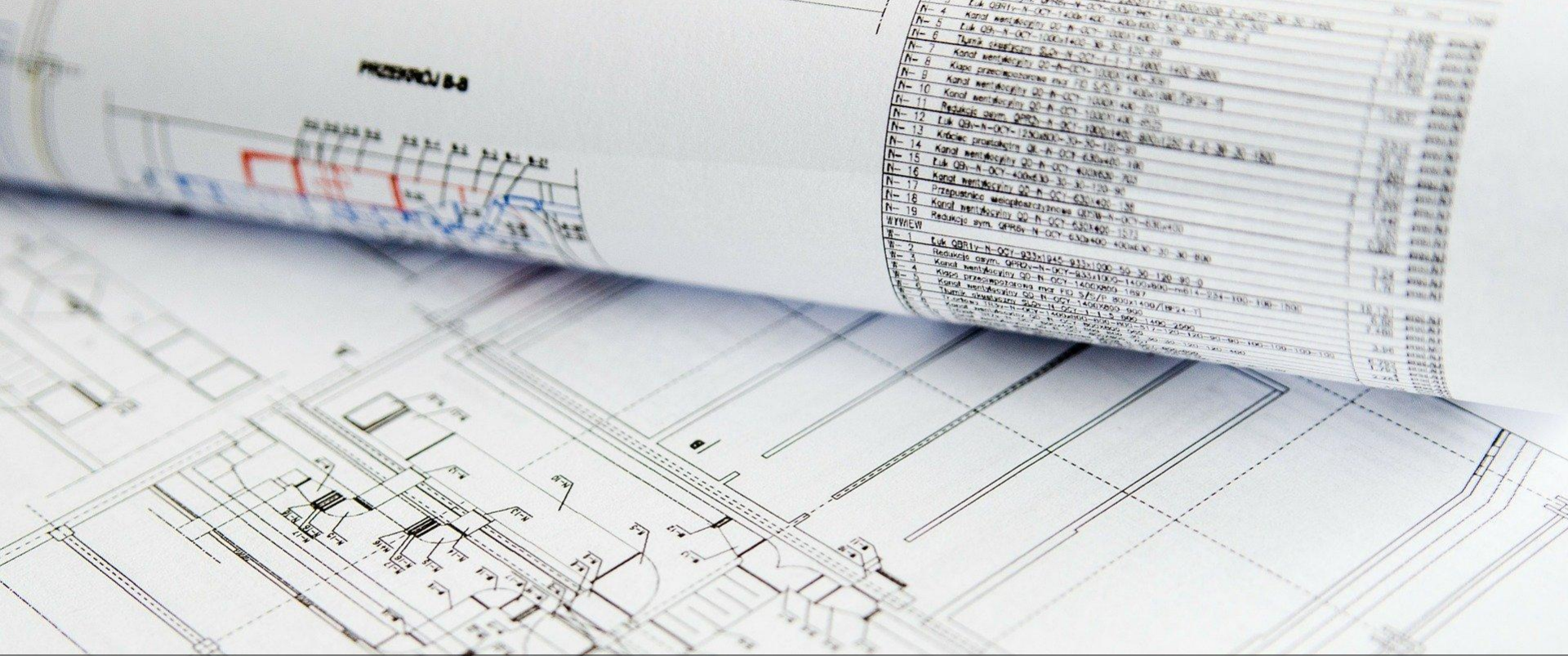


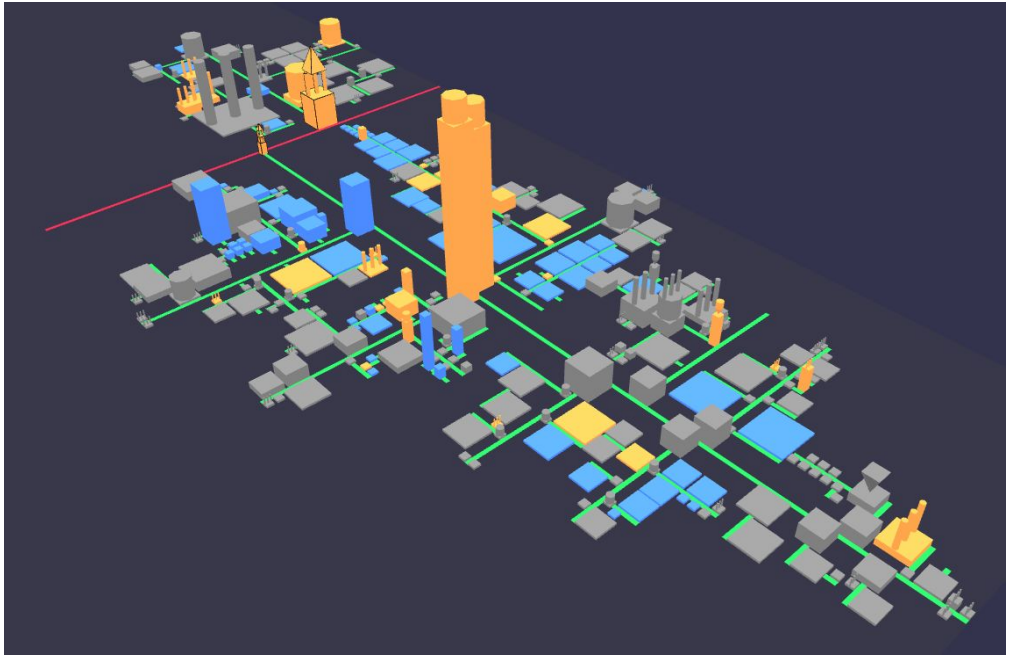
ROSE Festival @ ICSME 2021
September 2021, 29

“You’re an artifact, VariCity.”

Johann Mortara — Philippe Collet — Anne-Marie Dery-Pinna

