



# **Pyrrha: Navigate easily into your system binaries**

**How to build a firmware mapper & extend it**

---

**Eloïse Brocas**

**ESE, Paris — 25 mai 2024**

## Security R&D Engineer @ Quarkslab

- ▶ Vulnerability research
- ▶ Reverse-engineering
- ▶ Tooling

X @\_cryptocorn\_

## Automated Analysis Team

Tools		
Dynamic Analysis	 <b>QBDI</b>	dynamic binary instrumentation framework
	 <b>Qtracer</b>	dynamic trace generator and analysis
Symbolic Execution	 <b>Triton</b>	symbolic execution framework
	 <b>TritonDSE</b>	DSE and exploration library ( <i>whitebox fuzzing</i> )
Fuzzing	 <b>PASTIS</b>	collaborative/distributed fuzzing
	 <b>HF/QBDI</b>	Honggfuzz backed by QBDI
Firmware Analysis	 <b>Pandora</b>	whole firmware analysis engine
	 <b>Pyrrha</b>	firmware cartography
	 <b>QSig</b>	firmware 1-Day matching engine ( <i>discontinued</i> )
Diffing	 <b>python-bindiff</b>	python library wrapping Bindiff
	 <b>QBinDiff</b>	Binary Differ based on machine learning algorithm
Static Analysis	 <b>python-binexport</b>	python API to manipulate Binexport files
	 <b>Quokka</b>	IDA plugin and python API to manipulate IDA disassembly
Deobfuscation	 <b>Qsynthesis</b>	synthesis based deobfuscator ( <i>targeting MBAs</i> )

## Security R&D Engineer @ Quarkslab

- ▶ Vulnerability research
- ▶ Reverse-engineering
- ▶ Tooling

X @\_cryptocorn\_

## Pass the Salt Organizer

- ▶ Lille, July 3-5, 2024
- ▶ [2024.pass-the-salt.org](http://2024.pass-the-salt.org)

## Automated Analysis Team

Tools		
Dynamic Analysis	 <b>QBDI</b>	dynamic binary instrumentation framework
	 <b>Qtracer</b>	dynamic trace generator and analysis
Symbolic Execution	 <b>Triton</b>	symbolic execution framework
	 <b>TritonDSE</b>	DSE and exploration library ( <i>whitebox fuzzing</i> )
Fuzzing	 <b>PASTIS</b>	collaborative/distributed fuzzing
	 <b>HF/QBDI</b>	Honggfuzz backed by QBDI
Firmware Analysis	 <b>Pandora</b>	whole firmware analysis engine
	 <b>Pyrrha</b>	firmware cartography
	 <b>QSig</b>	firmware 1-Day matching engine ( <i>discontinued</i> )
Diffing	 <b>python-bindiff</b>	python library wrapping Bindiff
	 <b>QBinDiff</b>	Binary Differ based on machine learning algorithm
Static Analysis	 <b>python-binexport</b>	python API to manipulate Binexport files
	 <b>Quokka</b>	IDA plugin and python API to manipulate IDA disassembly
Deobfuscation	 <b>Qsynthesis</b>	synthesis based deobfuscator ( <i>targeting MBAs</i> )

# Definition

## Embedded into IoT objects

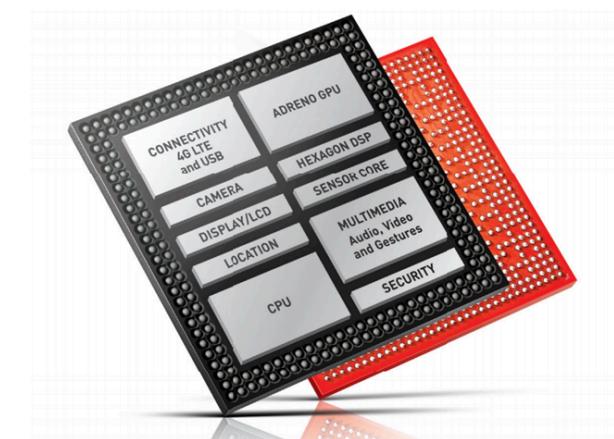
- ▶ routers
- ▶ smartphones
- ▶ automotive

