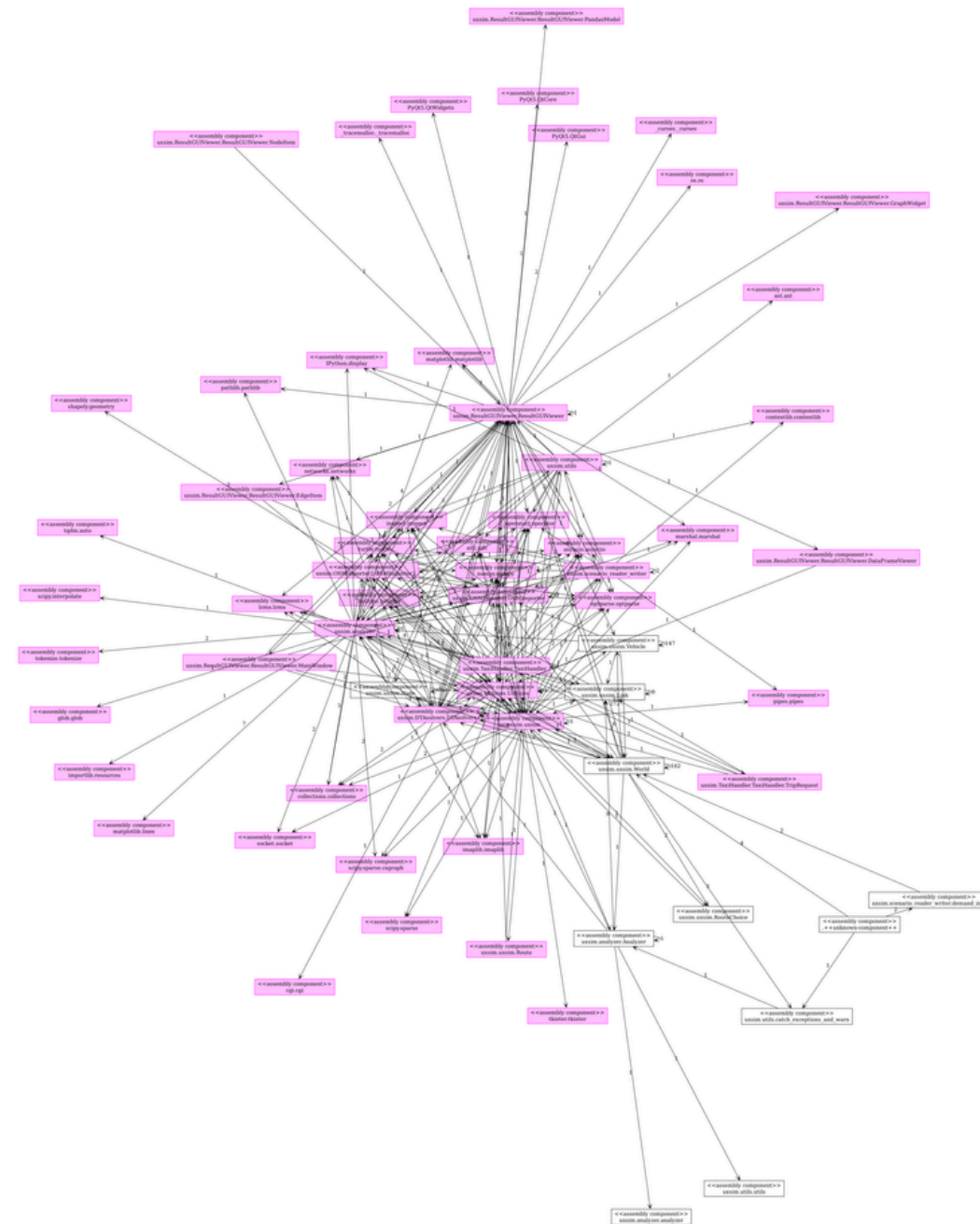
Thank you
for your attention



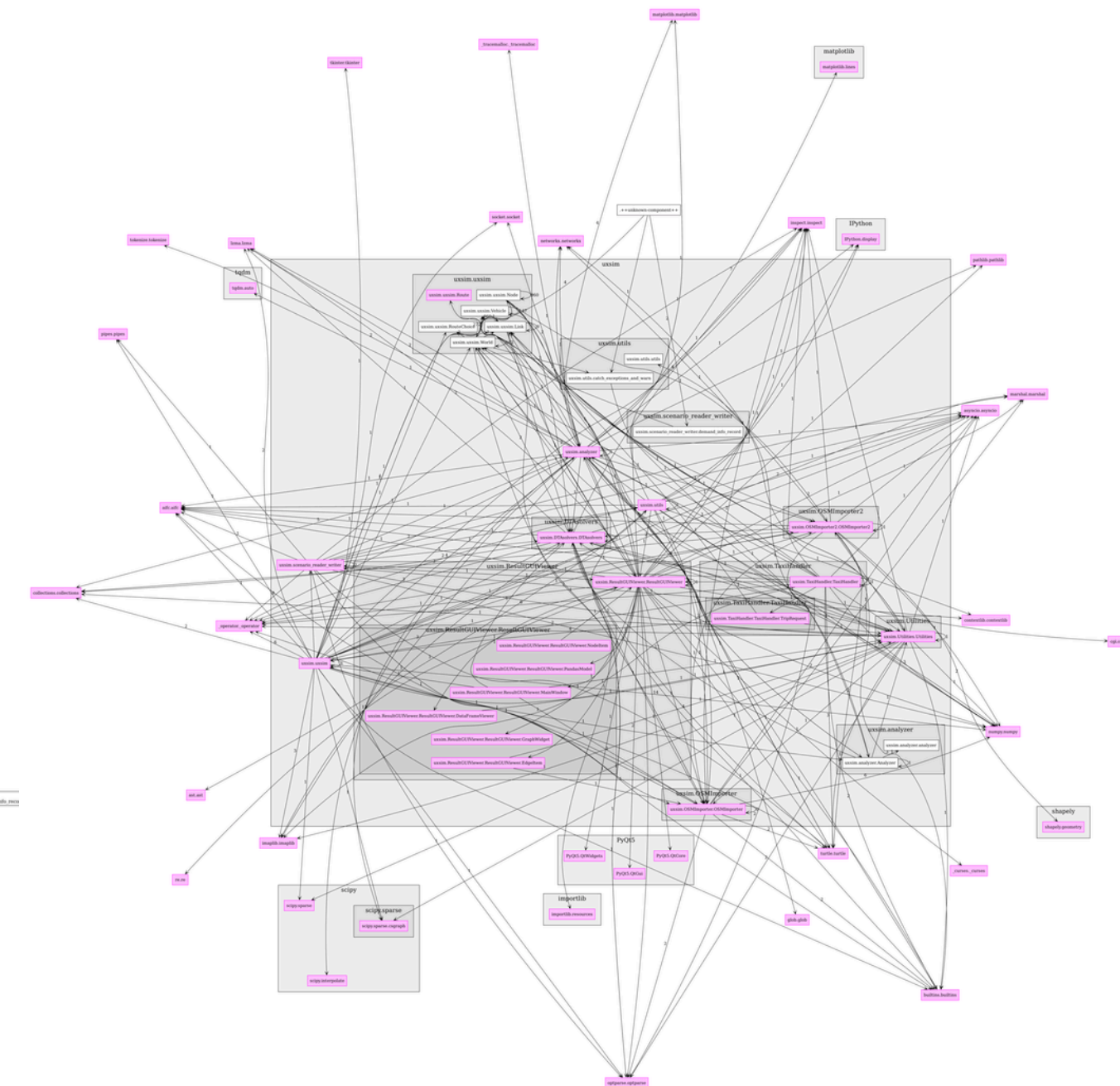
Graphviz: Fine for smaller apps



Uxsim, Graphviz dot flat



Uxsim, Graphviz fdp flat



Uxsim, Graphviz fdp grouped