

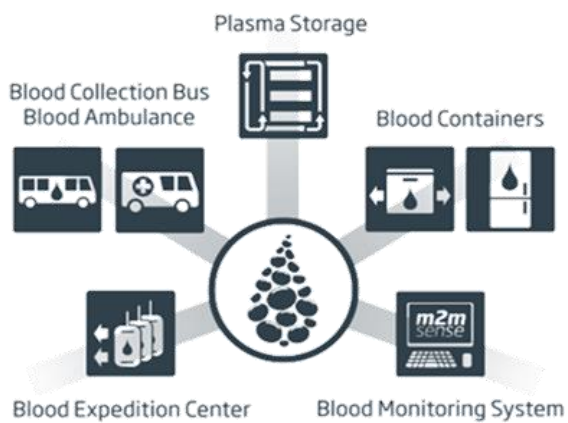
## M2M Team Sp. z o.o.

API Documentation for integration  
with MC System Software

Document version 1.0

MC GUI software version 7.0.1

MC web service software version 5.6.1.0



Warsaw 08.2024

## 1. TABLE OF CONTENTS

<b>1. DESCRIPTION OF DATA EXCHANGE BETWEEN MC SYSTEM AND BMS .....</b>	<b>3</b>
<b>2. EXAMPLES OF PRODUCT INFORMATION SENT TO THE BMS SYSTEM.....</b>	<b>3</b>
<b>3. COMMUNICATION BETWEEN THE MC SYSTEM AND THE BMS .....</b>	<b>4</b>
<b>4. API DESCRIPTION .....</b>	<b>4</b>
INTEGRATION LOGIC DIAGRAM .....	4
INTRODUCTION.....	5
API.....	5
<i>M2M Team MC – Im2mMonitoring (BB.Service)</i> .....	5
<i>BMS – MonitoringInput</i> .....	7
M2M TEAM - PROCESSORS .....	8
<i>MovementsProcessor</i> .....	8
<i>PingProcessor</i> .....	9
<b>5. EXAMPLE OF RECORD .....</b>	<b>9</b>
<b>6. EXAMPLE OF XML SCHEMA DEFINITION (XSD).....</b>	<b>10</b>
<b>7. EXAMPLE OF WSDL .....</b>	<b>11</b>
<b>8. INTRODUCTON TO MC SYSTEM.....</b>	<b>14</b>
MACHINE .....	14
SOFTWARE.....	15
<b>9. DOCUMENT CHANGE LOG .....</b>	<b>15</b>

## 1. DESCRIPTION OF DATA EXCHANGE BETWEEN MC SYSTEM AND BMS

The basic feature of MC is to automate the process of storage and providing to the user of warehouse products selected according to the given criteria. This enables the exchange of data with the local or central record system of the blood collection process and the production of blood products called the BMS system (Blood Management System). It is assumed that in the process of obtaining and producing blood products in the BMS collects basic information about the product. The MC system assumes the integration of data compliant with the ISBT 128 standard (International Society of Blood Transfusion, The Global Information Standard for Medical Products of Human Origin).

The basic unit of storage is the blood bag. It must be marked with a unique code in the whole system (Single-code or group of codes in the ISBT 128 is Donation, FinalCode, Division). It is not allowed, there are two storage units bearing the same code. A unique identification code is entered into the MC system during the product storage operation. In the process of data exchange with the BMS system, the following information describing the blood product is obtained on the basis of a unique identification code:

- date of execution of warehouse operations
- date of blood collection - ISBT compliant
- Valid until - ISBT compliant
- Donation code (string field, max 15 characters) - ISBT compliant
- Final code (string field, max 5 characters) - ISBT compliant
- Division (string field, max 3 characters) - ISBT compliant
- Class (string field, max 4 characters) - ISBT compliant
- Blood Group (string field, max 4 characters) - ISBT compliant
- Capacity (numeric field ) - ISBT compliant
- Blood Units (string field, max 5 characters) - ISBT compliant
- Fractionator - field for local/national code markings (string field, max 100 characters)
- Description - (string field, max 100 characters)