## Structured firmwares

- ▶ filesystem
- ▶ thousands of files

## Complete OSes (*Android, Linux, OpenWRT, ...*)

- ▶ base operating system
- ▶ libraries, SDK, etc.
- ▶ vendor specific code



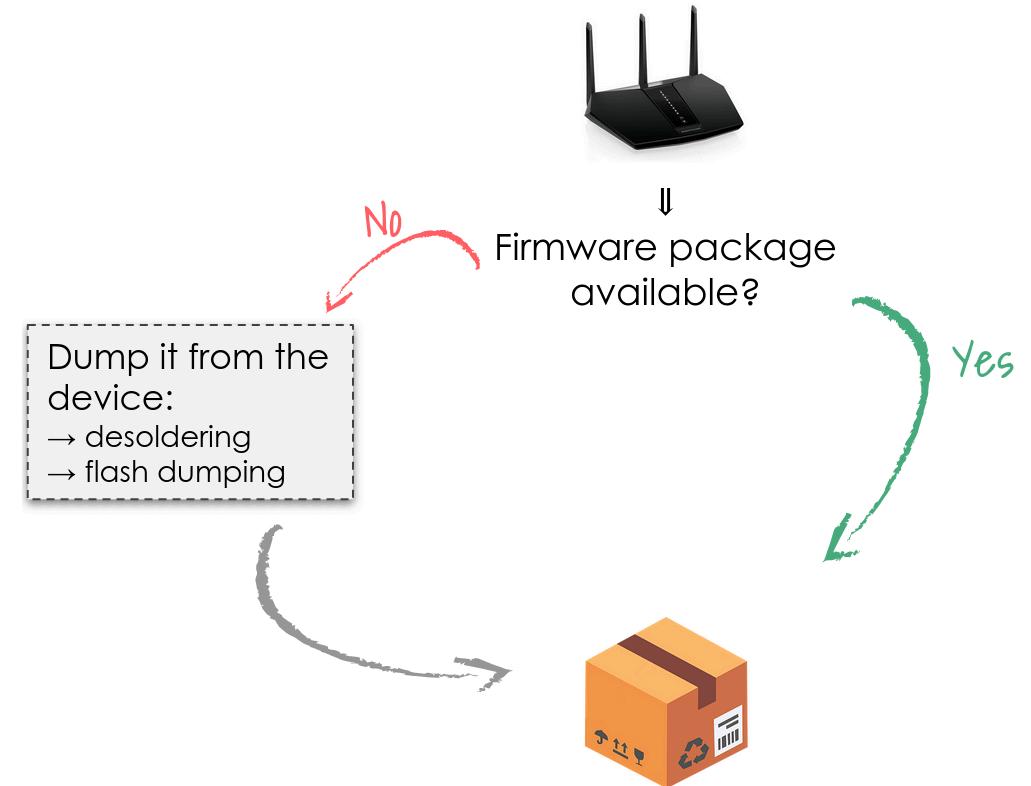
# Firmware Analysis use case

- ▶ Vulnerability Research
- ▶ Artifacts left (*private keys, build files, symbols*)
- ▶ Compliance:
  - ▶ Checking patches are applied
  - ▶ Checking firmware signatures (*trust management*)
- ▶ SBOM (Software Bill-of-Materials)

# Analysis workflow

## 1. Acquisition

- ▶ could require hardware reverse



# Analysis workflow

## 1. Acquisition

## 2. Extraction

- ▶ Custom *proprietary formats*
- ▶ Encrypted firmwares

### **Binwalk**

- ▶ developed by ReFirmLabs
- ▶ <https://github.com/ReFirmLabs/binwalk>

### **Unblob**

- ▶ developed by OneKey
- ▶ <https://unblob.org>

# Analysis workflow

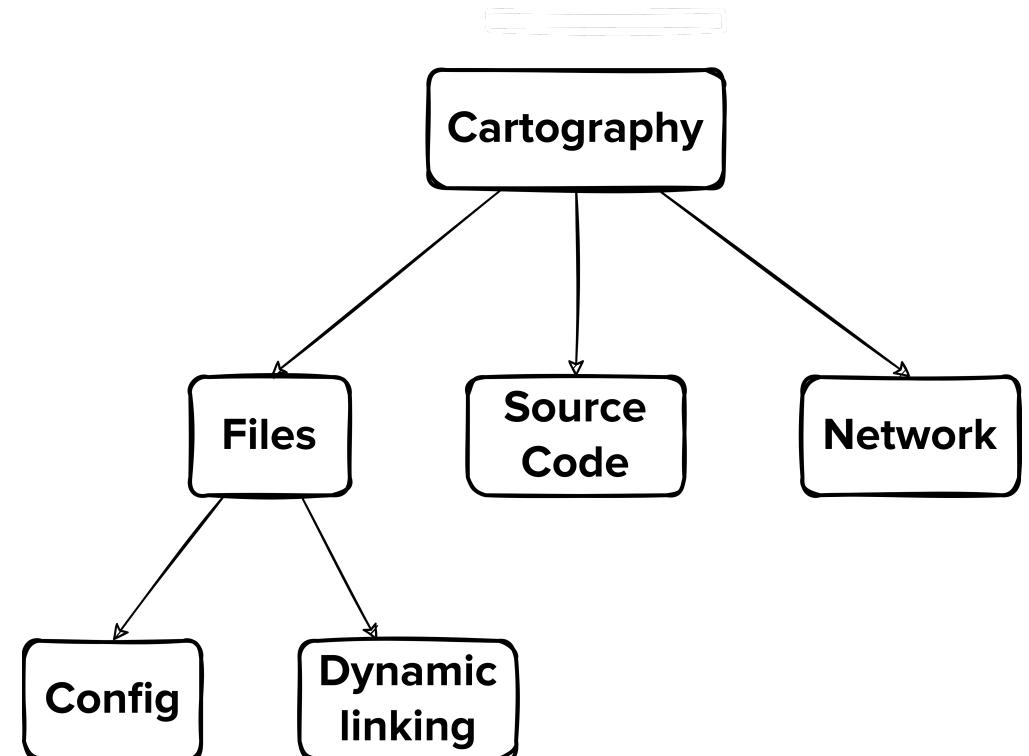
Q

## 1. Acquisition

## 2. Extraction

## 3. Cartography

- ▶ Make correlations
- ▶ Understand interactions between components
- ▶ Identify points of interest



# Analysis workflow

## 1. Acquisition

## 2. Extraction

## 3. Cartography

## 4. In-depth Analysis & Conclusions

The screenshot shows a blog post titled "Our Pwn2Own journey against time and randomness (part 2)" on the "Quarkslab's blog". The post was published on Tuesday, November 7, 2023, by Eloïse Brocas and Damien Cauquil, with contributions from Robin David and Benoît Forgette. It is categorized under "Vulnerability" and "Tags" for "vulnerability" and "2023". The sidebar includes links to the website, social media (Twitter, Mastodon, GitHub), and categories such as Android, Automotive, Blockchain, Challenge, Containers, Cryptography, Exploitation, Fuzzing, Hardware, Kernel Debugging, Life at Quarkslab, Math, Pentest, Program Analysis, Programming, Reverse-Engineering, Software, and Vulnerability.

**Our Pwn2Own journey against time and randomness (part 2)**

Tue 07 November 2023 By Eloïse Brocas, Damien Cauquil, Robin David, Benoît Forgette Category Vulnerability Tags vulnerability, 2023

Part 2 of a series about participation in the Pwn2Own Toronto 2023 contest. This blogpost is the second part of the series about our journey to the pwn2own Toronto 2022 contest. In Our Pwn2Own journey against time and randomness, part 1 we explained how we attacked the router from the WAN side and lost our battle against randomness and time just by a few seconds. Here we will describe two vulnerabilities that we found on the LAN side of the Netgear RAX30 router. Pwn2Own Toronto 2022 occurred on December 6-9, 2022, so why publish this now? Well, we reported the discovered vulnerabilities to the vendor and followed a coordinated vulnerability disclosure process that took a lot of time, and we decided not to rage-quit and just publish them, so after a lengthy coordination process we reached the agreed date for publication and here's the blog post.

