

Requirements

Initial Draft Version 1.0

Author: *jd*
Last reviewed: *January 03, 2023*

Table of Contents

1	Document Objectives and Purpose	1
2	Restrictions	1
3	Requirements	2
4	Assumptions.....	2
5	Use Cases.....	2
6	Architecture High-Level Overview	3
6.1	Get Forecast Service	4
6.1.1	Basic Flow	4
6.1.2	Messages	4
6.2	Train Model Service	4
6.3	View/Insert/Update Client Parameters.....	4
6.3.1	Update Client Parameters Flow	5
7	Database.....	5
8	Technical Choices and Technologies used for Development	5
9	Security/Privacy	5
10	Other	5

1 Document Objectives and Purpose

This document contains information about the requirements and the decisions related with the integration of an existing application, referred to in this document as *Work Force Management (WFM)*, and a *backend* that provides services to allow the use of various machine learning models and related algorithms to compute forecasts. The information in this document also serves as a base for the development of the *backend* mentioned and to establish its scope, purpose, objectives, concerns, and responsibilities. Therefore, the main purpose of this document is to establish a contract between *WFM* and the *backend* to be developed.

2 Restrictions

R1: The backend is installed and runs in a Windows operating system environment.

R2: WFM initiates and establishes connections with the backend. The backend responds to requests, it does not initiate requests.

R3: The backend does not provide mechanisms to authenticate clients. Client authentication is performed by the WFM client application.

R4: The backend does not provide mechanisms to create and or manage security tokens used for authorization purposes.

R5: The backend and its architecture, initially, do not consider functionalities like balance loading, distributed processing, and automatic service provisioning.

Commented [JD1]: Some restrictions may apply. This is the place where restrictions should be presented and discussed.

3 Requirements

RQ1: The integration of predictive (forecast) capabilities provided by machine learning models with the WFM client application, according to the objectives defined in this project, should be accomplished using web services. This set of web services will also be referred to in this document as the backend.

RQ2: The WFM client application and the backend are independent applications and may run (exist) in different machines and network domains.

RQ3: The WFM client application and the backend communicate using a secure encrypted communication channel, e.g., the HTTPS protocol, that provides secure communications, data integrity and privacy. Data is exchanged between the two applications using JSON.

RQ4: The backend should not need to know about real clients or how to access their data. The backend receives all the input data needed to provide its services, from the WFM client application in the corresponding (respective) requests.

RQ5: The backend can run (exist) in a container, virtual or physical machine, but its services and dependencies exist on a single node (machine). That is, initially, a distributed architecture is not considered (assumed).

RQ6: The backend expects and only accepts requests containing an Authorization Bearer Header token.

RQ7: The backend can handle multiple requests, possibly, simultaneous.

4 Assumptions

A1: Clients run the WFM client application on their systems (machines).

A2: The backend does not provide authentication and/or access control primitives and mechanisms. It is assumed that clients communicating with the backend are authenticated by an external entity, e.g., by WFM.

A3: The backend only accepts requests containing a valid authorization token. The authorization token is created (issued) and managed by an external entity, e.g., WFM.

A4: The backend runs in a single machine: this can be a physical or a virtual machine. This means, in particular, that a distributed architecture/system is not assumed. Several processes (instances) can however exist, e.g., several processes running (instances of) a service can exist, on the same machine.

A5: The communication between the backend and its clients uses the HTTPS protocol. The backend runs behind a reverse proxy that establishes and manages HTTPS communications with the clients.

A6: There are no restrictions on the technologies and architecture used to develop the backend other than the restrictions and requirements presented in Sections 2 and 3, that apply.

Commented [JD2]: Some assumptions may apply. This is the place where assumptions should be presented and discussed.

5 Use Cases

Figure 1 presents the main use cases considered¹.

¹ Retraining and monitoring model accuracy decay were not considered.

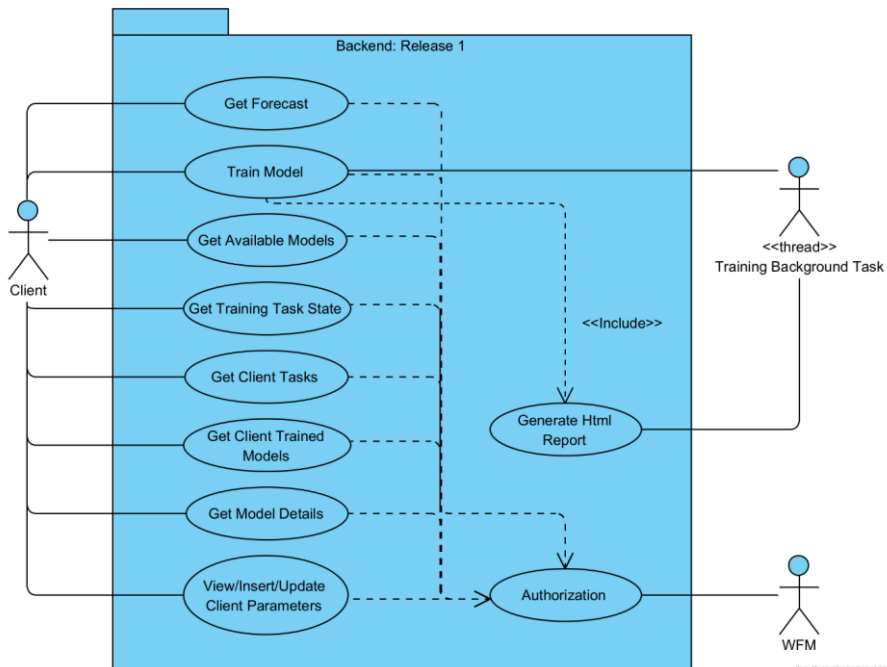


Figure 1 Main use cases.

