# Architecture Review Board Document
for
Firebase Elastic Integration

Prepared by:

Kai Fedin (3515541)
Majd Bousaad (3519015)
Alan Tofeq (3286019)
Nilusche Liyanaarachchi(3272466)

FH Aachen University of Applied Sciences, Germany
Interdisciplinary Project

Aachen, 24. October 2023

Contents

Chapter 1

Executive Summary

1.1 Purpose

The ARB is essential for ensuring that the high-level architectural decisions made for the project align with the organization's strategic goals and industry best practices. It plays a pivotal role in identifying and mitigating potential risks associated with architectural decisions, ultimately contributing to a more robust and reliable system. Ultimately, we use this document to discuss a solution and its merits to a business problem with stakeholders and software architects.

1.2 Intended Audience

This document is intended for product owners, management, developers, and other stakeholders who are involved in the planning, design, and execution of a software project.

1.3 Intended Use

The ARB serves as a guidance document, overseeing the architectural decisions of a software project, and provides a framework to adopt technical solutions to a business problem.
Additionally, the ARB serves as a resource for resolving architectural disputes and clarifying design choices, which can lead to smoother development processes. For this project in particular Stakeholders, which can include business owners and end-users, benefit from the ARB by gaining assurance that the project's architectural decisions are made with strategic alignment and compliance with industry standards.

1.4 Summary of Major Architectural Decisions

Elastic Cloud instance
Instance to run Kibana as an exploratory data analysis tool

- Supports Spaces and Role Creation
- Supports Exploratory Analysis Dashboard Tool
- Supports Horizontal Scalability
- Supports optional fast, database searches via elasticsearch

Data Pipelines
Pipelines that extract data from Firebase to upload to Elastic.

- Runs on Cloud Run and Cloud Scheduler of Google Cloud to support no cost in Google Cloud's free tier.
- Set up via Terraform to ensure minimal input from the developer's side

Chapter 2

Overview

2.1 Introduction

Dagmarverse is a web-based platform with a content management system regarding financial education. This MVP is an additional integration to this platform that features a tool with which the creators of the platform can perform exploratory data analysis on user behavior data.

2.2 Background

The application consists of several databases containing application data as documents in Firestore and additional content data in several other data storages.
Application data about user behavior and user demographics will be mocked in one Firestore database. This database serves as the foundation of all the data that can be tracked via a tracking solution and pipelined to the analysis tool Kibana.

2.3 Design Goals

1. **Compatibility**:
   The current application is built and sustained with the Google Cloud Platform (GCP). The components of the MVP will be set up for easy integration into GCP.
2. **Usabilty and User Experience**:
   Ensure that the tool is user-friendly and intuitive for creators to navigate and utilize effectively
3. **Scalability and Performance**:
   With the expected growth in customer base, the system architecture needs to handle a growing volume of user data and analysis requests without compromising data flow and processing.
4. **Maintenance and Supportability**:
   Create a system that is easy to maintain, with clear processes for troubleshooting and updates as needed over time.
5. **Security and Privacy**:
   Implement robust security measures to protect sensitive user data, ensuring accessibility for different types of customers

Chapter 3

System Design

3.1 Requirements

3.1.1 Business Requirements

- **Data Analysis Capabilities**: The integration provides creators with the ability to perform exploratory data analysis on user behavior and demographics data stored in one Firestore database.
- **User-Friendly Interface**: The tool should have an intuitive and user-friendly interface to facilitate easy navigation and utilization for creators
- **Integration Efficiency**: The integration process between Firebase and Kibana should be efficient and reliable to ensure timely access to the analysis tool.

3.1.2 Functional Requirements

- **Data Upload and Retrieval**: Creators must be able to choose user data from the Firebase database and retrieve it for analysis in Kibana
- **Data Analysis Function**: The tool should provide basic data analysis functions including filtering, aggregations, and visualization options.
- **Access Management**: Different Creators must be able to create independent analyses without breaching the data of other users
- **Report Extraction**: Additionally to the exploratory data analysis, the tool should provide a way to extract a snapshot of relevant reports and charts in an appropriate file format.