**LAN vulnerabilities**

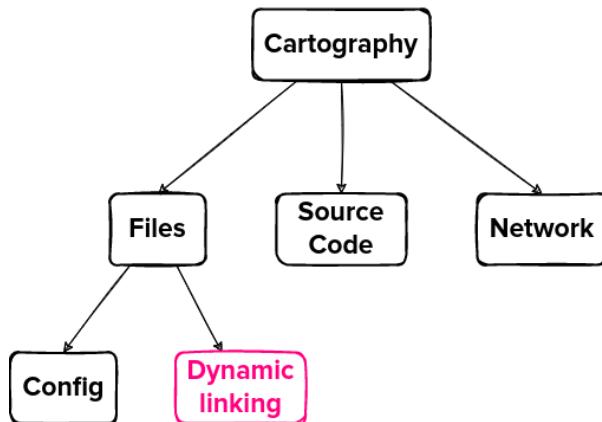
**Buffer Overflow inside soap server**

This vulnerability known as CVE-2023-27368 was not attributed to us, it seems that another competitor also found it. However, since we found this vulnerability during the competition, we will describe it nonetheless. On the LAN side, a server brought to our notice this program called [soap\\_serverd](#), it is a server that is exposed on all LAN interfaces

Quarkslab blogpost about [WAN vulnerabilities in Netgear RAX30 router](#).

# Pyrrha specifications

## Cartography tool



## How to display result?

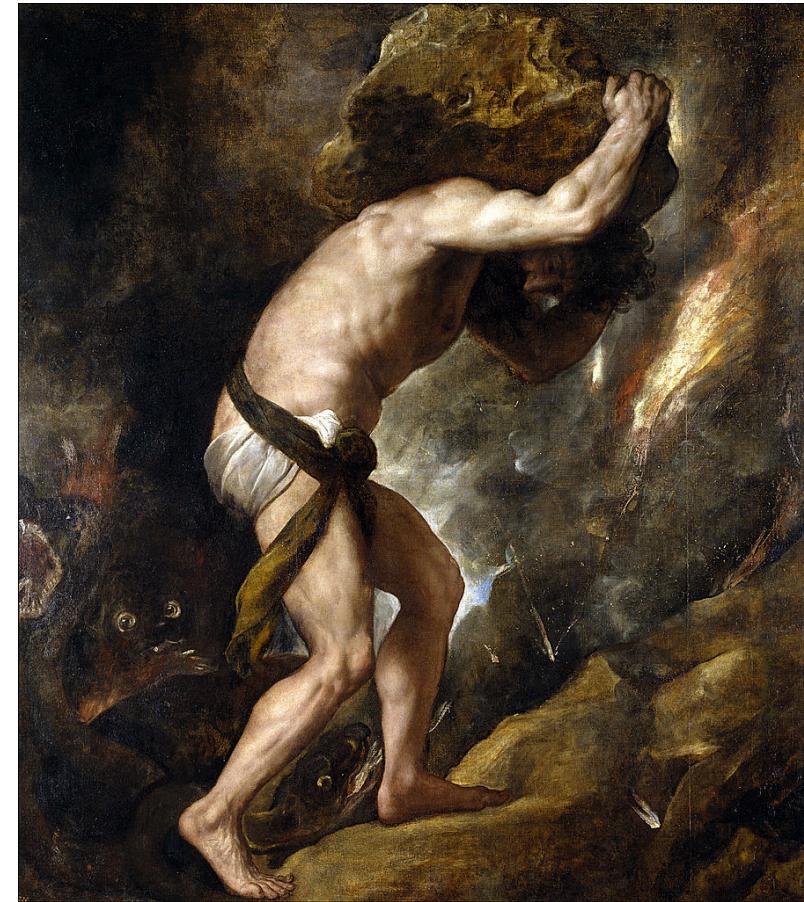
- ▶ JSON?
- ▶ console only?
- ▶ GUI?

# We also need visualization

What is the family link?



*Pyrrha*, J.-P. A. Tassaert. Source: [Musée du Louvre](#).



*Sisyphus*, Titian. Source: [Wikimedia](#).

