

# Sorting Hat: A Tool to Characterize the Architecture of Service-Based Systems

Author: Erick Rodrigues

Advisors: Alfredo Goldman João Francisco Daniel André van der Hoek

## Introduction

Developing large software systems is not trivial, as these systems must satisfy quality attributes such as scalability and maintainability. Therefore, it is important that they have an architecture that favors the fulfillment of these attributes. Understanding the architecture of a software, knowing its structural aspects and patterns, is essential to be able to satisfy the desirable quality requirements. However, there is little support for characterizing and evaluating systems architecture, especially those with service-based architecture.

The Sorting Hat is a tool under development that assists in the process of characterizing the architecture of service-based systems. It uses the CharM as supporting model. The first MVP of the visualization was built in 2021. However, the visualization had some issues related to its usability and user experience, with many context switching due to the large number of viewpoints and deep navigation flow. Also, the metrics the tool had implemented were not up-to-date with the current state of the CharM. Lastly, the first version does not have any support for automated data collection. The main goals of this work for the discipline MAC0499 (Final Capstone) are:

- (1) to improve the visualization of the tool and update its information to the most recent CharM metrics,
- (2) to provide an automated data collection to the Sorting Hat.

## International Experience

From April 2022 to July 2022, I had been in an international internship, with a FAPESP scholarship, at the Department of Informatics of the University of California, Irvine (UCI) advised by the Professor Dr. André van der Hoek. Some of the achievements and challenges of that period were:

- I have been outside my country for the first time
- I have had contact with another culture
- I have spoken a foreign language for 4 months
- I have had the opportunity to work with an expert in Software Engineering and his research group
- We have built the Sorting Hat's new visualization

## Results

In the first MVP, you had to access many pages in order to view all information

CharM metrics were not up-to-date, which compromised the tool's content

There was no support for automated data collection



Now, all information take place in a single page, with a reformulated, more elegant design

CharM metrics are up-to-date and automatically extracted for every new system in the tool

There are many filter options in order to make the visualization more interactive

### InterScity

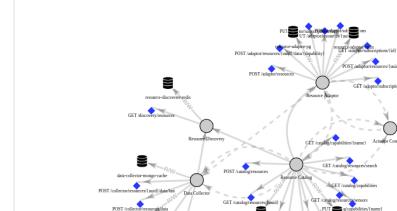
Dimensions:  Size  Data coupling  Sync coupling  Async coupling

Link synchronous communications through operations

Group services by deployment unit (Modules)

Click on a service and type a depth level you want to see.

Depth: 1



### Metrics

Global:

Number of system components

5 services and 5 modules

Number of services with deployment dependency

0

Largest service (12 operations)

Resource Catalog

Smallest service (1 operation)

Resource Discovery

Number of components that consume messages from the queue

• modules: 4

• services: 4

Number of components that publish messages in the queue



Now, the tool analyzes docker-compose and openapi files to collect the services, endpoints, databases and databases usages of a new service-based system registered in the tool

### Register new system

Repository URL:

Docker-Compose filename:

Register

Only a remote repository link and a docker-compose filename are needed to register a new system



## CharM - Characterization Model

To evaluate the current architecture of a microservices system, there is a characterization model called CharM, under development by the PhD Candidate Thatiane de Oliveira Rosa. That model, which is in constant evolution, currently has its core around four dimensions: size, data source coupling, synchronous coupling and asynchronous coupling, each one being a composition of a set of metrics.

