UNIVERSITÄT KOBLENZ · LANDAU

Combining Multiple Dimensions of Knowledge in API Migration

Thiago Bartolomei, Mahdi Derakhshanmanesh, Andreas Fuhr, Peter Koch, Mathias Konrath, Ralf Lämmel, Heiko Winnebeck

Contribution of this presentation

Present a *framework* combining *multiple dimensions* of knowledge to *support API migration*.

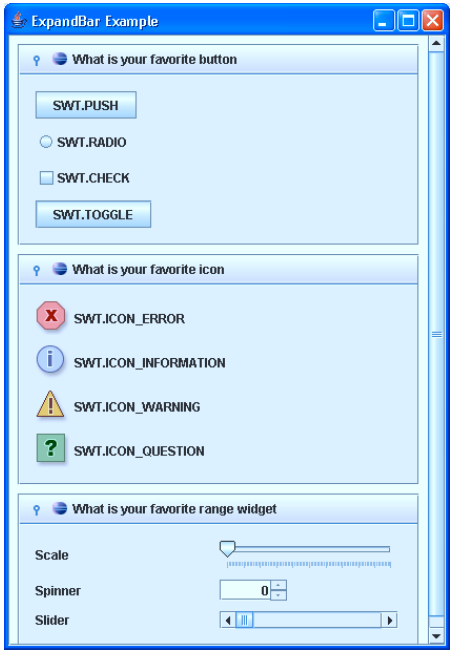
API Migration

- Special case of software migration
- Adapt software system to
  - replace old API (source API) by
  - new API (target API)
  - of the same domain (e.g., GUI development)

Why API Migration?

- Legacy APIs may be
  - outdated, not supported anymore
- New APIs may provide
  - new features
  - better performance, reliability, usability, …
  - support for new system environments