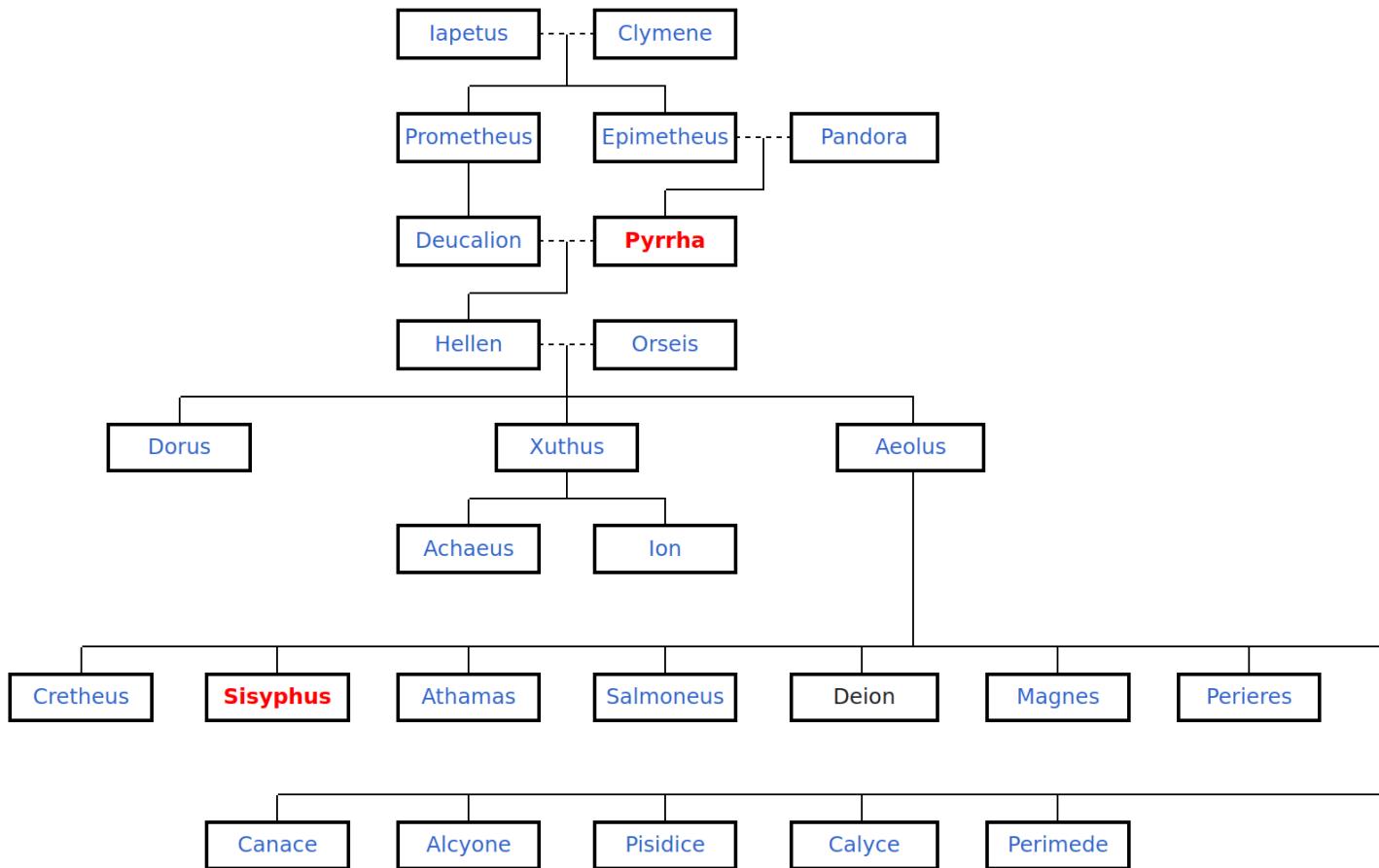
# Visualization is the key

«[1.7.2] And Prometheus had a son Deucalion. He reigning in the regions about Phthia, married Pyrrha, the daughter of Epimetheus and Pandora, the first woman fashioned by the gods. And when Zeus would destroy the men of the Bronze Age, Deucalion by the advice of Prometheus constructed a chest, and having stored it with provisions he embarked in it with Pyrrha. But Zeus by pouring heavy rain from heaven flooded the greater part of Greece, so that all men were destroyed, except a few who fled to the high mountains in the neighborhood. It was then that the mountains in Thessaly parted, and that all the world outside the Isthmus and Peloponnese was overwhelmed. But Deucalion, floating in the chest over the sea for nine days and as many nights, drifted to Parnassus, and there, when the rain ceased, he landed and sacrificed to Zeus, the god of Escape. And Zeus sent Hermes to him and allowed him to choose what he would, and he chose to get men. And at the bidding of Zeus he took up stones and threw them over his head, and the stones which Deucalion threw became men, and the stones which Pyrrha threw became women. Hence people were called metaphorically people (*laos*) from *laas*, “a stone.” And Deucalion had children by Pyrrha, first Hellen, whose father some say was Zeus, and second Amphictyon, who reigned over Attica after Cranaus; and third a daughter Protogenia, who became the mother of Aethlius by Zeus

[1.7.3] Hellen had Dorus, Xuthus, and Aeolus by a nymph Orseis. Those who were called Greeks he named Hellenes after himself, and divided the country among his sons. Xuthus received Peloponnese and begat Achaeus and Ion by Creusa, daughter of Erechtheus, and from Achaeus and Ion the Achaeans and Ionians derive their names. Dorus received the country over against Peloponnese and called the settlers Dorians after himself. Aeolus reigned over the regions about Thessaly and named the inhabitants Aeolians. He married Enarete, daughter of Deimachus, and begat seven sons, Cretheus, Sisyphus, Athamas, Salmoneus, Deion, Magnes, Perieres, and five daughters, Canace, Alcyone, Pisidice, Calyce, Perimede.»

*Apollodorus, 1.7.[2-3].*

# Visualization is the key

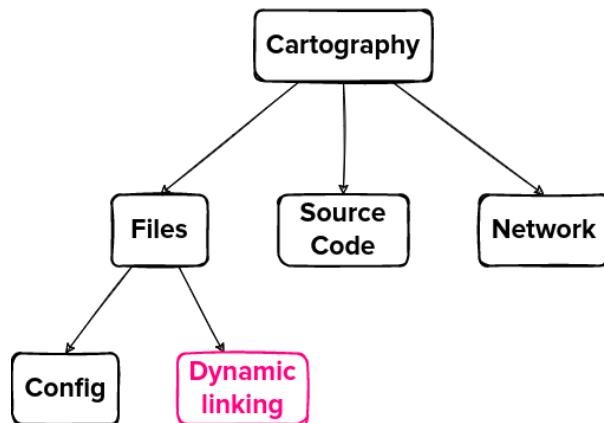


Genealogy of Hellenes. Source: [Wikipedia](#).