3.1.3 Technical Requirements

- **Firebase Database Setup**: Ensure that a Firestore database is properly configured to store user behaviour data
- **Kibana Integration Configuration**: Set up necessary configurations to establish a Server running the data analysis tool
- **Data Pipeline Implementation**: Develop a data pipeline mechanism to facilitate the transfer of user tracking data from Firebase to Kibana.
- **Security and Access Control**: Implement measures to safeguard sensitive data and control access to the data analysis tool.
- **Documentation and Knowledge Sharing**: Create comprehensive documentation to guide creators in utilizing the integration effectively.

3.2 Future Work (out of scope)

- **Real-Time Data Processing**: Explore options for real-time data processing and analysis to enable creators to monitor and analyze user behavior data as it is generated.
- **Integrate with Additional Data Sources**: Extend the integration capabilities to include other data sources or databases, allowing for a broader range of data to be analyzed
- **Optimization for Large Datasets**: As the application database grows with its users, address performance considerations for handling large volumes of tracking data.
- **User Feedback and Iterative Development**: Gather feedback from creators and stakeholders to identify areas for improvement and iterate on the tool's functionality and user experience.
- **Replace NoSQL Integration with Relational Database** In the event of a changed database schema, the data processing pipelines need to be extended to be compatible with the data analysis tool

3.3 Architecture

3.3.1 Legacy Architecture

The existing architecture consists of several different components:

- Frontend: The application interface for customers to consume content. The technologies here are irrelevant while extending the architecture
- Database: Firebase NoSQL database hosted on the Google Cloud Platform.
- Backend: Built with Sveltekit, connection to the database
- CRM: ActiveCampaign, Software as a Service Platform to manage contacts, workflows

To integrate the proposed Data Analysis tool Kibana into the application, only the Database needs to be considered.