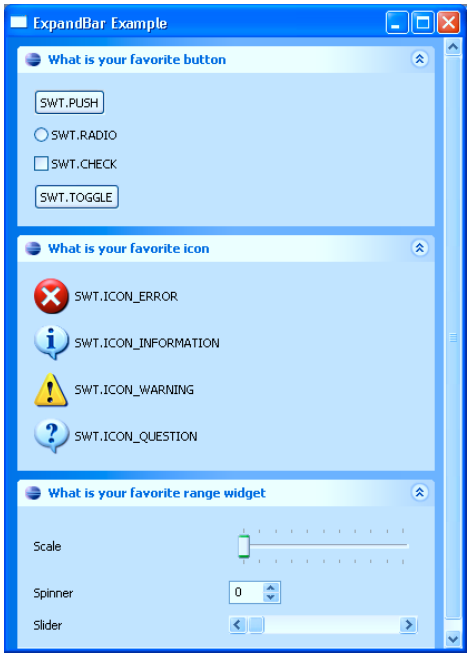
Why API Migration: Example



Swing

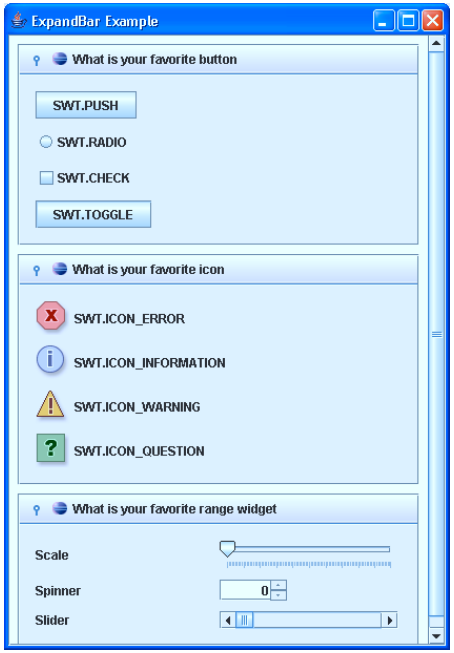
SwingWT

- Style



SWT

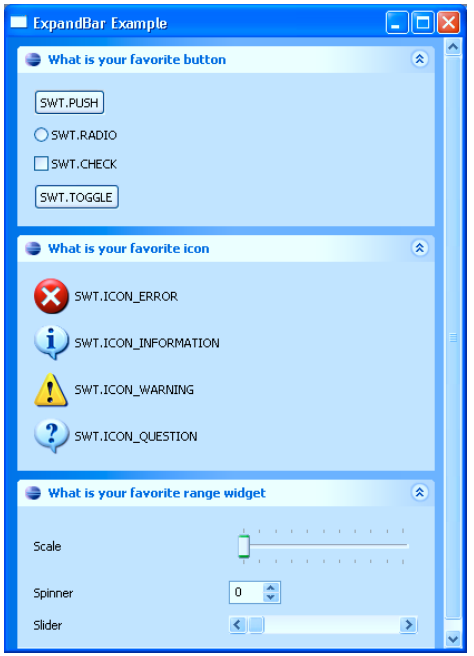
Why API Migration: Example



Swing

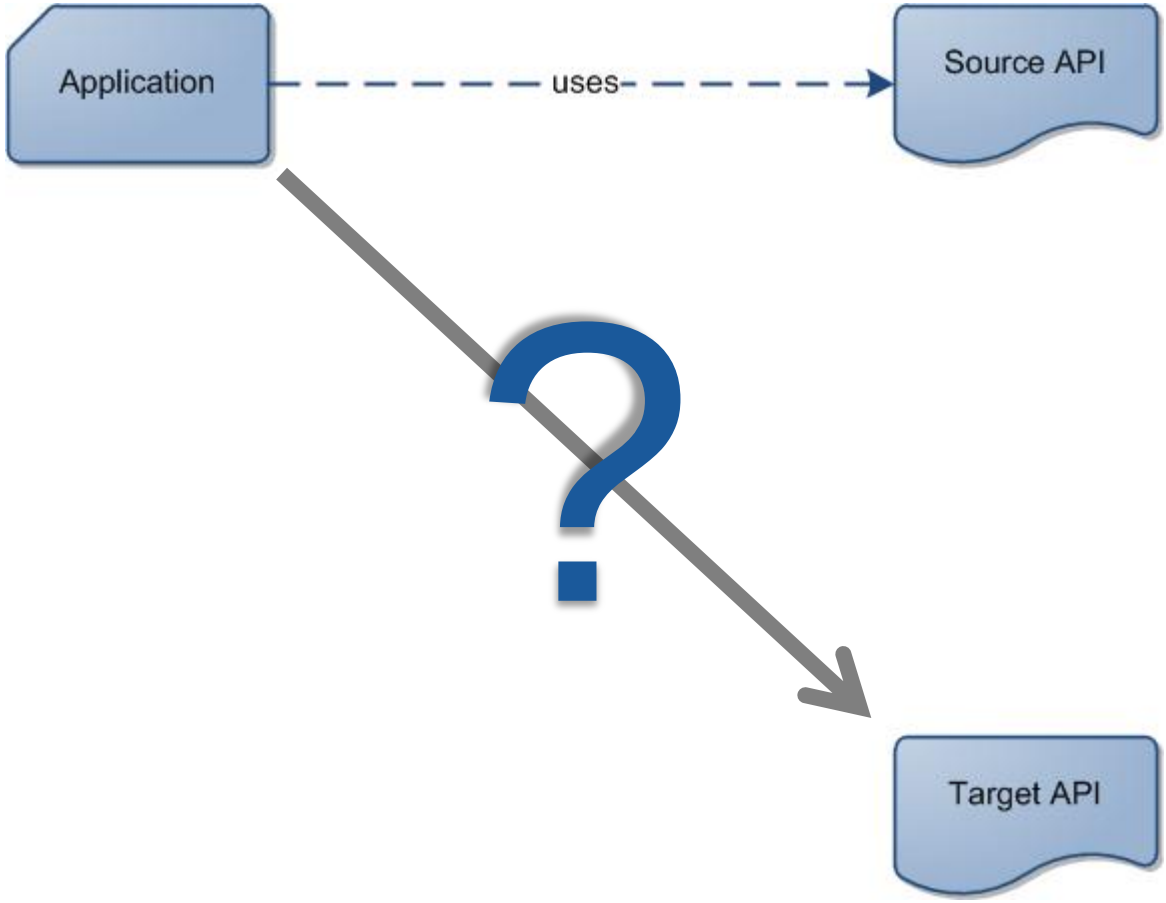
SWTSwing

- Compability

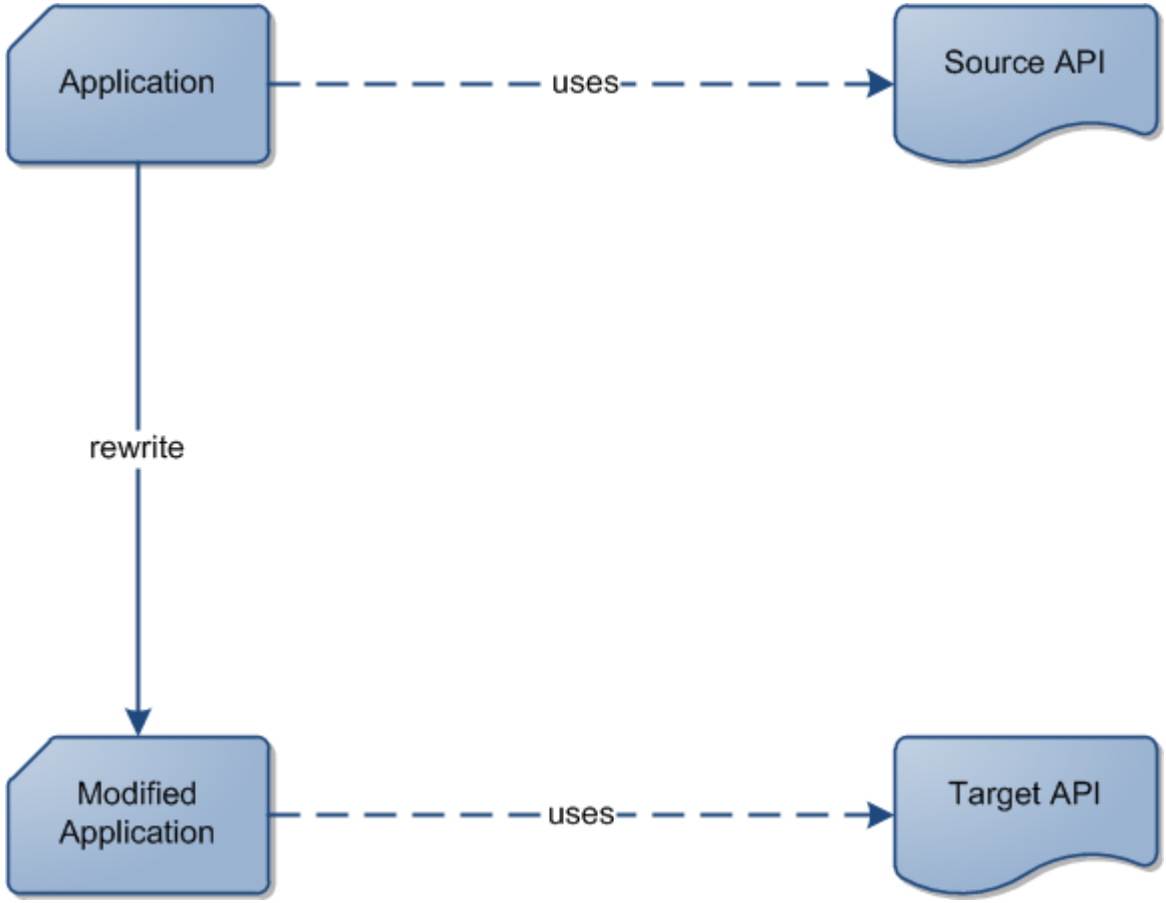


SWT

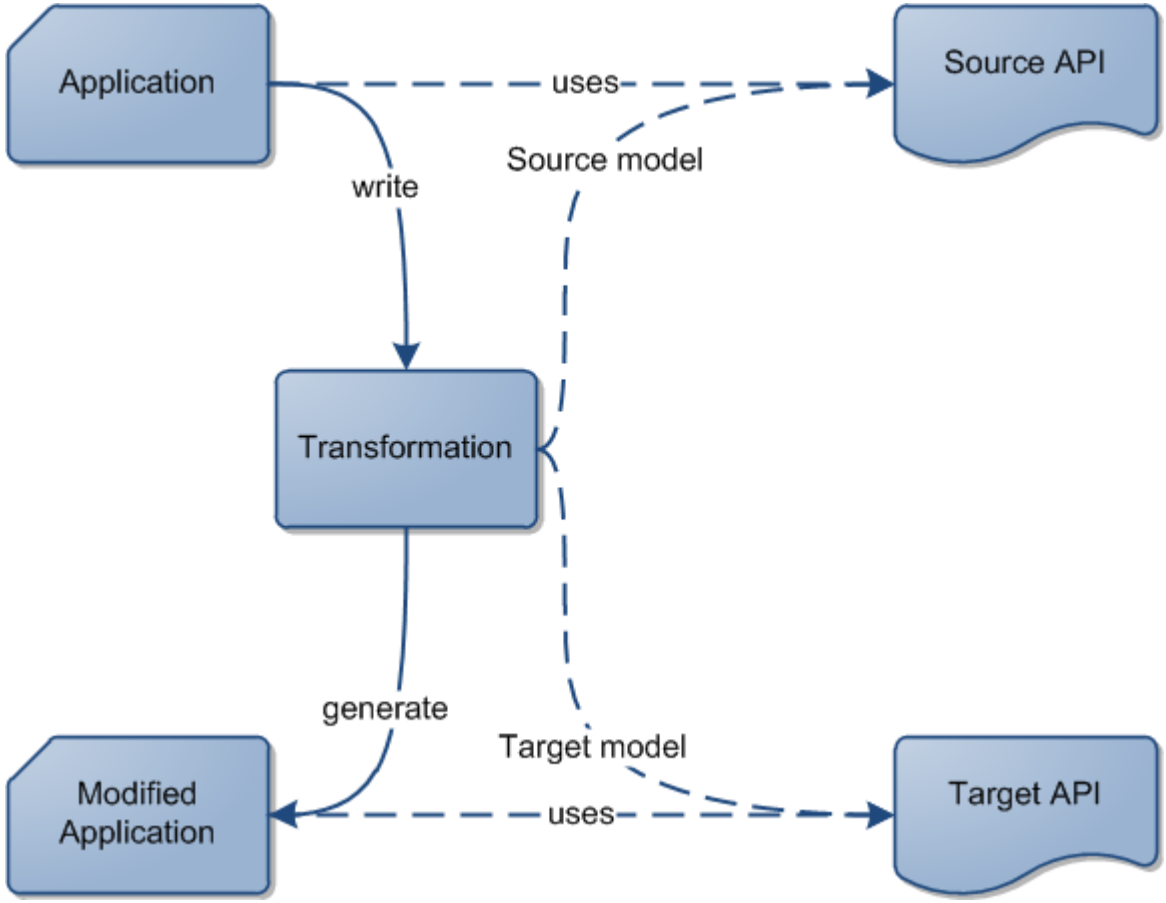
API Migration Approaches