VariCity

3D visualization of variability implementations (OO metrics)

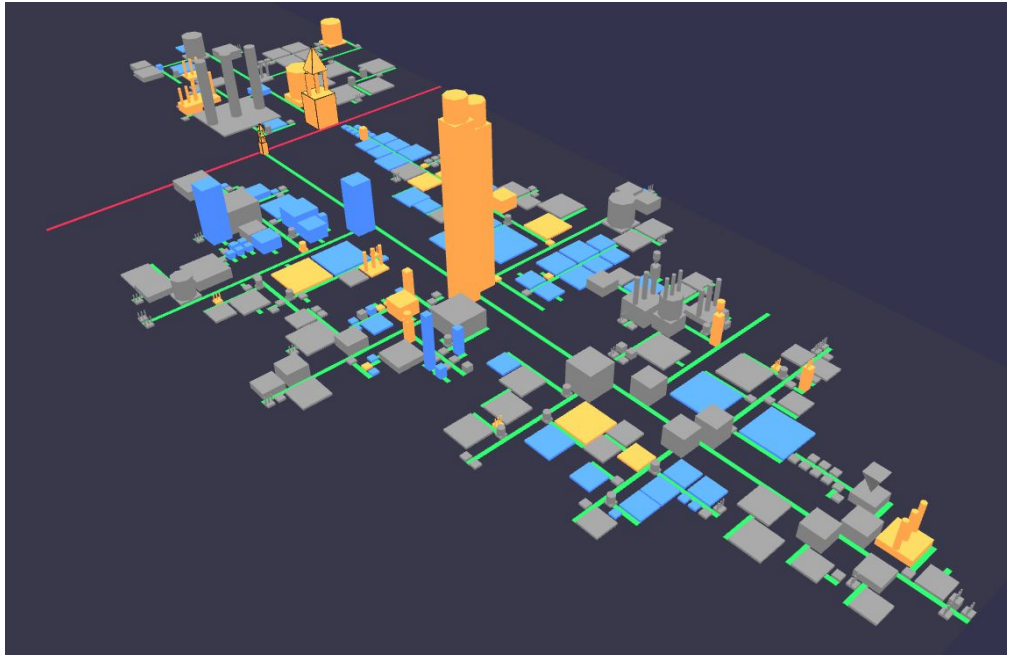


VariCity

3D visualization of variability implementations (OO metrics)

Technological stack:

