

Microservice Architecture Recovery

Dr Nour Ali

Keynote Talk@ Agility with Microservices Programming 2022
13th of June, 2022



nour.ali@brunel.ac.uk
<https://www.brunel.ac.uk/people/nour-ali>
<https://twitter.com/DrNourAliSE>

1

Microservice Architecture Recovery

About Me...

The timeline consists of a horizontal orange arrow pointing right. Five colored circles (orange, yellow, green, grey, blue) are connected to text boxes by dashed lines. The text boxes describe educational and professional milestones.

- B. A. in Computer Science from Bir-Zeit University Palestine.** (Orange circle)
- Research fellow at Lero-The Irish Software Engineering Research Centre** (Yellow circle)
- Senior Lecturer in Computer Science- Brunel University London and Head of Brunel Software Engineering Lab (BSEL):** <http://www.brunel-sweng.org/> (Green circle)
- PhD from Universidad Politecnica de Valencia- Spain** (Grey circle)
 - Ambient-PRISMA: Mobile Ambients in Aspect-Oriented Software Architecture
- Principal Lecturer in Software Engineering- University of Brighton.** (Blue circle)

6/28/2022 2

2

Acknowledgement

Dr Nuha Alshuqayran


3



Agenda

- Microservice Architectural Challenges
- Architecture Recovery
- MicroService Architecture Recovery (MiSAR)
- Future Directions

4



Monolithic Architecture

- Before the concept of microservices evolved, most applications were built using a monolithic architectural style.
- In **Monolithic architecture**, an application is delivered as a single deployable software artifact.

