UNIVERSITY OF
LEADING
THE WAY
WESTMINSTER

Towards Cloud Application Description Templates Supporting Quality of Service

Pierantoni, Kiss, Terstyanszky

UNIVERSITY OF LEADING THE WAY WESTMINSTER

Content

- Introduction to COLA
- COLA Architecture
- COLA and TOSCA
- QoS Policies Overview and Structure

UNIVERSITY OF LEADING THE WAY WESTMINSTER

Problems in the migration to Cloud

- Intrinsic complexity required to describe:
  - the correlated services,
  - the QoS that describe its execution,
  - the procedures to deploy, undeploy and migrate applications in different IaaS platforms.

UNIVERSITY OF LEADING THE WAY WESTMINSTER

The COLA (Cloud Orchestration at the Level of Application) Project

- COLA aims at addressing the previously mentioned difficulties to foster the adoption of cloud computing services.

- The COLA project will provide a reference implementation of a generic and pluggable framework that supports the optimal and secure deployment and run-time orchestration of cloud applications.

UNIVERSITY OF LEADING THE WAY WESTMINSTER

The COLA (Cloud Orchestration at the Level of Application) Project

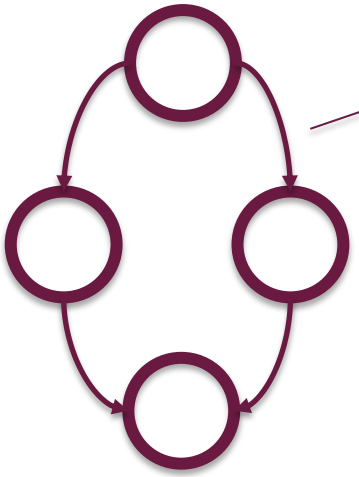
- Applications can then be embedded into workflows or science gateway frameworks to support complex application scenarios from user-friendly interfaces.

- A specific aspect of the cloud orchestration framework developed by COLA is the ability to describe complex application architectures incorporating several services.

- The framework will also support the definition of various Quality of Service (QoS) parameters related to performance, economic viability and security.

UNIVERSITY OF LEADING THE WAY WESTMINSTER

Cola Concept



Describe the overall topology.

What is the Application ?

Describe the Execution Modalities.

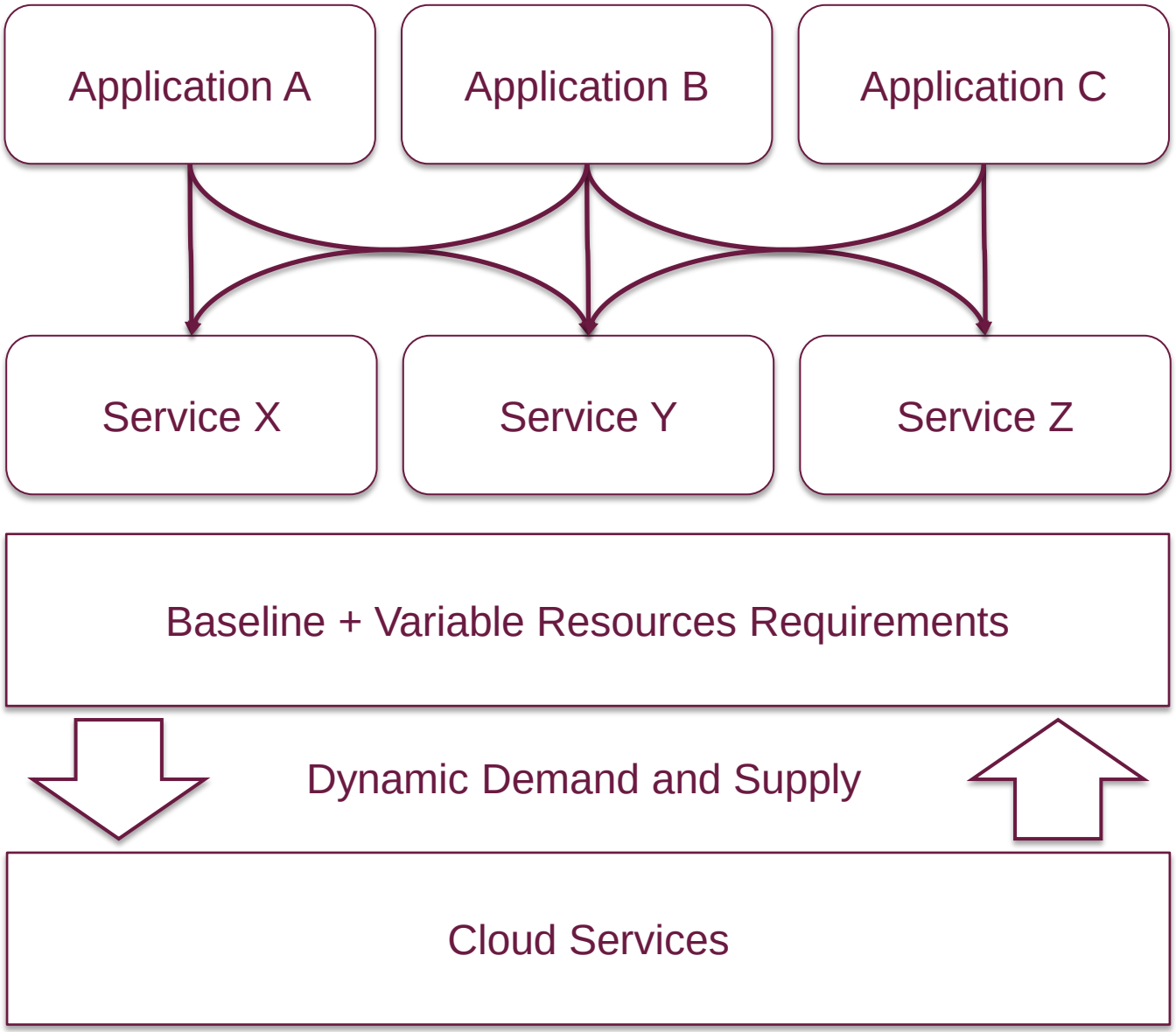
How do I execute the Application ?

