< CONTAINER MONITORING SYSTEM >
WORK PACKAGE DEFINITION AND DEVELOPMENT RESPONSABILITIES
MINIMUN VIABLE PRODUCT CHARACTERISTICS

WWW.PARADIGMA.CL
INFO@PARADIGMA.CL

25th of January, 2019

Contenido

1. GLOBAL CONNECT WORK PACKAGE DEFINITION AND DEVELOPMENT RESPONSABILITIES

(Based on Signed Form by Innotel, Paradigma and approved by Eureka)

	Responsability		
Work Package Definition	**Innotel**	**Paradigma**	**Jointly**
1. Design of TMS architecture: Starting point: from the beginning. Expected results: a document with technical and operational requirements and common quality attributes such as performance, security, and manageability.	X		Feedback by Paradigma
a. Are the expected results A) Not solved; B) partly solved (C) completely solved, ….explain b. Is the design up to date /competitive/valid c. What is the value preposition and what real problems does it solve. See also MVP.			
2. Developing the TMS Gateway that acts as a convertor between the central Global Connect Server and on-board devices that communicate via the TMS protocol Starting point: - a set of requirements, - a list of APIs including their specific interfaces, - an overview of the modules that together form the TMS Gateway. Expected results: a working version of the TMS Gateway.	X		Feedback by Paradigma
2.1 Are the developing results: A) not solved B) partly solved (C) completely solved, …explain 2.2 Is the design up to date /competitive/valid: explain. 2.3 What is the value proposition and what are real problems that it solves?			General algorithm paper revised
3. Develop a management information base (MIB) for managing the (predefined) messages and the CID field and dialed numbers. Starting point: - a set of requirements, - a list of possible predefined messages,	X		Feedback by Paradigma

- a list of free definable messages / fields, - a number plan structure to match the CID and the dialed numbers with the (predefined) messages. Expected results: a database that can convert the highly compressed messages using CID field and dialed numbers.			
3.1 Are the developing results: B) not solved B) partly solved (C) completely solved, …explain 3.2 Is the design up to date /competitive/valid:explain What is the value proposition and what are real problems that it solves.			
4. Develop the software for the ABC router that is based on a IoT (Internet of Things) application processor that has to be installed in a cargo container. This software will have the ability to capture the data from the different sensors, like temperature, pressure, and others also the GPS location and/or the mobile operator that is operating. From that information will send that information using the TMS (Travelling Messaging System) protocol based in GSM calls or the other transmission protocols as GPRS/3G/4G or SMS if needed. The ABC Router will be able to set up different rules for the communication depending on communication tariffs, amount of sent data and location and reliability.		X	Feedback by Innotel
4.3 Is the software development : C) not solved B) partly solved (C) completely solved, …explain 4.4 Is the design up to date /competitive/valid:explain What is the value preposition and what are real problems that it solves.		4.3 partly solved 4.4 The design is up to date. An IoT devices has been specified and posible producer has send a quotation for its development.	Received feedback from Innotel
5. Develop the Container Tracking Service. The Container Tracking Service (CTS) will receive and register the messages of that state of the container, GPS location and other sensor data in a block of information using Blockchain technology. Paradigma will develop the necessary software in order to receive from the TMS the data that has captured from the containers and will publish it using the Blockchain network and a virtual chain that is attached to it. This data will be accessible for the Fleet management systems and the different partners involved in the move of a container.		X	Feedback by Innotel
5.5 Is the software in development?: D) not solved B) partly solved (C) completely solved, …explain 5.6 Is the design up to date /competitive/valid:explain		5.5 Partly solved 5.6 The design is up to date, shared among Project participants	Received feedback from Innotel

What is the value proposition and what are real problems that it solves.		The value proposition is the reception of certified measurements nearly in realtime about the internal conditions of a container, facilitating the ontime comercial and distribution decision making	
6. To test the complete solution, a pilot project will be executed jointly with SEACO Global in one of the container carried by one of their ships.	TMS gateway must be operative	ABC routers should be available CTS must be operative	Conduct a real pilot test
6.7 Is the development: E) not solved B) partly solved (C) completely solved, …explain 6.8 Is the design up to date /competitive/valid:explain What is the value preposition and what are real problems that it solves as a whole.	Not solved	Not solved	Not solved
7 Are there any new developments? IF yes explain The sensor data must be captured during all the journey, when the container is travelling through the ocean the communication must be done by satellite. There is the need to add TMS gateway for satellite modules. The containers are loaded in a ship and are allocated in different positions, some on top of other containers, and/or surrounded by several containers. The communications by satellite or mobile communications can be affected for some containers. There is the need to add a solution to get the information from the containers that are blocked to communicate through satellite or mobile and send them in realtime. A new device ABC router has to be designed and develop as the satellite and the capacity to send information through neighbouring ABC routers was not considered. 8 How new developments cost are divided. 9 ETC		Innotel is addressing the development of the TMS gateway for Satellite transmition. Paradigma is addressing the development of the functionalities in the ABC Router to send the sensor data through neighbour ABC routers. Define who can develop this device. The cost of the device should be financed by both parties.	

2. CUSTOMER FEEDBACK OF THE GLOBAL CONNECT SOLUTION

A customer feedback from potential customers was conducted during January 2019. One of them representing the fruit industry and another representing he seafood industry of Chile.

