

Technical Design Document

Investment Workbench ESG

Status: Draft
Version: 2.4

Prepared by: Synechron
Project Manager: TBD
Project Sponsor: TBD

Date Created: 09/09/2020
Date Last Revised: 08/03/2021

DOCUMENT CONTRIBUTORS

Name	Project Role
Shridhar Shivaji Sarnobat	Solution Architect

DOCUMENT VERSION CONTROL

Version #	Revision Date	Revised By	Change Descriptions
0.1	09/10/2020	Shridhar Sarnobat	Draft created
1.0	03/12/2020	Shridhar Sarnobat	Review comments
1.1	19/01/2021	Shridhar Sarnobat	Class Diagram
2.0	08/03/2021	Shridhar Sarnobat	UI 2.0, Portfolio Performance, Benchmark score calculation
2.1	09/04/2021	Shridhar Sarnobat	Mathematical Calculation/DS Algorithms
2.2	14/6/2021	Shridhar Sarnobat	User Creation on RavenPack
2.3	15/6/2021	Shridhar Sarnobat	Centralized logging and Distributed Tracing

DOCUMENT REVIEW HISTORY

Version #	Date	Reviewed by	Comments
0.1	09/11/2020	Amit Saxena	

APPROVAL LIST

Name	Department	Role

Document Change Management and Distribution Procedures

Changes will be applied to this document according to the following procedure:

- ❖ Direct all change requests to the author of this document
- ❖ Each change request will be considered. If accepted, the change will be incorporated into a new draft of this document
- ❖ The new draft of this document will be circulated for review by appropriate project resources
- ❖ Approval of the new draft will be by concurrence of those individuals participating in the review
- ❖ Once concurrence is achieved, the draft becomes the latest version of this document, replacing any existing and previous versions
- ❖ New versions of this document are to be distributed to appropriate project resources or made accessible on-line for reference. Notification of a latest version will be communicated.

Deleted: pln

Table of Contents

1.	INTRODUCTION	6
2.	OBJECTIVE(S) AND/OR PURPOSE	6
3.	FEATURES	8
4.	SCOPE	8
4.1.	PHASE 1:	8
4.2.	OUT OF SCOPE FOR PHASE 1 AND PHASE 2:	8
5.	ASSUMPTIONS AND CONSTRAINTS	9
6.	DESIGN CONSIDERATIONS	9
6.1.	HIGH LEVEL ARCHITECTURE	9
6.2.	DOMAIN DRIVEN DESIGN	10
6.3.	PORTFOLIO HUB - COMMANDS/QUERIES	11
6.4.	TECHNICAL ARCHITECTURE DIAGRAM - CURRENT	12
6.5.	TECHNICAL ARCHITECTURE DIAGRAM – FUTURE	12
6.6.	DEPLOYMENT DIAGRAM	13
6.7.	MICROSERVICE ARCHITECTURE	16
6.8.	APPLICATION ARCHITECTURE DIAGRAM	17
6.9.	UI ARCHITECTURE	19
6.10.	ACTIVITY DESIGN	20
6.10. TECHNOLOGY CHOICES		21
7.	HAZELCAST IMPLEMENTATION	21
CACHE IMPLEMENTATION DETAILS:		21
LIST OF ESG BOOSTER APP DATA BEING CACHED IN HAZELCAST IMDG:.....		22
8.	CENTRALIZED LOGGING USING ELK	22
9.	DISTRIBUTED TRACING USING SLEUTH AND ZIPKIN SERVER	29

10.	INVESTABLE UNIVERSE	32
11.	PORTFOLIO DATA	33
12.	ESG SCORE CALCULATION AND INFLUENCE SCORE	35
13.	BENCHMARK SCORE CALCULATION	37
14.	INVESTSUITE	40
15.	ESG SCORE NORMALIZATION LOGIC AND CALCULATION	45
16.	ESG FORECASTING API	45
17.	RAVENPACK - NEWS SENTIMENTAL ANALYSIS	50
18.	MATHEMATICAL CALCULATION/DS ALGORITHMS	52
19.	REQUIREMENT VS. END POINTS - TRACEABILITY MATRIX	52
20.	END POINT DEFINITIONS	52
21.	TEST STRATEGY	53
17.1	SCOPE OF TESTING	53
17.2	ASSUMPTIONS	53
18.	INTEGRATION STRATEGY	54
18.1 INTEGRATION WITH PROPRIETARY SYSTEMS.....		54
18.2	INTEGRATION WITH THIRD PARTY SYSTEMS	54
19.	RISKS	54
20.	APPLICATION SETUP	54
20.1	PREREQUISITE	54
20.2 OBTAIN SOURCE CODE		54
20.3 OPEN APP IN INTELLIJ-IDEA		54
21.	RUNNING APP	54
22. CI/CD SETUP.....		54
22.1	CI /CD GITLAB SCRIPTS	55
23. CODE REVIEW GUIDELINES		55

24. HARDWARE REQUIREMENTS	55
25. SOFTWARE REQUIREMENTS	55
26. DATA INGESTION STEPS	56

1. Introduction

- The accelerator will demonstrate the functionality of showing the ESG score at portfolio
- It will perform Portfolio Analysis against different ESG metrics and benchmarking using different ESG scores and index
- It will also improve the portfolio ESG score by proposing an “alternative portfolio” with minimal changes and considering client preferences
- Finally, it will provide support for extracting report overviews of the recommendation/investment actions for further discussion or approval
- This document describes the Scope, Architecture Design and Microservices End-point definitions for the ESG Application. Also note that this document is applicable to development of ESG accelerator to be carried out by Synechron.

2. Objective(s) and/or Purpose

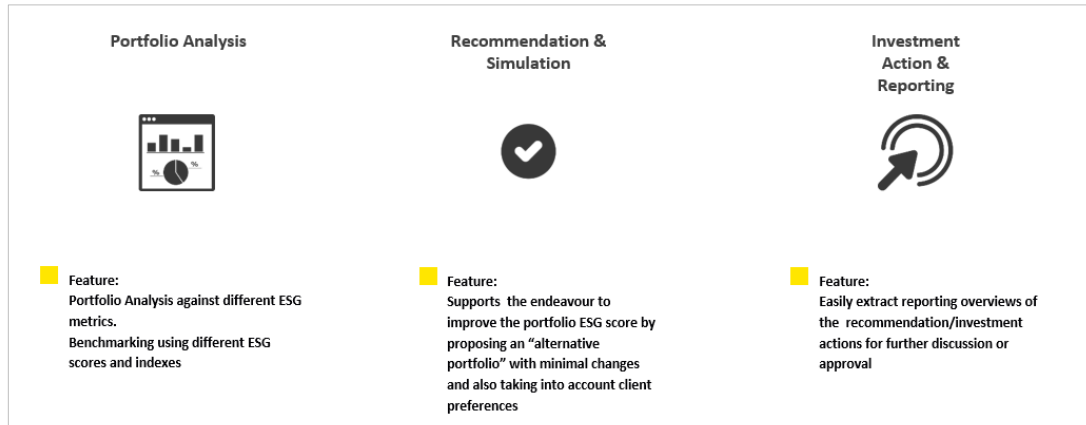
The accelerator has following objectives:

- Institutional firms have started to embrace ESG but are struggling with moving targets and lack of standards
- No uniformity and aligned definitions in terms of ESG data from the legislator
- Lots of data sources available but how to make sense of this data?
- No mainstream platform which provides the functionality of showing the ESG score at portfolio level and/or benchmarking of ESG data.

The solution will also demonstrate the following Microservices only benefits:

- Reduce time to market
- Increase reusability and flexibility
- Loose coupling and asynchronous communication between services.

3. Features



4. Scope

4.1. Phase 1:

- All Services like Portfolio, Forecasting, ESG calculation, News Sentiment, Investable universe will be deployed on AWS cloud, but no AWS cloud specific service will be used
- Following all business functionality will be delivered



- SSO integration
- Hazelcast cache

4.2. Phase 2:

- UI 2.0
- Implement distributed tracing
- API Management.

Commented [AS1]: Let's name the services in scope instead of generic word All.

Commented [SS2]: Done

Commented [AS3]: We can provide JIRA item list as business features.

Commented [SS4]: File attached

Deleted: A

Deleted: -

Deleted:

Formatted: Font colour: Black

Formatted: Indent: Left: 1.9 cm, No bullets or numbering

Commented [SS5]: @Sanup Kumar - Can you please add JIRA items here

Deleted: will be delivered.

Commented [AS6]: We can keep Phase1, Phase 2 & Not in scope items.

Commented [SS7]: Done

Deleted: Out of Scope for Phase 1 and

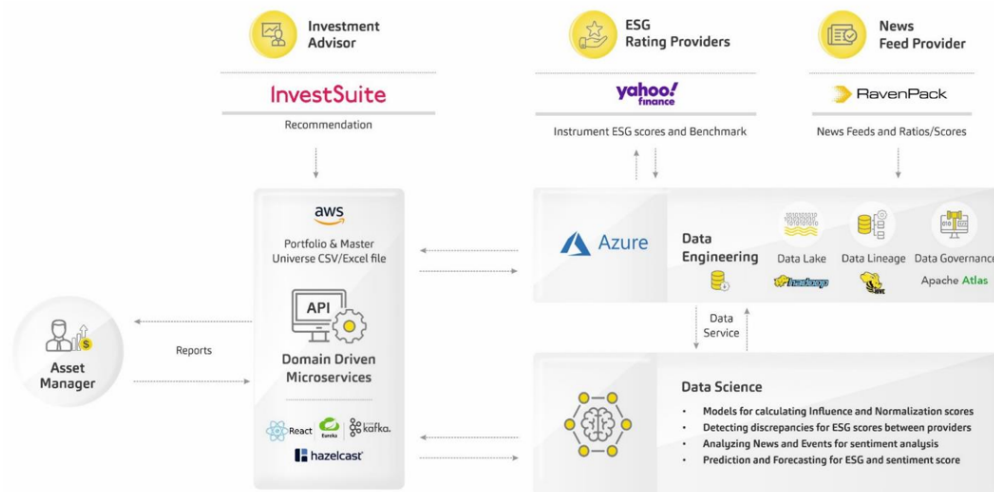
5. Assumptions and Constraints

TODO

6. Design Considerations

6.1. High Level Architecture

High level view of a system with different internal and external components that interact with each other.