COLA
Cloud Orchestration at the Level of Application

Execute the Application

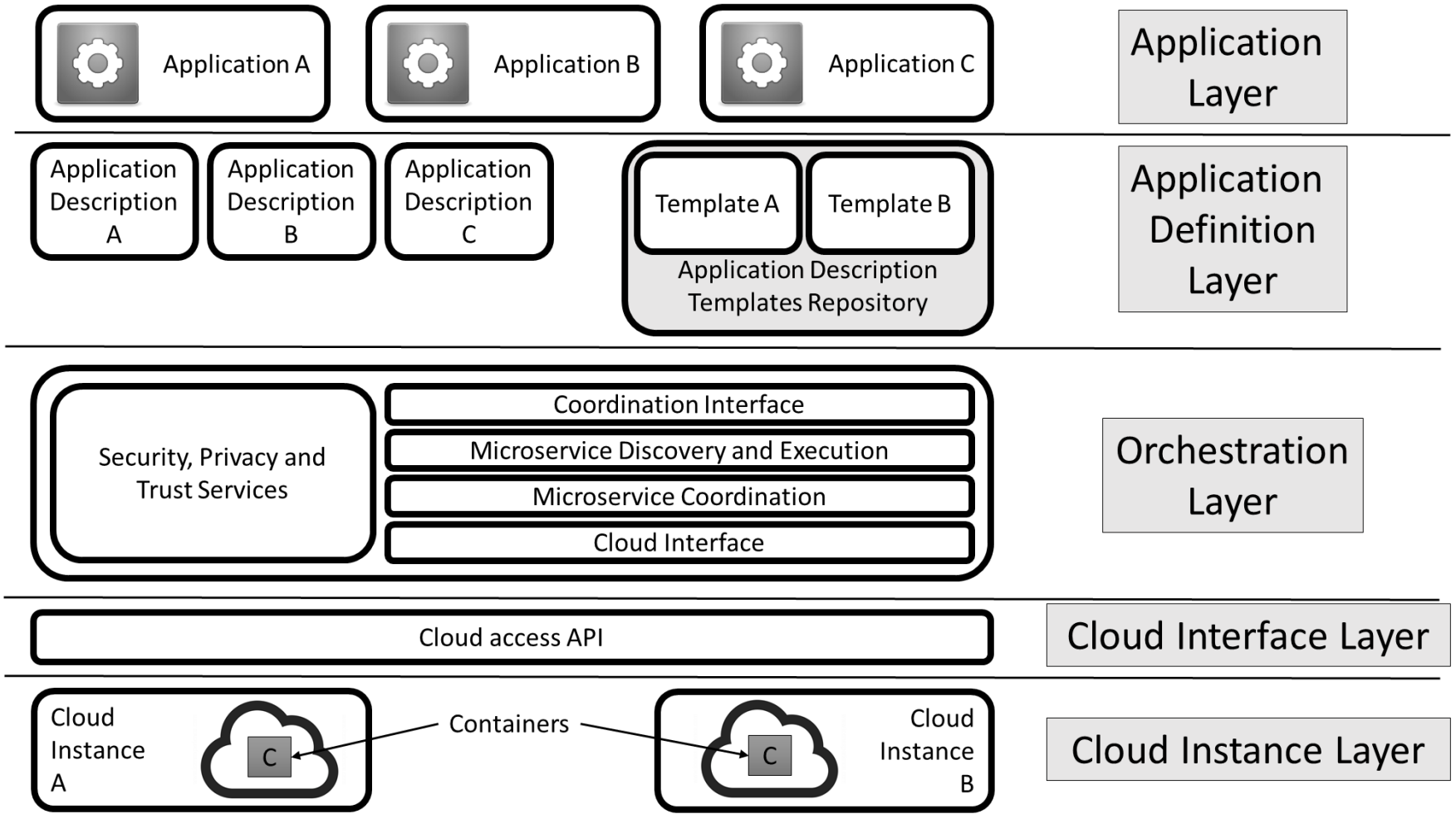
UNIVERSITY OF LEADING THE WAY WESTMINSTER

