UNIVERSITÀ DEGLI STUDI DI MILANO BICOCCA

DEPLOYMENT-AWARE ANALYSIS OF MICROSERVICE-BASED APPLICATIONS

Davide Rendina

Relatori: Francesca Arcelli Fontana, Antonio Brogi, Jacopo Soldani

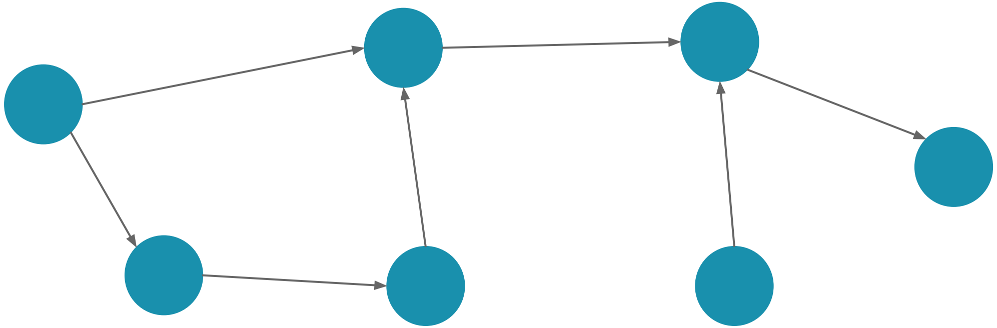
ESSeRE Lab

SOCC

01

INTRODUCTION

MICROSERVICE SYSTEMS



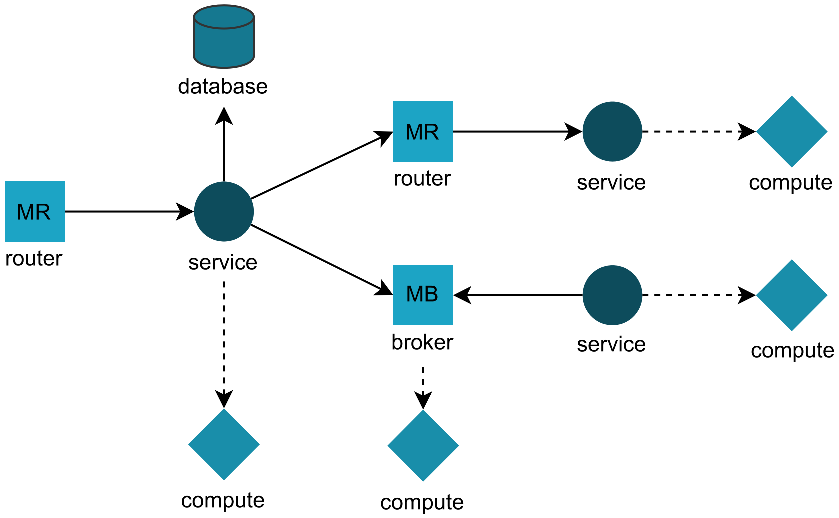
MAIN TECHNOLOGIES: Container & Kubernetes

ARCHITECTURAL SMELL

> "An **architectural smell** is a commonly used architectural decision that negatively impacts system quality"
>
> **– J. Garcia**

