

PROCESS AND SERVICE DESIGN



POLIMED

PoliMed

Edoardo Longo
841677
edoardo.longo@mail.polimi.it

Alberto Marini
862838
alberto2.marini@mail.polimi.it

June 24th, 2016

Politecnico di Milano

Contents

1	Presentation	3
2	Service BluePrint	4
3	Assumption About The Underlying Infrastructure	5
3.1	Overview	5
3.2	ERP - Enterprise resource planning	5
3.3	DBMS - Database management system	6
3.4	CRM - Customer relationship management	6
3.5	PoliMed official website	6
3.6	Vans	7
3.6.1	Vans & temperature sensor	7
3.6.2	Vans & GPS	7
3.6.3	Vans & Node-RED	8
3.7	Warehouse & sensors	9
3.8	Warehouse	9
3.9	Sensors & cameras	9
4	Website Description	10
4.1	Homepage	10
4.2	Order	11
4.2.1	Medicine order example	12
4.3	Tracking	13
4.3.1	Result tracking page & Google Maps	14
4.4	Contacts	15
5	Design and Analysis	16
5.1	Business Process Model and Notation	16
5.1.1	User Action	16
5.1.2	Order Medicine	16
5.1.3	Warehouse	17
5.1.4	Create package	17
5.1.5	Tracking	17
5.1.6	Truck	17
5.1.7	Check sensors	17
5.2	Choreography	18
5.3	Petri Net	18

Chapter 1

1 Presentation

PoliMed is a company located in the north-west of Italy and has been operating for years in the pharmaceutical logistic field.

In particular, *PoliMed* deals with the final part of the process of distribution of pharmaceutical products to small and medium-sized pharmacies throughout the territory. The latter has the opportunity to interact with *PoliMed*, through a platform installed on their devices, able to place orders, cancel them and check the status. Lots of individual medicines are stored inside the company's logistics warehouse and whenever a pharmacy requires a certain medicine will place an order.

Nowadays, this kind of service has already been presented, but the goal is to make it more intelligent to monitor the temperature throughout the delivery, not to avoid the medicine properties, and at the same time the pharmacy has the possibility to see where the ordered product is in real-time. In addition, the company ensures the delivery of medicines within the specified date during ordering. The distribution of lots of medicines occurs on road, thanks to the special *PoliMed* vans equipped with innovative sensors and a continuous remote control, so as to meet the requirements expressed above.

Goals:

- Network distribution at controlled temperature (for example in a range between: -10°C and 30°C)
- Package tracking
- Same-day medicine delivery

Clients:

- Small & medium size pharmacies in Italy

Layers involved:

- Pharmacy
- *PoliMed* booking operator
- *PoliMed* warehouse
- Internal *PoliMed* couriers

