POLYTECHNIQUE MONTRÉAL

Service Identification to Support the Migration of Legacy Systems to SOA

Manel Abdellatif
Polytechnique Montréal
Canada
manel.abdellatif@polymtl.ca

Supervised by:

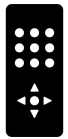
Naouel Moha
École de Technologie Supérieure
Canada

&

Yann-Gaël Guéhéneuc
Concordia University
Canada

Context

Maintenance tasks are central activities in many businesses



Legacy software systems are essential in many businesses

70% of today's transactional operations are managed by legacy systems

Legacy systems cannot simply be removed / replaced as they execute complex business logic

Legacy systems are critical

but challenging to maintain

Shortcomings

- Maintenance Costs
- Lack of Flexibility
- Lack of Scalability
- Lack of Support

Migrating your legacy system

to the SOA stack

Legacy systems migration remains essential to ease their maintenance and make them more evolvable without loosing their business values

Migration of legacy systems to SOA

Statelessness

Abstraction

Autonomy

Reusability

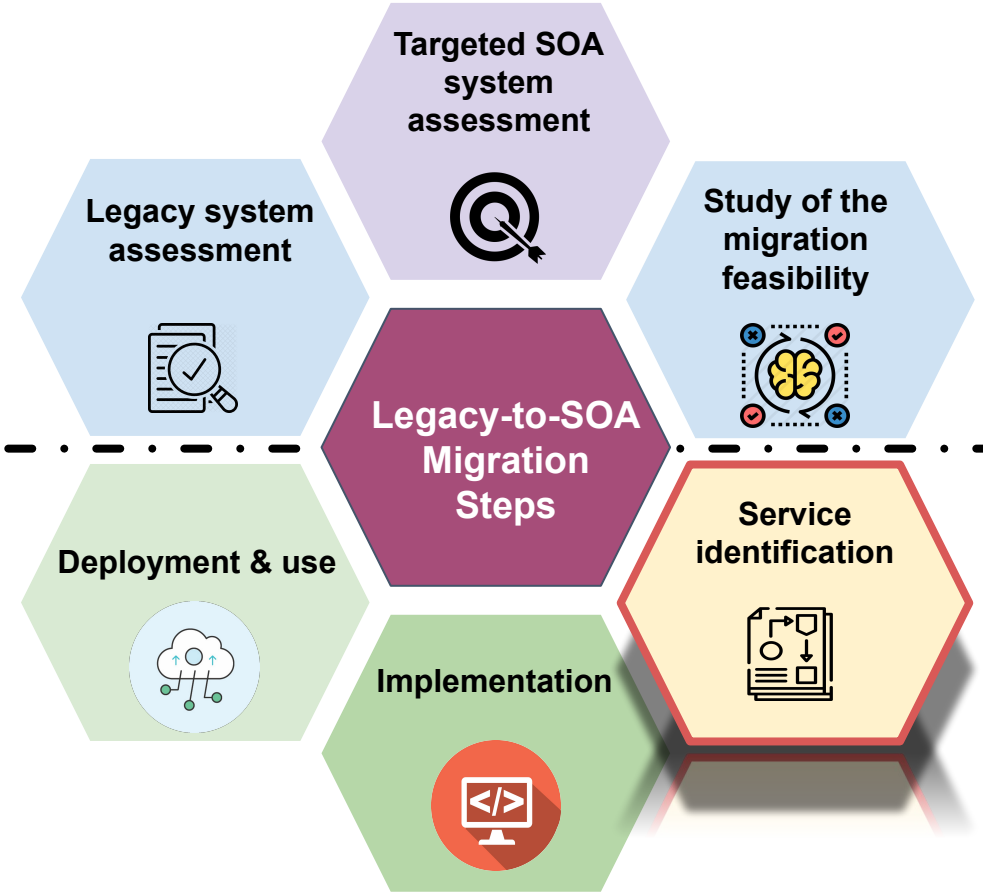
Interoperability

Composability

Migration steps

Planification of the migration

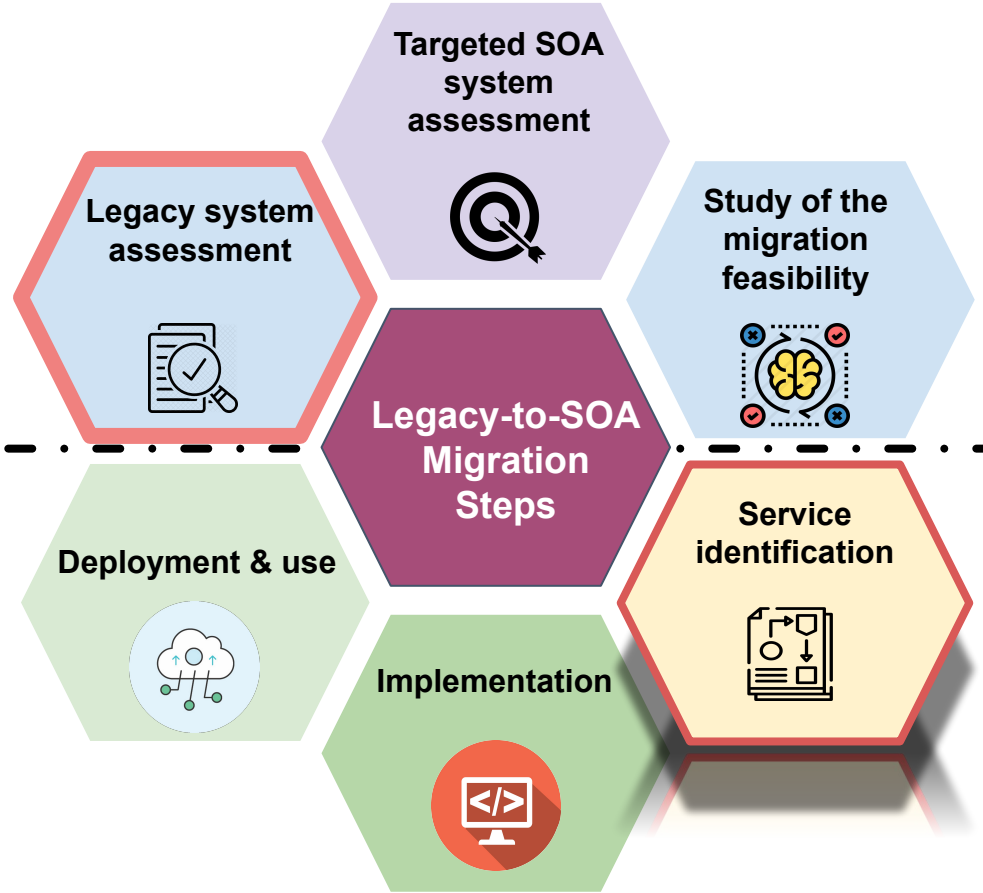
Implementation



Migration steps

Planification of the migration

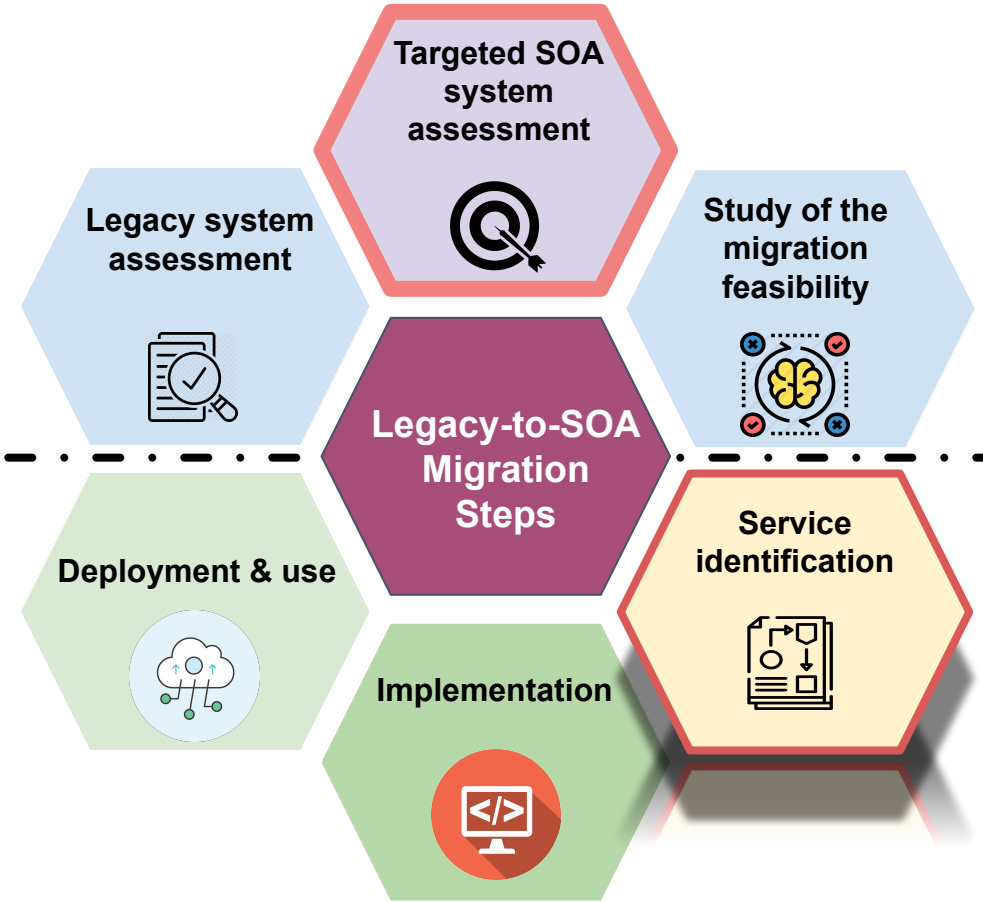
Implementation



Migration steps

Planification of the migration

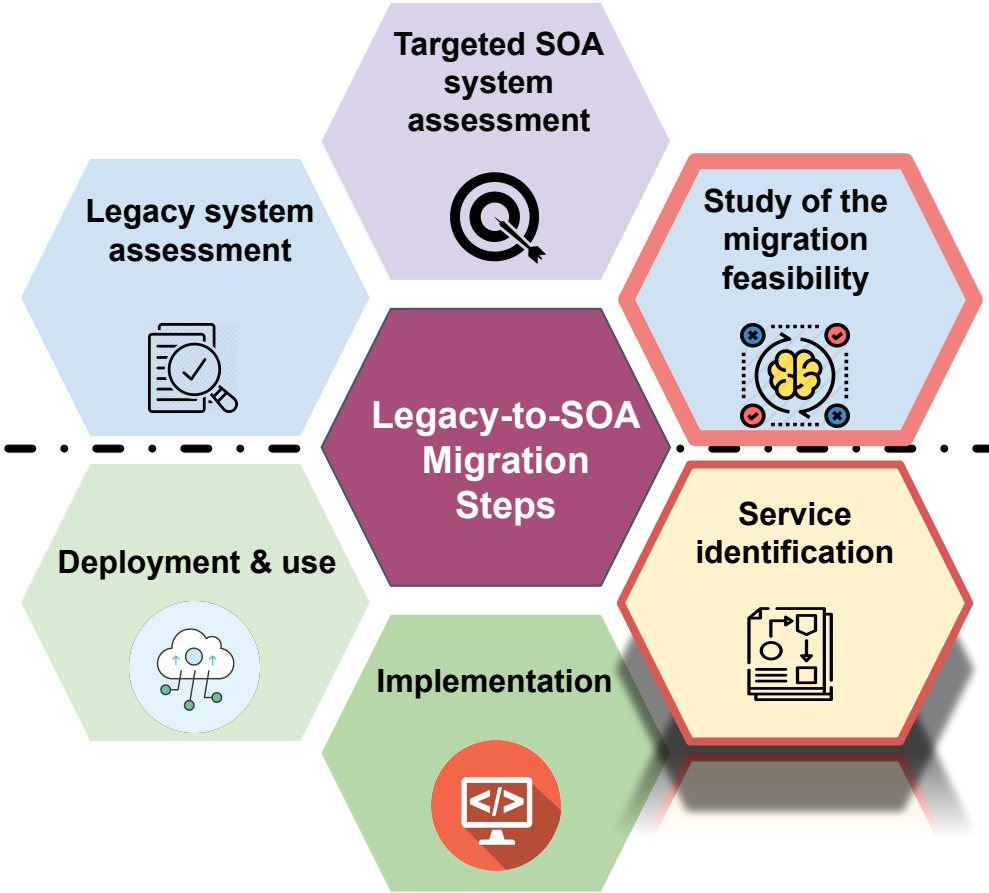
Implementation



Migration steps

Planification of the migration

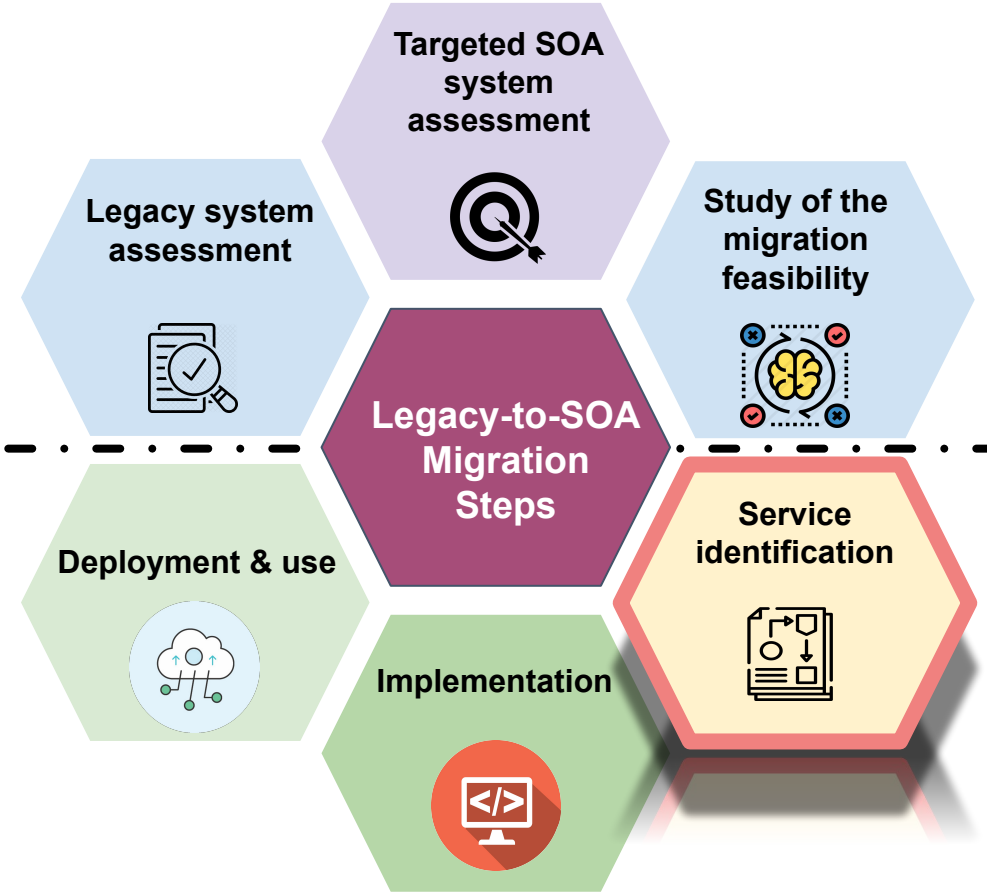
Implementation



Migration steps

Planification of the migration

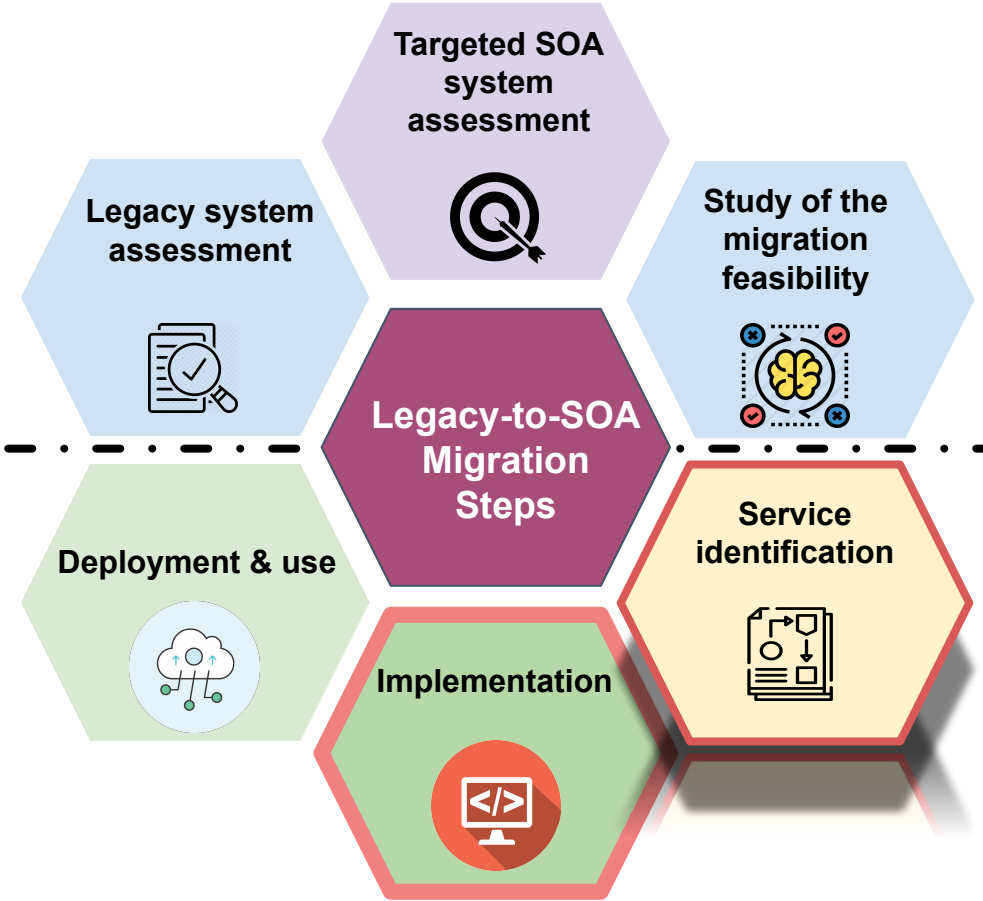
Implementation



Migration steps

Planification of the migration

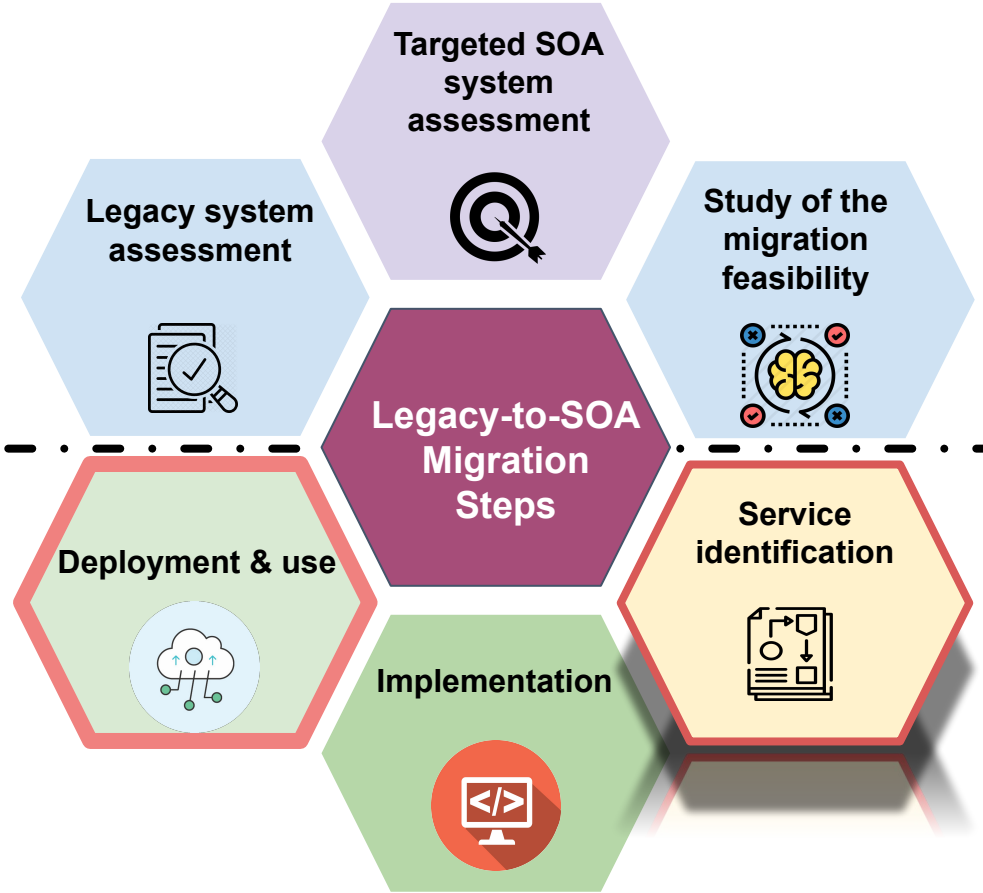
Implementation



Migration steps

Planification of the migration

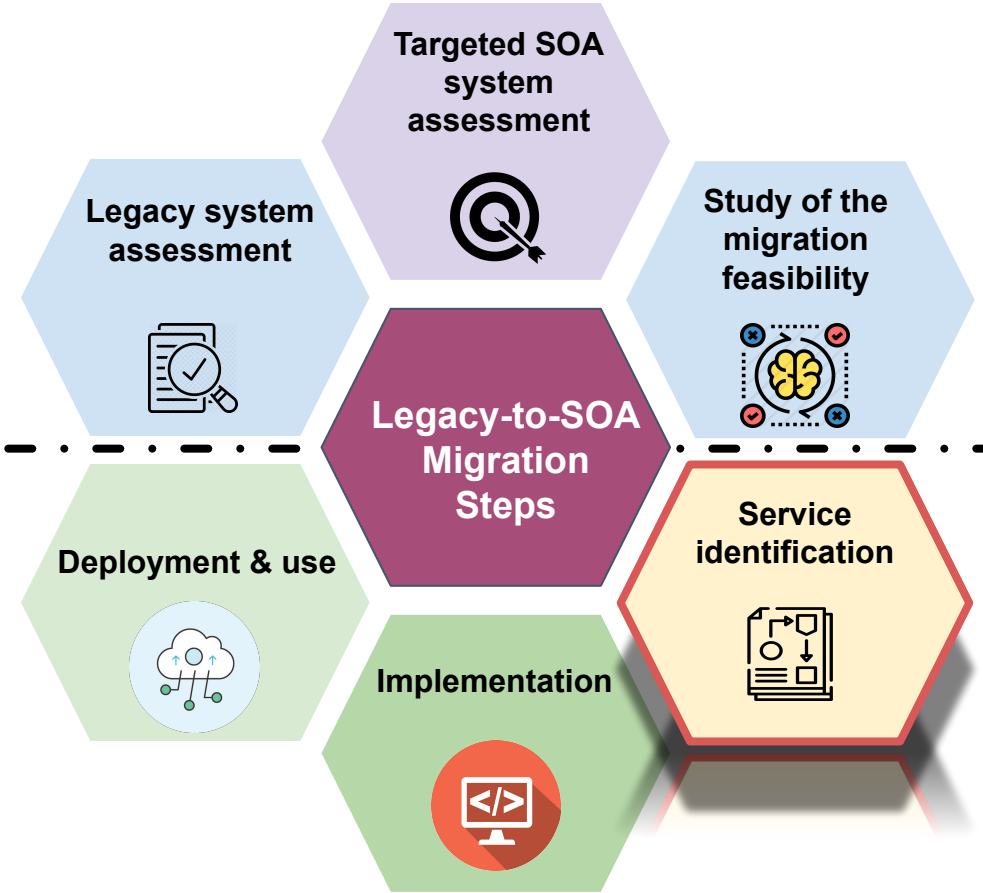
Implementation



Migration steps

Planification of the migration

Implementation



Problems

Service Identification

Is the most challenging phase of the overall modernization process.
How to promote software reuse and avoid development from scratch?

Gap between academia and industry
Several SI approaches have been proposed in the literature.

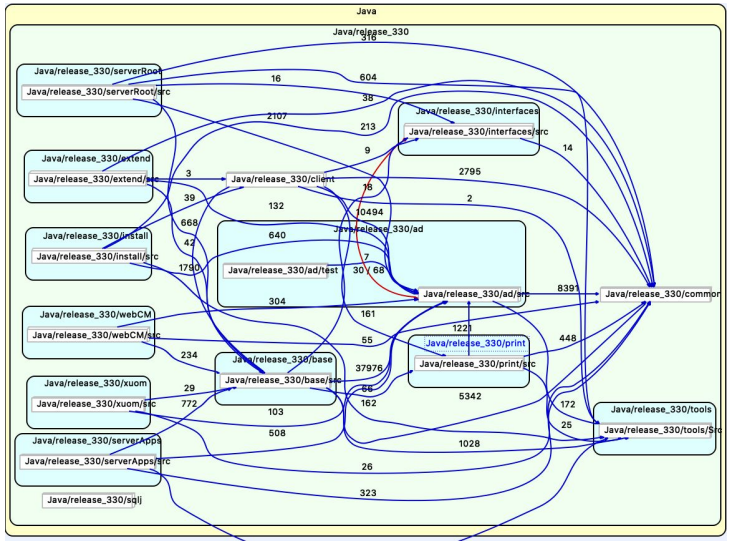
Identification accuracy
Most of them have limited identification accuracy.

Input types & availability
They usually require several types of inputs.

Lack of automation
Most of the existing approaches are semi-automated or manual.

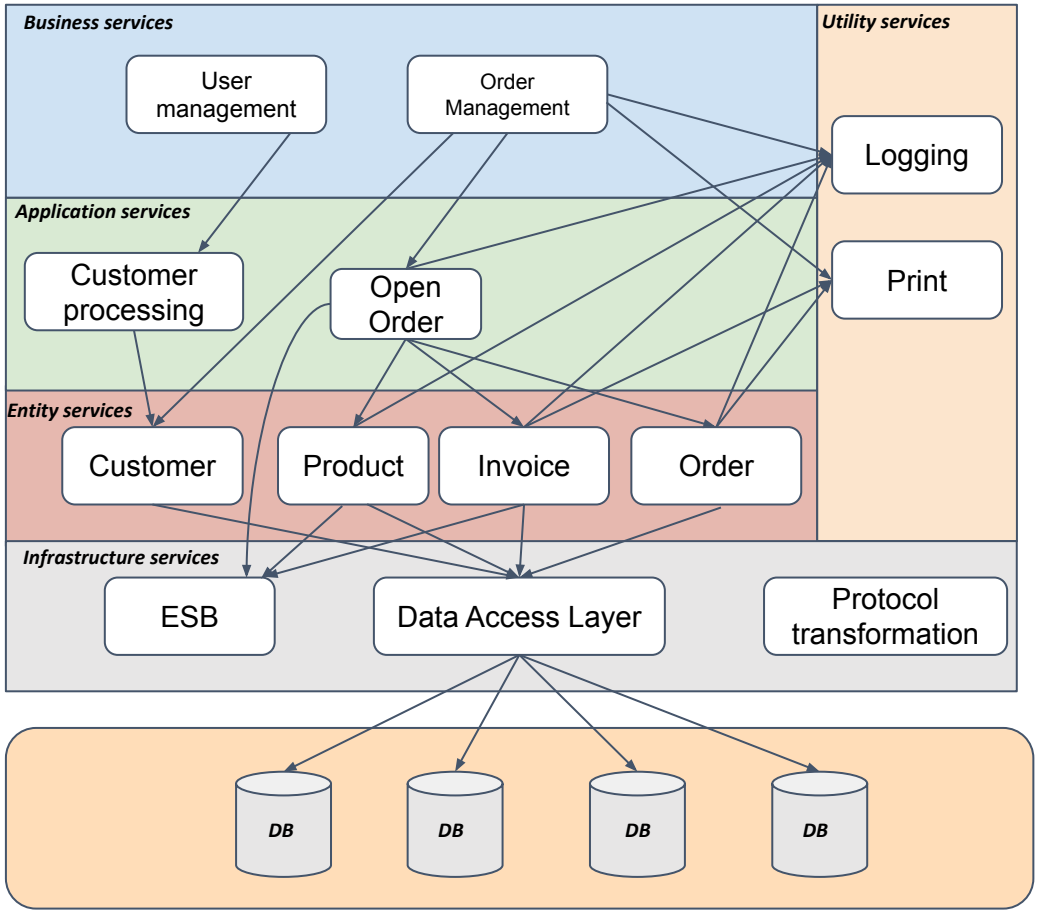
Lack of type-sensitive SIAs
Lack of the identification of specific types of services.

