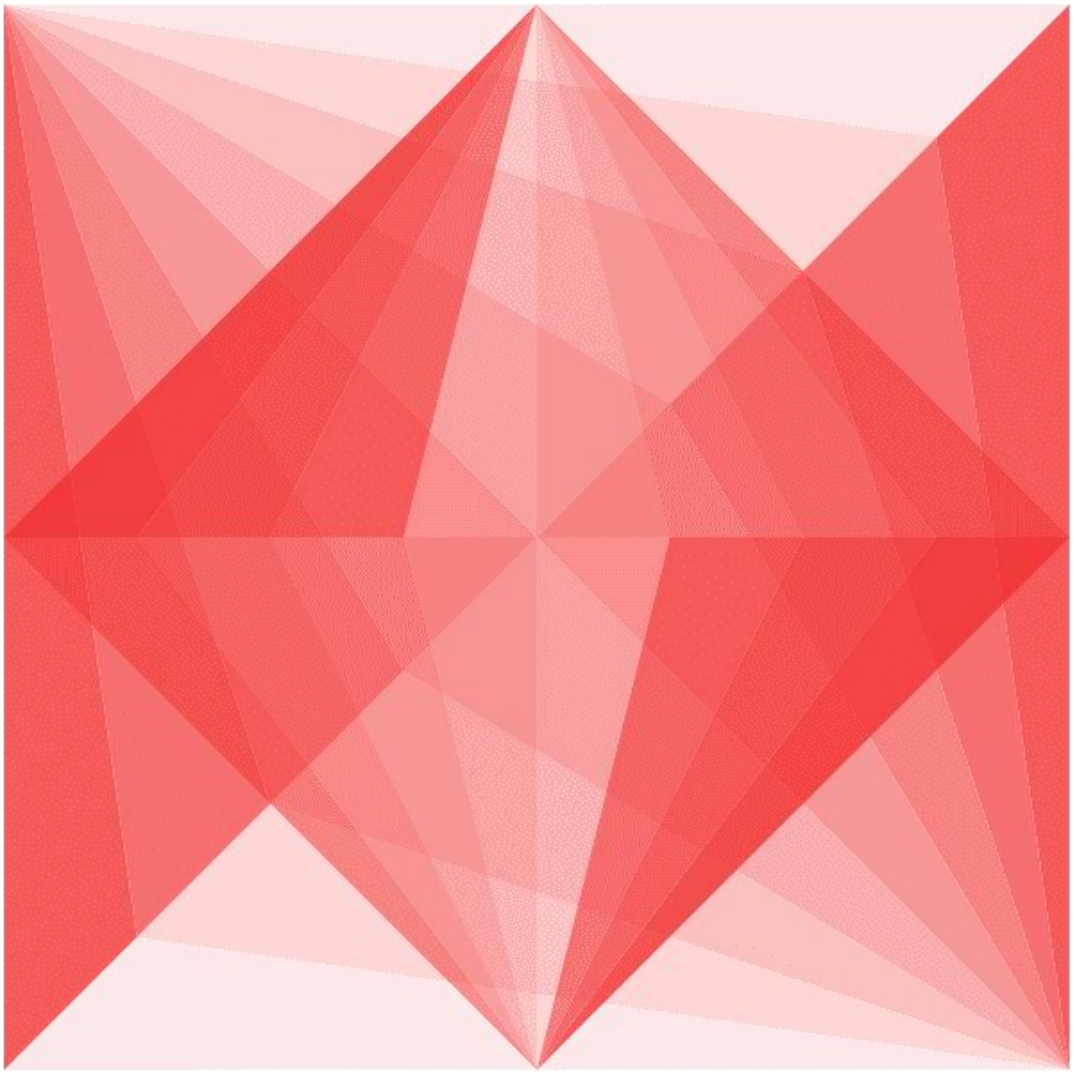


LIVE SMART LIVE CLEAN

Never Letting down

PROPOSAL FOR IR2

INTRODUCTION

IT companies are at the algid point of their development, technology is more accessible to everybody than ever and nobody contemplates a world without their smartphone, their high-tech oven or their household robot. 21st century has finally reached and is no longer something far that you are only able to watch on movies. Companies know that and right now is one of the most disputed markets. Because of that, if you want to become an over-the-top company, you must provide not only high-quality products, but be constantly developing new solutions to face the new client's requirements.