- **Language:** TypeScript built with Node.js
  ⇒ eased dependencies management / build
- **3D framework:** Babylon.js
- **Deployment:** Webpack
  ⇒ visualization accessible through a web browser

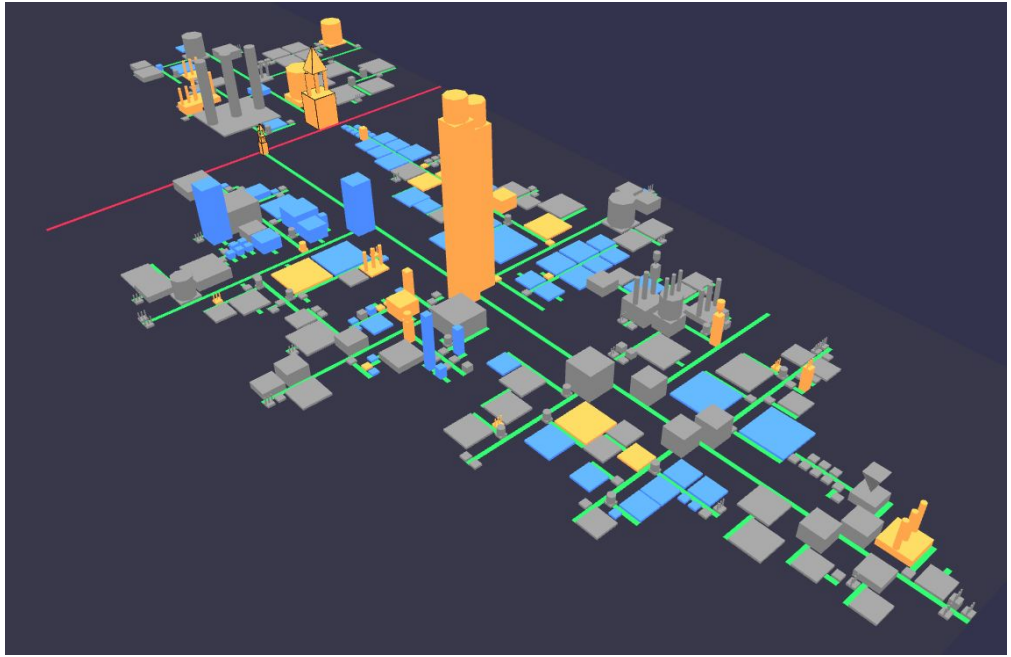


VariCity

3D visualization of variability implementations (OO metrics)