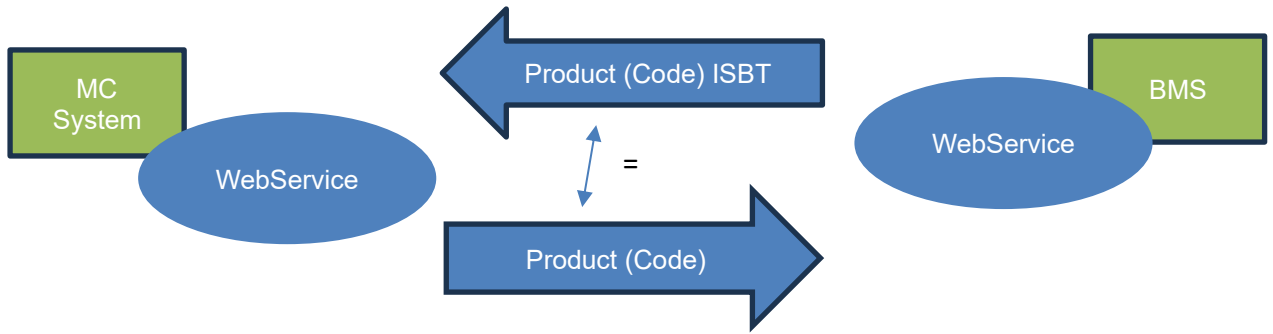
MC transfers to BMS information about the location of the product in a warehouse in the form of a unique code box (Unit) for the products.

## 2. EXAMPLES OF PRODUCT INFORMATION SENT TO THE BMS SYSTEM

- Adding a product:  
Donation='Z53331900594600';FinalCode='E6302';Division='V00';Targetunit='A26K646';User='KRA185';Description='';m2mUser='21'  
Donation='Z53331900590700';FinalCode='E6302';Division='V00';Targetunit='A26K646';User='KRA185';Description='';m2mUser='21'  
Donation='Z53331900591800';FinalCode='E6302';Division='V00';Targetunit='A26K646';User='KRA185';Description='';m2mUser='21'
- Removing product:  
Donation='Z53201908016800';FinalCode='E6183';Division='V00';Targetunit='';User='';Description='';m2mUser='1'  
Donation='Z53381901585800';FinalCode='E6183';Division='V00';Targetunit='';User='';Description='';m2mUser='1'  
Donation='Z53271900235300';FinalCode='E6183';Division='V00';Targetunit='';User='';Description='';m2mUser='1'

### 3. COMMUNICATION BETWEEN THE MC SYSTEM AND THE BMS

Communication between systems the MC and the BMS is realized in WCF (Windows Communication Foundation) technology using the Webservice mechanism.