5

Monolith vs Microservices



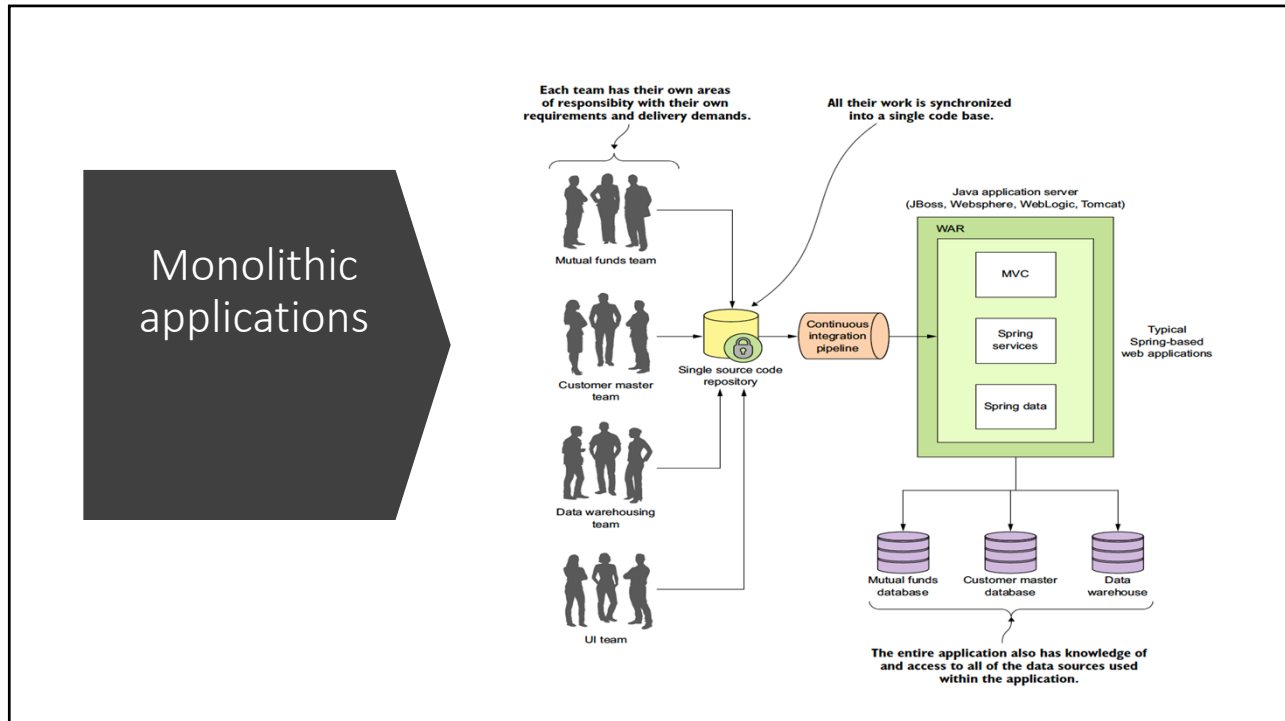
MONOLITH



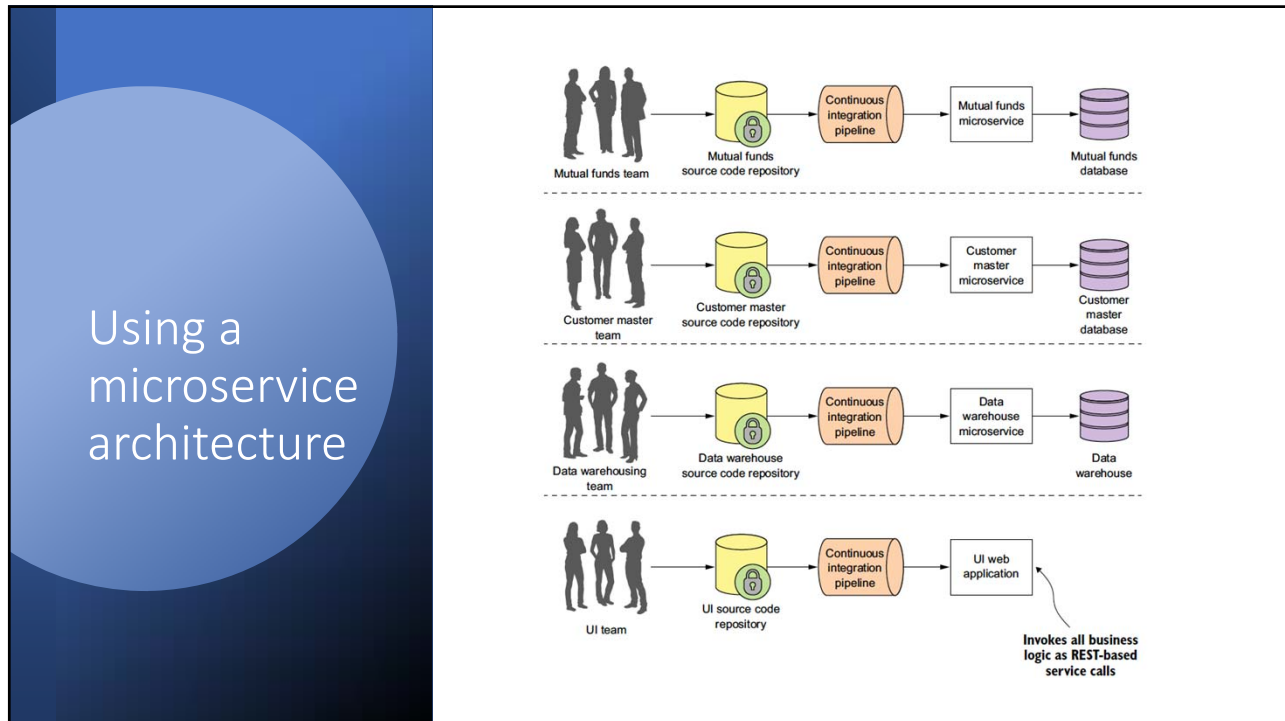
MICROSERVICES

6/28/2022 6

6



7



8

A microservice architecture characteristics:

Constrained—Microservices have a single set of responsibilities and are narrow in scope.



Abstracted—Microservices completely own their data structures and data sources.



Loosely coupled—A collection of small services interact with one another through specific interface using invocation protocol.



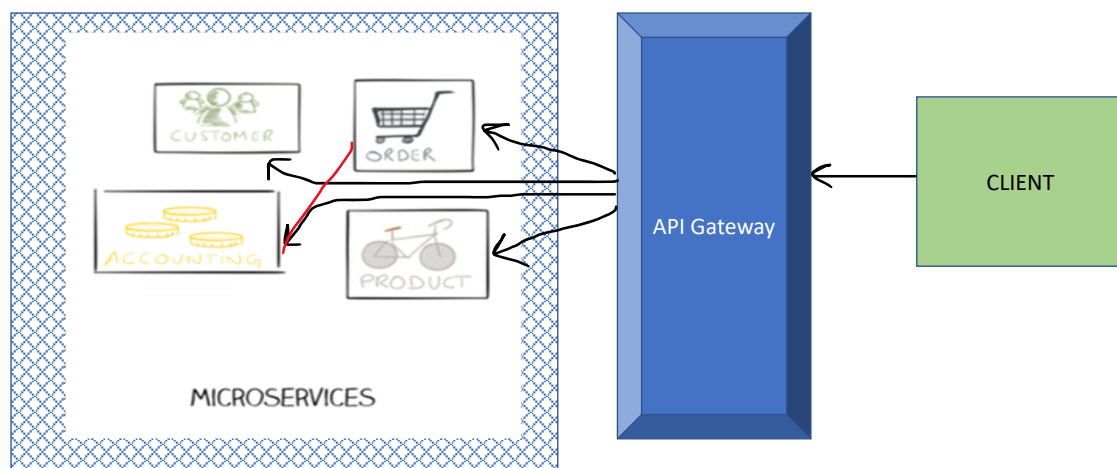
Independent—Each microservice in a microservice application can be compiled and deployed independently of the other services used in the application.