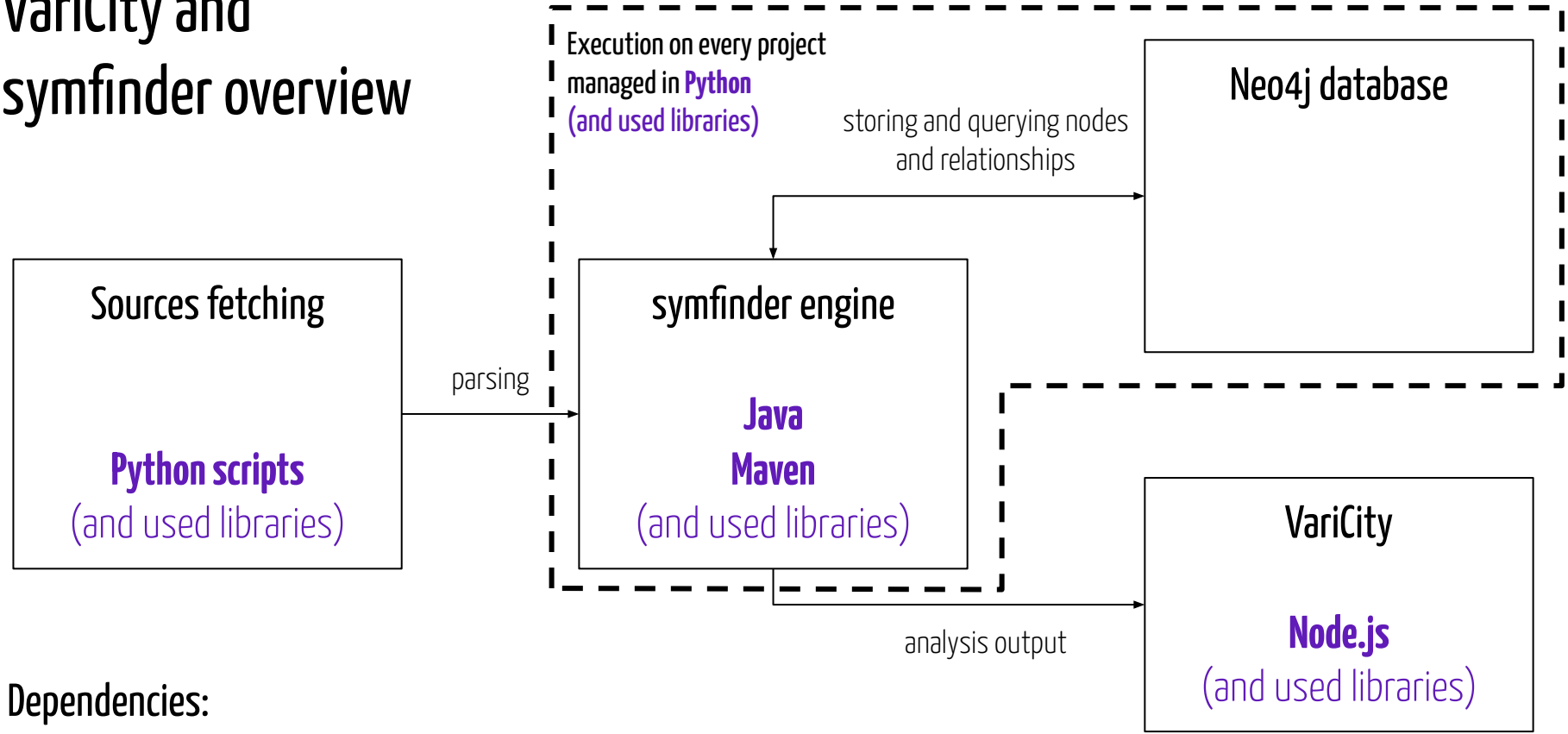
Technological stack:

- **Language:** TypeScript built with Node.js
  ⇒ eased dependencies management / build
- **3D framework:** Babylon.js
- **Deployment:** Webpack
  ⇒ visualization accessible through a web browser