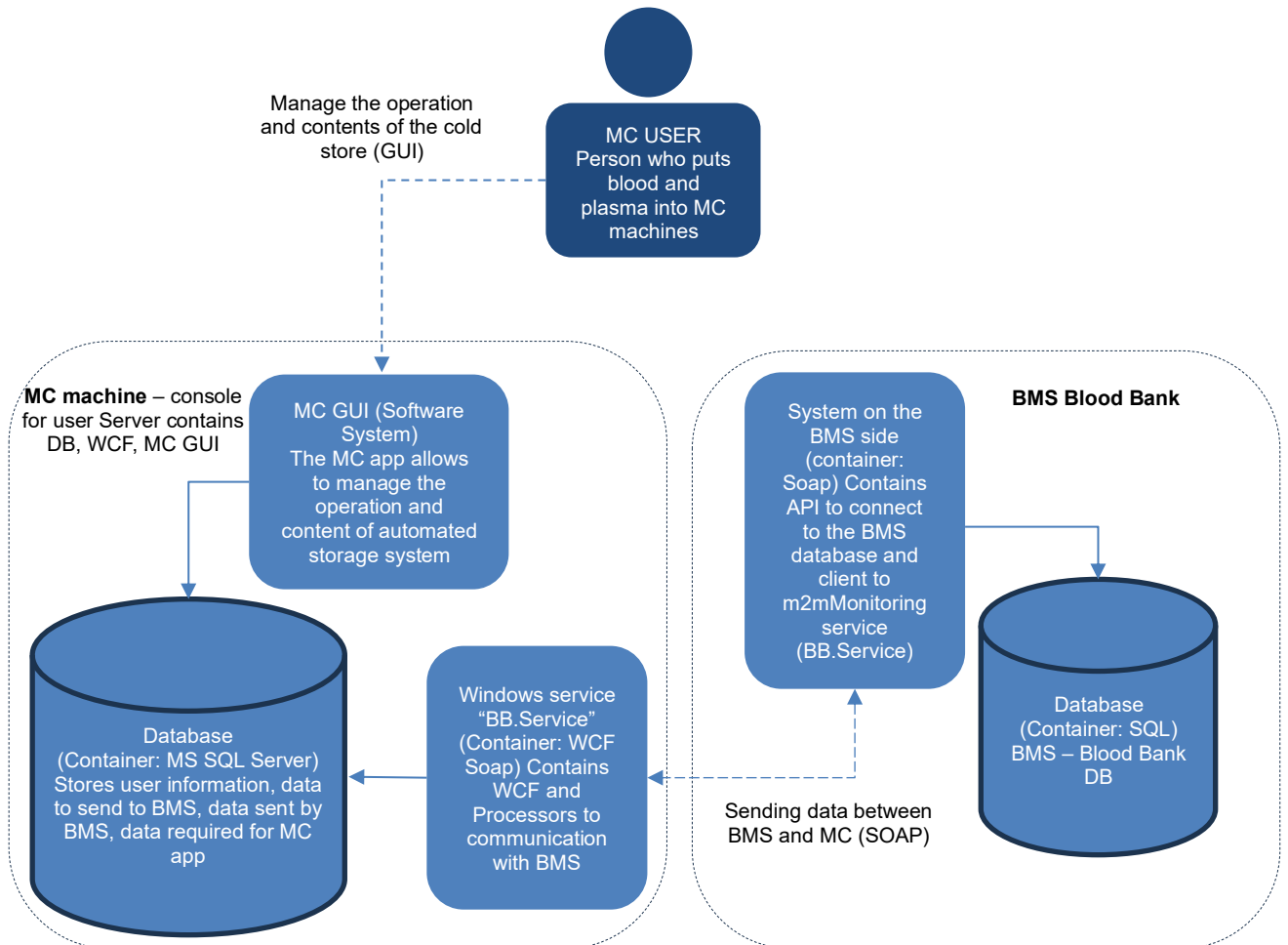


The "Product (Code) localization" command is executed when the product has been re-staged (the operation of adding or removing a product from the litter box).

The "Product (Code) ISBT details" command is executed after receiving the "Product (Code) localization" command and in case of changes to "ISBT details" in the BMS.

### 4. API DESCRIPTION

#### Integration Logic Diagram



## Introduction

The services described below are used for data exchange between the Blood Bank and the MCMC system. They use the SOAP communication protocol.

## API

### M2M Team MC – Im2mMonitoring (BB.Service)

The WCF service is used for communication between the Blood Bank and the MC system.

#### Authentication

The service uses Basic authentication. User and password will be provided.

#### Methods

##### DtTmCurrent

It takes no parameters. Returns the current server date. Format: “yyyy-MM-dd HH:mm:ss.fff”

##### SendUser (optional)

After the request is sent, a procedure is run that processes the record. In case it does not exist in the database, it adds a new user. You can also use it to deactivate a user.

#### Request

Parameter	Type	Example	Name in MC	Description
operationDate	string	2019-10-31 12:32:03.822	Time	
userName	string	KRA279	User	
firstName	string	KRA279	Name	
lastName	string	KRA279	LastName	
isActive	string	1	Status	
description	string	Anna Pecak	Description	

## DonationDetails

#### Request

Parameter	Type	Example	Name in MC	Description
operationDate	string	2020-11-16 21:56:22.397	Time	Date of operation
donation	string	Z53602005730200	Donation	Donation code
class	string	E070	Class	
division	string	V00	Division	
finalCode	string	E9794	FinalCode	
unit	string	A01K0003	Unit	Storage location at MC machine
userName	string	KRA244	User	The name of the user performing the operation
status	string	Z	Status	
description	string		Description	Additional description field
bloodGroup	string	66W0	Blood-Group	blood group
validUntil	string	2022-12-10 00:00:00.000	ValidUntil	expiration date
capacity	string	260	Capacity	Unit capacity in ml
bloodUnits	string	1.0	BloodUnits	Unit (can be half a unit or one and a half)
fracParams	string	M R	FracParams	Additional fractionation parameters according to which the hospital searches for a

				specific blood bag
bloodCollection-Date	string	2020-11-16 20:56:22.397		Date of blood collection

### Response

Communication status	Response
Success	OK
Failed	Some exception message

### Validation for DonationDetails

#### Parameter "bloodGroup"

"bloodGroup" code	Value in MC	Description
		if in your system no information about blood group send string empty
06WO	A-	
17WO	B-	
28WO	AB-	
51WO	O+	
55WO	O	
62WO	A+	
66WO	A	
73WO	B+	
77WO	B	
84WO	AB+	
88WO	AB	
95WO	O-	

If in your system no information about blood group send string empty "<bloodGroup></blood-Group>"

#### Parameter "status"

If similar statuses are not used in the blood bank, always send the value "Z" in the "status" field.