SI should depend on service types



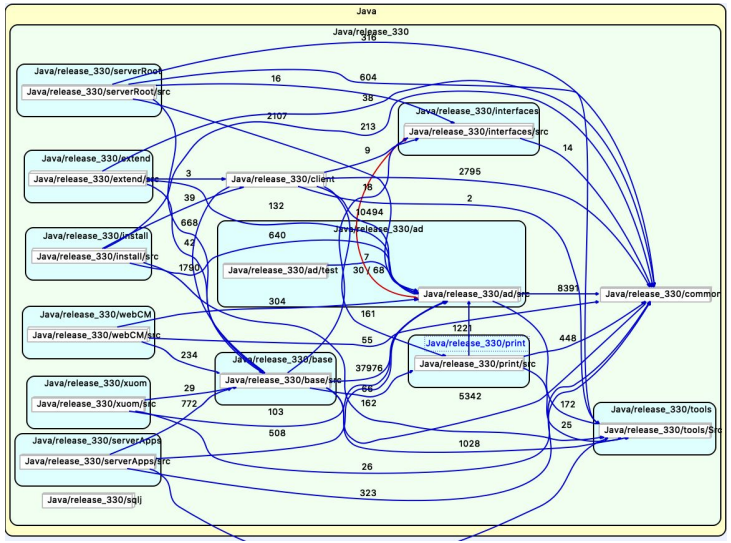
Legacy ERP

Migration



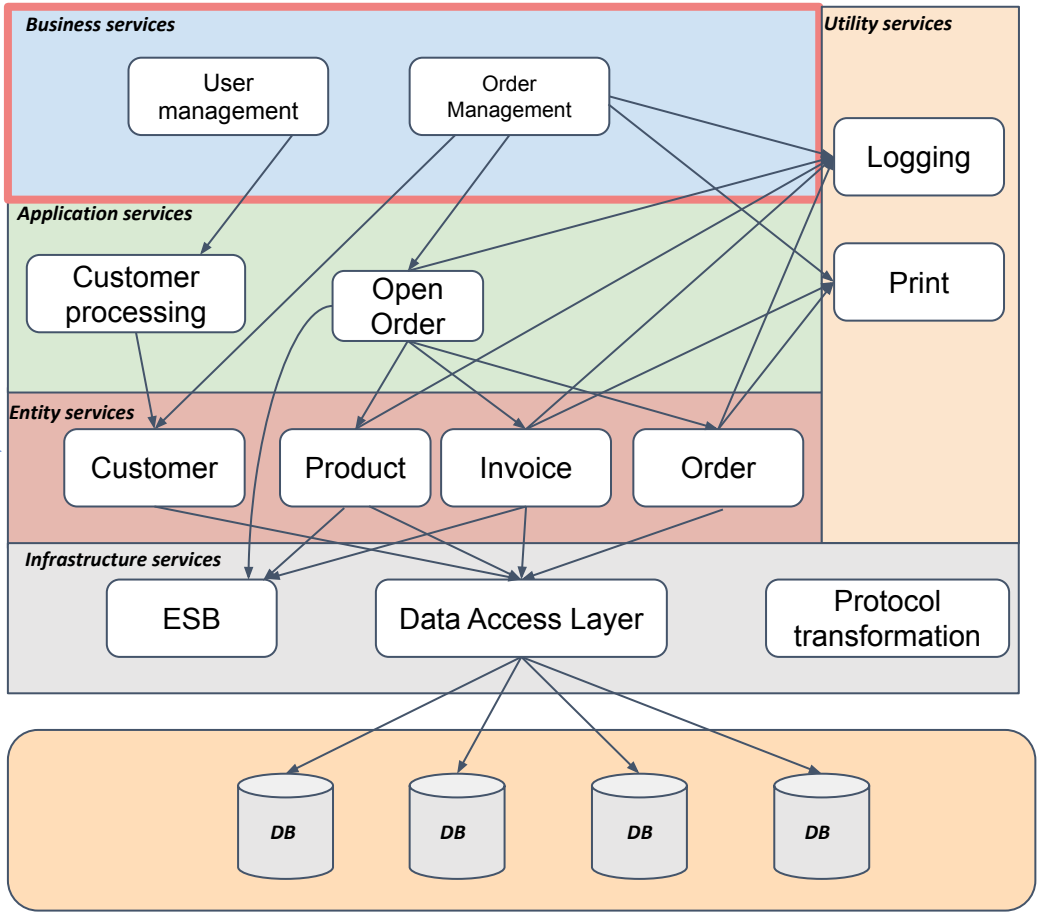
SOA-based ERP

SI should depend on service types



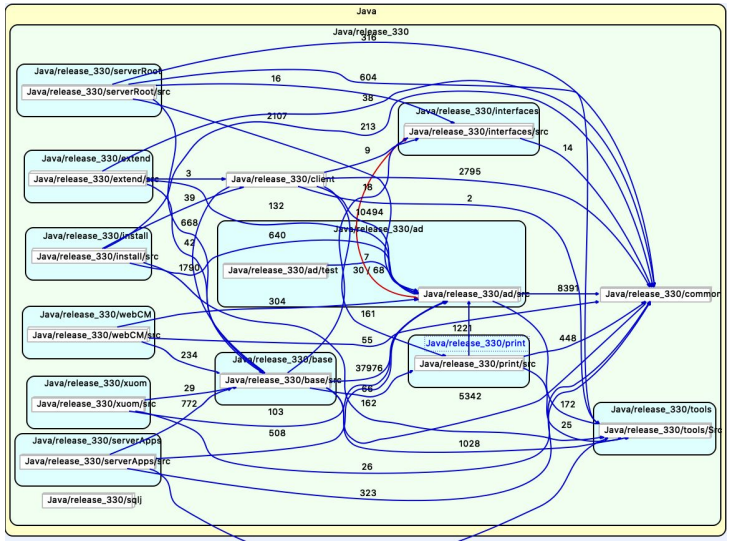
Legacy ERP

Migration



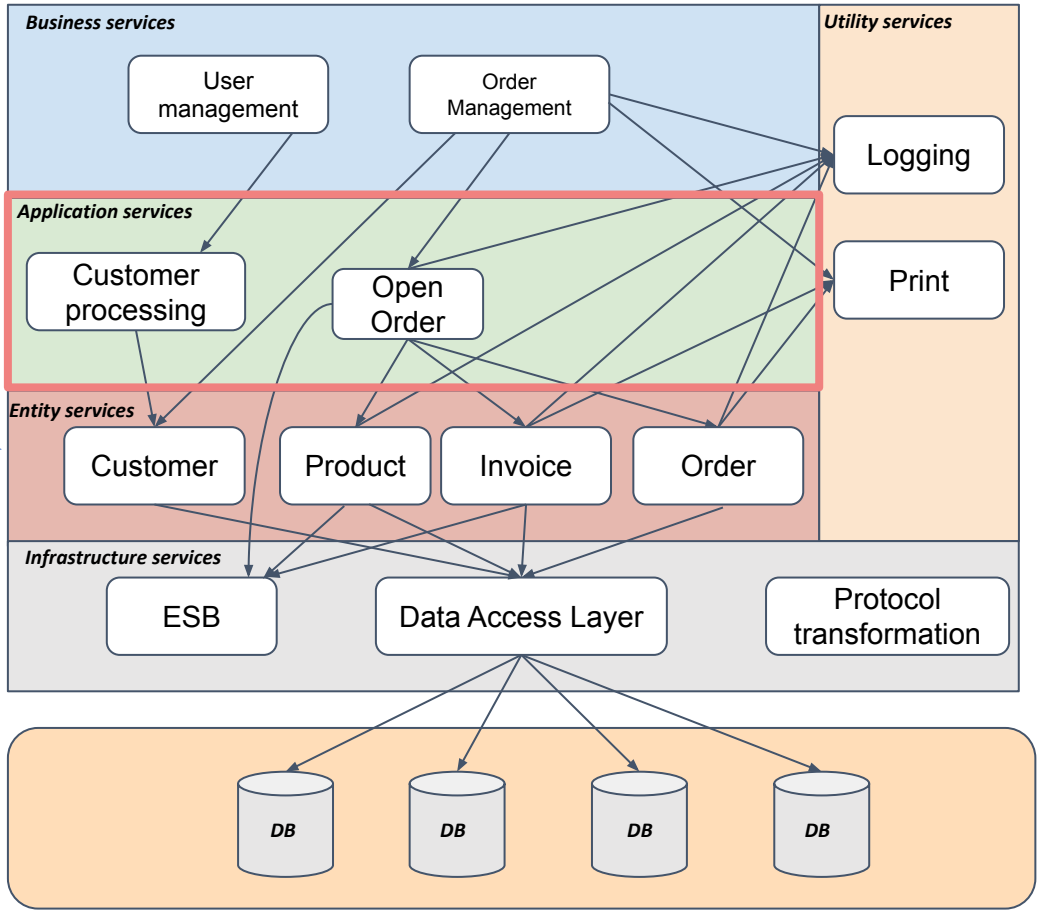
SOA-based ERP

SI should depend on service types



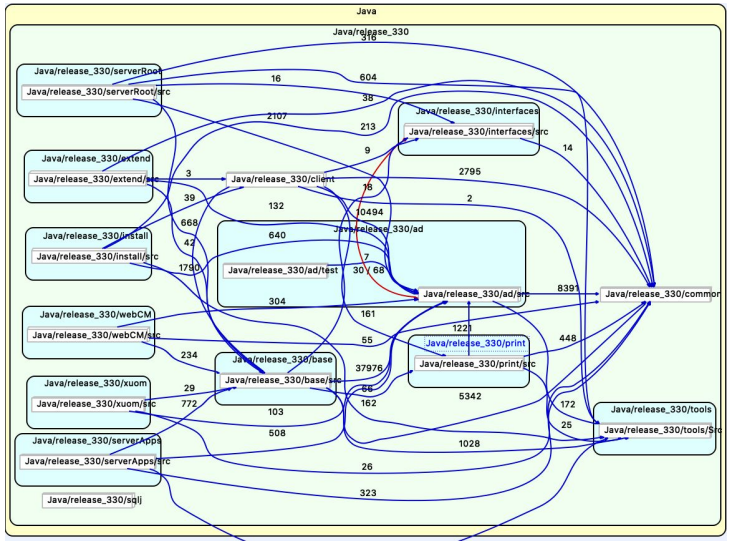
Legacy ERP

Migration



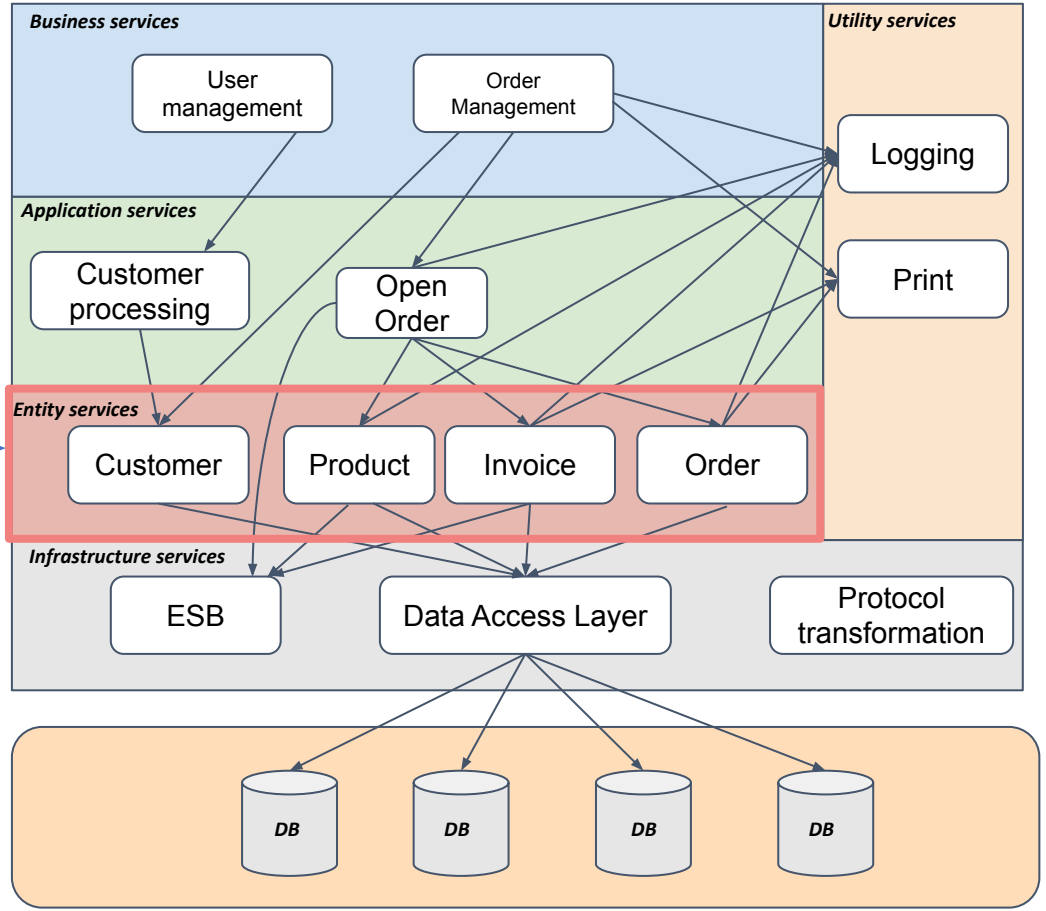
SOA-based ERP

SI should depend on service types



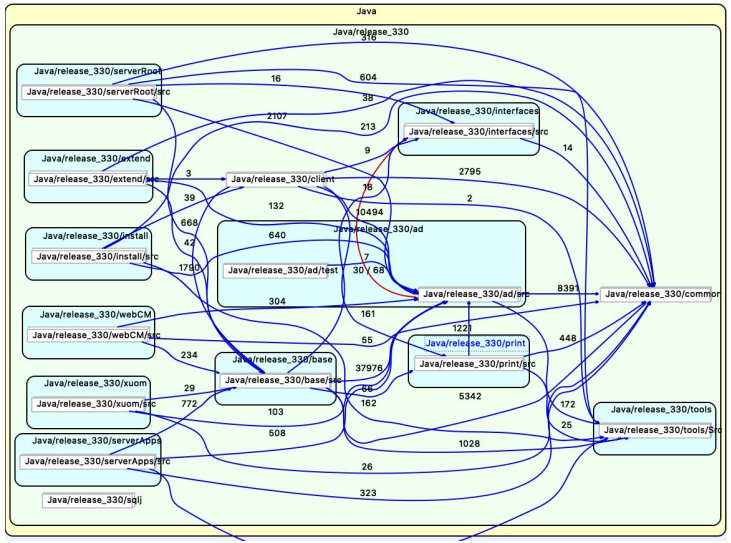
Legacy ERP

Migration



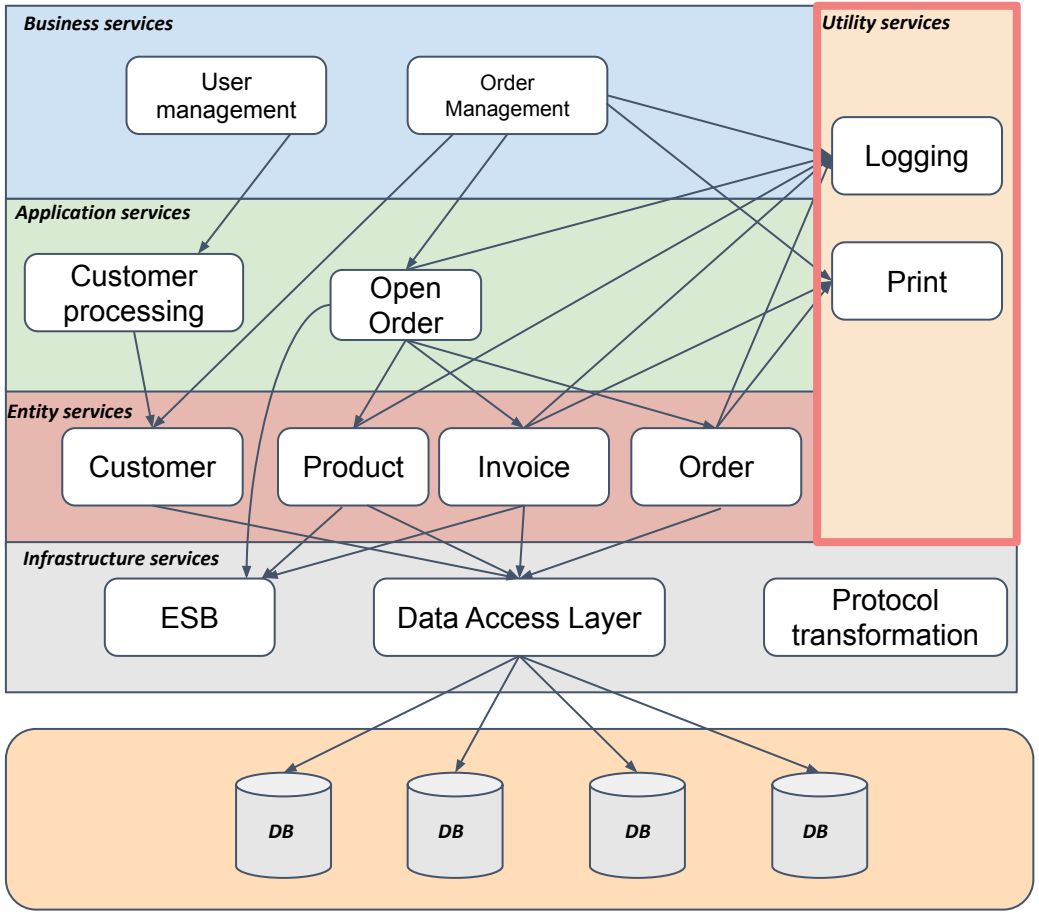
SOA-based ERP

SI should depend on service types



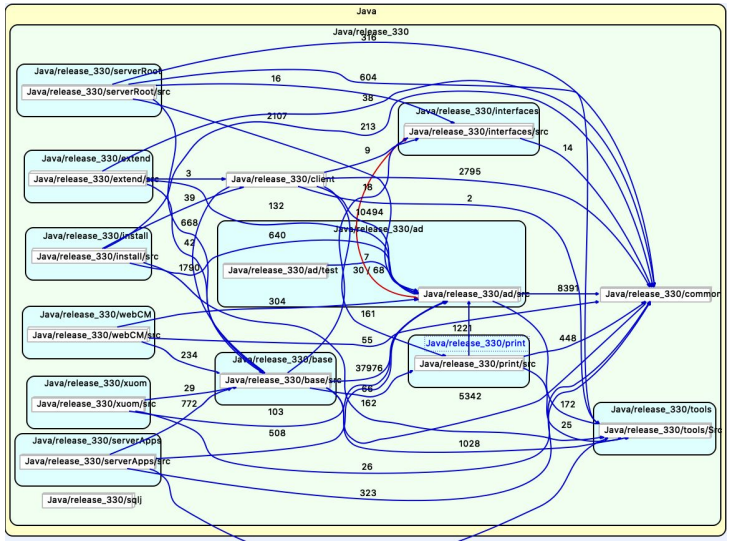
Legacy ERP

Migration



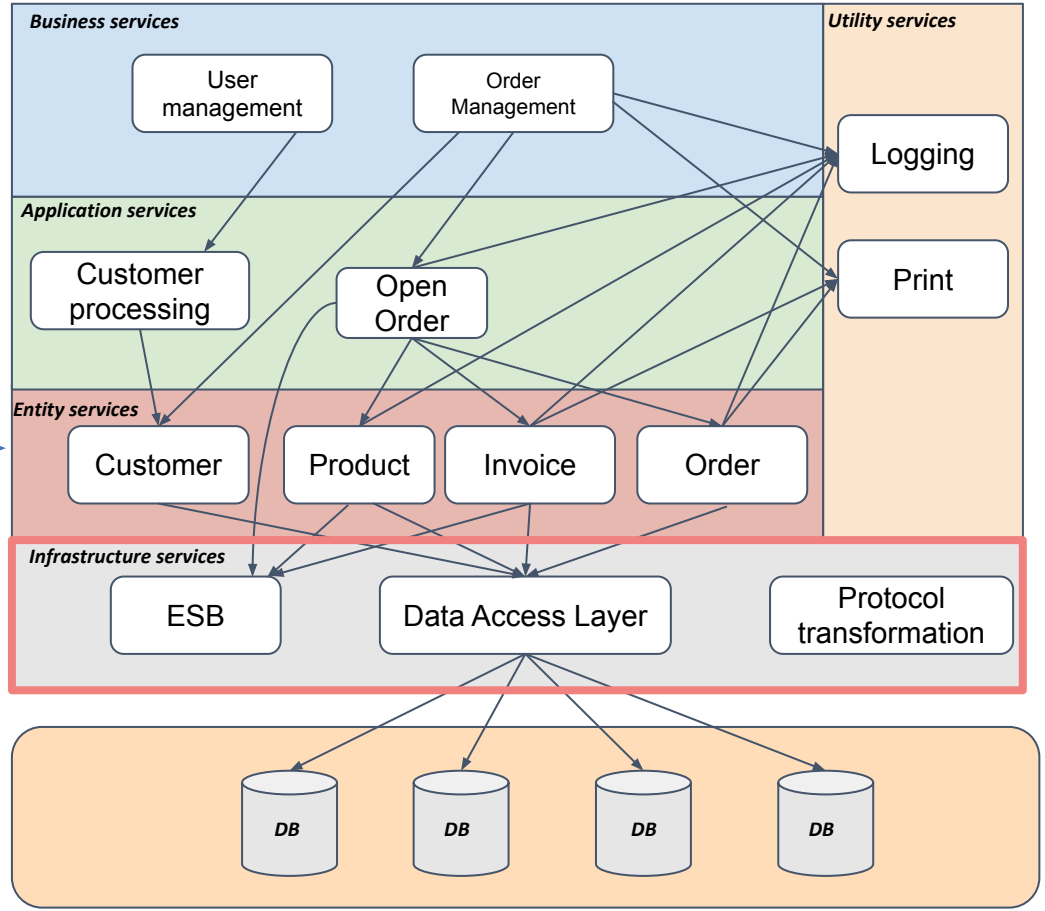
SOA-based ERP

SI should depend on service types



Legacy ERP

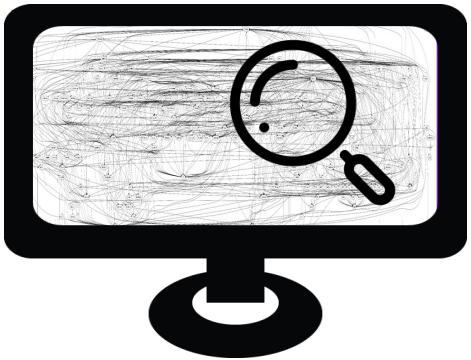
Migration



SOA-based ERP

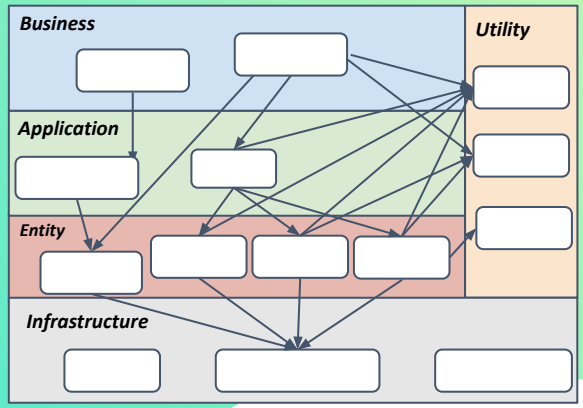
Thesis statement

Supporting legacy-to-SOA migration with a **highly automated type-sensitive SI** approach that relies on **static analysis** of legacy systems.



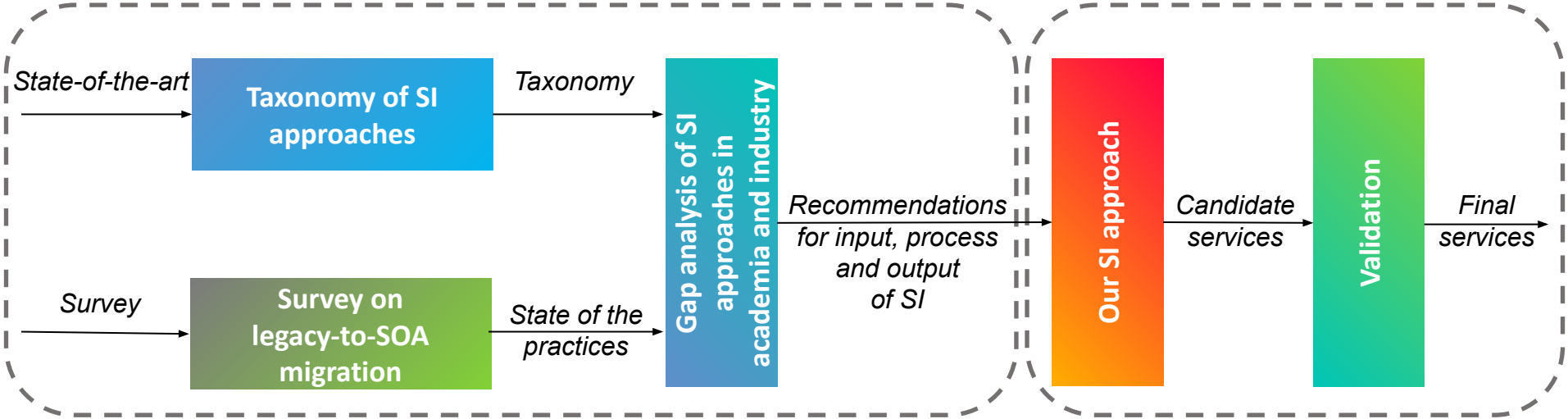
Legacy System

Service Identification



SOA-based System

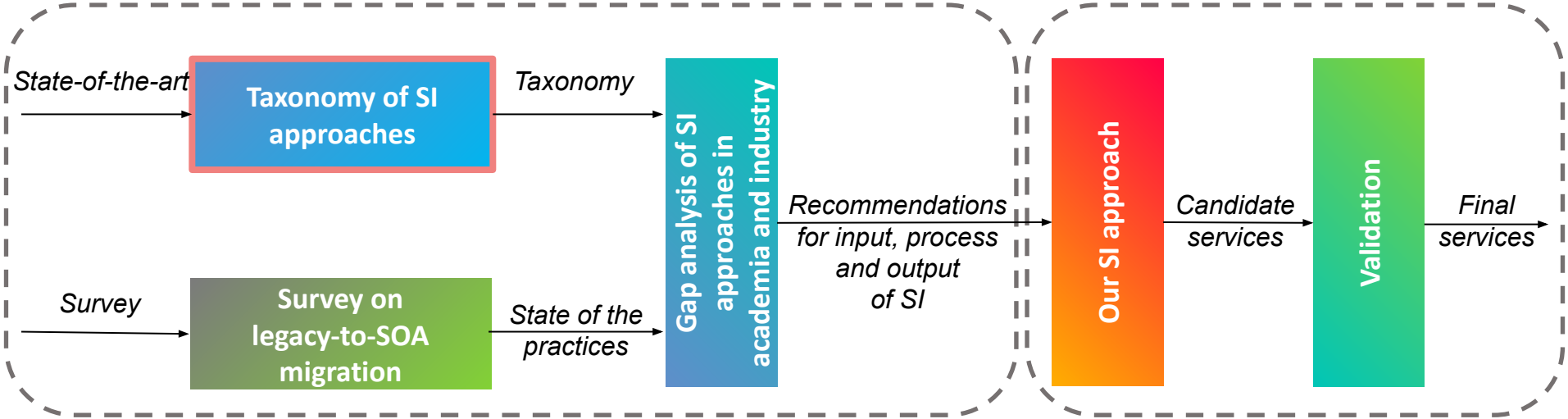
Research methodology



Step1: Gap analysis of SI approaches in academia and industry

Step 2: Type-sensitive service identification

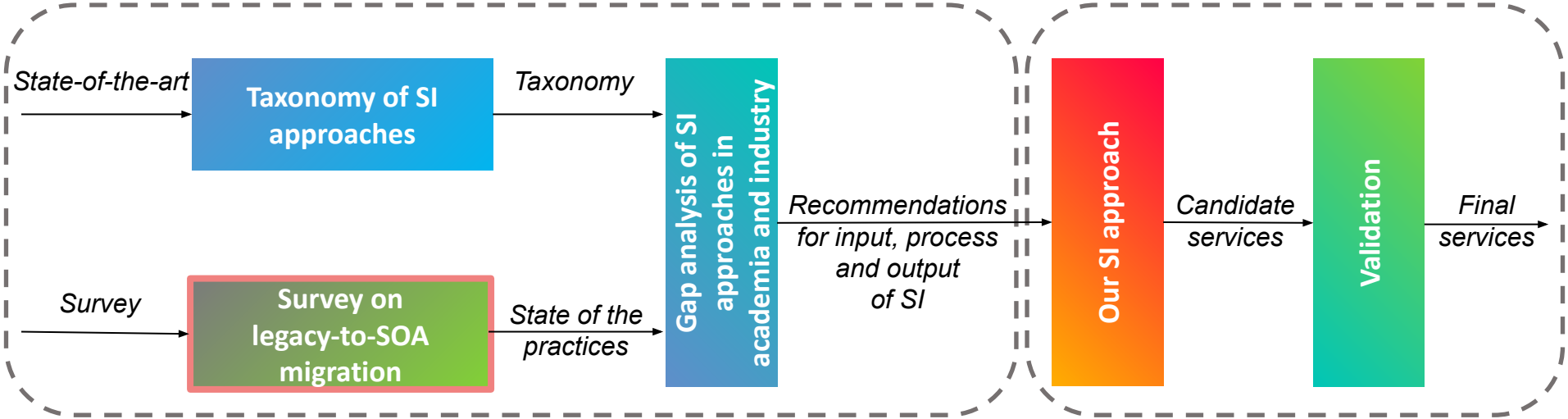
Research methodology



Step1: Gap analysis of SI approaches in academia and industry

Step 2: Type-sensitive service identification

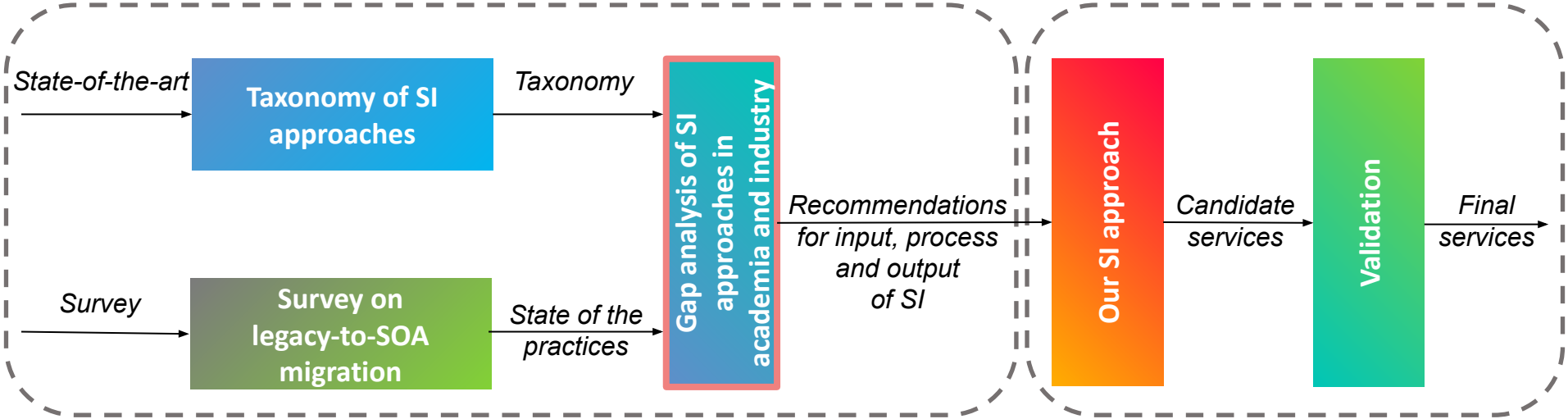
Research methodology



Step1: Gap analysis of SI approaches in academia and industry

Step 2: Type-sensitive service identification

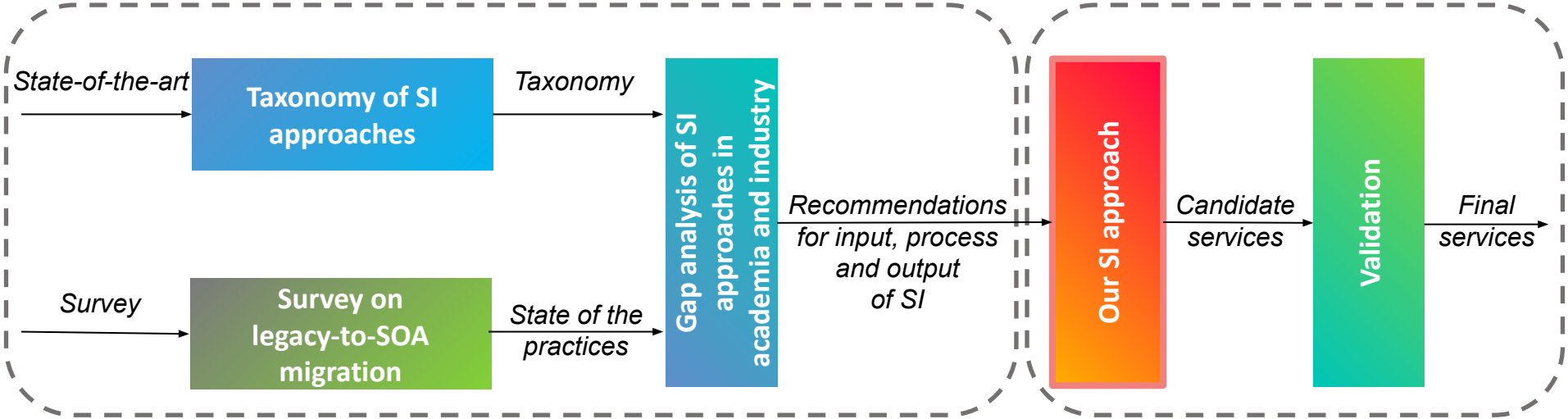
Research methodology



Step1: Gap analysis of SI approaches in academia and industry

Step 2: Type-sensitive service identification

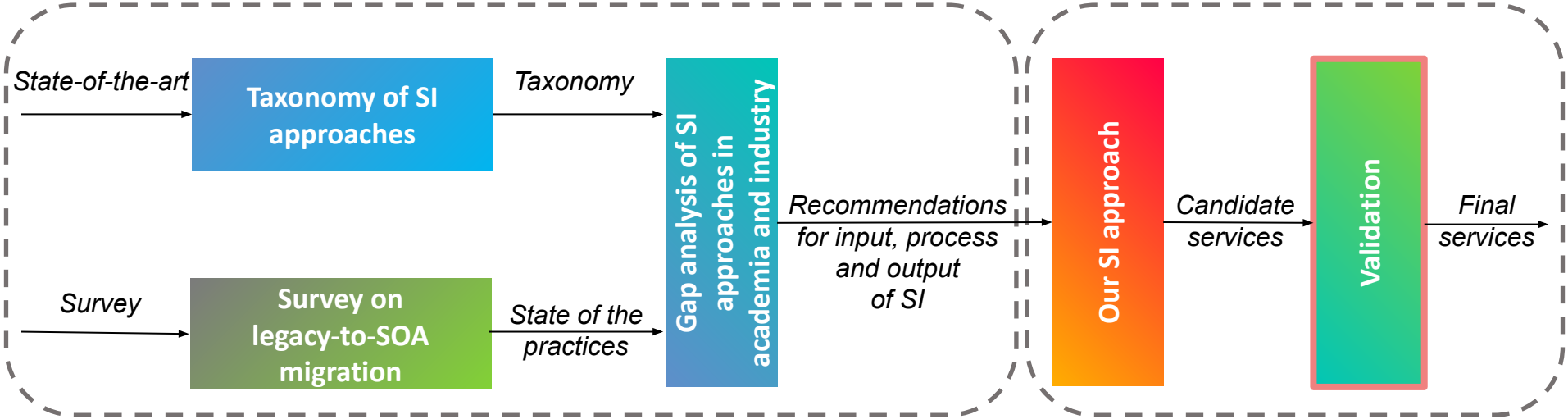
Research methodology



Step1: Gap analysis of SI approaches in academia and industry

Step 2: Type-sensitive service identification

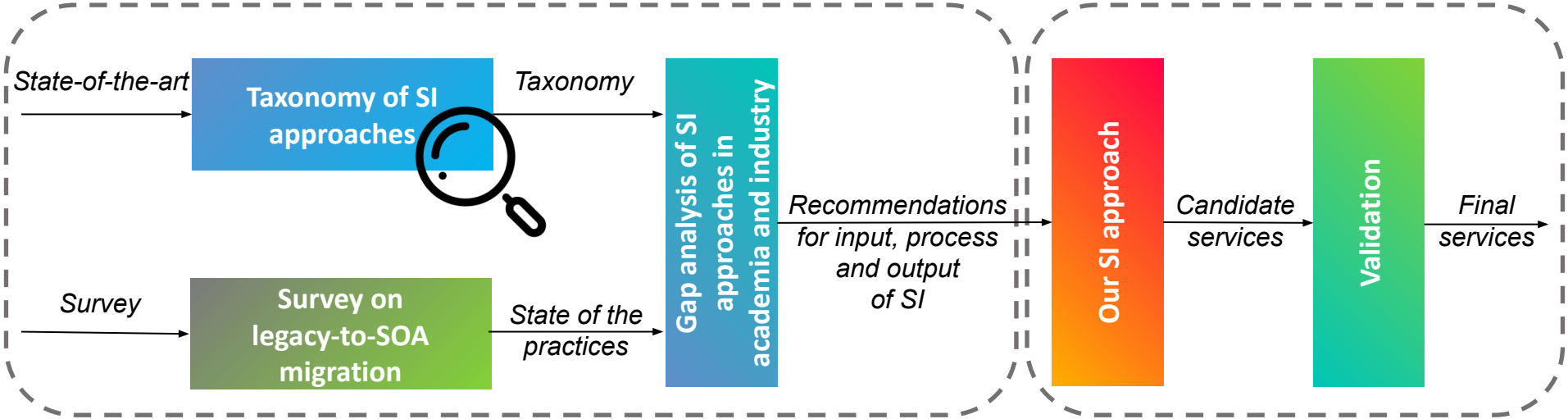
Research methodology



Step1: Gap analysis of SI approaches in academia and industry

Step 2: Type-sensitive service identification

Research methodology



Step1: Gap analysis of SI approaches in academia and industry

Step 2: Type-sensitive service identification

[Abdellatif et al., JSS 2021]