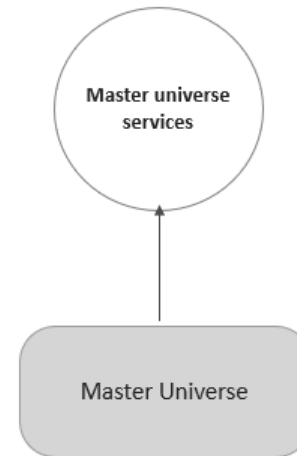
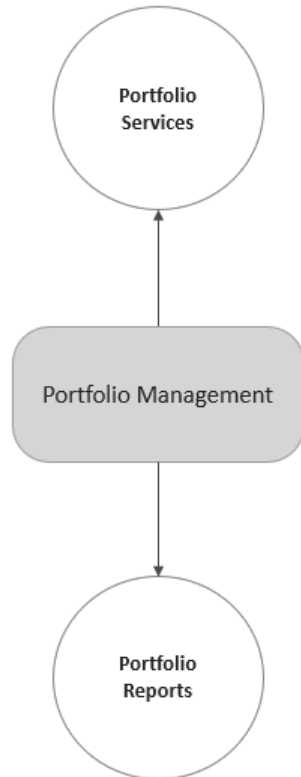


Commented [AS8]: We should remove Sustainability & Refinitiv

Commented [SS9]: Done

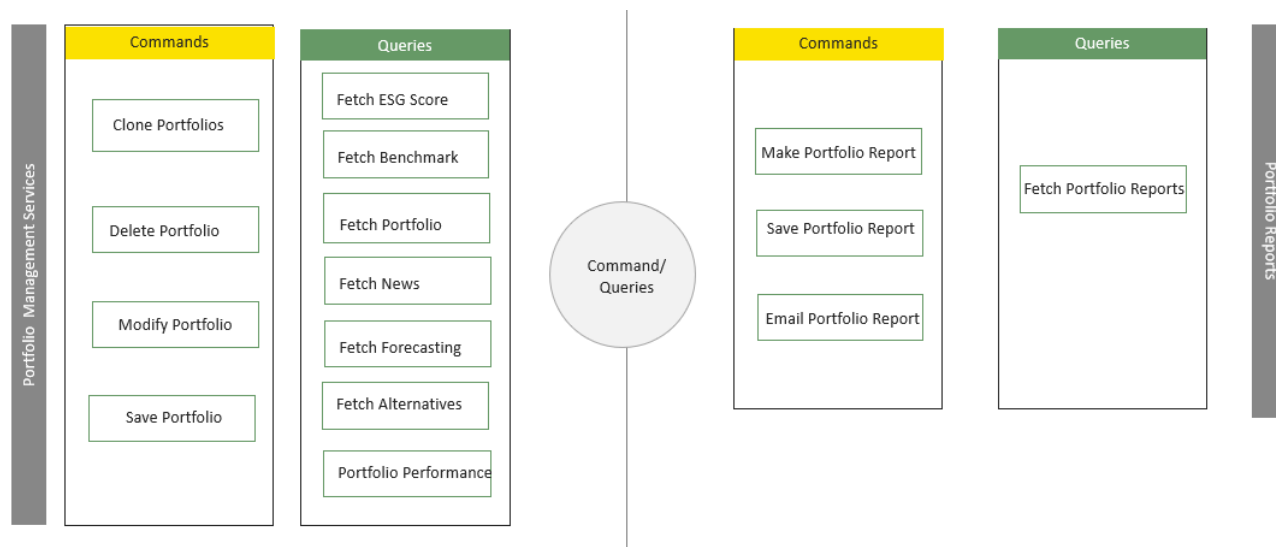
6.2. Domain Driven Design

Sub Domains & Bounded Context



6.3. Portfolio Hub - Commands/Queries

Commands and Queries represent any kind of operations within the bounded context.



Commented [AS10]: We should work on next level of this diagram to be more elaborative on Domain's command & query aspects. Many features are still missing here.

Commented [AS11]: In my view Create Portfolio & Save Portfolio are same. We do have features like Clone & delete, which can be included in Portfolio commands

Commented [SS12]: Done

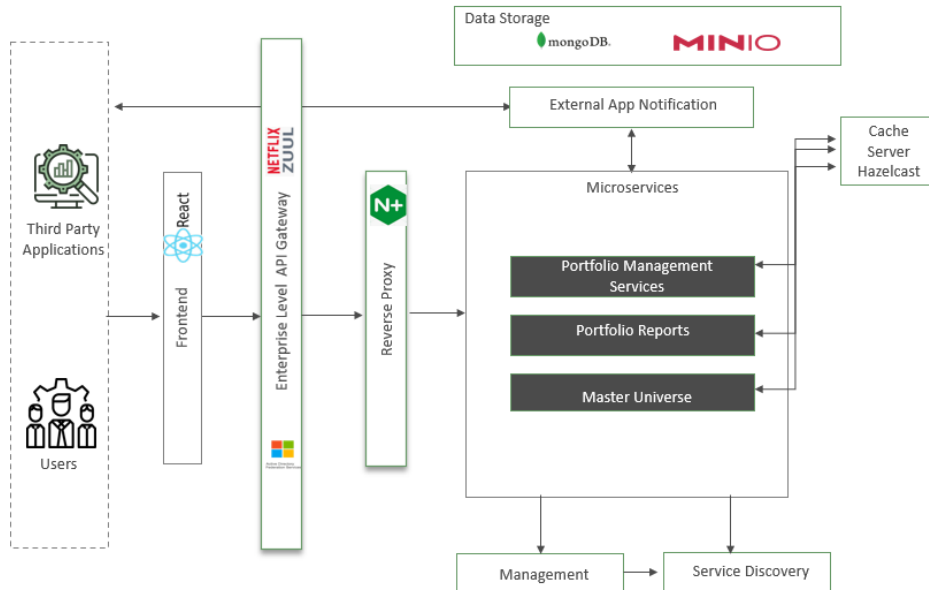
Commented [SS13]: Lets discuss

Commented [AS14]: Portfolio Queries : We fetch data about Benchmarks, NEWS, Forecasting & Alternates.

Commented [SS15]: Done

6.4. Technical Architecture diagram - Current

All bounded context becomes Service Family & Inside that there are microservices built in.



Overview

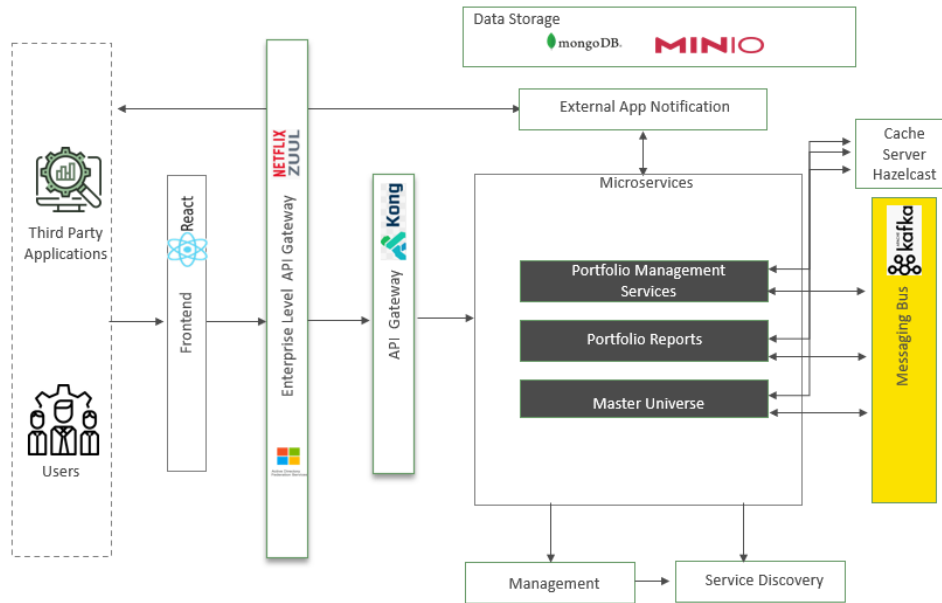
- **Enterprise Level API Gateway** – A single-entry point at enterprise level which is responsible for SSO with the help of ADFS.
- **Reverse Proxy**– It will act as Application-level Gateway which acts as an entry point for all backend services.
- **Management** – Balances services on nodes and identify failures.
- **Service Discovery** – A guide to find the route of communication between microservices.
- **Microservice** – Identified by domain and submain.
- **External App Notification** – This will be event-based system that will notify the data engineering component of any changes that have been made.
- **Cache Server Hazelcast** – To manage distributed cache.

Commented [AS16]: Deployment diagram needs to be added.

Commented [SS17]: Done

6.5. Technical Architecture diagram – Future

Commented [AS18]: Deployment diagram needs to be added.



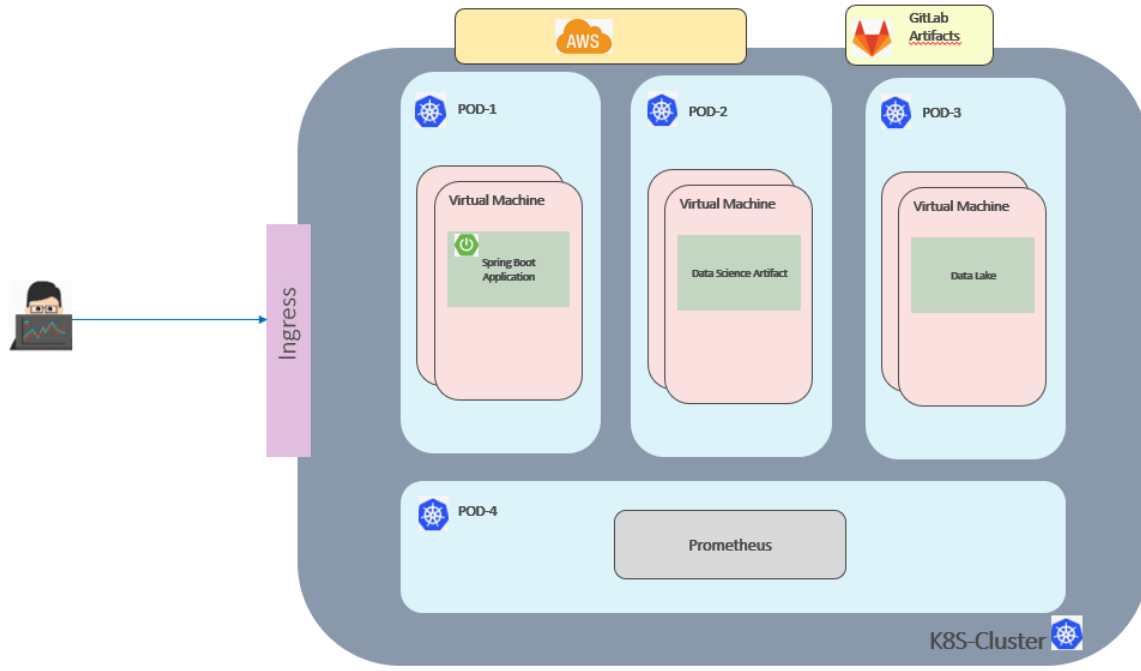
Overview

- **Enterprise Level API Gateway** – A single-entry point at enterprise level which is responsible for SSO with the help of ADFS.
- **API Management** – Application-level Gateway which acts as management tool for all backend services.
- **Management** – Balances services on nodes and identify failures.
- **Service Discovery** – A guide to find the route of communication between microservices.
- **Microservice** – Identified by domain and submain.
- **Messaging Bus** – A publisher subscriber implementation for propagating events between the microservices.
- **External App Notification** – This will be event-based system that will notify the data engineering component of any changes that have been made.
- **Cache Server Hazelcast** – To manage distributed cache.

6.6. Deployment diagram

Commented [AS19]: Deployment diagram needs to be added.

Commented [SS20]: Done



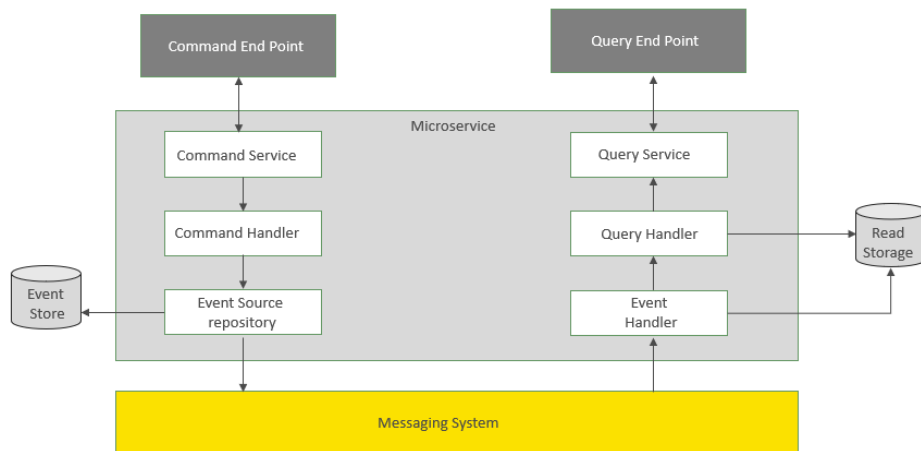
Description

- **Enterprise API Gateway**
A single-entry point at enterprise level which is responsible for SSO with the help of ADFS.
- **Application API Gateway**
Application-level Gateway which acts as an entry point for all backend services.
- **Management**
Balances services on nodes and identify failures.
- **Service Discovery**
A guide to find the route of communication between microservices.
- **Microservice**
Identified by domain and submain.
- **Messaging Bus**
A publisher subscriber implementation for propagating events between the microservices.
- **External App Notification**
This will be event-based system that will notify the data engineering component of any changes that have been made.
- **Cache server**
To manage distributed cache.