"status" code	Value in MC	Description
	App.	Approved - if empty value is sent
2	App.	Approved
8	Rej.	Rejected
4	Rest	Restricted
R	Rel.	Released
P	Proc.	Processed
Z	-	No value

Parameters "operationDate", "validUntil", "bloodCollectionDate"

Required date format: **yyyy-MM-dd HH:mm:ss.fff**

## Parameter "unit"

Each "unit" code is assigned to a specific box in Storage area in MC, e.g. A1712 ,

A1861 - A Warehouse code, 1861 – numer of the box

Range of units is from **A0001 – AXXXX** e.g.

XXXX – numer depends on capacity of the warehouse

A1712, A1861, (...), A1899

If the blood bank sends the value "<unit></unit>" the product will be removed from the MC system

## BMS - MonitoringInput

The service is used for communication between the MC system and the Blood Bank (BMS).

### Authentication

The service uses Basic authentication.

### Methods

### ChangeStorageLocation

After receiving a request via "ChangeStorageLocation" for donation, the BMS should send the donation details (via "DonationDetails") to MC. This should be done in the queue system and be repeated in case of error during sending (e.g. MC system not responding). Implementation "ChangeStorageLocation" should sets queue to send, NOT sending details of blood.

### Request

Parameter	Type	Example	Name in MC	Description
operationTime	string	2020-11-16 21:56:22.397	Time	Date of operation
donation	string	Z53602005730200	Donation	Donation code
class	string	E070	Class	
division	string	V00	Division	
unit	string	A01K0003	Unit	Storage location at MC machine
userName	string	KRA244	User	The name of the user performing the operation
description	string		Description	

### Response

If successful, the Blood Bank service should return "OK". If an exception has occurred, the response should be returned with the exception body.

Communication status	Response
Success	OK
Failed	[Some exception message]

### DtTmCurrent

#### Request

It takes no parameters. Returns the current server date. Format: "yyyy-MM-dd HH:mm:ss.fff"

Response

Returns the current server date as a string. E.g. „2021-03-18 07:05:10.158”

Communication status	Response example
Success	2021-03-18 07:05:10.158

### ChangeWarehouse

When a box is taken out of storage to change storage MC sends following request to the BMS. In this case field targetUnit is empty

Parameter	Type	Example	Name in MC	Description
dateTime	string	2020-11-16 21:56:22.397	Time	Date of operation
userCode	string	KRA244	User	The name of the user performing the operation
targetUnit	string		Targetunit	
unitCode	string	A1712	Unit	
Unittype	String	RU	Unittype	
description	string		Description	Additional description field

When a box is put into storage as a consequence of a warehouse change MC sends following request to the BMS.

Parameter	Type	Example	Name in MC	Description
dateTime	string	2020-11-16 21:56:23.399	Time	Date of operation
userCode	string	KRA998	User	The name of the user performing the operation
targetUnit	string	DP_MAG_P	Targetunit	
unitCode	string	A1844	Unit	
Unittype	String	RU	Unittype	
description	string		Description	Additional description field

### Parameter „Unittype”

"Unittype" code	Description
SN	Permanent superior
SO	Permanent final
RU	Movable final

### M2M team - Processors

Processors allow you to perform specific tasks at intervals. Currently, in the application configuration, this interval is set to 30 seconds. We have two "Processors", one is for blood bank data updates.

#### MovementsProcessor

Gets a list of all inventory movements from queue on the database to be sent to the Blood Bank. Then, they are sent using the "ChangeStorageLocation" and "ChangeWarehouse". If the sending is

successful, the record is marked as sent. Otherwise it will be marked for resend.