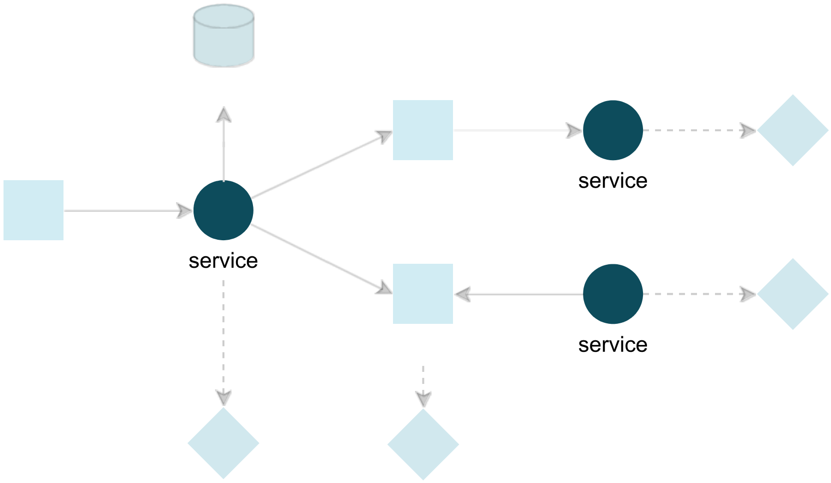
µTOSCA MODEL

- Language for **modelling** microservice architectures



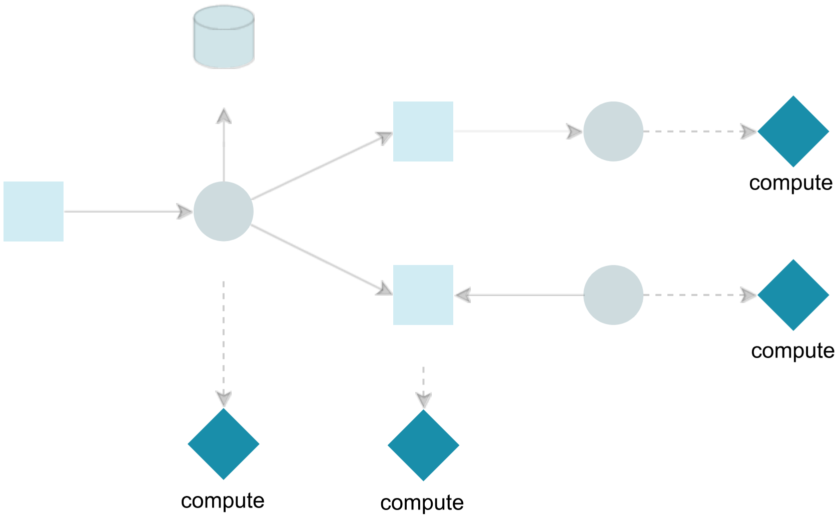
µTOSCA MODEL

- Language for **modelling** microservice architectures



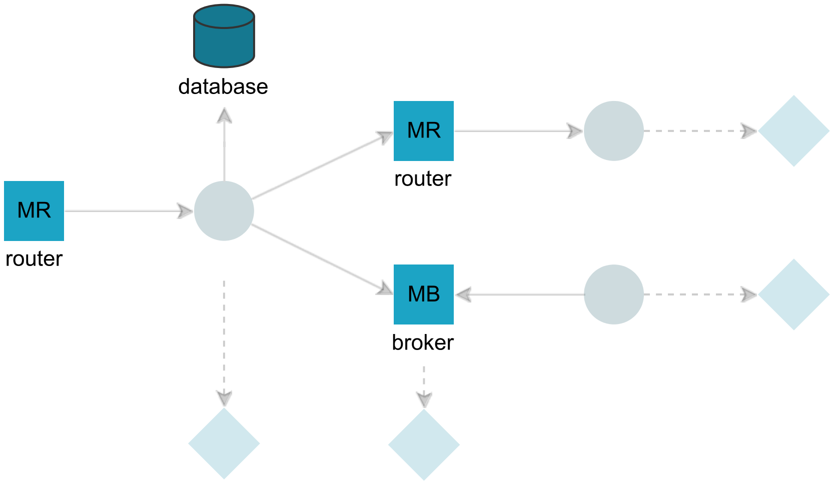
µTOSCA MODEL

- Language for **modelling** microservice architectures



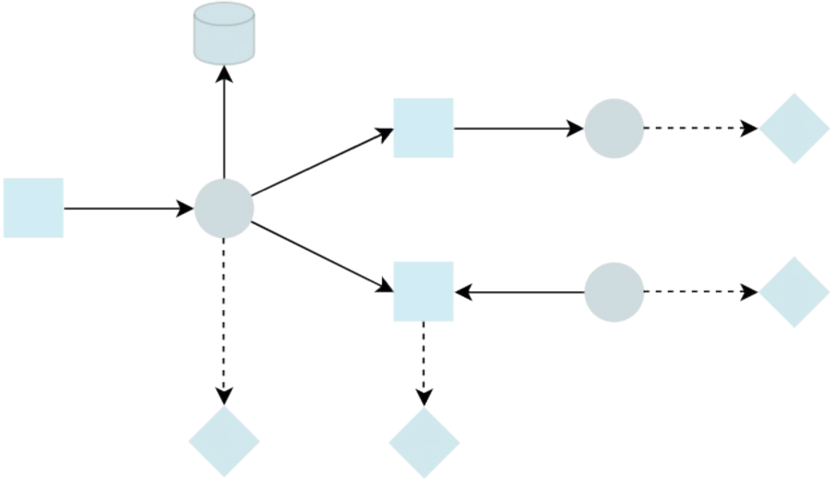
µTOSCA MODEL

- Language for **modelling** microservice architectures



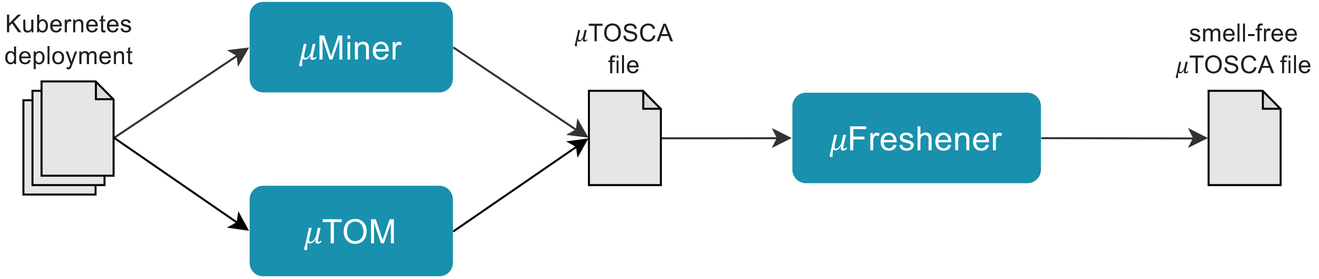
µTOSCA MODEL

- Language for **modelling** microservice architectures