6.7. Microservice Architecture

DDD with event sourcing and CQRS

We will use two patterns:



Event Sourcing

CQRS

- Event Sourcing – Everything should be traceable (Ex. Changes in configuration) If you are modifying portfolio for better ESG percentage
- What we captured is final state, but it helps to find out detail steps
- At any given point of time. Whenever there is change, that change will be recorded as an event and not as data
- **Axon framework** will be used for implementing CQRS and Event store.

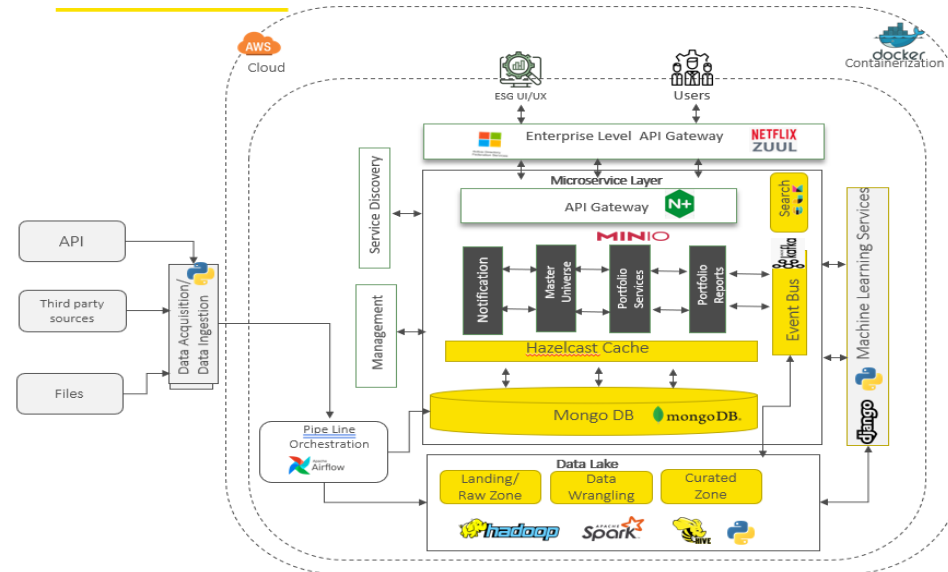
Commented [AS21]: Instead of Q Lets add relevant information. i.e how CQRS help in know detail steps.

Commented [SS22]: Done

Deleted: what if you want to know

Deleted: ?

6.8. Application Architecture Diagram



Description

- **UI**
 - The User Interface is a web application built with HTML 5 and CSS3
 - This is a Single Page Application (SPA) built using React, Redux and JSX
 - The user accesses the application using any popular web browser like chrome, IE, Mozilla Firefox and safari etc.
 - We have used popular libraries like **Highcharts**, **Moment**, **Lodash** etc.
 - This web application is responsive for all desktop monitors and tablet landscape version also.
 - Used **PDFtoJS** and **HTMltoPNG** library for exporting report from frontend

- We have used Synechron single sign on login authentication
- **Micro Services Layer**
 - The core application service is hosted on the server and accessed using Restful Web Service API.
 - Investable Universe service is responsible for fetching the master universe company list from Data lake.
 - Portfolio services are related to action taken against Portfolio like creating, modifying, saving, and fetching portfolio, similarly it will contain service for calculating ESG score against instruments from portfolio.
 - Portfolio reports are related to action taken against reports like generate, save, fetch, report.
- **Machine learning service**
 - **ESG Score Forecasting**
 - In house built econometric model for ESG score forecasting, that considers cross-sectional and time effects. Based on ESG historical data, yields prediction of the ESG score for the following year.
 - Input: ESG historical data
 - Output: ESG score predictions
- **Event Bus-Apache Kafka**
 - A publisher subscriber implementation for propagating events between the microservices.
 - Apache Kafka is highly available and fault tolerant.
 - It provides loose coupling through asynchronous communication.
- **Hazelcast Cache**
 - Distributed cache will be used where Cache is stored outside the service instance but accessible to all the service instances of given service
 - Consumer application services will start seeing the performance gain from the time service is launched
 - New service launched as part of auto scale will start having preloaded cache, hence it's good for micro-services kind of environment
 - Write-through mechanism is implemented where data will be sync up with persistent storage like MongoDB
 - We have written a map-store for each collection from MongoDB. Every map-store takes care of storing the record from Hazelcast cache to MongoDB collection
 - Locking is implemented to handle concurrency.
- **MongoDB database**
 - Database per service is implemented by created separate collection inside MongoDB schema
 - Temporary data of Portfolio configurations and portfolio reports will be stored in this.

Commented [AS23]: Master Universe has very low importance . Also we don't have such feature of uploading MU from app. We should rephrase this sentence.

Commented [SS24]: Lets discuss

Commented [SS25]: Done

Commented [AS26]: We are envisioning only generate pdf report. Email & save is out of scope in PH2 as well.

Commented [SS27]: Email is ok to omit, but then how can we show already generated reports without save functionality?

Deleted: make

Deleted: and email

Commented [SS29]: Removed

Deleted: <#>In house data science model for custom build recommendation engine. It will be developed in Python....

Deleted: <#>

Commented [AS30]: We should highlight how effectively are we utilizing distributed cache across services.

Commented [SS31]: @Amit Saxena1 it's already updated in below section.

- **Data Lake**
 - All the raw data will be stored in Hadoop Data Lake; any data intensive computations will be performed on Hadoop
 - Hive Tables will be created for all the datasets created on Hadoop so that application team, Developers, Business users and Data scientists can write SQL queries to explore the data using Hue
 - All data intensive pre-processing required for ML models will be performed on Hadoop before being consumed
 - Historical and System logs & Unstructured data can be moved to Hadoop.
- **Service Discovery and Management**
 - A guide to find the route of communication between microservices
 - Netflix Eureka will be used for Discovery.
- **External API gateway**
 - A single-entry point at enterprise level which is responsible for SSO with the help of ADFS
 - Enterprise gateway is implemented using Spring Zuul.
- **API management**
 - Application-level Gateway which acts as an entry point for all backend services
 - It is implemented using Nginx.

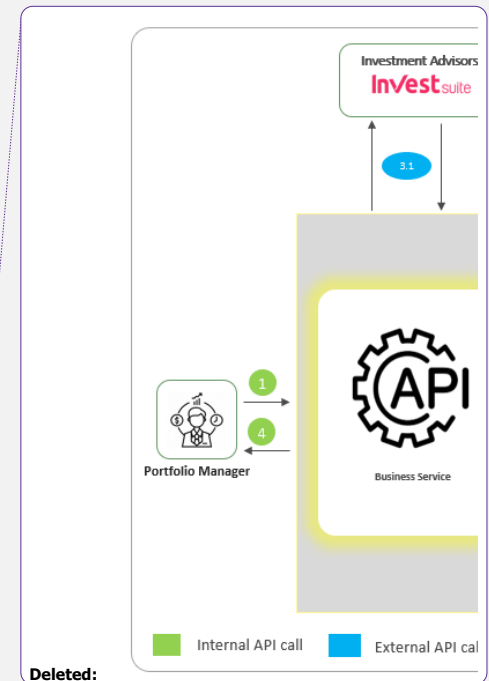
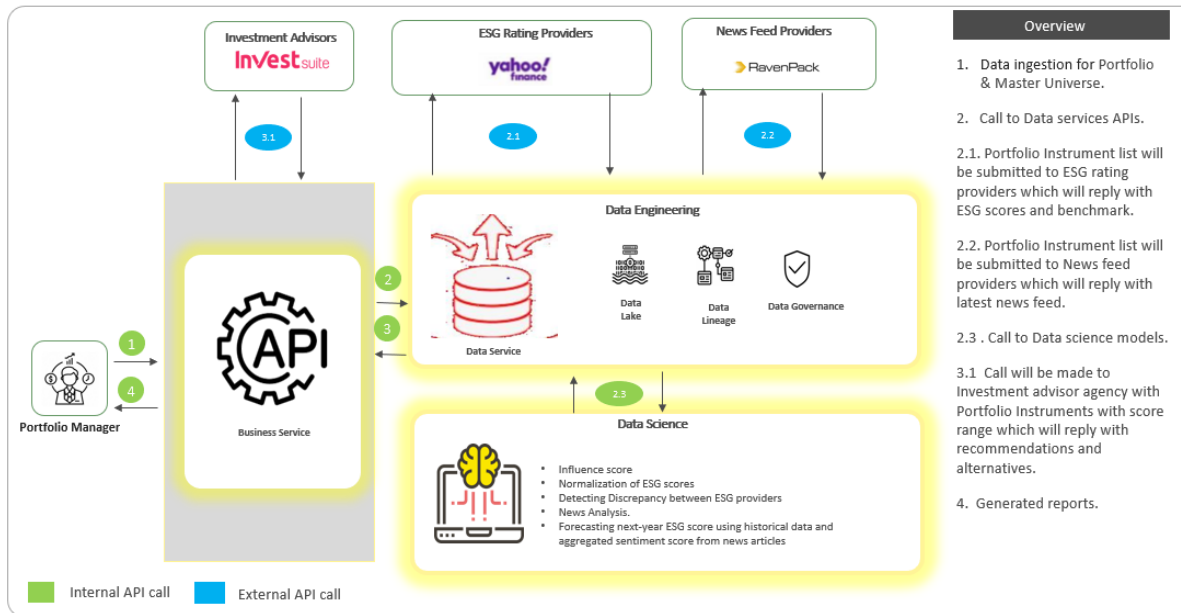
Deleted: Application level

6.9. UI Architecture

Description

The UI will be built on React and Redux with HTML 5, CSS3 and JSX. The UI will be designed as a Single Page Application (SPA) and uses less server traffic to fetch pages and data, which improves the response time and user experience. Re-useable parts of the UI are identified and made as component. These are then tailored to make the whole app. Also uses several Redux services (Actions, Reducers, Dispatcher) and custom-made services for fetching data and navigating pages.

6.10. Activity Design



6.10. Technology Choices



7. Hazelcast implementation

Cache Implementation Details:

- Hazelcast Version: 3.12.7
- Cache Access Pattern: Write Through
- Topology Used: Client Server.
- Eviction policy: LRU (Least Recently Used)
- Locking Mechanisms used for Updates: Key based Locking.

