University of Salerno

sesa lab
SOFTWARE ENGINEERING SALERNO

JADS Jheronimus Academy of Data Science

Infrastructure Intent Discovery via TOSCA-based Reverse Engineering

October 24, 2025

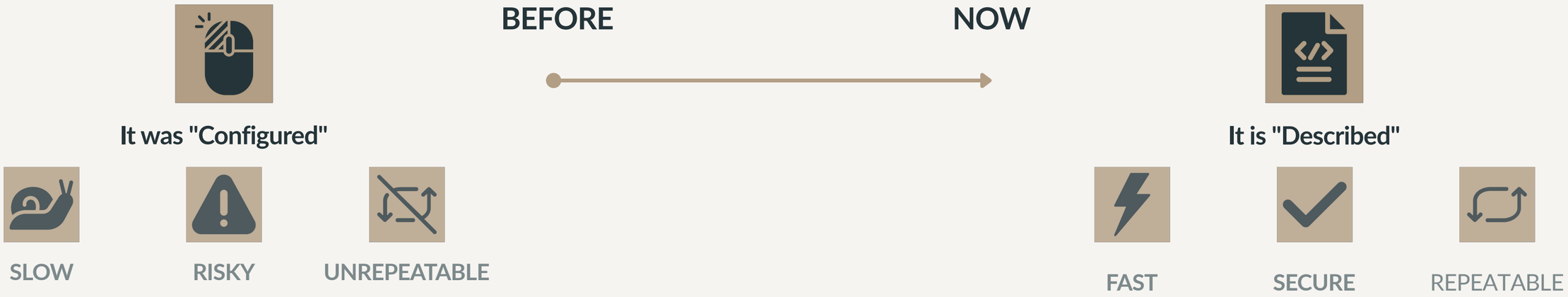
Prof. Fabio Palomba
University of Salerno

Prof. Damian A. Tamburri
Dott. Stefano Fossati
Jheronimus Academy of Data Science

Dario Mazza
Student ID: 0522501553

Introduction

The Evolution of IT Infrastructure

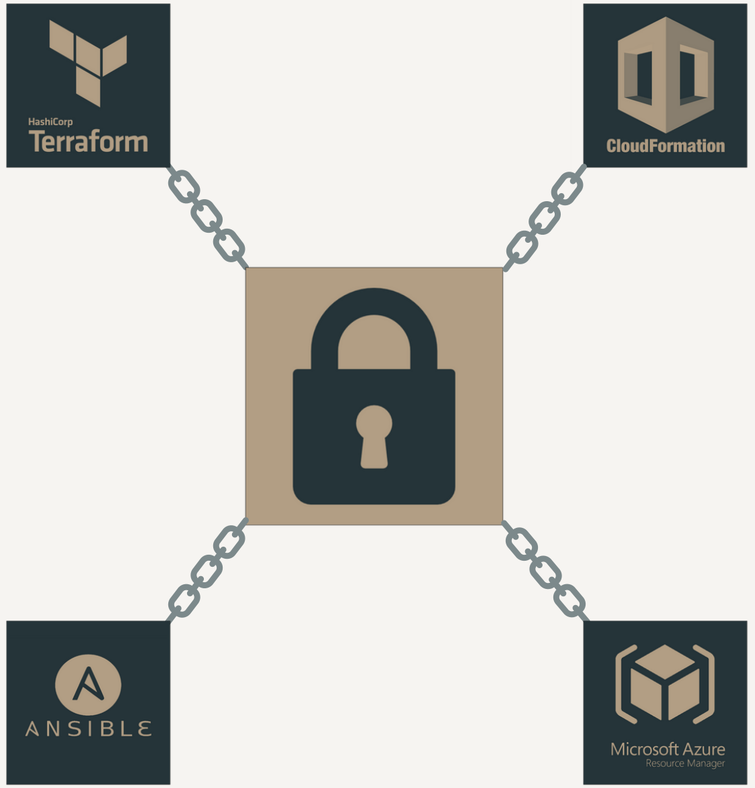


sesa lab
SOFTWARE ENGINEERING
SALERNO

JADS Jheronimus Academy of Data Science

The Problem

A Fragmented Ecosystem



Many tools, no common language.