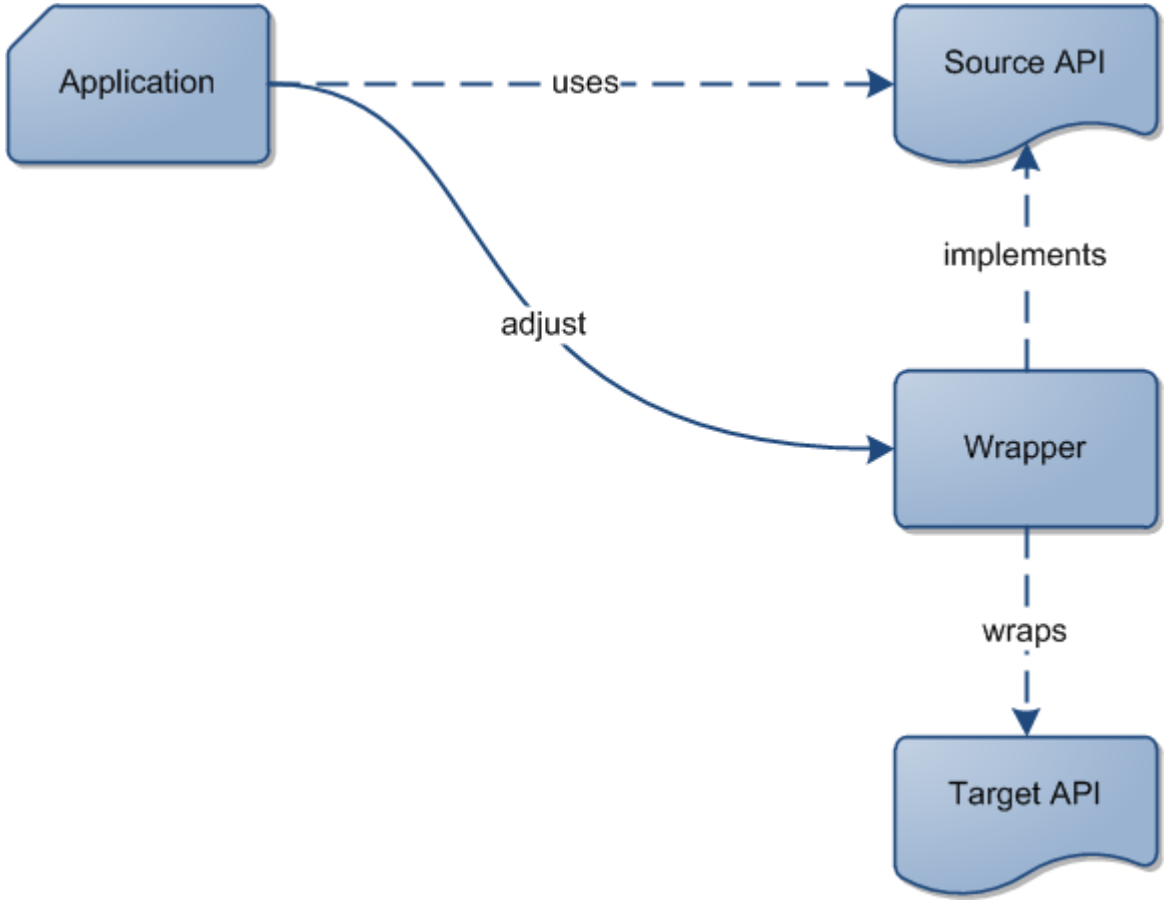
API Migration: Reeimplentation



API Migration: Transformation



API Migration: Wrapping



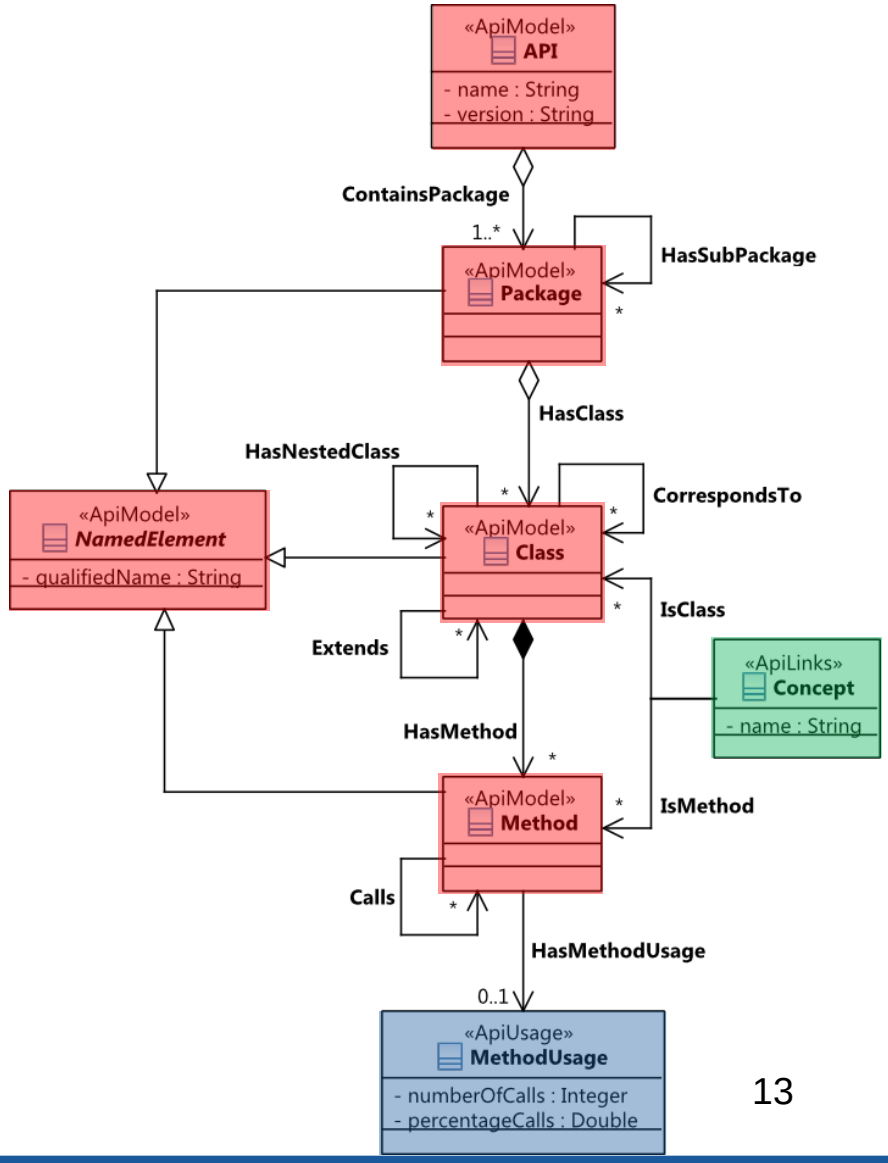
Roadmap

- Integrated Repository
  - Multiple dimensions of knowledge
- Wrapper Assessment
  - How to rate wrappers on their suitability?
- Guidance for Migration
  - How to extend/write own wrappers/transformations?

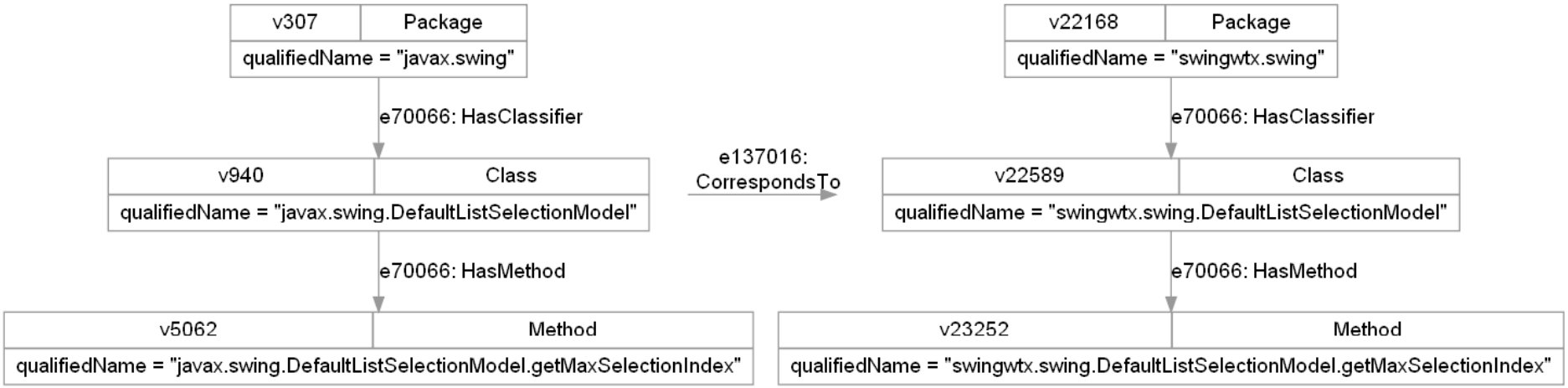
