



Assistant Engineer Internship Report

ARCHITECTURE AND DEVELOPMENT OF A BACK-OFFICE SERVICE FOR
THE MANAGEMENT OF CONNECTED OBJECTS (IOT DEVICES)

1ST OCTOBER 2018



IETA 2019 NOELE Elodie — Robotics Promotion Lieutenant-général Jean-Baptiste
Vaquette de Gribeauval

Contents

1	Abstract	2
2	Résumé	3
3	Acknowledgements	4
4	Introduction	5
5	Context	6
5.1	Technicolor Overview	6
5.1.1	What is Technicolor	7
5.1.2	Technicolor branches	8
5.1.3	Technicolors Rennes branch overview	9
6	Economic analysis of the company	10
7	Technological, economic and human challenges	11
7.1	Technological and economic challenges	11
7.2	Teamwork challenge	12
7.3	Personal challenge	13
8	Technological, economic and human challenges	14
8.1	Approach to the world of engineering	14
8.2	Project Management Approach	15
9	Achievements during the internship	16
9.1	Global overview of the project	16
9.2	Specifications of the project	17
9.2.1	First approach of the internship	17
9.2.2	Review of work already done by the other trainee	18
9.2.3	What is a REST API?[1]	19
9.2.4	Drafting of the Backend API documentation	20

9.3	Designing the app	21
9.3.1	Overview of a JavaScript web application	21
9.3.2	Architecture of the MariaDB database	22
9.3.3	Architecture of the Backend[2]	23
9.3.4	Architecture of the MQTT Broker	24
9.3.5	Docker container deployment[3]	25
9.3.6	Overall operation of the application in relation to the goals of the internship	26
9.4	Demos of the app	28
9.4.1	First demo	28
9.4.2	Second demo	28
9.4.3	Third demo	29
9.5	Testing the application	30
10	Conclusion	31
	Glossary	33
A	Global overview of the Meeting Assistant Application	35
B	UML diagram of the Meeting Assistant Database	36
C	Architecture of a web application	37

Abstract

This internship is part of the training of an ENSTA Bretagne student. The purpose is to understand the constraints of an engineer by taking the position of an assistant engineer. The aim of this trainee was to propose a POC of a web solution for booking meeting rooms which had to be supplementary to Outlook and to be connected with an IOT device, the "Time Keeper". During this internship, I had the opportunity to design the Backend of the web application. It was an opportunity to become more familiar with the world of software engineering through JavaScript.

Finally, I was able to implement a working Backend which fulfilled most of the specifications. Later, other features may be added.

It was truly challenging to attain the internship goals in such a short period of time as I was a beginner in web programming.

Résumé

Ce stage s'inscrit dans le cadre de la formation d'un ingénieur ENSTA Bretagne. Il vise à appréhender les contraintes d'un ingénieur en prenant le poste d'un assistant ingénieur. Le but de ce stage était de proposer une POC d'une solution web de réservation des salles de réunions complémentaire à Outlook et qui pourra être interfacé avec un objet connecté, le **Time Keeper**. Durant ce stage, j'ai eu l'occasion de m'occuper de la conception du Backend de l'application web. Ce fut l'occasion d'appréhender le monde de l'ingénierie logicielle au travers du langage JavaScript.

Le stage a finalement abouti sur un backend fonctionnel, remplissant une majeure partie du cahier des charges. Par la suite, d'autres fonctionnalités pourront lui être ajoutées.

Ce fut donc un réel challenge en raison des objectifs à atteindre dans un court laps de temps, surtout pour des débutants en programmation web.

Acknowledgements

I thank everyone who supported me during the internship, especially my tutor, Laurent Dauvin.

I would also like to thank Yoann Amicel and Paul Baron, the two other interns who worked with me on the project, as well as their tutors, Sébastien Fraleu and Christian Schahl.

I would also like to thank Morgan Le Floch and his tutor, Frédéric Poupard, for all the help they provided during the design of the Backend as well as the entire Technicolor IT team who helped me debug a lot of problems.

I also thank Jean-Claude Colmagro without whom this training course would not have taken place.

Finally, I thank all the people in my open space as well as those I met during my internship and who made my internship enjoyable.

Introduction

As part of the ENSTA Bretagne engineer's training, I undertook my engineering assistant internship at Technicolor Connected Home in Cesson-Sévigné from 27 May to 31 August 2018. This internship was an opportunity to get hands-on experience of the engineering world to prepare for my graduation project.

As part of the ENSTA Bretagne engineer's training, I undertook my engineering assistant internship at Technicolor Connected Home in Cesson-Sévigné from 27 May to 31 August 2018. This internship was an opportunity to get hands-on experience of the engineering world to prepare for my graduation project.

Please find at the end of the report, the assesment report filled by my tutor.

5.1 Technicolor Overview



Figure 5.1: Technicolor global overview [4]

5.1.1 What is Technicolor

Technicolor is a Creative Technology Company, focused on helping their clients deliver an ever-improving consumer experience. Technology underpins everything about the future of media and entertainment, and all the energies of their industry-leading experts go into creating seamlessly integrated, next-generation experiences for people everywhere. They harness the new economics of innovation, and help people create what people are envisioning, on-screen and off.

Their award-winning talent and highly credentialed researchers create, develop and deliver the technologies that enable amazing companies to bring next generation media and entertainment experiences to life for their customers. For the last 100 years, they have defined it for Hollywood, and for the industry around the globe. From video through to imaging and information, regardless of place, device or context, they partner with their clients to create a more vivid and seamless digital world for everyone.

Technicolor harnesses the new economics of innovation, in sync with the unrelenting rise in consumer expectations, and the constant desire to be wowed. They focus on creating and enabling the next experiences that will delight consumers, and sustainably bolster their clients businesses. That's why they deliver across the entire entertainment and media ecosystem, from content creation to consumption, and thus always stay at the forefront of imagination and engineering.