3.3.2 Architecture Diagram

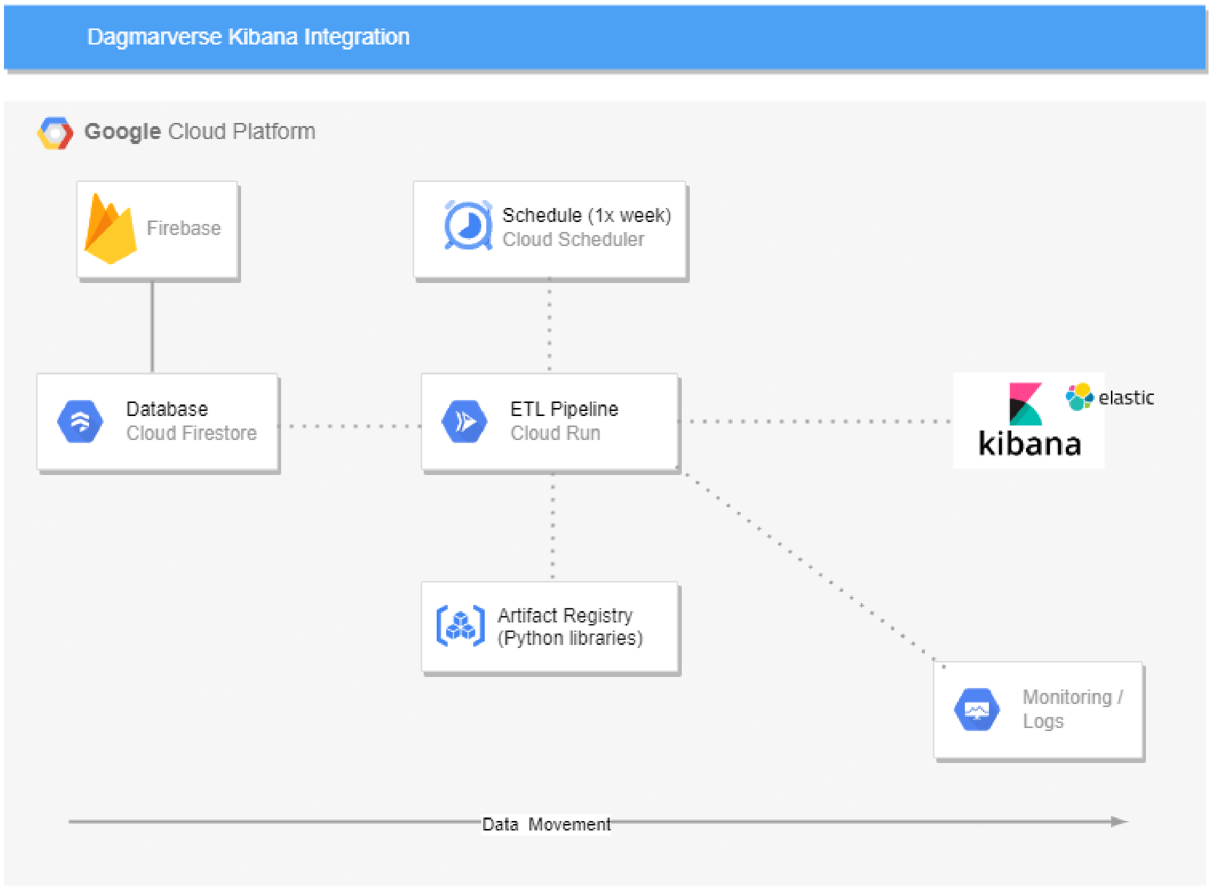


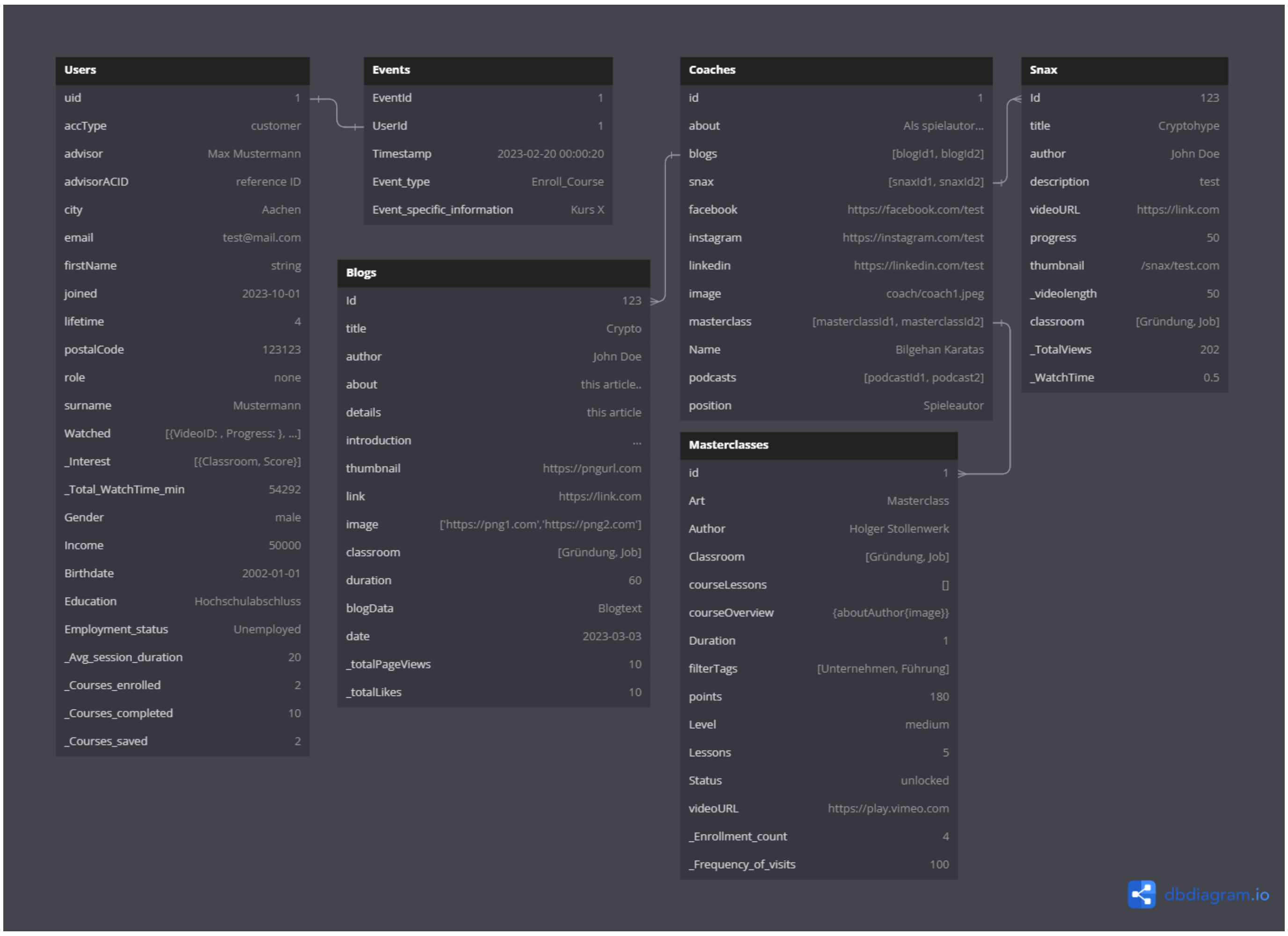
Figure 3.1: Architecture Diagram for proposed