## PingProcessor

Reads the date from BMS server (via "DtTmCurrent"). Logs an exception on failure. It's test method

### 5. EXAMPLE OF RECORD

SoapUI „DonationDetails” request example:

```
<soapenv:Envelope xmlns:soapenv="http://schemas.xmlsoap.org/soap/envelope/"
xmlns:tem="http://tempuri.org/">
  <soapenv:Header/>
  <soapenv:Body>
    <tem:DonationDetails>

      <!--date format: yyyy-MM-dd HH:mm:ss.fff-->
      <tem:operationDate>2021-05-27 00:00:00.000</tem:operationDate>

      <!--donation-->
      <tem:donation>Z53602005730200</tem:donation>
      <tem:class> E070</tem:class>
      <tem:division> V00</tem:division>
      <tem:finalCode> E6302</tem:finalCode>

      <!--value from range A01K0001 - A14K0350 or <tem:unit></tem:unit> if product should be
removed-->
      <tem:unit>A01K0001</tem:unit>

      <!--user login-->
      <tem:userName>test</tem:userName>

      <!--value from dictionary - send "Z", if missing in blood bank -->
      <tem:status></tem:status>

      <!--string empty for description -->
      <tem:description></tem:description>

      <!--value from dictionary-->
      <tem:bloodGroup>84WO</tem:bloodGroup>

      <!--date format: yyyy-MM-dd HH:mm:ss.fff-->
      <tem:validUntil>2022-02-03 00:00:00.000</tem:validUntil>

      <!--integer type, capacity in ml-->
      <tem:capacity>290</tem:capacity>

      <!--string empty for bloodUnits -->
      <tem:bloodUnits></tem:bloodUnits>
```

```

<!--string empty for fracParams or some bag marking that is used at the blood bank -->
<tem:fracParams></tem:fracParams>

<!--date format: yyyy-MM-dd HH:mm:ss.fff-->
<tem:bloodCollectionDate>2021-05-27 00:00:00.000</tem:bloodCollectionDate>

</tem:DonationDetails>
</soapenv:Body>
</soapenv:Envelope>

```

## 6. EXAMPLE OF XML SCHEMA DEFINITION (XSD)

```

<?xml version="1.0" encoding="utf-8"?>
<xs:schema xmlns:tns="http://tempuri.org/" elementFormDefault="qualified" targetName-
space="http://tempuri.org/" xmlns:xs="http://www.w3.org/2001/XMLSchema">
  <xs:element name="DtTmCurrent">
    <xs:complexType>
      <xs:sequence />
    </xs:complexType>
  </xs:element>
  <xs:element name="DtTmCurrentResponse">
    <xs:complexType>
      <xs:sequence>
        <xs:element minOccurs="0" name="DtTmCurrentResult" nillable="true" type="xs:string" />
      </xs:sequence>
    </xs:complexType>
  </xs:element>
  <xs:element name="SendUser">
    <xs:complexType>
      <xs:sequence>
        <xs:element minOccurs="0" name="operationDate" nillable="true" type="xs:string" />
        <xs:element minOccurs="0" name="userName" nillable="true" type="xs:string" />
        <xs:element minOccurs="0" name="firstName" nillable="true" type="xs:string" />
        <xs:element minOccurs="0" name="lastName" nillable="true" type="xs:string" />
        <xs:element minOccurs="0" name="isActive" nillable="true" type="xs:string" />
        <xs:element minOccurs="0" name="description" nillable="true" type="xs:string" />
      </xs:sequence>
    </xs:complexType>
  </xs:element>
  <xs:element name="SendUserResponse">
    <xs:complexType>
      <xs:sequence>
        <xs:element minOccurs="0" name="SendUserResult" nillable="true" type="xs:string" />
      </xs:sequence>
    </xs:complexType>
  </xs:element>
  <xs:element name="DonationDetails">
    <xs:complexType>
      <xs:sequence>
        <xs:element minOccurs="0" name="operationDate" nillable="true" type="xs:string" />