We consider IR2 has the potential to reach higher market quotes and become one of the most knowable IT companies in the mid-term future. To do it, IR2 must put an effort on their Research & Development department. Right now, the unavailability of reaching appropriately to customer feedback is harming the potential of the company, putting it in an unfavorable position for the near future that has to be corrected. That´s why the solution offered by ItalIT is perfect, because we provide a design that covers very mention area, what will lead to place IR2 in the first positions in the map. Our solution is aimed to:

- Promote the possibility of IR2 of using the reviews and feedbacks of the customers
- Improve the product development process having access to specialized data to make appropriate changes
- Create a platform that can be able to integrate and manage big loads of data coming from multiple sources

We will use our expertise in this area to offer a solution designed especially for you using only the best features in the current market, always aimed to offer IR2 and subsequently your clients the best experience possible.

Eventually, we also guarantee an opened communication channel with our support department, as it is one of the main stamps of our company, indefinitely. In addition, if we end up by having strong business relation, we will grant you access to our Data Mining & Machine Learning department, which will give you the features of how the proposed solution is developing in relation to the expectations generated and the possible solutions to hypothetical scope problems.

PROJECT SPECIFICATIONS

This project aims to provide the IR2 company a software application through which it will be possible to gather all the information related to the company coming from different sources, thus allowing to drive the development of the products currently researched in a more strategic way and with inputs coming both from the R&D department than from the real world.
In particular, we highlight, for each source, the type of data we collect.

SOURCE	DATA
WEBSITE	Customer reviews and feedbacks
MOBILE APP	Customer reviews and feedbacks
SOCIAL NETWORKS	Customer reviews and feedbacks on own social pages, additional data from user profiles mentioning the company
INTERNAL REPORTING	R&D data, tagged with the activities it belongs to and their status (running smoothly, if problems have arisen etc.)
INTERNAL DOCUMENTATION	R&D datasheets

The application will then provide a means of accessing the collected data in a useful manner.

- Feedbacks will be analyzed by a predictive model that will extrapolate the mood of the customers who left them: this will allow the sorting between positive and negative user reviews and will result, consequently, in better addressing the problems. Additional sorting by date and product type will be possible.
- Research & Development department information will be easily accessible by the project managers and the top management and the possibility of setting up some alerts will be provided: these could be triggered by problems arising during the activities.
- Through the analysis of the data, a forecasting system able to estimate the approval rate of the products will be available to use.
- R&D datasheet will be searchable through a centralized interface so that the gained knowledge is shared among the teams.

Employees will be able to access the data through a dashboard which will be implemented using the latest web standards, it will be responsive, user-friendly (our UX engineers will take care of that) and its compatibility will be ensured with both computer and mobile browsers. Even if the latter is not required, we think that accessibility on-the-go is a must-have in our time and since its implementation is not a significant overhead for us, we choose to add the functionality nonetheless.
The information requested by the employees will be displayed, depending on its type, with histograms, tag clouds and pie charts. Particular requests to add new data visualization tools can be done at any time.

The storage platform will not have problems with the management of the collected data and the integration of all the architecture into IR2 systems will be seamless.

All the proposed requirements will be fulfilled.

VENDOR SELECTION

As the requirements of the solution are really extensive, our team has developed a schema in which 4 main areas are identified and, subsequently, each one has to be covered via the usage of a different tool. Because of that, we have decided to divide it into the next areas.

Big data platform

To satisfy the needs of IR², which were described before, a big data platform is necessary so the data can be collected from different sources. Because providing those platforms are not our core activities and we also not have the know-how of doing this we decided to use an external one. Because of this we can develop a more flexible, cheaper and also robust solution for IR².

There are existing multiple big data platform solutions. To find the most fitting one a shortlist has to be developed, because thereby it is possible to compare the best possible solutions each other. The developed shortlist contains MongoDB, CouchDB and ElasticStack because all these possibilities are open source platforms and can be compliant with GDPR as well.

Despite these properties the platform has to have low overall costs, has to be compatible with Azure, has to be flexible and secure in relation to data losses as well. It's also helpful if our company have already worked with the platform. Wherefore these are the five criterions of the shortlist. They are weighted as follow because in our opinion it's most important that the data are safe, and that the platform can be integrated in the existing software:

- Overall costs: 0,2
- Compatible with Azure: 0,3
- Experience with working with the SW: 0,1
- Secure: 0,3
- Flexibility: 0,1

All possibilities are open source, that's why they are all cheap, but the effort to adapting the MongoDB platform is 10 % more than CouchDB and 20 % more than ElasticStack. In contrast to this the effort to integrate MongoDB in Azure is less than integrating ElasticStack (four working hours more) or CouchDB (5 working hours more). In addition to this our company has worked with MongoDB a couple of times (~150 times) with ElasticStack just approximately 30 times and with CouchDB just one time. The reason for this is, that MongoDB has a very robust code so it's very safe against data losses. Even so the other two solutions have a robust code as well, it's not that robust like Mongo DBs'. Last but not least MongoDB is very flexible because it's highly scalable. Because of these characteristics the result of the ranking is the following.