INTEGRATED REPOSITORY

Metamodel

- Simplified Java sources
- API Usage properties of 1476 SourceForge projects
- Ontology on API concepts

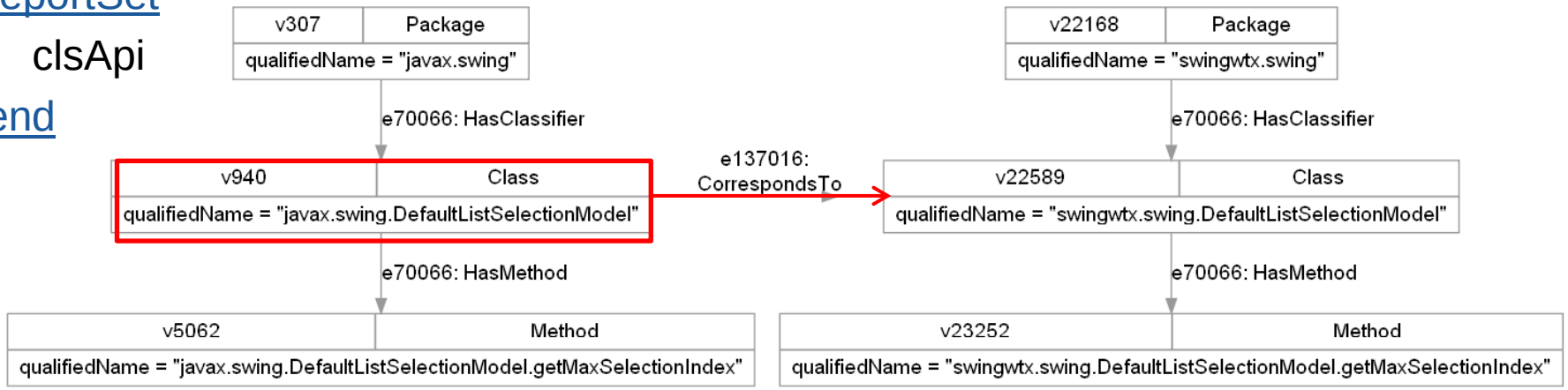


Repository Technology: TGraphs



Repository Technology: Graph Querying with GReQL

```
from
    clsApi: V{Class}
with
    clsApi.qualifiedName =~ “javax\.swing\..*" and
    count(clsApi-->{CorrespondsTo}) > 0
reportSet
    clsApi
end
```