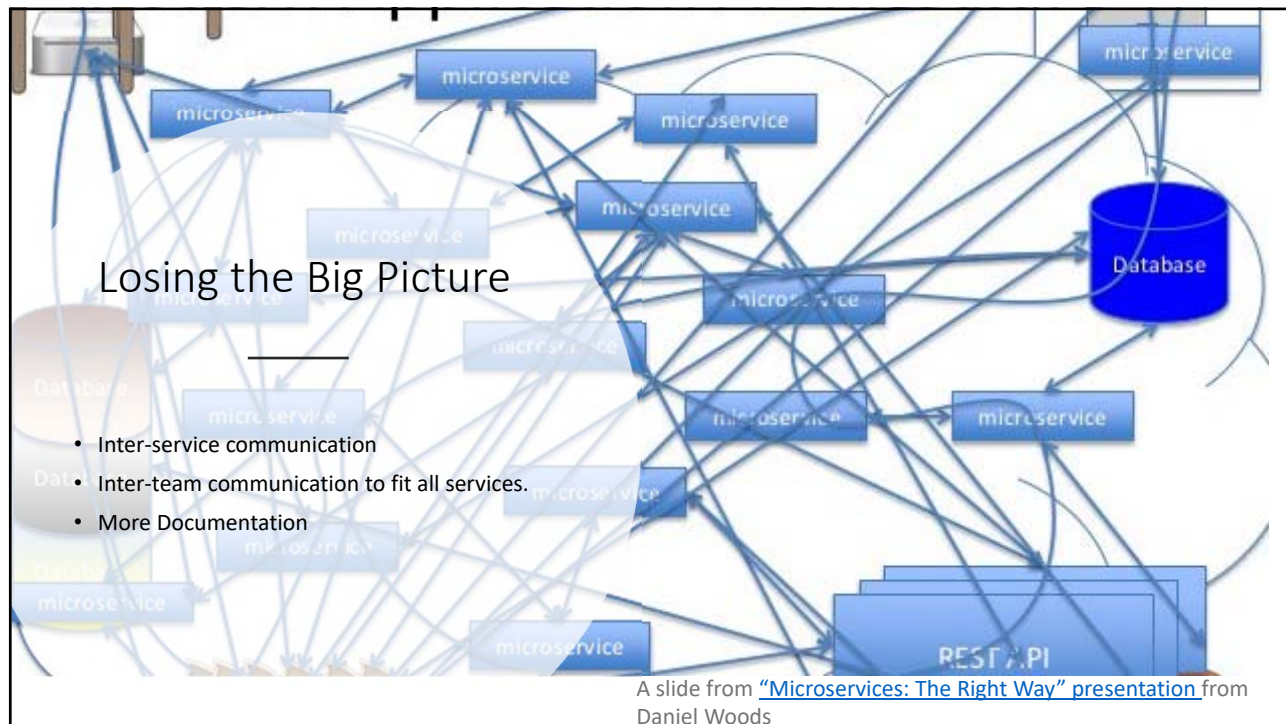
Have Small development teams with well-defined areas of responsibility.

9

Complexity



10



11

Technical Debt


- "A design or construction approach that's expedient in the short term but that creates a technical context in which the same work will cost more to do later than it would cost to do now (including increased cost over time)". McConnell (2013)

12

When poll is active, respond at pollev.com/nourali975

Technical debt is reduced in microservice systems

True False

Powered by  Poll Everywhere

Start the presentation to see live content. For screen share software, share the entire screen. Get help at pollev.com/app

13

Reasons For Microservice Technical Debt

- Bad smells and Anti-patterns (Taibi and Lenarduzzi, 2018)
- Architectural erosion: the gap between the planned and actual architecture software system Bogner et al. (2018)
- Postponed architectural decisions Bogner et al. (2019)
- Increased Dependencies between services de Toledo et al. (2019)
- Technical Debt was increased during the migration activities to a microservice architecture Lenarduzzi et al (2020)
- Continuous delivery a principle in microservice architecture (ME)