µTOSCA TOOLCHAIN

- Set of tools for **mining** microservice architectures, **detecting** 4 architectural smells, and **refactoring** the µTOSCA model



02

TOOLCHAIN UPDATE

RESEARCH OBJECTIVE

DETECTION

Detection of a new architectural smell

RESEARCH OBJECTIVE

DETECTION

Detection of a new architectural smell

EXTENSION

Correction and **enrichment** of the µTOSCA model

RESEARCH OBJECTIVE

DETECTION

Detection of a new architectural smell

EXTENSION

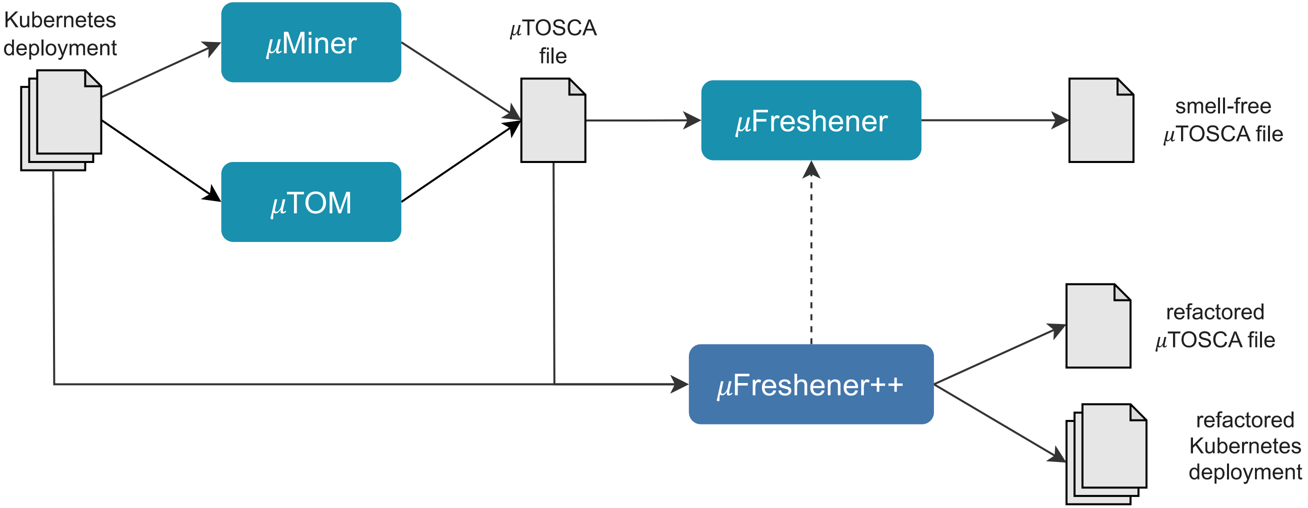
Correction and **enrichment** of the µTOSCA model