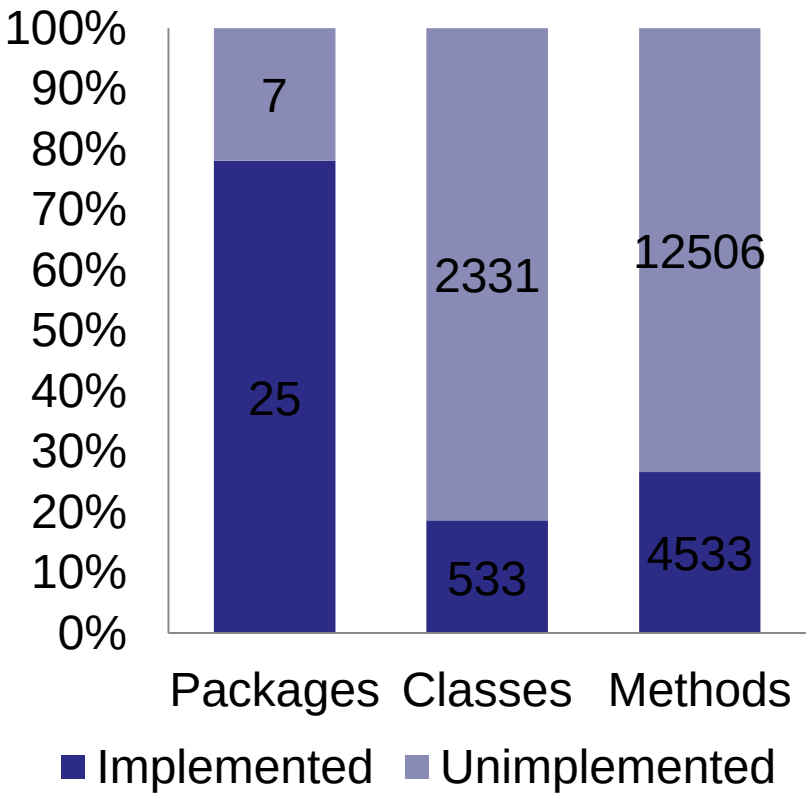
WRAPPER ASSESSMENT

Wrapper Assessment: Goals

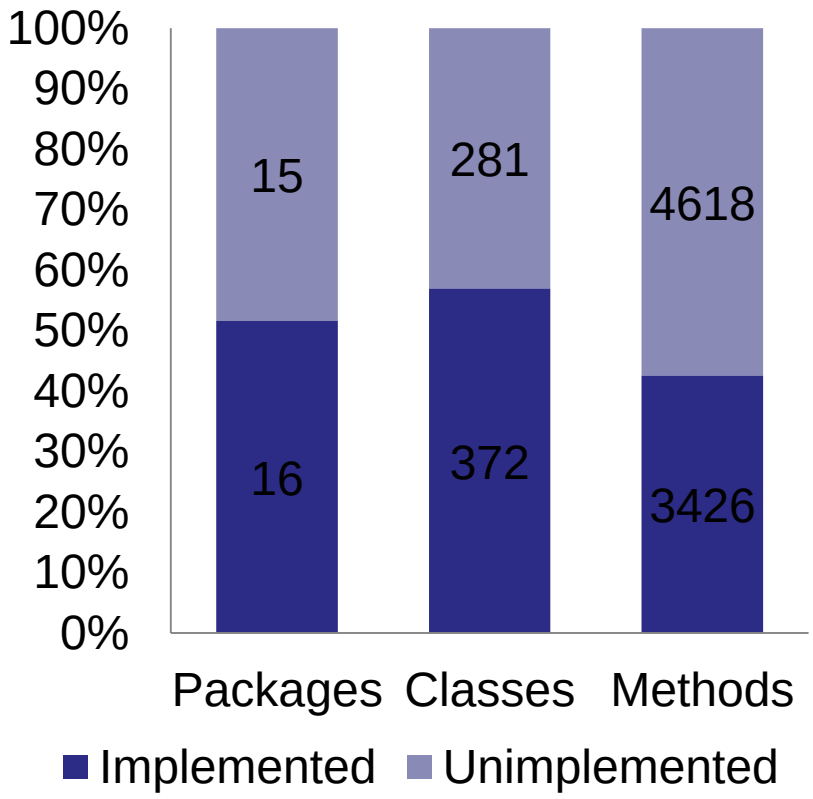
- Compare different wrappers for same wrapping task
- Track development of own wrappers

Wrapper Assessment: Source API Coverage

SwingWT



SWTSwing



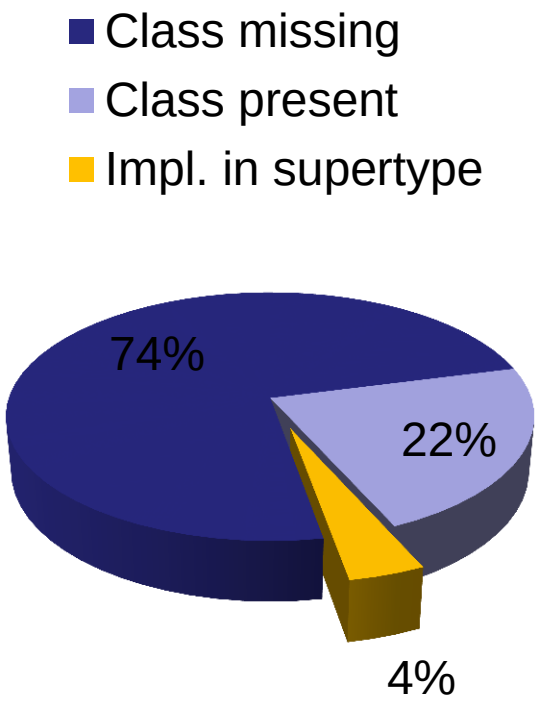
Wrapper Assessment: Wrapper Compliance

- Simple coverage statistics do not cover more complex dependencies
  - Declarations in supertypes
  - Empty implementations
- Simple covarage does not reflect usage of APIs in real projects