14

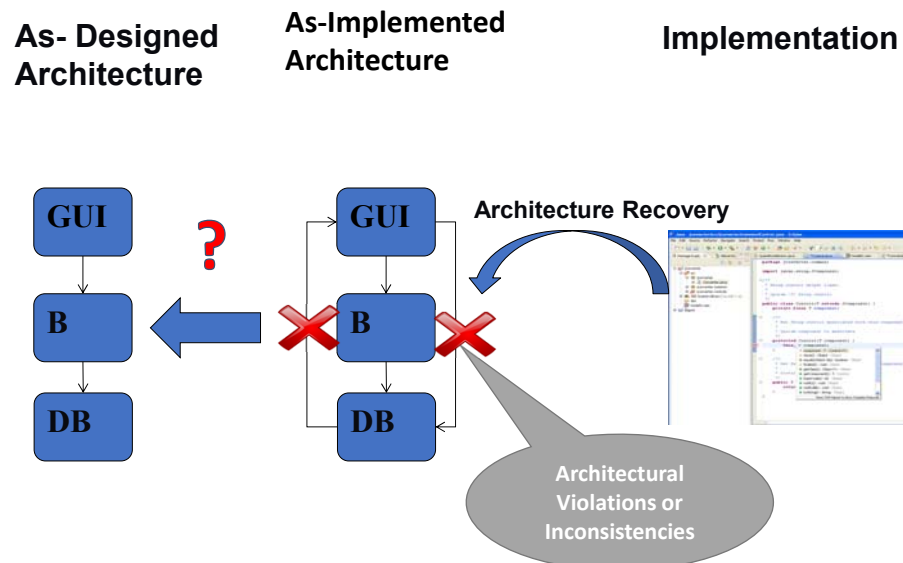


Agenda

- Microservice Architectural Challenges
- Architecture Recovery
- MicroService Architecture Recovery (MiSAR)
- Future Directions

15

Architecture Recovery and Erosion



16



6/28/2022

17

Architectural Drift/Erosion

17

Alternatives for Architecture Recovery

- We interviewed 19 senior software engineers from 17 different companies.
- RQ1: What practices are currently in place in practitioners' companies to address AC?
 - INFORMAL PRACTICES:
 - Code Reviews
 - Tools for ensuring rules are fulfilled in code.
 - Using naming conventions
 - Architecture Communication:
 - The usage of wikis, skype, forums and sometimes training to share architecture knowledge;
 - Conducting Scrum meetings.
 - Having an architecture steering committee within the company that is aware.
 - Providing heavyweight architectural documentation.

Ali, N., Baker, S., O'Crowley, R. et al. Architecture consistency: State of the practice, challenges and requirements. Empir Software Eng 23, 224–258 (2018).

18

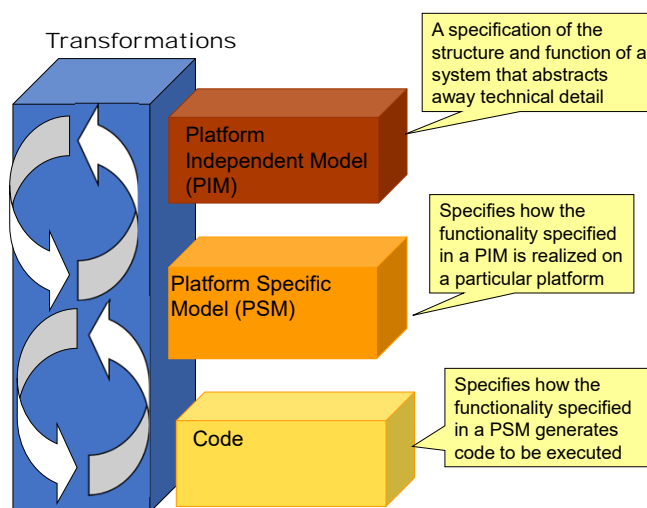
Advantages for Architecture Recovery and Consistency Awareness

- RQ3: Which software development situations would practitioners envision for AC approaches to be useful?
 - Increasing Architectural Knowledge Awareness
 - Stopping Inconsistency Introduction
 - Auditing In house and for Outsourcing
 - Evolving the Architecture.

Ali, N., Baker, S., O’Crowley, R. *et al.* Architecture consistency: State of the practice, challenges and requirements. *Empir Software Eng* **23**, 224–258 (2018).