REFACTORING

Support for automatic **refactoring** of smells

TOOLCHAIN UPDATE

- Introduced the **µFreshener++** tool



DETECTION

WHAT

Detection of the **Multiple Services in One Pod** architecture smell

DETECTION

WHAT

Detection of the **Multiple Services in One Pod** architecture smell

HOW

Search for a specific **pattern** of nodes in the graph

DETECTION

WHAT

Detection of the **Multiple Services in One Pod** architecture smell

HOW

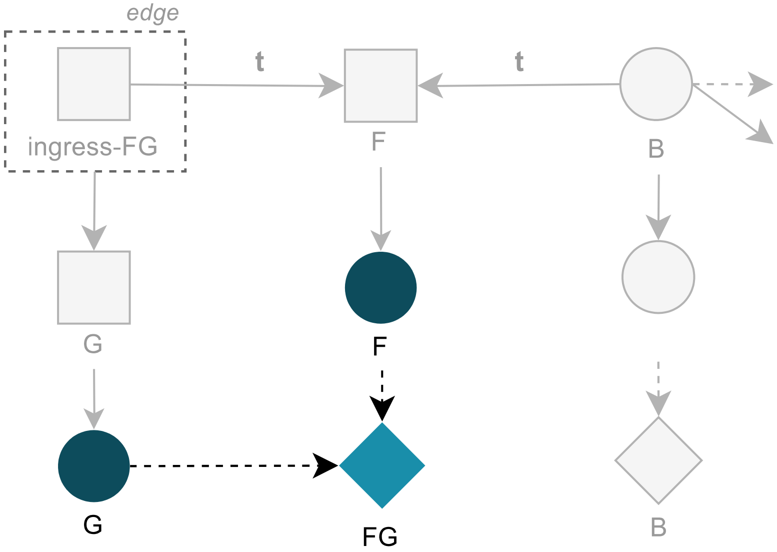
Search for a specific **pattern** of nodes in the graph

GOAL

Extend the number of **smell recognized** by the toolchain

DETECTION

- Search for compute nodes with **more than two edges** incoming



EXTENSION

WHAT

Correction and **enrichment** of the µTOSCA model

EXTENSION

WHAT

Correction and **enrichment** of the µTOSCA model

HOW

Adding, removing and modifying **nodes** of the model

EXTENSION

WHAT

Correction and **enrichment** of the µTOSCA model

HOW

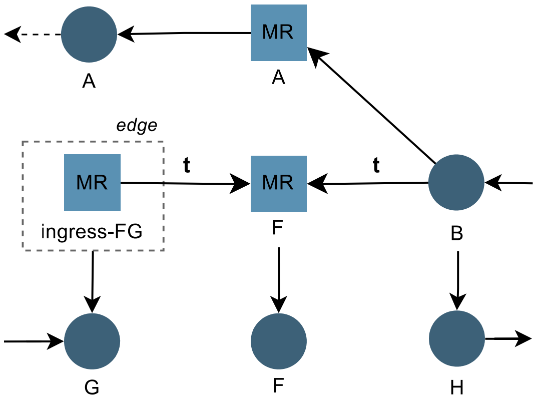
Adding, removing and modifying **nodes** of the model