Choice of technological stack driven by the ease of use and reuse

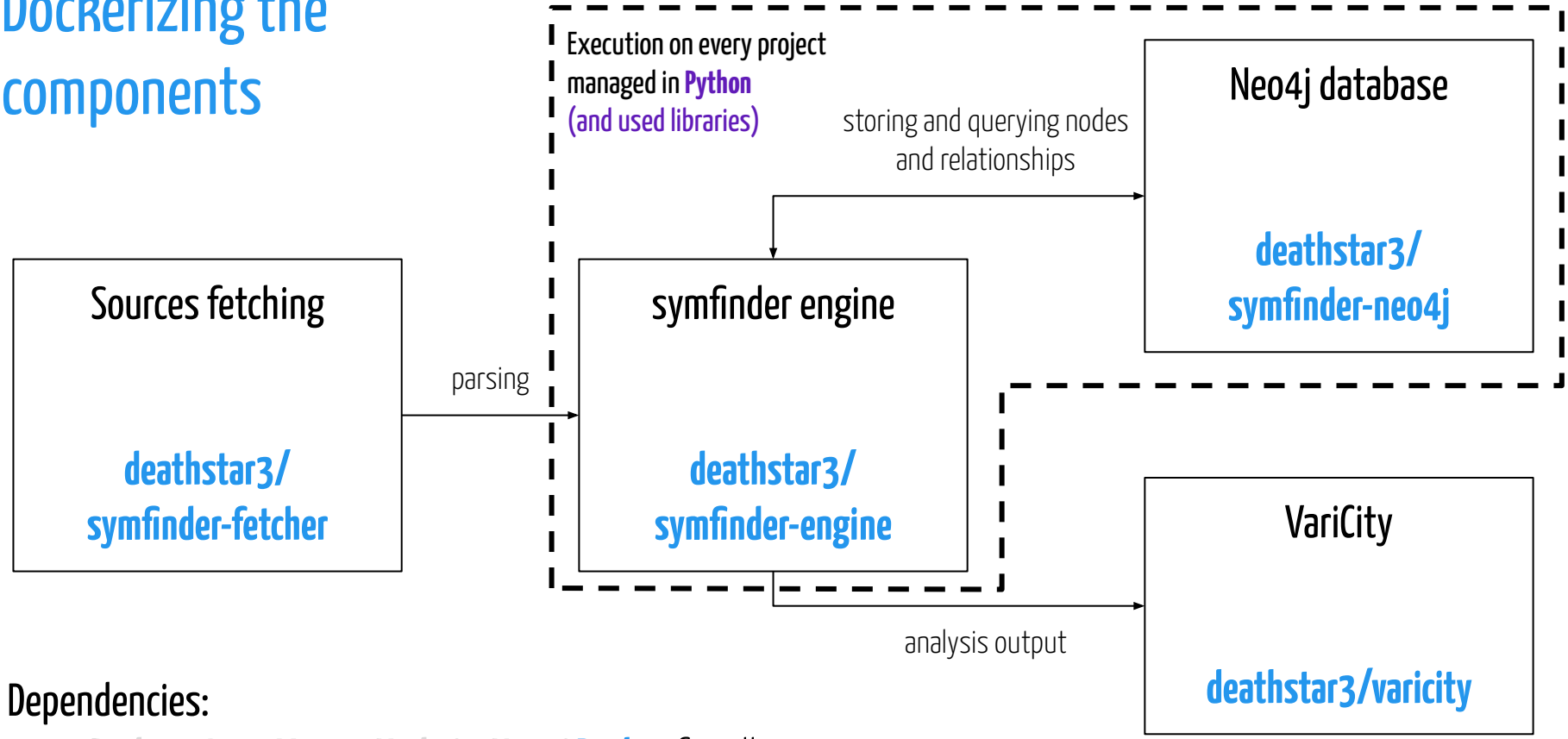
VariCity and symfinder overview



Dependencies:

- **Python, Java, Maven, Node.js, Neo4j** for all components
- **Python, Bash / Batch** for the execution

