Monolith Splitter

SFSCON 2024

Support for Migration of a Monolithic System to Microservices

SFSCON

MUNI Masaryk University

Speakers

Bruno Rossi

- Assistant Professor at the at the Faculty of Informatics, Masaryk University, Brno, Czech Republic
- Research interests in software evolution, open source software systems, and cyber-physical systems
- Integrating open source software in the courses taught at Masaryk University

Michal Skipala

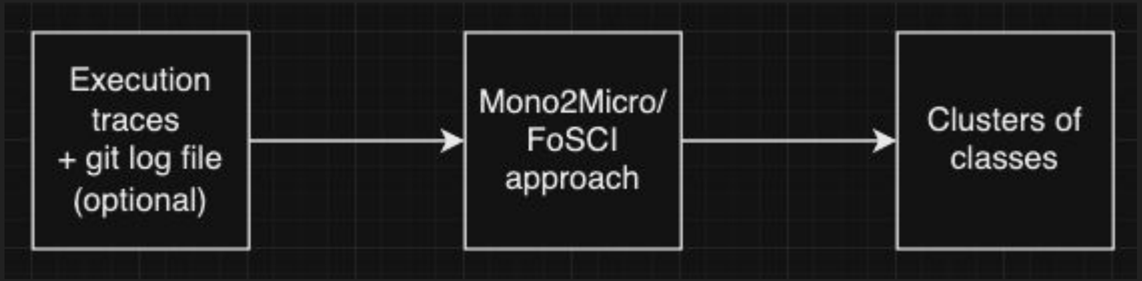
- Software Engineer at T-Mobile Czech Republic
- Student at Masaryk University, currently working on Diploma Thesis
- Specializes in Fullstack Web Development in recent technologies
- Proficient in Java and ReactJS

Context

- Migration from monolithic systems is of interest for many companies running on legacy systems, but…
  - Complexity of migration to microservices architecture
  - Numerous ways to split a monolith into a microservices system
  - Measuring the quality of the final migration is challenging
- Research papers suggesting ways to optimize the process often lack implementation

What is Monolith Splitter?

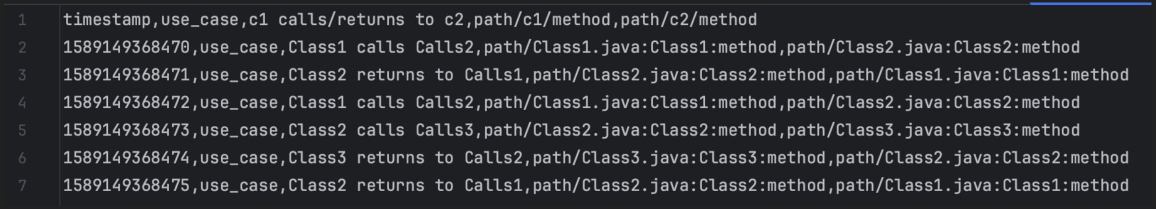
- A tool in development that suggests approaches for migrating monolithic applications to microservices
- Accepts execution traces and Git log files as input
- Provides recommendations for clustering classes from execution traces into microservice-aligned groups
- Tech Stack: Java 21 + Spring Boot 3.2, PostgreSQL, TypeScript, ReactJS


Execution traces + git log file (optional)
Mono2Micro/ FoSCI approach
Clusters of classes

Benefits

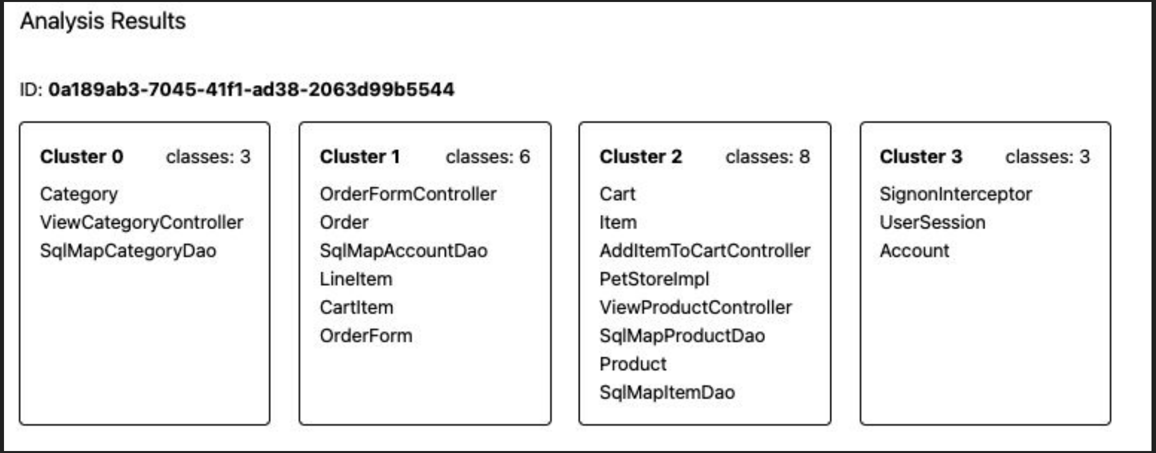
- Monolith Splitter provides support for software architects aiming to transition applications to a microservices architecture
- Algorithm implementations will be openly available, enabling practitioners to leverage the tool across various projects
- Accelerates the modernization of legacy applications by providing optimized strategies for monolith decomposition