	Weight	***MongoDB***		***Elastic Stack***		***CouchDB***	
		Rating	Weighted	Rating	Weighted	Rating	Weighted
Cost/price (e.g. time, licenses)	0,2	8	1,6	10	2	9	1,8
Compatible with Azure	0,3	10	3	8	2,4	7	2,1
Experience with working with the SW	0,1	9	0,7	5	0,5	1	0,1
Secure (in relation to data losses)	0,3	9	2,7	7	2,1	7	2,1
Flexibility	0,1	9	0,9	7	0,7	5	0,5
Overall mark:			**8,9**		7,7		6,6

Since MongoDB is the winner with a result of 8,9 points (out of possible 10 points) the offered customized solution is based on MongoDB.

Infrastructure

Because IR² already have an existing infrastructure with Azure, which is well proven, it's also the solution which our company suggest. Nevertheless, it's important to offer the most suitable version of Azure depending on IR²'s need. That's why the version GRS, LRS and RA-GRS are compared in relation to cost, quality of data transmission, secure and a low error rate. In this case the cost (weight = 0,3) but also the quality of data transmission (weight 0,4) is very important for the choice. The highest price has RA-GRS with 245 €, the price of GRS is 50 € lower and the cheapest possibility is LRF with 98 €. In conflict to this, RA-GRS has the best redundancy for what reason it has the highest quality of data transmission and the lowest error rate as well:

	weight	***GRS***		***LRS***		***RA-GRS***	
		Rating	Points	Rating	Points	Rating	Points
Cost/price (e.g. licenses)	0,3	4	1,5	9	2,7	2	0,6
Quality of data transmission	0,4	6	2,4	3	1,2	9	3,6
Secure (in relation to data losses)	0,1	6	0,6	6	0,6	6	0,6
low error rate	0,2	6	1,2	5	1	9	1,8
Overall mark:			5,7		5,5		**6,6**

Because the quality of our offer is for us very crucial the winner of this ranking is the version RA-GRS of Azure.

Dashboard

Because the designing of the dashboard based on the customer needs is one of our core activities and we have specialized software engineers which have regular training so that they understand the customer need in detail ItalIT develops the dashboard by itself. But depending on the needs a different language is to be chosen. That's why a comparison between the three most fitting languages is to be done. Since IR² needs a very interactive dashboard is the possibility of a language to design an interactive application the most crucial criterion (weight = 0,25) in the following ranking:

	weight	***ReactJS***		***AngularJS***		***HTML+CSS+PHP***	
		Rating	Points	Rating	Points	Rating	Points
Experience with working with the language	0,20	5	1	8	1,6	9	1,8
Adaptability	0,15	7	1,05	7	1,05	6	0,9
Easy to learn the language	0,05	6	0,3	7	0,35	8	0,4
Easy to customize the SW	0,15	7	1,05	8	1,2	7	1,05
Possibility for interactive applications	0,25	9	2,25	6	1,5	3	0,75
Easy to design in a team	0,05	9	0,45	7	0,35	6	0,3
Less constraints in designing	0,15	8	1,2	7	1,05	8	1,2
Overall mark:			**7,3**		7,1		6,4

The difference of the final score between ReactJS, AngularJS and HTML+CSS+PHP is not large but significant why ReactJS is chosen. It won the comparison even all languages are pretty similar and the workers have less experience in working with it since it has the best possibilities for designing interactive applications. In addition to that the choice of ReactJS is suitable for us because it's becoming more and more the language of the market and hence our aim is to get more experience with it.

Legal checkup

Because our company has no legal department but we want to guarantee that our solutions are compliant with the GDPR so IR² can fully trust our offer, our solutions is proved in relation to GDPR externally by a consultancy. Even so the price of the consultancy is very important because we want to stay in the desired budget (weight = 0,3) in this case is for us, the quality of the outcome more crucial (weight: 0,4):

	weight	***DPO-consulting***		***DPOCC***		***Iubenda***	
		Rating	Weighted	Rating	Weighted	Rating	Weighted
Cost/price	0,3	7	2,1	4	1,2	6	1,8
good relationship/experience with the company	0,15	5	0,75	5	0,75	9	1,35
flexibility	0,1	4	0,4	6	0,6	4	0,4
General customer satisfaction	0,05	8	0,4	9	0,45	7	0,35
Quality of the outcome	0,4	6	2,4	8	3,2	8	3,2
Overall mark:			6,05		6,2		**7,1**

Even so Iubenda is a bit more expensive than DPO-consulting (about 200 €) it's the chosen consultancy because their outcome of them is always correct and detailed and we have also a good relationship to them based on former collaboration.

