

Improving modifiability in microservices architectures



A reference implementation for a remote work platform in Software Startups

Author: Thiago Guerrero¹

Advisors: Prof. Dr. Eduardo Martins Guerra²,
MSc. João Francisco Lino Daniel²

¹ Instituto de Matemática e Estatística da Universidade de São Paulo (IME-USP)

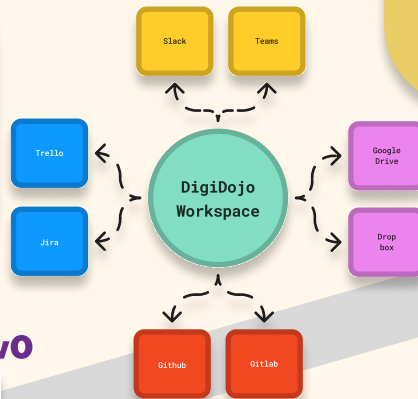
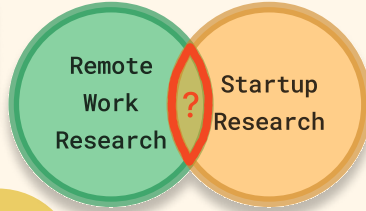
² Free University of Bozen-Bolzano (Unibz)

a remote work platform in Software Startups?



The Digi-dojo project is a platform designed to support development teams in Software Startups that work remotely. The goal is to provide the interactive and creative environment of a startup for these teams.

It's expected that the scientific results obtained will fill in the knowledge gap in the literature by adding a better understanding of remote work in startups.



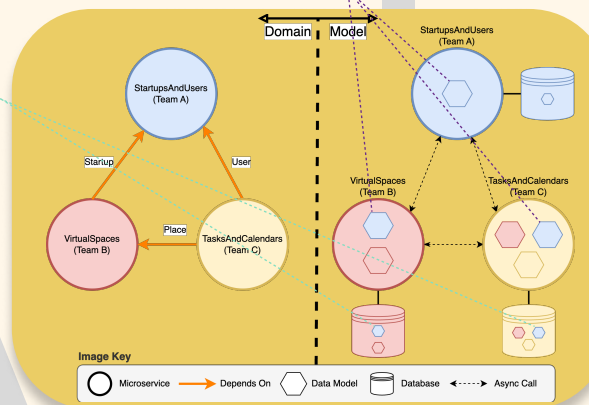
What is modifiability?

Modifiability is a non-functional attribute of the software architecture that measures the cost of making changes and reflects the ease with which a software system can accommodate modifications.

The view on modifiability we will work with consists of two types of costs: the cost of preparing the system for changes and the cost of making changes.

Spotting lack of Modifiability

Data Model Duplication!



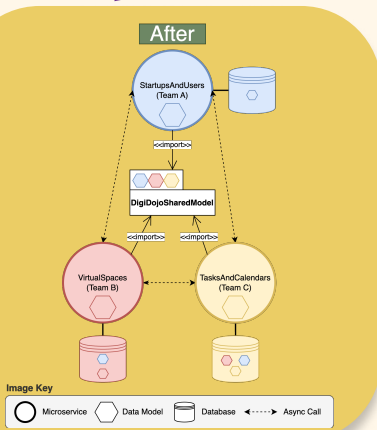
Digi-Dojo v0



Data Duplication!

Proposed Solutions

Data Model Duplication



Bad Architecture

```

EntityManagerService/src/main/java/
it/unibz/digidojo/entitymanagerservice
├── common/kafka
│   ├── BaseConsumer.java
│   ├── BaseProducer.java
│   └── KafkaConfig.java
├── startup
│   ├── application
│   │   ├── StartupController.java
│   │   └── StartupProducer.java
│   └── domain
├── teammember
├── user
├── util
└── EntityManagerServiceApplication.java
    └── ExceptionHandlerAdvice.java
    
```

```

ActivityPlannerService/src/main/java/
it/unibz/digidojo/activityplannerservice
├── application
│   ├── api
│   │   ├── CalendarEventController.java
│   │   ├── PlaceController.java
│   │   ├── TaskController.java
│   │   └── UserController.java
│   └── kafka
│       ├── consumer
│       │   ├── BaseConsumer.java
│       │   ├── PlaceConsumer.java
│       │   └── UserConsumer.java
│       └── producer
├── config
├── domain
├── util
└── ActivityPlannerServiceApplication.java
    
```

```

MService-StartupsAndUsers/src/main/java/com/startupsdigidojo/usersandteams
├── startup
│   ├── application
│   │   ├── kafka
│   │   │   ├── StartupKafkaConfiguration.java
│   │   └── TeamMemberKafkaConfiguration.java
│   └── domain
├── teamMember
│   ├── application
│   │   ├── kafka
│   │   └── UserKafkaConfiguration.java
│   └── domain
├── user
│   ├── application
│   │   ├── kafka
│   │   └── UserKafkaConfiguration.java
│   └── domain
├── ExceptionHandlerAdvice.java
└── UsersAndTeamsApplication.java
    
```

Bad Architecture!

```

MService-TaskAndCalendars/src/main/java/it/unibz/taskcalendarservice
├── calendar
│   ├── application
│   │   ├── KafkaConfiguration.java
│   │   ├── PlaceConsumer.java
│   │   ├── PlaceController.java
│   │   ├── UserConsumer.java
│   │   └── UserController.java
│   └── domain
├── task
└── TaskAndCalendarServiceApplication.java
    
```

Data Duplication