Example Input & Output



```
timestamp,use_case,c1 calls/returns to c2,path/c1/method,path/c2/method
1589149368470,use_case,Class1 calls Calls2,path/Class1.java:Class1:method,path/Class2.java:Class2:method
1589149368471,use_case,Class2 returns to Calls1,path/Class2.java:Class2:method,path/Class1.java:Class1:method
1589149368472,use_case,Class1 calls Calls2,path/Class1.java:Class1:method,path/Class2.java:Class2:method
1589149368473,use_case,Class2 calls Calls3,path/Class2.java:Class2:method,path/Class3.java:Class3:method
1589149368474,use_case,Class3 returns to Calls2,path/Class3.java:Class3:method,path/Class2.java:Class2:method
1589149368475,use_case,Class2 returns to Calls1,path/Class2.java:Class2:method,path/Class1.java:Class1:method
```

I: Execution traces



Analysis Results

ID: **0a189ab3-7045-41f1-ad38-2063d99b5544**

Cluster 0 classes: 3

Category
ViewCategoryController
SqlMapCategoryDao

Cluster 1 classes: 6

OrderFormController
Order
SqlMapAccountDao
LineItem
CartItem
OrderForm

Cluster 2 classes: 8

Cart
Item
AddItemToCartController
PetStoreImpl
ViewProductController
SqlMapProductDao
Product
SqlMapItemDao

Cluster 3 classes: 3

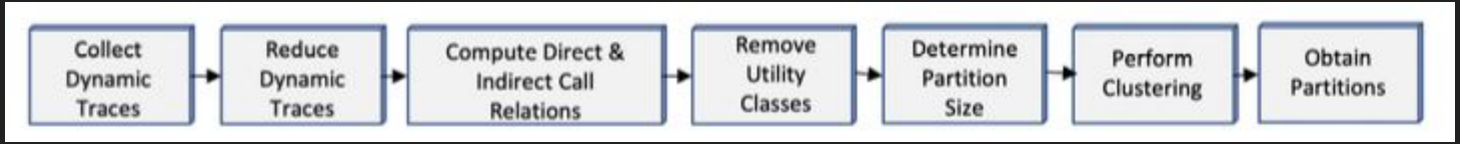
SignonInterceptor
UserSession
Account

O: Clusters

Mono2Micro

- Spatio-temporal decomposition
- Business cases
- Runtime call relations

[1]



Collect Dynamic Traces → Reduce Dynamic Traces → Compute Direct & Indirect Call Relations → Remove Utility Classes → Determine Partition Size → Perform Clustering → Obtain Partitions

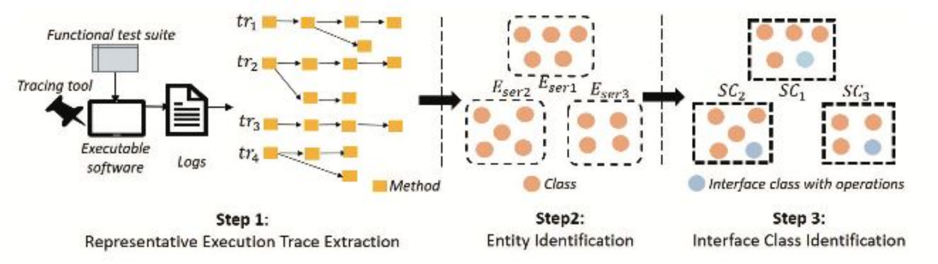
- Direct Call Relations, Direct Call Pattern
- Indirect Call Relations, Indirect Call Pattern
- Computation of Similarity
- Hierarchical Clustering

[1] Mono2Micro: A Practical and Effective Tool for Decomposing Monolithic Java Applications to Microservices
https://www.researchgate.net/publication/354057927_Mono2Micro_a_practical_and_effective_tool_for_decomposing_monolithic_Java_applications_to_microservices

FoSCI

- Functional Atom Generation (coherent, minimal functional unit)
- Assignment of Functional Atoms into Service Candidates
- Non-dominated Sorting Genetic Algorithm-II (NSGA-II) for optimization

[2]



- Structural and Conceptual Intra-Connectivity & Inter-Connectivity

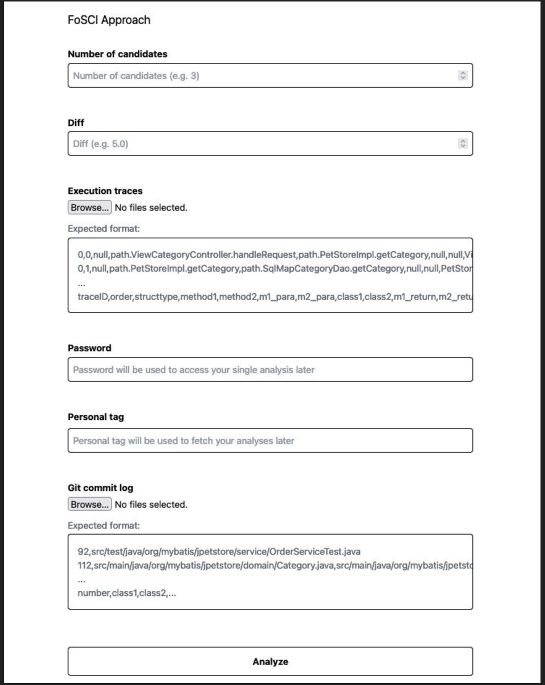
[2] Service Candidate Identification from Monolithic Systems Based on Execution Traces https://ieeexplore.ieee.org/document/8686152

Monolith Splitter Implementation & Integration

- Two reimplemented approaches
- Implementation based on research articles:

  [1] Mono2Micro: A Practical and Effective Tool for Decomposing Monolithic Java Applications to Microservices

  [2] Service Candidate Identification from Monolithic Systems based on Execution Traces



FoSCI Approach

Number of candidates

Number of candidates (e.g. 3)

Diff

Diff (e.g. 5.0)

Execution traces

Browse... No files selected.