3.3.3 Subsystems

Database (Cloud Firestore): Host database that includes all tracked data
ETL Pipeline: Cloud Run instance that runs a python script to extract data
Schedule: Cloud Scheduler instance that schedules the Cloud Run service
Artifact Registry: Contains all application code
Monitoring / Logs: Google Cloud's Monitoring service to watch out for service downtime
Kibana / Elastic: The exploratory data analysis tool

3.3.4 Data Model

As a NoSQL Database tool, Firebase relations between tables are implicit. Field attributes with an underscore prefix "_" are metrics that are calculated in the Pipeline.
We have excluded some tables (such as Podcasts etc.) as the following tables are the ones that have been discussed as necessary for an analysis.
We have added a new table (Events) to track a user's movement on the webpage further. Any Tracking measure for additional table fields and metrics that we deem helpful for analysis and are not captured by the application already, needs to be implemented outside of this project.



Users	
uid	1
accType	customer
advisor	Max Mustermann
advisorACID	reference ID
city	Aachen
email	test@mail.com
firstName	string
joined	2023-10-01
lifetime	4
postalCode	123123
role	none
surname	Mustermann
Watched	[{VideoID: , Progress: }, ...]
_Interest	[{Classroom, Score}]
_Total_WatchTime_min	54292
Gender	male
Income	50000
Birthdate	2002-01-01
Education	Hochschulabschluss
Employment_status	Unemployed
_Avg_session_duration	20
_Courses_enrolled	2
_Courses_completed	10
_Courses_saved	2

Events	
EventId	1
UserId	1
Timestamp	2023-02-20 00:00:20
Event_type	Enroll_Course
Event_specific_information	Kurs X

Blogs	
Id	123
title	Crypto
author	John Doe
about	this article..
details	this article
introduction	...
thumbnail	https://pngurl.com
link	https://link.com
image	['https://png1.com','https://png2.com']
classroom	[Gründung, Job]
duration	60
blogData	Blogtext
date	2023-03-03
_totalPageViews	10
_totalLikes	10

Coaches	
id	1
about	Als spielautor...
blogs	[blogId1, blogId2]
snax	[snaxId1, snaxId2]
facebook	https://facebook.com/test
instagram	https://instagram.com/test
linkedin	https://linkedin.com/test
image	coach/coach1.jpeg
masterclass	[masterclassId1, masterclassId2]
Name	Bilgehan Karatas
podcasts	[podcastId1, podcast2]
position	Spieleautor

Snax	
Id	123
title	Cryptohype
author	John Doe
description	test
videoURL	https://link.com
progress	50
thumbnail	/snax/test.com
_videolength	50
classroom	[Gründung, Job]
_TotalViews	202
_WatchTime	0.5

Masterclasses	
id	1
Art	Masterclass
Author	Holger Stollenwerk
Classroom	[Gründung, Job]
courseLessons	[]
courseOverview	{aboutAuthor{image}}
Duration	1
filterTags	[Unternehmen, Führung]
points	180
Level	medium
Lessons	5
Status	unlocked
videoURL	https://play.vimeo.com
_Enrollment_count	4
_Frequency_of_visits	100

Figure 3.2: Tables with Metrics to analyze

3.3.5 State Management

For each stateful system in your architecture diagram we outline its function and replication method.