COLA Concepts



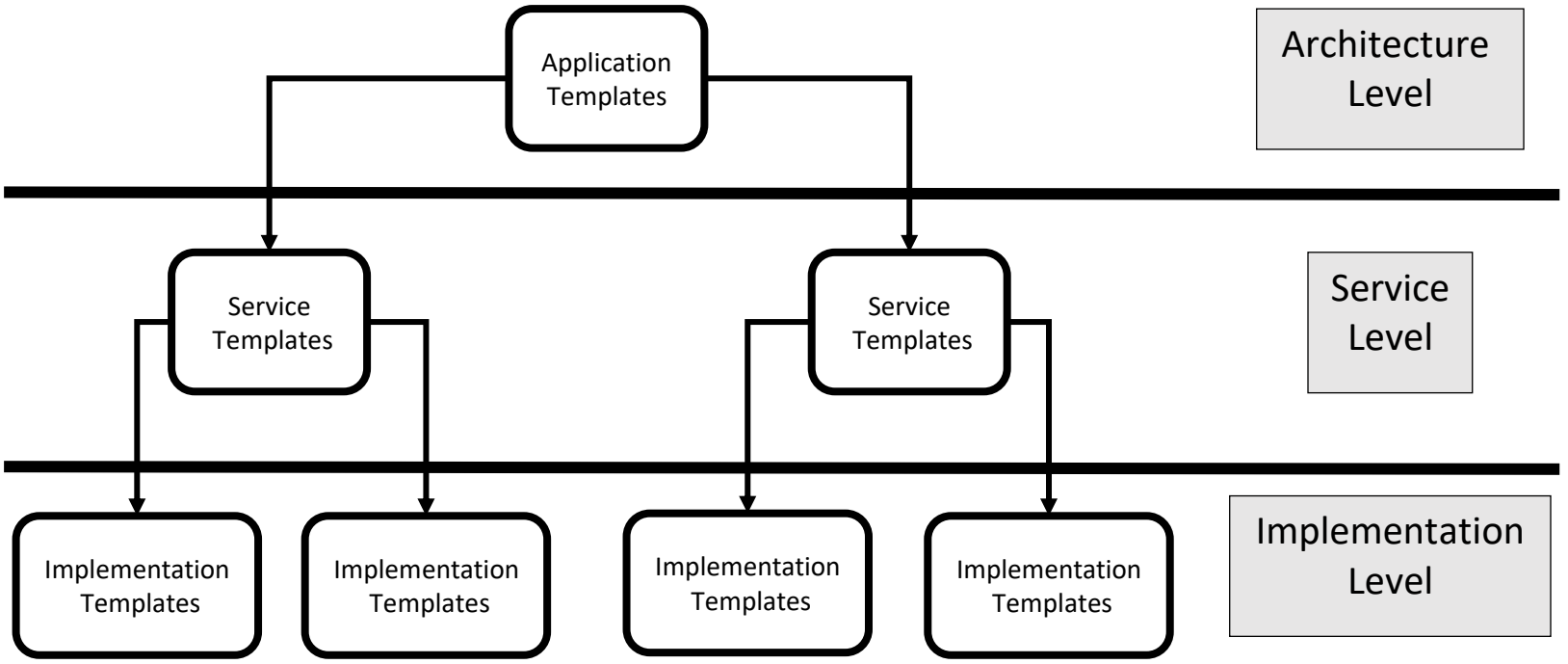
UNIVERSITY OF LEADING THE WAY WESTMINSTER