Expected format:

```
0,0,null,path.ViewCategoryController.handleRequest,path.PetStoreImpl.getCategory,null,null,Vi
0,1,null,path.PetStoreImpl.getCategory,path.SqlMapCategoryDao.getCategory,null,null,PetStor
...
traceID,order,structtype,method1,method2,m1_para,m2_para,class1,class2,m1_return,m2_retu
```

Password

Password will be used to access your single analysis later

Personal tag

Personal tag will be used to fetch your analyses later

Git commit log

Browse... No files selected.

Expected format:

```
92,src/test/java/org/mybatis/jpetstore/service/OrderServiceTest.java
112,src/main/java/org/mybatis/jpetstore/domain/Category.java,src/main/java/org/mybatis/jpetstc
...
number,class1,class2,...
```

Analyze

[1] Mono2Micro: A Practical and Effective Tool for Decomposing Monolithic Java Applications to Microservices https://www.researchgate.net/publication/354057927_Mono2Micro_a_practical_and_effective_tool_for_decomposing_monolithic_Java_applications_to_microservices

[2] Service Candidate Identification from Monolithic Systems Based on Execution Traces https://ieeexplore.ieee.org/document/8686152

Monolith Splitter Test Results

- Currently tested on the JPetStore [3] and AcmeAir [4] applications
- Execution traces sourced from publicly available Mono2Micro datasets on GitHub [5]
- Future testing planned on widely-used, monolithic open-source projects written in Java

[3] JPetStore https://github.com/KimJongSung/jPetStore
[4] AcmeAir https://github.com/acmeair/acmeair
[5] Mono2Micro-FSE-2021 https://github.com/kaliaanup/Mono2Micro-FSE-2021

Mono2Micro JPetStore

5 Clusters

Cluster 0 classes: 5

Cart
AddItemToCartController
RemoveItemFromCartController
ViewCartController
CartItem

Cluster 1 classes: 13

UpdateCartQuantitiesController
SearchProductsController
PetStoreImpl
SqlMapProductDao::ProductSearch
Product
SqlMapProductDao
ViewItemController
Item
ViewProductController
SqlMapItemDao
ViewCategoryController
SqlMapCategoryDao
Category

Cluster 2 classes: 7

AccountFormController
AccountForm
AccountValidator
SqlMapAccountDao
Account
SignonController
UserSession

Cluster 3 classes: 8

Order
OrderValidator
OrderFormController
OrderForm
SqlMapSequenceDao
Sequence
SqlMapOrderDao
LineItem

Cluster 4 classes: 1

SignonInterceptor

6 Clusters

Cluster 0 classes: 5

Cart
AddItemToCartController
RemoveItemFromCartController
ViewCartController
CartItem

Cluster 2 classes: 7

AccountFormController
AccountForm
AccountValidator
SqlMapAccountDao
Account
SignonController
UserSession

Cluster 4 classes: 3

ViewCategoryController
SqlMapCategoryDao
Category

Cluster 5 classes: 1

SignonInterceptor

Cluster 1 classes: 10

UpdateCartQuantitiesController
SearchProductsController
PetStoreImpl
SqlMapProductDao::ProductSearch
Product
SqlMapProductDao
ViewItemController
Item
ViewProductController
SqlMapItemDao

Cluster 3 classes: 8

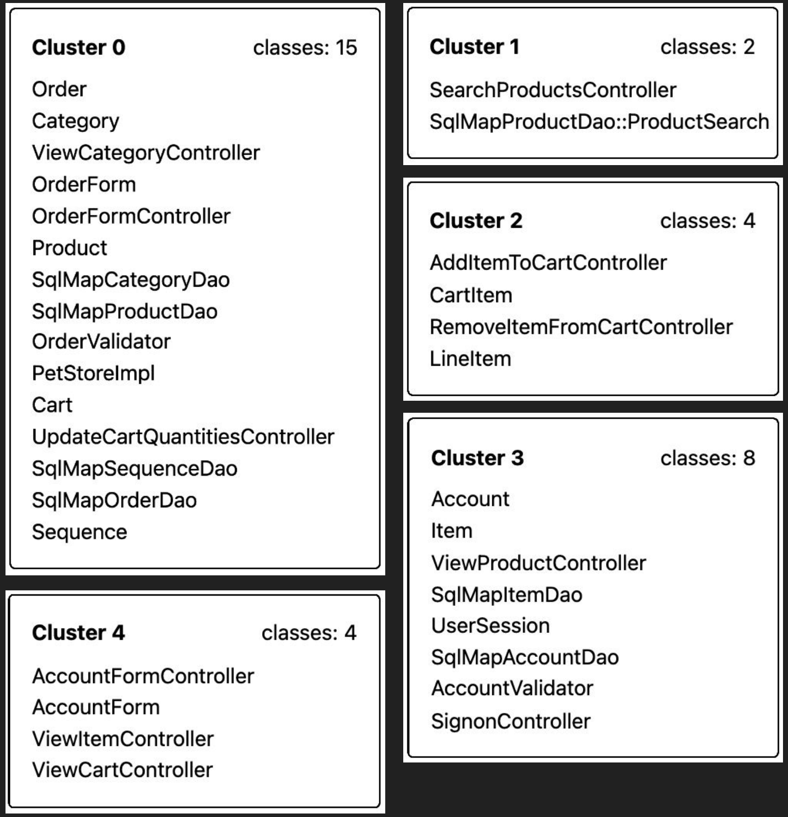
Order
OrderValidator
OrderFormController
OrderForm
SqlMapSequenceDao
Sequence
SqlMapOrderDao
LineItem

FoSCI JPetStore

5 Clusters

2 diff

6 Clusters



Cluster 0 classes: 15

Order
Category
ViewCategoryController
OrderForm
OrderFormController
Product
SqlMapCategoryDao
SqlMapProductDao
OrderValidator
PetStoreImpl
Cart
UpdateCartQuantitiesController
SqlMapSequenceDao
SqlMapOrderDao
Sequence