Cache Implementation Sample:

```
Config.addMapConfig(new MapConfig().setName("portfolios") //represents name of cache
```

Deleted:

```
.setMaxSizeConfig(new MaxSizeConfig(300, MaxSizeConfig.MaxSizePolicy.FREE_HEAP_SIZE)) // memory config
.setEvictionPolicy(EvictionPolicy.LRU) // Eviction Policy is Least Recently used
.setBackupCount(2) //Number of synchronous backups.
.setTimeToLiveSeconds(1800) // time after which the entry will be removed from cache
.setMapStoreConfig(portfolioMapStoreConfig))
// Mapstore Config that will synchronize data between cache and table/Collection in DB as we are using write through approach
```

List of ESG Booster App Data being cached in Hazelcast IMDG:

- Investable universe – Master universe is cached.
- Portfolio Configuration – Criteria met Investable universe is cached
- Portfolio Report – Portfolio specific reports are cached
- InvestSuite -To reduce frequency of calls to InvestSuite API for one-to-one alternative for instrument from portfolio
- Benchmark – Portfolio benchmark information is cached.
- Portfolio - Portfolio Performance and Companies response

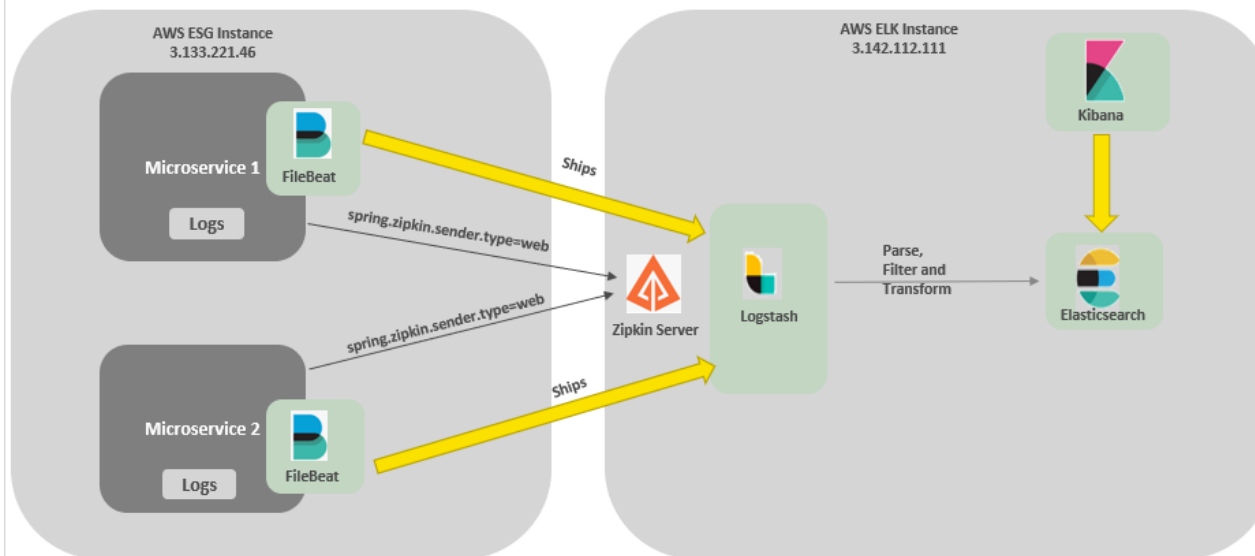
Commented [AS32]: We should add little more technical details
.i.e. evacuation policy of each line item.

Commented [SS33]: @Sachin Poojari - Please add technical
implemented in brief.

Commented [AP34]: Added new in 2.0

8. Centralized logging using ELK

Centralized logging is implemented using ELK stack and File Beat



- Elastic Search: - To store logs in specific index for specific application, this index will be per service per environment to have logs populated for all env differently.
- Logstash: - Logs will be inserted in Elastic Search via Logstash after filtering. Logstash will receive logs from Filebeat.
- Kibana: - Kibana will be used to display logs on dashboard.
- File beat - Filebeat will be used to forward logs to Logstash from individual service.

Following are the steps to configure ELK and File Beat

1. Modify docker file for each service to install File Beat component in it.

Ex. Alternative service docker file

File – Docker file

FROM openjdk:11

RUN apt-get update

```
RUN curl -L -O https://artifacts.elastic.co/downloads/beats/filebeat/filebeat-7.12.0-amd64.deb
RUN dpkg -i filebeat-7.12.0-amd64.deb
COPY filebeat.yml /etc/filebeat/filebeat.yml
ARG JAR_FILE=target/Alternative-0.0.1-SNAPSHOT.jar
ADD $JAR_FILE alternative-service.jar
COPY start.sh start.sh
RUN chmod +x start.sh
ENTRYPOINT ["/start.sh"]
#ENTRYPOINT ["java","-Dspring.profiles.active=${activatedProfile}","-jar","/alternative-service.jar"]
EXPOSE 8096
```

2. Start File Beat

This will be invoked when container is starting up. This file will get called from docker file.

Note that file beat is started before actual service is up, so that it will be ready to receive any logs as soon service starts up.

File - start.sh

```
#!/bin/bash
```

```
#service filebeat start
```

```
cd /etc/filebeat
```

```
filebeat -c /etc/filebeat/filebeat.yml &
```

```
cd ..
```

```
cd ..
```

```
java -Dspring.profiles.active=${activatedProfile} -jar alternative-service.jar
```

3. File - filebeat.yml

```
filebeat.inputs:
```

```
- type: log
```

```
  enabled: true
```

```
paths:
```

```
  - /logs/alternative.log
```



```
output.logstash:  
  # The Logstash hosts  
  hosts: ["3.142.112.111:5044"]
```

As stated in above file, input path points to service log file location (from where file beat will fetch logs) and destination path points to Logstash location (to where this service logs will be transferred to for centralize logging)

4. File – logstash.conf

Once logs are collected in Logstash it will get forwarded to Elastic search which applies indexing to it.

```
input {  
  beats {  
    port => 5044  
    host => "0.0.0.0"  
  }  
}  
  
# Specify an Elasticsearch instance  
output {  
  elasticsearch {  
    hosts => ["localhost:9200"]  
  
    #index => "nginx-logs-dev"  
    #index => "%{[@metadata][target_index]}"  
    index => "%{[service]}-%{+YYYY.MM}"  
  }  
}
```

5. File – kibana.yml

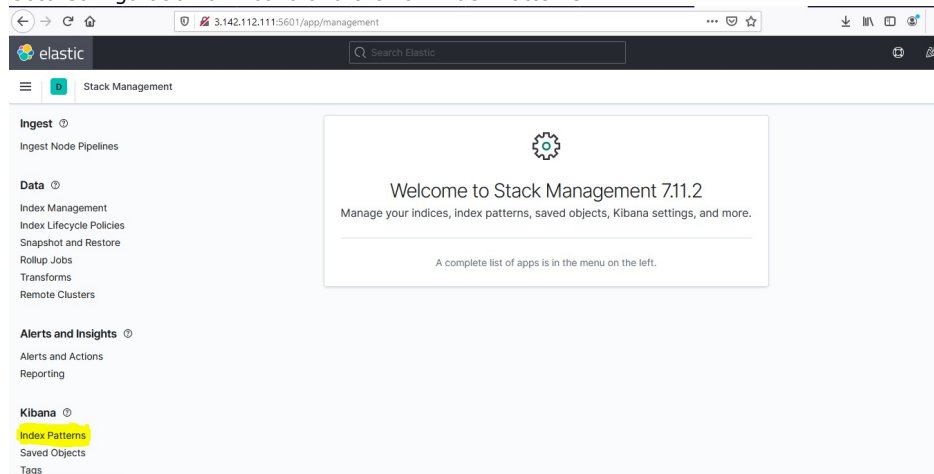
It will specify Kibana host details.

6. Kibana dashboard will be used for checking logs.

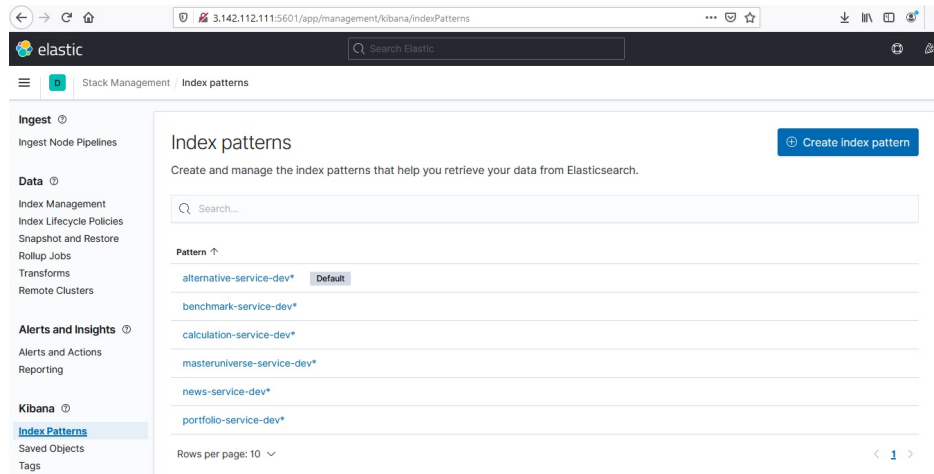
<https://3.142.112.111:5601>

To setup initial index in Kibana

A. Goto Configuration of Kibana and click on Index Patterns




B. Click on Create Index Patterns





The screenshot shows the Elastic Kibana interface for managing Index Patterns. The browser address bar indicates the URL is 3.142.112.111:5601/app/management/kibana/indexPatterns. The Elastic logo and a search bar are at the top. The left sidebar contains navigation links for Ingest, Data, Alerts and Insights, and Kibana. The main content area is titled 'Index patterns' and includes a 'Create index pattern' button. Below the title is a search bar and a table of existing patterns. The table lists five patterns: 'alternative-service-dev*', 'benchmark-service-dev*', 'calculation-service-dev*', 'masteruniverse-service-dev*', and 'news-service-dev*', with 'alternative-service-dev*' marked as 'Default'. The 'portfolio-service-dev*' pattern is also listed but not visible in the current view. The bottom of the page shows 'Rows per page: 10' and a pagination control for page 1.


elastic Search Elastic

Stack Management Index patterns

Ingest 
Ingest Node Pipelines


Data 
Index Management
Index Lifecycle Policies
Snapshot and Restore
Rollup Jobs
Transforms
Remote Clusters

Alerts and Insights 
Alerts and Actions
Reporting


Kibana 
Index Patterns
Saved Objects
Tags

Index patterns


Create and manage the index patterns that help you retrieve your data from Elasticsearch.

 Create index pattern

Search...

Pattern 

alternative-service-dev*	Default
benchmark-service-dev*	
calculation-service-dev*	
masteruniverse-service-dev*	
news-service-dev*	
portfolio-service-dev*	

Rows per page: 10  < 1 >

c. Give index patterns name like servicename-env ex. portfolio-service-dev

← → ↻ 🏠 3.142.112.111:5601/app/management/kibana/indexPatterns/create

elastic Search Elastic

Stack Management / Index patterns / Create index pattern

Index Lifecycle Policies
Snapshot and Restore
Rollup Jobs
Transforms
Remote Clusters

Alerts and Insights ⓘ
Alerts and Actions
Reporting

Kibana ⓘ
[Index Patterns](#)
Saved Objects
Tags
Spaces
Advanced Settings

Stack ⓘ
License Management
8.0 Upgrade Assistant

Step 1 of 2: Define an index pattern

Index pattern name

port*

Next step >

Use an asterisk (*) to match multiple indices. Spaces and the characters \, /, ?, *, <, >, [are not allowed.