COLA Architecture



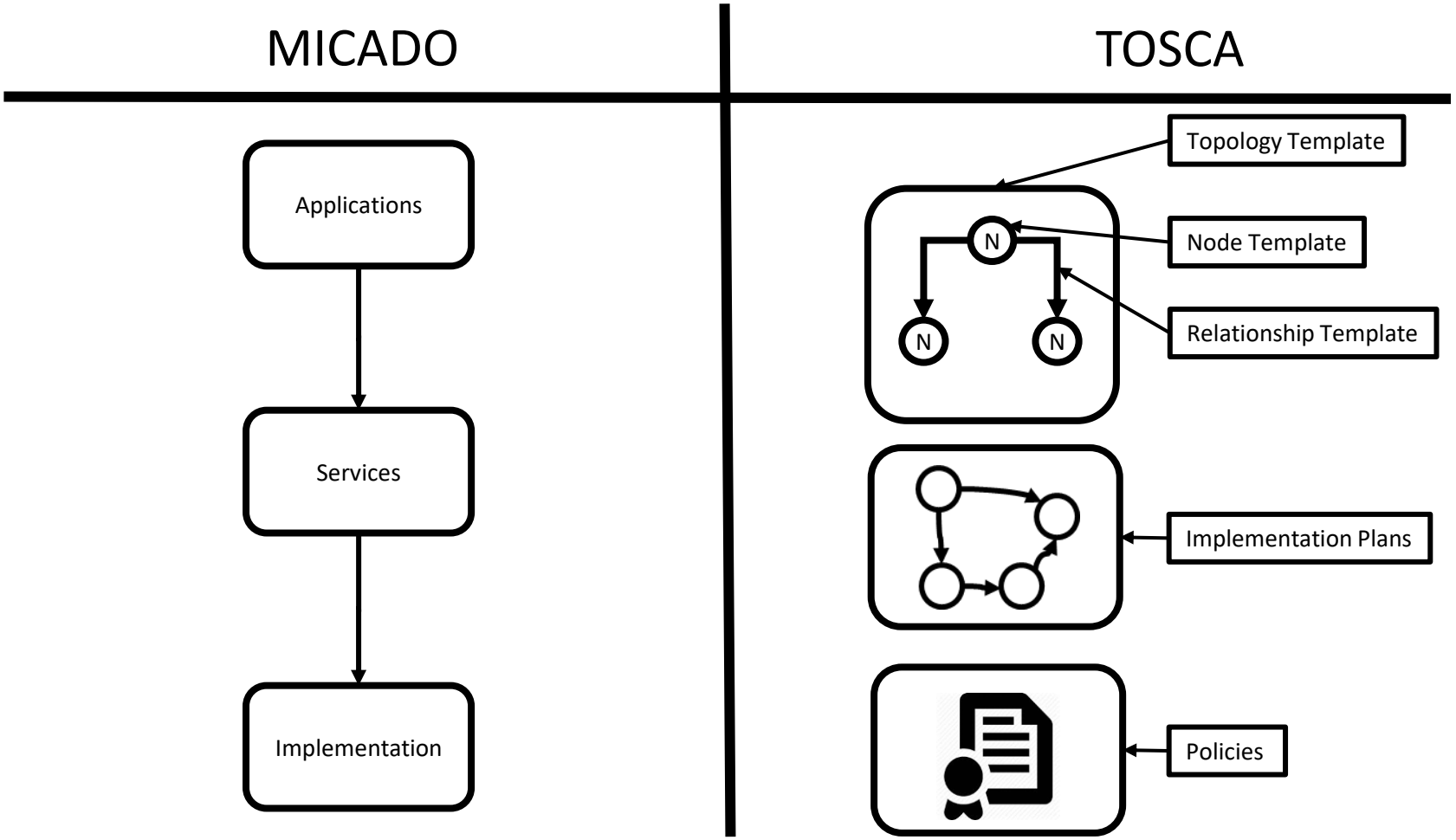
UNIVERSITY OF LEADING THE WAY WESTMINSTER