Cluster 1 classes: 2

SearchProductsController
SqlMapProductDao::ProductSearch

Cluster 2 classes: 4

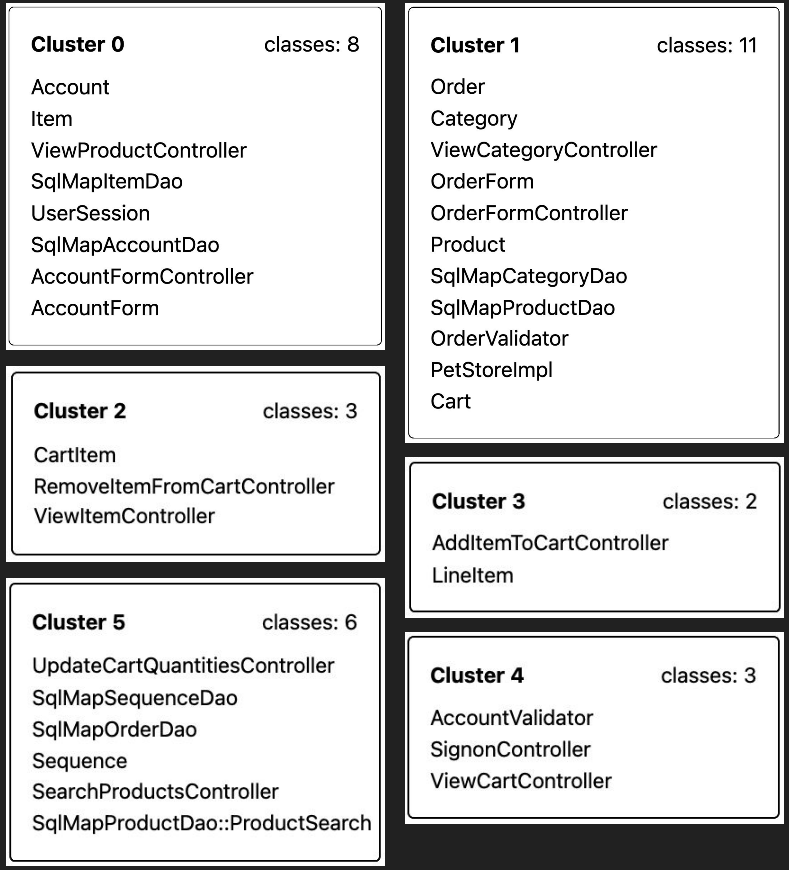
AddItemToCartController
CartItem
RemoveItemFromCartController
LineItem

Cluster 3 classes: 8

Account
Item
ViewProductController
SqlMapItemDao
UserSession
SqlMapAccountDao
AccountValidator
SignonController

Cluster 4 classes: 4

AccountFormController
AccountForm
ViewItemController
ViewCartController



Cluster 0 classes: 8

Account
Item
ViewProductController
SqlMapItemDao
UserSession
SqlMapAccountDao
AccountFormController
AccountForm

Cluster 1 classes: 11

Order
Category
ViewCategoryController
OrderForm
OrderFormController
Product
SqlMapCategoryDao
SqlMapProductDao
OrderValidator
PetStoreImpl
Cart

Cluster 2 classes: 3

CartItem
RemoveItemFromCartController
ViewItemController

Cluster 3 classes: 2

AddItemToCartController
LineItem

Cluster 5 classes: 6

UpdateCartQuantitiesController
SqlMapSequenceDao
SqlMapOrderDao
Sequence
SearchProductsController
SqlMapProductDao::ProductSearch

Cluster 4 classes: 3

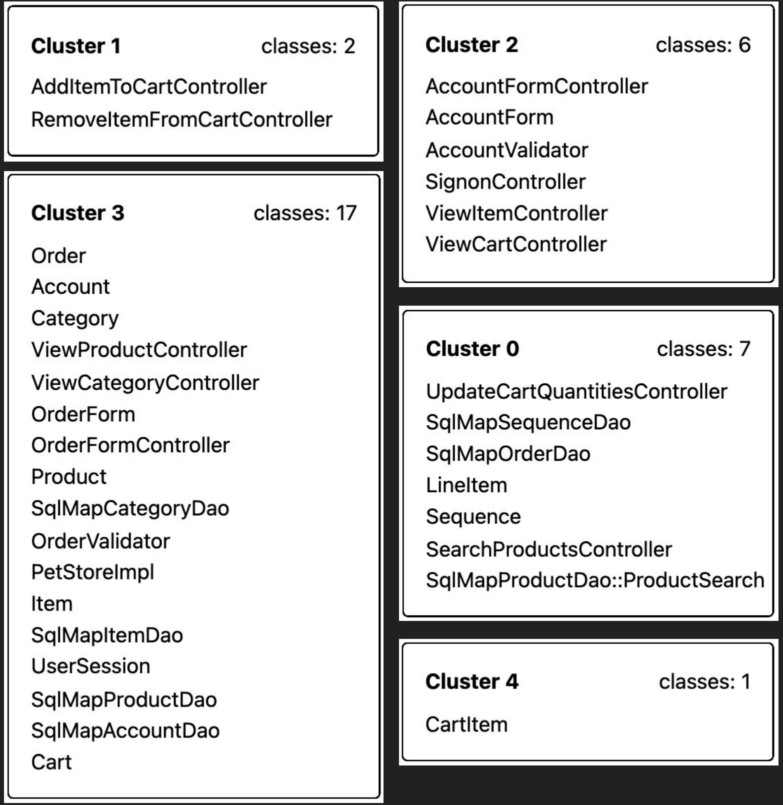
AccountValidator
SignonController
ViewCartController