GOAL

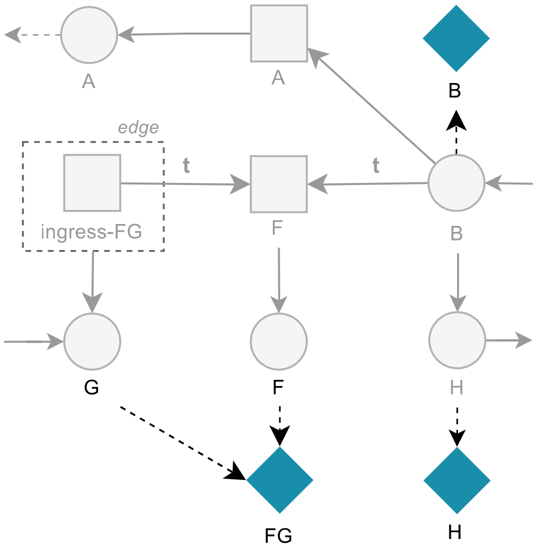
Improve the µTOSCA toolchain **smell detection**

EXTENSION

BEFORE



AFTER



REFACTORING

WHAT

Support for automatic **refactoring** of architectural smells

REFACTORING

WHAT

Support for automatic **refactoring** of architectural smells

HOW

Adding, modifying and removing **Kubernetes** resources

REFACTORING

WHAT

Support for automatic **refactoring** of architectural smells

HOW

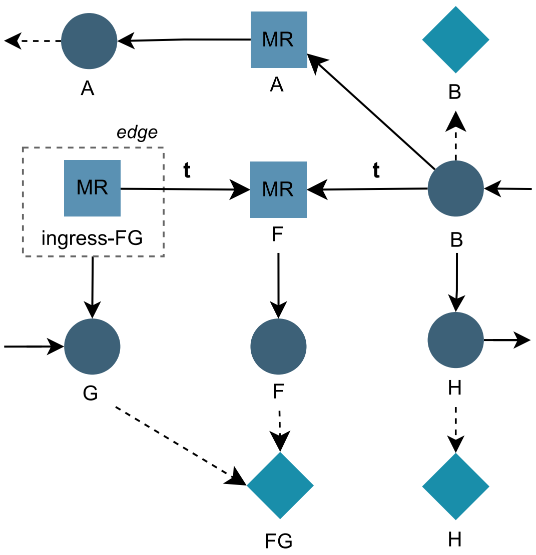
Adding, modifying and removing **Kubernetes** resources

GOAL

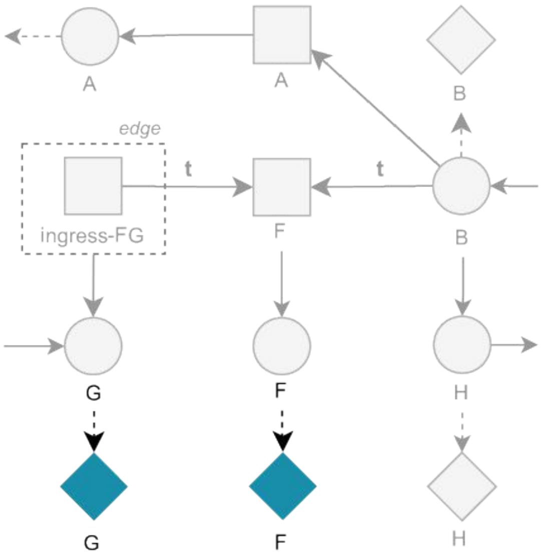
Improve the actual **refactoring** applied by the toolchain