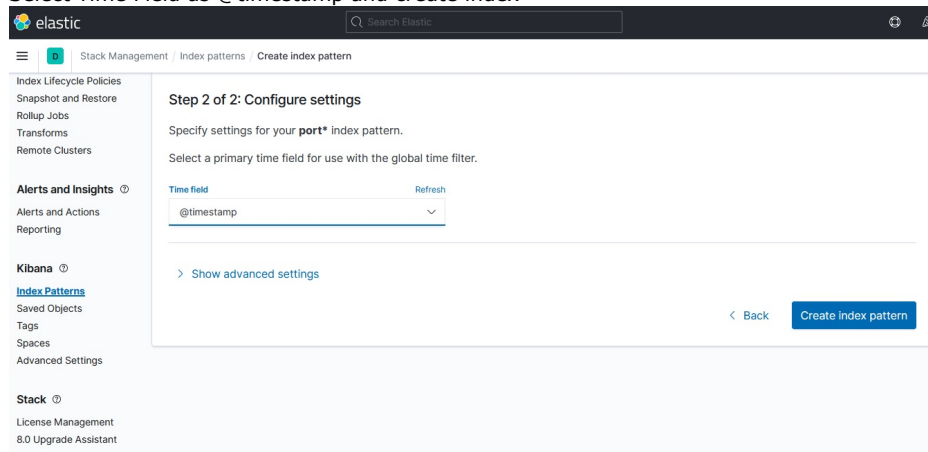
☐ Include system and hidden indices

✓ Your index pattern matches 1 source.

portfolio-service-dev-2021.05	Index
-------------------------------	-------

Rows per page: 10 ▾

D. Select Time Field as @timestamp and create index



elastic Search Elastic

Stack Management / Index patterns / Create index pattern

Index Lifecycle Policies
Snapshot and Restore
Rollup Jobs
Transforms
Remote Clusters

Alerts and Insights ⓘ
Alerts and Actions
Reporting

Kibana ⓘ
Index Patterns
Saved Objects
Tags
Spaces
Advanced Settings

Stack ⓘ
License Management
8.0 Upgrade Assistant

Step 2 of 2: Configure settings

Specify settings for your **port*** index pattern.

Select a primary time field for use with the global time filter.

Time field Refresh
@timestamp

> Show advanced settings

< Back Create index pattern



Logging.zip

9. Distributed Tracing using Sleuth and Zipkin Server

1. Add dependency spring-cloud-sleuth-zipkin in pom.xml

```
<dependency>
  <groupId>org.springframework.cloud</groupId>
  <artifactId>spring-cloud-starter-sleuth</artifactId>
</dependency>
<dependency>
  <groupId>org.springframework.cloud</groupId>
  <artifactId>spring-cloud-sleuth-zipkin</artifactId>
</dependency>
```

2. Modify application.properties file

```
#zipkin server location  
spring.zipkin.base-url=http://3.142.112.111:9411
```

```
spring.zipkin.enabled=true  
spring.zipkin.service.name={accelerator-name}-{service-name}-{profile}
```

```
#How info will be sent over to Zipkin – HTTP or Rabbit or Kafka  
spring.zipkin.sender.type=web
```

```
#How much info needs to be sent over to Zipkin Ex. For 10% its 0.1 and for 100% its 1. And with this trace id and span  
id will be attached to logging info that application produces.  
spring.sleuth.sampler.probability=1
```

3. Add logging in Java class for microservice A

```
Logger logger = LoggerFactory.getLogger(this.getClass());
```

```
logger.info("Before calling microservice B");
```

```
//Call to another microservice B  
logger.info("After calling microservice B");
```

4. Download Zipkin server

Option 1: Docker

```
docker run -d -p 9411:9411 openzipkin/zipkin
```

Option 2: Java

```
#Download  
curl -sSL https://zipkin.apache.org/quickstart.sh | bash -s  
or  
curl -sSL https://zipkin.io/quickstart.sh | bash -s
```

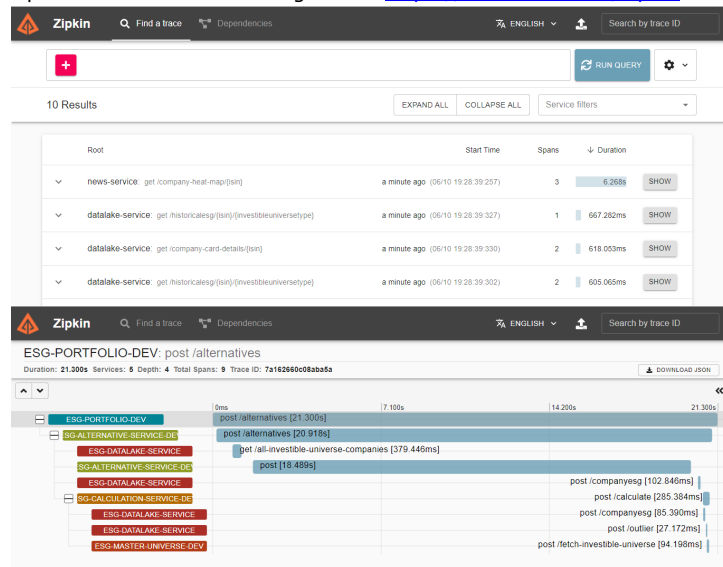
Run executable jar file
`java -jar zipkin.jar`

Ex.

Run Zipkin in background and pass some environment variables which specify Elasticsearch as the storage type (assumes a locally running ELK Stack/Elastic search)

`sudo nohup java -DSTORAGE_TYPE=elasticsearch -DES_HOSTS=http://127.0.0.1:9200 - DES_INDEX=zipkin-dev -jar zipkin.jar &`

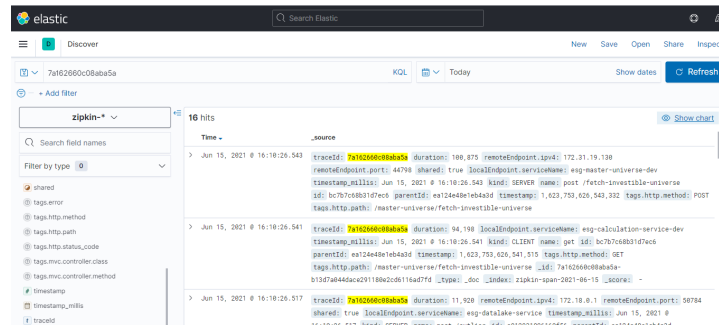
5. Open dashboard for checking Traces <https://localhost:9411/zipkin>



Trace Id: **7a162660c08aba5a**

Now the same Zipkin logs can also be retrieved using Kibana dashboard using above traceId.
Create Index Pattern: Home -> Manage -> Index Patterns under Kibana -> Create Index pattern

Access Kibana using url: <http://3.142.112.111:5601>



10. Investable Universe

- Investable universe is the master data and contains list of all the companies. Portfolio manager can invest in these instruments. PFA schema.
- Investable universe can be set using the "Investable Universe" tab in the ESG Booster app.
- Portfolio Manager can use various filters to produce a strong Investable universe which will be saved against the portfolio and would contribute to finding best alternatives for low scoring companies in the portfolio.
- Investable universe data is stored in data lake and data lake API's are invoked by services which returns all Investable universe companies in form of objects as per the filter which was saved by the Portfolio manager using the "Investable Universe" tab.
- This list of Investable Objects contains ISIN, CompanyName, ESG Score and E, S, G Pillar Scores for each of the scoring methodology.
- These objects are stored into the Hazelcast cache for fast retrieval.

Once user uploads a portfolio, he/she must set the investable universe as per his/her requirement.

There are following parameters on which user can set the filters:

- ESG Score
- Environmental Score
- Social Score
- Government Score
- Countries (This is list of countries from which user would like to get the recommendations.)

Business Service API's:

We have two API's here namely:

1. GET -> /investibleUniverse/{portfolioId}

This API hits the data lake and gets the total investable universe companies and returns as response.

Before sending the response, we are caching the result in hazelcast, so that next time the result is checked in the cache and if cache is empty then only, we make a call to the data lake.

2. POST -> /investibleUniverse/{portfolioId} + Request Body

The input to this API is portfolioId and Request body which consists of the filter values for the above parameters.

Instead of storing these large number of companies, we are storing only the filters.

Data Lake API's:

GET API: /all-investible-universe-companies

this returns the whole investable universe from the data lake.

This response is the one in which the portfolio specific filters are applied, and the portfolio specific investable universe is generated.



investable_universe_
Schema.csv

- Application will have 2 default portfolios uploaded by default from Synthetic data and [Yahoo Finance](#) respectively.
- Investable universe consisting of 455 companies are shared with InvestSuite for them to provide alternative from this list.
PFB sheet



Investable_Universe_
549.xlsx

11. Portfolio Data

[MinIO which is used for storing Portfolio data files which is hosted on AWS cloud.](#)

▲ Please follow below guidelines for uploading portfolio file:

Deleted: Sustainalytics

Deleted: 549

Deleted: s

Commented [AS35]: Below information is purely functional. We should add technical details about how are we managing / mapping these Objects. i.e any technical implementation.

Commented [SS36]: @Sachin Poojari - Please add relevant technical details

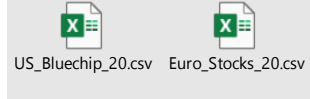
Commented [SS37]: Done

Formatted: Normal, Space Before: 0 pt, No bullets or numbering

Formatted: Font: Calibri, Font colour: Auto

- Below columns are mandatory columns in the portfolio upload csv file whether there is value present for them or not.
Columns: ISIN, Ticker, Company, Weightage, Amount Invested
Note: Column names should be exact matches (They are case-sensitive), while the order of the columns can change.
- ISIN is a mandatory field, if it is empty or not specified, application will throw error.
Error: "Uploaded file contains incorrect company data records."
- Weightage and Amount Invested, either of them should be present. If both are empty or 0, application will throw error.
Error: "Uploaded file contains incorrect company data records."
Note: Step 2 and 3 having same errors as these will be checked for each company and after parsing through all the companies in the file, application will throw this error.
- Company Name is an optional field, if it is empty or given name is incorrect, the correct name will be fetched from the data lake and populated in the application.
- If in portfolio uploaded, for some company weightages are given and for some amount invested is given, application will throw error.
Error: "Multiple components are missing in the file. Kindly re-check Weightage / Amount Invested again and re-upload."
- If in portfolio uploaded, weightages for some companies are not given but amount invested for all companies is given, application will calculate weightages on basis of amount invested.
- If in portfolio uploaded, weightages for all companies are given and for some company amount invested are not given, application will consider the weightages.
- If in portfolio uploaded, weightages and amount invested for all companies is given, application will consider the weightages.
- If in portfolio uploaded, sum of weightages does not satisfy below condition, application will throw error.
Condition: total Weightage = 100.00 OR (total Weightage >= 99.51 AND total Weightage <= 100.00)
Error: "Total weightage of all the companies from the uploaded portfolio must be equal to 100%."
- While fetching ESG data from data lake for the companies, first we will match by ISIN and get the data. If we have few companies whose ISIN does not match we will query the data lake by matching company name. Still if there are companies whose ISIN and company name does not match, these company's weightages will be normalized among the rest of the companies and portfolio will get uploaded.
- Weightage should be given as only digits without percentage sign, as it is by default considered weightage to be in percentage.

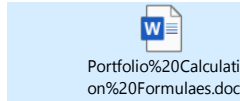
PFA sample portfolio csv files



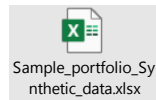
12. ESG score calculation and influence score

The Tool allows user to select the data source i.e., Synthetic data & [Yahoo Finance](#) and accordingly the Portfolio data should reflect the specific ESG scoring Methodology.

- Influence score - measure the influence of companies and their underlying factors to overall portfolio scores
 - decomposition of weighted average formula
 - influence score has been calculated based on holding % and ESG score
- PFB ESG portfolio calculation formula sheet.



- Sample portfolio ESG and influence calculation for Synthetic data.



- Sample portfolio ESG and influence calculation for [Yahoo Finance](#)

Deleted: Sustainalytics

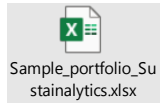
Commented [AS38]: These are functional requirement . We should add technical details APIs which has been built to serve these data.