FoSCI JPetStore

5 Clusters

3 diff

6 Clusters



Cluster 1 classes: 2

AddItemToCartController
RemoveItemFromCartController

Cluster 3 classes: 17

Order
Account
Category
ViewProductController
ViewCategoryController
OrderForm
OrderFormController
Product
SqlMapCategoryDao
OrderValidator
PetStoreImpl
Item
SqlMapItemDao
UserSession
SqlMapProductDao
SqlMapAccountDao
Cart

Cluster 2 classes: 6

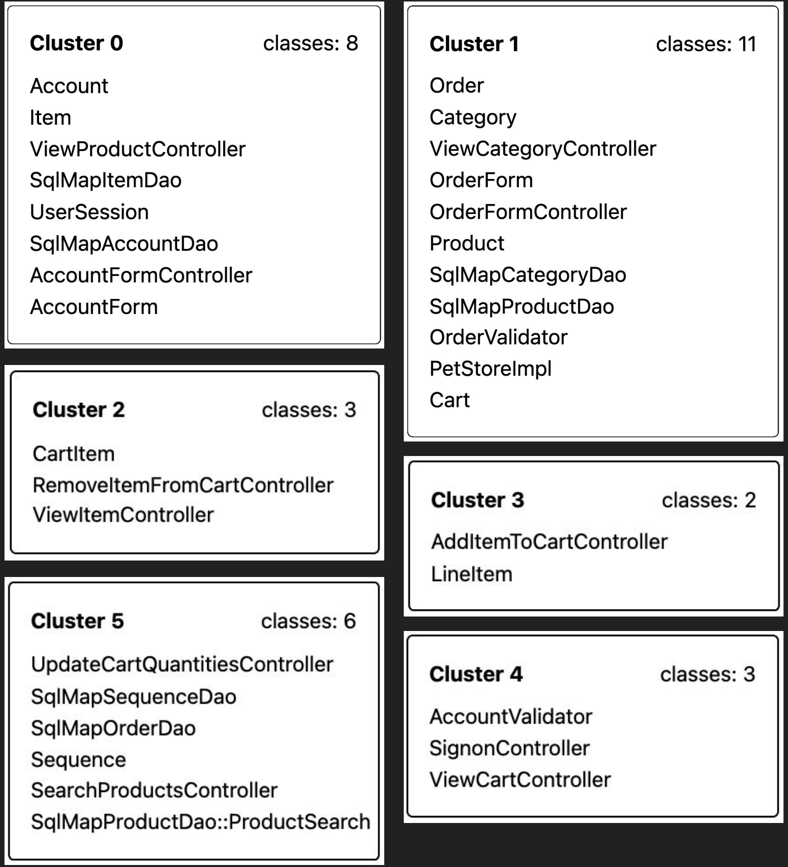
AccountFormController
AccountForm
AccountValidator
SignonController
ViewItemController
ViewCartController

Cluster 0 classes: 7

UpdateCartQuantitiesController
SqlMapSequenceDao
SqlMapOrderDao
LineItem
Sequence
SearchProductsController
SqlMapProductDao::ProductSearch

Cluster 4 classes: 1

CartItem



Cluster 0 classes: 8

Account
Item
ViewProductController
SqlMapItemDao
UserSession
SqlMapAccountDao
AccountFormController
AccountForm

Cluster 2 classes: 3

CartItem
RemoveItemFromCartController
ViewItemController

Cluster 5 classes: 6

UpdateCartQuantitiesController
SqlMapSequenceDao
SqlMapOrderDao
Sequence
SearchProductsController
SqlMapProductDao::ProductSearch

Cluster 1 classes: 11

Order
Category
ViewCategoryController
OrderForm
OrderFormController
Product
SqlMapCategoryDao
SqlMapProductDao
OrderValidator
PetStoreImpl
Cart

Cluster 3 classes: 2

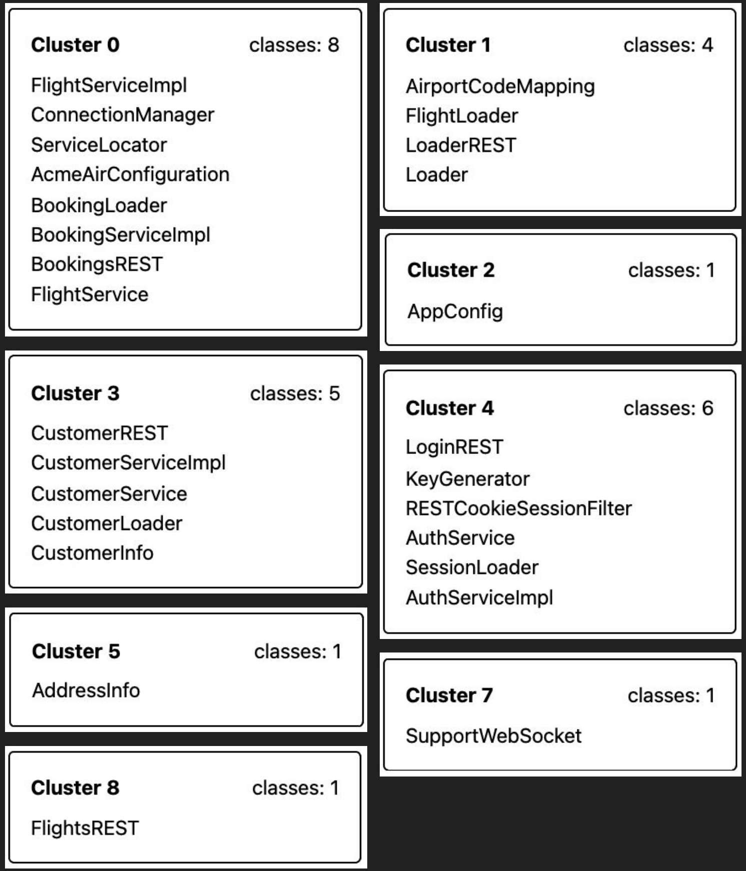
AddItemToCartController
LineItem

Cluster 4 classes: 3

AccountValidator
SignonController
ViewCartController

Mono2Micro AcmeAir

8 Clusters



Cluster 0 classes: 8

FlightServiceImpl
ConnectionManager
ServiceLocator
AcmeAirConfiguration
BookingLoader
BookingServiceImpl
BookingsREST
FlightService