Firestore Database:
Function: Firestore is used to store data from your Firebase application.
Replication Method: Firestore automatically replicates data across multiple regions for high availability.

Cloud Run (Executing Python ETL Script):
Function: Cloud Run executes a Python ETL script responsible for extracting data from Firestore, transforming it, and loading it into Kibana (Elastic) for further analysis and visualization.
Replication Method: Cloud Run instances can be horizontally scaled to handle varying ETL workloads. The statelessness of Cloud Run allows for easy scaling, and data consistency is maintained through the ETL process. Since the ETL job will only run once a week, replication might not be necessary to avoid additional cost.

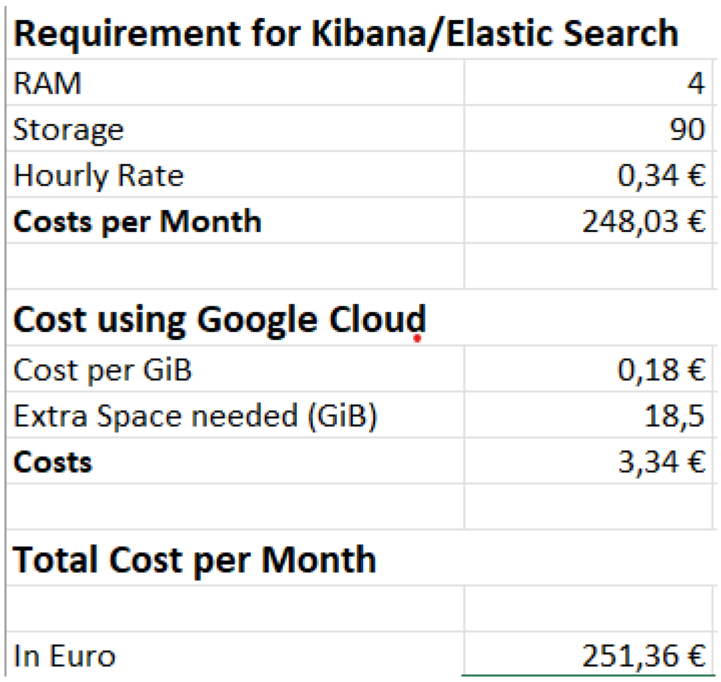
Google Cloud Monitoring:
Function: Google Cloud Monitoring provides insights into the performance of your Cloud Run instances, Firestore database, and other monitored resources.
Replication Method: Google Cloud Monitoring itself is a managed service and doesn't require explicit replication. It operates across multiple regions for redundancy. Ensure that monitoring is configured for both Firestore and Cloud Run to track resource usage, errors, and other relevant metrics during the ETL process.

Kibana (Elastic):
Function: Kibana is used for data visualization and analysis. It receives the transformed data from the ETL process and provides a user-friendly interface for exploring and understanding the data.
Replication Method: Elasticsearch, which is likely the underlying data store for Kibana, can be configured for replication and sharding to ensure high availability and fault tolerance. Check Elasticsearch's settings for index replication and distribution across nodes.

3.3.6 Scalability & Limits

To ensure the reliable availability of the application the elastic instances should be Hosted to scale horizontally. This means that additional servers will handle increased data volume and traffic as the application grows.
During testing, we have narrowed the number of Kibana instances down to at least 2 instances to ensure sufficient concurrent use of more than 100 users.

3.3.7 Performance & Cost Estimates



Requirement for Kibana/Elastic Search	
RAM	4
Storage	90
Hourly Rate	0,34 €
Costs per Month	248,03 €
Cost using Google Cloud	
Cost per GiB	0,18 €
Extra Space needed (GiB)	18,5
Costs	3,34 €
Total Cost per Month	
In Euro	251,36 €