Commented [SS39]: @Sachin Poojari to add required details

Commented [SS40]: Sachin is currently working on making it as API endpoint.

Commented [SS41]: Done

Deleted: Sustainalytics



- Once the given portfolio is validated the portfolio is sent to the Calculation Service layer which first retrieves all the REFINTIV and Yahoo Finance ESG scores and E,S,G pillar scores along with outlier score for all the companies in the portfolio and then each of the Scoring methodology is applied over these scores to calculate the overall Portfolio ESG Score and the E,S,G Pillar score and also the influence score for each of the company in the portfolio which shows how each of them contribute to the overall portfolio ESG Scores .
- Also using the outlier score the outlier companies for the given companies are calculated

Business Service API's:

POST API: portfolio-mgt/upload-portfolio

this posts the portfolio file to the ESG Booster API's where the validation of the portfolio happens and once its validated s sent to the calculation layer for calculating the ESG Scores, influences and the outliers.

Data Lake API's:

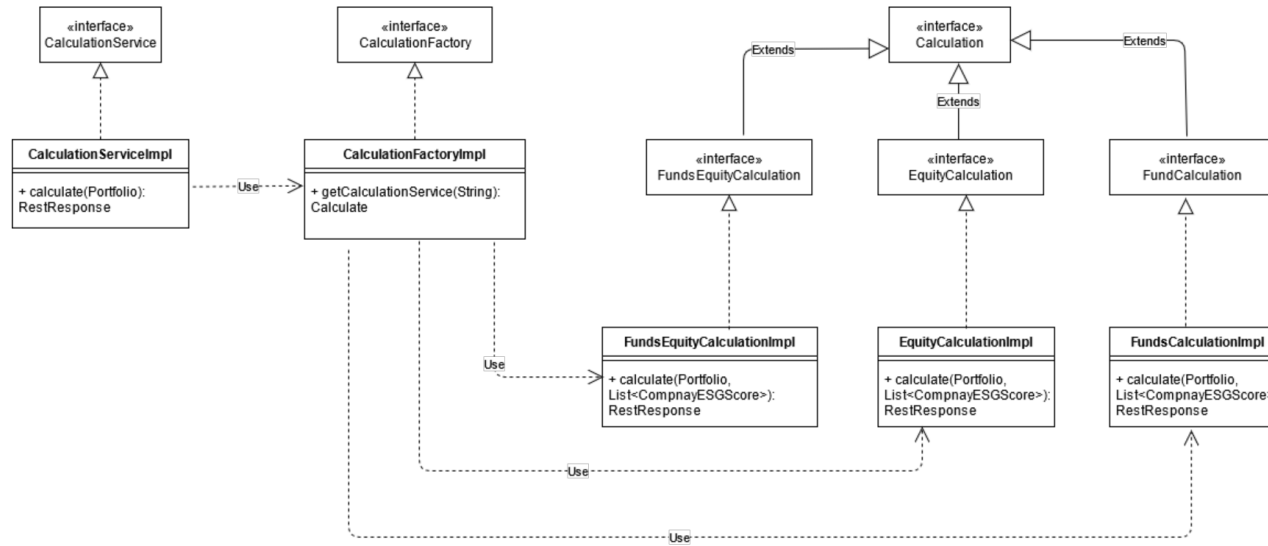
GET API: /datalake/companyesg(sending the List of ISIN's as the request params)

this returns the list of Company objects that contains ISIN, name details as well as the ESG Scores for Synthetic data and the Yahoo Finance and the outlier scores which are the pre-requisites for the scoring methodology and getting the outliers

GET API: /datalake/companyEsgByName (sending the List of Company Names as the request params)

If the above API does not return result for some of the ISIN, this API returns the list of Company objects that contains ISIN, name details as well as the ESG Scores for Synthetic data and the Yahoo Finance and the outlier scores which are the pre-requisites for the scoring methodology and getting the outliers by matching the name of company with that in data lake.

Class diagram:



13. Benchmark Score Calculation

Benchmark is calculated at the data engineering side. Relative Calculation for Portfolio with Each Benchmark is done to represent how portfolio is performing against each Benchmark Index in E,S,G and ESG.

Formula $((\text{Portfolio Score} - \text{Benchmark Score}) / \text{Benchmark Score}) * 100$.

Business Service API: Get call </portfolio-mgmt/benchmark-data/{portfolio-id}>

Response Body:

```

{
  "portfolioId": "1",
  "portfolioName": "Diversified Portfolio",
  "totalESG": 56.55,
}

```

Commented [AP42]: Added new Relative Benchmark calculation

Formatted: Font: Not Bold, Font colour: Blue

Formatted: Indent: Left: 1.27 cm

Formatted: Font: Not Bold, Font colour: Blue

Formatted: Font: Not Bold, Font colour: Blue

```

"environmentalScore": 72.05,
"governanceScore": 58.43,
"socialScore": 74.05,
"benchmarkdata": [
  {
    "indexName": "DOW JONES30",
    "indexTotalScore": 51.76,
    "envIndexScore": 73.62,
    "socialIndexScore": 78.8,
    "govIndexScore": 67.27
  },
  {
    "indexName": "NASDAQ100",
    "indexTotalScore": 43.38,
    "envIndexScore": 56.47,
    "socialIndexScore": 68.37,
    "govIndexScore": 65.98
  },
  {
    "indexName": "S&P500",
    "indexTotalScore": 47.19,
    "envIndexScore": 55.65,
    "socialIndexScore": 66.65,
    "govIndexScore": 61.56
  },
  {
    "indexName": "EURO Stocks",
    "indexTotalScore": 61.79,
    "envIndexScore": 80.27,
    "socialIndexScore": 86.27,
    "govIndexScore": 69.77
  }
]

```

We are calculating benchmark for Synthetic data and Sustainability concerning Nasdaq 100, Dowjones, S&P 500 and EURO Stocks.
In particular "Total ESG combined," "Total E score," "Total S score" and "Total G score."

For example, following table shows ESG details for the company "Activision Blizzard," where ESG combined, E score, S score and G scores are

Formatted	... [1]
Formatted	... [2]
Formatted	... [3]
Formatted	... [4]
Formatted	... [5]
Formatted	... [6]
Formatted	... [7]
Formatted	... [8]
Formatted	... [9]
Formatted	... [10]
Formatted	... [11]
Formatted	... [12]
Formatted	... [13]
Formatted	... [14]
Formatted	... [15]
Formatted	... [16]
Formatted	... [17]
Formatted	... [18]
Formatted	... [19]
Formatted	... [20]
Formatted	... [21]
Formatted	... [22]
Formatted	... [23]
Formatted	... [24]
Formatted	... [25]
Formatted	... [26]
Formatted	... [27]
Formatted	... [28]
Formatted	... [29]
Formatted	... [30]
Formatted	... [31]
Formatted	... [32]
Formatted	... [33]
Formatted	... [34]
Formatted	... [35]
Formatted	... [36]
Commented [SS43]: @Vrushali Shelke -Update benchm	... [38]
Commented [VS44]: Updated the response with 'EURO	... [39]
Commented [SS45]: Done	
Deleted: {¶	... [37]
Deleted: ¶	... [40]
Deleted: and	
Deleted: t	

fetches against Synthetic data or Sustainalytics if the portfolio is uploaded against Synthetic data or Sustainalytics data providers, respectively. Script is written to fetch weight from Nasdaq 100, Dowjones and S&P 500.

NAME	Activision Blizzard
ESG Combined	60.76473888
E Score	21.00775194
S Score	53.14421419
G Score	79.25395345
Weights	0.517

Weights have been derived from the market cap of the benchmark indices. These weights are not static because it depends on stock price and outstanding shares. They can be found at below sites for Dowjones, S&P 500 and Nasdaq 100 resp.

<https://www.slickcharts.com/dowjones>
<https://www.slickcharts.com/sp500>
<https://www.slickcharts.com/nasdaq100>

Now combined score will be calculated by following formula:

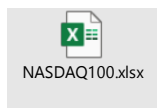
- $\text{Total_ESG_Combined} = (\text{ESG Combined} * \text{Weights})/100$
- $\text{Total_E_Score} = (\text{E Score} * \text{Weights})/100$
- $\text{Total_S_Score} = (\text{S Score} * \text{Weights})/100$
- $\text{Total_G_Score} = (\text{G Score} * \text{Weights})/100$

For example, after applying above formula, we will get below figures.

NAME	Activision Blizzard
Total_ESG_Combined	0.3141537
Total_E_Score	0.108610078
Total_S_Score	0.274755587
Total_G_Score	0.409742939

Above process needs to carry for each company and then summation will be performed against each component to arrive at final scores as shown below.

Total_ESG_Combined	42.64046011
Total_E_Score	56.55467451
Total_S_Score	68.16526071
Total_G_Score	65.51377456



14. InvestSuite

Business API Specification - </portfolio-mgmt/alternatives/{portfolioId}/{isin}>

InvestSuite API Specification <https://dev.storyteller.investsuite.com/api/esg>

Request:

```
{
  "current_pf": {
    "US67066G1040": 0.1,
    "US4781601046": 0.1,
    "US46625H1005": 0.1,
    "US2546871060": 0.1,
    "US7427181091": 0.1,
    "US0378331005": 0.1,
    "US00724F1012": 0.1,
    "US0231351067": 0.1,
    "US0605051046": 0.1,
    "US7170811035": 0.1
  },
  "instr_to_replace": [
    "US0378331005"
  ],
  "investable_universe": [],
  "esg_source": "SYNECHRON_1",
  "esg_target": "ESG"
}
```

Commented [AS46]: Let's add detail & significance of each field here. From where are we fetching them for request. i.e on screen parameter / db field.

Commented [SS47]: @Sachin Poojari to add required details

Request Params Details:

current_pf : all companies in the portfolio except the ones for whom we are finding the alternatives (comes from the ingested portfolio which is further stored as a part of Portfolio object)

instr_to_replace : the companies in the portfolio for whom we need an alternative from the invest suite (comes from the UI as per the user selection)

investable_universe : this is the criteria met list companies which was selected by the Portfolio manager using the various filters and then was further applied to portfolio.

esg_source : comes from the scoring methodology selected from the UI (SYNECHRON_1 for REFINTIV and SYNECHRON_2 for Yahoo Finance)

esg_target : comes from the UI as per user's selection (by default value is ESG)

Response:

```
{
  "recommended_trades": {
    "US0378331005": -0.1,
    "US4851703029": 0.1
  },
  "esg_differences": {
    "original_portfolio": {
      "ESG": 44.75,
      "ENVIRONMENTAL": 77.39999999999999,
      "SOCIAL": 82.89999999999999,
      "GOVERNANCE": 60.95
    },
    "recommended_portfolio": {
      "ESG": 47.0,
      "ENVIRONMENTAL": 76.89999999999999,
      "SOCIAL": 80.64999999999999,
      "GOVERNANCE": 58.65
    }
  },
  "allocation_differences": {
    "sector": {
      "BASIC_MATERIALS": 0.0,
```

```

"CONSUMER_CYCLICAL": 0.0,
"CONSUMER_DEFENSIVE": 0.0,
"COMMUNICATION_SERVICES": 0.0,
"ENERGY": 0.0,
"FINANCIAL_SERVICES": 0.0,
"HEALTHCARE": 0.0,
"INDUSTRIALS": 0.1,
"REAL_ESTATE": 0.0,
"TECHNOLOGY": -0.1,
"UTILITIES": 0.0
},
"region": {
  "EQUITY_EUROPE_DEVELOPED": 0.0,
  "EQUITY_NORTH_AMERICA": 0.0,
  "EQUITY_ASIA_PACIFIC_DEVELOPED": 0.0,
  "EQUITY_EMERGING": 0.0,
  "EQUITY_FRONTIER": 0.0,
  "EQUITY_OTHER": 0.0
},
"currency": {
  "USD": 0.0
}
},
"historical_performance": {
  "original_portfolio": {
    "2015-09-30 00:00:00": 1.0,
    "2016-01-01 00:00:00": 1.153955388435782,
    "2017-01-02 00:00:00": 1.6631301503915857,
    "2018-01-01 00:00:00": 2.1501235627142867,
    "2019-01-01 00:00:00": 2.1242551909653986,
    "2020-01-01 00:00:00": 3.177805083664914,
    "2020-09-28 00:00:00": 4.594329671479895
  },
  "recommended_portfolio": {
    "2015-09-30 00:00:00": 1.0,
    "2016-01-01 00:00:00": 1.1402758443271715,

```

```

        "2017-01-02 00:00:00": 1.649782657882474,
        "2018-01-01 00:00:00": 2.1126055943231325,
        "2019-01-01 00:00:00": 2.0838393377291348,
        "2020-01-01 00:00:00": 3.0709042050794257,
        "2020-09-28 00:00:00": 4.377960534064353
    },
    "performance_differences": {
        "tracking_error": 0.02059818152747542
    }
}

```

Commented [AS48]: We should add details , how these responses been consumed on our screen.