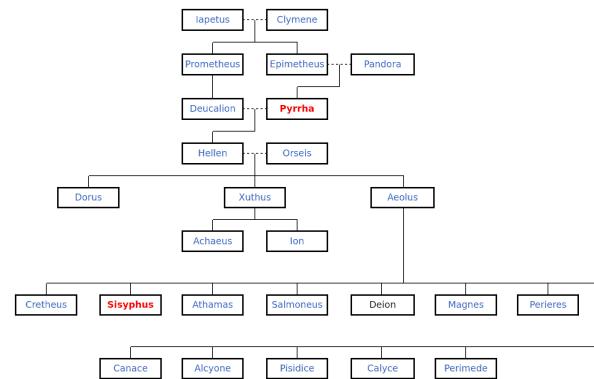
# Pyrrha specifications

Q

## Cartography tool



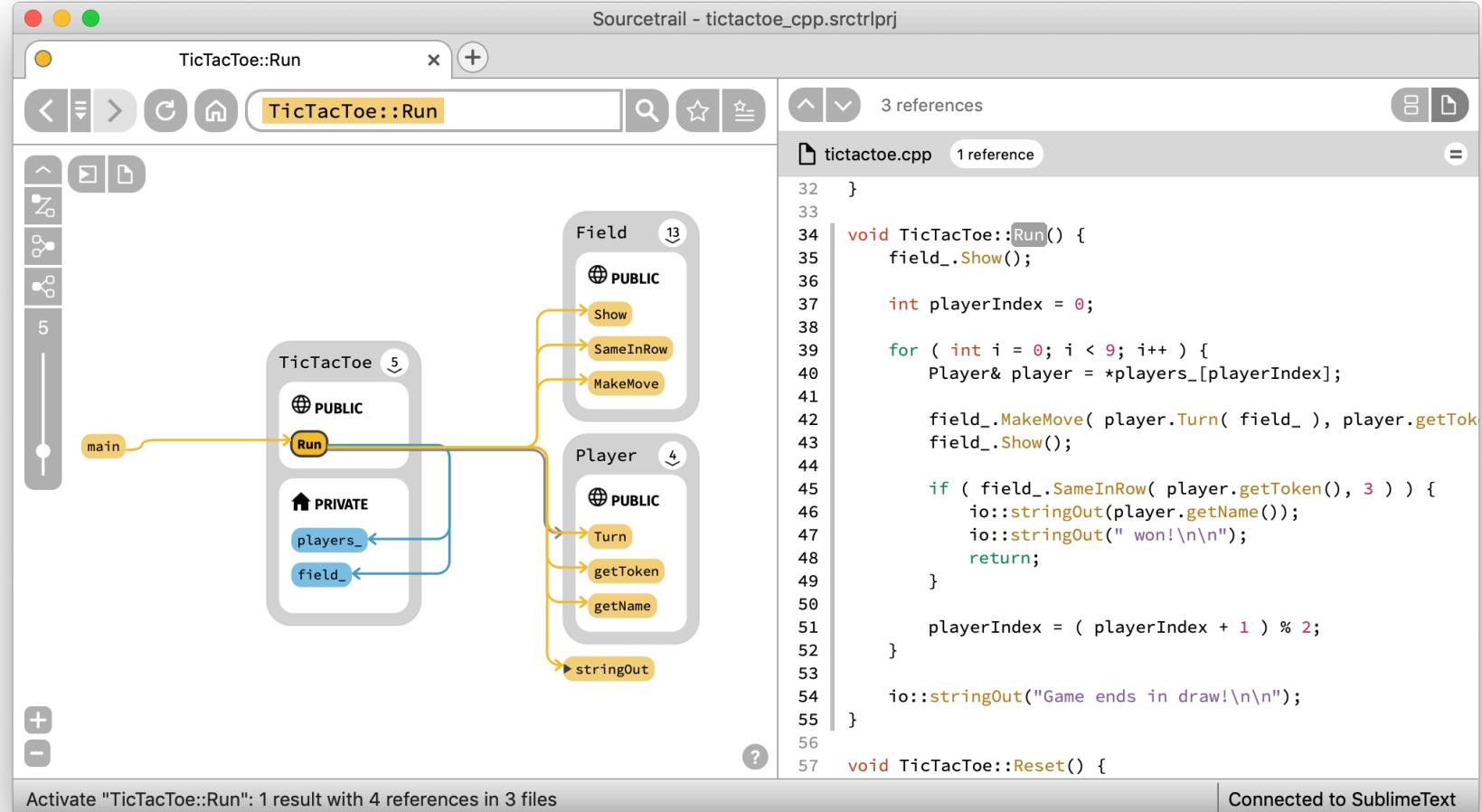
## Result visualization



## Reuse existing project?

# Let's try not to reinvent the wheel!

Q



Sourcetrail interface. Source: [Sourcetrail README](#).