5.1.2 Technicolor branches

TECHNOLOGY GROUP

The primary source of technology, research and intellectual property development for Technicolor, their clients and their partners. As a leader in technology and patent licensing, their world-class researchers and scientists help all actors of the M and E industry fully participate in today's Innovation Economy.

ENTERTAINMENT SERVICES

From the creation of animated content, the visual and aural wonder of post-production to the manufacturing and distribution of packaged media, they set industry standards for excellence. Their unrivaled teams of creative talents and acclaimed experts develop standard-setting technologies, create awe-inspiring moments and deliver compelling entertainment experiences to consumers.

CONNECTED HOME

They partner with their customers to deliver vibrant connected life services. From connectivity excellence to immersive video experiences to Internet of Things, their work spans the digital life ecosystem for which they create smarter devices and applications that enable intelligent interactions between people and their environment.

RESEARCH AND INNOVATION GROUP

They connect research and development across Technicolor to deliver on their commitment to innovate.



Figure 5.2: Technicolor in figures

5.1.3 Technicolors Rennes branch overview



Figure 5.3: Technicolor Rennes Campus

Their support the delivery of exciting new experiences for consumers in theatres, homes and on-the-go. Technicolor Rennes offers a lot of solutions to meet the needs of Content Creators, particularly Hollywood studios, network operators and consumer electronics manufacturers.

Technicolor in Rennes hosts its largest Research and Development Centre in the world. Its 500 researchers and engineers invent the future services and products of entertainment and the connected home, innovating in new-generation technologies and video experiences.

Economic analysis of the company

In € Million	2017	2018	At constant rate
Revenues from continuing operations	2,098	1,769	(9.3) %
Adjusted <u>EBITDA</u> from continuing operations	82	57	(20.0) %
As a % of revenues	3.9%	3.2%	
Free Cash Flow from continuing operations	(109)	(150)	(41)

Figure 6.1: Economic figures of Technicolor for 2017-2018

Technicolor Connected Home is part of a large group whose results are rather good. Technicolor is a company strongly focused on research and innovation. It has sites, partners and customers all over the world. Concerning the Rennes site, the company is being reorganised in various ways:

- by accelerating the implementation of research programmes, through the creation of patents in partnership with other groups;
- with financial support agreements for research, for example, through the sale of patents to Interdigital;
- with simplification of the group structure;
- by increased partnerships with other groups to reach more customers and increase profits.

Despite the difficult market conditions, Technicolor is still a large company with great potential, offering good working conditions to its employees.

Technological, economic and human challenges

7.1 Technological and economic challenges

To carry out their mission, Rennes Technicolor employees hold numerous meetings on various topics. These meetings take place in Rennes Technicolor site and require a constant use of the limited number of meeting rooms. This leads to conflicts of a temporal nature (meetings ending late), of an object management nature (meetings planned every week that sometimes do not take place) and other issues. These problems have two main sources.

The first is related to the lack of assessment of the "Time Keeper" role in meetings. During a meeting, there are three important roles: the meeting creator, the scribe and the "Time Keeper". The latter is responsible for time management and ensuring that the meeting does not go beyond the specified end time. This role is often not addressed. This is where one of TCH's managers came up with the idea of creating an electronic hourglass whose main function would be to perform the "Time Keeper" function by warning of the time remaining during a meeting.

The second is the lack of convenience of the meeting booking tool, Outlook via Office 365, because this tool has several shortcomings. It is complicated to book a meeting room, since it is first necessary to choose a meeting room to see the availability of that room, which also does not guarantee the availability of participants. In addition, it is often necessary to extend the duration of a meeting during that meeting, but Outlook does not have this feature. Often, meeting creators must recreate a meeting, going through the room search process again depending on room availability and participants, to continue their meeting. This solution is obviously not feasible, since it is costly and time-consuming for employees. To overcome this problem, it would be useful to add a web application connected to the "Time Keeper" device via an IOT type communication.

7.2 Teamwork challenge

It was in this context that my internship took place - more precisely, the design of the Backend of the Web application. This includes integration with the Frontend, link with Outlook and interfacing with the Time Keeper. This internship is part of a series of four training courses on the theme of "Time Keeper". These internships follow a previous internship where a trainee had been asked to design a hardware and software prototype to demonstrate the feasibility of the project and then to improve it.

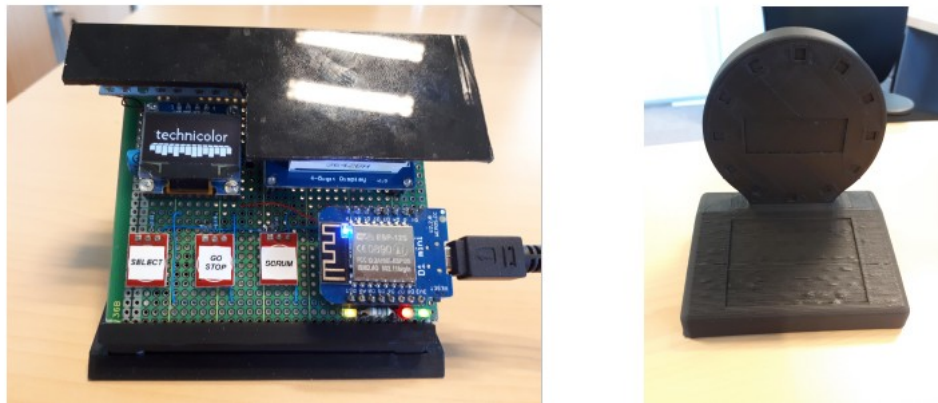


Figure 7.1: Initial prototype (left) and current prototype (right) of the Time Keeper

Hence, it is a team project involving a trainee on the Backend part, a trainee on the Frontend part, a trainee on the Hardware part and lastly a trainee on the Software part, who will not be present during this course and will have to connect the electronic object with the Backend designed during this internship. This internship is software development-oriented and includes a small part concerning IOT communication, using IOT protocols such as MQTT. Moreover, this course required reporting skills, to specify the need, write the documentation of the code and web application for potential trainees or employees of the company, to demonstrate the application to show the features developed during the course and to ensure the fulfilment of the client's request (here, TCH).