ARCHITECTURE

Microsoft Azure is the provider of the IaaS on top of which we will develop and install our software.

Considering that cloud computing and storage providers have similar costs when providing the same level of service and since IR2 is using and it is satisfied with Azure services, we choose to extend the already in-place infrastructure. The integration cost of a different IaaS would have easily outgrown the eventual price difference.

To satisfy the need for our application, a high availability server with some additional data storage is needed.

We identified the first as an instance A4 (8 core, 14 GB of RAM, 605 GB of temporary storage), while we deemed enough a 10 TB Data Lake account for the storage. They can be both easily scaled if the need arises.

As already discussed, the chosen big data platform is MongoDB which we will install on the server and will handle all the data coming to and from the database.

The SPD software, which we will develop, is divided into 3 modules: analytics, data collection and dashboard.

The data collection module will take care of getting all the data from the user reviews, the social pages of the company and the R&D department. In particular, the reviews left on the company's site will be easily collected from the servers, the ones left of social pages, together with comments and reactions to the company public communications, will be retrieved using the relevant social network APIs.

R&D department status and updates will be collected through a particular section of the module, which will offer the possibility to tag and sort the data so that it will be more easily manageable.

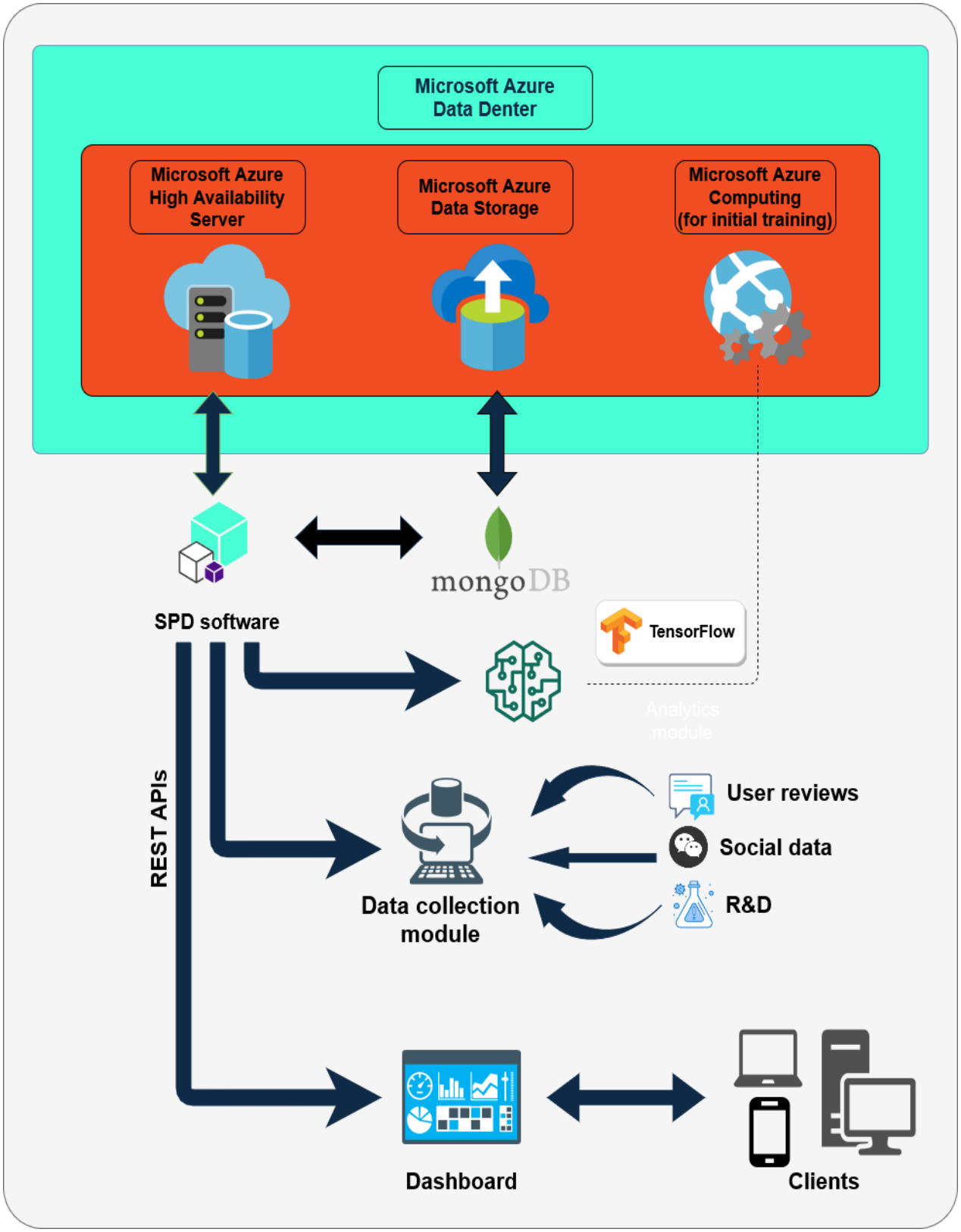
To achieve GDPR compliance, the SPD software will also aggregate the data so that it won't be possible to identify which user is responsible for a certain piece of information, apart from the case in which he has explicitly opted-in. Regarding this last type of user data we store, the user will have full control over them and could request their removal at any time.