19

Model Driven Architecture



20



Agenda

- Microservice Architectural Challenges
- Architecture Recovery
- MicroService Architecture Recovery (MiSAR)
- Future Directions

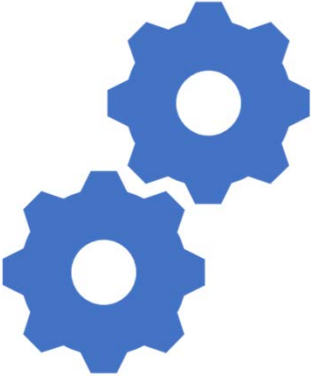
21

Research Problems



- ❑ Microservice architecture composed of many microservices that are dynamic, small, distributed and operated by multiple teams.
- ❑ Microservices are developed quickly and provides more agility of the system, which result in **continuous architectural changes**.
- ❑ Software architects may lose the knowledge of their architectural systems, and often the documentation of the architecture is not kept up to date.

22



Solution: Microservice Architecture Recovery

- Architecture recovery is a promising approach to aiding comprehension of the complexity of microservice architecture in a way that allows developers/architects to understand an architecture's implemented structure.
- We defined Microservice Architectural Recovery (MiSAR) approach, which supports the recovery of architectural models of microservice systems and that can unveil their architectural aspects.

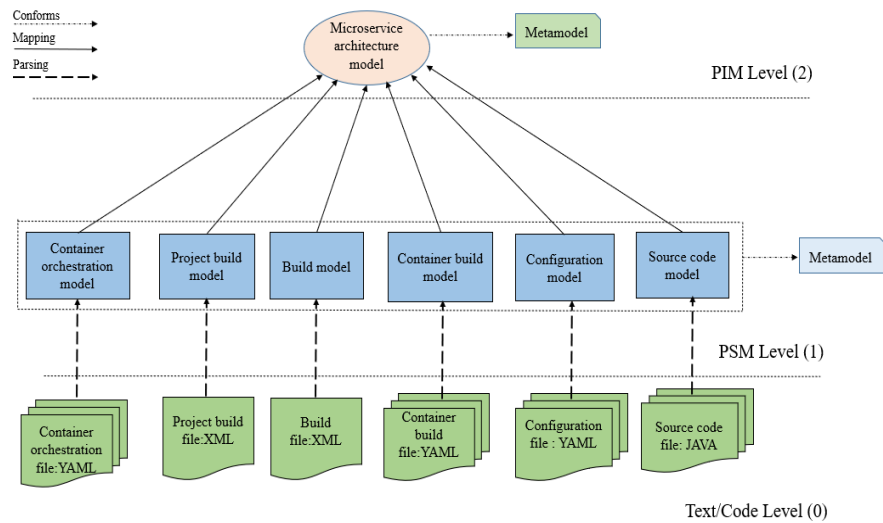
23

Microservice Architecture Recovery (MiSAR)

- MiSAR** is an approach which supports the recovery of architectural models of microservice systems and that can unveil their architectural aspects.
- The approach aims to recover the architecture of microservice-based systems from the **implementation** level to the **architecture level**.
- MiSAR follows a Model-Driven Architecture (MDA) framework.
- MiSAR was developed from empirical data to define **metamodels** and the **mapping rules** that support the architectural recovery of a microservice system.

24

MiSAR abstraction Levels



25

Empirical studies to define MiSAR



Study 1: The aim is to identify the concepts needed to build a metamodel of the microservice-based system, and to develop mapping rules that derive a target model from the source model.



Study 2: This study focuses on validating and enhancing (metamodel and mapping rules) incrementally and achieve improved artefacts for architecture recovery.