```

```

<xs:element minOccurs="0" name="donation" nillable="true" type="xs:string" />
<xs:element minOccurs="0" name="class" nillable="true" type="xs:string" />
<xs:element minOccurs="0" name="division" nillable="true" type="xs:string" />
<xs:element minOccurs="0" name="finalCode" nillable="true" type="xs:string" />
<xs:element minOccurs="0" name="unit" nillable="true" type="xs:string" />
<xs:element minOccurs="0" name="userName" nillable="true" type="xs:string" />
<xs:element minOccurs="0" name="status" nillable="true" type="xs:string" />
<xs:element minOccurs="0" name="description" nillable="true" type="xs:string" />
<xs:element minOccurs="0" name="bloodGroup" nillable="true" type="xs:string" />
<xs:element minOccurs="0" name="validUntil" nillable="true" type="xs:string" />
<xs:element minOccurs="0" name="capacity" nillable="true" type="xs:string" />
<xs:element minOccurs="0" name="bloodUnits" nillable="true" type="xs:string" />
<xs:element minOccurs="0" name="fracParams" nillable="true" type="xs:string" />
<xs:element minOccurs="0" name="bloodCollectionDate" nillable="true" type="xs:string" />
</xs:sequence>
</xs:complexType>
</xs:element>
<xs:element name="DonationDetailsResponse">
  <xs:complexType>
    <xs:sequence>
      <xs:element minOccurs="0" name="DonationDetailsResult" nillable="true" type="xs:string" />
    </xs:sequence>
  </xs:complexType>
</xs:element>
</xs:schema>

```

## 7. EXAMPLE OF WSDL

```

<?xml version="1.0" encoding="utf-8"?>
<wsdl:definitions xmlns:wsap="http://schemas.xmlsoap.org/ws/2004/08/addressing/policy"
xmlns:wsa10="http://www.w3.org/2005/08/addressing" xmlns:tns="http://tempuri.org/"
xmlns:msc="http://schemas.microsoft.com/ws/2005/12/wsd/contract" xmlns:soap-
penc="http://schemas.xmlsoap.org/soap/encoding/" xmlns:wsx="http://schemas.xml-
soap.org/ws/2004/09/mex" xmlns:wsp="http://schemas.xmlsoap.org/ws/2004/09/policy"
xmlns:wsam="http://www.w3.org/2007/05/addressing/metadata" xmlns:wsa="http://sche-
mas.xmlsoap.org/ws/2004/08/addressing" xmlns:wsaw="http://www.w3.org/2006/05/ad-
dressing/wsd" xmlns:soap="http://schemas.xmlsoap.org/wsd/soap/" xmlns:soap12="http://sche-
mas.xmlsoap.org/wsd/soap12/" xmlns:wsu="http://docs.oasis-open.org/wss/2004/01/oasis-
200401-wss-wssecurity-utility-1.0.xsd" xmlns:xsd="http://www.w3.org/2001/XMLSchema"
name="m2mMonitoringInt" targetNamespace="http://tempuri.org/" xmlns:wSDL="http://sche-
mas.xmlsoap.org/wsd/">
  <wsp:Policy wsu:Id="BasicHttpBinding_Im2mMonitoring_policy">
    <wsp:ExactlyOne>
      <wsp>All>
        <http:BasicAuthentication xmlns:http="http://schemas.microsoft.com/ws/06/2004/po-
licy/http" />
      </wsp>All>
    </wsp:ExactlyOne>
  </wsp:Policy>
</wsdl:types>