Initially, the objectives of the course were to establish a backend architecture and to manage a group of electronic objects of the "Time Keeper" type. As a trainee, it is necessary to analyse the requirement, define the software architecture of the backend, develop, make a POC and design a backend allowing new features as well as to interface it with the electronic object.

Since the trainee managing the electronics came later, the part concerning the interfacing with the electronic system could not be carried out, but the specifications concerning this interface were written.

7.3 Personal challenge

During the internship, I had to learn new technologies such as JavaScript, Docker, a virtualization tool, and to handle Git, a collaborative development tool via the SourceTree software through the Git Flow. Git Flow is a set of processes to follow in order to use Git correctly in a project involving at least two people.

More specifically, to develop the Backend I used the Express.js framework, derived from Node.js, a server-oriented JavaScript working environment, while the frontend was developed with the Vue Material framework, an overlay of Vue.js, itself being a Node.js framework.

This internship allowed me to work both on the server side and the client side, especially during the integration of the backend in the frontend which had to be performed in a rather limited time, therefore the working constraints were similar to those of a real engineer.

During my three-month internship, I created the Backend, notably the CRUD, part of the backend integration in the frontend, the backend documentation and part of the frontend documentation, as well as the specifications for the Backend integration with Outlook and with the "Time Keeper" software.

It was a challenge to produce the work required in a relatively short period of time, given my software development experience and that of the second trainee. However, it is a very good way to become familiar with the working constraints and work environment of a large company, to be on an interesting project which is important for optimizing the management of meeting rooms.

Technological, economic and human challenges

8.1 Approach to the world of engineering

This internship was an opportunity to immerse myself in the working environment of engineers as well as to understand their issues. Indeed, although the aim of the internship has no true effects on the company other than to ease the reservation of meeting rooms, it is nonetheless an important project since it involves many actors.

First, from a development point of view, this internship was an opportunity to put teamwork into practice, since any function performed by the Backend must properly meet the client's needs as well as allow the Frontend to easily make server requests.

In terms of deployment, I have had the opportunity to call on the company's IT department often. In the case of project management, this highlights the issues that may or may not delay the progress of the project. Indeed, the IT team was often asked for help, so I had to adapt to any delays in my deadlines and reorganize my action plan.

8.2 Project Management Approach

This training is also a practical application of group project management. The project allowed the implementation of a functional backend, which meets all the goals. However, there is always work to do in order to optimize as much as possible the application that I was not able to realize for the following reasons :

- As a beginner in JavaScript, I made design errors that coerce me to postpone my action plan for the project.
- The short duration of the internship for a rather large project, but it was very rewarding for beginners.

Achievements during the internship

9.1 Global overview of the project

Since this internship was limited to the realization of the web application, only the global operation of the application will be explained, which excludes communication with the physical object.

The project is an application running under Docker. We use Docker to have a lightweight web application, easily deployable on any other machine and running on a UNIX-like OS. The application is broken down into four components:

- the Frontend, created by another trainee,
- the Backend, which I created,
- the MariaDB database,
- the MQTTWS broker (MQTT Over Web sockets).

Each of these elements is deployed in a separate docker container. They communicate with each other via HTTP protocol for Frontend and Backend, MQTT protocol for Backend and Broker MQTT as well as Frontend and MQTT protocol, SQL for Backend and MariaDB. The overall architecture of the project is shown in [A](#).

9.2 Specifications of the project

9.2.1 First approach of the internship

The first task I was given during this course was to grasp the stakes of the subject. So, I had a first stage of adaptation where I was able to assess the issues of the internship to understand the work to come. To do this, I first studied the documentation already written about the project. This involved studying the project specifications, reading the Frontend documentation written by the other trainee, and studying the features of the original Time Keeper.

Thus, I could see the major features of the object that had to be reproduced in the application as well as the possible complements that the web application would bring to the electronic object. This adaptation stage was also an opportunity to have a first look at the technological tools to be used. Being new to software and web engineering, I started by doing a tutorial around JavaScript to understand what was expected from the Backend of an application.

9.2.2 Review of work already done by the other trainee

Once the first stage was completed, I had the opportunity to review the documentation written by the Frontend trainee to suggest possible evolutions of the database architecture (see [B](#)) as well as starting to design the Backend architecture. According to the project specifications, the backend must be based on a REST API.

9.2.3 What is a REST API?[1]

The REST standard designates a means of conceiving inter-software communications in the manner of Internet-style communication. It is based on the HTTP communication protocol and allows simple web requests that a client can make to a server to be written, i.e. CRUD requests (GET, POST, PUT, DELETE). The architectural constraints of this type of API are the following:

- *A REST API is stateless.* It is not possible for the backend server to identify the client. For a server, each request is separate from another, i.e. when sending a request, the client must attach all the information allowing the server to respond to its request without needing external elements to this request.
- *A REST API has a cache.* This cache makes it possible not to display on the computer of the client the various pages necessary to the success of its request. This means that only the information explicitly requested by the customer will be displayed on the frontend. Any subsidiary information called by the backend server to respond to the client request will not be displayed.
- *A REST API is client-server oriented.* This type of API is based on the principle of multi-threaded sockets, which means a single server responds to requests from several clients. This architecture allows the server to be the only SQL database client that contains all the data needed to run the application.
- *A REST API has a uniform interface and is layered.* A request will be made when the client has entered the correct URL. This URL will then be broken down according to the type of request made (GET, POST, PUT, DELETE). An HTTP request contains the following information:
 - A Header specifying the necessary permissions for the Frontend client to make the request to the Backend,
 - A Header specifying the type of data sent. In our case, for any request, we used the JSON format to send our data,
 - A Body with the data. For example: 'field: 'this is a data.