Compliance: Declarations on supertypes

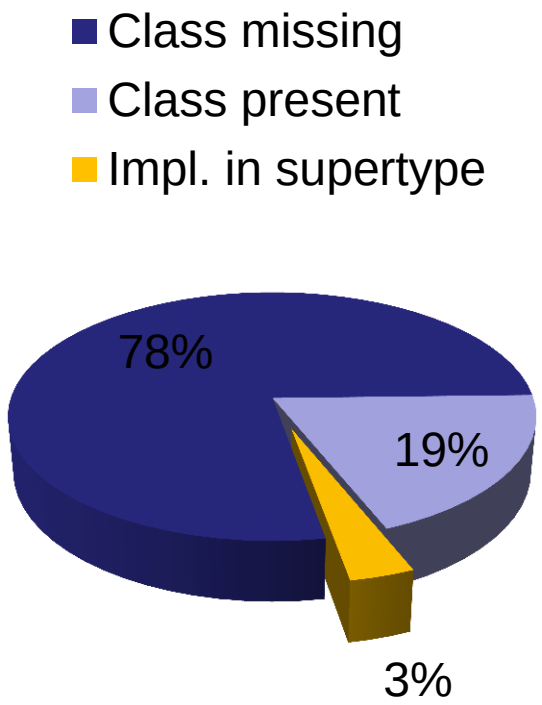
SwingWT

Missing methods



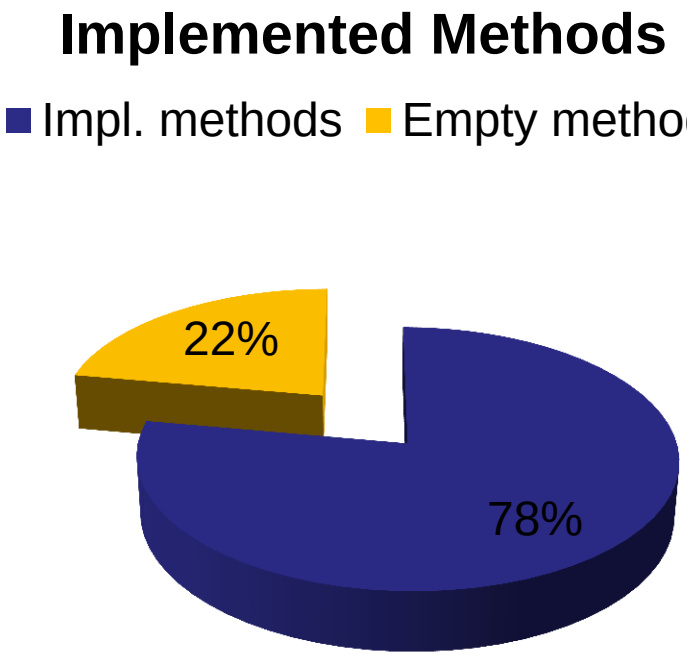
SWTSwing

Missing methods

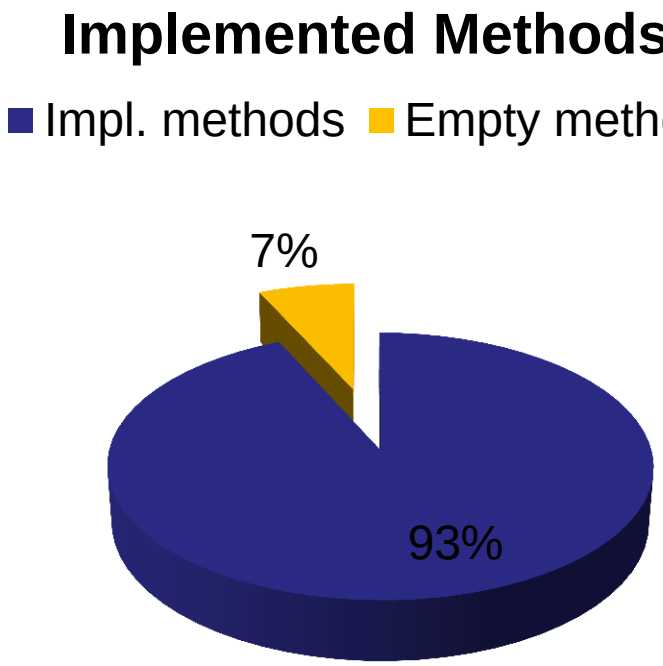


Compliance: Empty methods

SwingWT

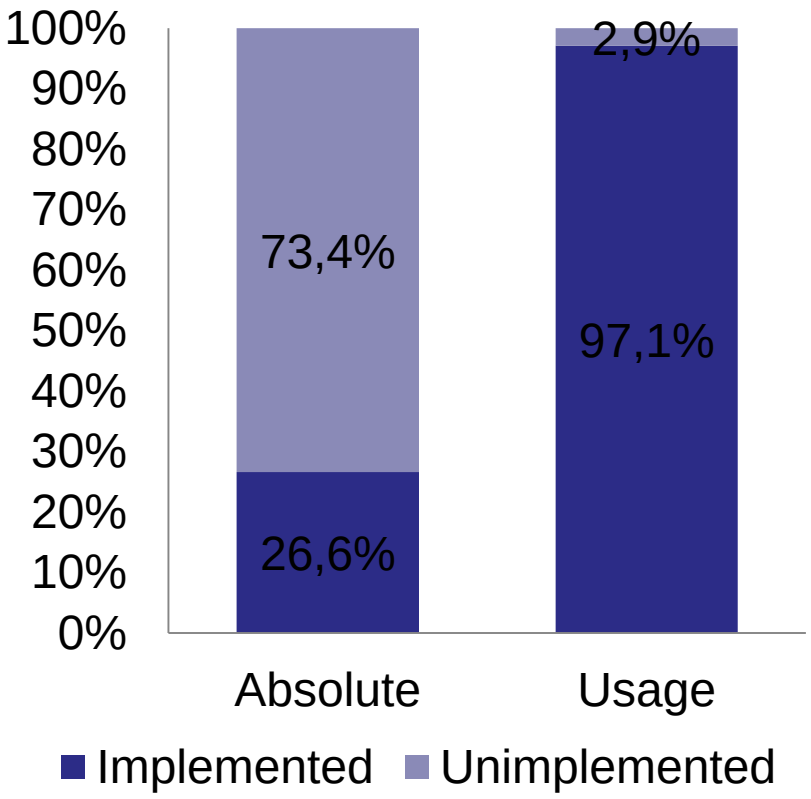


SWTSwing

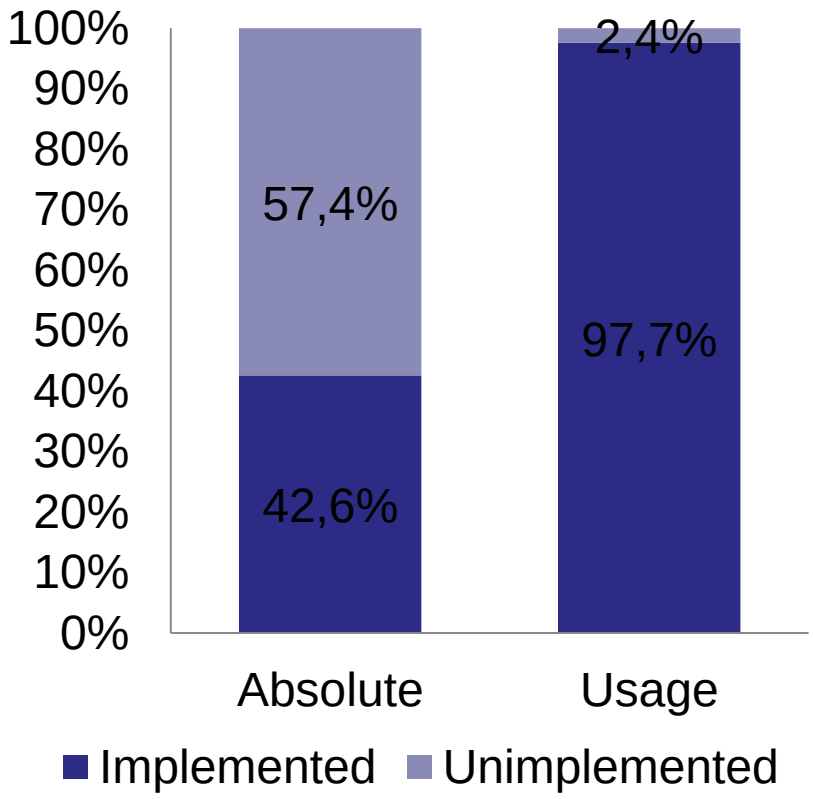