The analytics module is the core of the architecture. From all the data we have collected, through some machine learning techniques, it will extract the relevant information for the company (see Project specification)

The last module of our software is the dashboard, which will expose to the company employees, in a clean and ordered way, all the data acquired and elaborated by the previous modules.

It will be developed using ReactJS, a state-of-the-art framework that will allow it to be completely responsive and usable from whatever device the users want. The communication between the dashboard and the SPD software, installed on the Azure instance, is achieved through the implementation of a REST APIs interface. This will allow a more rapid development since, from the moment they are defined, even if the relative functions are not implemented yet, the frontend developers can almost finalize their work: only minor adjustments and testing will be necessary until deployment can be done.

On the next page it can be seen a graph summarizing the designed architecture integrating all the features of the solution proposed.


Microsoft Azure
Data Denter
Microsoft Azure
High Availability
Server
Microsoft Azure
Data Storage
Microsoft Azure
Computing
(for initial training)
mongoDB
SPD software
TensorFlow
Analytics
module
REST APIs
User reviews
Social data
R&D
Data collection
module
Dashboard
Clients

USE CASE

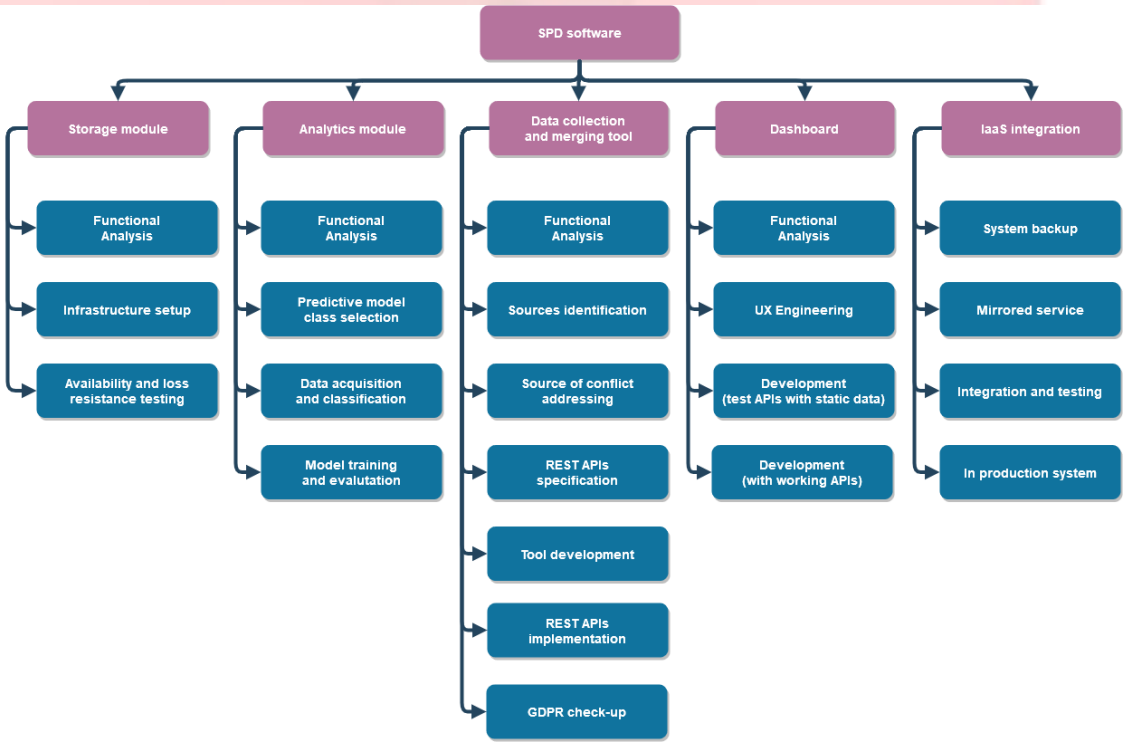
This fully customized software architecture leads to a development of the robots based on the customer's needs. Because of this the robots are fully customized why the satisfaction of the customers will be increased.