The Application Description Architecture



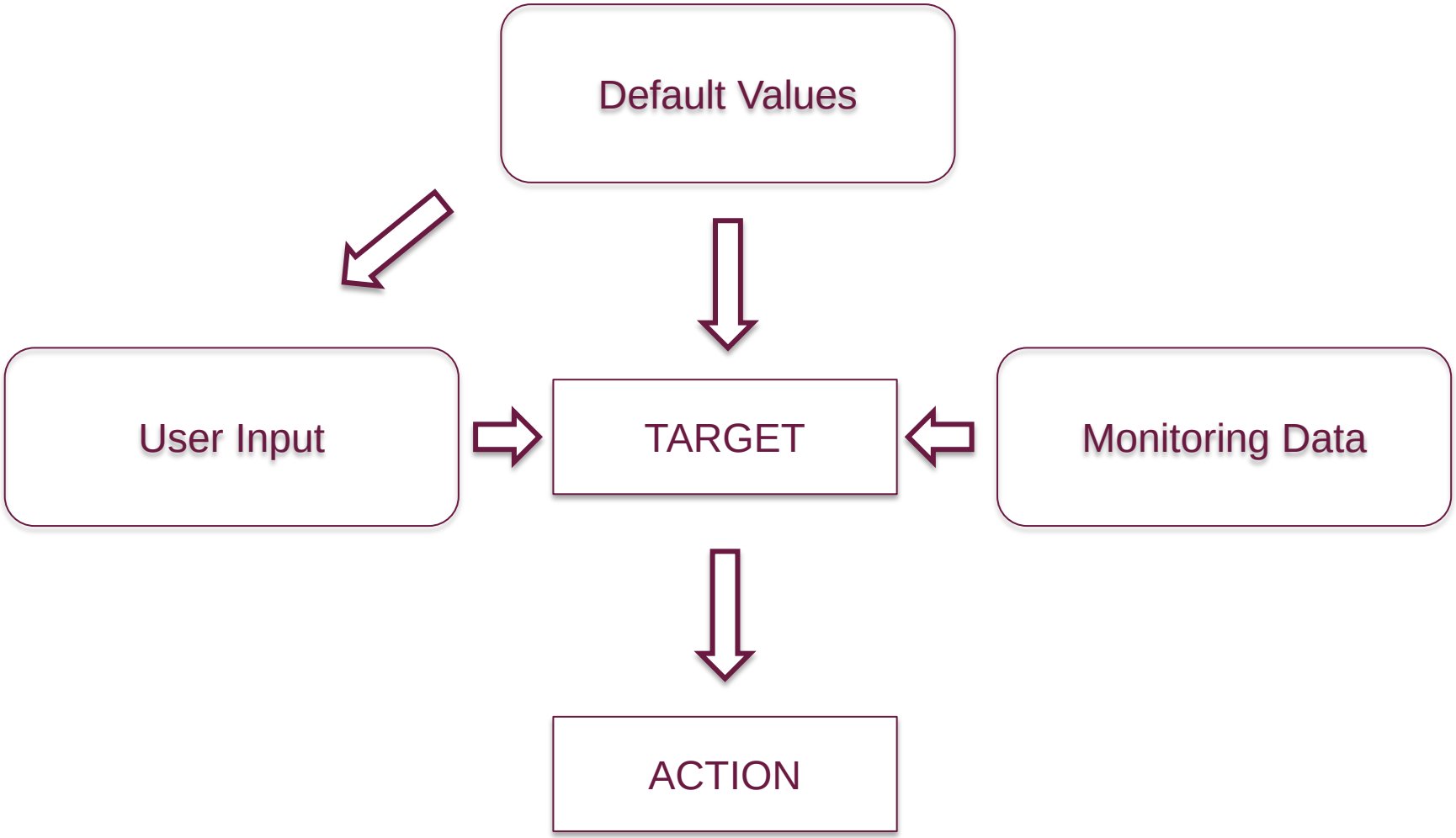
UNIVERSITY OF LEADING THE WAY WESTMINSTER

COLA, MICADO and TOSCA



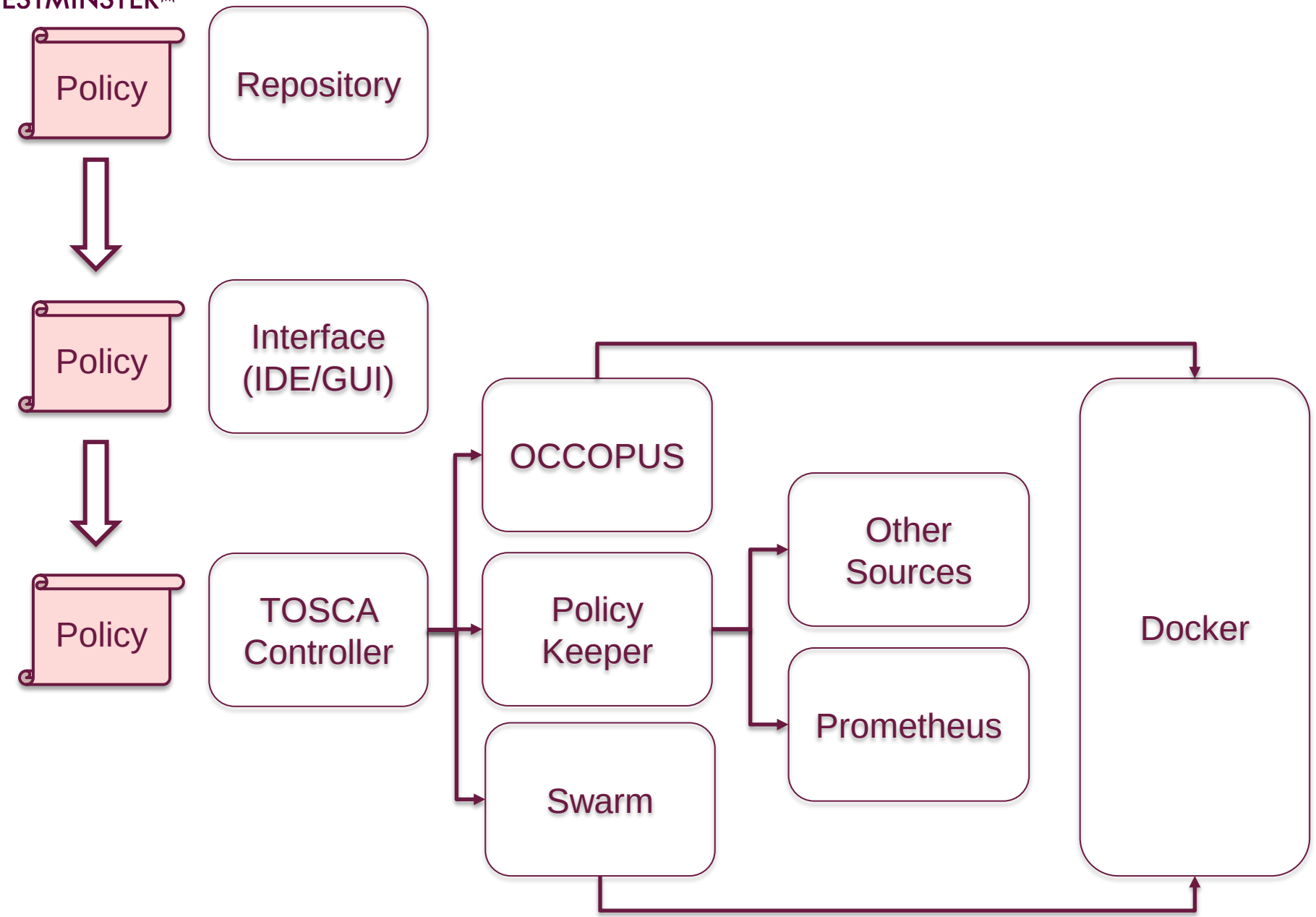
UNIVERSITY OF LEADING THE WAY WESTMINSTER