26

Empirical study 1

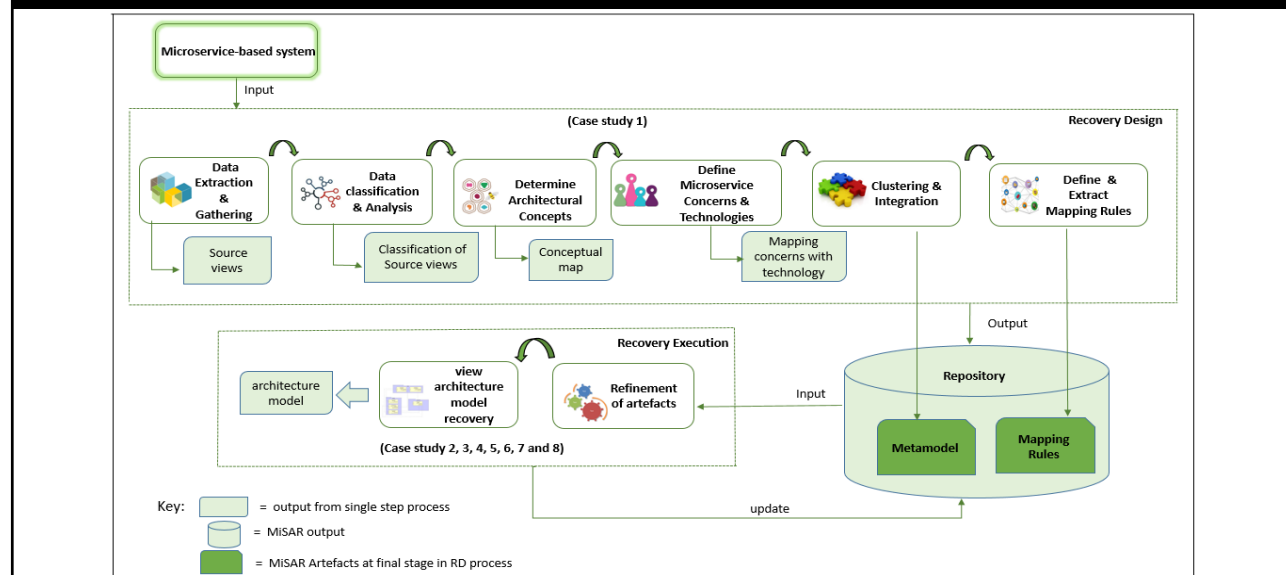
An empirical study to define an initial version of the MiSAR artefacts: the **metamodel** and **mapping rules**.

We selected **8 open-source projects** from the GitHub repository that employed microservice architecture.

Was designed as a manual architecture recovery process, which includes two main phases: **Recovery Design** and **Recovery Execution**.

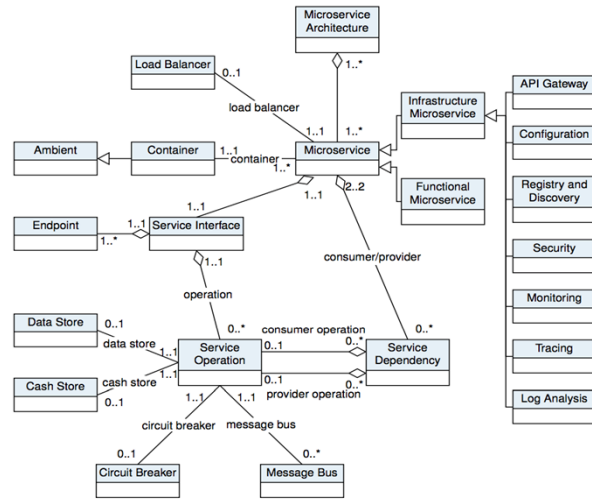
27

Empirical Study 1 steps.



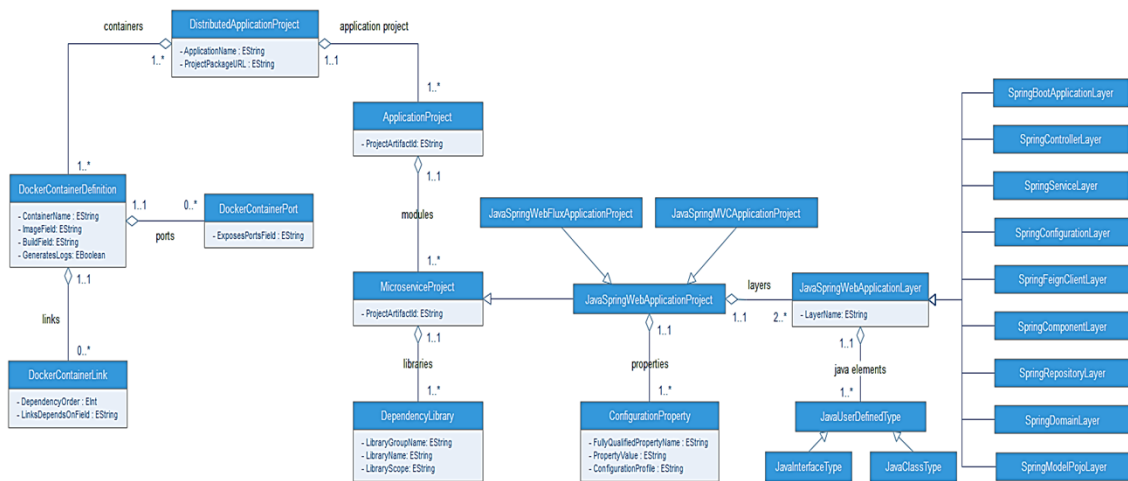
28

Results: RQ1(PIM metamodel)



29

PSM metamodel (Java, Spring boot/ Cloud framework)