Taxonomy of service identification approaches

1. **Search query**

(Service identification OR service mining OR service packaging) AND (migration OR modernization OR transformation OR re-engineering) AND (legacy OR existing systems OR Object-Oriented)

2. **Execution of the search query on several databases (3246 papers)**

ACM DL DIGITAL LIBRARY

WEB OF SCIENCE™

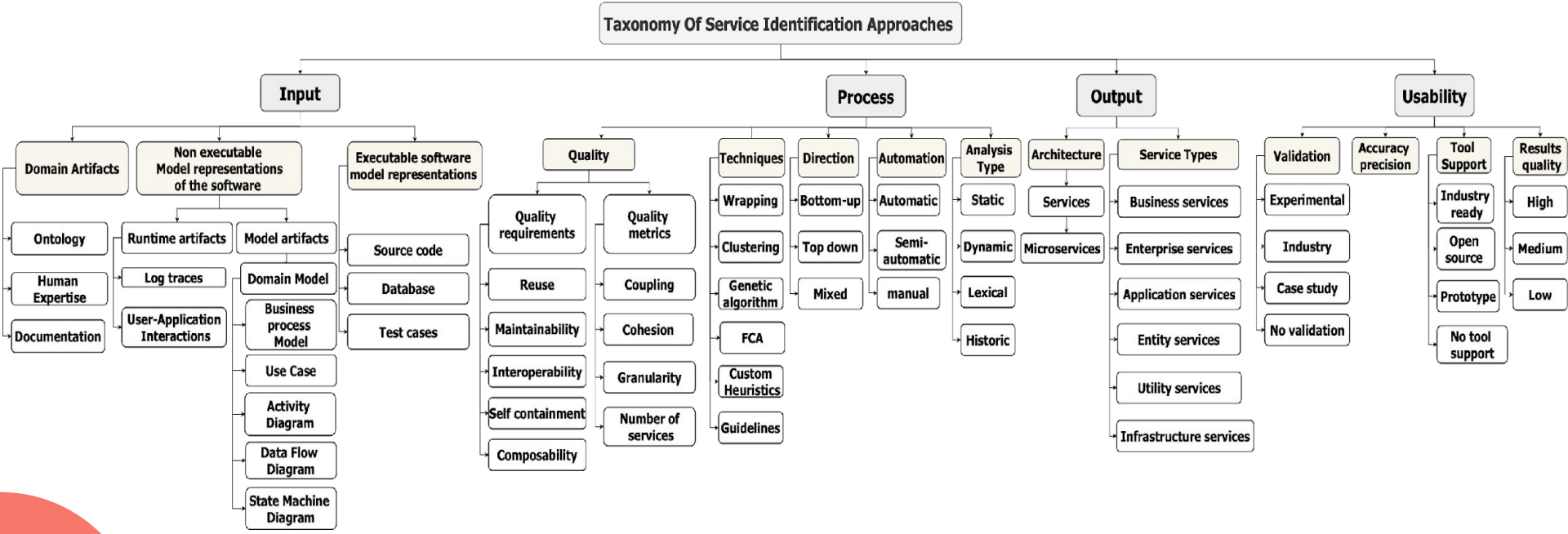
Compendex on Engineering Village

IEEE Xplore® Digital Library

3. **Filtering process**

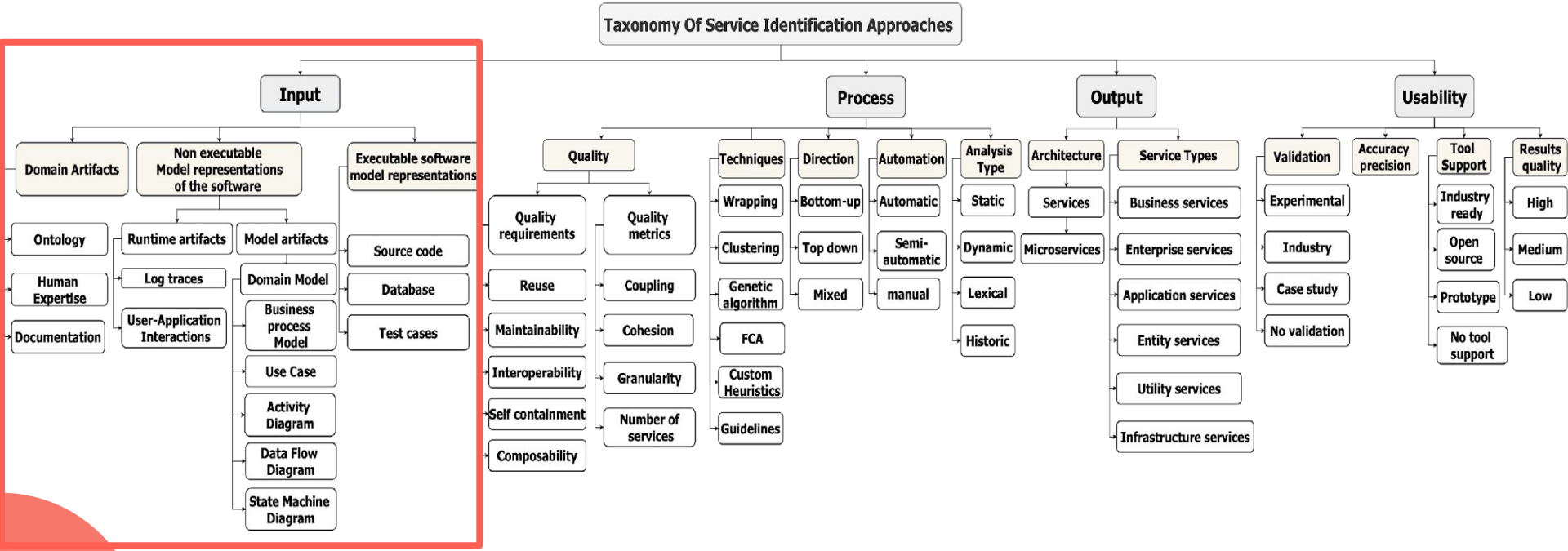
4. **Backward and forward snowballing (9 iterations)** → **Final 41 papers**

Taxonomy Of Service Identification Approaches



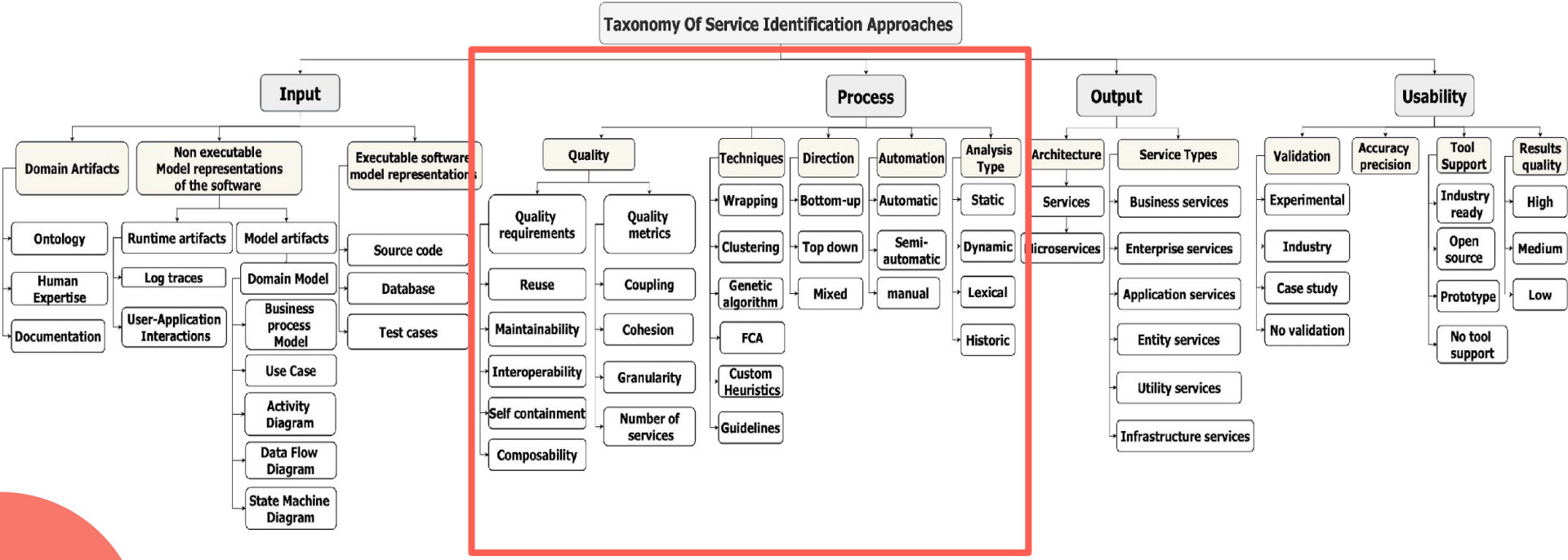
(Abdellatif et al., JSS 2021)

Taxonomy Of Service Identification Approaches



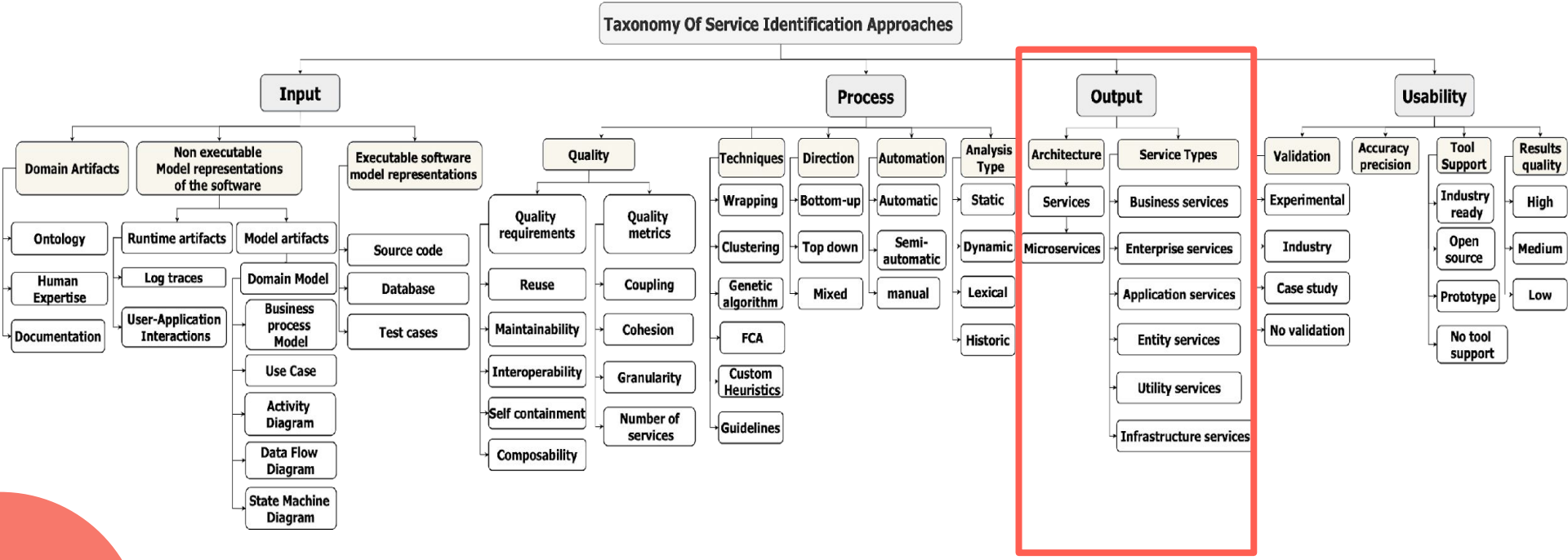
(Abdellatif et al., JSS 2021)

Taxonomy Of Service Identification Approaches



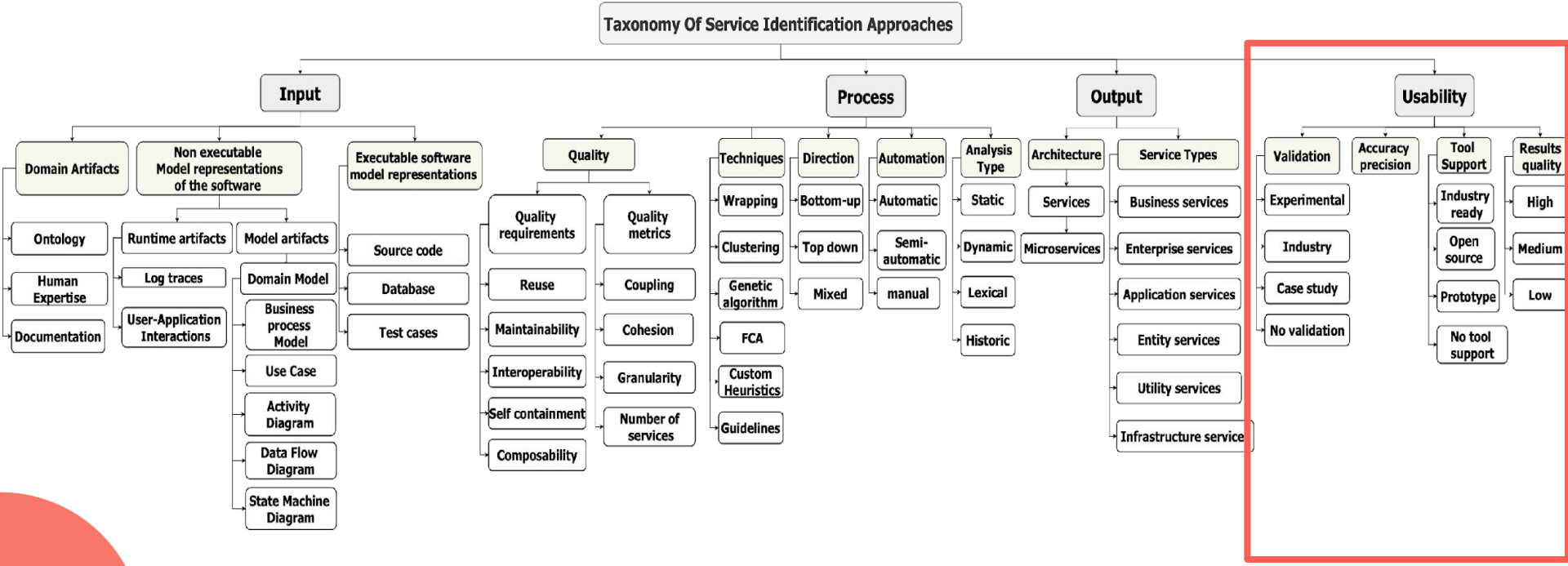
(Abdellatif et al., JSS 2021)

Taxonomy Of Service Identification Approaches



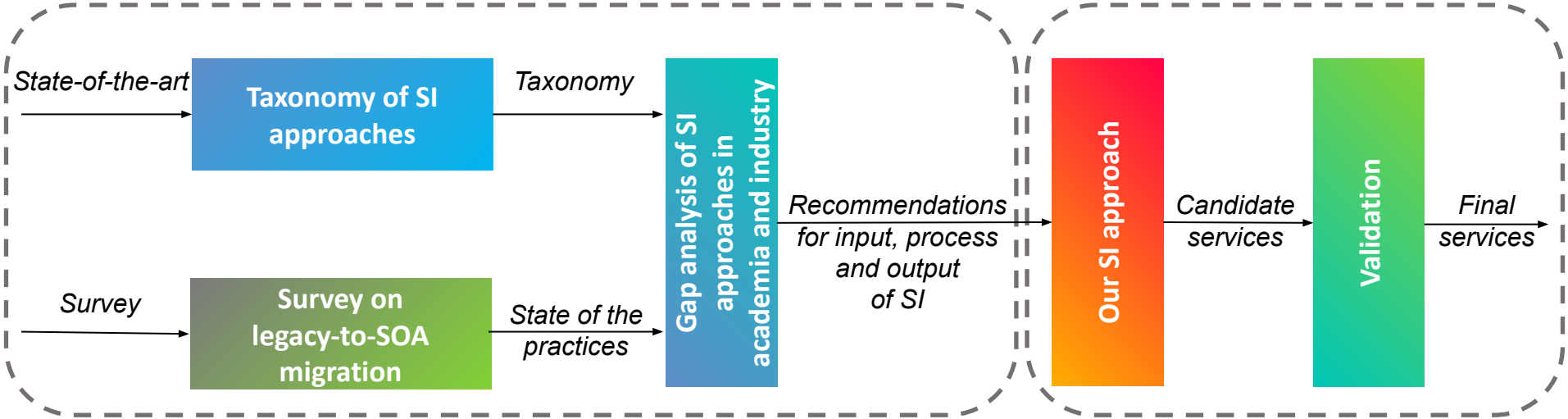
(Abdellatif et al., JSS 2021)

Taxonomy Of Service Identification Approaches



(Abdellatif et al., JSS 2021)

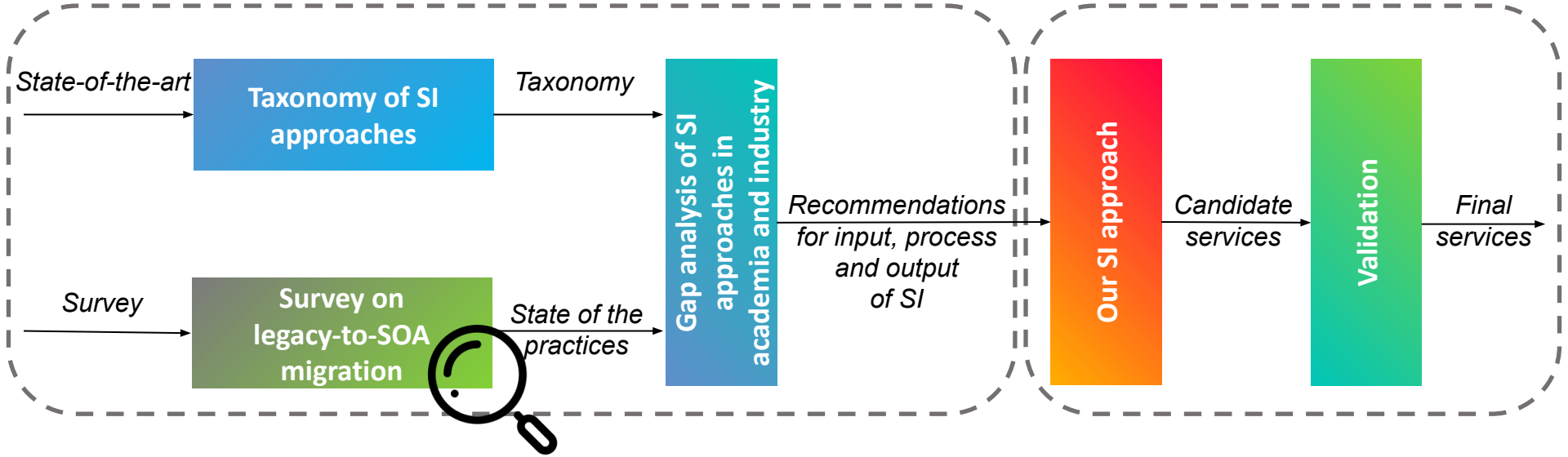
Research methodology



Step1: Gap analysis of SI approaches in academia and industry

Step 2: Type-sensitive service identification

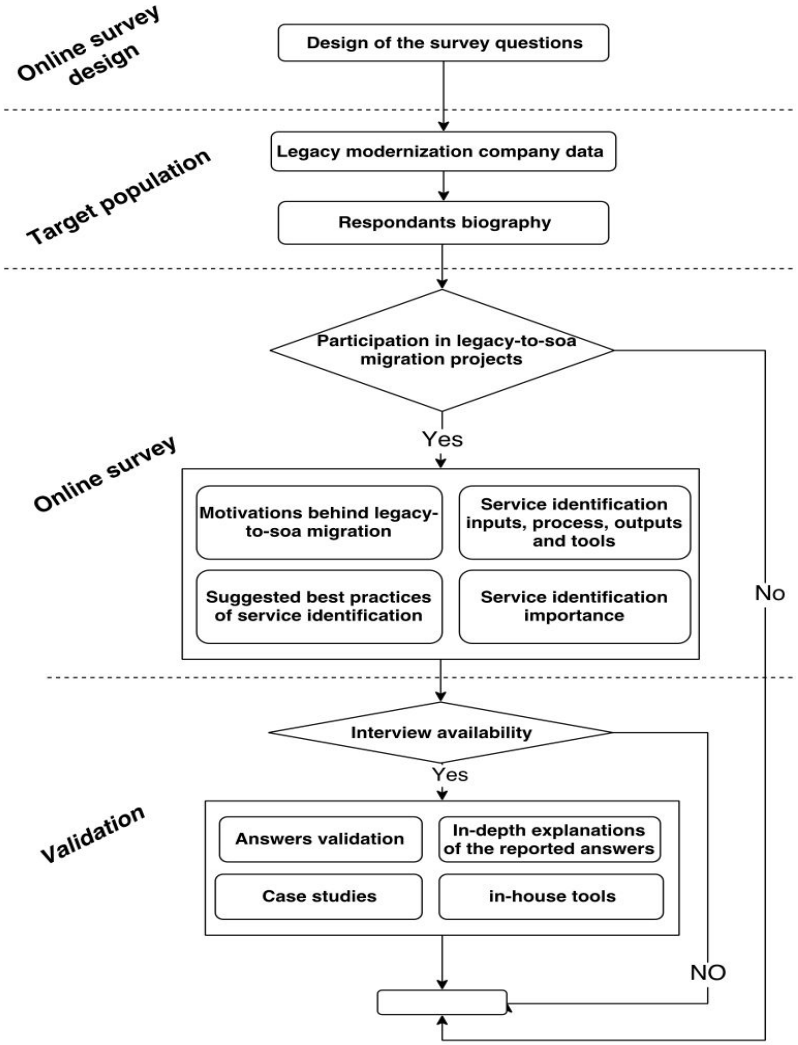
Research methodology



Step1: Gap analysis of SI approaches in academia and industry

Step 2: Type-sensitive service identification

(Abdellatif et al., ICSOC 2018)


Online survey design
Design of the survey questions
Target population
Legacy modernization company data
Respondants biography
Participation in legacy-to-soa migration projects
Yes
No
Online survey
Motivations behind legacy-to-soa migration
Service identification inputs, process, outputs and tools
Suggested best practices of service identification
Service identification importance
Interview availability
Yes
Validation
Answers validation
In-depth explanations of the reported answers
Case studies
in-house tools
NO

Collection of potential participants

Publication of the survey

Target companies for legacy modernization

Rely on LinkedIn, Facebook, And Twitter

Voluntary participation (no incentives)

We reached around 298 practitioners

45 participants fully completed the survey + 8 interviews

Companies


IBM


accenture


Hydro
Québec


hp


benchmark

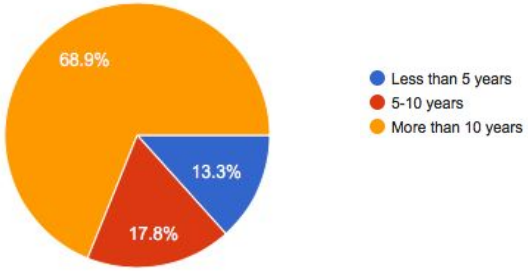

FRESCHE
LEGACY


Infosys

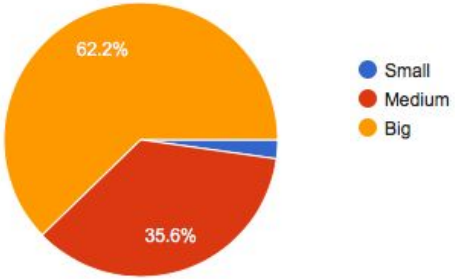

NTT DATA


TATA
TATA CONSULTANCY SERVICES

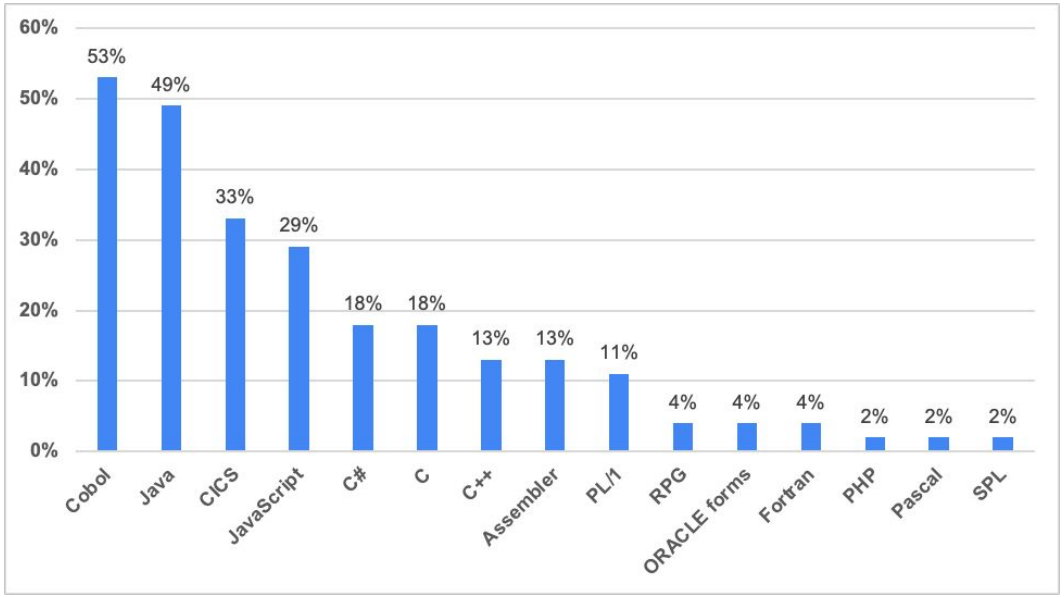
Types of the migrated systems



(a) Age of the migrated systems



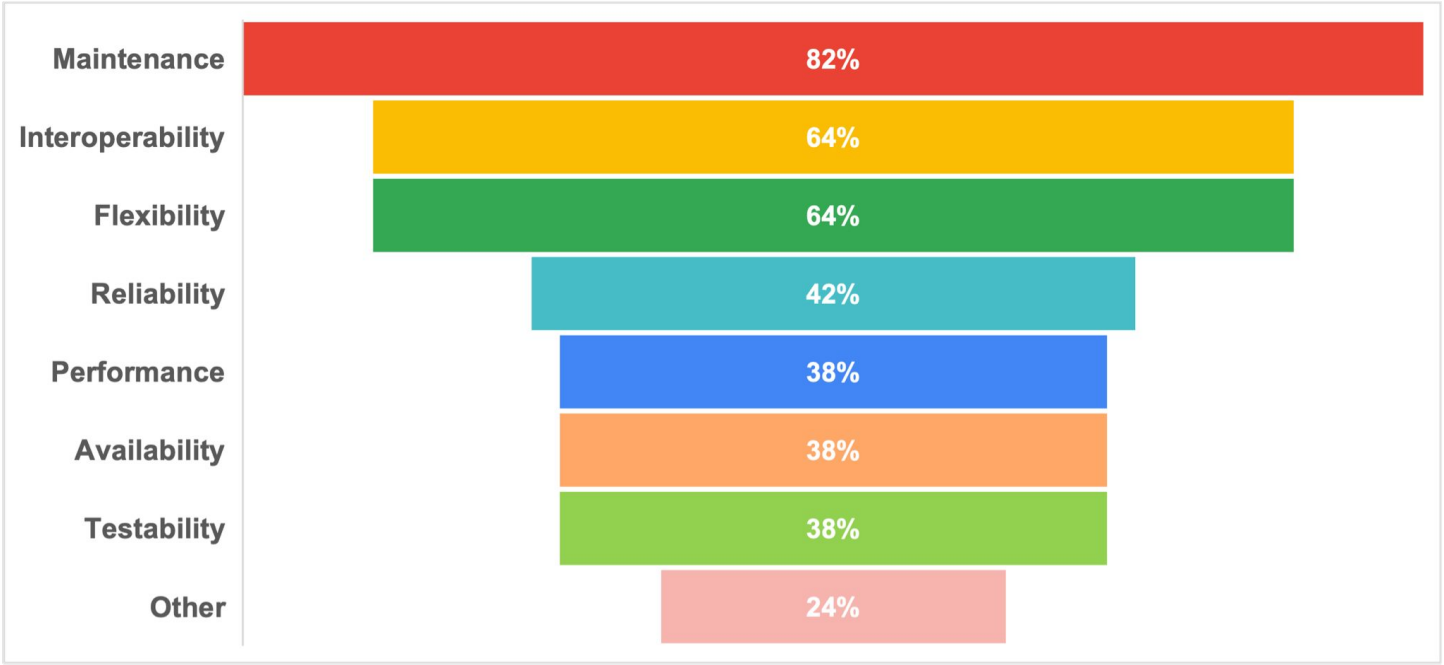
(b) Size of the migrated systems



(c) Programming Languages of migrated systems

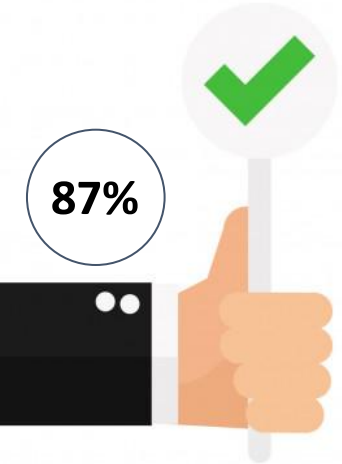
*Practitioners migrate **different types** of old legacy systems implemented mainly in **Cobol** and **Java.***

Motivations for legacy-to-SOA migration

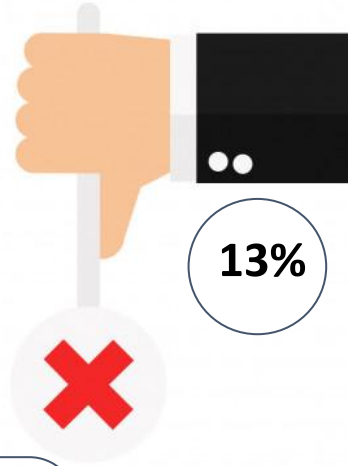


*Reducing **maintenance costs**, improving the **flexibility** and **interoperability** of legacy systems are the main **motivations** to migrate legacy systems to SOA.*

Do you think that service identification from legacy systems is important for legacy-to-SOA migration?