Application Description and Policies



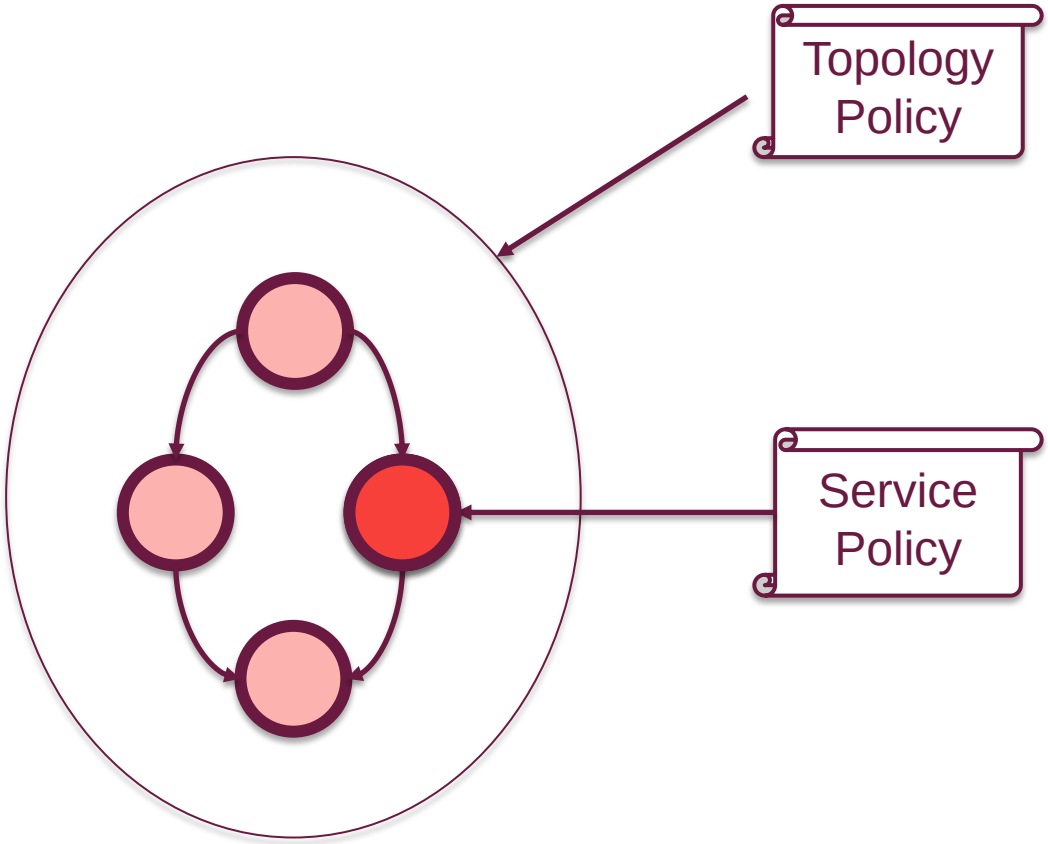
UNIVERSITY OF LEADING THE WAY WESTMINSTER