Java PSM metamodel is reduced)

30

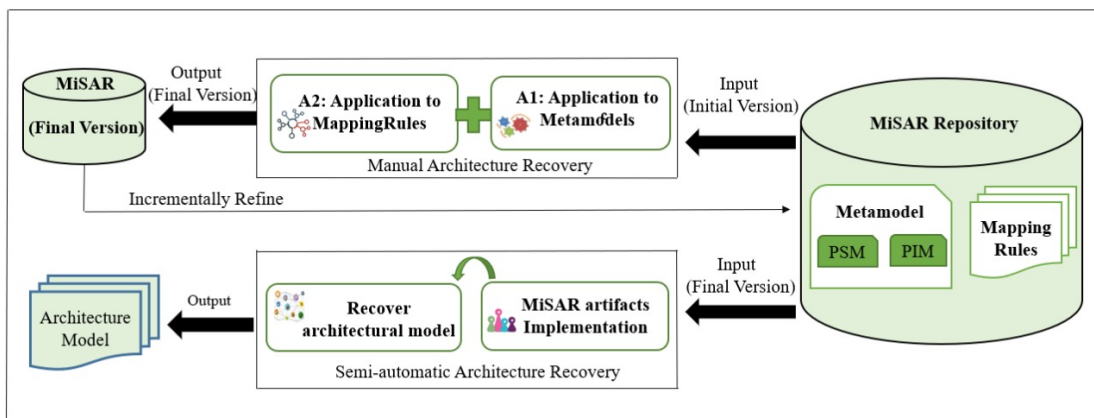
Empirical study 2

We designed a new empirical study based on 9 microservice projects.

The aim was to define the final MiSAR artefacts in order to be able to generate architectural models of implemented microservice systems.

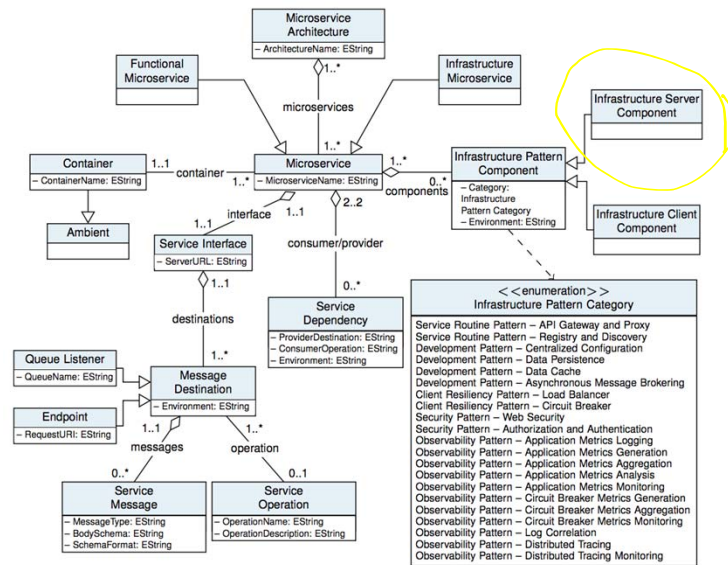
31

Empirical Study 2 steps



32

Results: MiSAR PIM Metamodel



33

MiSAR Implementation



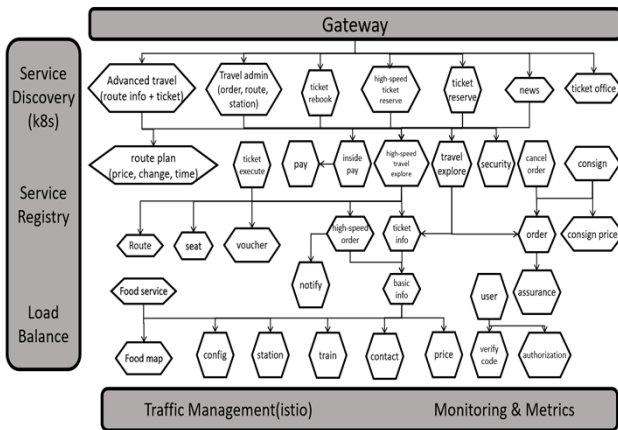
Metamodels were implemented as Ecore models using the Eclipse Modelling Framework (EMF).



Mapping rules were implemented by operational QVT transformation language (QVTo).

34

TrainTicket :a Case Study



- **TrainTicket**: is a train ticket booking system based on a microservice architecture.