Chapter 2

2 Service BluePrint

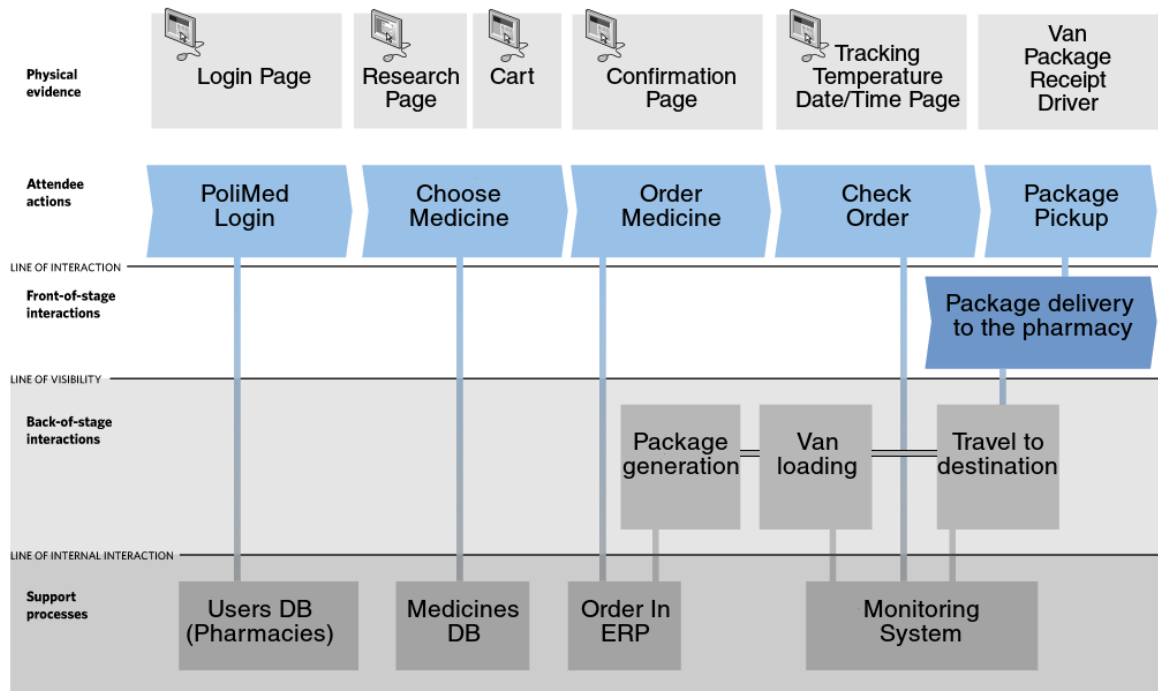


Figure 1: Service Blueprint chart

Blueprint description

Pharmacies will see a login page, from where they could login inside *PoliMed* platform.

At this point, they can research medicines and put them inside a cart. After the order confirmation page, *PoliMed* will get the order inside the ERP.

From that moment, *PoliMed* will create a package with the medicines ordered, load it on the van, then send it to the destination.

Pharmacies will be able to monitor the temperature and the GPS position, by using the dedicated page.

The delivery will be completed by the package pickup action.

Chapter 3

3 Assumption About The Underlying Infrastructure

3.1 Overview

In this document we are going to introduce the **infrastructure** that will let *PoliMed* handle the services about medicines shipment, check temperature and position. In particular, we are going to analyze our components of the underlying infrastructure:

- ERP
- DBMS
- CRM
- Website
- Van & sensors
- Warehouse

Note that **IBM Bluemix** platform is used for our website and services involved.

3.2 ERP - Enterprise resource planning

ERP software modules can help administrators in organization to monitor and manage supply chain, procurement, inventory, finance, projects, human resources and other mission-critical components of a business through a series of interconnected executive dashboards. Moreover, the ERP solution is contained in the operational layer of a *Service Oriented Architecture* (SOA).

Inside *PoliMed* we are going to use a **SAP ERP system**. Our ERP solution, named *SAP Business ByDesign* is a Cloud-based ERP for mid-market companies. It's a complete and integrated *SaaS* suite that supports financials, human resources, sales, procurement, customer service, supply chain management, and more.

3.3 DBMS - Database management system

The **database management system** is another important component inside the underlying infrastructure. For *PoliMed* needs, the decision is connected with the ERP solution on a **SQL relational database**. Also the DB is included in the **SOA Operational layer**.

The database will contain medicines data like:

- Name
- Typology
- Available Quantity
- Price

3.4 CRM - Customer relationship management

By choosing an ERP as the main component of our underlying infrastructure, we can also have a customer relationship management system. The **CRM** included in *SAP ByDesign* let us provide improved supports and services to our customers. Also, developing a CRM strategy is what will give us the long-term benefits.

Moreover, our CRM is a cloud-based system that allows us to access, organize, and analyse information about customers and sales made on our platform.

3.5 PoliMed official website

The *PoliMed* **official website** is the main entrance to our service. As technology is progressing faster and faster in modern world, a website is one of the most important elements for the companies. All pharmacies can use our website with the related services to create new orders, check tracking or send requests.

REST, JSP, J2EE & Java Beans

PoliMed official website offers a flexible platform for all pharmacies to use our services. In particular, the implementation of the website is based on **J2EE**. *PoliMed* official website is created by using **JavaBeans**, **JSP web pages** and **HTML pages**.