REFACTORING

BEFORE

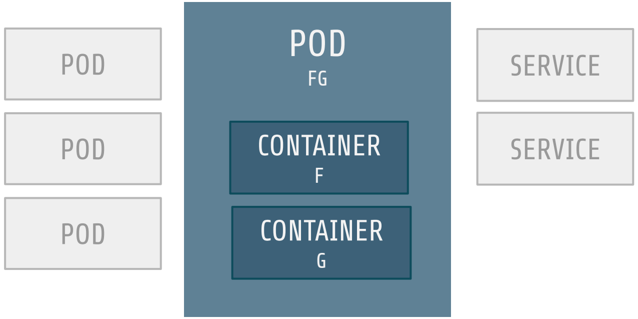


AFTER

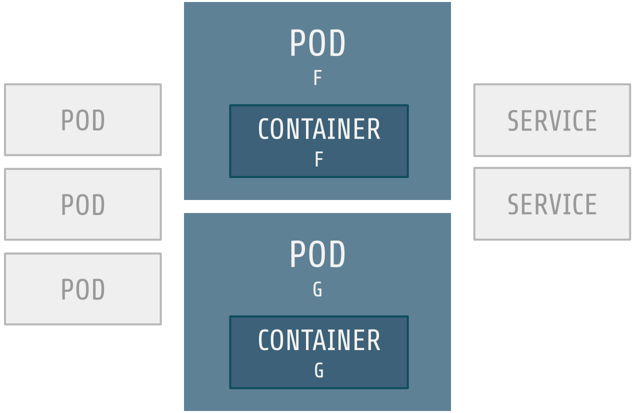


REFACTORING

BEFORE



AFTER



03

VALIDATION

VALIDATION

🔍 DETECTION

Validation of **Multiple Services in One Pod** smell detection

VALIDATION

DETECTION

Validation of **Multiple Services in One Pod** smell detection

EXTENSION

Validation of µTOSCA model **extension**

VALIDATION

DETECTION

Validation of **Multiple Services in One Pod** smell detection

EXTENSION

Validation of μTOSCA model **extension**

IMPACT

Study of the **extension impact** on smell detection

VALIDATION

DETECTION

Validation of **Multiple Services in One Pod** smell detection

EXTENSION

Validation of µTOSCA model **extension**

IMPACT

Study of the **extension impact** on smell detection

REFACTORING

Validation of architectural smell automatic **refactoring**

DETECTION

METHODOLOGY

- Run smell detector
- Analyze detected smells

DETECTION

METHODOLOGY

- Run smell detector
- Analyze detected smells

RESULTS

5 Instances detected

3 True positives

2 False positives

EXTENSION

METHODOLOGY

- Issues injection
- Run extension
- Verify issues
- Compare graphs

EXTENSION

METHODOLOGY

- Issues injection
- Run extension
- Verify issues
- Compare graphs

RESULTS

92% Issues solved (23/25)

100% Extension controls tested

IMPACT

METHODOLOGY

- Issues injection
- Detection on both models
- Compare results

IMPACT

METHODOLOGY

- Issues injection
- Detection on both models
- Compare results

RESULTS

9 False positive removed (9/19)

5 True negatives detected

REFACTORING

METHODOLOGY

- Smell detection
- Apply refactoring
- Verify applied refactoring
- Analyze resources

REFACTORING

METHODOLOGY

- Smell detection
- Apply refactoring
- Verify applied refactoring
- Analyze resources

RESULTS

100% Smells refactored (19/19)

FUTURE WORKS

IMPROVEMENT

Of the features implemented

FUTURE WORKS

IMPROVEMENT

Of the features implemented

VALIDATION

On a running system

FUTURE WORKS

IMPROVEMENT

Of the features implemented

VALIDATION

On a running system

SUPPORT

For others deployment system

FUTURE WORKS

IMPROVEMENT

Of the features implemented

VALIDATION

On a running system

SUPPORT

For others deployment system

EXTENSION

To detect and refactor other types of smells

Thank you

Davide Rendina

CREDITS: Template by **Slidesgo** and icons by **Icons8**