A development based on the customer needs is possible because the feedback of the customers is gathered, stored and analyzed by our suggested SPO-Software and it's presented afterwards to the R&D department by our intuitive designed dashboard. Hence the workers know exactly for which features they should increase and for which they should decline their effort and which features they may should eliminate. Even if they search for some special feedback, they can do it by keyword which is very fast and very efficient. Because of the software architecture, not only the exchange with the clients is increased but the interchange and communication within the R&D department and the whole organization as well. This leads also to a better development of the products and again consequently to a higher customer satisfaction.

But the architecture helps not only the R&D department but the sales and marketing department as well. This is firstly because our solution analyses the customers' needs why they can adjust their marketing mix which includes the Promotion-Mix and secondly the customers are segmented automatically why they can target the most profitable customers easily. This will lead to a more market-oriented strategies that can focus on those segments that will provide more benefits to the company or the ones that are in risk to fall in a decline area so the company will be able to decide if more efforts are needed or, otherwise, abandon that segment in order to the lack of positive future previsions

Taking everything into account, the proposed architecture enables both fully customized marketing activities and a development, which generate the most fitting product in relation to customer's needs, which both leads to higher sales and happier customers as well.

WBS



Given the complexity of the project, a functional analysis is a required initial step for all the different modules.

Storage-wise, we first have to set up the infrastructure and then test if it is working properly.

Concerning the analytics module, we have to be sure to use the right machine learning algorithm that best fits our data: an accurate selection of the predictive model to be used is conducted, data is then acquired and classified, and lastly the artificial intelligence model is trained and its results evaluated. It has to be noted that some of these activities don't require employees to work constantly. In particular, the last one should be done with little to no effort since there are huge waits for the compute nodes to elaborate the data.

The data collection and merging tool is the most significant part of the project: gathering the data from the different sources is not an easy task, conflicts will arise and they must be programmatically solved to avoid downtime when the software is in production. Since it is the module responsible to provide all the data to the dashboard, the development of the latter, at some point, will halt and will wait for all the functionalities to be implemented. To ensure the GDPR compliance, this module has to reviewed by an external legal consulting agency. Even if the check-up will not take much time per se, the overall activity is going to take 2 weeks. This is because it is outsourced and our vendor agreed to deliver in this timespan.

The internal communication of the modules is achieved through the usage of some REST APIs endpoints: to speed up the dashboard development, we will determine as quickly as possible their definition.

We want the user experience to be as pleasant as possible so that the employees will not get frustrated using our software: apart from the initial functional analysis, a phase of UX engineering is conducted by our experts which will outline the main views the dashboard will have.

The actual development of the dashboard can start as soon as the REST APIs specification are written down. This should not be a problem if we follow our schedule, however, at a certain point, the actual implementation will be needed.

When all the modules are finished, we have to make sure the system is stable and that we won't incur in any data losses during the integration with the already existent IR2 infrastructure: we make sure to back up the system and mirror the services on different servers, then the integration tests are conducted and, finally, the system with our integrated software can be deployed in production.

SCHEDULE

The project is set to start on 22 June 2020.

The full system design specification will be ready as soon as all the functional analysis will be completed, 25 June (deadline 29 June)

The full system architecture requires additional time since we first need to identify all the possible sources of data, 30 June (deadline 6 July)

The dashboard mock-ups and its design will be ready as soon as the UX Engineering complete their task, 6 July (deadline 13 July)

The first software demo will require the infrastructure to be up and running, some data acquired and analyzed and the dashboard to be usable, at least using some dummy data. It will be possible by 20 July (deadline 20 July).

It has to be noted that tests are performed during all the activities and at the end of the development (together with integration tests).

After the deadline for the first demo, for which we have no slack time, the successive ones are not a problem.