Wrapper Assessment: Relevance in Terms of Usage

SwingWT



SWTSwing



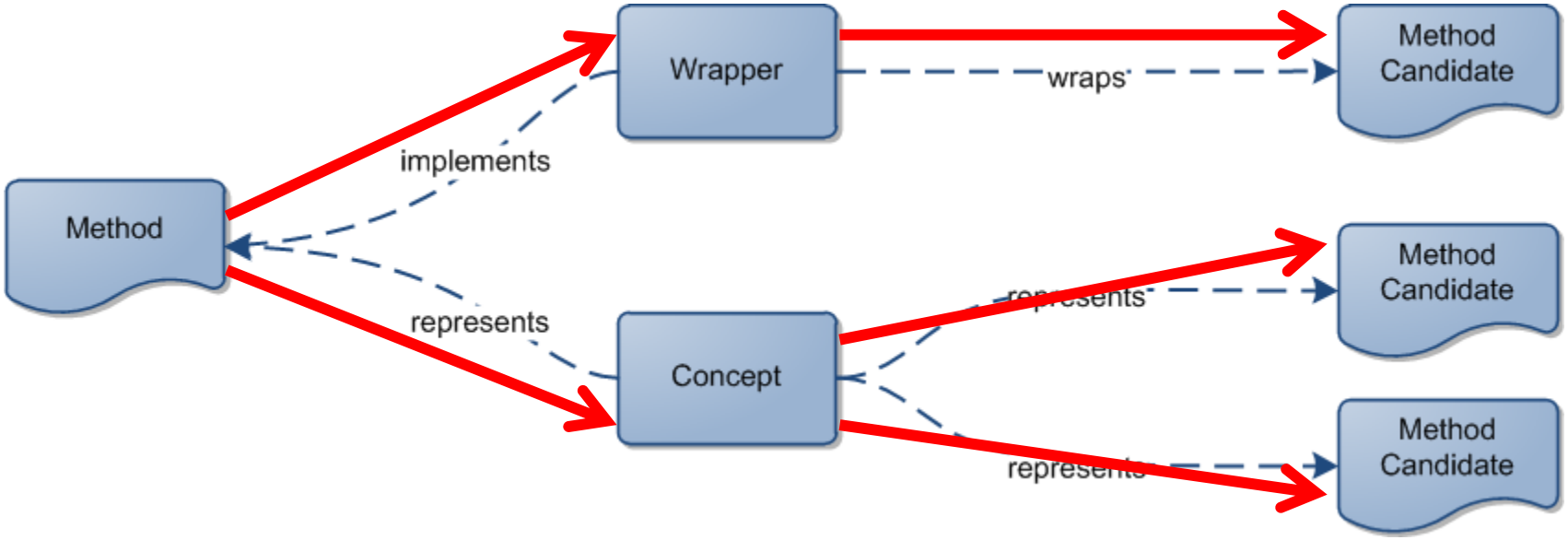
GUIDANCE FOR MIGRATION

Guidance for Migration: Goals

- Extend existing wrappers
- Write own wrappers / reimplementations / transformations

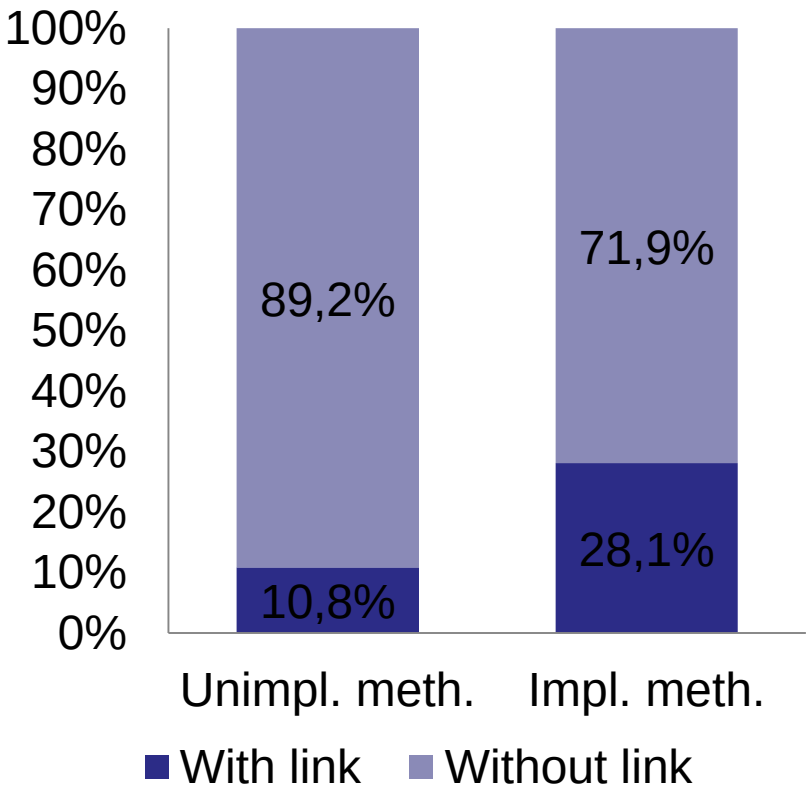
→ Identify target API code suited to implement source API methods