Important focus have been given to **REST** implementation (*Representational State Transfer*), a stateless communication protocol, client-server based on HTTP.

Thanks to REST, we can send server request to retrieve our resources, in particular the creation of the order and the tracking. The **POST** and **GET** methods permit us to create, modify and access all the resources in the database.

Google Maps APIs

Millions of websites and apps use *Google Maps APIs* to power location experiences for their users. Also *PoliMed* includes Google Maps APIs inside the website to shows the real-time tracking on a map, by using a GPS sensor inside our vans.

At the same time, Google Maps APIs offer flexibility and accuracy. Google Maps APIs are available for Android and iOS for future implementations.

3.6 Vans

Vans are an extremely important component to provide the best service quality. We decide to use *Volkswagen vans* which guarantee great reliability and comfort for the driver. *PoliMed* uses an official laboratory to keep each van under maintenance.

3.6.1 Vans & temperature sensor

Inside each van there are 6 temperature sensors: 2 on the trunk **bottom**, 2 close to the **doors** and two in the **front** part close to the cockpit. In this way, *PoliMed* can ensure that the temperature is uniform in each trunk part. At the same time, *PoliMed* uses 2 sensor for each zone, for redundancy reasons.

Moreover, all temperature sensors are calibrated at the factory, using high quality standards.

3.6.2 Vans & GPS

GPS is another important underlying component of our infrastructure. *PoliMed* and our customers can both track the vans via GPS, getting in real-time the position via the web platform because of *Google Maps APIs*.

3.6.3 Vans & Node-RED

The sensors management is entrusted to the NODE-RED¹, a modern *IoT system* that keeps track of the various temperatures measured by sending them to our platform.

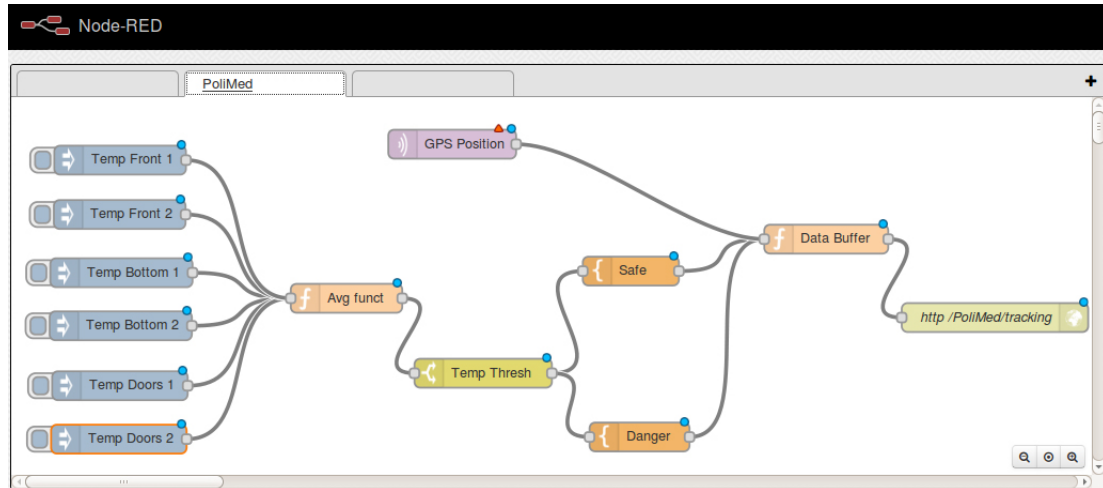


Figure 2: Node-RED chart

Figure 2 represents the flow of the data from the sensor to the HTTP request.

On the left there are the six sensors; after the average operation and the temperature checking, the data are sent to a buffer that addresses them to the system via HTTP. As you can see in Node-RED diagram, a similar path is done by the GPS data.

¹<http://nodered.org/>

3.7 Warehouse & sensors

3.8 Warehouse

Last but not the least, warehouse is the main place where *PoliMed* keeps medicines. Our warehouse has high level of automation. in particular, pallets and product move by using a system of automated conveyors, cranes and automated storage and retrieval systems coordinated by programmable logic controllers and computers running logistics automation software.