Cluster 1 classes: 4

AirportCodeMapping
FlightLoader
LoaderREST
Loader

Cluster 2 classes: 1

AppConfig

Cluster 3 classes: 5

CustomerREST
CustomerServiceImpl
CustomerService
CustomerLoader
CustomerInfo

Cluster 4 classes: 6

LoginREST
KeyGenerator
RESTCookieSessionFilter
AuthService
SessionLoader
AuthServiceImpl

Cluster 5 classes: 1

AddressInfo

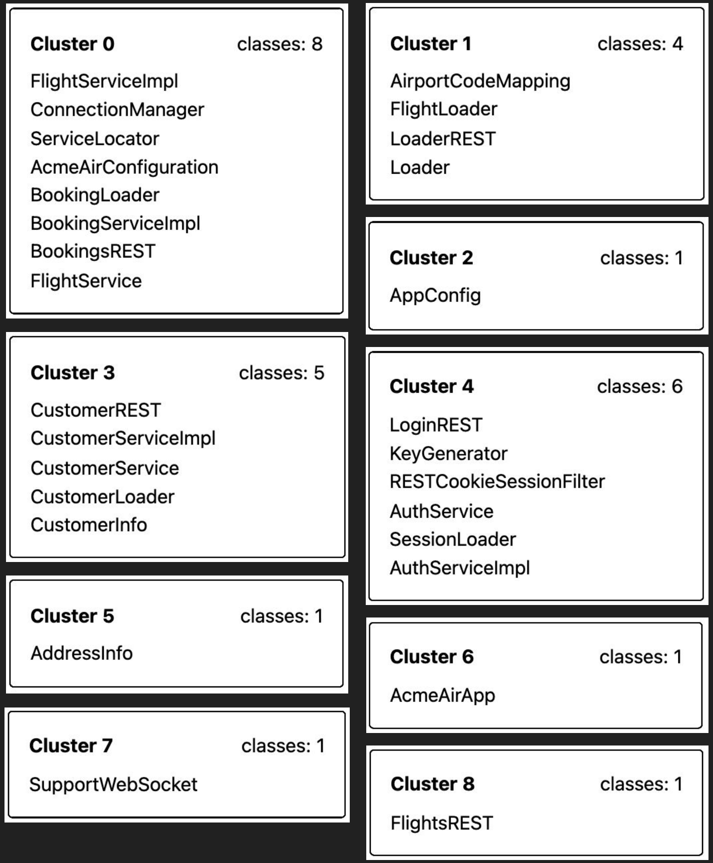
Cluster 7 classes: 1

SupportWebSocket

Cluster 8 classes: 1

FlightsREST

9 Clusters



Cluster 0 classes: 8

FlightServiceImpl
ConnectionManager
ServiceLocator
AcmeAirConfiguration
BookingLoader
BookingServiceImpl
BookingsREST
FlightService

Cluster 1 classes: 4

AirportCodeMapping
FlightLoader
LoaderREST
Loader

Cluster 2 classes: 1

AppConfig

Cluster 3 classes: 5

CustomerREST
CustomerServiceImpl
CustomerService
CustomerLoader
CustomerInfo

Cluster 4 classes: 6

LoginREST
KeyGenerator
RESTCookieSessionFilter
AuthService
SessionLoader
AuthServiceImpl