9.2.4 Drafting of the Backend API documentation

Before starting to code, it was necessary to explain how the Backend application would be designed. This fastidious but very important task makes it easier for anyone who will take over the application to become familiar with the issues of the application as well as the way it was designed. In this documentation, all the functions allowing the Frontend to make HTTP requests have been highlighted. It thus aims to highlight the GET, PUT, POST and DELETE requests that the Frontend will make according to possible Use Cases. Any subsidiary functions for CRUD queries are not included.

In our application, there are 3 major use cases.

- The user wishes to join a meeting: in this case, (s)he must get the list and information of the meetings to which (s)he has been invited using search filters.
- The user wishes to book a meeting: in this case, (s)he searches for a meeting room available according to the date and time at which (s)he wishes to book the meeting. Then, (s)he can reserve this room.
- Or a meeting is in progress and the user wishes to extend the duration of this meeting. Then, (s)he can make a request to know whether it is possible to extend the meeting for a specified duration.

Subsequently, as the application was being developed, the Backend and Frontend documentation was significantly modified, while keeping the same use cases.

9.3 Designing the app

9.3.1 Overview of a JavaScript web application

a) Components of a web architecture A web architecture is based on three different types of languages. These three languages are generally HTML, CSS and a web programming language such as JavaScript, Python (with Django), etc.

In our case, we decided to build our entire application with JavaScript, both for the Frontend and the Backend. The Frontend is based on the Vue.js framework, which is an overlay of Node.js, whilst the Backend is entirely coded in Node.js.

Node.js is a free software platform, based on the JavaScript language. It is a low-level environment that facilitates JavaScript execution for servers. Many frameworks such as Express.js, Vu.js, Angular.js, React.js and many others are derived from node.js.

b) A web architecture A web architecture is generally dynamic, since data processing is regularly refreshed depending on the requests made by the Backend. Typically, a website is organized as described in [C](#).

9.3.2 Architecture of the MariaDB database

In our project, we set up a relational SQL database which is composed of five data tables, see :

- Meeting, to manage meetings. A meeting is linked to a single room and has several clients.
- Survey, which was not implemented during my internship. This table should be used to get the users feedback for a given meeting in a given room. This should enable the general services to improve the room conditions so that people will no longer have problems with air conditioning, power supply or the Internet.
- MAClient, to manage Technicolor users.
- Room, to manage the meeting rooms. In a meeting room, several meetings take place, but a meeting room has only one "TimeKeeper" or MAIOT.
- MAIOT, to manage Time Keepers.

9.3.3 Architecture of the Backend[2]

The backend is similar to the submerged part of an iceberg. It allows the application to run without the user having direct access to it. The backend is used to send the Frontend the resources needed to operate the Frontend via HTTP requests such as GET, POST, PUT and DELETE.

To write backend servers, most developers use the Express.js framework. This framework is very simple to use since it allows an application variable that will be overloaded with all the methods to perform queries, called "routes", to be generated easily. In addition, this application can inherit methods from other modules. In 9.1, we can see the basic architecture of a Backend using the Express.js framework.

```
var express = require('express'); // import the express module

var bodyParser = require('body-parser'); // we import the body-parser module which automatically applies the JSON.parse method
on any object sent by the Frontend. This makes it possible to work automatically with parsed variables.

var app = express(); // application object is generated.

app.listen(3000); // application must listen on port 3000

console.log('Server listening on port 3000');
```

Figure 9.1: Example of a Backend Architecture Code

9.3.4 Architecture of the MQTT Broker

The main function of the application is to allow its users to have a quick look at the elapsed time of a meeting. Our application must be able to promptly and easily warn users of the remaining time of a meeting. To do this and in accordance with the desire to not have to set up a chat server, which is expensive in terms of disk space, we decided to use the MQTT protocol.

This protocol, specific to the IOT, allows messages to be sent and received with or without memory. It is based on the publish/subscribe principle and uses a broker to forward messages.

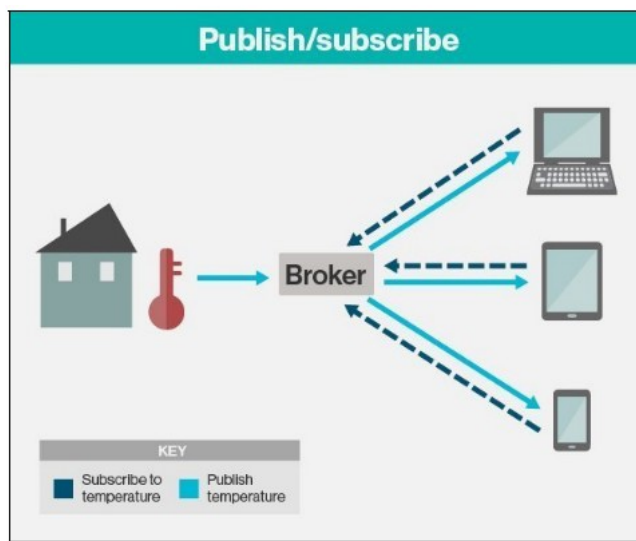


Figure 9.2: MQTT Architecture

An MQTT client, called a "publisher", will send a message to the MQTT broker in a defined topic. If other MQTT clients, called "subscribers", have subscribed to this topic, they will be notified of the message sent by the publisher. This protocol is based on the TCP/IP communication protocol but uses much less data.