3.9 Sensors & cameras

PoliMed uses many types of sensors inside the warehouse. We have temperature sensors, to monitor the medicines temperature in real-time. At the same time, in the warehouse perimeter there are many security cameras, with movement detection system during night time.

Chapter 4

4 Website Description

PoliMed website is available on IBM BlueMix at <http://polimed.mybluemix.net/Polimed/>.

4.1 Homepage

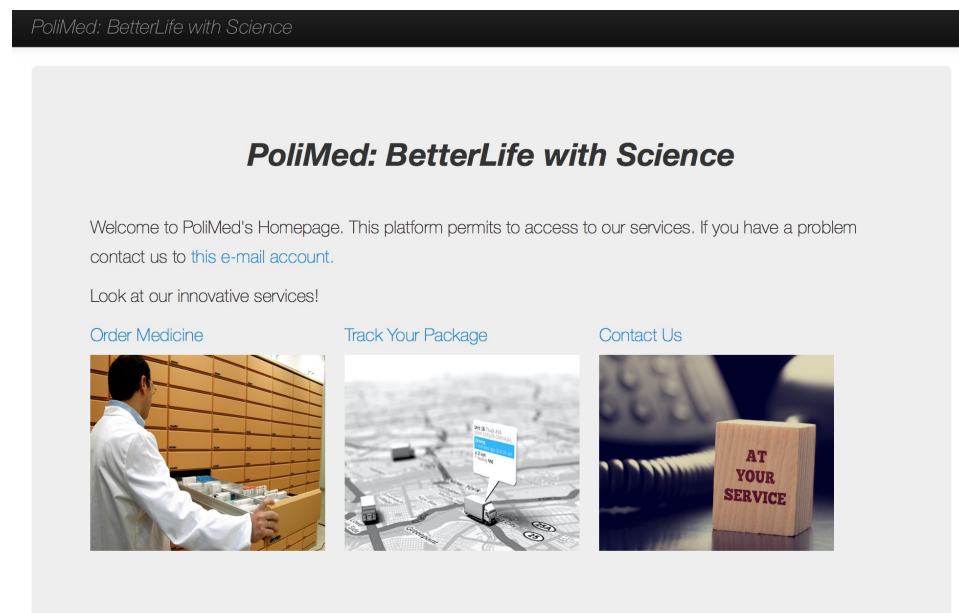


Figure 3: *PoliMed* homepage

PoliMed homepage


Once entering inside the *PoliMed* homepage, you can find three different kinds of services:

- **Order Medicine:** this page will let you create a new **order**, by specifying name and quantity.
- **Track Your Package:** this page will let you **track** the package, check the real-time **temperature** and also get a detailed map, implemented by using *Google APIs*.
- **Contact Us:** here you will find a form to fill for contacting *PoliMed* staffs.

Another way to contact *PoliMed* is to press the link named *this e-mail account*, which will let you send a direct email to our company: polimedpolimi@gmail.com.

4.2 Order

Order Medicine



Order the medicine you need

Insert the name of the medicine and the quantity you want to buy.

Reservation:

Medicine:

Quantity:

Figure 4: Order medicine page

Order Medicine

Inside this page you can insert the name of medicine and also the quantity. By pressing the *Submit* button, our system will immediately create an order connected to a tracking number.

Order the medicine you need

Insert the name of the medicine and the quantity you want to buy.

Reservation:

Medicine:

Quantity:

Figure 5: Medicine order example

4.2.1 Medicine order example

For example after inserting medicine and quantity, the JSP page sends the two parameters to the `@POST` method and creates a new order instance. After this operation the method stores the medicine order inside the database (in our case, for simplicity, an `ArrayList`) and it returns to the *Result Page*. The code below shows these operations:

```
1  @POST
2      @Produces(MediaType.TEXT_HTML)
3      @Consumes(MediaType.APPLICATION_FORM_URLENCODED)
4      public String output(@FormParam("drug") String drug,
5                          @FormParam("quantity") int quantity,
6                          @Context HttpServletResponse servletResponse) throws
7                          IOException {
8
9      drugOrder = new Drug();
10     drugOrder.setName(drug);
11     drugOrder.setQuantity(quantity);
12     drugOrder.setTrackingNumber(DataBase.getTrackingNumber());
13     DataBase.addDrug(drugOrder);
14
15     ....
16 }
```

You have ordered 150 package(s) of Voltaren.