```

```

<xsd:schema targetNamespace="http://tempuri.org/Imports">
  <xsd:import schemaLocation="m2mMonitoring.xsd" namespace="http://tempuri.org/" />
  <xsd:import schemaLocation="m2mMonitoring1.xsd" namespace="http://schemas.microsoft.com/2003/10/Serialization/" />
</xsd:schema>
</wsdl:types>
<wsdl:message name="Im2mMonitoring_DtTmCurrent_InputMessage">
  <wsdl:part name="parameters" element="tns:DtTmCurrent" />
</wsdl:message>
<wsdl:message name="Im2mMonitoring_DtTmCurrent_OutputMessage">
  <wsdl:part name="parameters" element="tns:DtTmCurrentResponse" />
</wsdl:message>
<wsdl:message name="Im2mMonitoring_SendUser_InputMessage">
  <wsdl:part name="parameters" element="tns:SendUser" />
</wsdl:message>
<wsdl:message name="Im2mMonitoring_SendUser_OutputMessage">
  <wsdl:part name="parameters" element="tns:SendUserResponse" />
</wsdl:message>
<wsdl:message name="Im2mMonitoring_DonationDetails_InputMessage">
  <wsdl:part name="parameters" element="tns:DonationDetails" />
</wsdl:message>
<wsdl:message name="Im2mMonitoring_DonationDetails_OutputMessage">
  <wsdl:part name="parameters" element="tns:DonationDetailsResponse" />
</wsdl:message>
<wsdl:portType name="Im2mMonitoring">
  <wsdl:operation name="DtTmCurrent">
    <wsdl:input wsaw:Action="http://tempuri.org/Im2mMonitoring/DtTmCurrent" message="tns:Im2mMonitoring_DtTmCurrent_InputMessage" />
    <wsdl:output wsaw:Action="http://tempuri.org/Im2mMonitoring/DtTmCurrentResponse" message="tns:Im2mMonitoring_DtTmCurrent_OutputMessage" />
  </wsdl:operation>
  <wsdl:operation name="SendUser">
    <wsdl:input wsaw:Action="http://tempuri.org/Im2mMonitoring/SendUser" message="tns:Im2mMonitoring_SendUser_InputMessage" />
    <wsdl:output wsaw:Action="http://tempuri.org/Im2mMonitoring/SendUserResponse" message="tns:Im2mMonitoring_SendUser_OutputMessage" />
  </wsdl:operation>
  <wsdl:operation name="DonationDetails">
    <wsdl:input wsaw:Action="http://tempuri.org/Im2mMonitoring/DonationDetails" message="tns:Im2mMonitoring_DonationDetails_InputMessage" />
    <wsdl:output wsaw:Action="http://tempuri.org/Im2mMonitoring/DonationDetailsResponse" message="tns:Im2mMonitoring_DonationDetails_OutputMessage" />
  </wsdl:operation>
</wsdl:portType>
<wsdl:binding name="BasicHttpBinding_Im2mMonitoring" type="tns:Im2mMonitoring">
  <wsp:PolicyReference URI="#BasicHttpBinding_Im2mMonitoring_policy" />
  <soap:binding transport="http://schemas.xmlsoap.org/soap/http" />
  <wsdl:operation name="DtTmCurrent">
    <soap:operation soapAction="http://tempuri.org/Im2mMonitoring/DtTmCurrent" style="document" />
  <wsdl:input>

```

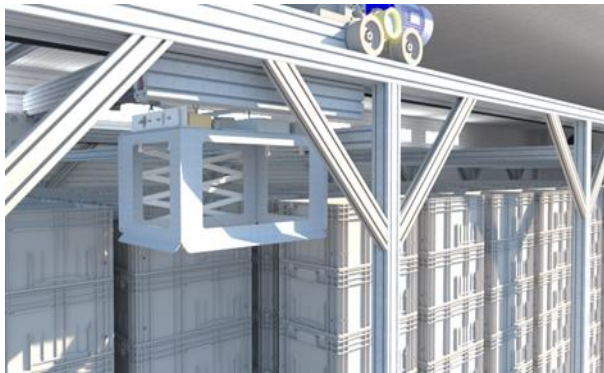
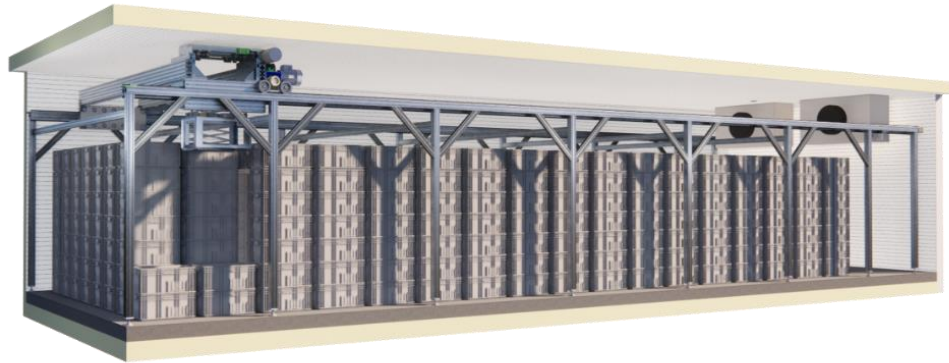
```

    <soap:body use="literal" />
  </wsdl:input>
  <wsdl:output>
    <soap:body use="literal" />
  </wsdl:output>
</wsdl:operation>
<wsdl:operation name="SendUser">
  <soap:operation soapAction="http://tempuri.org/Im2mMonitoring/SendUser" style="document" />
  <wsdl:input>
    <soap:body use="literal" />
  </wsdl:input>
  <wsdl:output>
    <soap:body use="literal" />
  </wsdl:output>
</wsdl:operation>
<wsdl:operation name="DonationDetails">
  <soap:operation soapAction="http://tempuri.org/Im2mMonitoring/DonationDetails" style="document" />
  <wsdl:input>
    <soap:body use="literal" />
  </wsdl:input>
  <wsdl:output>
    <soap:body use="literal" />
  </wsdl:output>
</wsdl:operation>
</wsdl:binding>
<wsdl:service name="m2mMonitoringInt">
  <wsdl:port name="BasicHttpBinding_Im2mMonitoring" binding="tns:BasicHttpBinding_Im2mMonitoring">
    <soap:address location="http://localhost:8443/m2mMonitoring" />
  </wsdl:port>
</wsdl:service>
</wsdl:definitions>