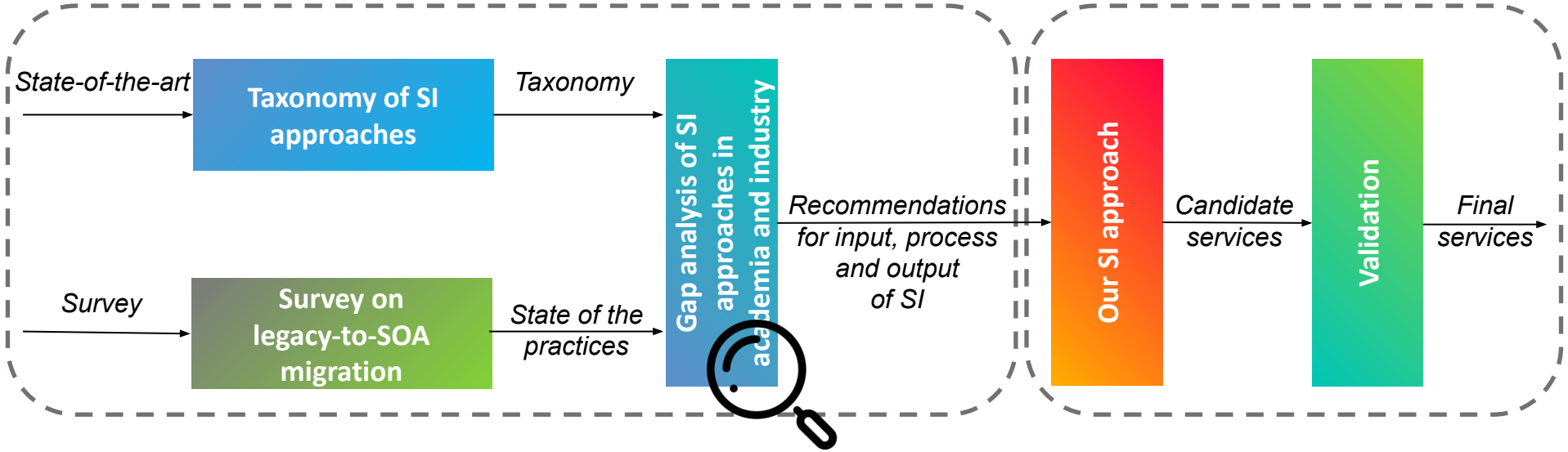


- Reduce maintenance cost
- Reduce software development cost
- Increases software productivity by shortening software development time
- Benefits of software reuse



- Source code not reusable
- Top-down approach
- Not suitable for integration problems

Research methodology

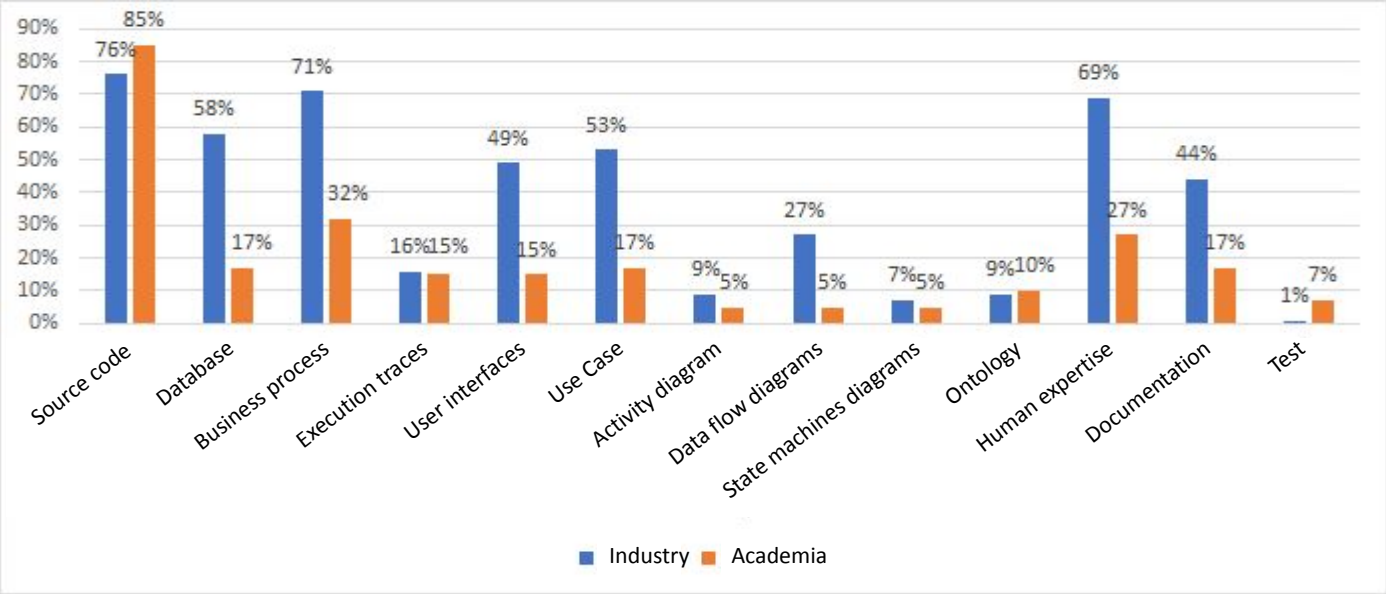


Step1: Gap analysis of SI approaches in academia and industry

Step 2: Type-sensitive service identification

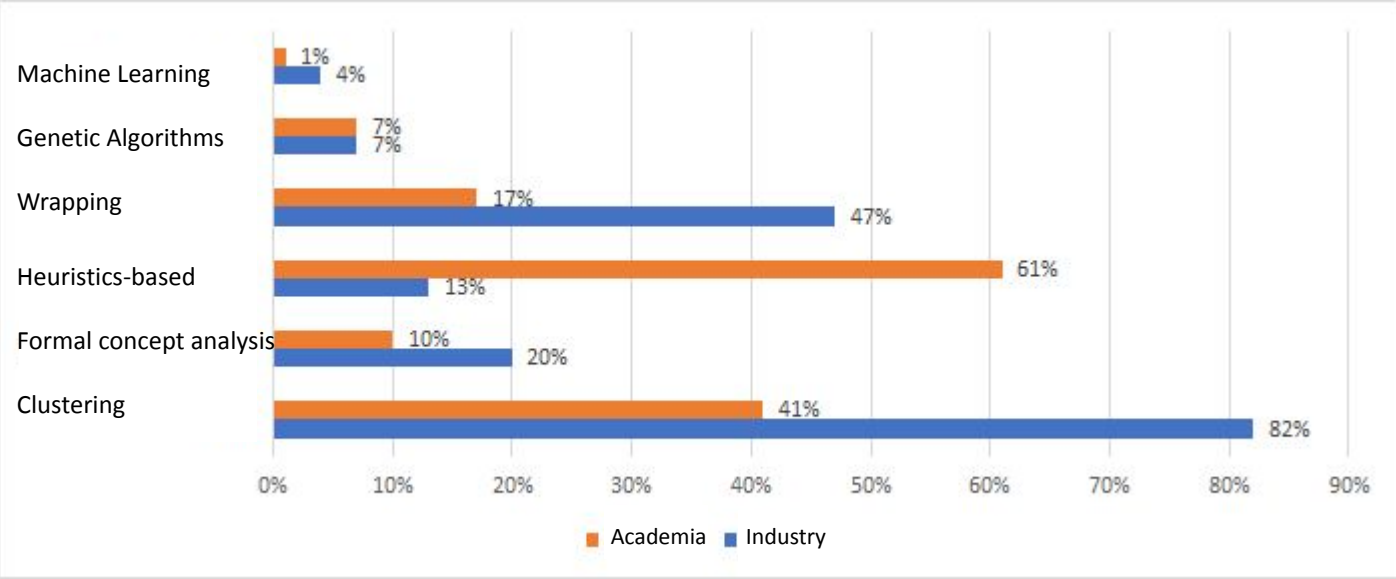
[Abdellatif et al., TSE 2021 - Under review]

Inputs used for service identification



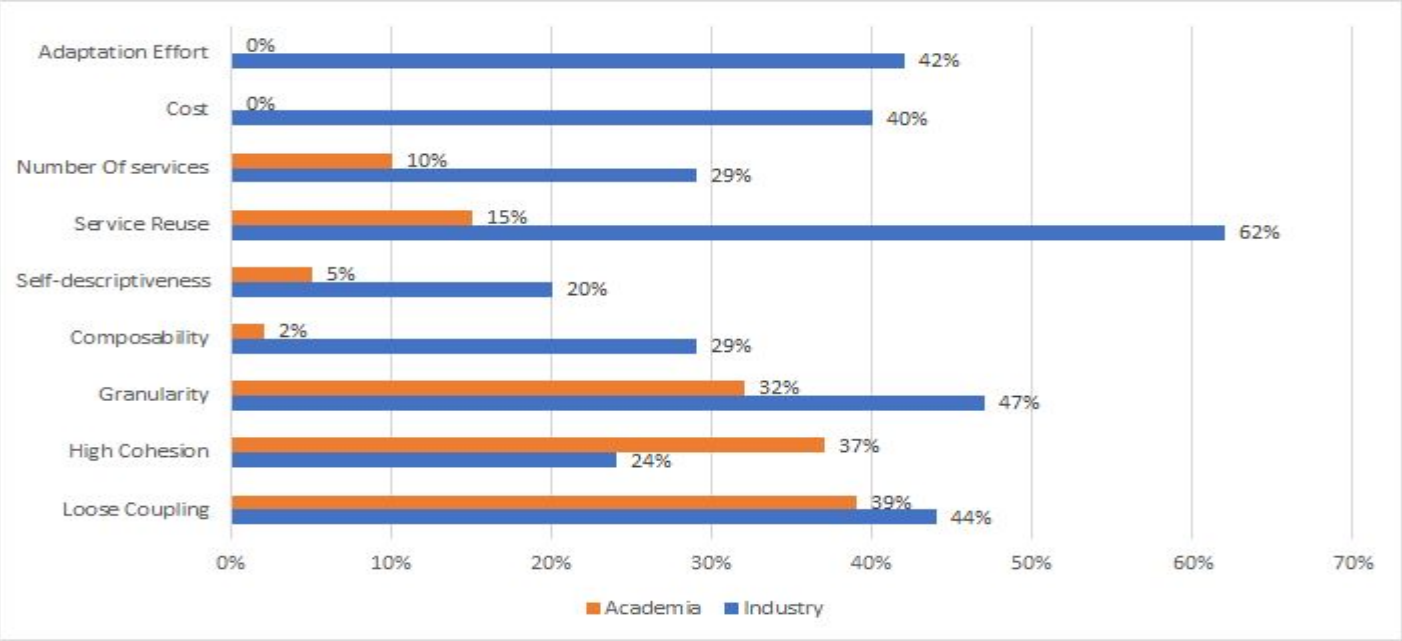
- *The most used inputs by industrial and academic SIAs are source code, business process, and human expertise.*
- *Ontologies, activity diagrams, state machines, and execution traces are seldom used to identify services.*

Techniques used for service identification



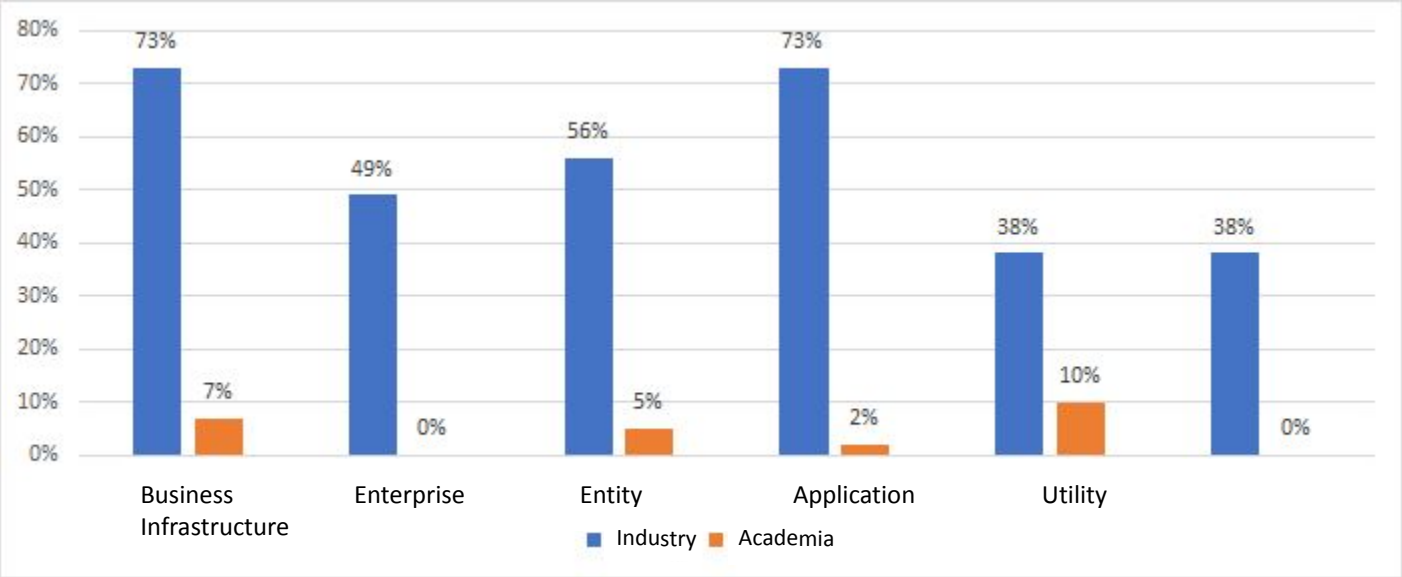
- ***Clustering*** *is the technique most used by academia and industry to identify services.*
- *In second place, practitioners favour* ***wrapping and*** *researchers* ***custom heuristics.***

Desired service quality criteria



- ***Cost and adaptation efforts*** *are widely considered in industry vs widely ignored in academia.*
- ***Service coupling and granularity*** *are widely considered in both academia and industry.*

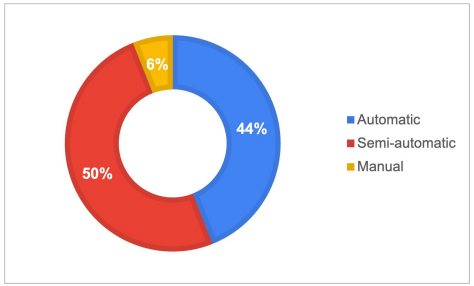
Targeted service types



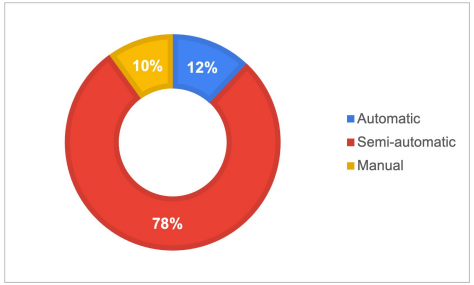
- *Lack of **academic type-sensitive** SI approaches.*
- *Service identification is **business driven**.*
- *Practitioners **prioritise** the identification of **domain-specific** rather than domain-neutral services.*

Automation of Service Identification

SI automation in academia

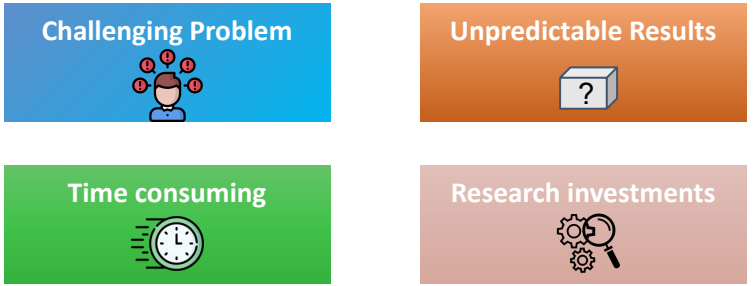


SI automation in industry



Observations

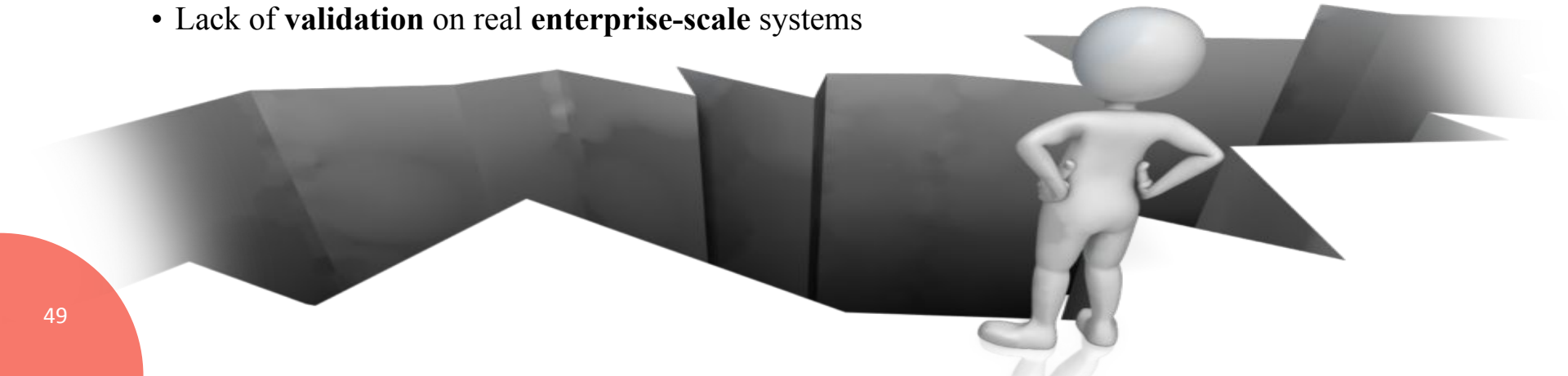
- The **full automation** of SI process **is not** the primary **focus** of practitioners.
- Automation in **wrapping** and **reverse engineering** techniques to document and extract the business logic of legacy systems.
- Practitioners do not take the **risk** to try to **fully automate** the SI process.



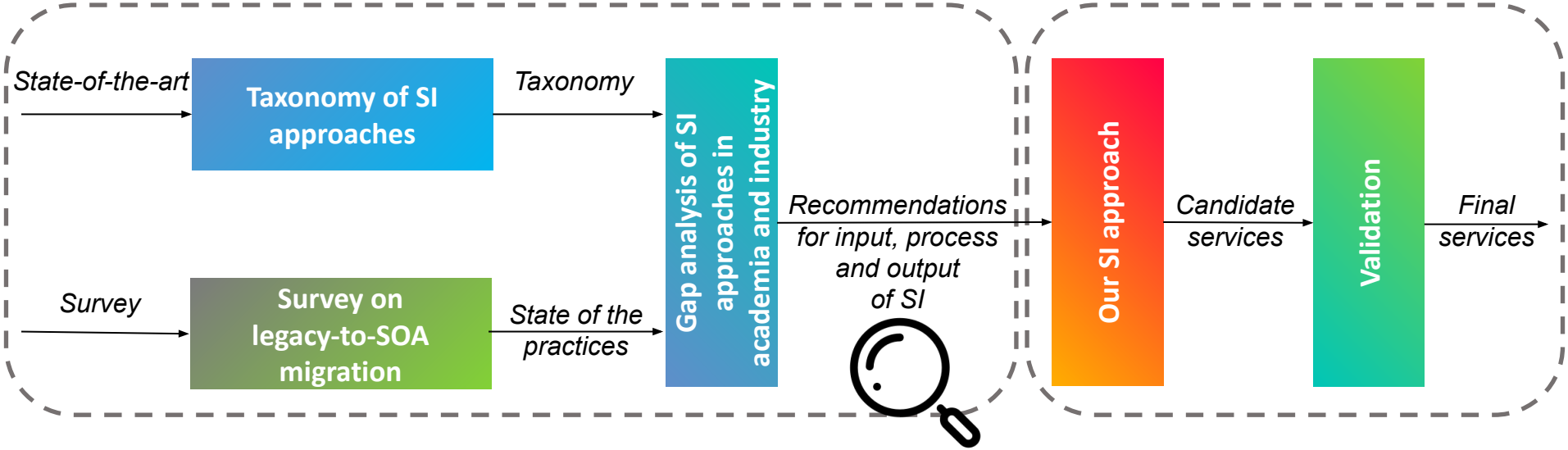
- **Academic** SIAs **focus** on the **automation** of the identification process.

Gap Analysis

- **Gap** between academia and industry
- **Poor knowledge transfer** between academia and industry in the context of legacy-to-SOA migration
- **Lack** of **cost-effective** academic service identification techniques
- Lack of **validation** on real **enterprise-scale** systems



Research methodology



Step1: Gap analysis of SI approaches in academia and industry

Step 2: Type-sensitive service identification

Recommendations

The input must be source code and production data

A deep understanding of the domain and a great familiarity with the legacy systems are necessary

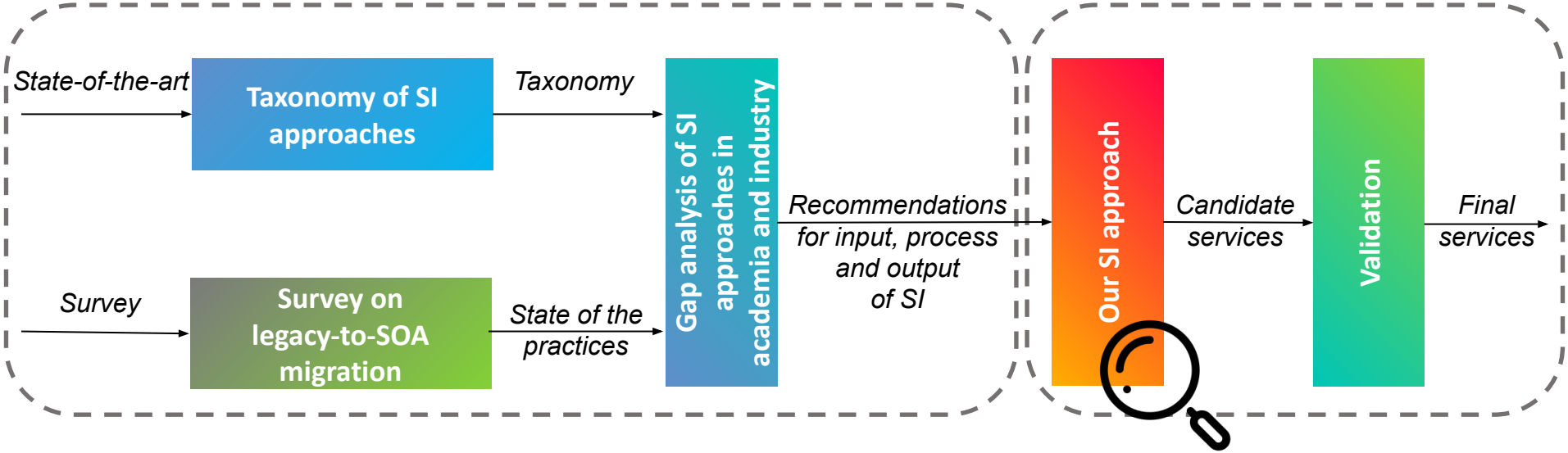
Service identification should be a business-value driven process

The process must follow a proven clustering technique

The output must be high-value, coarse-grained services

Service identification should be type-sensitive

Research methodology

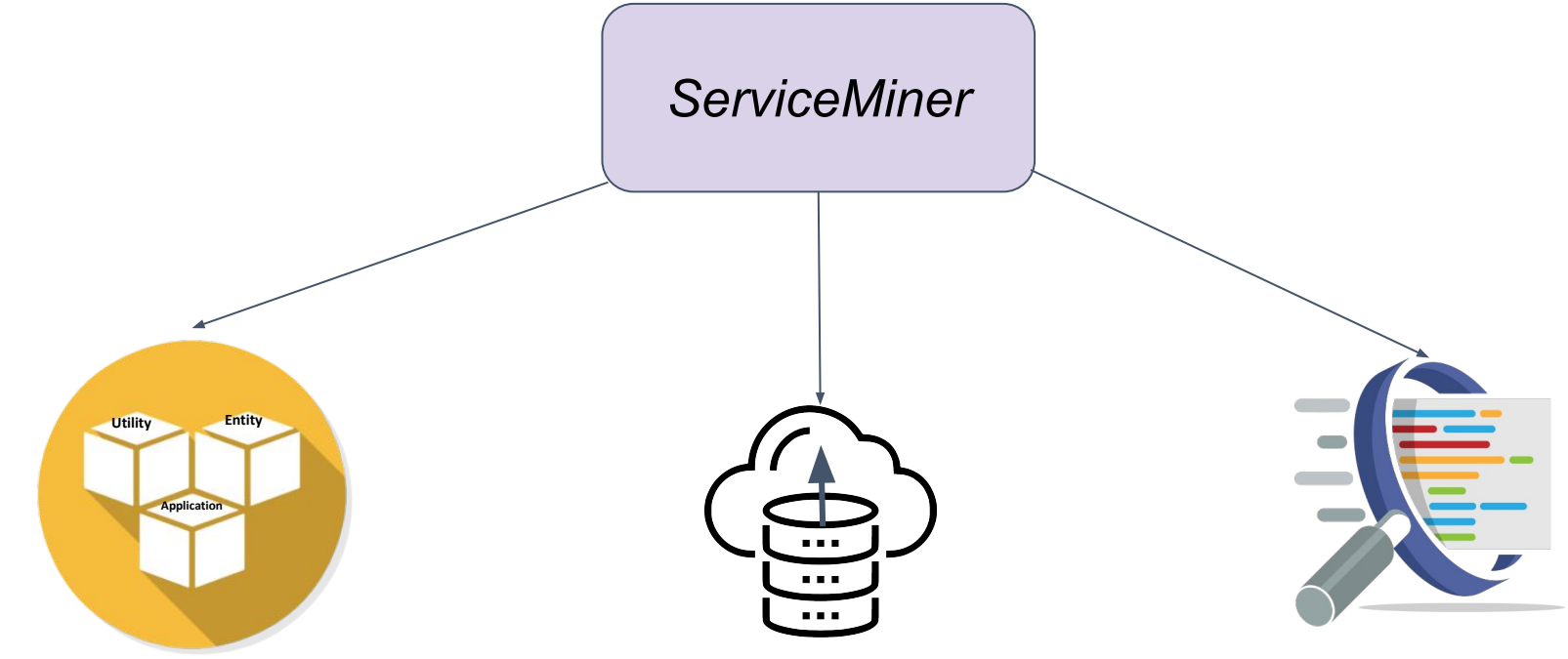


Step1: Gap analysis of SI approaches in academia and industry

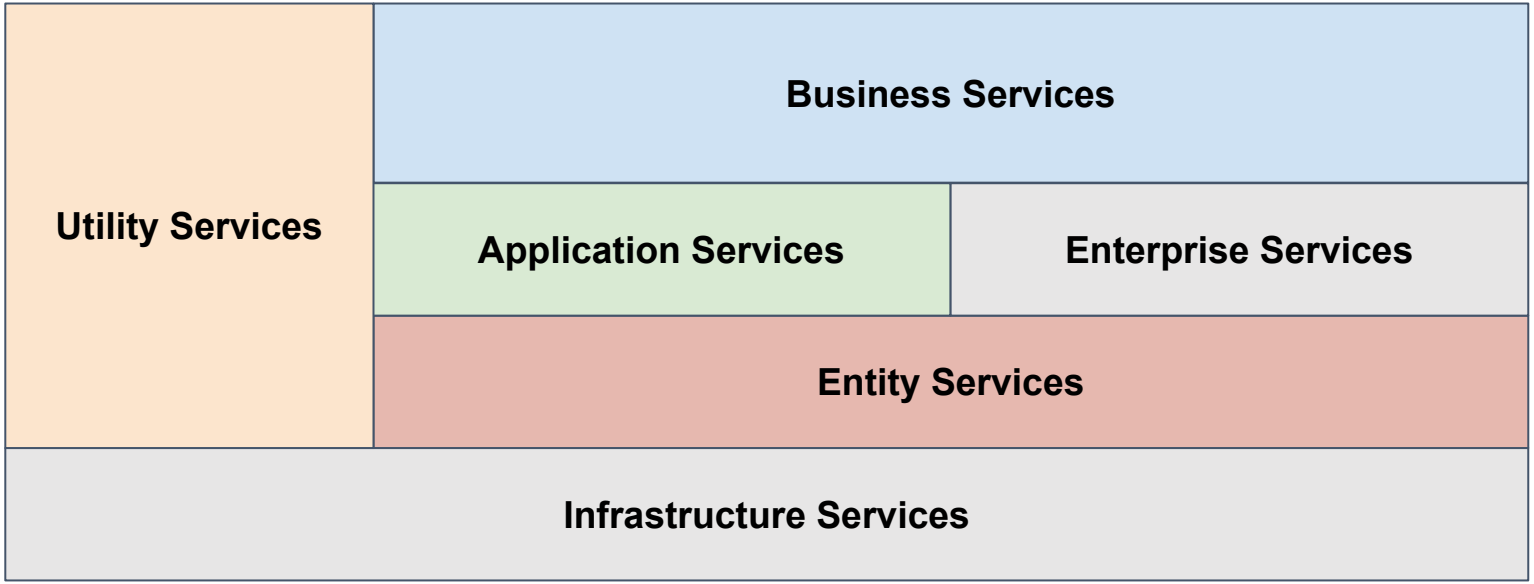
Step 2: Type-sensitive service identification

[Abdellatif et al., TSE 2021 - Under review]
[Abdellatif et al., ICSOC 2019]

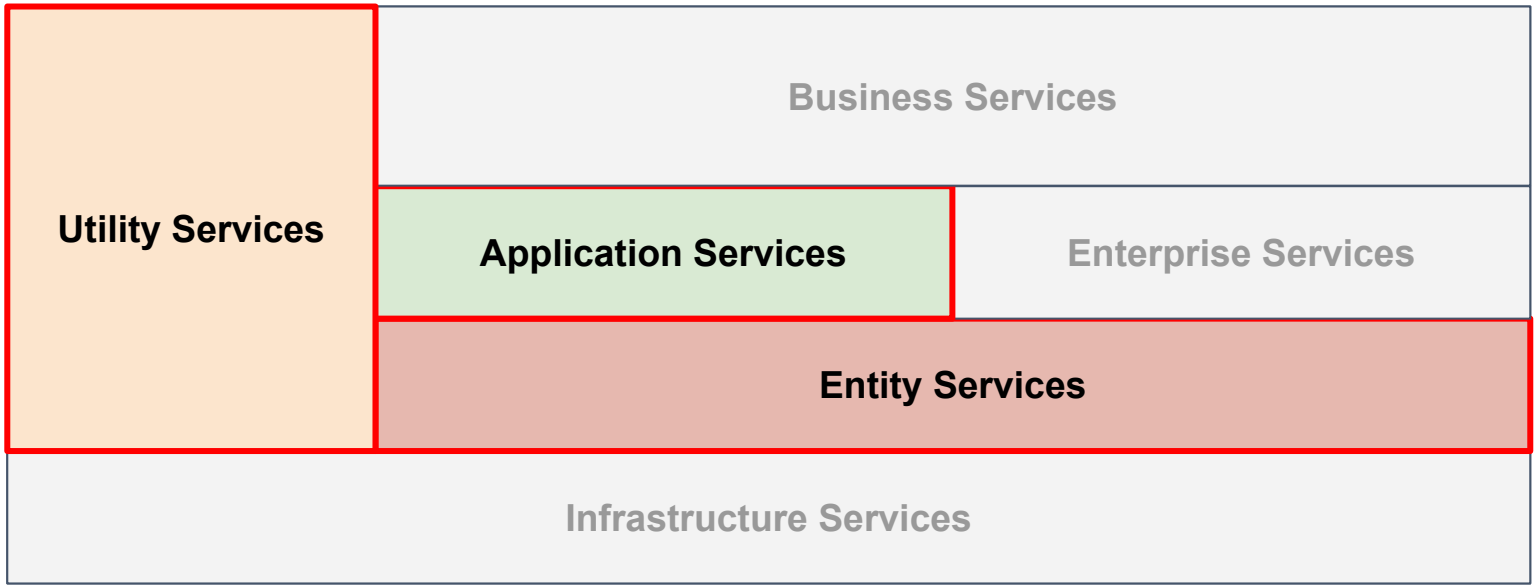
Solution


ServiceMiner
Utility
Entity
Application
Type-sensitive SI
Bottom-up Approach
Static Analysis