Commented [SS49]: @Sachin Poojari - To add required details

Commented [SS50]: Done

Parts of response consumed:

recommended_trades: These are the alternatives suggested by the Invest suite for the given set of companies. This set of stocks replaces the companies for whom the alternatives were searched for and, they are used in calculating the simulated portfolio for the original portfolio.

historical_performance: These are the historical performance year wise breakup between the original portfolio and the InvestSuite recommended portfolio. These points are used to plot a historical graph in the portfolio simulation screen.

Business Service API's:

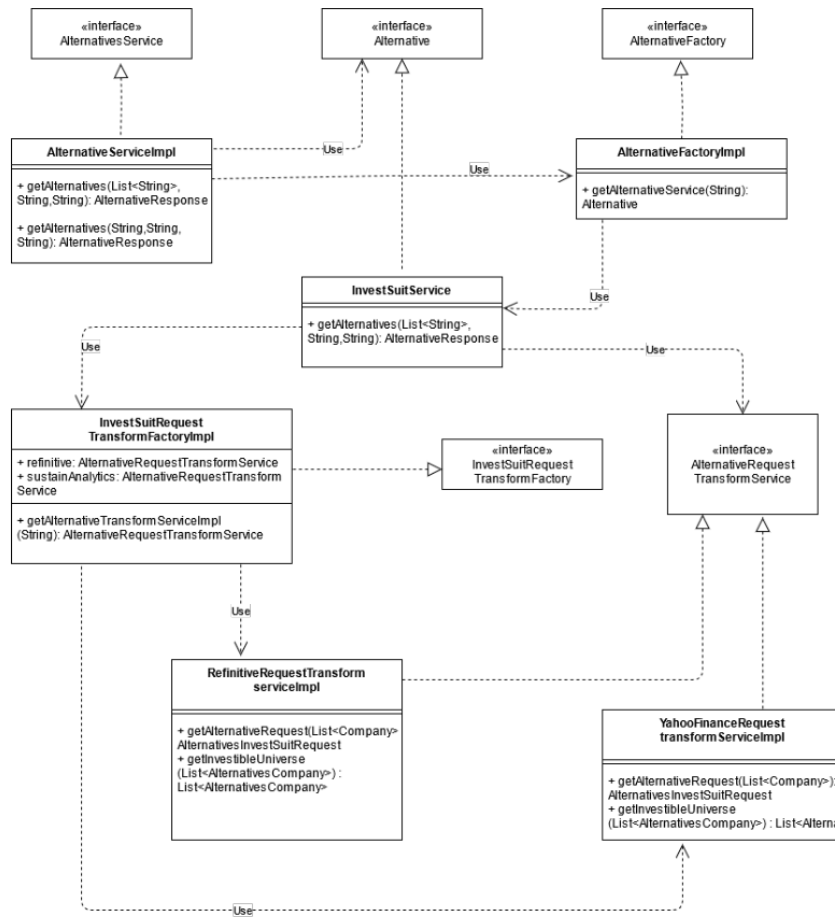
/alternatives/{portfolioId}/{isin}:

this is the API which is used to find alternatives for single company.

/alternatives (accepts the ISIN's of companies for which the alternatives are to be found)

this is the API used to find alternatives for the one or multiple companies and calculate the response for historical graphs and the simulation portfolio using the calculation service layer

Class Diagram:



Commented [AS51]: We should add details of responsibility of each layers DE/DS/MS involved producing correct information.

15. ESG score normalization logic and calculation

What is the need?

- The asset or portfolio managers who are already advance in sustainability and ESG scoring, they often use 2 or 3 providers to check ESG score/rating because relying only on one provider is not enough. Different providers have different ratings, so this Normalization feature will give signal to the analyst to look at the company when the difference between ratings is too big.
- Scaling factor is different amongst various ESG data providers like Synthetic data (0 to 100) and Sustainalytics (100-0). The purpose is to compare two different data providers on the same scale and find outlier.
- The purpose of Outlier is to highlight to portfolio manager that there is a difference in normalized score between Synthetic data and Sustainalytics and they might need to perform further research on this company.
- In case of Synthetic data ESG score for a company is better if its near to 100 and in case of Sustainalytics ESG score is better if its near to '0'.

Calculation logic:

- First calculate Average and Std deviation based on Synthetic data industry wise.
- After this Zscore can be calculated for company by formula – $(\text{ESG score} - \text{Avg score}) / \text{Std Deviation}$
- Four-point range - Divide the z score series into four range points that are from 1.25 to -1.25
- Normalize score - Divide the values (four range points) by 5 to make it 1-point range and add 0.5 (so that mean of both the series can be the same and comparable)
- Outlier - At last when one score is above avg and other score is below avg, we need to calculate the outlier on absolute difference. By taking the difference between the normalized score of both the methodologies, outliers could be derived.

Detecting discrepancy between ESG providers using different outlier's detection methods

Formula -> $\text{Absolute} (\text{Normalized score of Sustainalytics} - \text{Normalized score of Synthetic data}) * 100$
and this will be calculated on the Synthetic data industry wise.

- For Gradient purpose, we are calculating the 10th percentile of the outlier on portfolio basis. Only those companies which has outlier more than this 10th percentile value will be marked as outlier in dashboard.



Normalization_Method_2.0.xlsx

Commented [AS52]: Lets add details of , Which service handles these calculation . i.e name of service

16. ESG Forecasting API

- Forecast the ESG score for the following year by using ESG historical data and aggregated sentiment score from news articles.

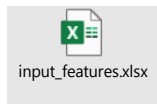
- The service is based on ESG Predictor inference class, that loads the model from the provided local directory, processes the data in the form convenient for inference and forecasts the scores for the following year.
- At backend Data science component performs integration with Synthetic data for forecasting.

It performs following activities:

- Using econometric models to capture the impact across different industries
- Showing the difference between models
- Explain how the sentiment score is driving prediction

The predications are made by two models:

- One that includes sentiment feature.
- One that does not include sentiment feature.



Business API Specification - </portfolio-mgmt/forecast/{isin}>

this is the API which is used to get the predictions with sentiments and the actual predictions for a given company in a portfolio. It also returns the response for the feature importance (Score, Currency, Aggregated Annual News Sentiment Score) for the ESG and E, S, G Pillars.

Data service API - /esg_predict/

Sample request:

```
{
  "TRBC Industry Group Name":"Investment Banking & Investment Services",
  "INSTRUMENT(NAME)":"3I GROUP",
  "ESG Currency":"£",
  "historical_scores":[
    {
      "Date":"2013-12-31",
```

```
        "ESG Score":41.729385553,  
        "sentiment_score_avg":-0.190465116  
    },  
    {  
        "Date":"2014-12-31",  
        "ESG Score":60.588878261,  
        "sentiment_score_avg":0.405238095  
    },  
    {  
        "Date":"2015-12-31",  
        "ESG Score":52.180134381,  
        "sentiment_score_avg":0.33  
    },  
    {  
        "Date":"2016-12-31",  
        "ESG Score":65.045879743,  
        "sentiment_score_avg":null  
    },  
    {  
        "Date":"2017-12-31",  
        "ESG Score":67.517471728,  
        "sentiment_score_avg":null  
    },  
    {  
        "Date":"2018-12-31",  
        "ESG Score":70.909259628,  
        "sentiment_score_avg":0.48125  
    },  
    {  
        "Date":"2019-12-31",  
        "ESG Score":62.155937306,  
        "sentiment_score_avg":0.472  
    }  
]  
}
```

Commented [AS53]: Let's add details of each field . i.e importance & source

Commented [SS54]: @Sachin Poojari - to add details

Commented [SS55]: Done

Request Params Details:

TRBC Industry Group Name: name of the company as extracted from the data lake

INSTRUMENT(NAME): ticker extracted from the data lake

ESG Currency: currency extracted from the data lake

historical_scores : the historical ESG scores and the news sentiment scores for a given company extracted from data lake

Sample response:

```
{
  "3I GROUP": {
    "predictions_without_sentiment": [
      {
        "Date": "2014-12-31T00:00:00",
        "ESG Score": 39.74907214465777
      },
      {
        "Date": "2015-12-31T00:00:00",
        "ESG Score": 57.47433741587677
      }
    ],
    "predictions_with_sentiment": [
      {
        "Date": "2014-12-31T00:00:00",
        "ESG Score": 39.78787245482832
      },
      {
        "Date": "2015-12-31T00:00:00",
        "ESG Score": 57.3892990269041
      }
    ]
  }
}
```



```
}
```

Parts of response consumed:

predictions_with_sentiment : these are the predictions which along with actual/historical predictions are plotted onto the forecasting graph.

Data service API - /feature_importance

Request: No Params

Response:

```
{
  "esg_feature_importance": {
    "ESG Combined Score": 18.0171689633384,
    "ESG Currency": 0.8473637374991291,
    "Aggregated Annual News Sentiment Score": 0.01393994837467
  },
  "e_feature_importance": {
    "Environment Pillar Score": 25.7066469069308,
    "ESG Currency": 0.6640434564048371,
    "Aggregated Annual News Sentiment Score": 0.056191966230039
  },
  "s_feature_importance": {
    "Social Pillar Score": 23.309829251348802,
    "ESG Currency": 0.772724979451769,
    "Aggregated Annual News Sentiment Score": 0.038437375574497
  },
  "g_feature_importance": {
    "Governance Pillar Score": 18.615534648321603,
    "ESG Currency": 1.09137315756875,
    "Aggregated Annual News Sentiment Score": 0.025678589567727995
  }
}
```

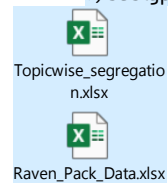
17. RavenPack -News Sentimental Analysis

It performs following activities:

- Applying Named Entity Recognition to identify company name for news articles
- Classifying articles to filter out only ESG related news (Labor issues, Pollution, natural disasters) etc.
- Measure sentiment score.

Sentiment score calculation on news article. Following points needs to be considered while calculating sentiment score, in particularly TOPIC, GROUP and TYPE.

- 1) Entity (Company name)
- 2) Origin
- 3) Event sentiment score
- 4) **TOPIC**
- 5) **Group**
- 6) **TYPE**
- 7) Subtype.



Data engineering is extracting news from RavenPack on daily basis. Batch process will trigger and will extract the data from RavenPack.