Cluster 5 classes: 1

AddressInfo

Cluster 6 classes: 1

AcmeAirApp

Cluster 7 classes: 1

SupportWebSocket

Cluster 8 classes: 1

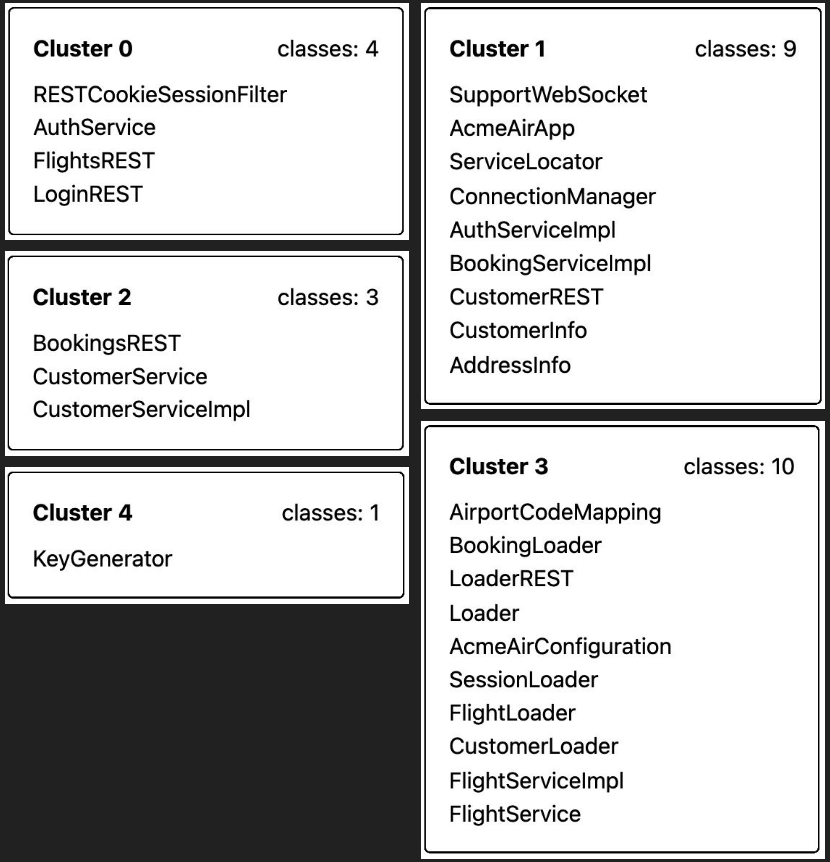
FlightsREST

FoSCI AcmeAir

5 Clusters

2 diff

6 Clusters



Cluster 0 classes: 4

RESTCookieSessionFilter
AuthService
FlightsREST
LoginREST

Cluster 2 classes: 3

BookingsREST
CustomerService
CustomerServiceImpl

Cluster 4 classes: 1

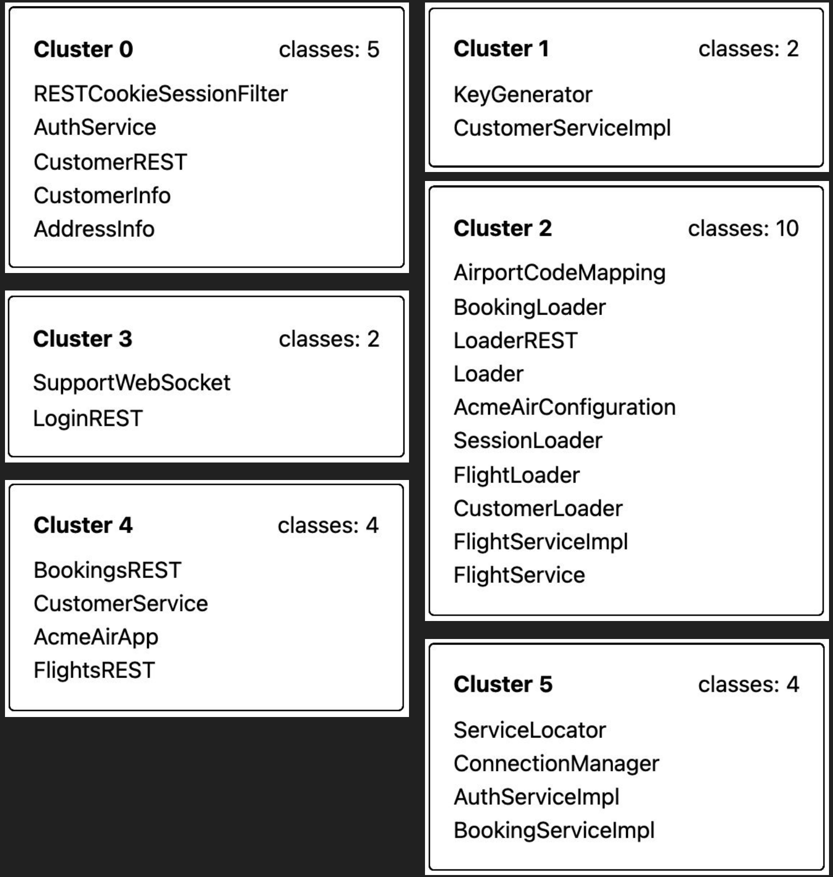
KeyGenerator

Cluster 1 classes: 9

SupportWebSocket
AcmeAirApp
ServiceLocator
ConnectionManager
AuthServiceImpl
BookingServiceImpl
CustomerREST
CustomerInfo
AddressInfo

Cluster 3 classes: 10

AirportCodeMapping
BookingLoader
LoaderREST
Loader
AcmeAirConfiguration
SessionLoader
FlightLoader
CustomerLoader
FlightServiceImpl
FlightService



Cluster 0 classes: 5

RESTCookieSessionFilter
AuthService
CustomerREST
CustomerInfo
AddressInfo

Cluster 3 classes: 2

SupportWebSocket
LoginREST

Cluster 4 classes: 4

BookingsREST
CustomerService
AcmeAirApp
FlightsREST

Cluster 1 classes: 2

KeyGenerator
CustomerServiceImpl

Cluster 2 classes: 10

AirportCodeMapping
BookingLoader
LoaderREST
Loader
AcmeAirConfiguration
SessionLoader
FlightLoader
CustomerLoader
FlightServiceImpl
FlightService