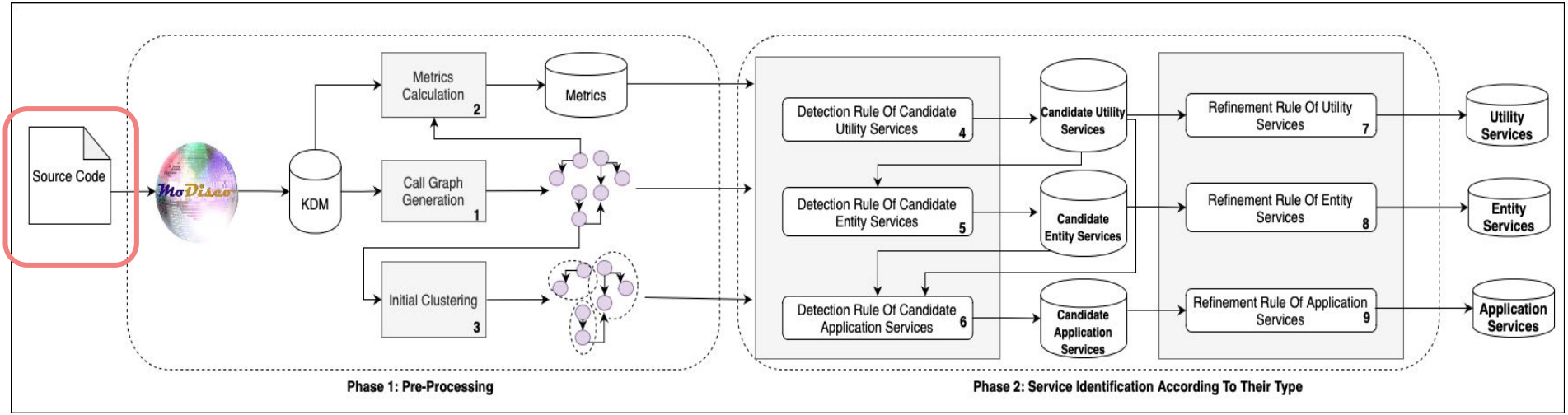
Taxonomy of service types



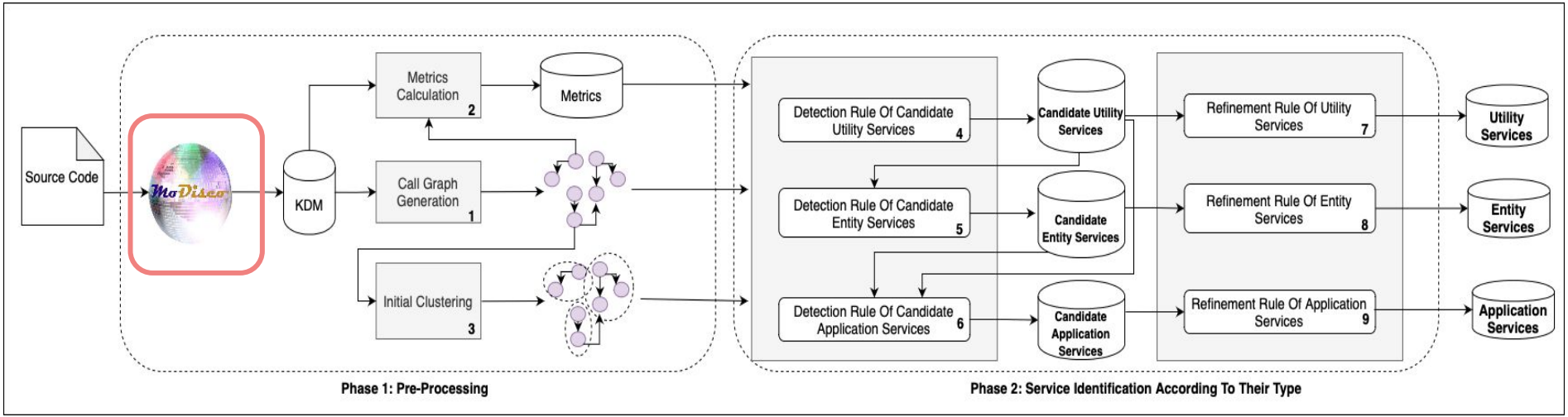
Targeted service types



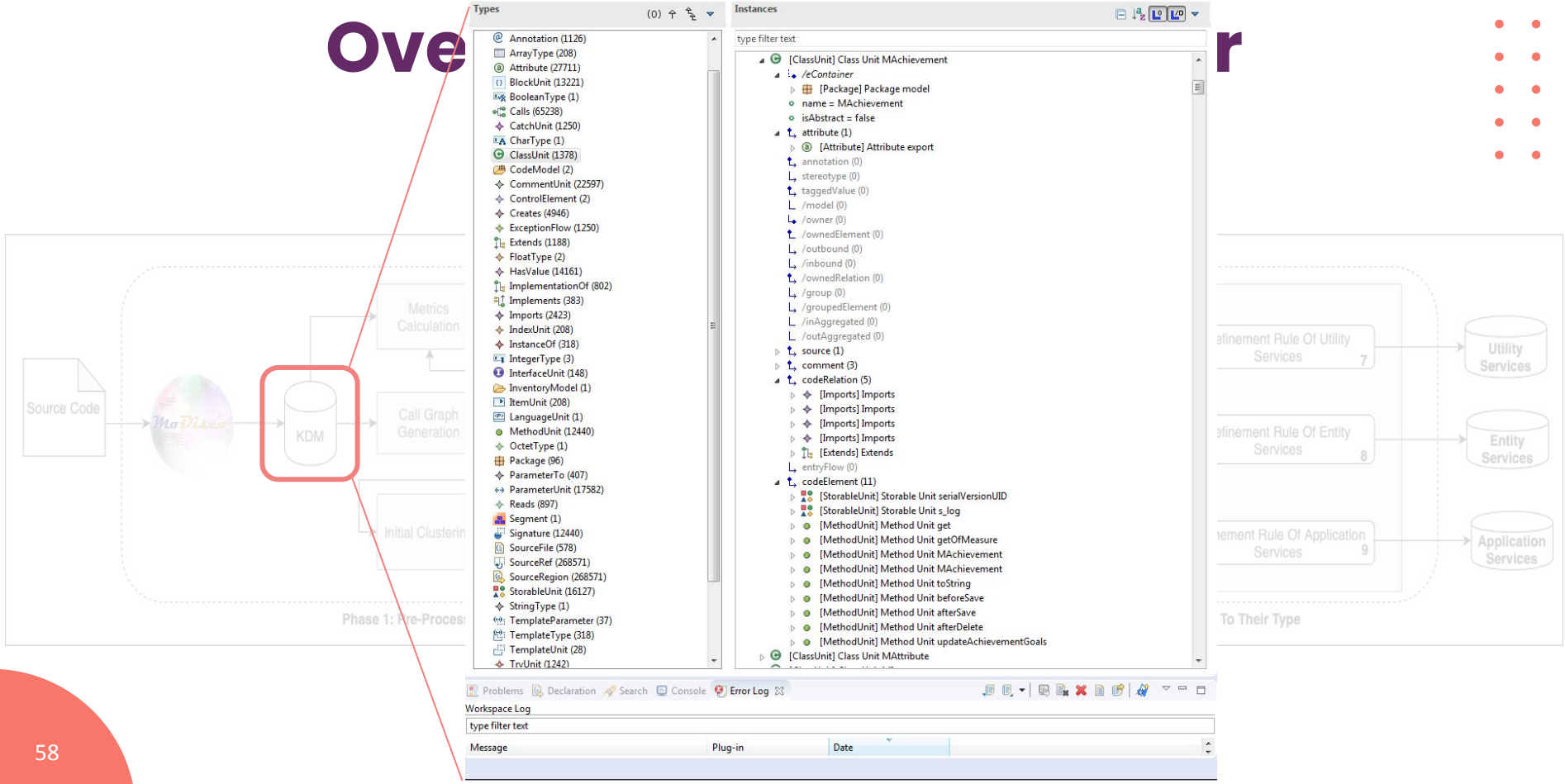
Overview of ServiceMiner



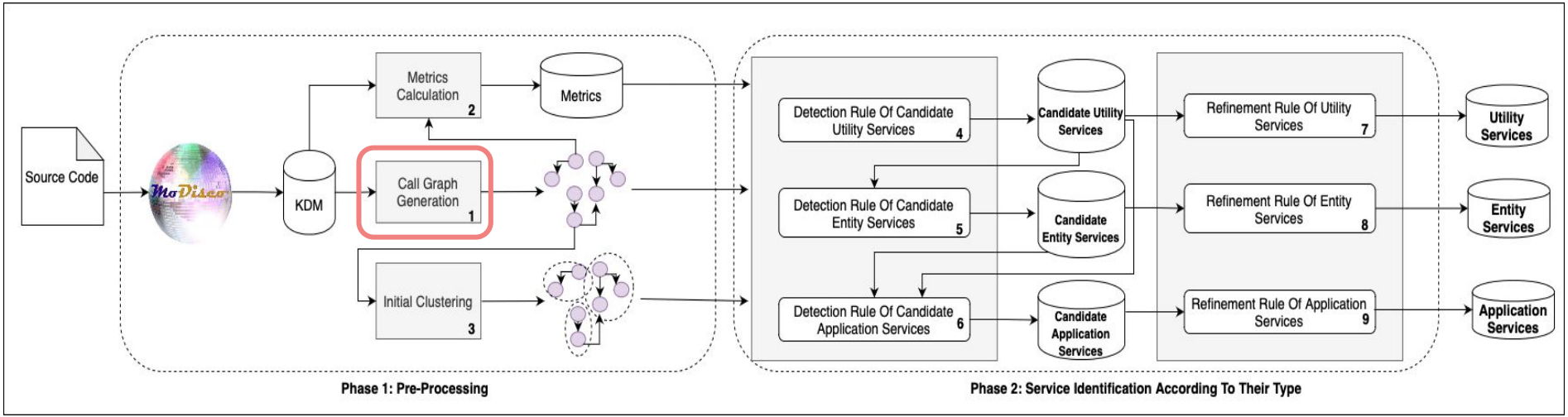
Overview of ServiceMiner



Ove r

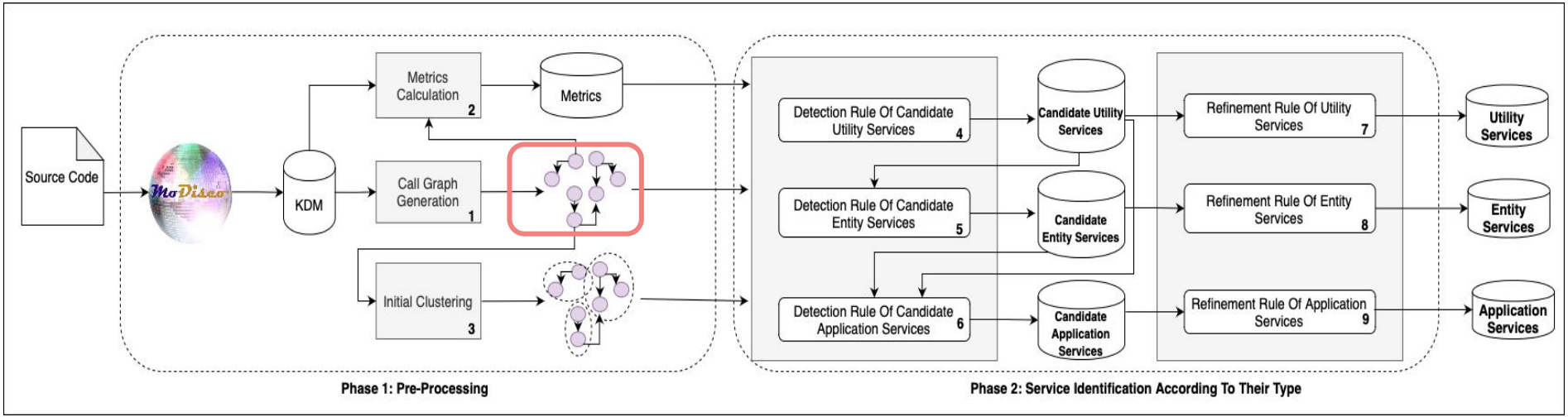

Types
(0)
Annotation (1126)
ArrayType (208)
Attribute (27711)
BlockUnit (13221)
BooleanType (1)
Calls (65238)
CatchUnit (1250)
CharType (1)
ClassUnit (1378)
CodeModel (2)
CommentUnit (22597)
ControlElement (2)
Creates (4946)
ExceptionFlow (1250)
Extends (1188)
FloatType (2)
HasValue (14161)
ImplementationOf (802)
Implements (383)
Imports (2423)
IndexUnit (208)
InstanceOf (318)
IntegerType (3)
InterfaceUnit (148)
InventoryModel (1)
ItemUnit (208)
LanguageUnit (1)
MethodUnit (12440)
OctetType (1)
Package (96)
ParameterTo (407)
ParameterUnit (17582)
Reads (897)
Segment (1)
Signature (12440)
SourceFile (578)
SourceRef (268571)
SourceRegion (268571)
StorableUnit (16127)
StringType (1)
TemplateParameter (37)
TemplateType (318)
TemplateUnit (28)
TryUnit (1242)
Instances
type filter text
[ClassUnit] Class Unit MAchievement
/eContainer
[Package] Package model
name = MAchievement
isAbstract = false
attribute (1)
[Attribute] Attribute export
annotation (0)
stereotype (0)
taggedValue (0)
/model (0)
/owner (0)
/ownedElement (0)
/outbound (0)
/inbound (0)
/ownedRelation (0)
/group (0)
/groupedElement (0)
/inAggregated (0)
/outAggregated (0)
source (1)
comment (3)
codeRelation (5)
[Imports] Imports
[Imports] Imports
[Imports] Imports
[Imports] Imports
[Extends] Extends
entryFlow (0)
codeElement (11)
[StorableUnit] Storable Unit serialVersionUID
[StorableUnit] Storable Unit s_log
[MethodUnit] Method Unit get
[MethodUnit] Method Unit getOfMeasure
[MethodUnit] Method Unit MAchievement
[MethodUnit] Method Unit MAchievement
[MethodUnit] Method Unit toString
[MethodUnit] Method Unit beforeSave
[MethodUnit] Method Unit afterSave
[MethodUnit] Method Unit afterDelete
[MethodUnit] Method Unit updateAchievementGoals
[ClassUnit] Class Unit MAttribute
Problems
Declaration
Search
Console
Error Log
Workspace Log
type filter text
Message
Plug-in
Date
Source Code
KDM
Metrics Calculation
Call Graph Generation
Initial Clusterin
Phase 1: Pre-Process
efinement Rule Of Utility Services 7
efinement Rule Of Entity Services 8
ement Rule Of Application Services 9
Utility Services
Entity Services
Application Services
To Their Type

Overview of ServiceMiner

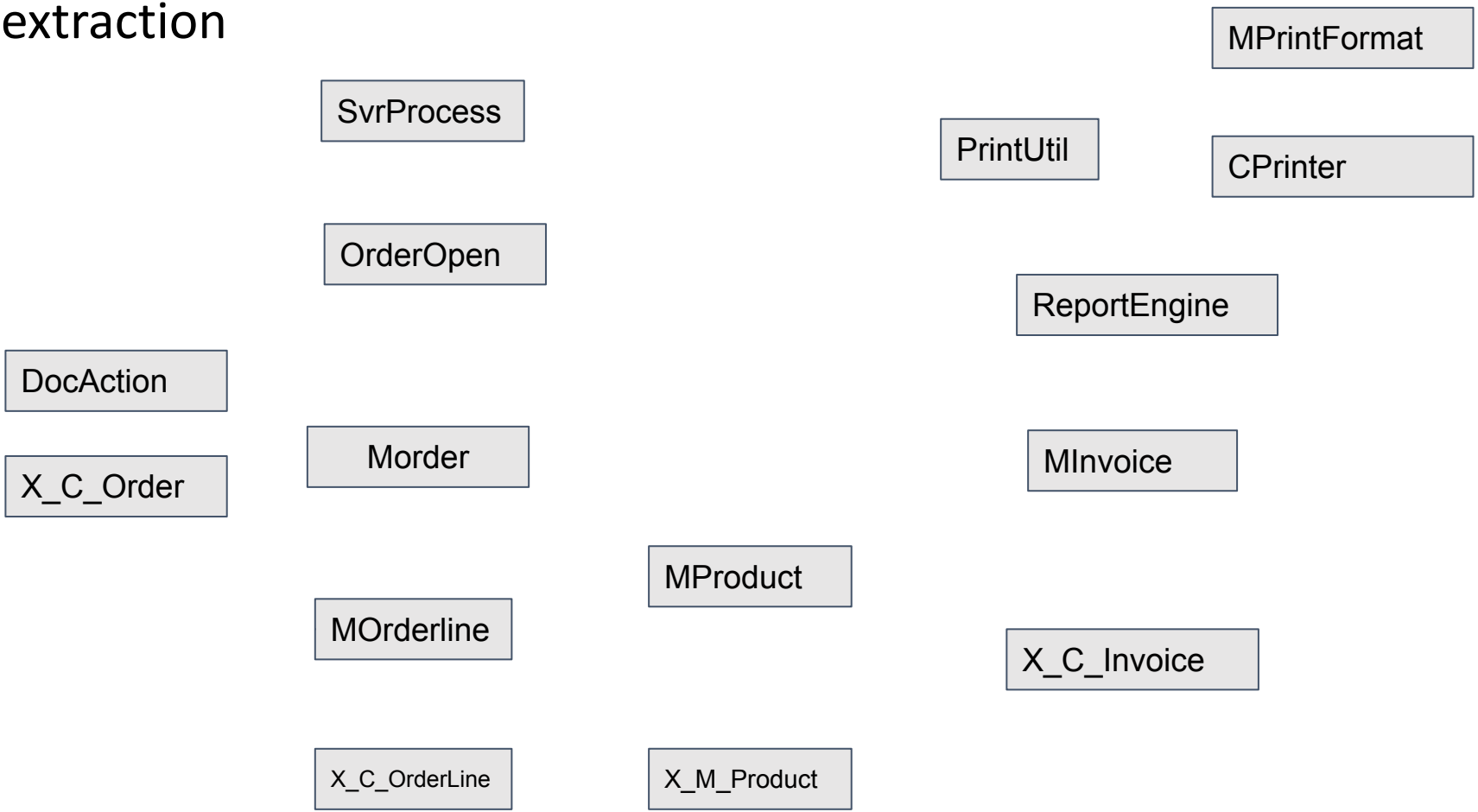


[Hecht et al., APSEC 2018]
[Shatnawi et al., MSR 2017]