In our project, MQTT allowed us to inform clients during a meeting of the remaining time. For example, by notifying the meetings clients that there are only 5 minutes left before the end. In addition, this technology was used to inform who joined or did not join a given meeting.

9.3.5 Docker container deployment[3]

To ease the deployment of the application on a server, we were advised to use Docker technology.

Docker automates the deployment of applications in software containers. It allows to "freeze" the state of an application at a given time. To do this, Docker consists of containers, which are closed systems in which virtual machines are running. These virtual machines are like Debian operating systems, only they run in a simplified version, i.e. not all the packages needed to start the operating system are installed and it does not have a running operating system.

To create our application, we used the Docker-compose tool from Docker which allows you to manage the links in a group of containers. In the first stage, these containers were deployed under localhost, then, after having built a bridge between the Docker Ethernet interface and the network, we deployed them with our own IP address.

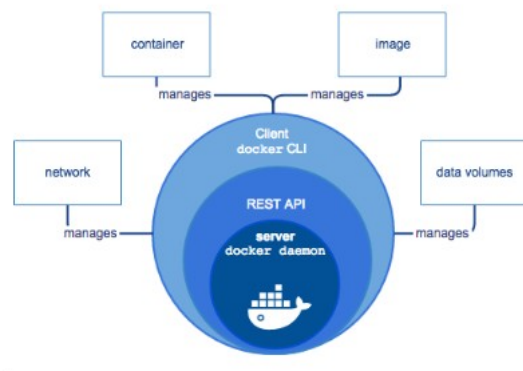


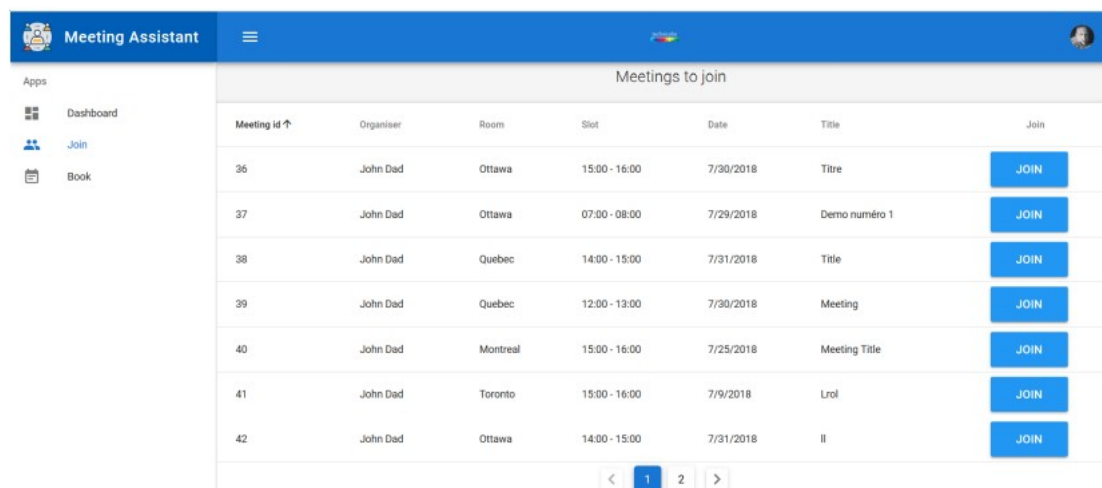
Figure 9.3: Docker Architecture

9.3.6 Overall operation of the application in relation to the goals of the internship

To explain the overall operation of the application, we will take a hypothetical use case. Lets take a Technicolor employee who wants to join a meeting.

The user will then connect to the application with his/her Technicolor credentials. If the user has entered the correct credentials. The Backend will retrieve their information from Technicolor's Active Directory LDAP, which provides a unique database that maintains Technicolor user credentials for all Technicolor applications. This user will then be assigned an authentication token that would be sent within each Frontend request. This is how the Backend will be able to authenticate the client by querying the MariaDB database.

From this token, the Backend can retrieve the user's email and when the user searches for current or upcoming meetings, the Frontend will only display the meetings to which the user has been invited.



The screenshot shows a web application interface titled "Meeting Assistant". On the left, there is a sidebar menu with options: Dashboard, Join, and Book. The main content area is titled "Meetings to join" and displays a table with the following data:

Meeting id ↑	Organiser	Room	Slot	Date	Title	Join
36	John Dad	Ottawa	15:00 - 16:00	7/30/2018	Titre	JOIN
37	John Dad	Ottawa	07:00 - 08:00	7/29/2018	Demo numéro 1	JOIN
38	John Dad	Quebec	14:00 - 15:00	7/31/2018	Titre	JOIN
39	John Dad	Quebec	12:00 - 13:00	7/30/2018	Meeting	JOIN
40	John Dad	Montreal	15:00 - 16:00	7/25/2018	Meeting Title	JOIN
41	John Dad	Toronto	15:00 - 16:00	7/9/2018	L'rol	JOIN
42	John Dad	Ottawa	14:00 - 15:00	7/31/2018	Il	JOIN

At the bottom of the table, there are navigation controls: a left arrow, a page number "1" (highlighted), a page number "2", and a right arrow.

Figure 9.4: Screenshot of the Join Meeting webpage

The user can therefore select one of these meetings and join it. Once in the meeting, (s)he will be informed if other people join the meeting through the MQTT Broker. In addition, as time goes by, any person who has joined the meeting will be informed of the remaining time 15 and 5 minutes before the end of the meeting.

Once the meeting is over, all MAClients who were attendees in the meeting will be deleted. Only the meeting will remain in the database. Any message sent by the Backend to the Broker will not be saved, due to the message sending mode (QoS 0, which means that messages are sent once, but the Broker does not ensure that the subscribers receive the messages sent).