The result? Vendor lock-in

The Way Out

The Standardization

Among the various attempts to create a “lingua franca” fot the cloud TOSCA

is the most promising OASIS standard for describing architecture in a

vendor-neutral way

```yaml
tosca_definitions_version: tosca_2_0

imports:
  - tosca_simple_profile:2.0

service_template:
  description: Esempio di un'applicazione web a due livelli.

  node_templates:

    # Definizione del Web Server
    web_server:
      type: Compute
      properties:
        # Proprietà che definiscono le risorse della macchina
        num_cpus: 2
        mem_size: 4 GB
        disk_size: 20 GB
      capabilities:
        # Specifica le caratteristiche del sistema operativo
        os:
          properties:
            architecture: x86_64
            type: linux
            distribution: ubuntu
            version: 20.04
      requirements:
        # Definizione della relazione: il web server ha bisogno di un database
        - database_endpoint:
            node: mysql_database # Si connette al nodo del database
            relationship: ConnectsTo

    # Definizione del Database
    mysql_database:
      type: DBMS
      properties:
        # Proprietà specifiche per un nodo di tipo Database
        name: MySQL
        version: 8.0
        port: 3306
```

What is TOSCA in Brief?

An Example

- **Nodes:** The infrastructure components (e.g., *web_server*, *mysql_database*)
- **Types:** The classification of a node (e.g., *Compute*, *DBMS*), imported from **Profiles**
- **Properties:** The specific characteristics of a node (e.g., *num_cpus*, *port*).
- **Relationships:** The connections between nodes, defined through **requirements** (e.g., the server *requires* a database).

The Adoption Paradox

Immature Ecosystem

The ecosystem of tools is still niche and not natively supported by the main industrial orchestrators.

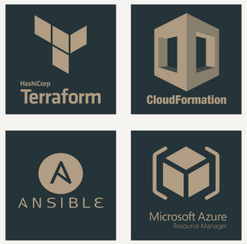
De Facto Competition

The market is dominated by consolidated solutions, which make it difficult to abandon technologies already in use.

Prohibitive Migration Costs

Companies have already invested heavily in their infrastructures. Rewriting everything from scratch in TOSCA is an economically impractical option.

The Proposed Solution



Reverse Engineering

The goal: to discover the architectural intent to enable the new paradigm of

Intent-Based Orchestration

Existing Approaches and Their Limits

Horizontal Approaches

Current solutions operate in a **"horizontal"** way: they reconstruct IaC from active resources or translate between proprietary formats, effectively **preserving vendor lock-in**.

AI for Generation and Repair

AI research focuses on **IaC code generation**, but it is still immature for **reverse engineering**.

The Gap: Vertical Abstraction

A clear gap emerges: a **"vertical"** approach is missing, one capable of raising the abstraction from proprietary code to a **standard** in order to discover the true architectural intent.

The Research Questions

RQ_1

What are the **design principles** for a **deterministic** mapping?

RQ_2

To what extent can **Large Language Models** (with RAG) support the **automatic** translation from IaC to TOSCA?

RQ_3

What are the **observed trade-offs** between the deterministic approach and the AI-based one in the context of the translation?

Methodology
Action Design Research

Problem Formulation

Define the problem and the objectives.

Building & Evaluation

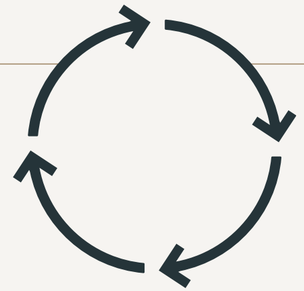
Build and evaluate the artifact.

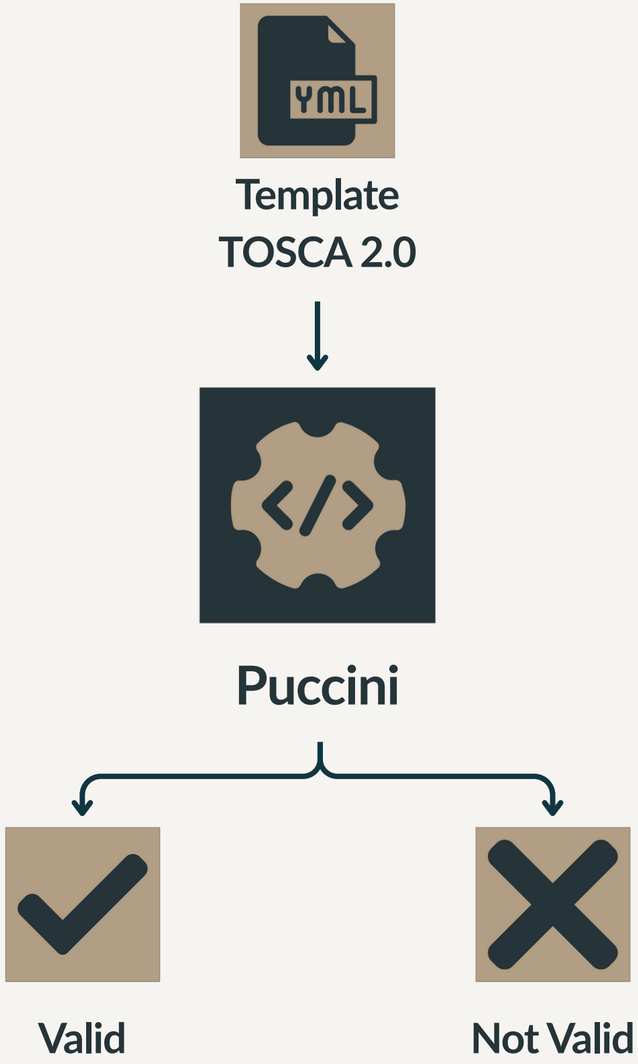
Reflection & Learning

Reflect on the results to learn.