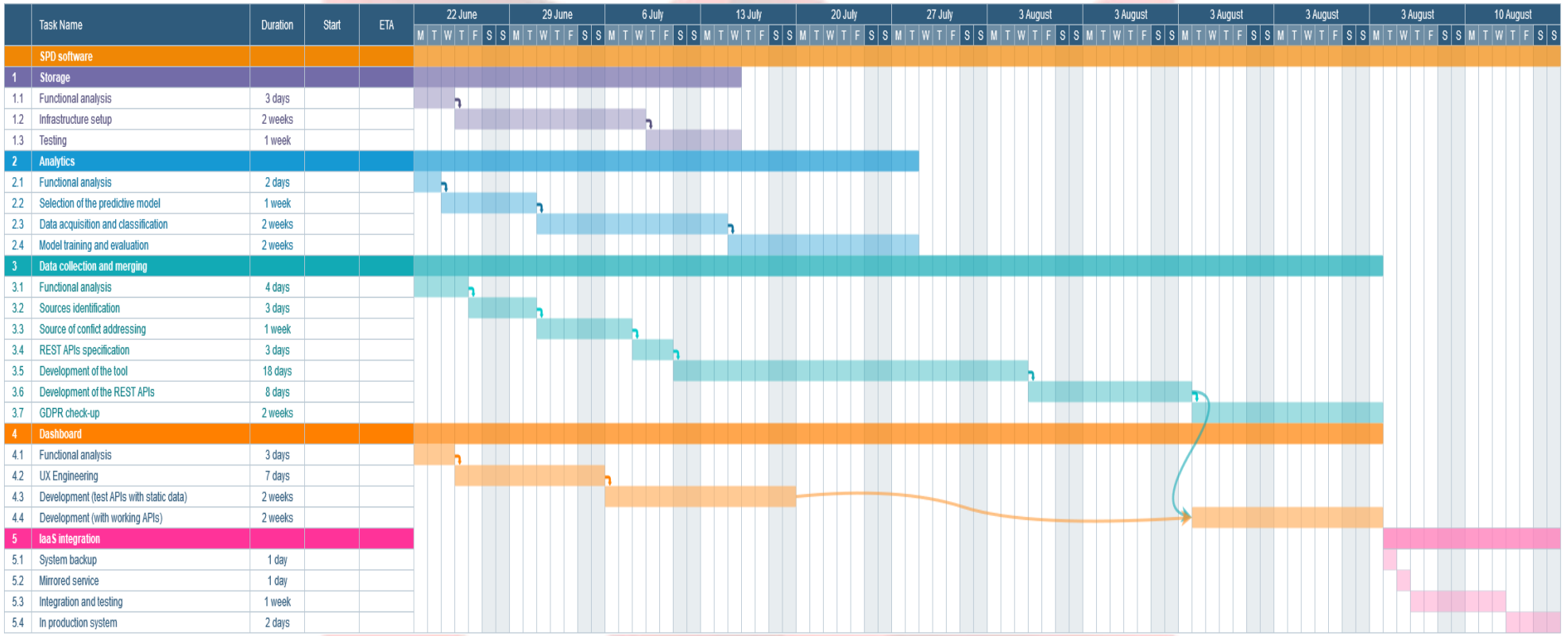
We plan to have the predictive model trained and working by the 29 July, which concludes the analytics module.

The development of the data collection and merging tool, the most demanding activity of the project is estimated to take a little more than 3 weeks. The successive REST APIs implementation will allow continuing the development of the dashboard.

When this last activity is concluded and the consulting agency has emitted its verdict on the GDPR compliance, we can integrate our software with the already existing architecture.

In the figure, we show our implementation plan if the aim is to deliver the project as soon as possible. However, since time could not be a concern for IR2, we would like some slack time to better manage our resources.

On the next page it can be seen an image that summarizes the main points of the explained scheduling.



	Task Name	Duration	Start	ETA
	SPD software			
1	Storage			
1.1	Functional analysis	3 days		
1.2	Infrastructure setup	2 weeks		
1.3	Testing	1 week		
2	Analytics			
2.1	Functional analysis	2 days		
2.2	Selection of the predictive model	1 week		
2.3	Data acquisition and classification	2 weeks		
2.4	Model training and evaluation	2 weeks		
3	Data collection and merging			
3.1	Functional analysis	4 days		
3.2	Sources identification	3 days		
3.3	Source of conflict addressing	1 week		
3.4	REST APIs specification	3 days		
3.5	Development of the tool	18 days		
3.6	Development of the REST APIs	8 days		
3.7	GDPR check-up	2 weeks		
4	Dashboard			
4.1	Functional analysis	3 days		
4.2	UX Engineering	7 days		
4.3	Development (test APIs with static data)	2 weeks		
4.4	Development (with working APIs)	2 weeks		
5	IaaS integration			
5.1	System backup	1 day		
5.2	Mirrored service	1 day		
5.3	Integration and testing	1 week		
5.4	In production system	2 days		

RESOURCES NEEDED

The resources needed are listed next:

- STORAGE MODULE
  - Functional analysis
    - 1x Senior software architect
  - Infrastructure setup
    - 1x Senior network engineer
    - 1x Junior network engineer
  - Availability and loss resistance testing
    - 1x Junior network engineer
- ANALYTICS MODULE
  - Functional analysis
    - 1x Business Analyst
    - 1x Senior software architect
    - 1x Senior data scientist
  - Selection of the predictive model
    - 1x Senior data scientist
    - 1x Junior data scientist
  - Data acquisition and classification
    - 1x Junior data scientist
    - 1x Senior data scientist supervising, 1-2h/working day
  - Model training and evaluation
    - First two days:
      - 1x Senior data scientist
      - 1x Junior data scientist
    - After the first two days till the end:
      - 1x Junior data scientist (sporadically checking the progress, 1-2h/working day)
- DATA COLLECTION AND MERGING TOOL
  - Functional analysis
    - 1x Senior software architect
    - 1x Junior software architect
    - 1x Senior data scientist
  - Sources identification
    - 1x Senior data scientist
  - Source of conflict addressing
    - 1x Senior data scientist
  - REST APIs specification
    - 1x Senior software architect
    - 1x Senior software developer
  - Development of the tool
    - 1x Senior software developer
    - 1x Junior software developer
  - Development of the REST APIs
    - 1x Senior software developer
    - 1x Junior software developer
  - GDPR check-up
    - External consultancy
- DASHBOARD

    - Functional analysis
        - 1x Senior software architect
        - 1x Senior UX Engineer
    - UX Engineering
        - 1x Senior UX Engineer
        - 1x Junior UX Engineer
    - Development (test APIs with static data)
        - 1x Senior frontend developer
        - 1x Junior frontend developer
    - Development (with working APIs)
        - 1x Senior frontend developer
        - 1x Junior frontend developer
- IaaS integration
    - System backup
        - 1x Senior network engineer
    - Mirrored service
        - 1x Senior network engineer
    - Integration and testing
        - 1x Senior network engineer
    - In-production system
        - 1x Senior network engineer
        - 1x Senior software architect

It has to be noted that the cost of software testing is included in the development activity (unless otherwise specified) since we usually carry them together in a continuous manner.

To sum up, we need:

- 17d Senior software architect
- 4d Junior software architect
- 19d Senior network engineer
- 15d Junior network engineer
- 2d Business Analyst
- 23d Senior data scientist
- 19d Junior data scientist
- 29d Senior software developer
- 26d Junior software developer
- 10d Senior UX Engineer
- 7d Junior UX Engineer
- 20d Senior frontend developer
- 20d Junior frontend developer

To which we have to add the external legal consultancy.

FINANCIAL

Our solution includes a platform for storing big data and an analytics module to help the company analyze the feedback in a productive way. The costs consist of some fixed costs, such as labor or one-time payments, and also some monthly payments related to the requirements of the project which are provided by our partners or other companies.

Based on the architecture of this software we need to rent a VM to run our main software and it consists of monthly payments; a storage platform was also needed to fulfill the requirements of this software.

- Storage module:
  - Analyzing, setup and testing: 7000€
  - Infrastructure renting 390 €/monthly
- Analytics module:
  - renting 400 €
  - Applying 12000 €
- Data collection and Merging tool 15000 €
- Dashboard 13000€
- IaaS 5000 €
  - VM 440€/monthly

Total cost: 52400€ + 830€/monthly

Area	*Costs*
Salary	26000€
Overhead	2000€
Labor hours	1690
Material Costs	24400€ + 830€/month
Overall Costs	52400€ + 830€/month

SUMMARY

As final summary, the solution we have designed will provide your company with a high customized software that will solve every kind of problem related to the precedent unavailability of gathering correctly customer´s feedback and its correspondent Research & Development resources will start to be useful for the company.

The development of this software will be divided in 5 main areas: Storage, Analytics, Data collection and merging, Dashboard and IaaS integration, each with its own deliverables and sub-products, all together will form an integrated and user-friendly solution that will answer to all the requirements proposed

For our architecture we use MongoDB as our big data platform, because it's an open source application and it's very robust against data losses. We use Azure RA-GRS because IR² already use Azure and with the version of RA-GRS, we can offer the best quality in relation to data transmission. Third we develop the dashboard with ReactJS because in that way we can offer interactive an application, which is fully tailored on the needs of IR². Last but not least our solution is checked by Iubenda in relation if our solution is in agreement with GDPR because we gained excellent experience while collaboration with them.

The overall costs of the designed solution are 52400€ + 830€ monthly divided between Salaries, and overhead. We consider that the total amount of Labor hours will increase to around 1700h.