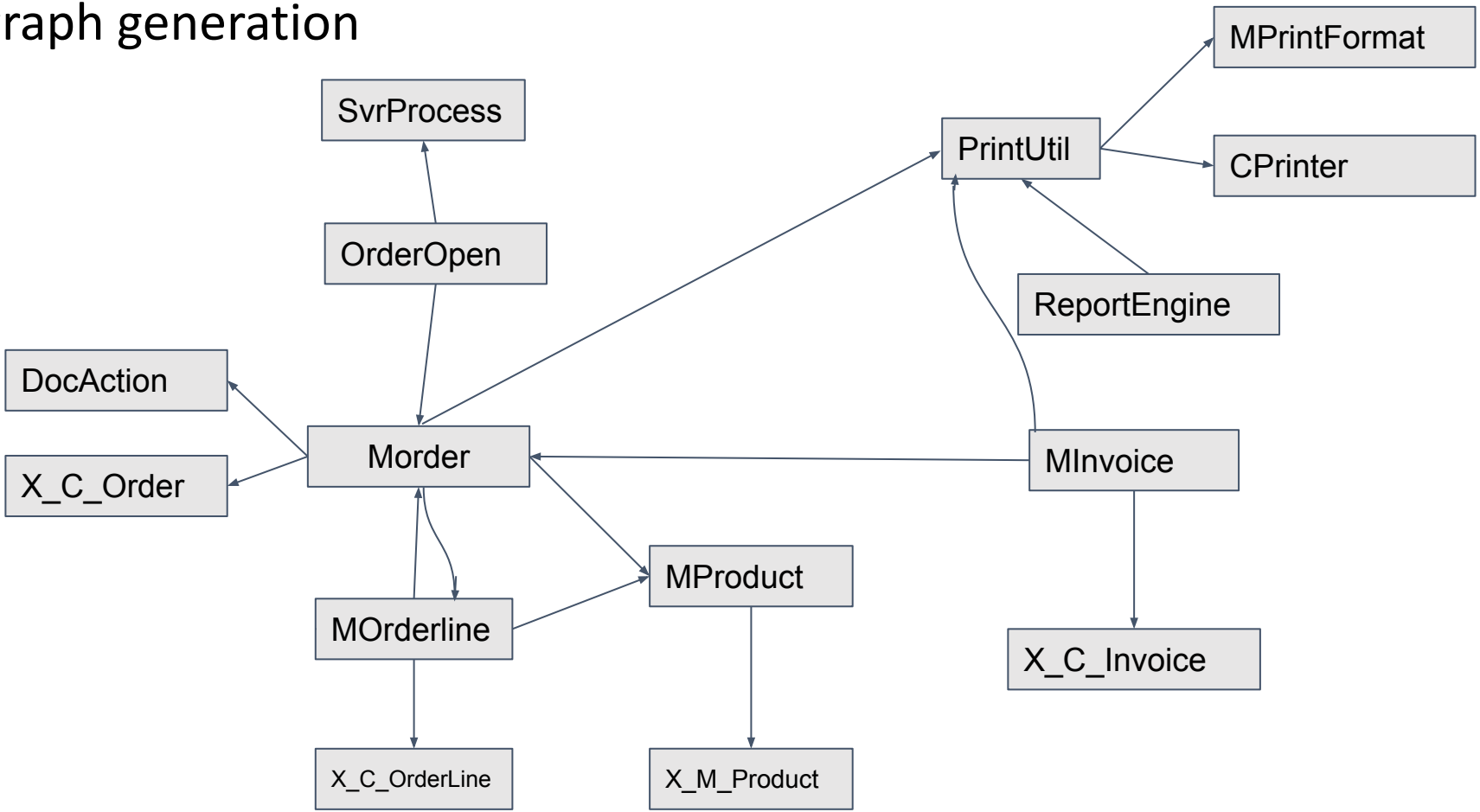
Overview of ServiceMiner



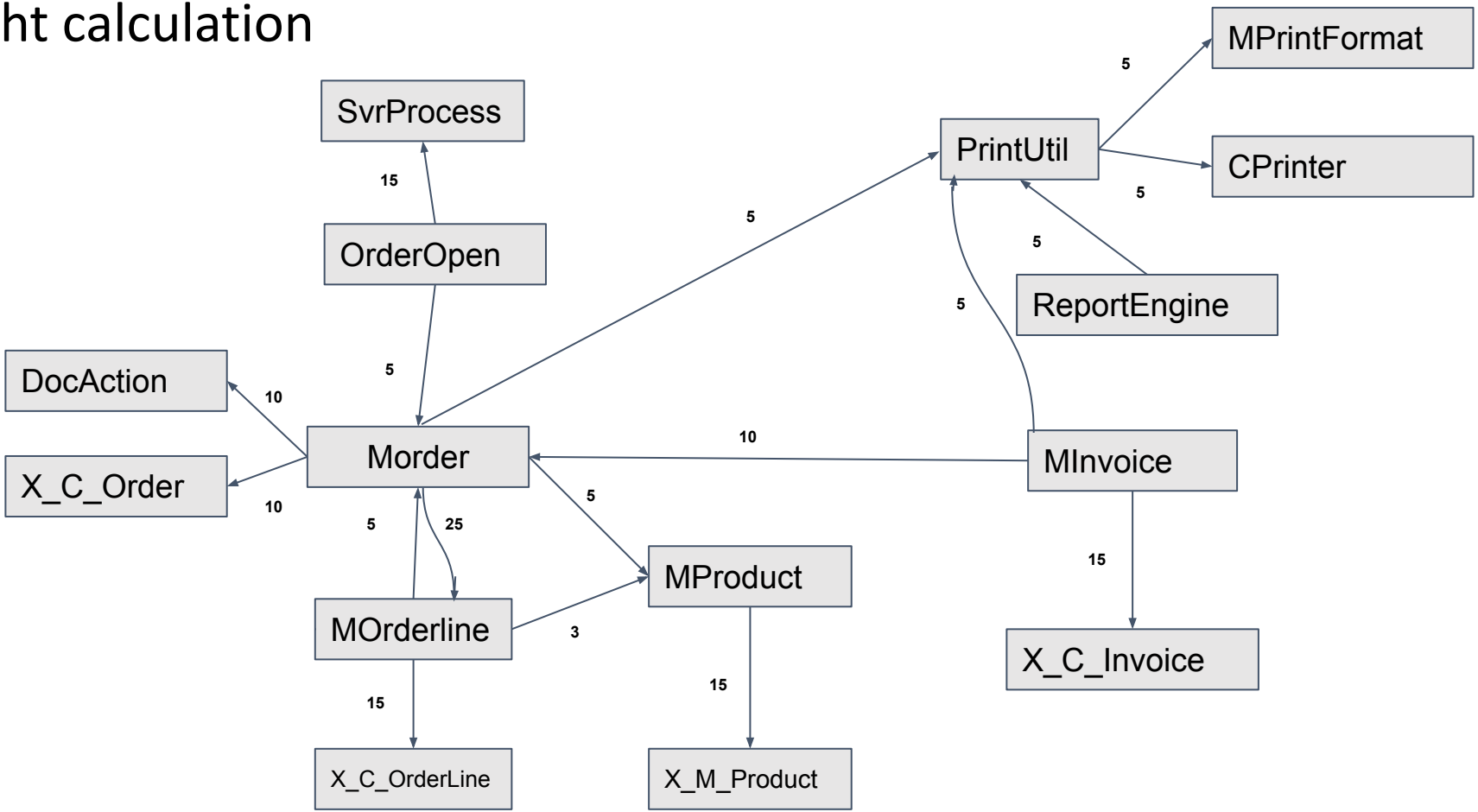
Class extraction



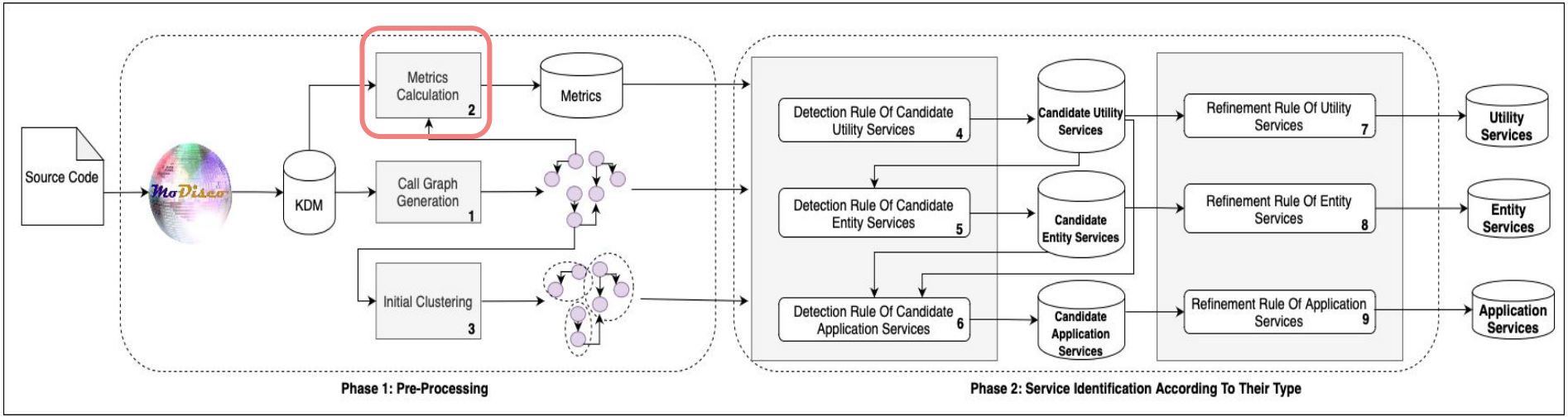
Call graph generation



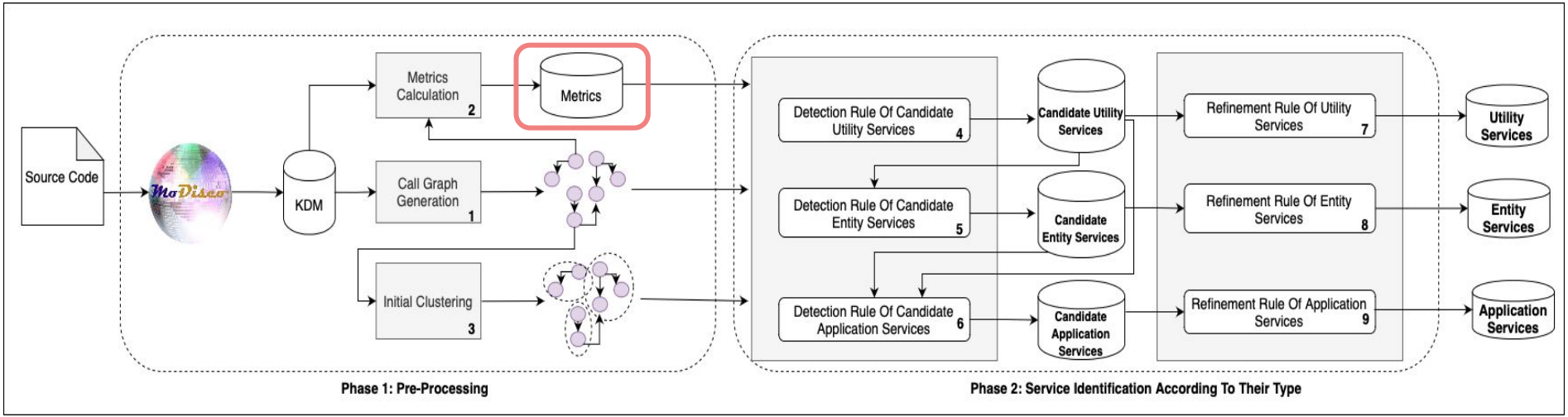
Weight calculation



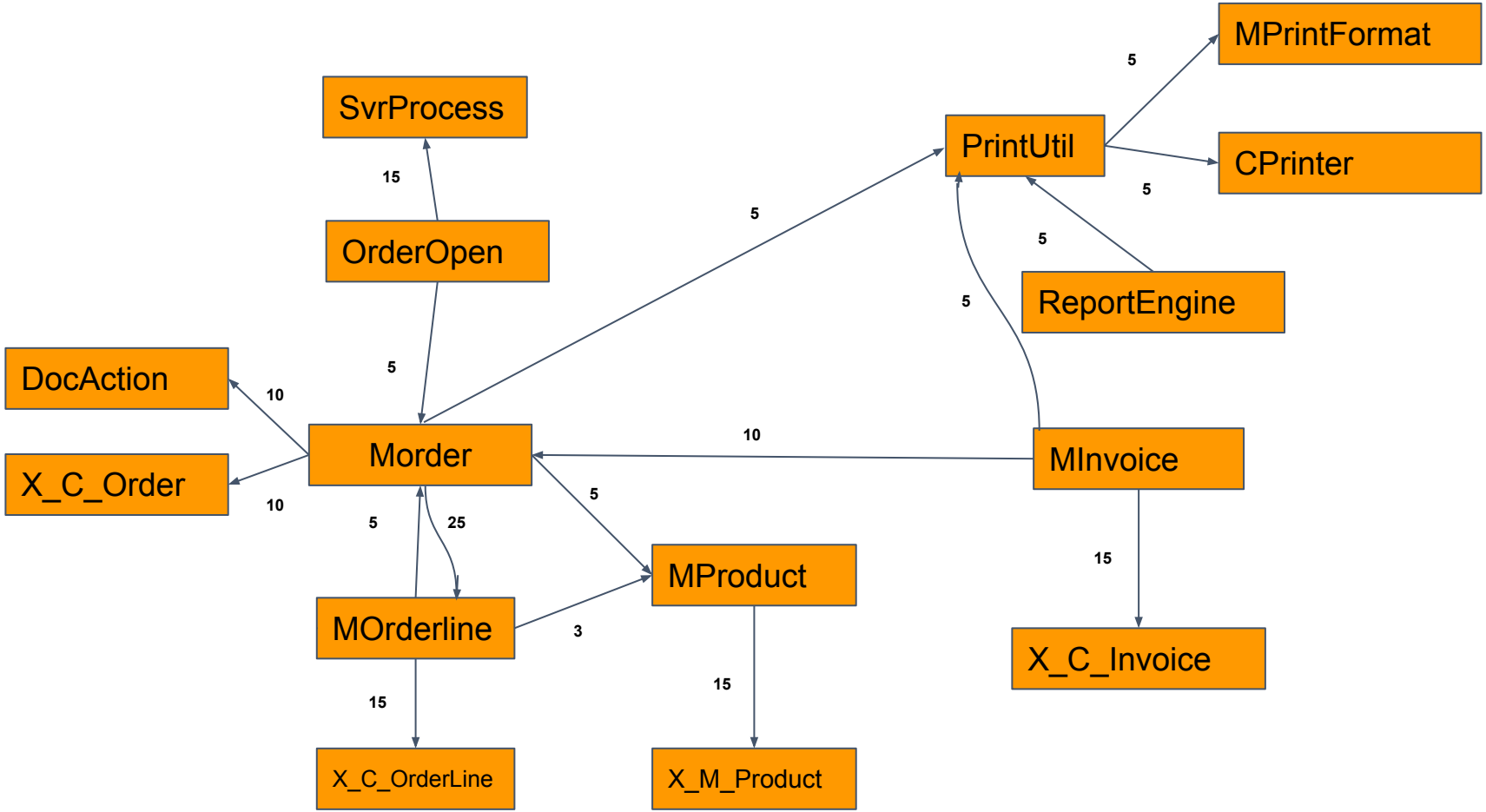
Overview of ServiceMiner



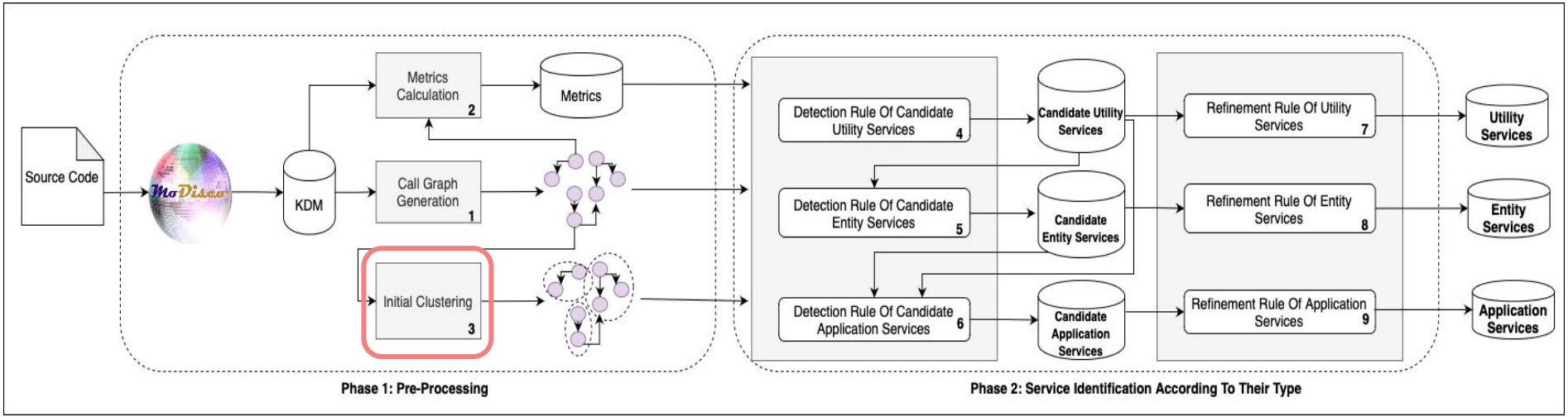
Overview of ServiceMiner



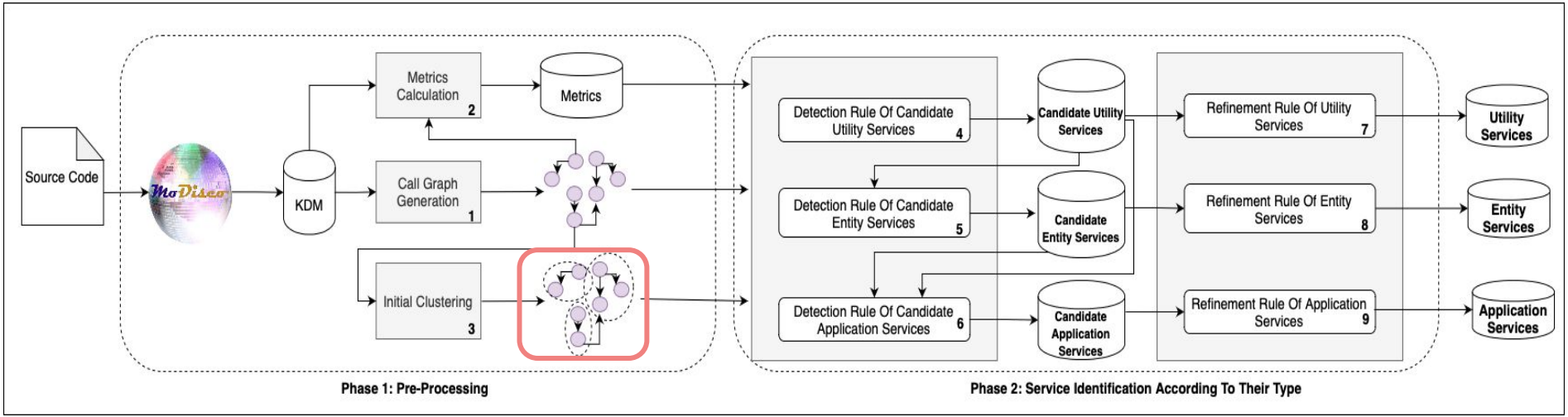
Metrics calculation for each class



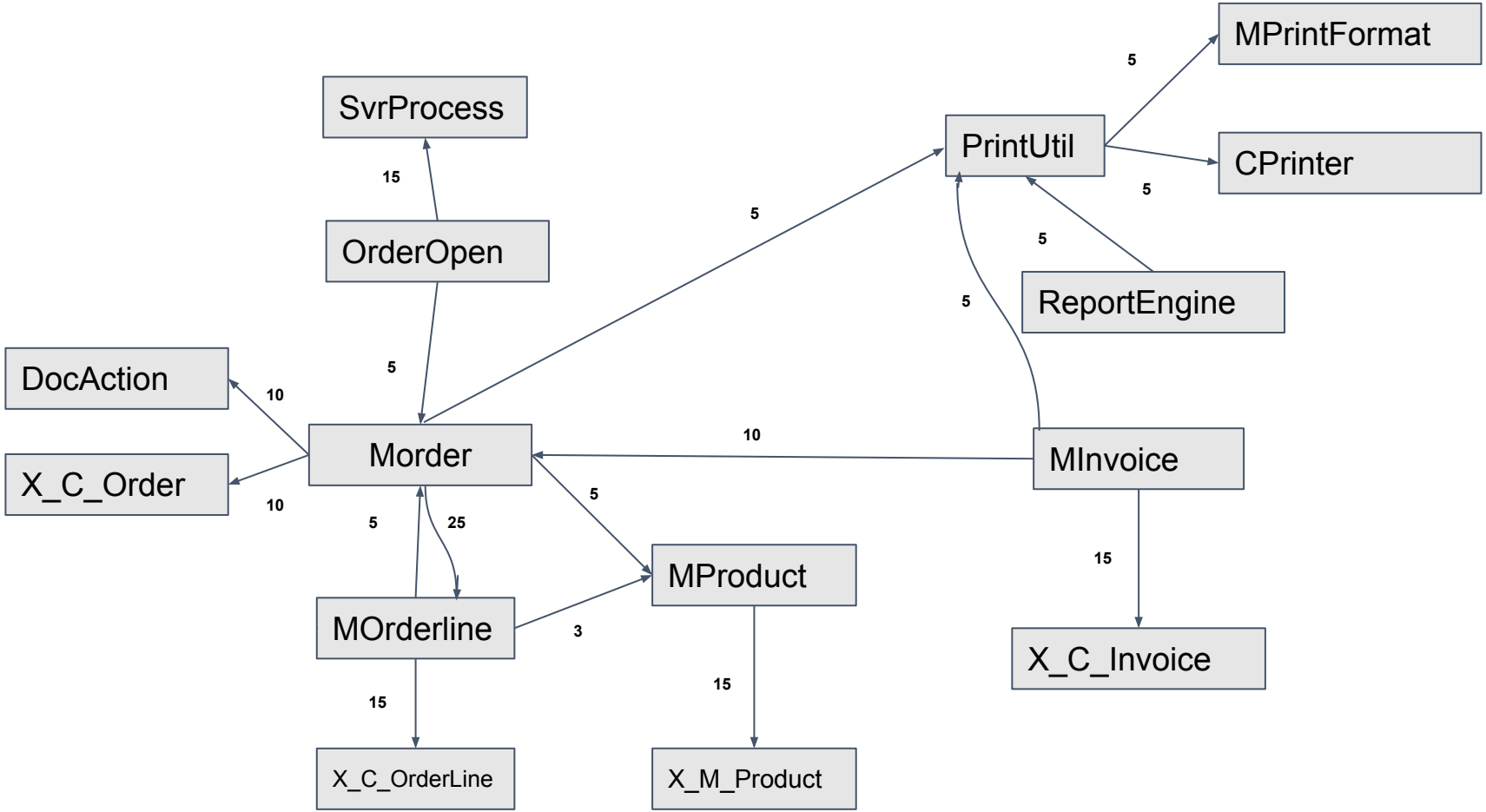
Overview of ServiceMiner



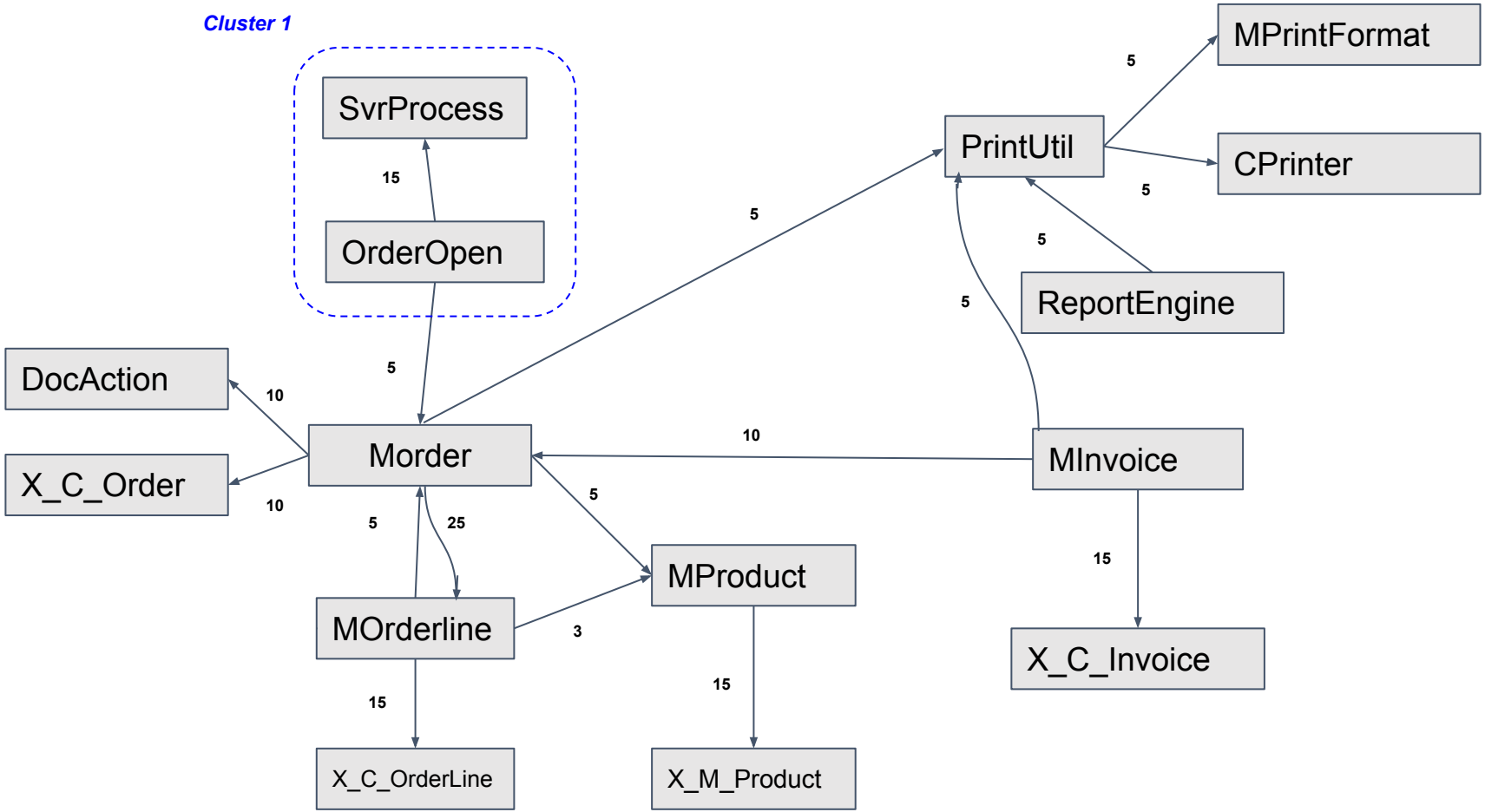
Overview of ServiceMiner



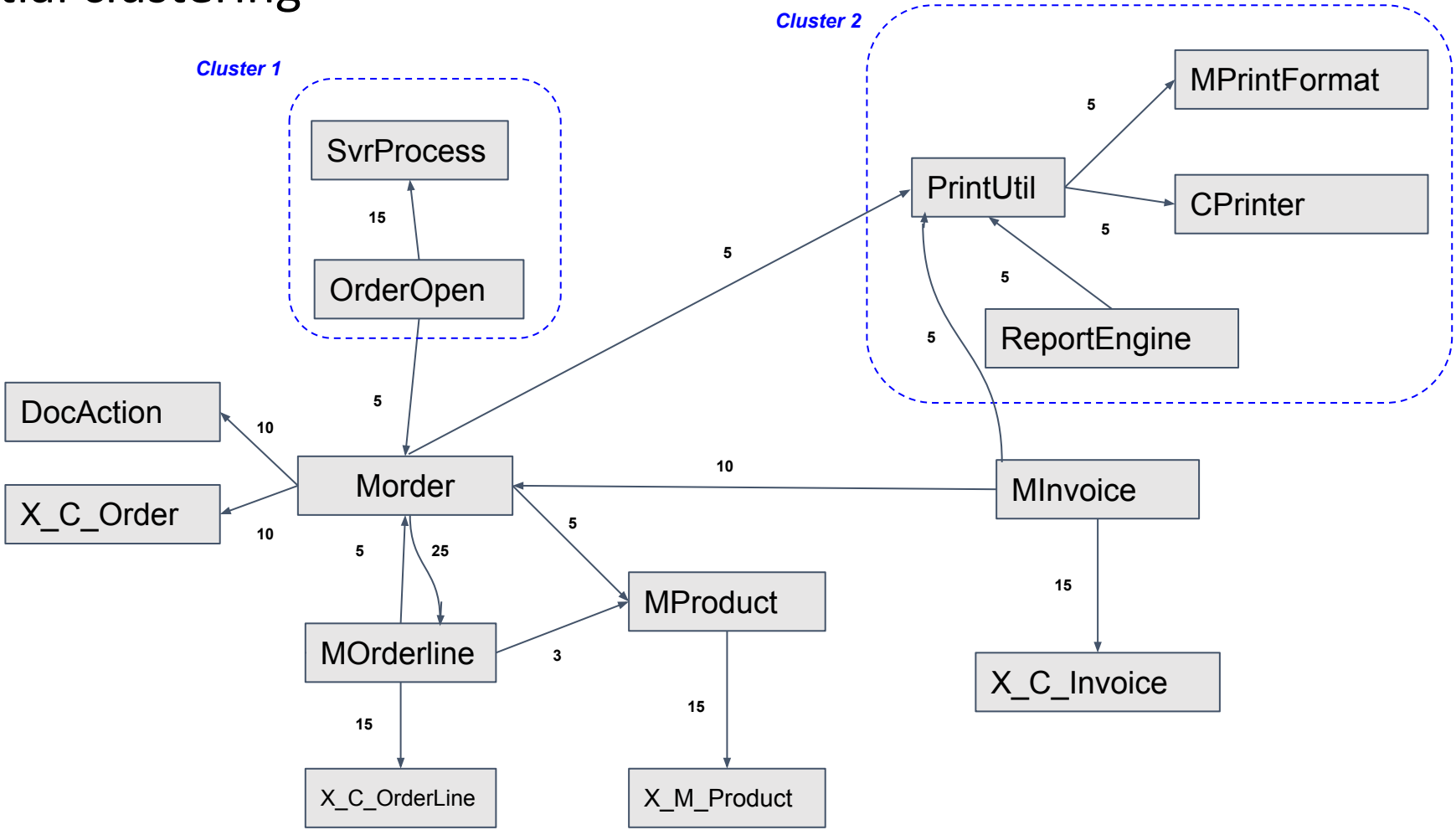
Running Example



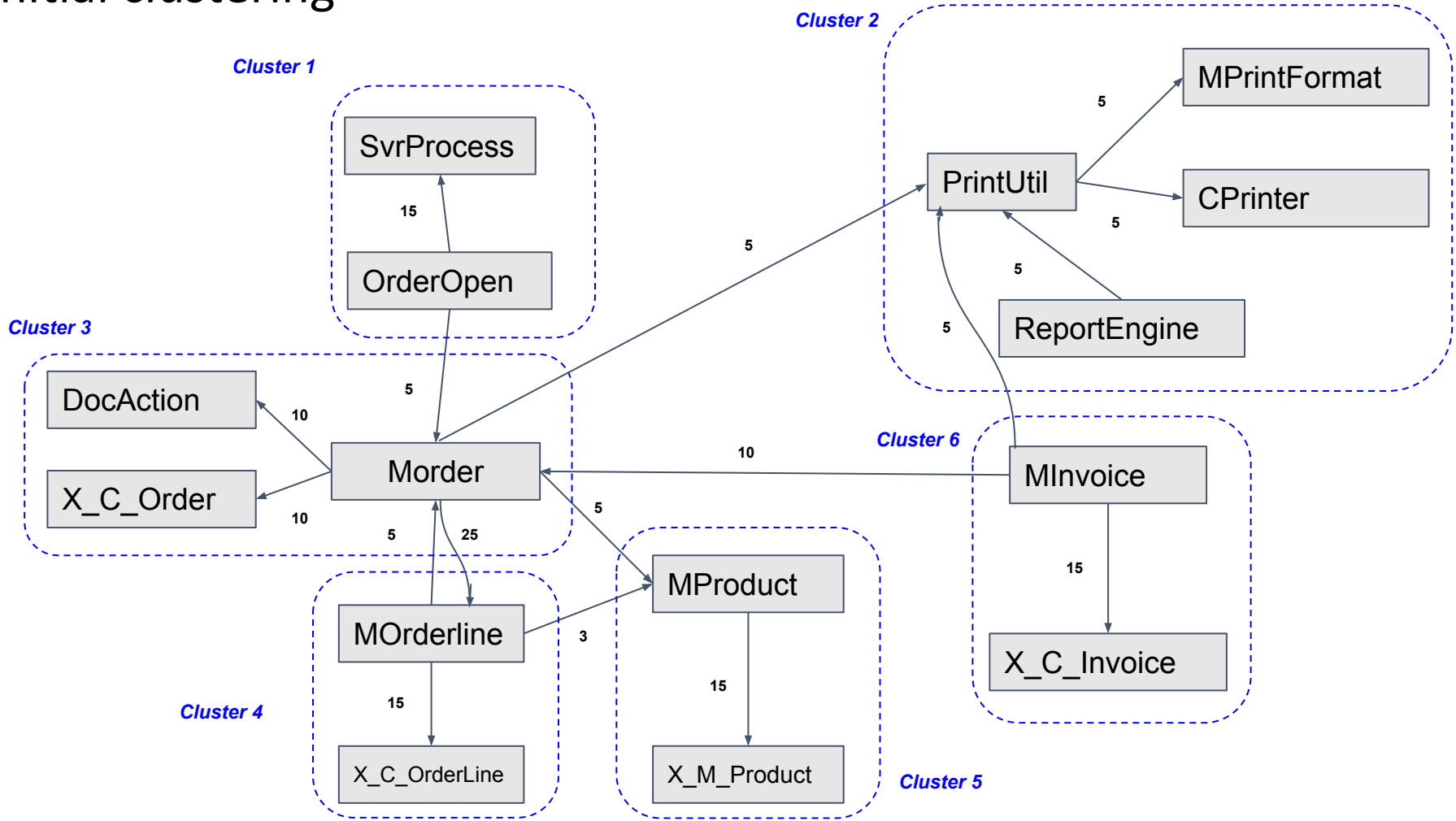
Initial clustering



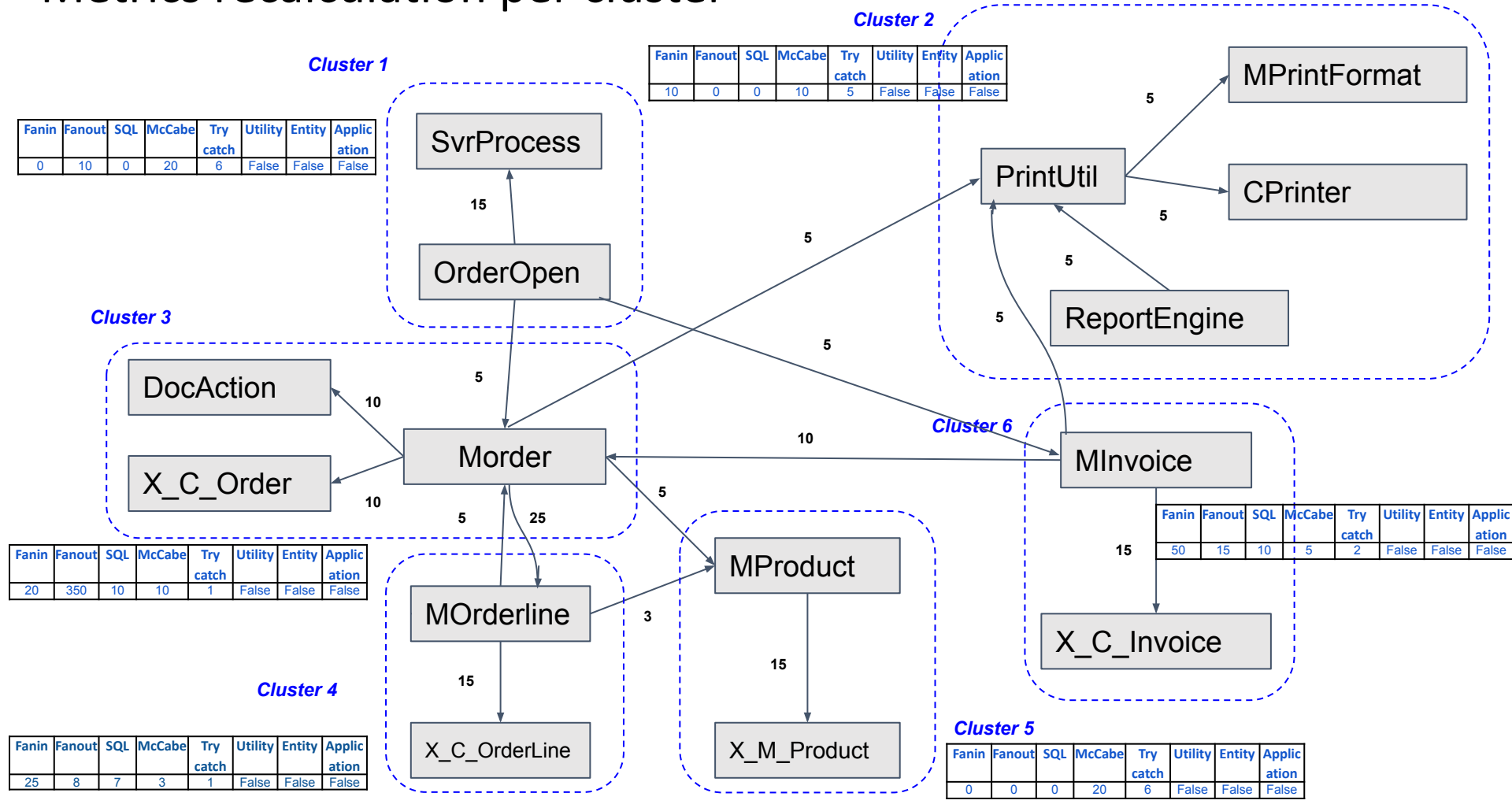
Initial clustering



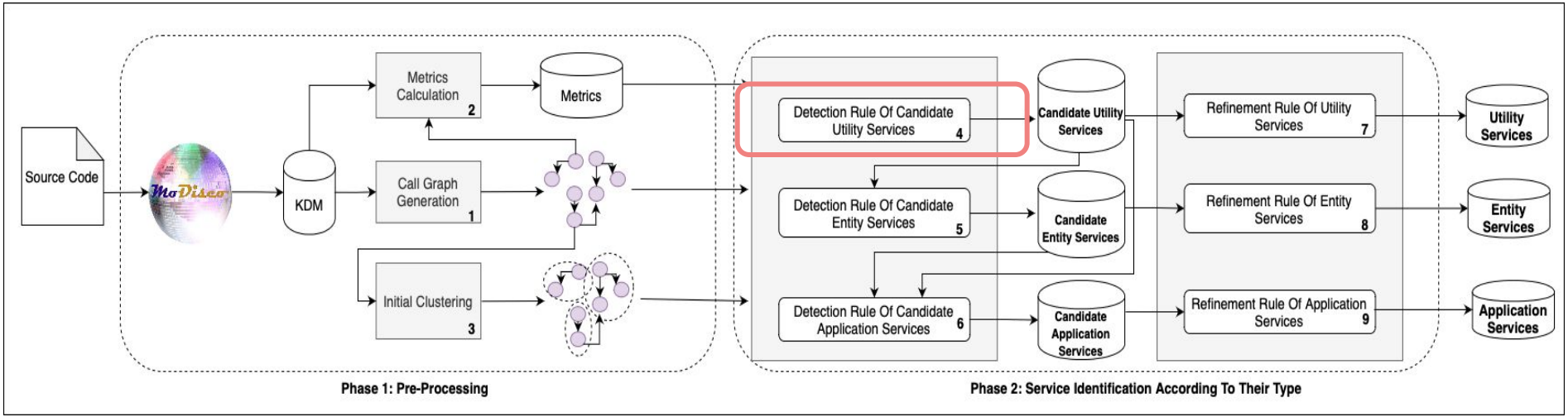
Initial clustering



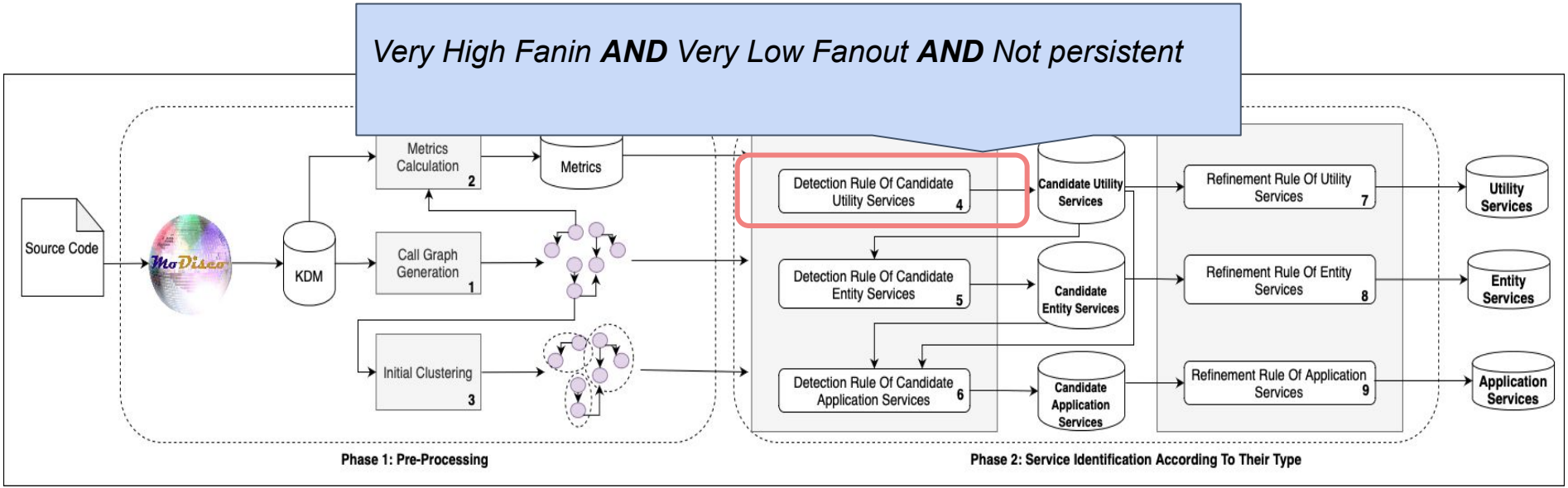
Metrics recalculation per cluster



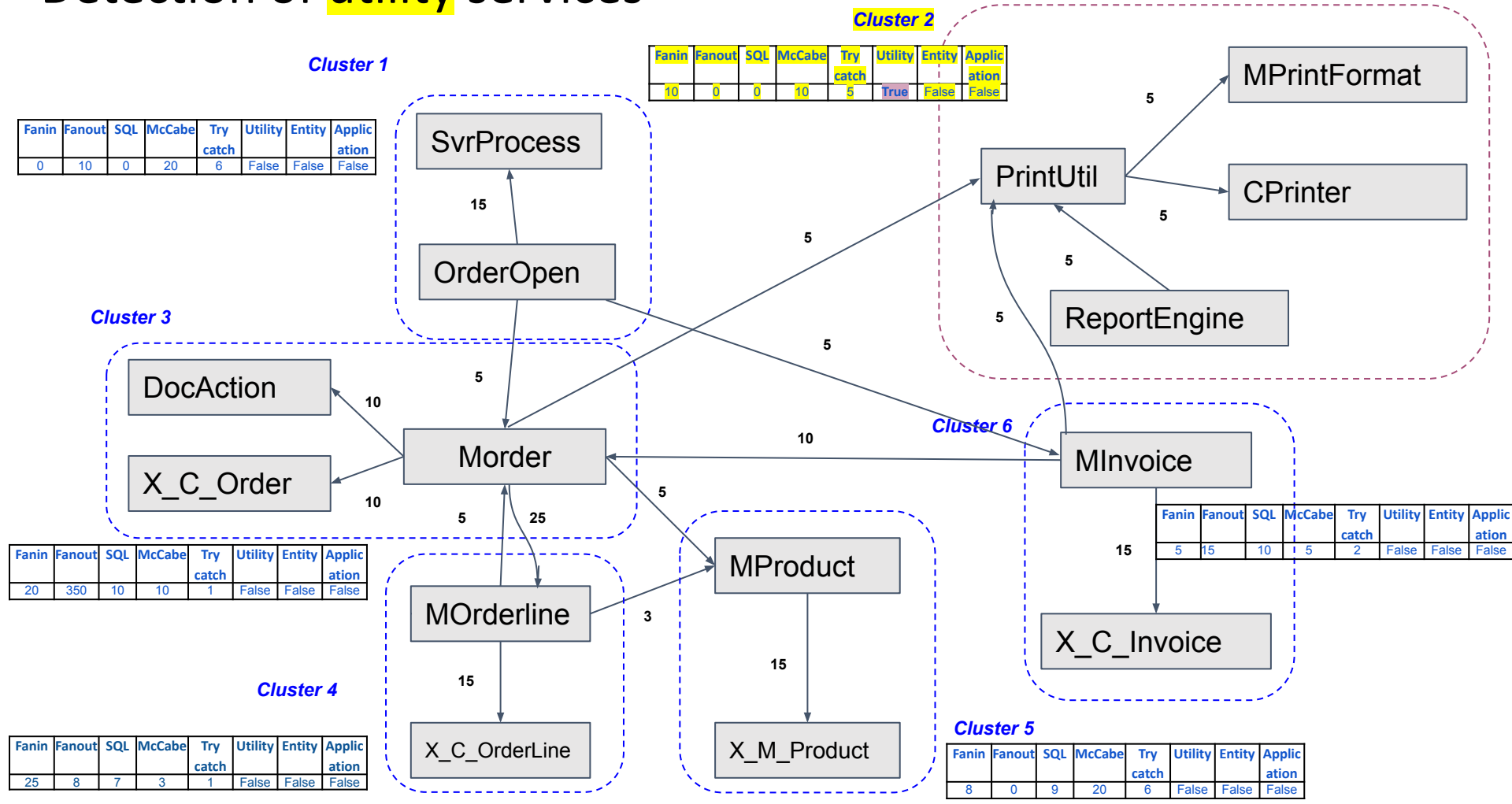
Overview of ServiceMiner



Overview of ServiceMiner



Detection of utility services



Fanin	Fanout	SQL	McCabe	Try catch	Utility	Entity	Application
0	10	0	20	6	False	False	False

Fanin	Fanout	SQL	McCabe	Try catch	Utility	Entity	Application
10	0	0	10	5	True	False	False

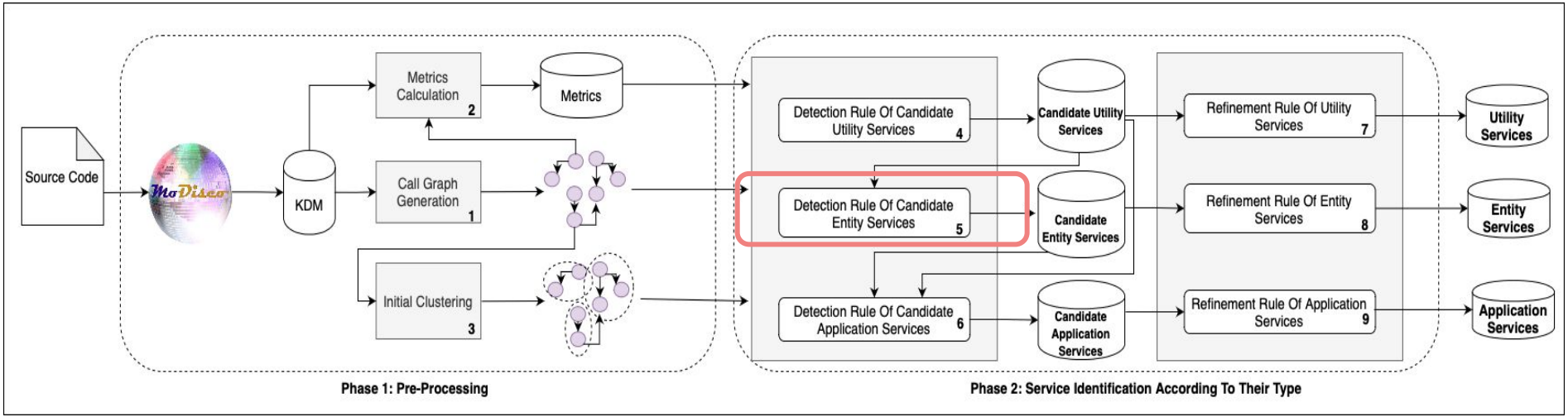
Fanin	Fanout	SQL	McCabe	Try catch	Utility	Entity	Application
20	350	10	10	1	False	False	False