Formalization of Learning

Extract generalizable knowledge.





A Fundamental Contribution

The "Puccini" Compiler

The "Tooling Gap":

A reference compiler was missing to rigorously validate the new TOSCA 2.0 standard.

Solution:

The open-source Puccini compiler **has been extended and updated** to fully support the TOSCA 2.0 standard.

The Research Cycles

Cycle 1: Alpha

The Monolithic prototype

The architecture is not scalable.

Cycle 2: Beta

The Modular Framework

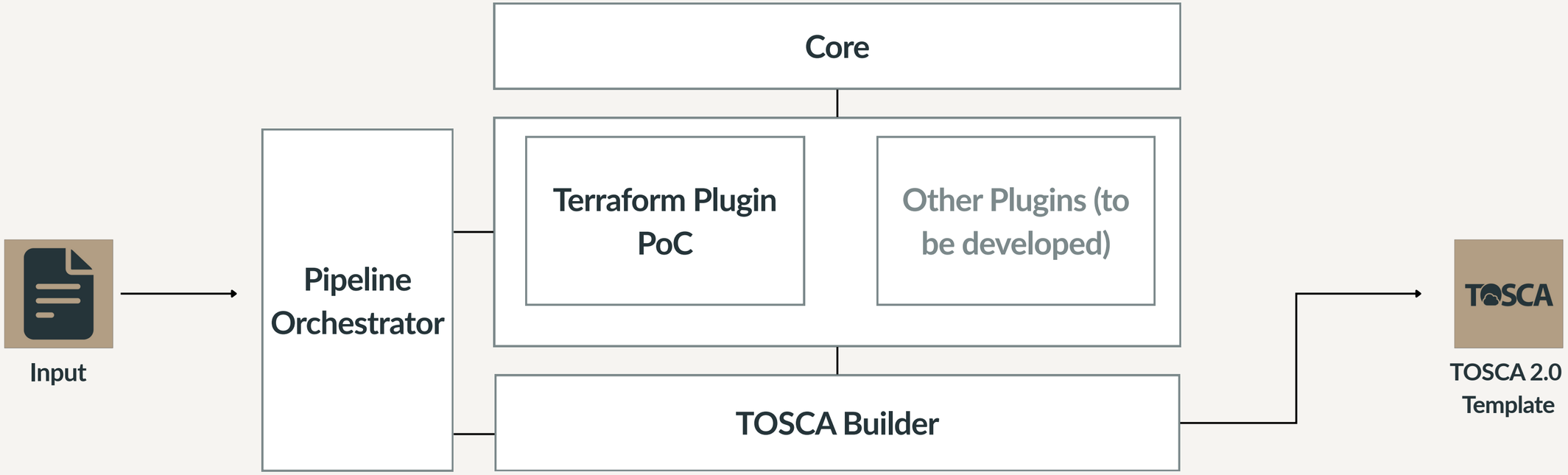
A robust, extensible, and reliable solution.