Your tracking number is: 15. Remind it!

[Home](#)[Back](#)

Figure 6: Result page

4.3 Tracking

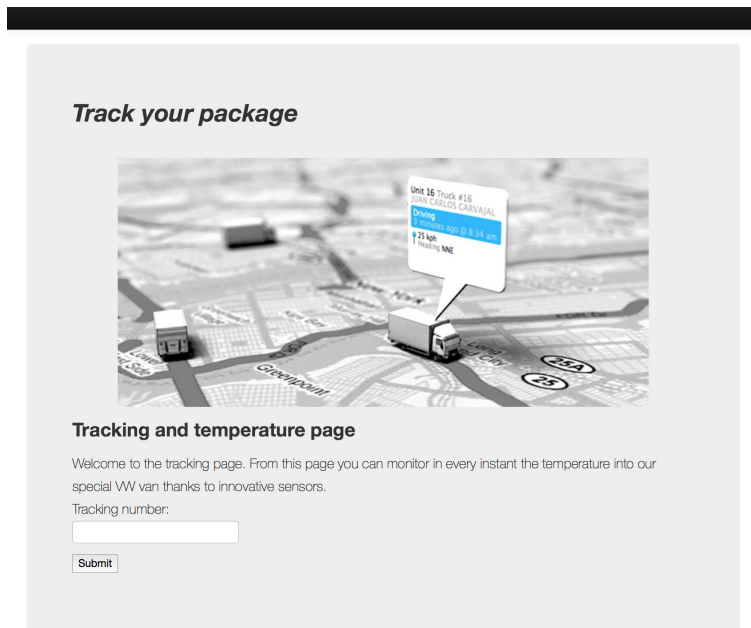


Figure 7: Tracking page

Tracking page

Inside the *tracking page*, our customers can put the tracking number, given after the medicine order. Tracking number is the only data required and

from this page customers can monitor the temperature inside our customized *Volkswagen* vans. Another important feature is the integration with *Google Maps APIs*, to show on a map where is the van.

Tracking and temperature page

Welcome to the tracking page. From this page you can monitor in every instant the temperature into our special VW van thanks to innovative sensors.

Tracking number:

Figure 8: Insert tracking number page

Order n°: 15

Details

Name: Voltaren

Quantity: 150

Location: Torino - Via Castellamonte, 10



For this medicine the temperature must be between -10° and 30°

Actual temperature: 8°

Everything is alright!

The package will arrive before the end of the day.

[Home](#)

[Back](#)



Figure 9: Tracking result page

4.3.1 Result tracking page & Google Maps

As you can see, in the *Tracking result page* there are the position and the temperature.

As regarding the position, in our implementation, we decided to use an `enum`

that includes the city and the street. However, it is pretty important to display the map showing the position. This feature has been developed using Google Maps API and an HTML `<iframe>`.

As regard the temperature, we decided to highlight the importance of having valid temperature values and some values out of limits.

4.4 Contacts

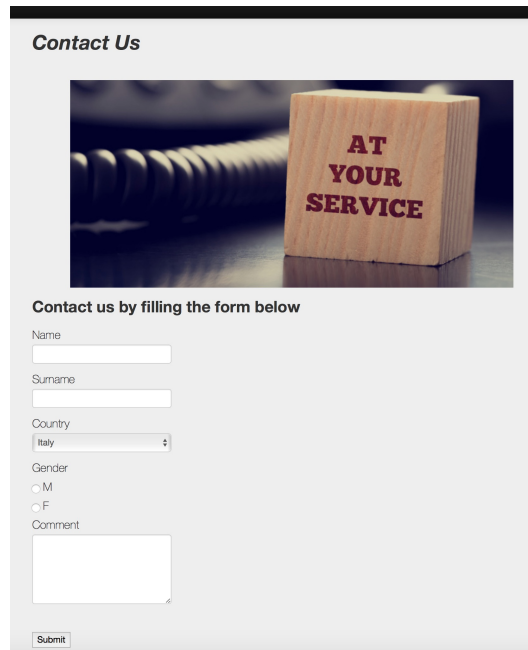


Figure 10: Contact us page

Contact us

The contact us page gives the customers the chance of asking *PoliMed* questions. Pharmacies should fill the form, in case of assistance.