# Filesystem as a language

**Binaries    Symlinks**

---

Class

TypeDef

**Exported functions    Exported symbols**

---

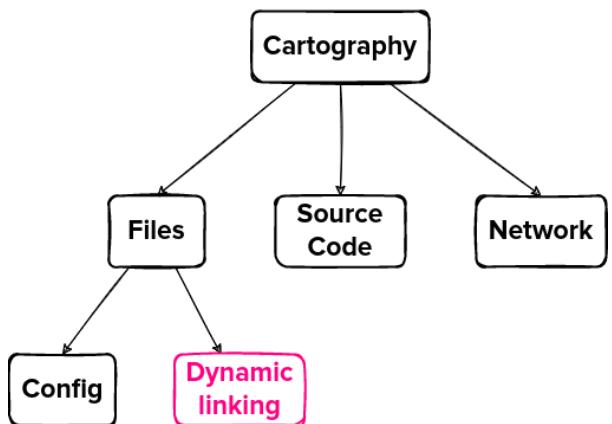
function

variable

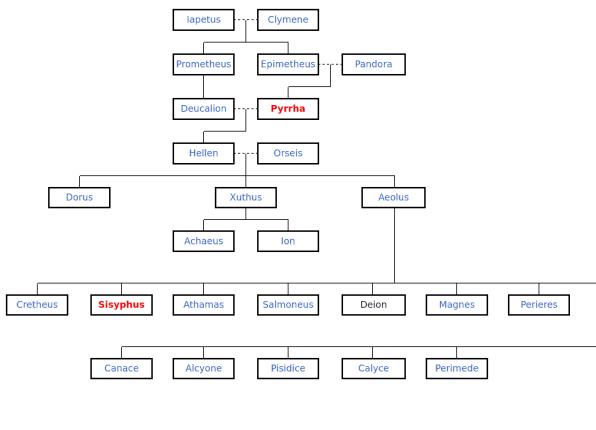
# Pyrrha specifications

Q

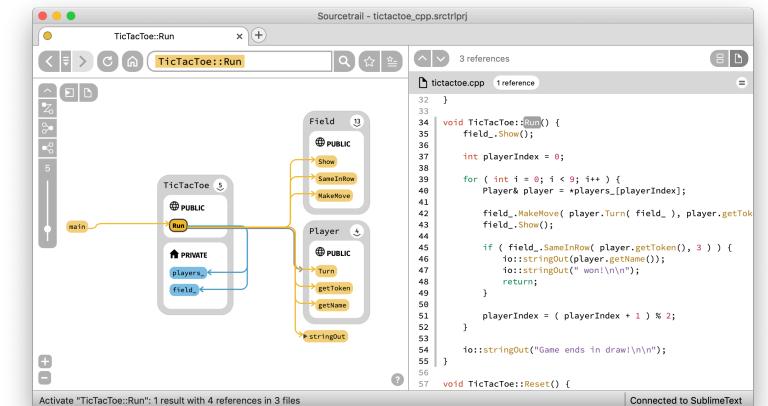
## Cartography tool



## Result visualization



## Extend Sourcetrail

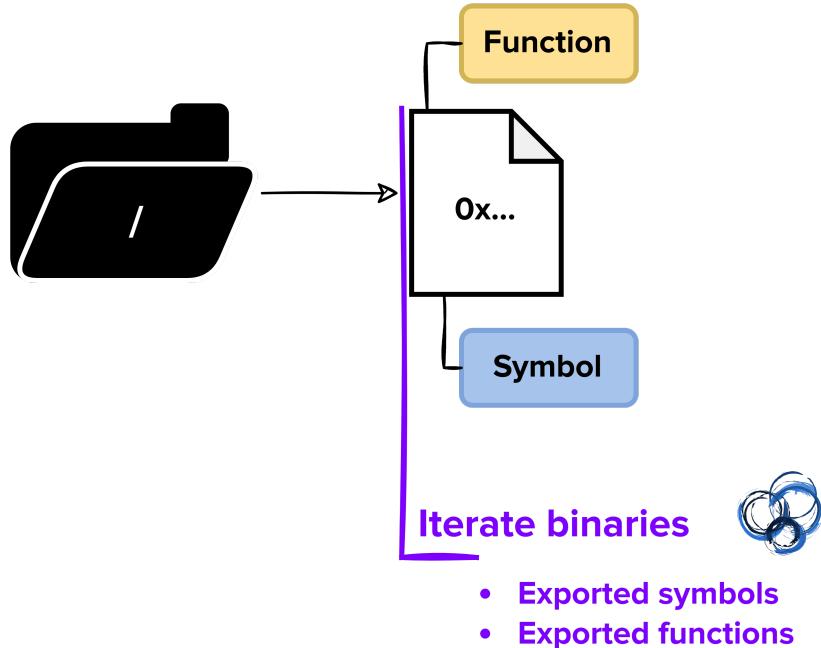




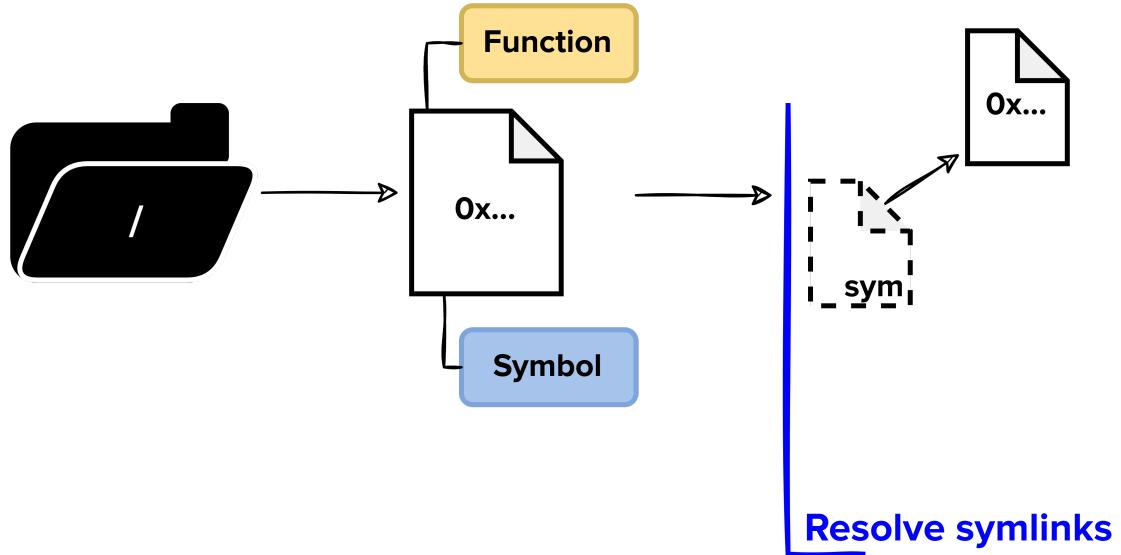
## Filesystem to map

- already extracted
- entry: root directory

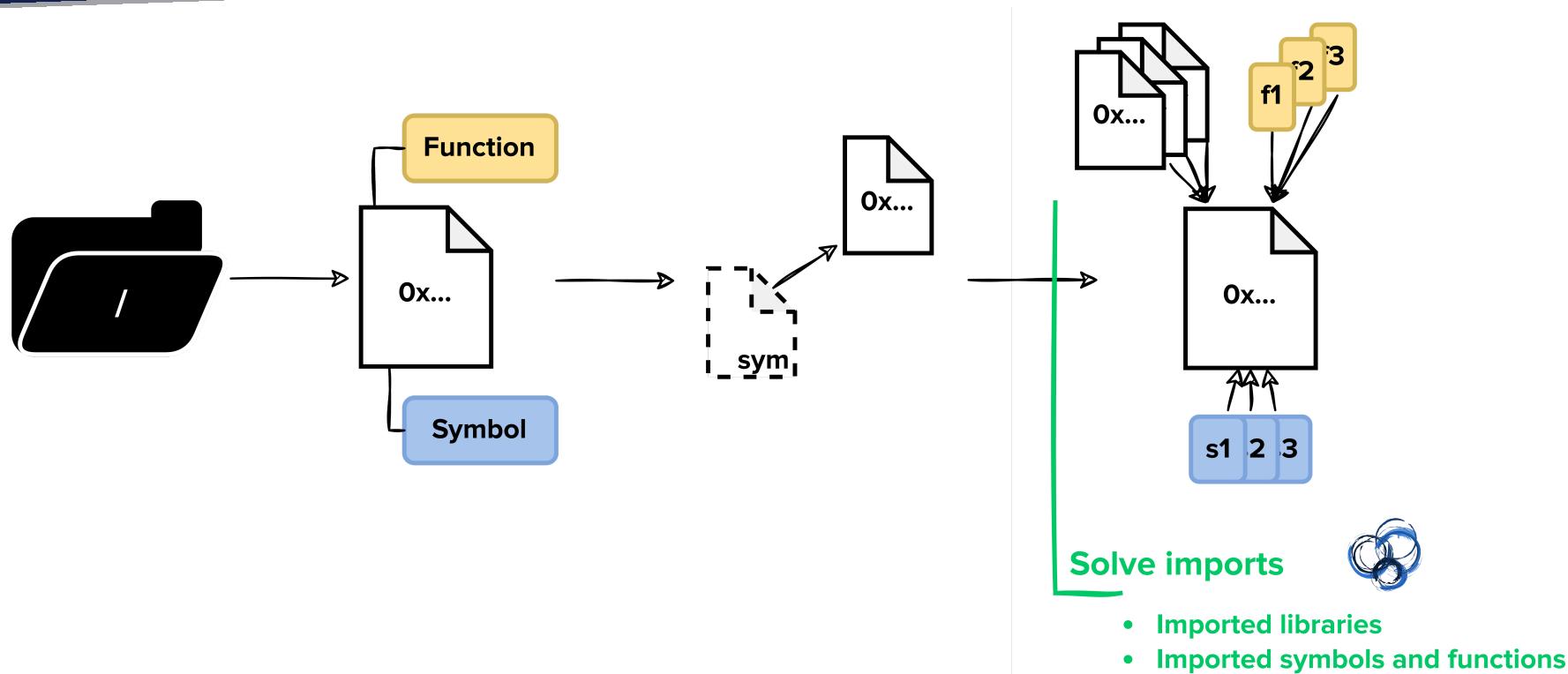
# Pyrrha workflow



# Pyrrha workflow



# Pyrrha workflow



# Demo Time



Q



Netgear RAX30

# Entrypoint identification: theory

secure

Command line

```
$ curl URL
```



insecure

```
$ curl --insecure URL
```



## Fetching an URL

Using the library in C

```
curl_easy_setopt(p1, CURLOPT_URL, url);
curl_easy_setopt(p1, CURLOPT_SSL_VERIFYHOST, 1);