9.4 Demos of the app

During the internship, three demonstrations were organised. The first mainly concerned the graphic interface of the Frontend to match the design potential of the "Time Keeper", which had not yet been redesigned at the time of the internship, since the trainee in charge of this part was not here yet.

The second concerned the integration between the Backend and the Frontend to ensure that the features required by the use cases have been provided.

The third and last demonstration was intended to show that the application fulfilled the specifications and was also an opportunity to suggest possible ways of improving and optimizing the application as well as to carry out an overall review of the internship.

9.4.1 First demo

The first demonstration was an opportunity to briefly show the Backends features: this implies the ability to retrieve the right information from the database, to format it correctly so that the Frontend can retrieve and display its information. However, this demonstration was firstly a means of validating the graphic presentation used to create the Frontend and to ensure that all the features defined in the Use Cases were covered.

9.4.2 Second demo

The second demonstration consisted of highlighting the integration between the Backend and the Frontend. To demonstrate the proper operation of the application, the database was populated with fake meeting rooms, fake meetings and fake users. This demonstration was mainly used to establish the action plan to be carried out for the end of the course, which included the integration with Outlook, to allow the preparation of the Backend to connect it with the future software "Time Keeper".

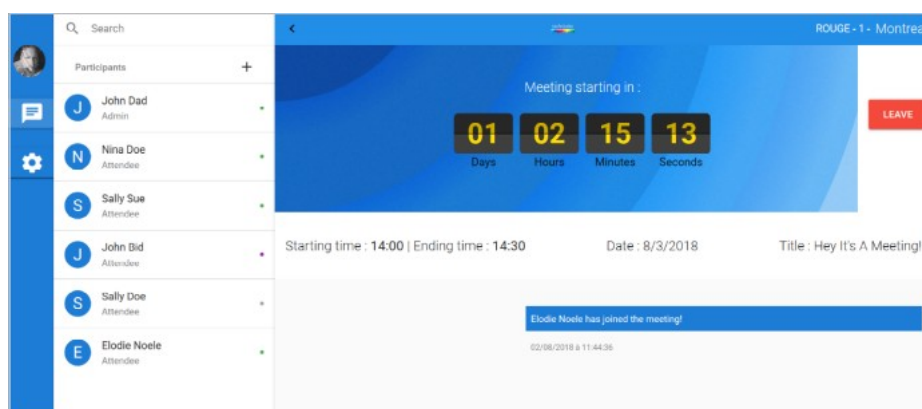


Figure 9.5: Screenshot of the Meeting Window webpage

9.4.3 Third demo

The third demonstration was a POC of the project to ensure that the specifications had been met. During this demonstration, the future of the application was decided, in particular, to know how the next trainee, who will take care of the MAIOT software, will have to interface with the Backend.

9.5 Testing the application

During the last two weeks of my internship, I had the opportunity to submit my application for user feedback to evaluate different aspects of the application such as ergonomics, aesthetics, speed of use and other factors.

In order to manage possible bugs and improvement tracks, I was led to use the JIRA collaborative problem management tool, allowing each user to open "tickets" explaining their problem and allowing them to suggest solutions or see its progress according to a defined timeline (Not Started, In Progress, Completed)

Finally, in order to forward the application to Technicolor employees, I created the following documents:

- Backend Documentation
- Project Implementation Documentation
- Test documentation

The latter documentation concerned written test procedures to ensure that the application was functional. So, if someone wants to take the code, he will have a reliable tool to know if the application is still functional or not and if not, to know where the bug is.

To do this, I used the Postman application which allowed me to make HTTP requests on the Backend to make sure that each route of the Backend worked and to get back the status code of the request as well as a possible error message.

In addition, the application backs up daily log files containing 4 levels of information:

- info: for information purposes only. Example : 'Add a new user'.
- warn: for preventive information. Example : 'Depreciation of a possible module'.
- error : for fatal errors.
- debug : for debug information

Conclusion

This internship was an opportunity to acquire an in-depth knowledge of the web development application and IOT objects, to have a thorough vision of a large company, to assimilate development processes, to experience project management in challenging situations, to face and resolve blocking situations as well as to develop technical and human skills.

It was a real pleasure to work at TCH, as the environment is attractive and ripe to be productive. It allowed me to develop skills such as autonomy, teamwork and adaptability. Also, I could reinforce or acquire new technical skills through the use of Git, the learning of the JavaScript language and Docker technology. Even though I was able to set up a functional Backend, I wish I had went even further and attained all the goals of the project.

As a DGA engineer and in comparison with my initial year of human and military training, my internship at TCH enables me to perceive the constraints of civilian companies. This knowledge will be useful to me during my future positions at the DGA, especially in the context of project design between the DGA and defence industry companies.

For my future internship, I would rather focus on a sector which is closer to my future major which is robotics or embedded systems. However, the skills developed in this software engineering internship will be useful in robotics. In both cases the goal will be to interact with the physical world: in one case the robot, in the other case a human client.

API Application Programming Interface.

Backend Backbone of a web application in charge of communicating with the resource needed by the Frontend to meet customer needs. [2].

Broker MQTT Message storage server [5].

CRUD Create, Read, Update, Delete queries: basic architecture of a Backend application..

CSS Cascading Style Sheets.

Docker Software to automate the deployment of applications in software containers..

EBITDA Earnings Before Interest, Taxes, Depreciation, and Amortization.

Express.js Web application infrastructure for HTTP and middleware methods. [6] [7].

Frontend Part of the code received by the client's browser that allows the deployment of a web application's graphical interface.[8] [7].

HTTP(GET, POST, PUT, DELETE) HyperText Transfer Protocol, client server communication protocol allowing different types of requests..

IOT Internet Of Things.

IT Information Technology.

JavaScript Object-oriented web programming language [9] [10].

LDAP Lightweight Directory Access Protocol, TCP/IP-based protocol for querying directory services, i.e. databases containing user IDs..