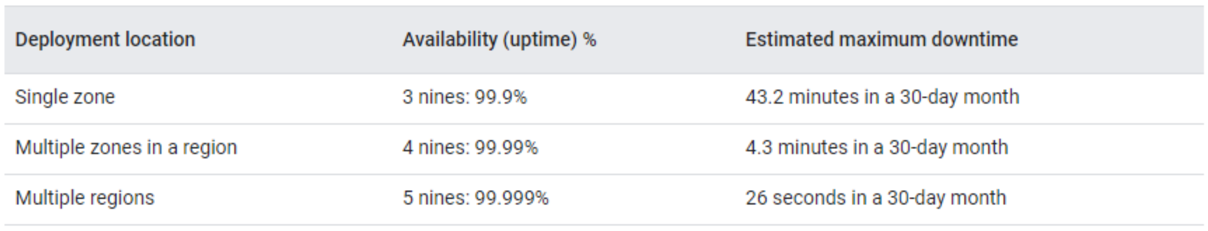
Figure 3.3: Cost Analysis

Chapter 4

Operations & Management

4.1 Availability & Reliability

Google Cloud infrastructure is designed to support the following target levels of availability for most customer workloads:



Deployment location	Availability (uptime) %	Estimated maximum downtime
Single zone	3 nines: 99.9%	43.2 minutes in a 30-day month
Multiple zones in a region	4 nines: 99.99%	4.3 minutes in a 30-day month
Multiple regions	5 nines: 99.999%	26 seconds in a 30-day month

Figure 4.1: Google Cloud Reliability across Zones

Since the Elastic Cloud infrastructure will run on at least 2 different Zones we can expect a 4 nines Reliability and the Pipeline will have at least 3 nines availability.
Therefore a downtime of 43.2 minutes in a 30-day month is acceptable.

4.2 Incident Response

Does degradation or downtime on your system impact other processes?

- Main Application System NO
- Main Application Experience NO
- CRM System NO

How will the application be monitored?
Google Cloud Monitoring will be implemented and additional uptime checks can be activated for the computing instances.
How will the application owners be notified in case of downtime?
Notifications to code owners in Google Cloud are set up automatically when creating uptime checks.

4.3 Testing

To ensure that the application is production-ready, it is essential to perform rigorous testing and benchmarking. This includes several types of testing, such as unit testing, integration testing, performance testing, and security testing.

Unit testing involves testing individual components of the pipeline scripts to ensure that they are working correctly. Integration testing involves testing the interaction Systems such as Kibana and Elasticsearch to ensure that are working together as expected.

Once the testing is complete, benchmarking can be performed to measure the performance and scalability of the application. This involves measuring key performance indicators such as response time, throughput, and resource utilization, under different levels of load.

4.4 Deployment

The Data Pipelines will be deployed via Terraform - An Infrastructure as Code Tool, which can be used on the already existing Google Cloud instance.Configuration for this method is found in the main repository.
The main Elastic instance can be deployed easily on `https://cloud.elastic.co/`

Chapter 5

Instrumentation

5.1 Logging

Google Cloud Logging captures stdout and stderr of services (containers) deployed to Google Cloud Run. You should be able to view these logs either through the Cloud Console's Logs Viewer `https://console.cloud.google.com/logs/query` or using gcloud.

5.2 Monitoring

Monitoring is done via Google Cloud. Appropriate resource utilization dashboards need to be set up first.

Chapter 6

Security

6.1 Authentication

Confirm that the application will be using standard SSO authentication for both user and M2M authentication

6.2 Authorization

Outline how authorization is performed. What roles and permissions are used to restrict access to services/resources: YES
Can customers and/or vendors authenticate? Choose One: YES

- If YES, what is the expected behaviours if the user changes their email address?
  The user cannot change their email address and/or other credentials.
- If mixed (employees + external users) how are you validating that the logged-in user is a Dagmarverse Employee?
  Designated user credentials are created by adminusers and submitted to the Dagmarverse Employees.

6.3 Sanitization

Outline any sanitization measures being applied to systems to ensure regular cleaning of bash histories, and and old folders. Is data being retained only as long as it is legally required?

- Data is retained only as legally required, in compliance with applicable regulations such as the DSVGO or other data retention policy.

6.4 Code Analysis

Provide all project GitHub repositories:

- `https://github.com/majdbousaad/MVPTracking`

Confirm that the following are enabled for each git repository:

- Depandabot:Howto YES

- CodeQL: Howto YES
- Secret Scanning: Howto YES

Confirms all critical and high vulnerabilities are remediated: YES