curl_easy_setopt(p1, CURLOPT_SSL_VERIFYPEER, 1);
```



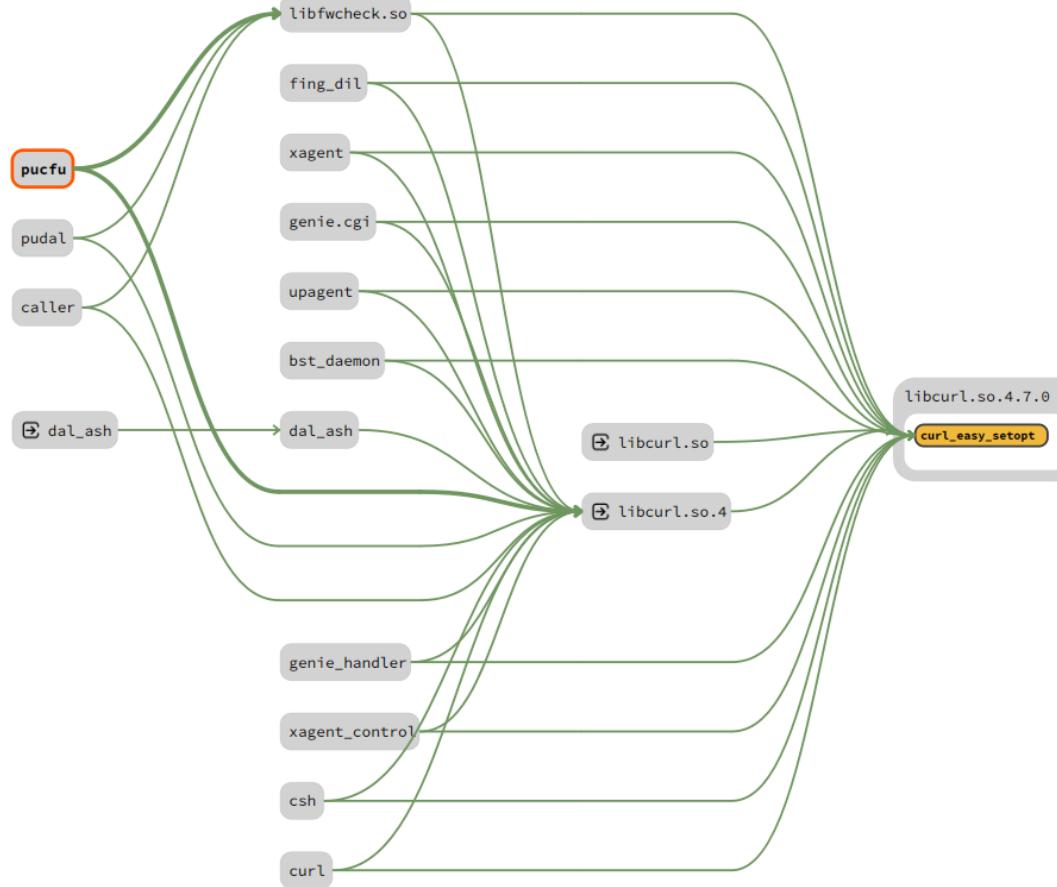
```
curl_easy_setopt(p1, CURLOPT_URL, url);
curl_easy_setopt(p1, CURLOPT_SSL_VERIFYHOST, 0);
curl_easy_setopt(p1, CURLOPT_SSL_VERIFYPEER, 0);
```



# Entrypoint identification: using Pyrrha



Q



# Result Export for intercompatibility

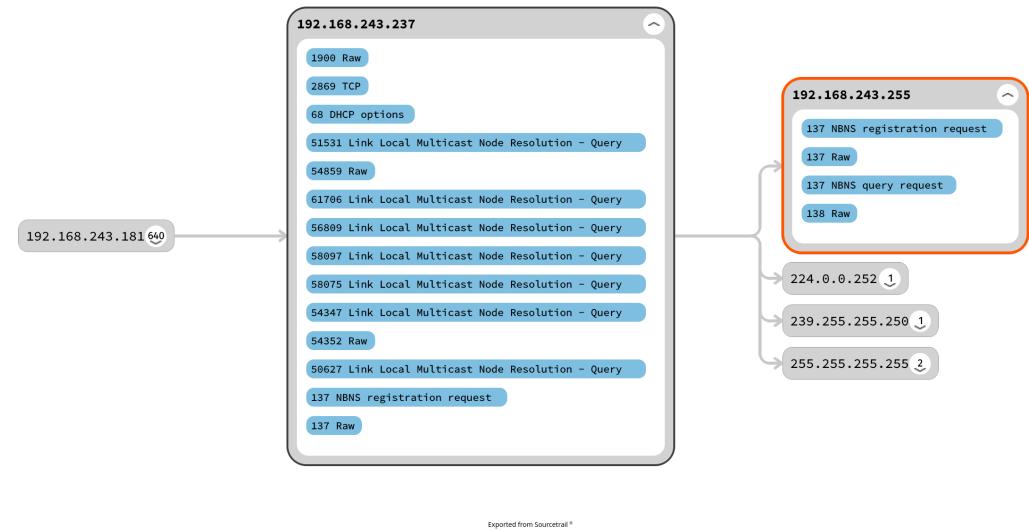
```
Binaries no longer in RAX30-V1.0.9.90_3.json:  
- /lib/libcurl.so.4.6.0  
- /bin/pppoe-relay  
- /bin/websockd  
  
Binaries added in RAX30-V1.0.9.90_3.json:  
- /[KERNEL_VERSION]/kernel/net/netfilter/xt_connlimit.ko  
- /lib/libcurl.so.4.7.0  
[...]  
  
Common binaries that have changed:  
pudil have changed:  
- symbols removed: {'sprintf'}  
- symbols added: {'sleep', 'snprintf'}  
fing_dil have changed:  
- lib removed: {'libcurl.so.4.6.0'}  
- lib added: {'libcurl.so.4.7.0'}  
[...]  
  
Total having changed: 108
```

Diffing use case example.



