presentation 09.03.2023

Construction of a software solution for detection of end-of-support dependencies

- ***Context***:
  dependency management

- ***Goal***:
  tooling to problem-solve

- ***Demarcation***:
  not full-fletched, rather pioneering

Problem Definition

- ***Focus***:
  end-of-support dependencies

- ***Why***:
  risk, expense & cost
  in management, development & planning

... e.g. **issues** with, and **concerns** about

- common vulnerabilities and exposures
  - compliance
- stability
  - bugs without fixes or patches
- compatibility
  - out-datedness and actuality

-> risk

… consequenting

- replacement
- upgrades
- independent fixes and patches

-> expense & cost

- ***Solution***:
  mitigating the impact of consequences
  from end-of-support dependency

Methods

- ***What***:
  a software solution
  to detect end-of-support dependencies

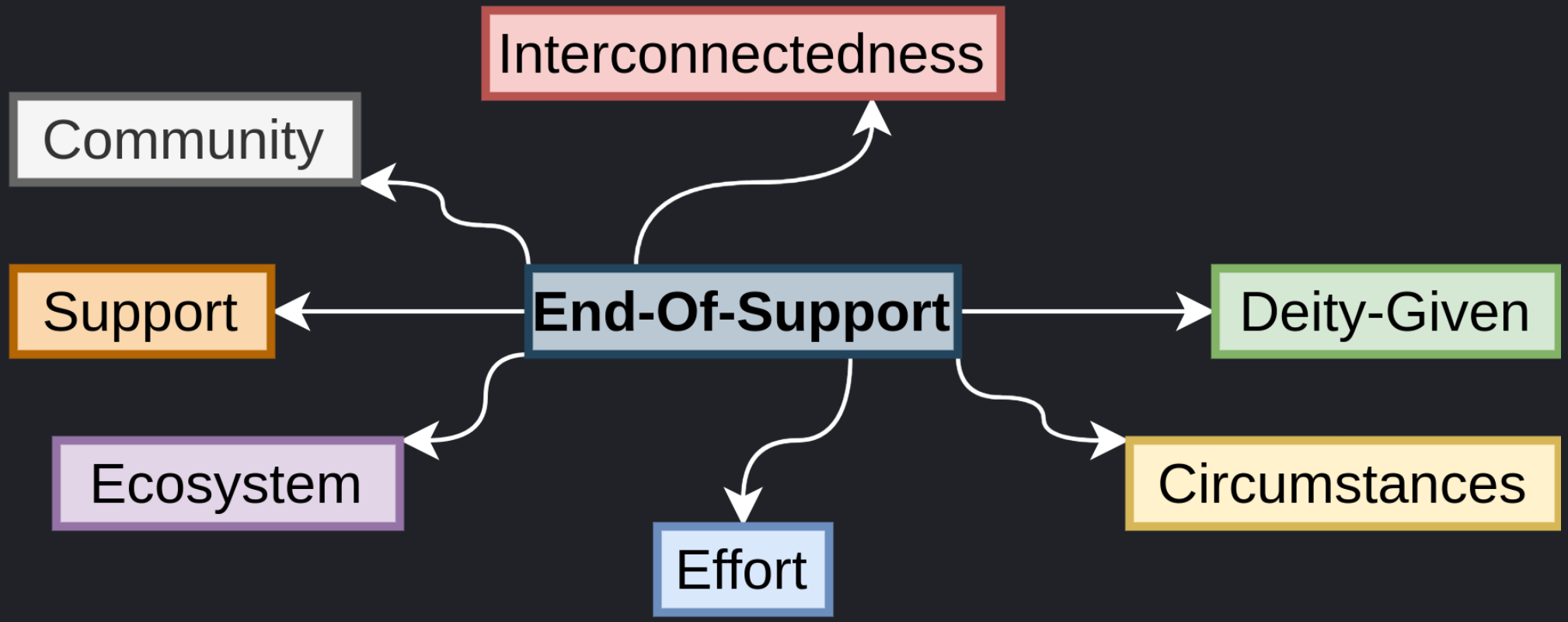
- ***How***:
  conception → requirements → design → implementation → evaluation

