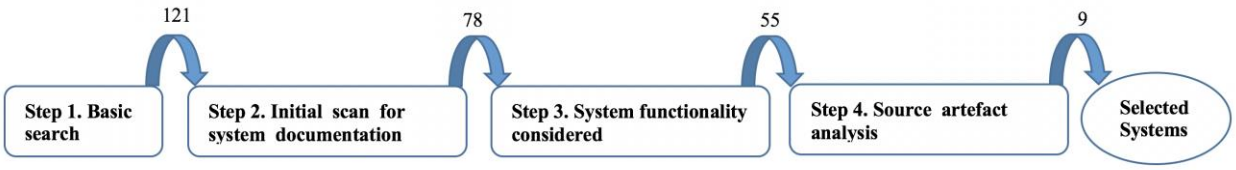
Systems Selection Analysis



Step 1: 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. The table below illustrates instances of search queries that contributed in the final selected systems. Those result sets are then additively joined while increasing the score of repeated results according to their accumulative occurrences in the search

Population	Input	Language	Selected systems appeared in the results
Microservice	"microservice"[1]	-	[1], [2], [3], [4], [5], [6], [7], [8], [9]
Microservice	"microservice" "architecture"[2]	-	[1], [2], [6]
Microservice	"microservice" "architecture"[3]	Java	[2], [6]
Microservice AND Framework	"microservice" "spring"[4]	Java	[1], [2], [3], [4], [5], [6], [7], [8], [9]
Microservice AND Framework	"microservice" "spring" "boot"[5]	Java	[2], [3], [4], [5], [6], [8], [9]
Microservice AND Framework	"microservice" "spring" "cloud"[6]	Java	[3], [4], [5], [6], [7], [9]
Microservice AND Framework	"microservice" "spring" "docker"[7]	Java	[3], [4], [9]

[1] https://github.com/search?o=desc&q=%22microservice%22&s=&type=Repositories

[2] https://github.com/search?o=desc&q=%22microservice%22+%22architecture%22&s=&type=Repositories

[3] https://github.com/search?l=Java&o=desc&q=%22microservice%22+%22architecture%22&s=&type=Repositories

[4] https://github.com/search?l=Java&o=desc&q=%22microservice%22+%22spring%22&s=&type=Repositories

[5] https://github.com/search?l=Java&o=desc&q=%22microservice%22+%22spring%22+%22boot%22&s=&type=Repositories

[6] https://github.com/search?l=Java&o=desc&q=%22microservice%22+%22spring%22+%22cloud%22&s=&type=Repositories

[7] https://github.com/search?o=desc&q=%22microservice%22+%22spring%22+%22docker%22&s=&type=Repositories

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. The outcome result set of this step turned out to have a length of 121 systems.

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: Two researchers were appointed to every remaining project to examine the functionality of the systems. We included projects which implement at least two business functionality and excluded projects that either implement infrastructure services only, (e.g., "Microservice Monitoring" system was excluded since it only implements monitoring infrastructure services[8]), or only one business microservice (e.g., "Stock Price Viewer" system was excluded since it implements only one business service named "stock-service[9]").
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: Code analysis, subjective by checking the code and analyze the source artifacts. The outcome result set for this step turned out to have a length of 9

[8] https://github.com/xeraa/microservice-monitoring
[9] https://github.com/TechPrimers/stock-price-viewer-microservices-part1