- ▶ Quarkslab implementation of Sourcetrail API
- ▶ Fully Pythonic
  - ▶  <https://github.com/quarkslab/numbat>
  - ▶ pip install numbat

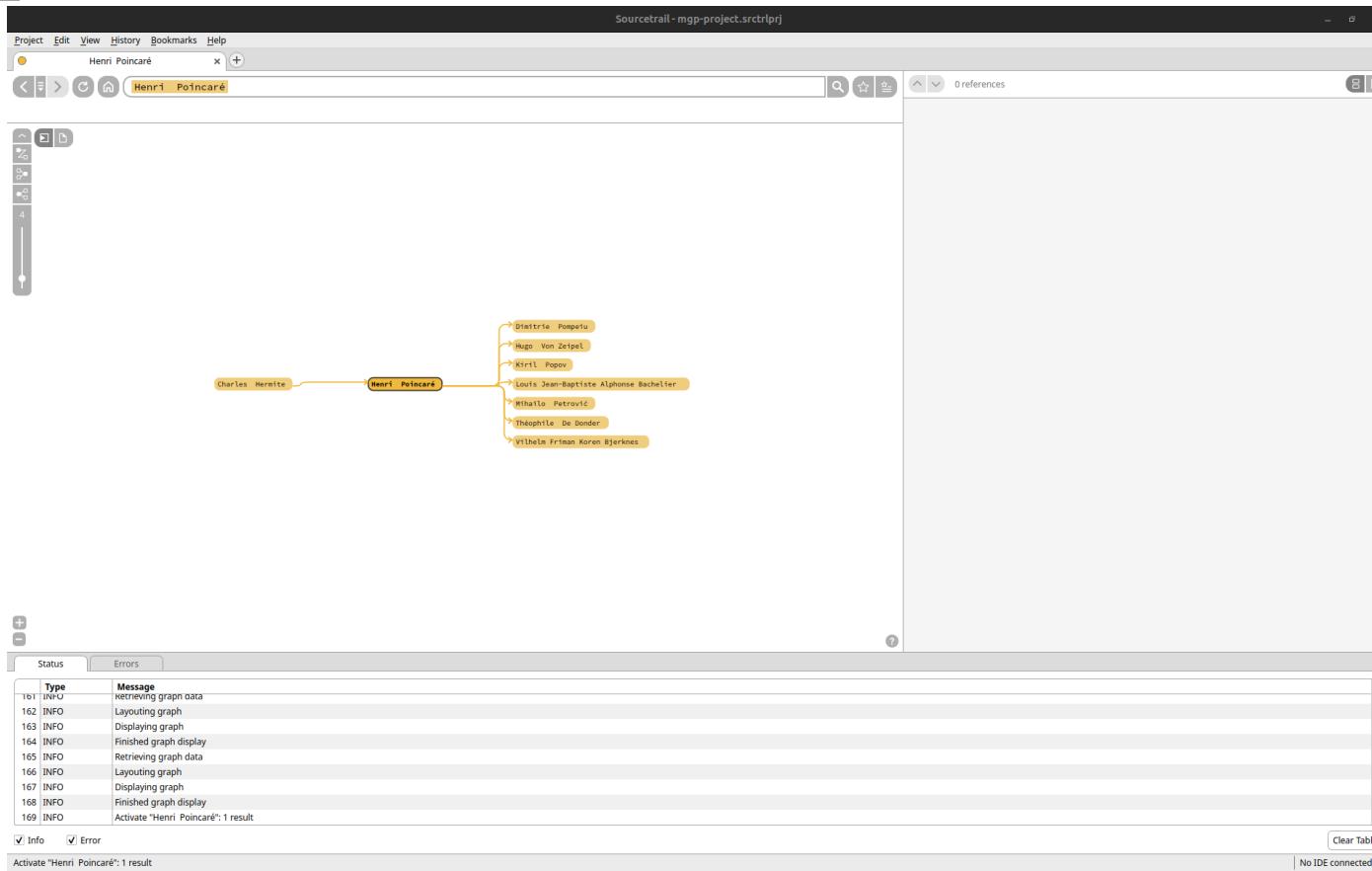
Visualize any graph-like data



# Numbat Example



Q



Data Source: Mathematics Genealogy Project.

# Numbat Usage

```
from numbat import SourcetrailDB

# Create DB
db = SourcetrailDB.open('my_db', clear=True)

# Create a first class containing the method 'main'
my_main = db.record_class(name="MyMainClass")
meth_id = db.record_method(name="main", parent_id=my_main)

# Create a second class with a public field 'first_name'
class_id = db.record_class(name="PersonalInfo")
field_id = db.record_field(name="first_name",
                           parent_id=class_id)

# The method 'main' is using the 'first_name' field
db.record_ref_usage(meth_id, field_id)

# Save modifications and close the DB
db.commit()
db.close()
```



# Conclusion

Q

**Pyrrha:** Firmware mapper (extensible)

- ▶  <https://github.com/quarkslab/pyrrha>
- ▶  pyrrha-mapper

**Numbat:** Map any graph into Sourcetrail

- ▶  <https://github.com/quarkslab/numbat>
- ▶  numbat

## Automatization

- ▶ Speed up some tasks to give more time for specialized analyses
- ▶ Cannot solve everything

## Future Work

- ▶ Customized fork of Sourcetrail
- ▶ Add more features to Pyrrha

# Thank you!

Contact information:

Email: [ebrocas@quarkslab.com](mailto:ebrocas@quarkslab.com)

Phone: +33 1 58 30 81 51

Website: <https://www.quarkslab.com>