Guidance for Migration: Concept-based Method Candidates

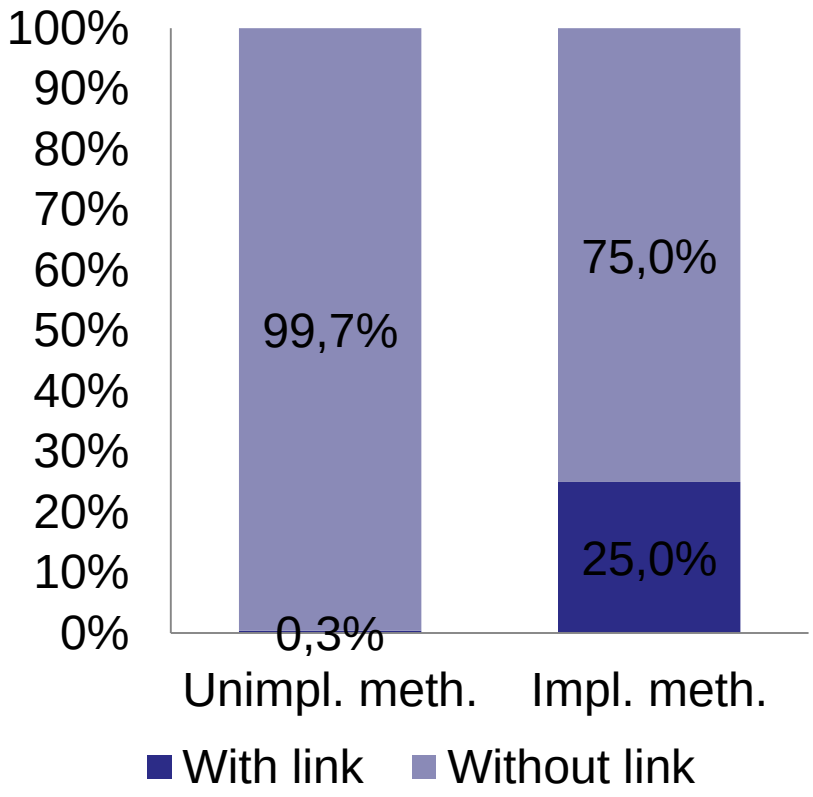


Guidance for Migration: Assessment of the Ontology

SwingWT

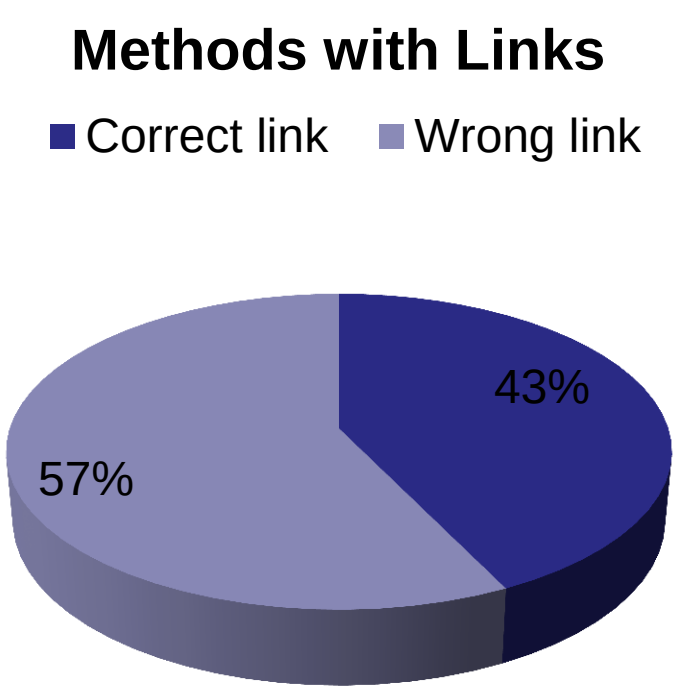


SWTSwing

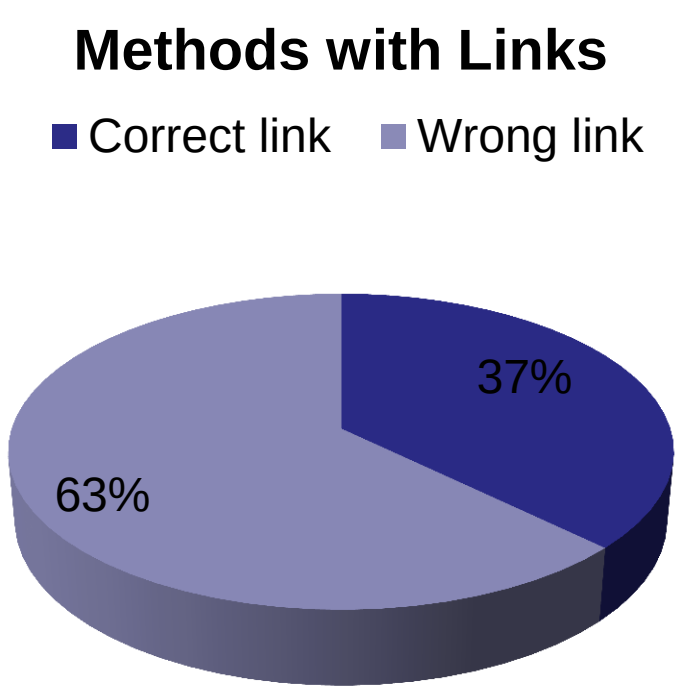


Guidance for Migration: Ontology Correctness

SwingWT



SWTSwing



THANK YOU FOR LISTENING!

QUESTIONS?