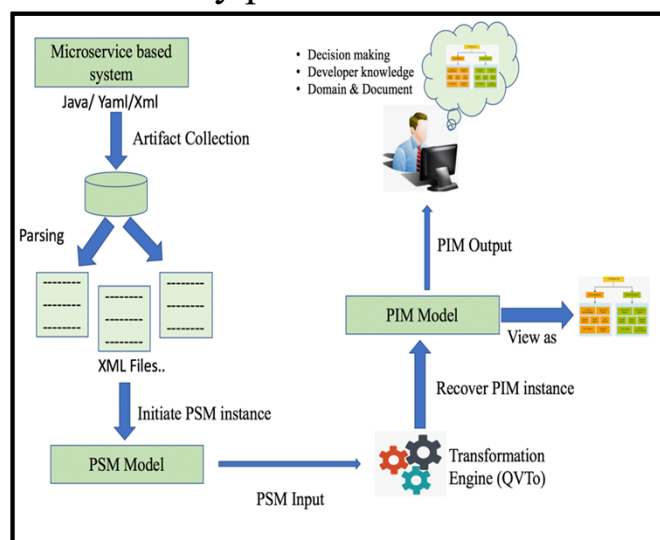
- **Large-size benchmark microservice system**: The case study has 69 microservices.

- **The objective**: is to evaluate the MiSAR approach in terms of recovering an architectural model of a microservice system

35

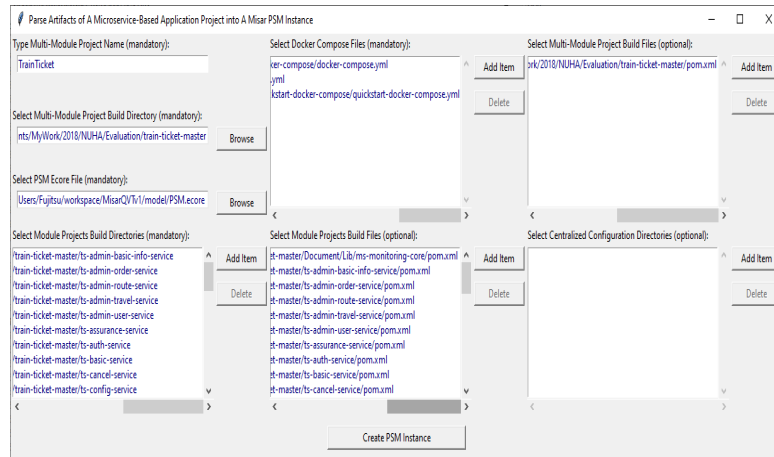
Steps of the MiSAR architecture recovery process:

- **Step 1** – Artefact collection (semi-automatic)
- **Step 2** – Instantiate PSM instance (automatic)
- **Step 3** – Recover PIM instance (automatic)



36

Step 1- Artefact collection



37

Step 2- Instantiate PSM instance



38

Step 3- Recover PIM instance



39

Results (architecture recovery)

(Successful recovery)

- We have recovered **36** instances of **functional microservices**.
- We have recovered **27** instances of **infrastructure microservices**.




(Partial recovery)

- We have recovered **6** instances of the supertype microservice.

Reason: the source artefacts of these microservices belong to non- JVM projects.

40

Consistency Checking:
(between the MiSAR Recovered Model and the Documentation)

- 
Additional elements: MiSAR recovered more architectural elements compared to the documentation.
- 
Inconsistent elements: There were recovered elements in MiSAR which were inconsistent with the ones represented in the documentation
- 
Missed elements: An example of a missed component is related to tracing infrastructure, which according to the documentation implements Jaeger. MiSAR does not yet support Jaeger in its repository of mapping rules.

41

Positive Aspects of MiSAR

- Support the consistency/conformance checking:**
 - We identified that the TrainTicket documentation diverged from its actual implementation.
 - MiSAR architecture recovery approach can obtain an up-to-date as implemented architectural model.
- Architectural Expressiveness:** MiSAR was found to recover extra architecture elements compared to the documentation.
- The Ability of MiSAR to Discover the Existence of Non-JVM Applications.**

42



Agenda

- Microservice Architectural Challenges
- Architecture Recovery
- MicroService Architecture Recovery (MiSAR)
- Future Directions

43

Future of Microservice Architecture Recovery

- Enrich/Complement the Architectural models with:
 - Economics.
 - Quality attribute analysis
 - Security property checks in architectural models
 - Conformance/Consistency Checking
 - Auditing

44

Thank you

Questions?

Dr. Nour Ali

Home page: <http://www.brunel.ac.uk/people/nour-ali/>

Email: Nour.Ali@brunel.ac.uk

Twitter <https://twitter.com/DrNourAliSE>