Fanin	Fanout	SQL	McCabe	Try catch	Utility	Entity	Application
5	15	10	5	2	False	False	False

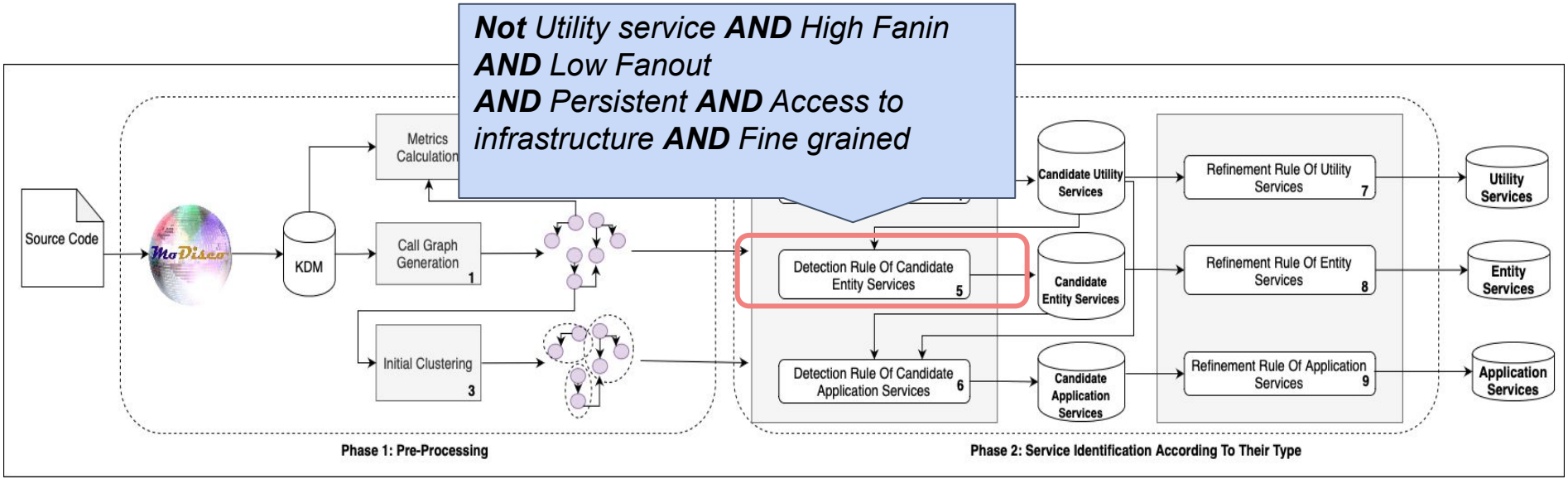
Fanin	Fanout	SQL	McCabe	Try catch	Utility	Entity	Application
25	8	7	3	1	False	False	False

Fanin	Fanout	SQL	McCabe	Try catch	Utility	Entity	Application
8	0	9	20	6	False	False	False

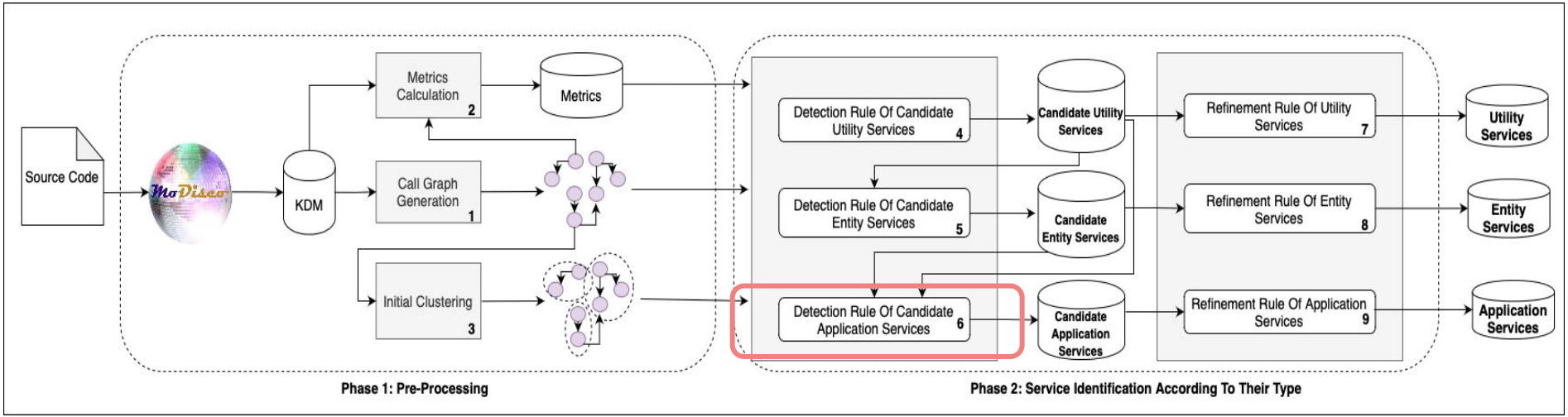
Overview of ServiceMiner



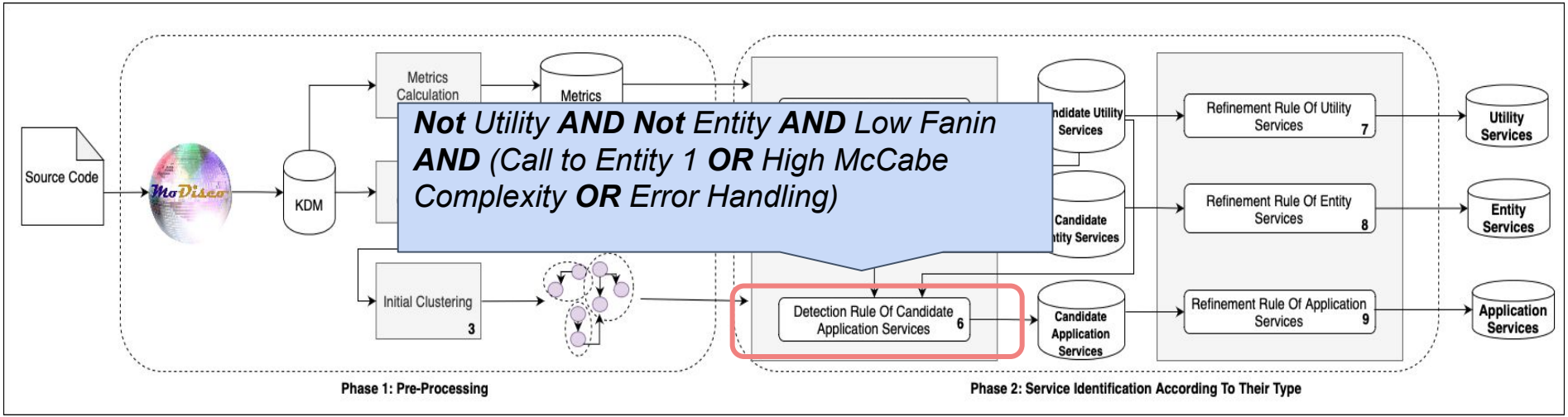
Overview of ServiceMiner



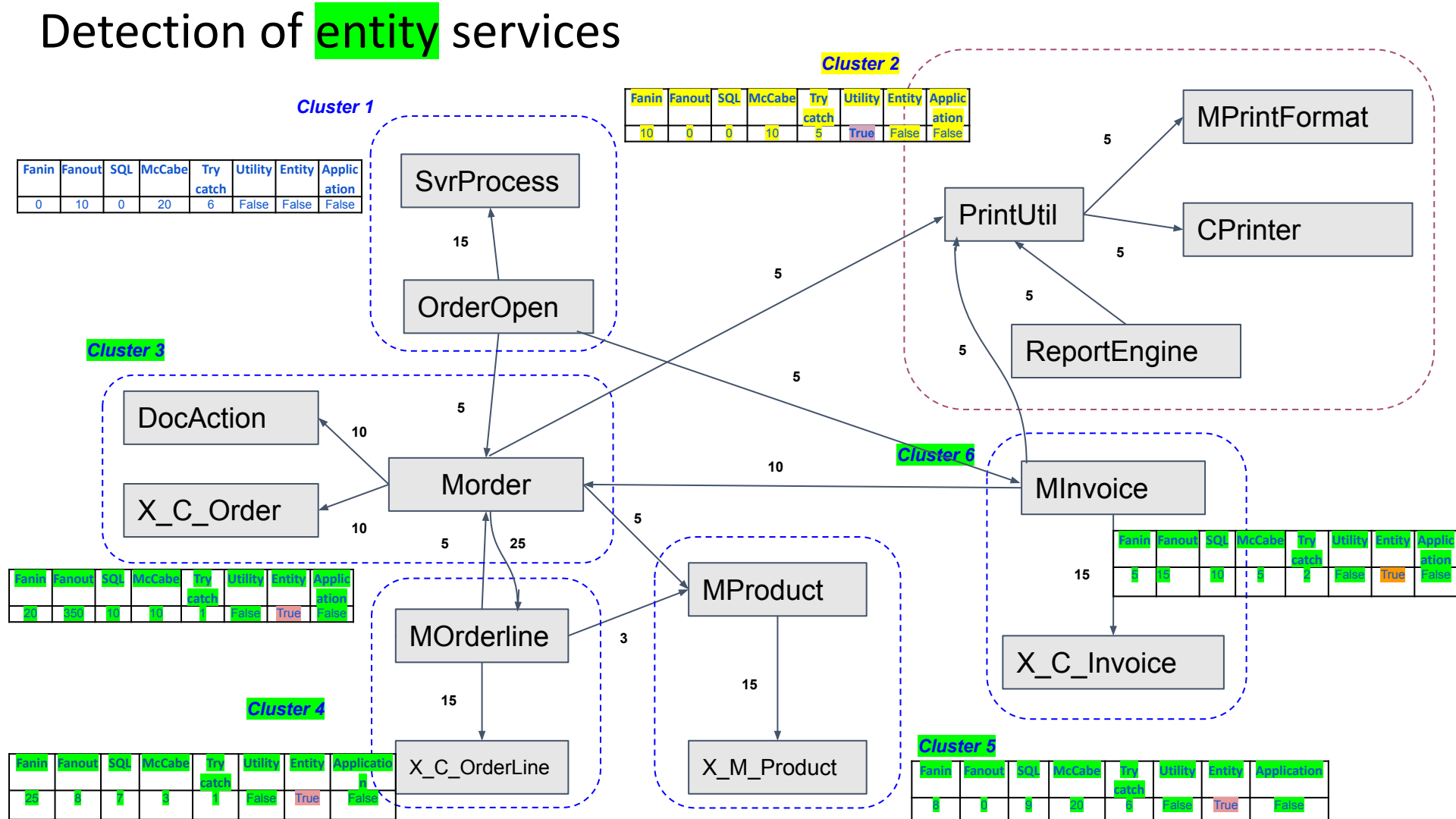
Overview of ServiceMiner



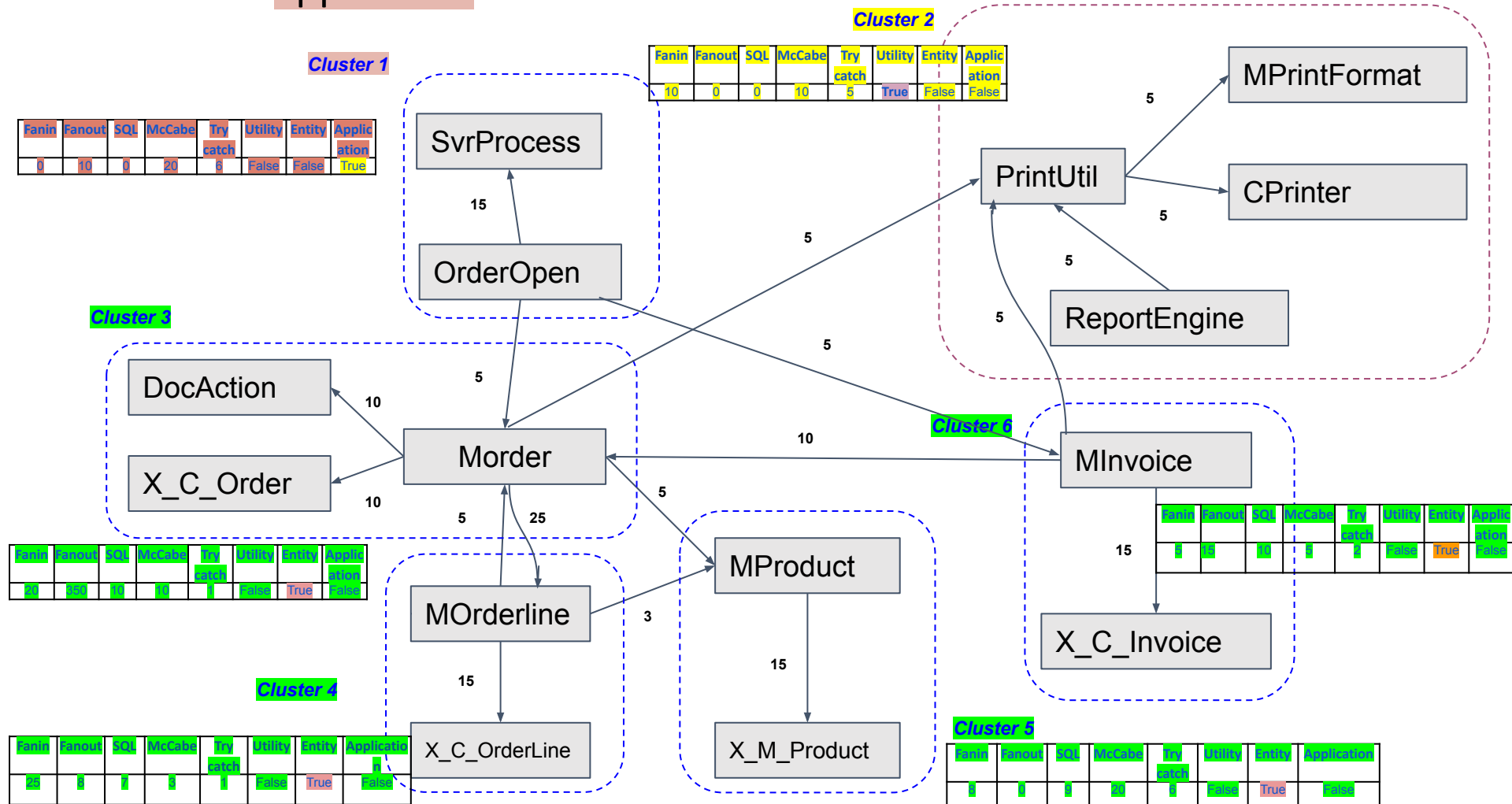
Overview of ServiceMiner



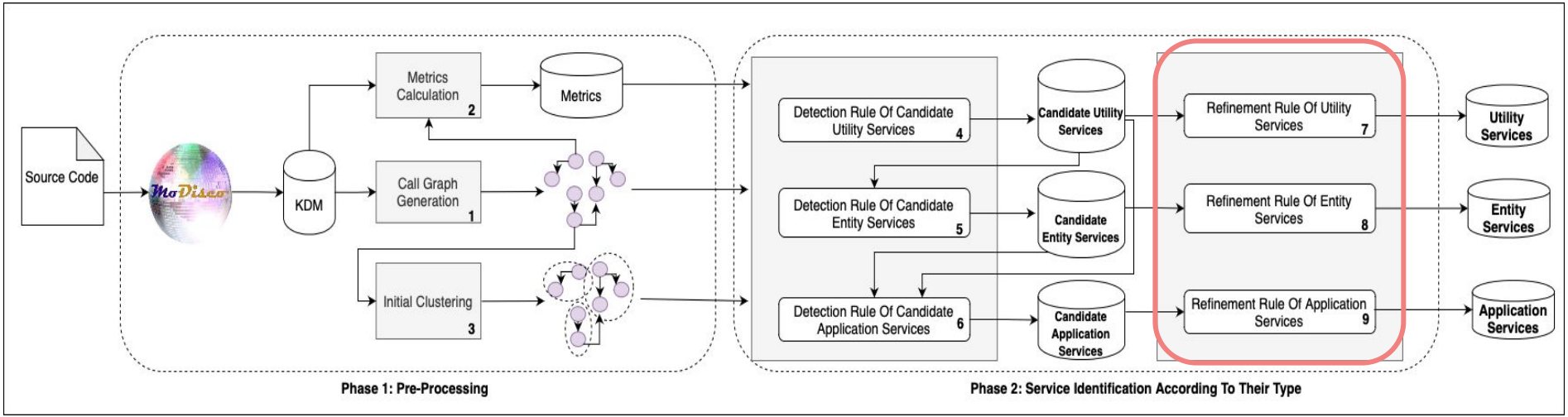
Detection of entity services



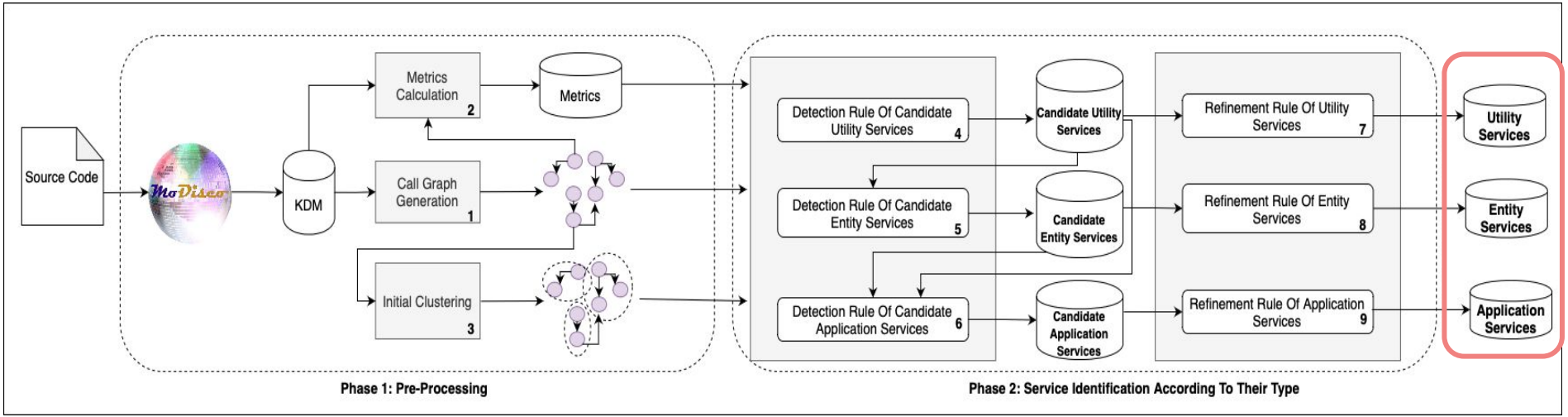
Detection of application services



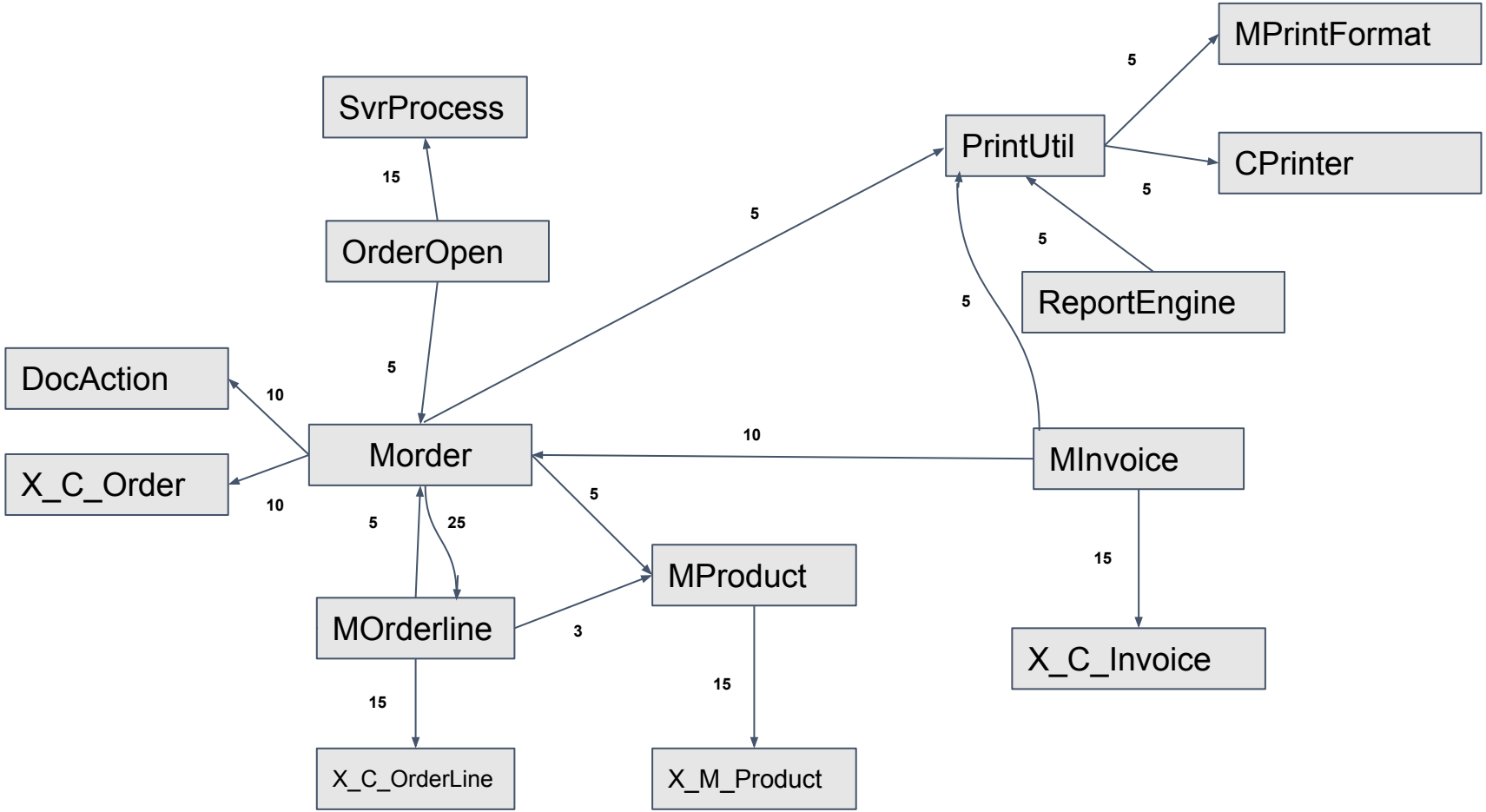
Overview of ServiceMiner



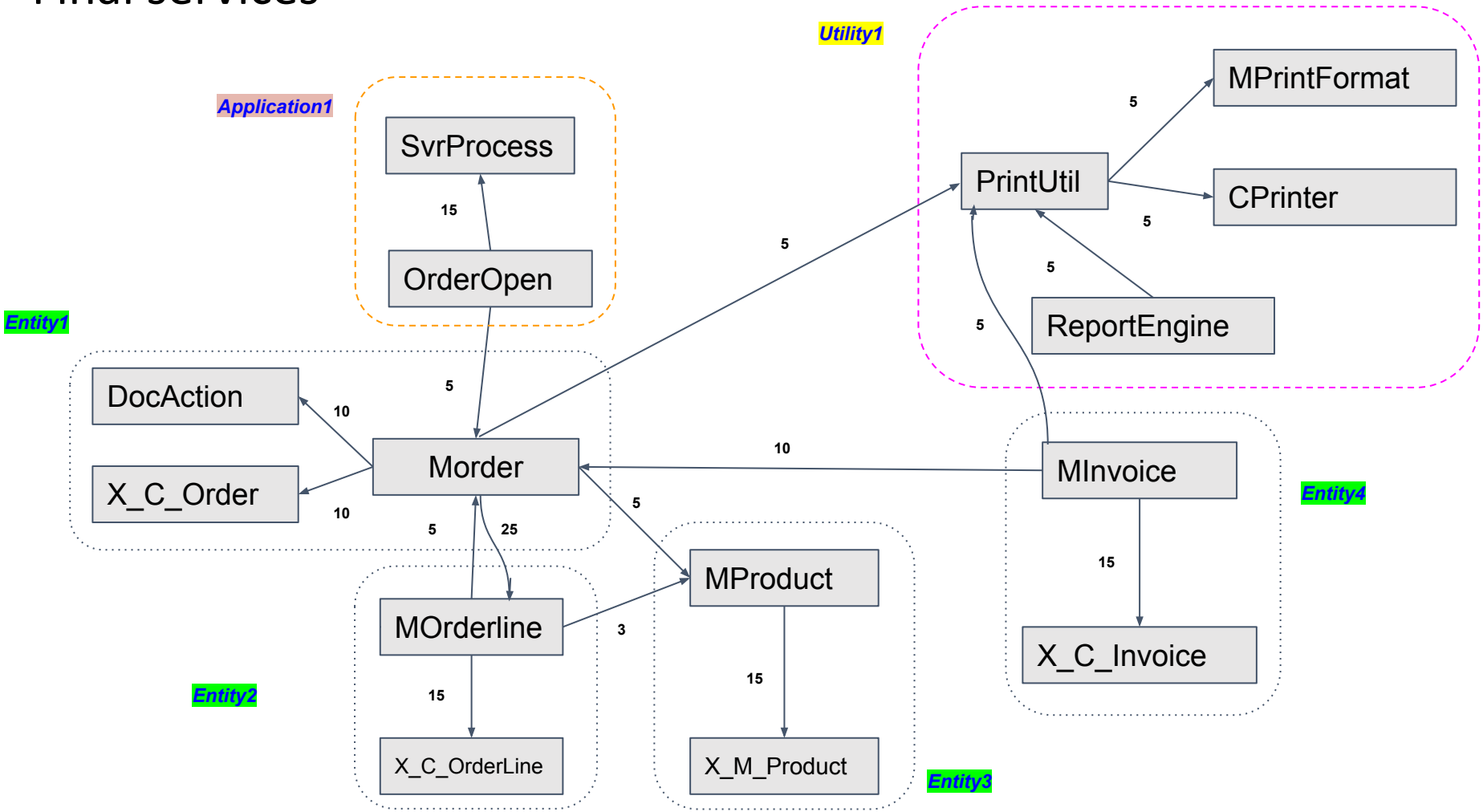
Overview of ServiceMiner



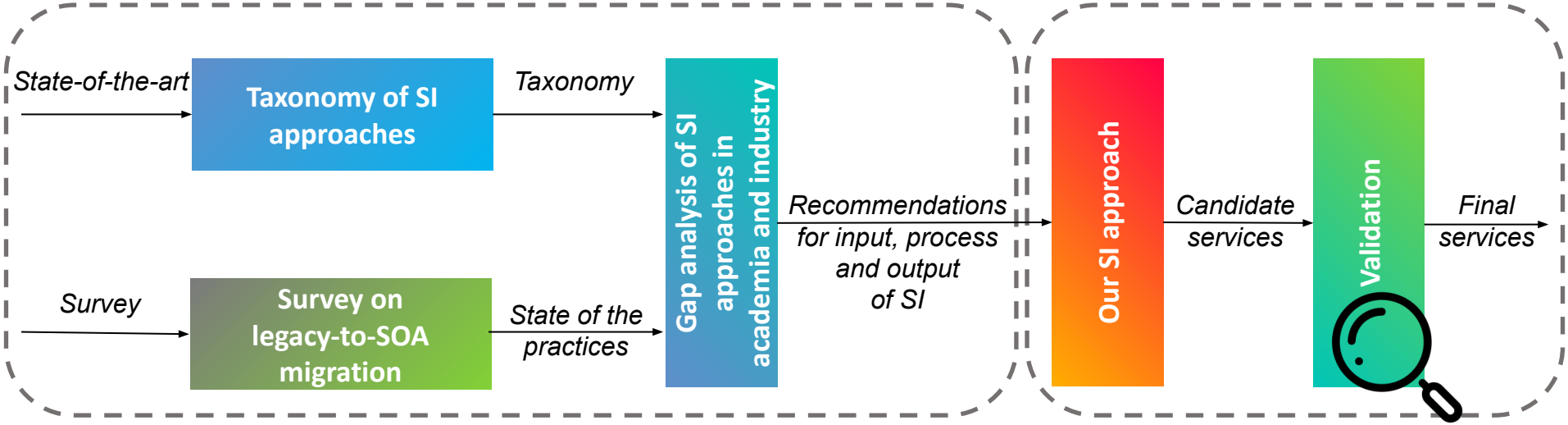
Original call graph



Final services



Research methodology



Step1: Gap analysis of SI approaches in academia and industry

Step 2: Type-sensitive service identification

Case studies

We validate our approach on **Compiere** and **POS**

- **POS** is a Java-based point of sales system
- **Compiere** is a **legacy** Java Enterprise **ERP** system
- Compiere was first introduced by **Aptean** in **2003**
- Provide **several features**
- **Not service-oriented**
- Supports different **databases**
- **2,716 classes** and **530 KLOC** in Compiere
- **54 classes** in POS

Compiere®

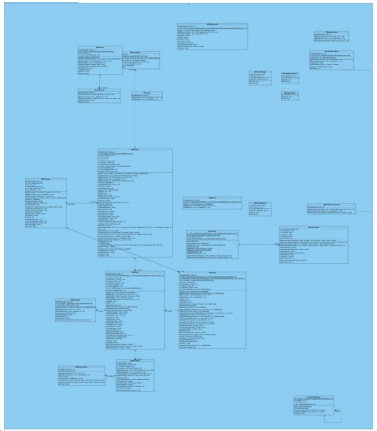
ERP

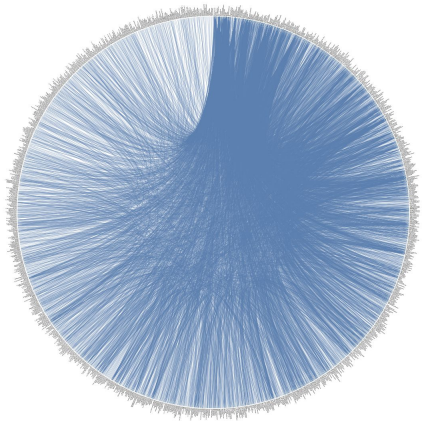
POS

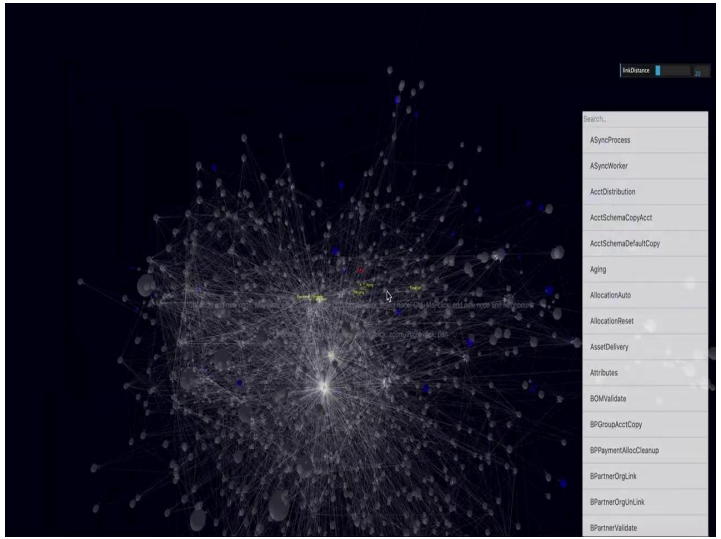
Ground truth

We build the **ground-truth** service-oriented architecture of the case studies to assess our approach.

