Systems Selection Analysis

Step 1: Basic search: We designed our search string to be a conjunction of two corresponding populations: Microservice population AND Frameworks technology population. Concerning the Microservice population, we have considered that Microservice itself should be a recurrent keyword, just to make sure, we have widened the search by including a more open keyword with the prefix service: Microservice population = (microservice* OR “microservice”* OR “micro service”* OR “microservice architecture”*). As for the Frameworks technology population, the situation may be more diverse. As a result, the string is: Frameworks technology population = (spring* AND java* AND (docker-compose OR docker) * OR netflix* OR asynchronous* OR reactive*). We applied the search string over the Github repository. The outcome result set of this step turned out to have a length of 121 items.

In GitHub: This step starts by applying individual search queries filtered by ‘Repositories’ and sorted by ‘Best match’ (using GitHub’s UI) each resulted in a maximum of 100 pages. Those result sets are then additively joined while increasing the score of repeated results according to their accumulative occurrences in the search queries’ hits. After that, the result list is sorted by hit score in descending order and its length is reduced out to 150 items. Finally, all results with score that is less than 3 are excluded from the result set.

Step 2: The title, description architectural diagram and documentation of the projects were reviewed by a member of a team. Projects were discarded for not having clear description (or written in other language than English, e.g., Chinese, Spanish … etc.), any documentation, paper or tutorial pages.

In the Excel sheet: apply the filter for (documentation type column) by unchecking N/A, N/A[Chinese] and N/A[Spanish] which refers to systems that do not include documentation. The outcome result set for this step turned out to have a length of 78 systems.

Step 3. System functionality considered: Two researchers were appointed to every remaining project to examine the functionality of the systems. We included projects which implement at least two business functionalities (e.g., “Stock Price Viewer” system was excluded since it implements only one business service named “stock-service”) and excluded projects that only implement infrastructure services, (e.g., “Microservice Monitoring” system was excluded since it only implements monitoring infrastructure services). The outcome result set of this step turned out to have a length of 55 items.

In the Excel sheet: apply the filter for (Business microservices column) by unchecking N/A which refers to systems that do not include Business microservices and uncheck [1] which refers to systems that include only one Business/functional microservice. The outcome result set for this step turned out to have a length of 55 systems.

Step 4. Source artefact analysis: All three researchers were involved in an analysis which checks the files of the artefacts in full and searches through source files to classify the systems as included or excluded based on existence of essential framework artefacts as pointed in the inclusion/exclusion criteria in Table 1.