Results

- Theoretical Concept
- Requirements Analysis
- Technical Concept
- ***deprec* / *deprec-cli***

Theoretical Concept

EOS Abstraction Framework



EOS Factor | → *Statements* | → *Signals & Metrics*

Requirements Analysis

- applicability in practice
- suitability for effective use

-> project-based & automation

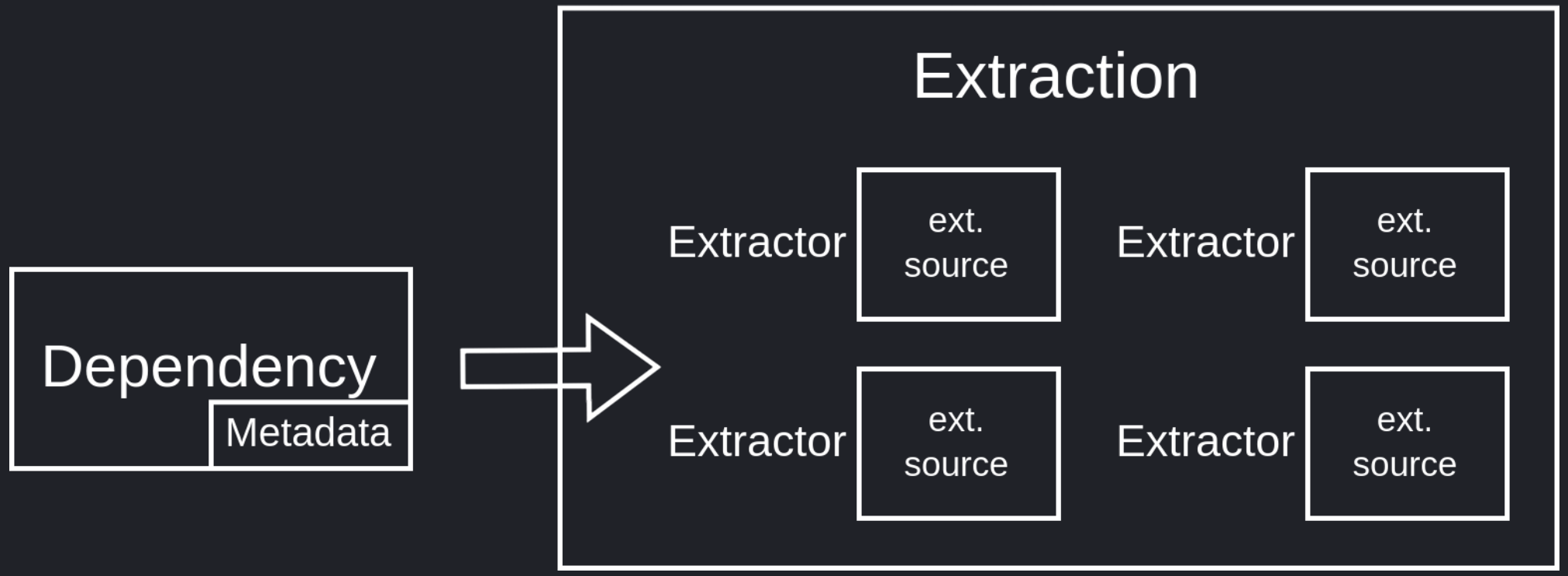
- achieve continuance
- enable further proceedings

-> independence

Technical Concept

- software bill of materials (sboms)
  ↪ CycloneDX

- extraction


Dependency
Metadata
Extraction
Extractor
ext. source
Extractor
ext. source
Extractor
ext. source
Extractor
ext. source

- data model

- combination and conclusion

(EOS Abstraction Framework)

deprec

- integrating the **theoretical** concept
- implementing the **technical** concept

open source

written in

supporting dependencies
from

extracting data
from

GitHub

extracting data
from

deprec-cli

```
$> deprec-cli <sbom> <opts>
```

open source

written in

THANKS