```

## Machine

General view of the MC plasma storage system:



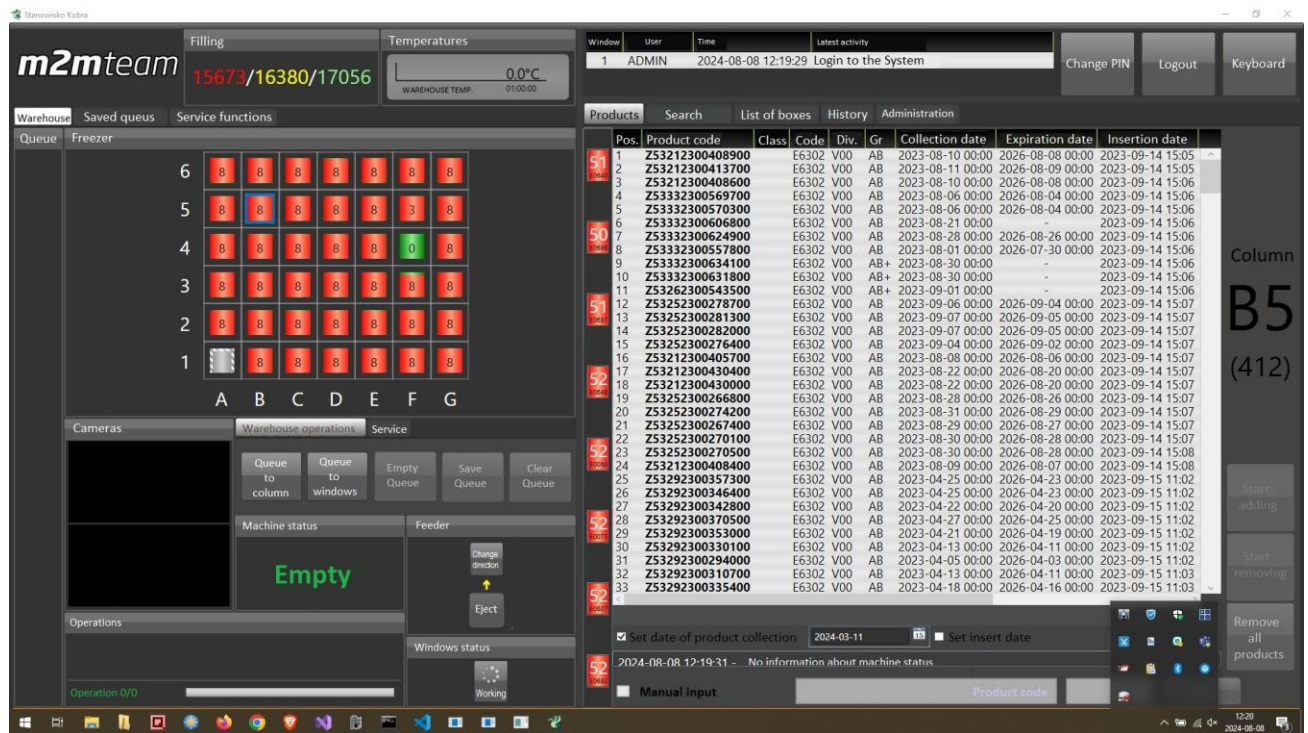
Inside the freezer, there is a device designed to transport plasma collection containers and deliver them to the access window.

The use of a computerized system for stacking containers means that there is no need to enter the freezer. A specially designed algorithm ensures that the most frequently used containers are available at the highest levels.

Service is performed from a computer station. Goods are collected at the access window to which the robot delivers the containers.

## Software

MC application is native software installed on PC Workstation with Windows operating system. Below view of main window MC application:



## 9. DOCUMENT CHANGE LOG

Changelog of the documentation below:

Version	Description of changes
1.0	First edition of document