MariaDB Database management system operating on the same principle as MySQL.

MQTT Message Queuing Telemetry Transport, publish-subscribe messaging protocol based on TCP/IP protocol.[5].

MySQL Relational database management system.

Node.js Software platform written in JavaScript and allowing easy writing of web applications.[11].

OTT Over The Top, Internet media delivery service without traditional network operator.

POC Proof Of Concept.

QoS Quality of Service..

QoS 0 The message is sent, we do not ensure that someone has received the message.

QoS 1 The message is sent only once..

QoS 2 The message is sent at least once.

Refactoring Improvement of the source code of an application to improve its structure, but without modifying its general behaviour..

REST REpresentational State Transfer.

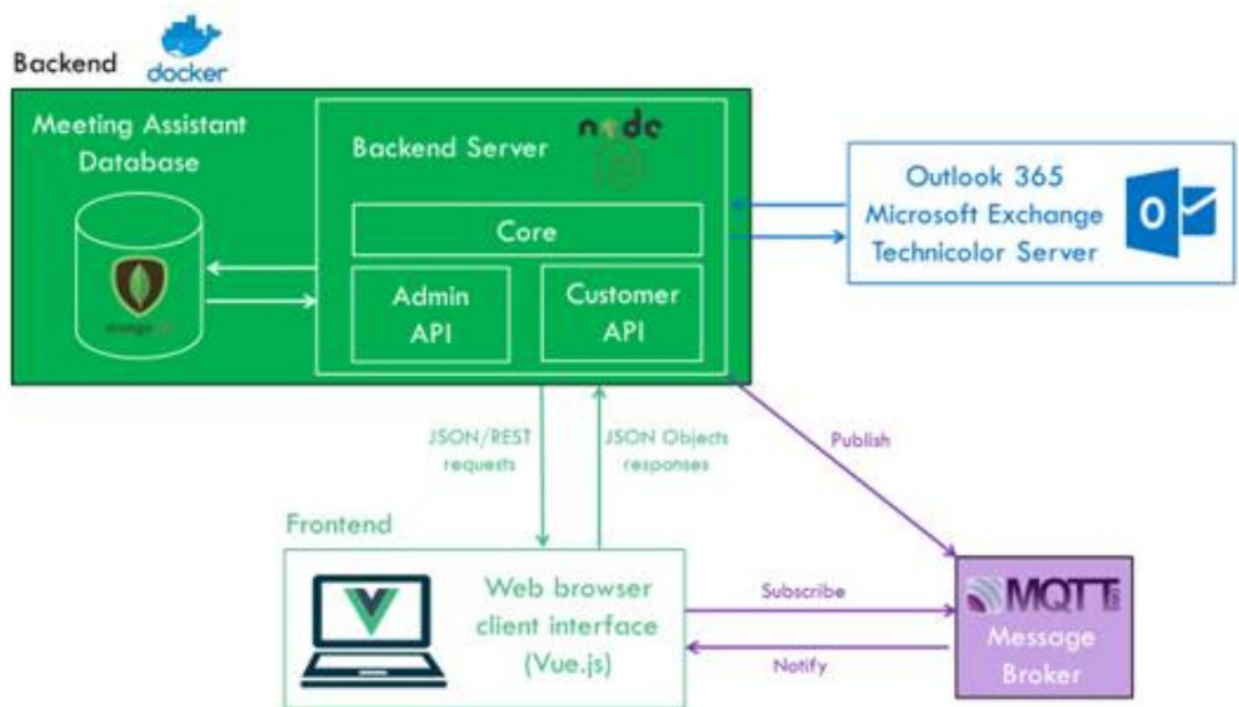
SourceTree GUI for Git collaborative tool management under Windows.

TCH Technicolor Connected Home.

UML Unified Modeling Language.