Cluster 5 classes: 4

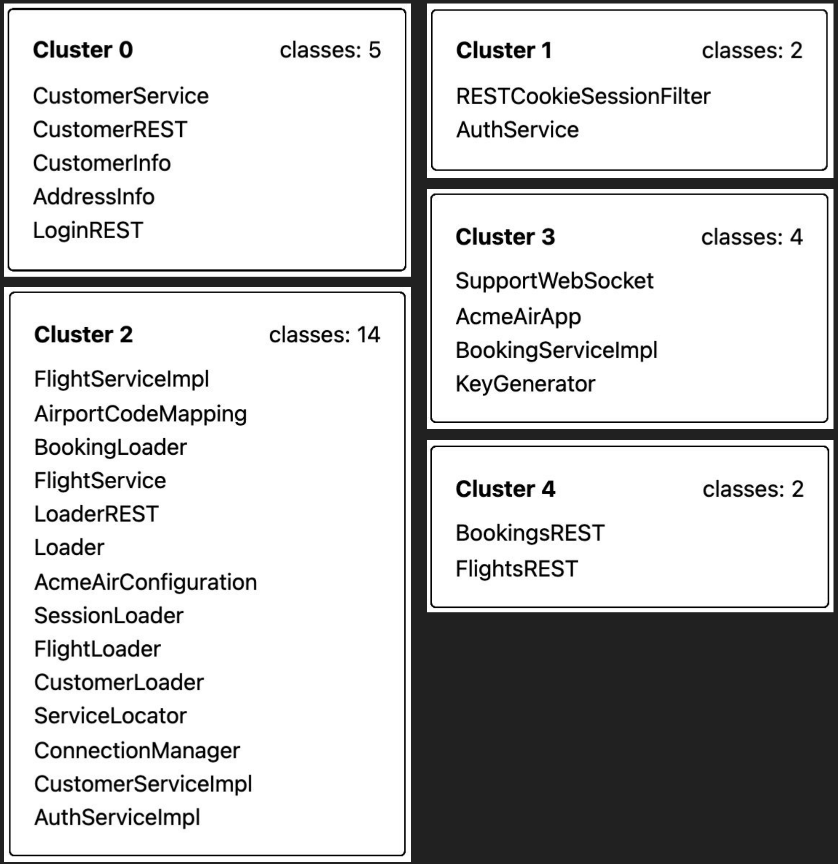
ServiceLocator
ConnectionManager
AuthServiceImpl
BookingServiceImpl

FoSCI AcmeAir

5 Clusters

3 diff

6 Clusters



Cluster 0 classes: 5

CustomerService
CustomerREST
CustomerInfo
AddressInfo
LoginREST

Cluster 2 classes: 14

FlightServiceImpl
AirportCodeMapping
BookingLoader
FlightService
LoaderREST
Loader
AcmeAirConfiguration
SessionLoader
FlightLoader
CustomerLoader
ServiceLocator
ConnectionManager
CustomerServiceImpl
AuthServiceImpl

Cluster 1 classes: 2

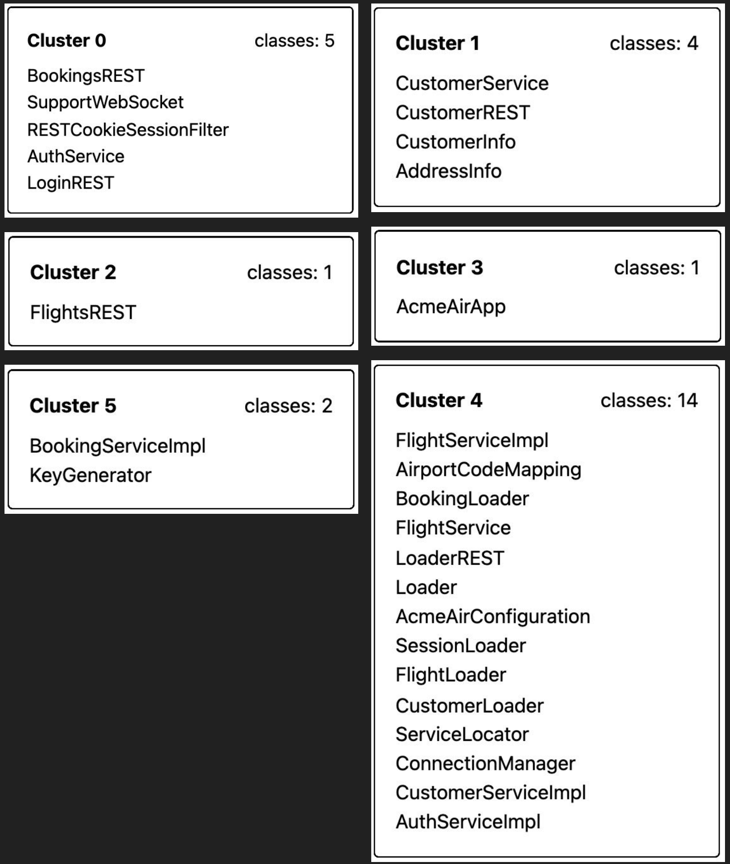
RESTCookieSessionFilter
AuthService

Cluster 3 classes: 4

SupportWebSocket
AcmeAirApp
BookingServiceImpl
KeyGenerator

Cluster 4 classes: 2

BookingsREST
FlightsREST



Cluster 0 classes: 5

BookingsREST
SupportWebSocket
RESTCookieSessionFilter
AuthService
LoginREST

Cluster 2 classes: 1

FlightsREST

Cluster 5 classes: 2

BookingServiceImpl
KeyGenerator

Cluster 1 classes: 4

CustomerService
CustomerREST
CustomerInfo
AddressInfo

Cluster 3 classes: 1

AcmeAirApp

Cluster 4 classes: 14

FlightServiceImpl
AirportCodeMapping
BookingLoader
FlightService
LoaderREST
Loader
AcmeAirConfiguration
SessionLoader
FlightLoader
CustomerLoader
ServiceLocator
ConnectionManager
CustomerServiceImpl
AuthServiceImpl

Evaluation - Which approach is optimal?

- Determining the most effective clustering approach for the monolith presents a significant challenge
- Each clustering method is designed to optimize specific metrics
- The prototype includes multiple clustering approaches to enable a more informed decision when migrating a monolithic project

Current Limitations and Challenges

- Optimization of metrics computation for explainability on the way
- Varying approaches require specific input sets
- Some legacy projects lack access to Git logs
- FoSCI analysis is time-intensive on large data sets (over 10 MB)
- Capturing quality execution traces in legacy systems is challenging due to complexity

Thank you!

- Questions

Reach Out for collaboration!

Bruno Rossi
brossi@mail.muni.cz

Michal Skipala
skipala3@gmail.com