Required data are:

- Name
- Surname
- Country
- Gender
- Comment

Chapter 5

5 Design and Analysis

5.1 Business Process Model and Notation

BPMN (*Business Process Model and Notation*) is a graphical representation for specifying business processes in a business process model. BPMN is used for designing processes by many companies, including hospitals like *San Raffaele* in Milan. Also *PoliMed*, during the design phase, uses BPMN to represent the business processes, which involves pharmacies, warehouse, robots and truck driver.

In our project we have divided the processes in seven BPMNs:

- **User Action**
- **Order medicine**
- **Warehouse**
- **Create package**
- **Tracking**
- **Check sensors**

5.1.1 User Action

The *User Action BPMN* represents the login phase and the possible options showed to the user. The pharmacy could order a medicine, check the tracking status or contact *PoliMed*.

If the action is concluded and the order exists, the pharmacy will wait the package and pick it when the truck is arrived.

5.1.2 Order Medicine

The *Order Medicine BPMN* represents all the actions that the user should do to order a medicine. If the medicine is in the warehouse, which means the medicine's name and the medicine's quantity are both right, the order will be inserted inside DB. If the order cannot be satisfied, it will be canceled.

After the order phase, the pharmacy will be redirect to the homepage.

5.1.3 Warehouse

The *Warehouse BPMN* is composed of warehouse man, robot and informational system. Robots pick the medicine and bring it to the warehouse man. After the *check medicine* phase, a package will be created and brought to the shipping area by the robot.

5.1.4 Create package

The *Create package BPMN* represents the actions that made by the warehouse man to pick the-right-size box, insert items, choose to apply the fragile adhesive label, attach and scan the shipping code.

Once the package has been created, it will be handled by warehouse robots in the Warehouse BPMN.

5.1.5 Tracking

The *Tracking BPMN* shows the check tracking process. The pharmacy can insert the tracking number inside the loaded page, and if the tracking number exists, it can check the order status and van's temperature.

5.1.6 Truck

Inside the *Truck BPMN*, the truck driver reaches the shipping area and picks up the package. During the “travel to destination”, the truck is always monitored by the sensors system. Once the package has been given to the pharmacy, the truck comes back to the warehouse.

5.1.7 Check sensors

The *Check sensors BPMN* represents the truck sensor system, composed of GPS sensor, temperature sensors and the main controller, whose job is to collect data and calculate the average temperature.

5.2 Choreography

The process orchestration is focused on the interaction among two or more subjects, highlighting the exchange of messages among parties. For *Polimed* have the following choreography diagram:

- **5.2.1 User**
- **5.2.2 PoliMed warehouse**
- **5.2.3 Check tracking**
- **5.2.4 Order Medicine**

The Choreography Diagram is appended.

5.3 Petri Net

The last phase is the **validation**. *PoliMed* handles very complex processes, specified by BPMN. So, we need to control the absence of behaviors problems.

For this scope we use a particular type of Petri net, called *Workflow Net*, to check the semantic correctness of business processes.

In particular, in the WF net we want to assure that the soundness property respects the rules of that property. Soundness property clarifies if a process terminates in the right way, by executing the final state and without leaving any other states active. In the WF net, property is respected by having just one token in the final place \circ and any tokens inside other place. It means that *liveness* and *1-boundness* properties are respected.

We have analyzed the package creation made by the warehouse man, and we have converted it in a Petri net. As result, we can say that our Petri net respects the propriety previously cited for the Workflow net. We can ensure that the process behaves properly respecting the soundness property.

The Workflow net is appended.