Cycle 3: Gamma

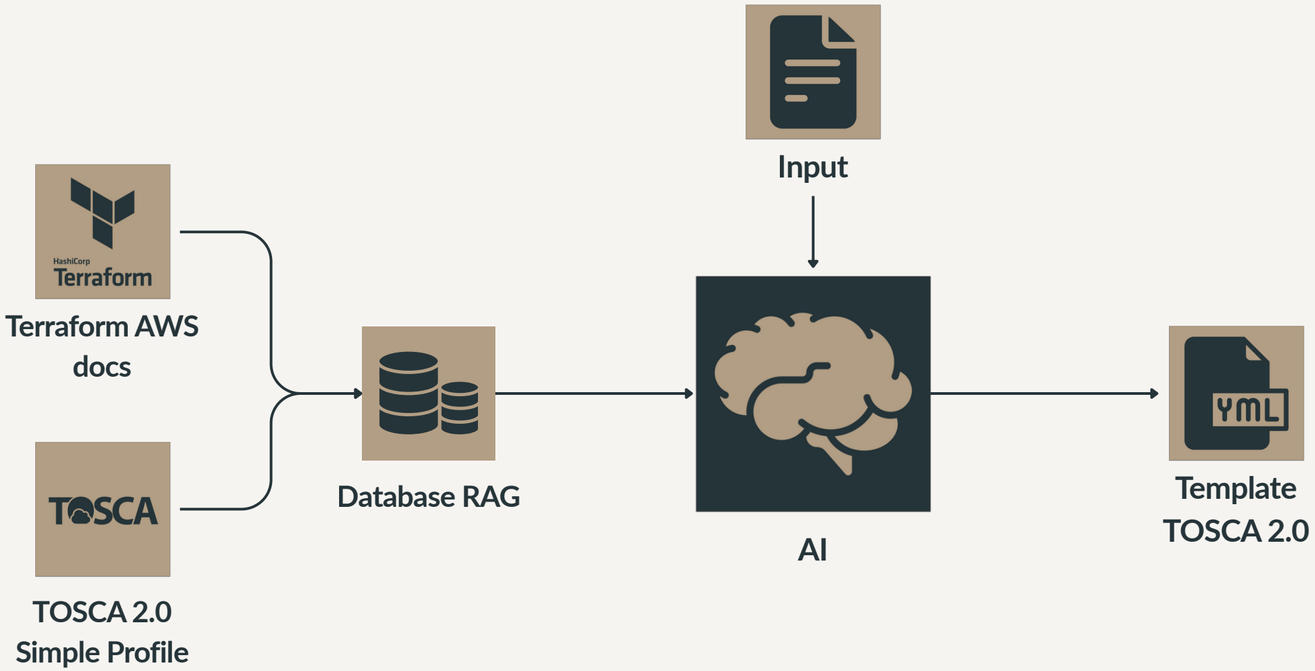
The AI experiment

Promising for simple tasks, but unreliable and incomplete.

The Deterministic Framework
The Beta Artifact



The AI Experiment
The Gamma Artifact



- ✔ **Promising:**
  Correctly maps single resources.
- ✘ **Restricted Scope (by Design):**
  The pipeline is focused only on single resources; it does not attempt to infer the topology (the relationships).
- ✘ **Not always reliable:**
  The same resource could be mapped in different ways.

Conclusions
Answers to the Research Questions

RQ_1 Principles for Deterministic Mapping

- Modular & semantic framework
- Preserves metadata & multi-cue inference
- Type-safe builder

RQ_2 AI Support (Exploration)

- Potential for single resources
- Critical limits: Topology inference & reliability (non-deterministic)

RQ_3 The Observed Trade-offs (In the Context of the PoC)

- Deterministic: Offers high precision and reliability, **but requires high manual effort.**
- AI: Has potential for low effort, but is incomplete (ignores topology) and unreliable.

Lessons Learned

01

The Limit is the Simple Profile, not the Translation

The standard type profile is **too generic** for modern cloud concepts, causing a loss of semantic detail in the translation.

02

AI: Limited Graph Awareness

The AI uses the graph of TOSCA types (Neo4j) for the nodes, but ignores both the **source graph** (Terraform) and the **target graph** under construction, preventing the inference of **relations**.

03

AI Unreliability

Inconsistent AI classifications make this approach unsuitable for production (non-deterministic).

Main Contributions

01

Methodological & Practical Framework (Beta)

Defined the **design principles** for "vertical" reverse engineering and developed a **modular and extensible PoC framework**.

02

Empirical Comparison of Approaches

Provided the first comparative analysis between the deterministic and the AI-augmented approach in this context, identifying key **trade-offs**.

03

Contribution to the Open Source Ecosystem

Extended the Puccini compiler to support TOSCA 2.0. The contribution has been **validated and integrated** into the official project for the benefit of the community.

Limits and Validity of the Research

Architectural, not Empirical, Extensibility

The multi-technology validation is only **architectural** (design review), not **empirical** (second plugin not implemented).

Qualitative Evaluation of Semantics

The "semantic correctness" was evaluated **qualitatively** (review with the ADR Team), not with quantitative metrics.

Exploratory AI Study

The AI evaluation was **exploratory**, aimed at identifying critical limits, not at measuring quantitative performance.

Future Developments

01

Empirical Extension and Validation

Implement a second plugin (e.g., Ansible) and support a new cloud (e.g., Azure) to **empirically validate** the framework's extensibility.

02

Evolution of the AI Approach

Develop a **"topology-aware"** AI and investigate a **"human-in-the-loop"** system to combine AI's efficiency with human reliability.

03

Maturing the TOSCA Ecosystem

Propose a more complete type profile to fill the semantic "gaps", **extend more tools to TOSCA 2.0** (against the adoption paradox), **and investigate "round-trip engineering".**

University of Salerno

sesa lab
SOFTWARE ENGINEERING SALERNO

JADS Jheronimus Academy of Data Science

Thank You!


dariomazza24@gmail.com


github.com/xDaryamo


linkedin.com/in/dario-mazza/