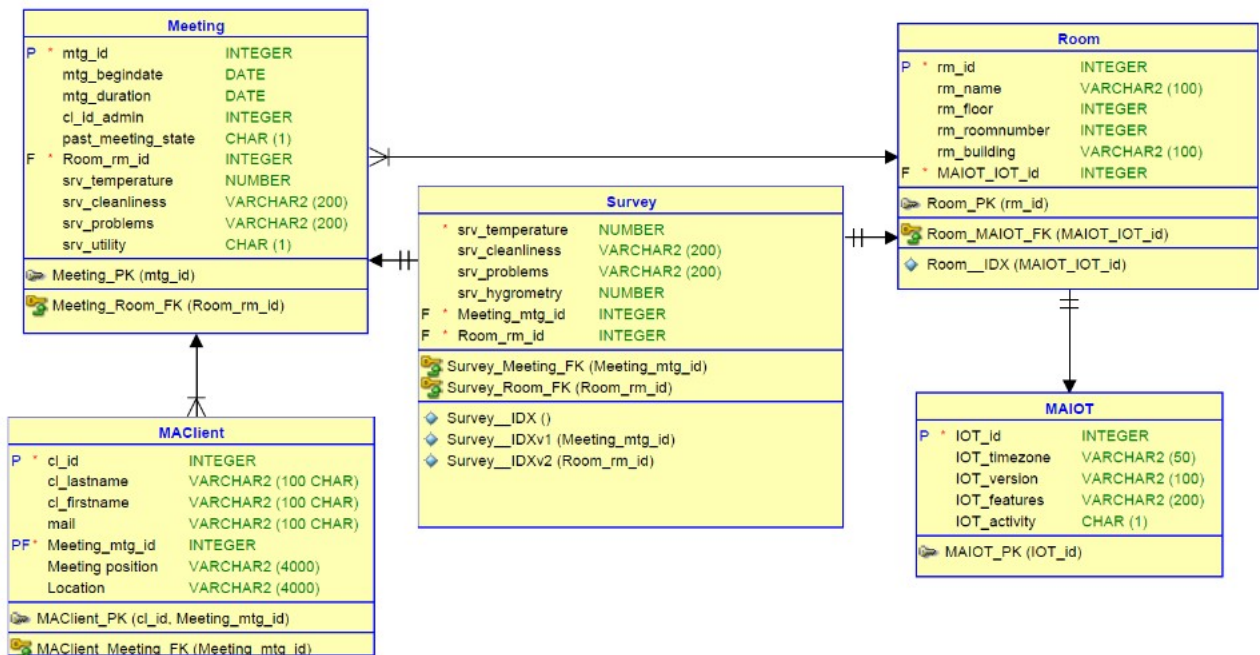
APPENDIX A

Global overview of the Meeting Assistant Application



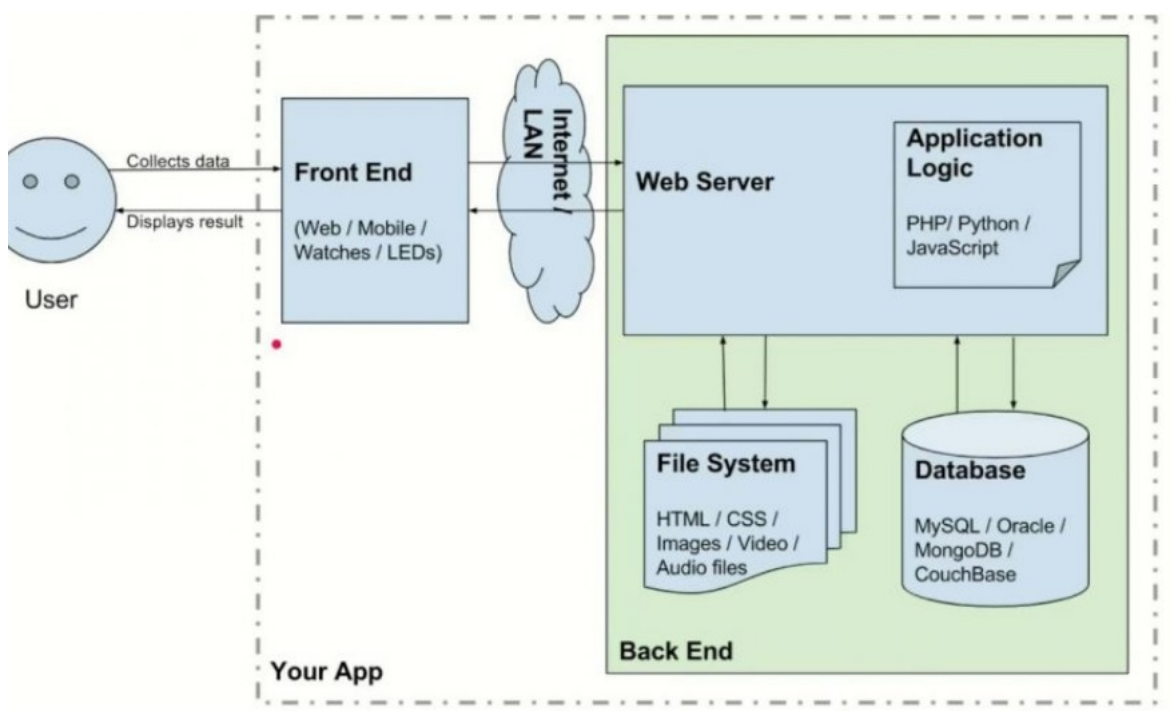
APPENDIX B

UML diagram of the Meeting Assistant Database



APPENDIX C

Architecture of a web application



List of Figures

5.1	Technicolor global overview [4]	6
5.2	Technicolor in figures	9
5.3	Technicolor Rennes Campus	9
6.1	Economic figures of Technicolor for 2017-2018	10
7.1	Initial prototype (left) and current prototype (right) of the Time Keeper	12
9.1	Example of a Backend Architecture Code	23
9.2	MQTT Architecture	24
9.3	Docker Architecture	25
9.4	Screenshot of the Join Meeting webpage	26
9.5	Screenshot of the Meeting Window webpage	28

Bibliography

- [1] OpenClassroom. Api rest. [Online]. Available: <https://openclassrooms.com/fr/courses/3449001-utilisez-des-api-rest-dans-vos-projets-web/3501901-pourquoi-rest>
- [2] M. Rousse. Back-end. [Online]. Available: <https://whatis.techtarget.com/fr/definition/Back-end>
- [3] Wikipédia. Docker. [Online]. Available: [https://fr.wikipedia.org/wiki/Docker_\(logiciel\)](https://fr.wikipedia.org/wiki/Docker_(logiciel))
- [4] Technicolor. Technicolor. [Online]. Available: <https://www.technicolor.com/>
- [5] Wikipédia. Mqtt. [Online]. Available: <https://fr.wikipedia.org/wiki/MQTT>
- [6] M. Firefox. Express. [Online]. Available: https://developer.mozilla.org/en-US/docs/Learn/Server-side/Express_Nodejs/Introduction
- [7] D. Lartet. Définition et avantages d'un framework web. [Online]. Available: <https://larlet.fr/david/biologeek/archives/20070117-definition-et-avantages-d-un-framework-web/>
- [8] Anon. Vue.js. [Online]. Available: <https://vuejs.org/>
- [9] Wikipédia. Javascript. [Online]. Available: <https://fr.wikipedia.org/wiki/JavaScript>
- [10] ——. Html. [Online]. Available: https://fr.wikipedia.org/wiki/Hypertext_Markup_Language
- [11] OpenClassroom. Node.js. [Online]. Available: <https://openclassrooms.com/courses/des-applications-ultra-rapides-avec-node-js/node-js-mais-a-quoi-ca-sert>