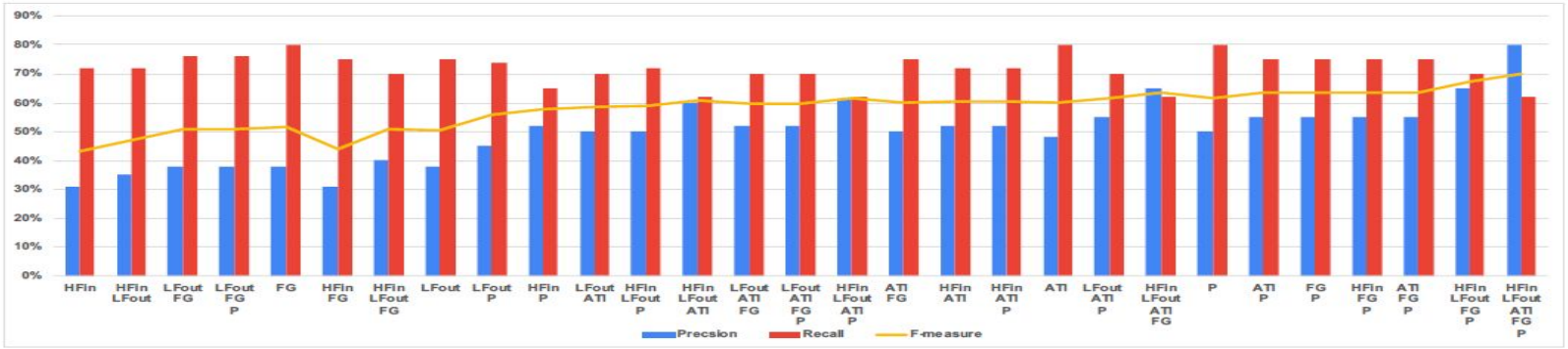
- Analyzed the systems
- Reviewed extensively the **documentations**
- Generated **several views** of the call graphs
- Found **477 services**, in Compiere and **22** in POS
- A**nnotated the services manually** according to their **type**.



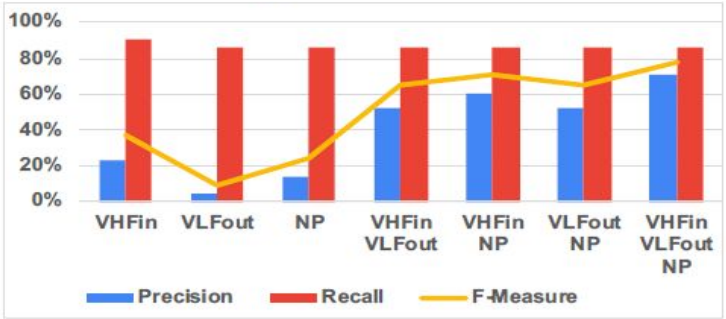




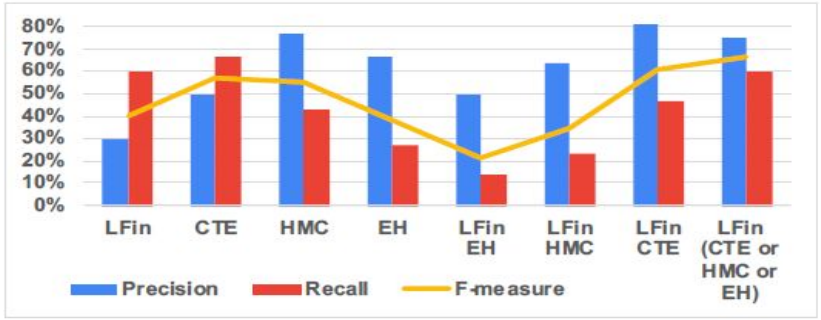
Evaluation of the detection rules



Evaluation of the detection rules of Entity Services

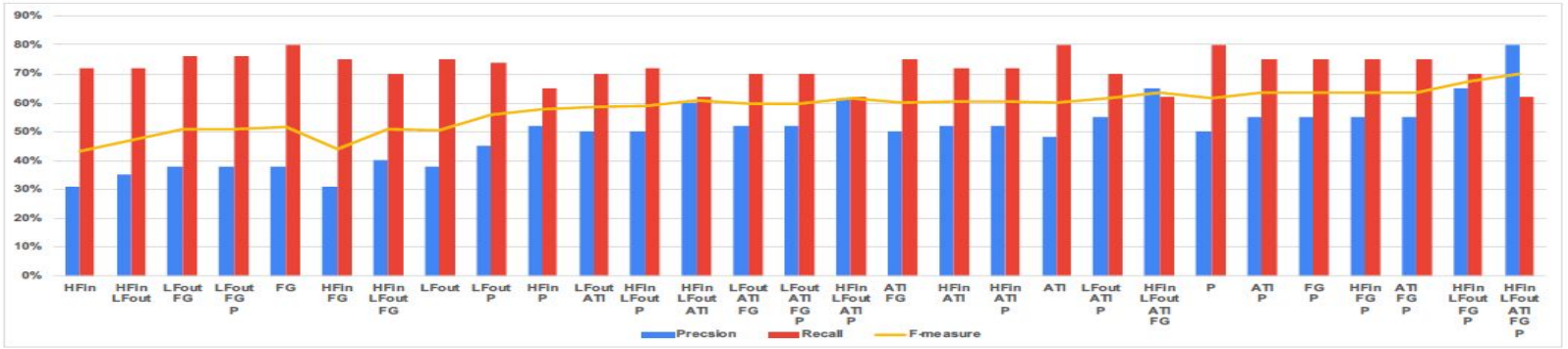


Evaluation of the detection rules of Utility Services

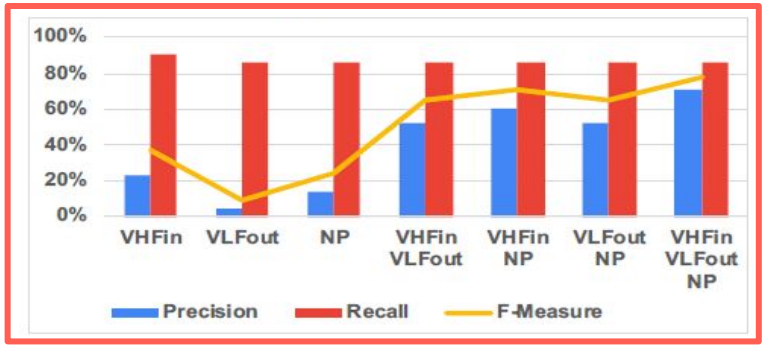


Evaluation of the detection rules of Application Services

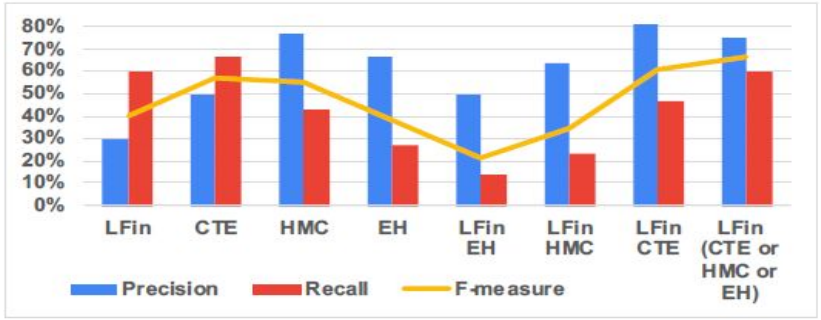
Evaluation of the detection rules



Evaluation of the detection rules of Entity Services

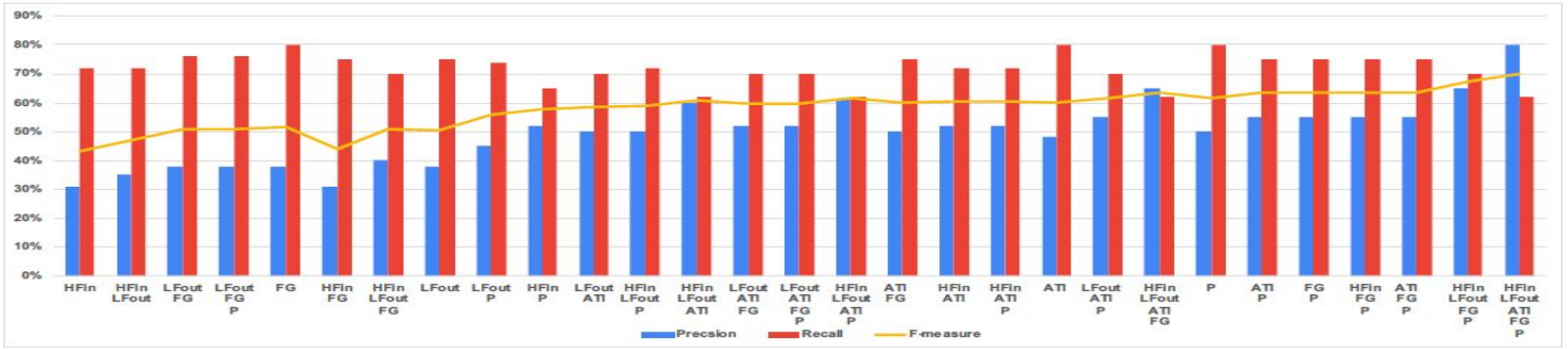


Evaluation of the detection rules of Utility Services

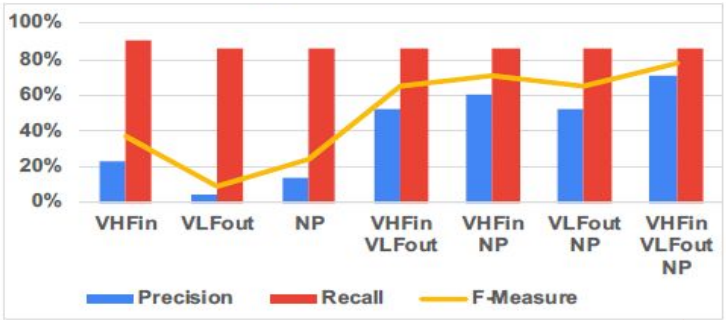


Evaluation of the detection rules of Application Services

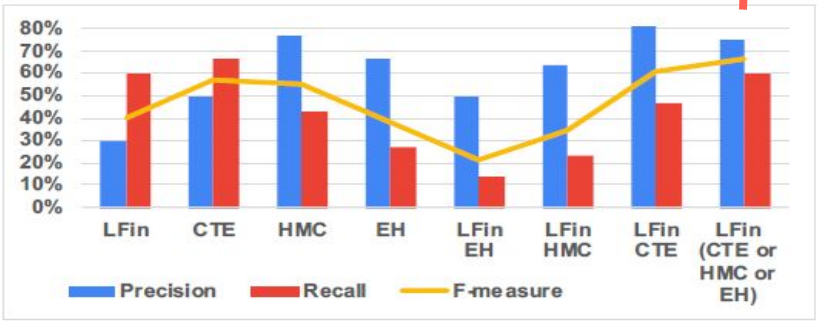
Evaluation of the detection rules



Evaluation of the detection rules of Entity Services

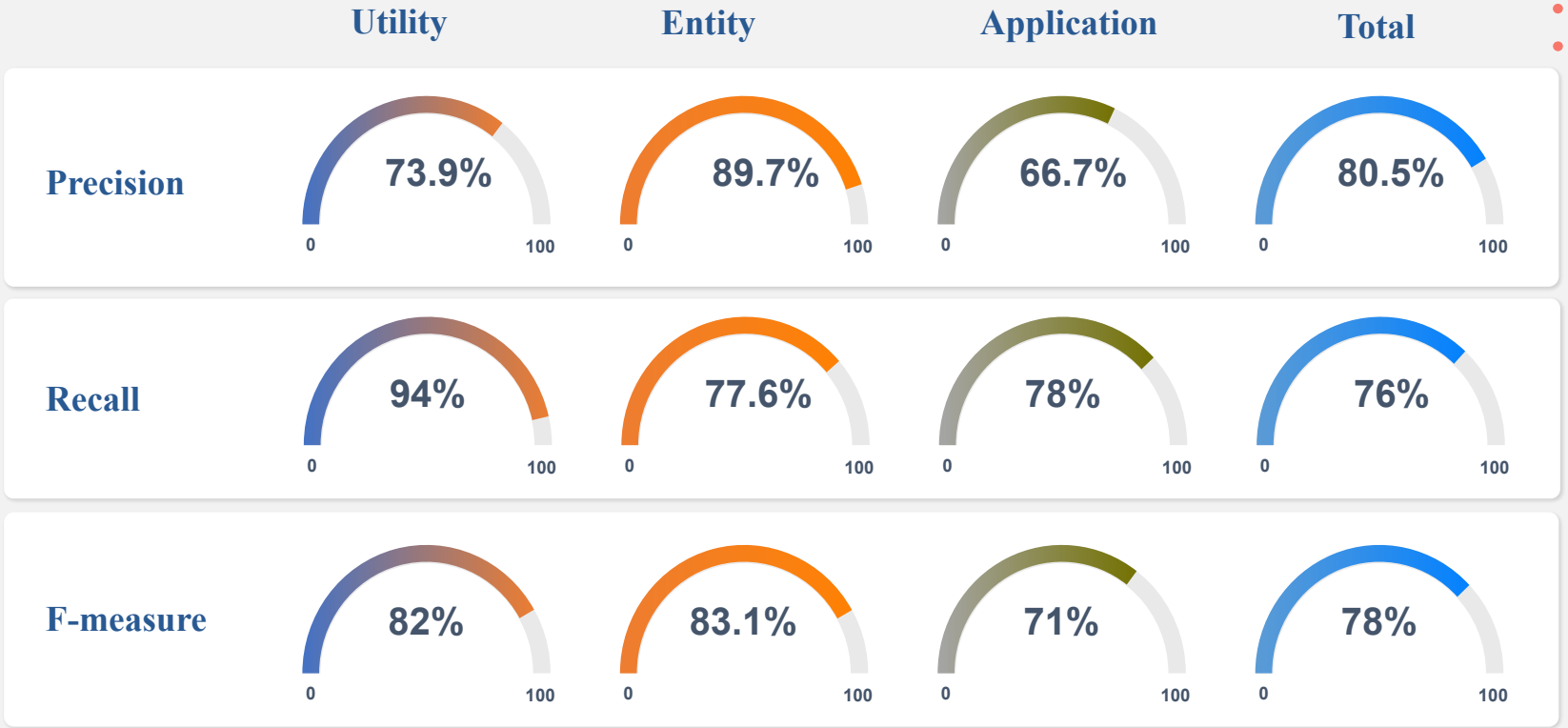


Evaluation of the detection rules of Utility Services



Evaluation of the detection rules of Application Services

Service identification accuracy



Comparison with existing SI approaches

Approach	Precision	Recall	F-measure
MOGA-WSI	10.3%	11.1%	10.6%
Service Cutter (EPL)	15.6%	21.1%	17.5%
Service Cutter (GN)	13%	11.7%	13%
Micro-Extraction	63%	31.8%	42.3%
ServiceMiner	80.5%	76%	78.2%

- **Better** identification **results** with **ServiceMiner**
- **Unbalanced** service candidates with **MOGA-WSI** and **Service Cutter**

Discussions

Type-sensitive Service Identification

Rely on real open-source post-migration SOA system

No open-source enterprise-scaled system that was **migrated** to SOA

Depending on several algorithms and thresholds

Our detection rules are **easily customized**, being **flexible** and **extensible**

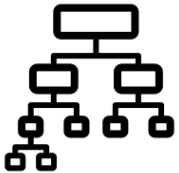
Embedded poor design in legacy systems

Reducing the **accuracy** of static-based SI approaches

Service types coverage

Extensibility to **new** services **types**

Conclusion

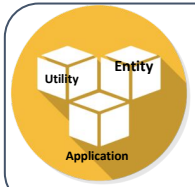


Taxonomy of service identification approaches

Study of the state of the practices of service identification in industry

Support the migration of legacy systems to SOA

Study the gap in service identification approaches in academia and industry



Type sensitive SIA that relies on static analysis of the source code of legacy systems

Conclusion

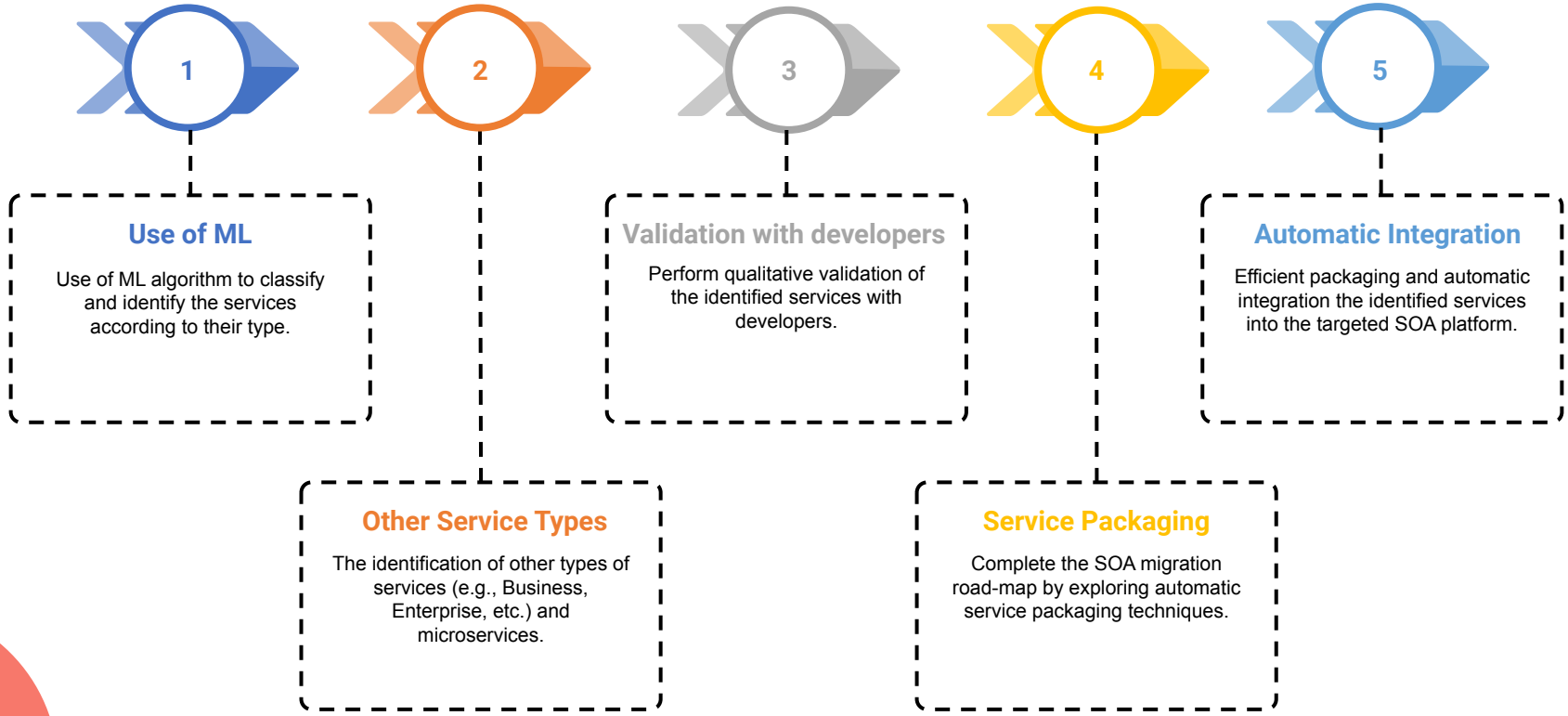
Automation of SI process

Identification of **architecturally significant** services

Possibility to **prioritize** the identification of specific **service types**

Extensibility to new technologies/languages

Future Work


1
2
3
4
5
Use of ML
Use of ML algorithm to classify and identify the services according to their type.
Validation with developers
Perform qualitative validation of the identified services with developers.
Automatic Integration
Efficient packaging and automatic integration the identified services into the targeted SOA platform.
Other Service Types
The identification of other types of services (e.g., Business, Enterprise, etc.) and microservices.
Service Packaging
Complete the SOA migration road-map by exploring automatic service packaging techniques.

Publications

1. Manel Abdellatif, Rafik Tighilt, Naouel Moha, Yann-Gaël Guéhéneuc, Hafedh Mili, Ghizlane Elboussaidi, Jean Privat: Identifying Reusable Services in Legacy Object-oriented Systems: A Type-sensitive Identification Approach. **IEE Transactions on Software Engineering 2021** (submitted)
2. Manel Abdellatif, Anas Shatnawi, Hafedh Mili, Naouel Moha, Ghizlane El-Boussaidi, Geoffrey Hecht, Jean Privat, Yann-Gaël Guéhéneuc: A taxonomy of service identification approaches for legacy software systems modernization. **J. Syst. Softw 2021,** 173: 110868
3. Manel Abdellatif, Rafik Tighilt, Naouel Moha, Hafedh Mili, Ghizlane El-Boussaidi, Jean Privat, Yann-Gaël Guéhéneuc: A Type-Sensitive Service Identification Approach for Legacy-to-SOA Migration. **ICSOC 2020**: 476-491
4. Geoffrey Hecht, Hafedh Mili, Ghizlane El-Boussaidi, Anis Boubaker, Manel Abdellatif, Yann-Gaël Guéhéneuc, Anas Shatnawi, Jean Privat, Naouel Moha: Codifying Hidden Dependencies in Legacy J2EE Applications. **APSEC 2018:** 305-314
5. Manel Abdellatif, Geoffrey Hecht, Hafedh Mili, Ghizlane El-Boussaidi, Naouel Moha, Anas Shatnawi, Jean Privat, Yann-Gaël Guéhéneuc: State of the Practice in Service Identification for SOA Migration in Industry. **ICSOC 2018**: 634-650
6. Anas Shatnawi, Hafedh Mili, Ghizlane El-Boussaidi, Anis Boubaker, Yann-Gaël Guéhéneuc, Naouel Moha, Jean Privat, Manel Abdellatif: Analyzing program dependencies in JavaEE applications. **MSR 2017**: 64-74

The following publications are not directly related to the material in this thesis, but they were produced in parallel to the research contained for this thesis.

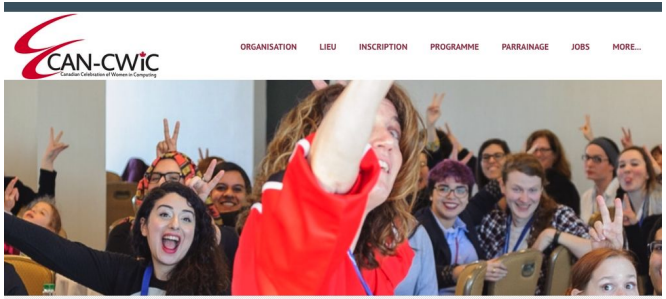
1. Manel Abdellatif, Rafik Tighilt, Abdelkarim Belkhir, Naouel Moha, Yann-Gaël Guéhéneuc, Éric Beaudry:A multi-dimensional study on the state of the practice of REST APIs us- age in Android apps. **Autom. Softw. Eng. 2020,** 27(3): 187-228
2. Rafik Tighilt, Manel Abdellatif, Naouel Moha, Hafedh Mili, Ghizlane El-Boussaidi, Jean Privat, Yann-Gaël Guéhéneuc: On the Study of Microservices Antipatterns: a Catalog Proposal. **EuroPLoP 2020:** 34:1-34:13
3. Abdelkarim Belkhir, Manel Abdellatif, Rafik Tighilt, Naouel Moha, Yann-Gaël Guéhéneuc, Éric Beaudry: An observational study on the state of REST API uses in Android mobile applications. **MOBILESoft@ICSE 2019**: 66-75

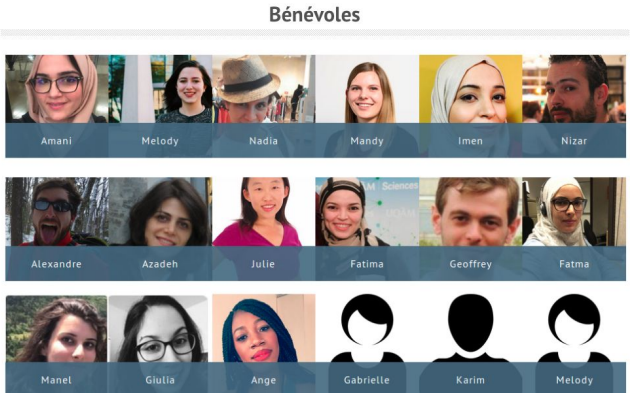
Community services

Publicity Chair
ICSOC 2019

Publicity Chair
ICSOC 2021

Student Volunteer
CAN-CWIC 2017





POLYTECHNIQUE MONTRÉAL

UQÀM

UNIVERSITÉ Concordia UNIVERSITY

ÉTS
Le génie pour l'industrie

Our Team

Manel Abdellatif
Polytechnique Montreal

Yann-Gaël Guéhéneuc
Concordia University

Naouel Moha
ÉTS

Hafedh Mili
UQÀM

Ghizlane El Boussaidi
ÉTS

Jean Privat
UQÀM

Anas Shatnawi
UQÀM

Geoffrey Hecht
UQÀM

Rafik Tighilt
UQÀM

LATECE

Ptidej

Problem

Service Identification

Is the most challenging phase of the overall modernization process

The identified services must meet a range of expectations (capability, quality, efficiency, etc.)

Gap between academia and industry
Several SI approaches have been proposed in the literature

Identification accuracy
Most of them have limited identification accuracy

Input types & availability
They usually require several types of inputs

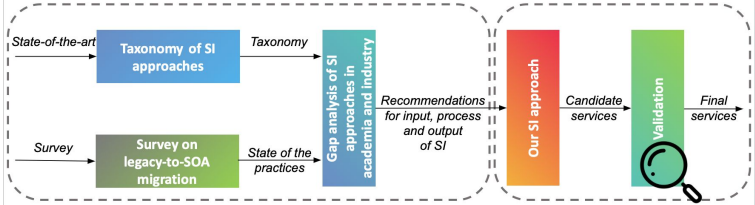
Lack of automation
Some **input types** may **not be always available** in the context of legacy systems

SI should depend on service types

- Improve the identification accuracy
- Narrowing the search space through the types and their associated code-patterns.
- Possibility to prioritize the identification of specific types of services

Propose a **highly automated type-sensitive SI** approach that relies on **static analysis** of legacy systems.

Research Methodology

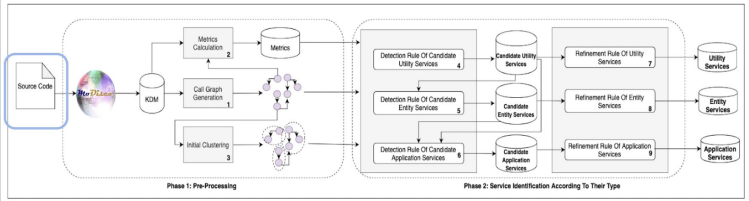


Step1: Gap analysis of SI approaches in academia and industry

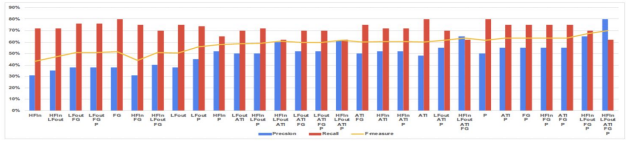
Step 2: Type-sensitive service identification

Thank you !

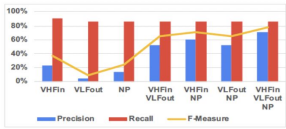
Overview of ServiceMiner



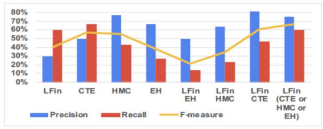
Evaluation of the Detection Rules



Evaluation of the detection rules of Entity Services



Evaluation of the detection rules of Utility Services



Evaluation of the detection rules of Application Services

References

[1] Khadka, R., Saeidi, A., Jansen, S., & Hage, J. (2013, September). A structured legacy to SOA migration process and its evaluation in practice. In 2013 IEEE 7th International Symposium on the Maintenance and Evolution of Service-Oriented and Cloud-Based Systems (pp. 2-11). IEEE.

[2] A. Marchetto and F. Ricca, “From objects to services: toward a stepwise migration approach for java applications,” STTT, vol. 11, no. 6, p. 427, 2009.

[3] M. Gysel, L. Kölbener, W. Giersche, and O. Zimmermann, “Service cutter: A systematic approach to service decomposition,” in ESOCC, 2016, pp. 185–200.

[4] H. Jain, H. Zhao, and N. R. Chinta, “A spanning tree based approach to identifying web services,” International Journal of Web Services Research, vol. 1, no. 1, p. 1, 2004.

[5] S. Alahmari, E. Zaluska, and D. De Roure, “A service identification framework for legacy system migration into soa,” in SCC. IEEE, 2010, pp. 614–617.

[6] A. Fuhr, T. Horn, and V. Riediger, “Using dynamic analysis and clustering for implementing services by reusing legacy code,” in WCRE. IEEE, 2011, pp. 275–279.

[7] R. S. Huergo, P. F. Pires, and F. C. Delicato, “Mdcsim: A method and a tool to identify services,” IT Convergence Practice, vol. 2, no. 4, pp. 1–27, 2014.

[8] M. Grieger, S. Sauer, and M. Klenke, “Architectural restructuring by semiautomatic clustering to facilitate migration towards a service-oriented architecture,” Softwaretechnik-Trends, vol. 34, no. 2, 2014.

[9] Zhuopeng Zhang, Ruimin Liu, and Hongji Yang. Service identification and packaging in service oriented reengineering. In SEKE, volume 5, pages 620–625, 2005.

References

[10] Zhuopeng Zhang and Hongji Yang. Incubating services in legacy systems for architectural migration. In 11th Asia-Pacific Software Engineering Conference, 2004., pages 196–203. IEEE, 2004.

[11] Andreas Fuhr, Tassilo Horn, and Volker Riediger. Using dynamic analysis and clustering for implementing services by reusing legacy code. In Reverse Engineering (WCRE), 2011 18th Working Conference on, pages 275–279. IEEE, 2011.

[12] Diptikalyan Saha. Service mining from legacy database applications. In Web Services (ICWS), 2015 IEEE International Conference on, pages 448–455. IEEE, 2015.

[13] Juan Manuel Rodriguez, Marco Crasso, Cristian Mateos, Alejandro Zunino, and Marcelo Campo. Bottom-up and top-down cobol system migration to web services. IEEE Internet Computing,17(2):44–51, 2013.

[14] Harry M Sneed, Chris Verhoef, and Stephan H Sneed. Reusing existing object-oriented code as web services in a soa. In Maintenance and Evolution of Service-Oriented and Cloud-Based Systems (MESOCA), 2013 IEEE 7th International Symposium on the, pages 31–39. IEEE, 2013.

[15] Lerina Aversano, Luigi Cerulo, and Ciro Palumbo. Mining candidate web services from legacy code. In 10th International Symposium on Web Site Evolution, pages 37–40. IEEE, 2008.

[16] Gerardo Canfora, Anna Rita Fasolino, Gianni Frattolillo, and Porfirio Tramontana. Migrating interactive legacy systems to web services. In Conference on Software Maintenance and Reengineering (CSMR'06), pages 10–pp. IEEE, 2006.

[17] Mostefai Abdelkader, Mimoun Malki, and Sidi Mohamed Benslimane. A heuristic approach to locate candidate web service in legacy software. International Journal of Computer Applications in Technology, 47(2-3):152–161, 2013.

[18] Mohammad Javad Amiri, Saeed Parsa, and Amir Mohammadzade Lajevardi. Multifaceted service identification: Process, requirement and data. Computer Science and Information Systems, 13(2):335–358, 2016.

References

[19] Zhuopeng Zhang, Hongji Yang, and William C Chu. Extracting reusable object-oriented legacy code segments with combined formal concept analysis and slicing techniques for service integration. In 2006 Sixth International Conference on Quality Software (QSIC'06), pages 385–392. IEEE, 2006.

[20] Concettina Del Grosso, Massimiliano Di Penta, and Ignacio Garcia-Rodriguez de Guzman. An approach for mining services in database oriented applications. In 11th European Conference on Software Maintenance and Reengineering, 2007. CSMR'07., pages 287–296. IEEE, 2007.

[21] M.A.M. Djeloul. Locating services in legacy software: information retrieval techniques, ontology and fca based approach. WSEAS Transactions on Computers, 11(1):19 – 26, 2012

[22] Seza Adjoyan, Abdelhak-Djamel Seriai, and Anas Shatnawi. Service identification based on quality metrics - object-oriented legacy system migration towards SOA. In The 26th International Conference on Software Engineering and Knowledge Engineering, Hyatt Regency, Vancouver, BC, Canada, July 1-3, 2013., pages 1–6, 2014.

[23] Hemant Jain, Huimin Zhao, and Nageswara R Chinta. A spanning tree based approach to identifying web services. International Journal of Web Services Research, 1(1):1, 2004.

[24] Cmarix, Tangible Benefits of Service Oriented Architecture: How it Works for IT Projects?, https://www.cmarix.com/blog/tangible-benefits-of-service-oriented-architecture-how-it-works-for-it-projects/, Last access on December 15th, 2020.