Application Description and Policies



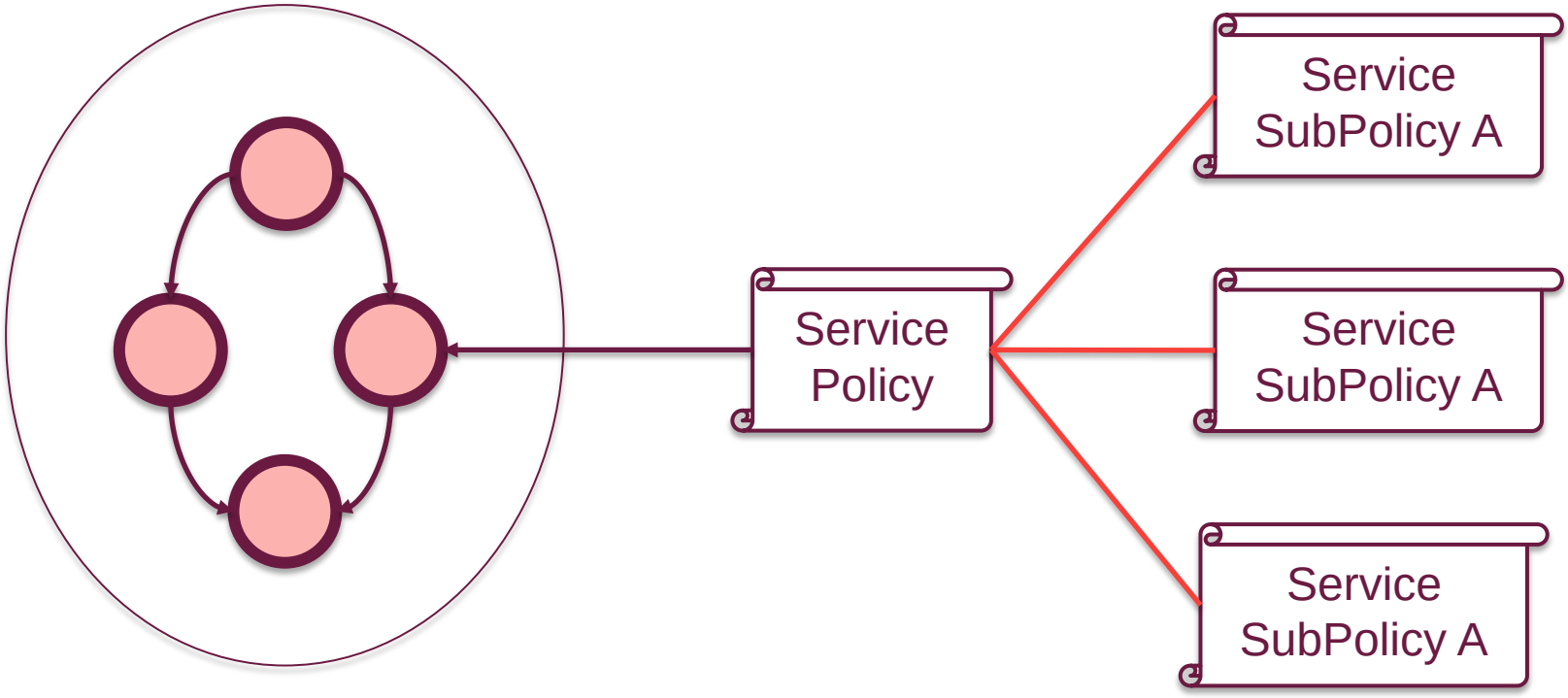
UNIVERSITY OF LEADING THE WAY WESTMINSTER

Application Description and Policies



UNIVERSITY OF LEADING THE WAY WESTMINSTER

Application Description and Policies



UNIVERSITY OF LEADING THE WAY WESTMINSTER

Policies Parameters

- **Stage**: Deployment, Runtime, Undeployment
- **Topological Layer**: Global (all nodes), Local (single node)
- **Technological Layer** (Target): Service, Operative System, Virtual Machine, etc…

UNIVERSITY OF LEADING THE WAY WESTMINSTER

Design Constraints/Guidelines

– **Declarative Model**. The policies description follows the Declarative Model. E.g. they describe a policy but they do not imperatively describe how to implement the policy, which is left to the proper components of the COLA architecture

UNIVERSITY OF LEADING THE WAY WESTMINSTER

Design Constraints/Guidelines

- **Pick and Choose Policies**. Users can select policies from a basket of existing policies but cannot create new policies

UNIVERSITY OF LEADING THE WAY WESTMINSTER

Design Constraints

– **Parameter-Based Policies**. Users can define parameters of the selected policies but cannot change the structure of the selected policies.