Dockerizing the components

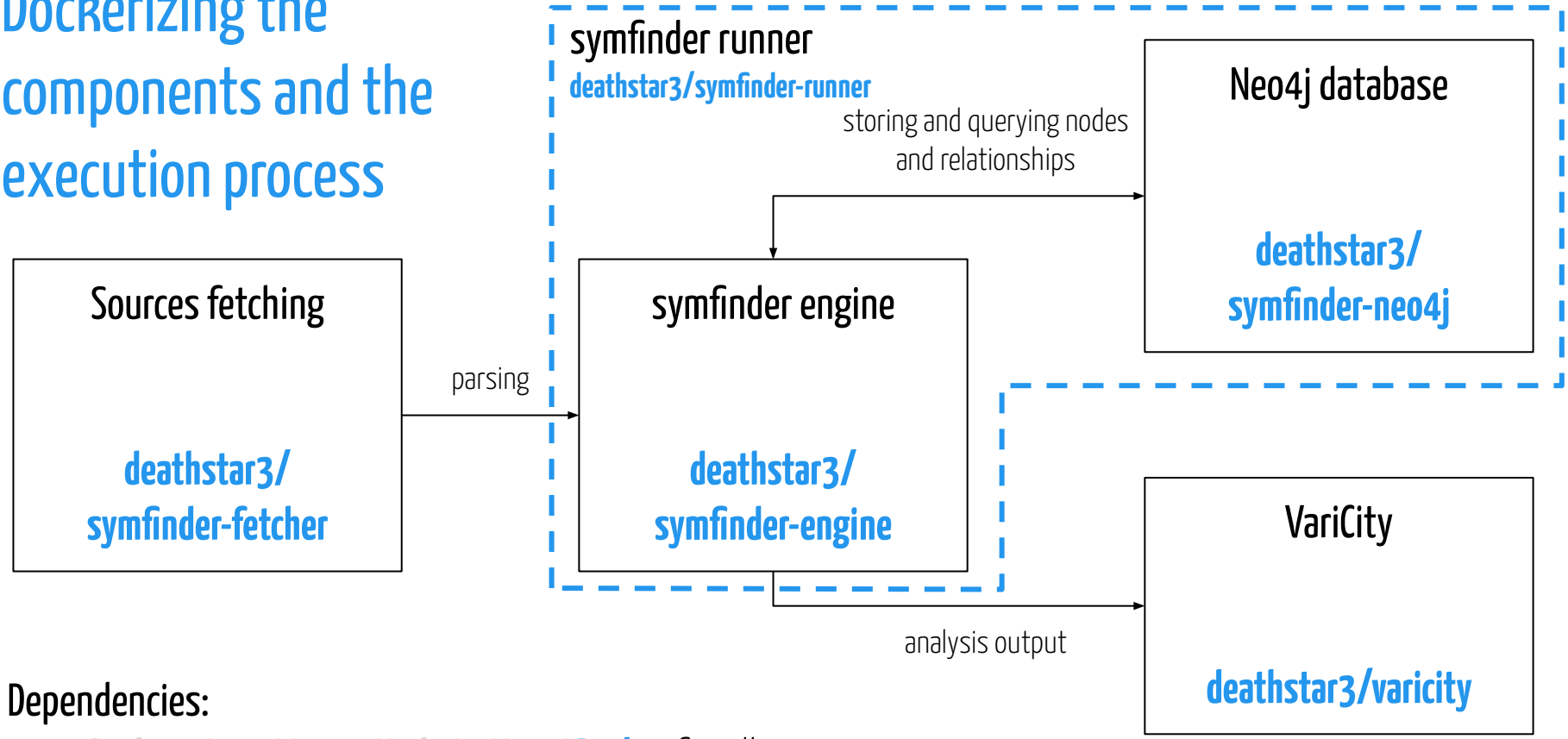


Dependencies:

- ~~Python, Java, Maven, Node.js, Neo4j~~ **Docker** for all components
- **Python, Bash / Batch** for the execution

Used Docker images

Dockerizing the components and the execution process



Dependencies:

- ~~Python, Java, Maven, Node.js, Neo4j~~ **Docker** for all components
- ~~Python,~~ **Bash / Batch, Docker** for the execution

Used Docker images

Functional & Reproducible

Functionality goal: the reviewer can run the visualization

The tool must be easy to setup and run

Already built docker images available on Docker Hub

⇒ no build required

Visualization data generated by symfinder already given

⇒ the reviewer is not obliged to run symfinder on all projects as it may take time

Trials on clean machines having ≠ OSes

⇒ made us realize that Docker on Windows and MacOS needed additional settings, that we added to the documentation of the tool

Functional & Reproducible

Reproducibility goal: the reviewer can reproduce the city exploration scenarios presented in the paper

<u>The usage instructions must be clear</u>

Scenarios are detailed step by step, from the setup to the expected output

⇒ only the needed technical details are given to keep it simple

Trials by people not knowing the project

⇒ are the instructions clear enough for someone external to run the tool?

Functional & Reproducible ⟶ Reusable

Reusability goals: practitioners can easily apply symfinder / VariCity on their own projects

<u>The tool must be easy to reuse</u>

Configuration external to the code

⇒ no modification inside the code needed

Detailed guide for reuse given

⇒ how to setup a new project to analyse...

Detailed technical documentation

⇒ the tool can be modified for other needs

“You’re an artifact, VariCity.”

Johann Mortara — Philippe Collet — Anne-Marie Dery-Pinna

Reproduction package:

https://doi.org/10.5281/zenodo.5034199

Get the paper on VariCity:

https://hal.archives-ouvertes.fr/hal-03312487

Obtained reproducibility badges

Open Research Objects 

Research Objects Reviewed 

symfinder obtained an ACM Reusable badge at SPLC’19 

VariCity website:

https://deathstar3.github.io/varicity-demo/