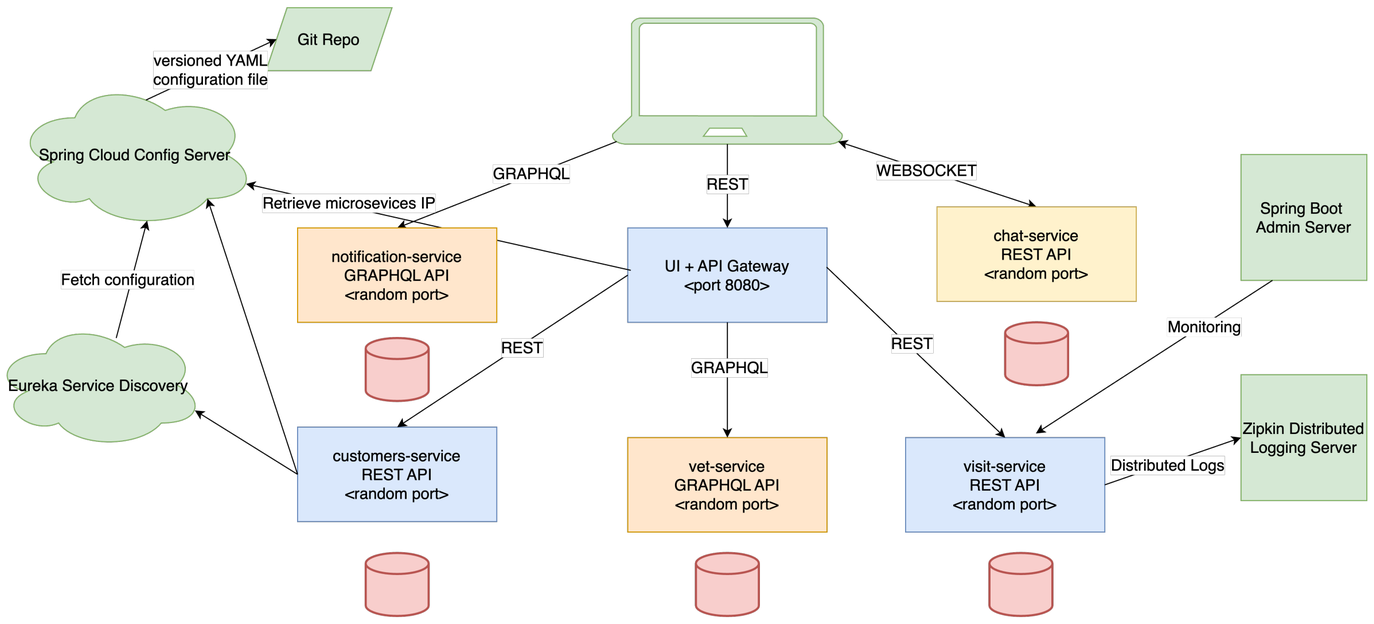
Static Analysis of Microservices using GraalVM

Vsevolod Pokhvalenko
xpokhv00@stud.fit.vutbr.cz

BRNO UNIVERSITY OF TECHNOLOGY
FACULTY OF INFORMATION TECHNOLOGY

Excel@FIT 2025

Microservices & Cloud Development



Independent, modular services

Despite their advantages in scalability and flexibility, microservices introduce non-trivial challenges in managing complexity and ensuring maintainability.

- Hidden service dependencies
- Obscure communication paths
- Protocol diversity (REST, WebSocket, GraphQL)
- Insufficient observability of distributed data flows

Service Communication: Code Examples

REST Endpoint

```
@GetMapping("{userId}")
public List<CatalogItem> getCatalog(
  @PathVariable("userId") String userId) {
  UserRating ratings = restTemplate.getForObject(
    "http://rating-data-service/ratingsdata/users/" + userId,
    UserRating.class);
}
```

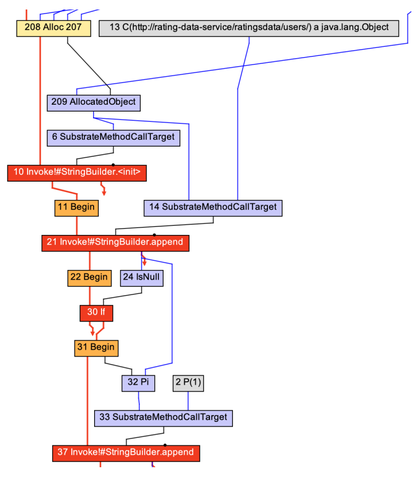
WebSocket (STOMP)

```
@Override
public void registerWebSocketHandlers(
  WebSocketHandlerRegistry registry) {
  String[] rooms = {"room1", "room2", "room3"};
  for (String room :  rooms) {
    registry.addHandler(
      new MyWebSocketHandler(room),
      "ws" + room).setAllowedOrigins("*")
  }
```