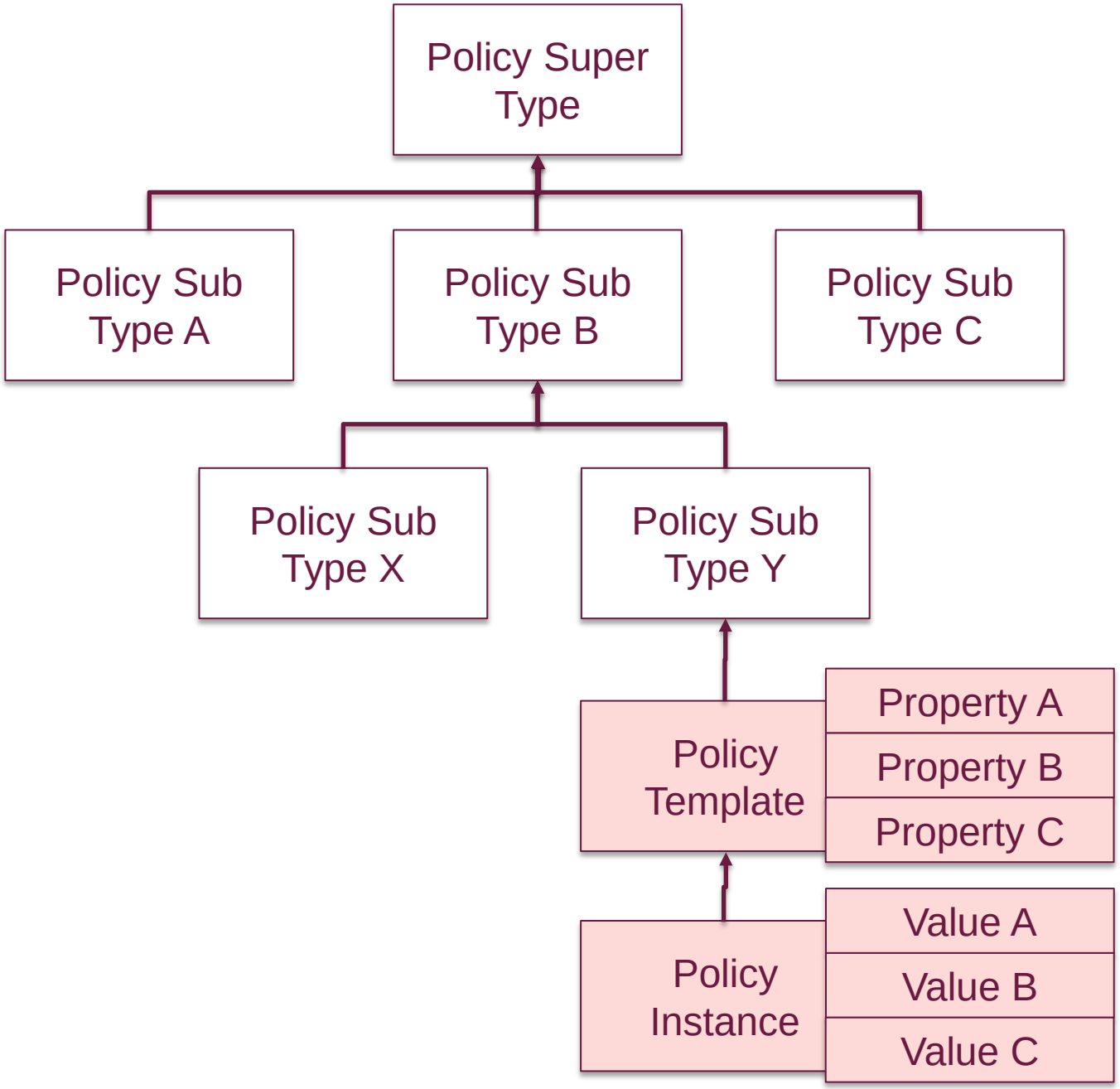
UNIVERSITY OF LEADING THE WAY WESTMINSTER

Design Constraints

- **No Consistency is ensured**. Policies may be contradictive.
- At this stage of the policies description, only a priority level will be offered to the user to define which policy has the highest priority but this priority may not always be respected.

UNIVERSITY OF LEADING THE WAY WESTMINSTER

Generic Policy Structure



UNIVERSITY OF LEADING THE WAY WESTMINSTER

Generic Policy Structure

- **Name:** A String that represents the name of the Policy
- **Type:** A String that defines the type of the policy, as an example: Authentication, Location, etc…
- **Description:** A textual description of the policy
- **Stage:** Define at which stage the policy applies. As an example, a policy may apply at deployment stage, or execution stage.
- **Priority:** This is an arbitrary integer of 0 to 100 used to define the priority with which the policy will be implemented.
- **Targets:** Targets define the technological element to which the policy has to be applied.
- **Parameters:** Description: Each policy is described by a list of parameters

UNIVERSITY OF LEADING THE WAY WESTMINSTER

Generic Policy Structure - Parameters

***Accepted values* + Triggers**:

Name	Description	Default Value	User Input	Constraints	Required	Trigger

UNIVERSITY OF LEADING THE WAY WESTMINSTER

Conclusions and Questions

- First Investigation of the Extension of TOSCA to describe policies will be completed in the summer

- Micado Architecture is in its third iteration and already supports simple policies although not expressed in TOSCA.

- Once the TOSCA extension will be tested, further research could investigate:
  - Solution of policies conflicts beyond simple priorites
  - Extending the trigger data sources namespaces beyond the available services