In Figure 1 *WFM* refers to the SISQUAL Work Force Manager application.

6 Architecture High-Level Overview

Figure 2 presents a high-level overview of the backend architecture and Figure 3 presents an example of a possible deployment architecture.

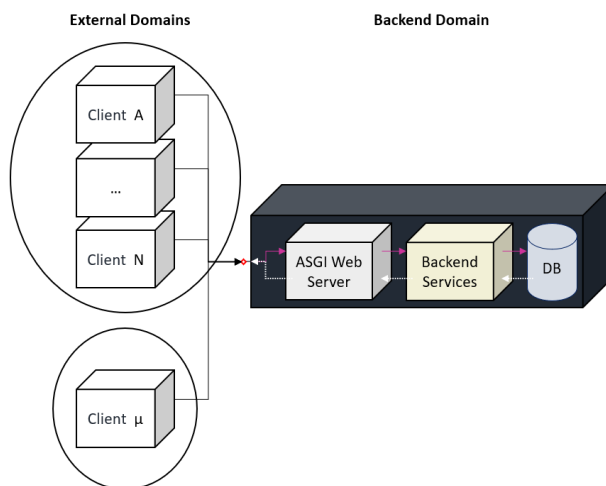


Figure 2 Backend architecture high-level overview.

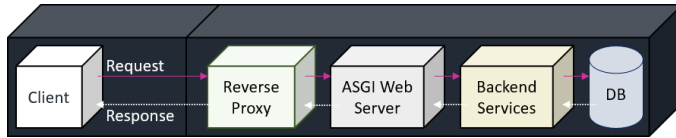


Figure 3 Example of a deployment architecture.

6.1 Get Forecast Service

6.1.1 Basic Flow

1. A client executes (starts) WFM. This implies that the client is authenticated (goes through an authentication process).
2. A client asks for a forecast in WFM using a previously trained model.
3. WFM makes a request to the forecast service running in the backend. The request contains an authorization token and all the input data needed (required) by the model to compute a forecast.
4. The forecast service loads the model and gets a forecast that is sent to WFM. The response includes the authorization token.

6.1.2 Messages

Direction	<i>Request</i>
Message Type/Format	<i>JSON</i>
Content	<i>Client Id</i> <i>Model Id</i> <i>Input data needed by the model to get a forecast</i>
Direction	<i>Response</i>
Message Type/Format	<i>JSON</i>
Content	<i>Forecast</i>

6.2 Train Model Service

6.3 View/Insert/Update Client Parameters

A client has a set of parameters associated with his/her requests, that are stored in the backend database. These parameters can be viewed and updated.

Known Information:

All clients have the same number and type of parameters which are already defined.

The insertion process has not yet been defined.

A client can have more than one store. If so, how does that impact the backend implementation?

6.3.1 Update Client Parameters Flow

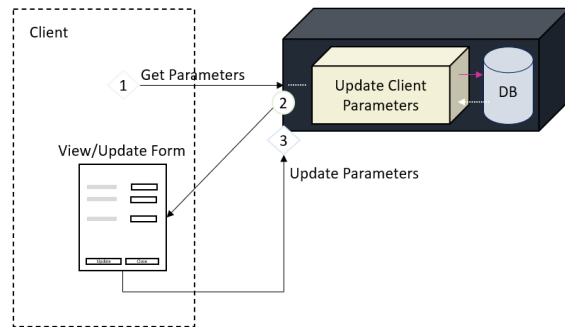


Figure 4

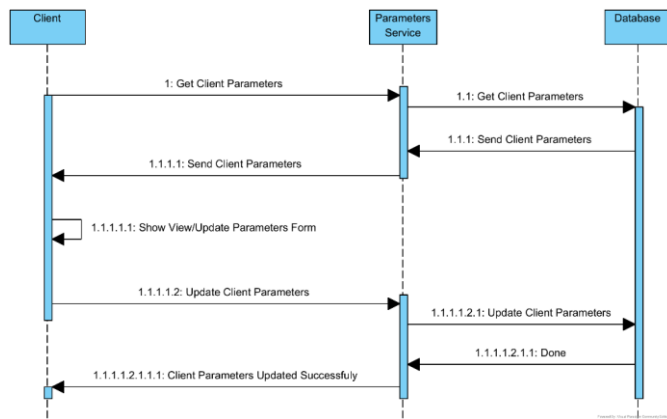


Figure 5

7 Database

The backend uses a database to store information about clients, models and training tasks, among other things.

8 Technical Choices and Technologies used for Development

Technical choices, restrictions and limitations, and technologies used to develop and integrate the *backend* with the client *WFM*.

9 Security/Privacy

Architecture overview...

10 Other

...