

Workflow Management System Through Microservice Approach

Snehal Shyam Nrupnarayan

Sipna College of Engineering and
Technology
Amrawati, India
payalnr05@gmail.com

Seema Rathod

Sipna College of Engineering and
Technology
Amrawati, India
seemarathodf47@rediffmail.com

Abstract— The hereby study explains the functioning of a workflow management system. A key aspect in successful project development is proper planning and timely execution. Workflow management system mainly comes into tracking project development to have detailed insights into whole development process. The goal is to develop a lightweight, easily accessible application to ease out workflow process. This review paper highlights the efficiency of microservice architecture and use of Golang for development.

Keywords— workflow, planning, management, lightweight, microservice

I. INTRODUCTION

Project development is a planned activity with continuous integration and deployment of functional patches. A typical software product goes through a complete development cycle which involves Designing, Development, Testing and finally Deployment. As far from coding perspective, modularity is a key to cover all possible scenarios, same applies when the whole integrated product is considered. The project or a product is broken down into different components and modules at an architectural level. Designing, Development and Testing of all these components are done and then the final integration is done which again goes through rigorous testing. This in turn indicates that project development is an interdependent process of different components.

II. FUNCTIONING OF WORKFLOW MANAGEMENT SYSTEM

As explained earlier, interdependent components have further granularized into independent tasks. These tasks are completed through a workflow which in turn results into a component completion. The more are the granular tasks, more details development can be achieved.

The project deals with a way of providing methodical approach to carry out workflow and thus help to provide precise execution. The project aims at developing lightweight, highly interactive and easily accessible web application. The workflow will typically consist of following states:

- a) TO DO
- b) IN PROGRESS
- c) IN REVIEW

d) COMPLETED



Fig 1. Workflow States

III. MICROSERVICES

Microservice is an architectural of structuring application into loosely coupled services, which implement business logic. The very concept of microservice arrived due to the need to have isolation between the services for there uninterrupted usage.

A monolithic service serves different requests received in a single server through in-process calls. This is not feasible for a highly scalable project as these calls will be much time consuming. On the other hand microservices run inside their own dedicated server inside a container. They are in complete isolation of other services and hence serves the purpose of seamless request handling.

A process running all the services once killed or restarted will make all services shut down. This central dependency makes request handling heavy and continuous availability of data will not be possible. Microservices as running separately will be available even in breakdown of other services.

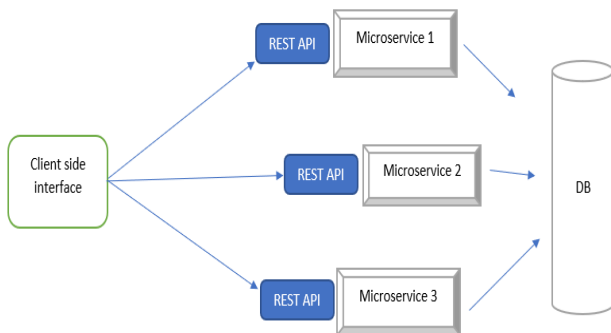


Fig 2. Microservice architecture

IV. GOLANG

The proposed idea is to develop micro-services in Golang, a recent programming language developed by Google. Go is really fast as it is compiled to machine code. It compiles in tradition with C. Hence it outperforms the languages which require virtual runtime. The compilation is also very fast and the resulting binary API size is also way too small. The proposed work is to implement micro-services in Golang for a workflow management system used to track project development.

V. CONCLUSION

Thus a thorough study of microservices architecture and associated survey of research papers reveal, how micro-services can be advantageous and efficient. The another coining factor is using Go for developing these services. It minimizes development duration, eases the deployment process and provides lightweight services hosting REST API.

REFERENCES

- [1] Jose Ignacio Ferná'ndez-Villamor, Carlos A. Iglesias, Mercedes Garijo, "MICROSERVICES: LIGHTWEIGHT SERVICE DESCRIPTIONS FOR REST ARCHITECTURAL STYLE"
- [2] "Concurrency in Go and Java: Performance Analysis" by Naohiro Togashi Software Engineering Lab University of Aizu Aizuwakamatsu, Fukushima, Japan , Vitaly Klyuev Senior Associate Professor University of Aizu Aizuwakamatsu, Fukushima, [978-1-4799-4808-6 /14/ ©2014 IEEE]
- [3] Fahim Shariar Shoumik, Md. Ibna Masum Millat Talukder, Ahmed Imtiaz Jami, Neeaz Wahed Protik, Md. Moinul Hoque, " Scalable Micro-Service based Approach to FHIR Server with Golang and No-SQL", 2017 20th International Conference of Computer and Information Technology (ICCIT), 22-24 December, 2017
- [4] G.Alonso ,D. Agrawal ,A. El Abbadi, C. Mohan , " Functionality and Limitations of Current Workflow Management Systems"
- [5] " An Introduction to Programming in Go" Copyright c 2012 by Caleb Doxsey
- [6] The Go Programming Language Official Website, <http://golang.org>