API specification for news feed

- Business API Specification - /portfolio-mgt/company-news-feeds/{isin}
- Data service API - /datalake/company-news-feeds/{isin}

API specification for heat map

- Business API Specification - /portfolio-mgt/company-heat-map/{isin}
- Data service API - /datalake/company-heat-map/{isin}

Commented [AS56]: We should make this NEWS analysis MS , which could be reusable further in any other 'accelerators.

Commented [SS57]: WIP

Sample query for heat map:



Sample query for latest top 5 news:



Here are Ravenpack credentials for accessing the console with API key for data extraction.

Amlesh Kulkarni

=====

Username: Amlesh.Kulkarni@synechron.com

Password: [Create Password](#)

API Key: QyiWkDESKzLRigYAsUIW5F

*You will need to populate any API request made with this key. Further information can be found in the API Documentation link below.

Vaibhav Ranbhare

=====

Username: Vaibhav.Ranbhare@synechron.com

Password: [Create Password](#)

API Key: deoLVcptSH9BAwUjSOTw4o

*You will need to populate any API request made with this key. Further information can be found in the API Documentation link below.

Kumar Rahul

=====

Username: Kumar.Rahul@synechron.com

Password: [Create Password](#)

API Key: IzxFtv0w2iSiEmVNIthOJ9

*You will need to populate any API request made with this key. Further information can be found in the API Documentation link below.

You will be able to access any of the links below with your username and your new password:

Product Site:

<https://app.ravenpack.com/>

Help Center (includes user guides and reference files):

<https://app.ravenpack.com/help/>

API Documentation (for programmatic access):

<https://app.ravenpack.com/api-documentation/>

Research Library:

<https://app.ravenpack.com/insights/research/>

18. Mathematical Calculation/DS Algorithms

ESG Booster Application which uses specific Mathematical Calculation/DS Algorithms. Please find document which contains all key features and the details



ESG%20Booster_Feature_Calculation_Logic

19.Requirement Vs. End Points - Traceability Matrix

TODO

20.End Point Definitions

Following Swagger URL contains endpoint definitions.

<https://investtech-datalake-dev.synechron.net/datalake/swagger-ui.html>

<https://investtech-esg-booster-services.synechron.net/portfolio-mgmt/swagger-ui.html>

Commented [AS58]: Let's use add this important information for API call details

Commented [SS59]: Done

21. Test Strategy

17.1 Scope of Testing

Following type of testing will be carried out

ADD DESCRIPTIONS ON unit test framework and strategy

- Unit Testing
- Service Testing

17.2 Assumptions

Following are the assumptions that may impact testing

- Swagger UI will be used as a test harness for Service level testing.

18 Integration Strategy

18.1 Integration with proprietary systems

Not Applicable

18.2 Integration with third party systems

Not applicable

19 Risks

Severity: High

Severity: Low

20 Application Setup

20.1 Prerequisite

- a. Java 1.8 update 131
- b. Git
- c. IntelliJ IDEA

20.2 Obtain Source Code

From a terminal window execute following command.

TODO

20.3 Open app in IntelliJ-IDEA

21 Running App

TODO

- <https://investtech.esg.synechron.net>

22. CI/CD Setup

TODO

22.1 CI /CD Gitlab scripts

23. Code review guidelines

- Java code guidelines standard



- React code guidelines standard

TODO

24. Hardware requirements

- AWS r5.2xlarge instance

25. Software requirements

- Ubuntu Linux 16.04 (LTS)
- JDK 8.0
- Spring 4.3
- React
- Docker CE 17.0
- Docker Compose 1.15
- Google Chrome 62 or above.

26. Data ingestion steps

For Booster new database ingestion, we have to follow the below mentioned steps which we have used for our client Dowjones and other details are given below:

Mapping Sheet Name: ESG-Booster-Data-Catalogue-DowJones.xlsx

Old Column Name: Column C (Old Column Name)

New Column Name: Column H (DowJones Fields)

Data File Name: Column I (Refer DowJones JSON file)

1) Table Name : ESG_InfoDS1

use existing dowjones "Investable_Universe" table to populate below columns data isin,Ticker,Company_name

To insert the data for Remaining columns refer ESG-Booster-Data-Catalogue-DowJones.xlsx sheet and use aggregate.json file

2) Table Name: Investable Universe

To populate the below columns data use "ESG_InfoDS1" table with latest date DS1_ESGScore,DS1_EnvironmentScore, DS1_ESGConsolidatedScore,DS1_ESGDisputableScore,DS1_SocialScore,DS1_BusinessModelInnovation,DS1_HumanCapital,DS1_GovernanceScore

To populate the below columns data use "existing dowjones Investable_Universe" table ISIN,Country_Code,ISIN_Type,Currency_Code, Currency_Symbol, Environment_Outlier_Score,

Social_Outlier_Score,Governance_Outlier_Score,HumanCapital_Outlier,BusinessModelInnovation_Outlier,ESG_Outlier,

To insert the data for Remaining columns, refer ESG-Booster-Data-Catalogue-DowJones.xlsx sheet and use universe_full.json file

For DS2 columns populate with '0', as we don't have the data

3) Table Name : news_data_agg_month

To insert isin column data use Investable_Universe table from old schema.

To insert the data for Remaining columns, refer ESG-Booster-Data-Catalogue-DowJones.xlsx sheet and use asp.json file

To update below columns description in news_data_agg_month table refer 20220307_sasb_names file

topic = general_issue_category
news_group = sustainability_dimension

4) Table Name: news_data_filtered_top5

use "Investable_Universe" table to populate below columns data isin,entity_name,country_code
To insert the data for Remaining columns, refer ESG-Booster-Data-Catalogue-DowJones.xlsx sheet and use asp.json file
To update `type` column in news_data_filtered_top5 table refer 20220307_sasb_names file

type = general_issue_category

5) Table Name: esg_hist_co2

Use "Investable_Universe" table to populate below columns data isin,Company_name as instrument_name
To insert the data for Remaining columns, refer ESG-Booster-Data-Catalogue-DowJones.xlsx sheet and use
long_gic.json
long_gic_un_adj
total_gic.json files

Note:

- 1) Use existing Benchmark,company_card_data tables in dowjones_new database also
- 2) To populate esg_hist_co2 table refer dowjones_new-esg_hist_co2.sql file
- 3) To populate ESG_InfoDS1 table refer to dowjones_new-ESG_InfoDS1.sql file.
- 4) To populate Investable_Universe table refer dowjones_new-Investable_Universe.sql file.
- 5) To populate news_data_agg_month table refer dowjones_new-news_data_agg_month.sql file.
- 6) To populate news_data_filtered_top5 table refer dowjones_new-news_data_filtered_top5.sql file.

Page 38: [1] Formatted Vrushali Shelke 27/11/2020 07:27:00

Font: Not Bold, Font colour: Blue

Page 38: [2] Formatted Vrushali Shelke 27/11/2020 07:27:00

Font: Not Bold, Font colour: Blue

Page 38: [3] Formatted Vrushali Shelke 27/11/2020 07:27:00

Font: Not Bold, Font colour: Blue

Page 38: [4] Formatted Vrushali Shelke 27/11/2020 07:27:00

Font: Not Bold, Font colour: Blue

Page 38: [5] Formatted Vrushali Shelke 27/11/2020 07:27:00

Font: Not Bold, Font colour: Blue

Page 38: [6] Formatted Vrushali Shelke 27/11/2020 07:29:00

Indent: Left: 2.54 cm

Page 38: [7] Formatted Vrushali Shelke 27/11/2020 07:27:00

Font: Not Bold, Font colour: Blue

Page 38: [8] Formatted Vrushali Shelke 27/11/2020 07:27:00

Font: Not Bold, Font colour: Blue

Page 38: [9] Formatted Vrushali Shelke 27/11/2020 07:27:00

Font: Not Bold, Font colour: Blue

Page 38: [10] Formatted Vrushali Shelke 27/11/2020 07:27:00

Font: Not Bold, Font colour: Blue

Page 38: [11] Formatted Vrushali Shelke 27/11/2020 07:27:00

Font: Not Bold, Font colour: Blue

Page 38: [12] Formatted Vrushali Shelke 27/11/2020 07:27:00

Font: Not Bold, Font colour: Blue

Page 38: [13] Formatted Vrushali Shelke 27/11/2020 07:27:00

Font: Not Bold, Font colour: Blue

Page 38: [14] Formatted Vrushali Shelke 27/11/2020 07:27:00

Font: Not Bold, Font colour: Blue

Page 38: [15] Formatted Vrushali Shelke 27/11/2020 07:27:00

Font: Not Bold, Font colour: Blue

Page 38: [16] Formatted Vrushali Shelke 27/11/2020 07:27:00

Font: Not Bold, Font colour: Blue

Page 38: [17] Formatted Vrushali Shelke 27/11/2020 07:27:00

Font: Not Bold, Font colour: Blue

Page 38: [18] Formatted Vrushali Shelke 27/11/2020 07:27:00

Font: Not Bold, Font colour: Blue

Page 38: [19] Formatted Vrushali Shelke 27/11/2020 07:27:00

Font: Not Bold, Font colour: Blue

Page 38: [20] Formatted Vrushali Shelke 27/11/2020 07:27:00

Font: Not Bold, Font colour: Blue

Page 38: [21] Formatted Vrushali Shelke 27/11/2020 07:27:00

Font: Not Bold, Font colour: Blue

Page 38: [22] Formatted Vrushali Shelke 27/11/2020 07:27:00

Font: Not Bold, Font colour: Blue

Page 38: [23] Formatted Vrushali Shelke 27/11/2020 07:27:00

Font: Not Bold, Font colour: Blue

Page 38: [24] Formatted Vrushali Shelke 27/11/2020 07:27:00

Font: Not Bold, Font colour: Blue

Page 38: [25] Formatted Vrushali Shelke 27/11/2020 07:27:00

Font: Not Bold, Font colour: Blue

Page 38: [26] Formatted Vrushali Shelke 27/11/2020 07:27:00

Font: Not Bold, Font colour: Blue

Page 38: [27] Formatted Vrushali Shelke 27/11/2020 07:27:00

Font: Not Bold, Font colour: Blue

Page 38: [28] Formatted Vrushali Shelke 27/11/2020 07:27:00

Font: Not Bold, Font colour: Blue

Page 38: [29] Formatted Vrushali Shelke 27/11/2020 07:27:00

Font: Not Bold, Font colour: Blue

Page 38: [30] Formatted Vrushali Shelke 27/11/2020 07:27:00

Font: Not Bold, Font colour: Blue

Page 38: [31] Formatted Vrushali Shelke 27/11/2020 07:27:00

Font: Not Bold, Font colour: Blue

Page 38: [32] Formatted Vrushali Shelke 27/11/2020 07:27:00

Font: Not Bold, Font colour: Blue

Page 38: [33] Formatted Vrushali Shelke 27/11/2020 07:27:00

Font: Not Bold, Font colour: Blue

Page 38: [34] Formatted Vrushali Shelke 27/11/2020 07:27:00

Font: Not Bold, Font colour: Blue

Page 38: [35] Formatted Vrushali Shelke 27/11/2020 07:27:00

Indent: Left: 1.27 cm

Page 38: [36] Formatted Vrushali Shelke 27/11/2020 07:27:00

Font: Not Bold, Font colour: Blue

Page 38: [37] Deleted Vrushali Shelke 27/11/2020 07:28:00

▼

Page 38: [38] Commented [SS43] Shridhar Sarnobat 27/11/2020 11:58:00

[@Vrushali Shelke](#) -Update benchmark data for Euro stock

Page 38: [39] Commented [VS44] Vrushali Shelke 27/11/2020 13:02:00

Updated the response with 'EURO Stocks'. Please Review [@Shridhar Sarnobat](#)

Page 38: [40] Deleted Vrushali Shelke 27/11/2020 07:28:00

I