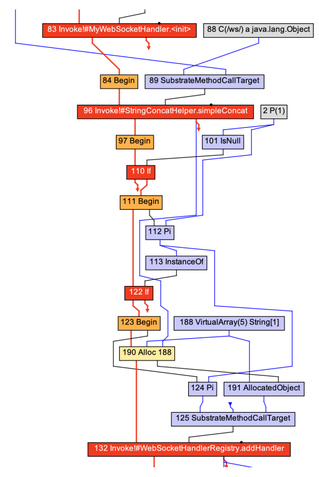
GraphQL Resolver

```
List<ProductResponse> productList = graphQlClient
  .document(productGraphQLQuery)
  .operationName(productGraphQLOperationName)
  .variable(productGraphQLVariableName, productIdList)
  .retrieve("getProductGraphQLByListId")
  .toEntityList(ProductResponse.class).block();
```

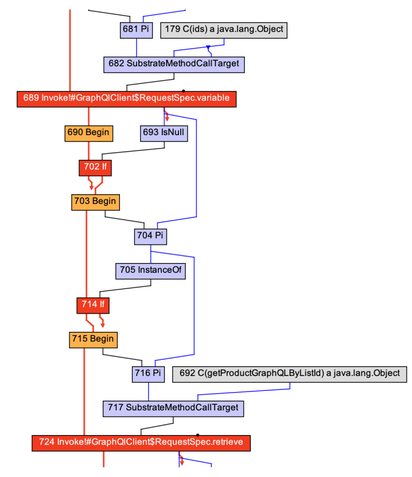
Graal Intermediate Representation



Graal IR for REST service endpoint logic

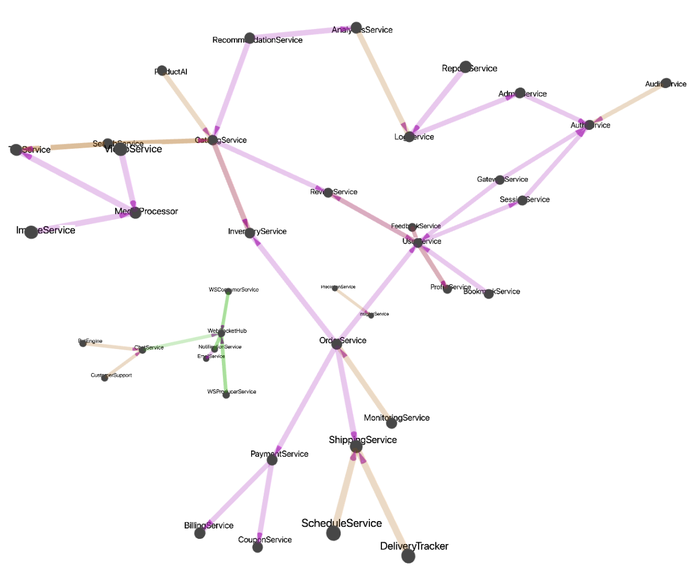


Graal IR for WebSocket setup and handlers

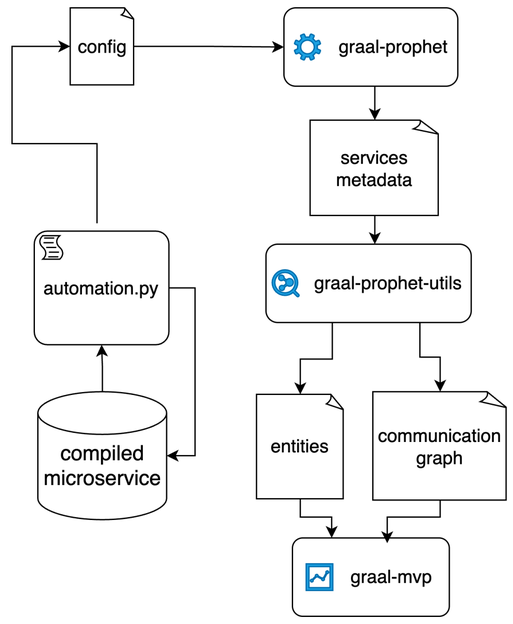


Graal IR for GraphQL query resolver logic

Extracted Architecture Visualization



Graal-MVP visualizing service view from extracted metadata



MicroGraal architecture overview showing extraction pipeline

Graal-Prophet analyzes microservices during the **native-image** build, extracting communication and type metadata.

Graal-MVP turns this data into clear visualizations of REST, GraphQL, and WebSocket communication.

Domain and **Service** views assist in understanding and documenting the distributed system architecture.

Bachelor thesis presented at **FIT BUT** Static Analysis of Microservices using GraalVM Supervised by **Ing. David Kozak**