The fruit industry representative indicated that it is essential to receive the information about the temperature and humidity, but the opportunity is very important. To get the information before the arrival to the destination is essential to prevent possible problems on the way to market. There is no solution being used right now, only checking the final logs at the end of the journey, but it was too late to prevent the damage of the cargo.

The companies are using the devices to log the mainly humidty and tempreature, but have realized that the information stored in the devices is good, but does not add sufficient value for the commercialization of the products, as when the products have arrived to the destination, it is too late to know that in the middle of the journey there was some temperature problems and the products are not considered the best, they loose value or it is lost. It is expected that if the companies have more real time information about what is going on with their cargo, they could make better decisions before the merchandise arrives to the destination and final distribution. They have realized that it is difficult to get the information of the containers that are in middle of the pile, and that to solve that is essential.

In the seafood industry, their needs are similar as the fruit industry, but their actual state of advance is more initial, as their level of digitalization in the distribution process is starting. They distribute their products by ships and by planes. Their cargo is frozen (70%) and fresh (30%), so their tracking conditions are different.

Feedback conclusions

i) There is an essential need to have the information of the conditions of transportation of their products from the origin to the destination closest to realtime. It can improve their sales revenues and reduce the loss of products. Nowadays, they are getting the information at the end of the journey, but it is considered as it does not add sufficient value, as the state of the cargo sent is known at that moment and only adds statistical information ratifying what it was found.

ii) Not all the containers will be able to transmit using mobile or satellite operators as there are piles of containers on top the others or at the sides.

iii) There is a need to receive the information of the conditions of transportation of their products for each and all their containers. Knowing that specific container is not meeting the rules of acceptance for a certain period of time can help to define an intervention or to check a specific container during the voyage in order to mitigate the situation. Also could help to define during the journey the final use or commercialization of the products of the container.

iv) The costs indications considered in the transportation of container are the goods that were damaged or the loss of its value during the transportation. Probably when the costs for the tracking service are know it Will be put into consideration depending on the losses the customer is having and can be prevented with the accesability of the sensor data.

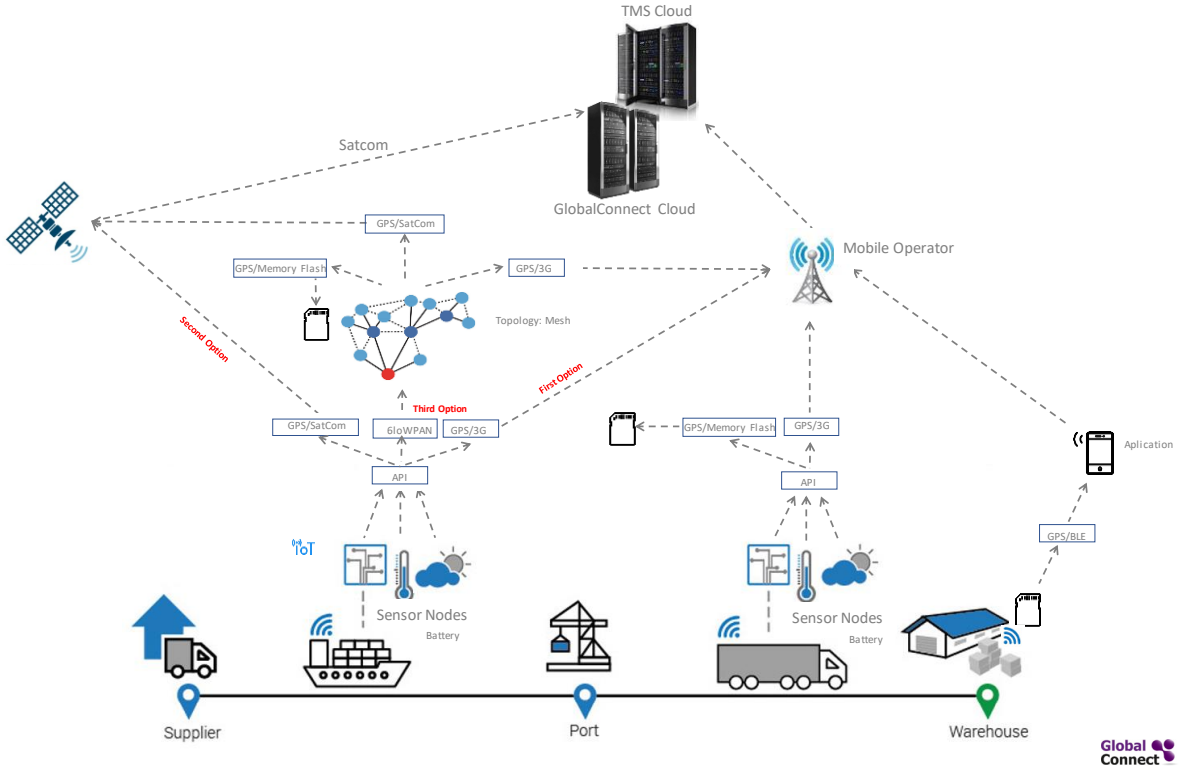
3. THE GLOBAL CONNECT COMPLETE PRODUCT DEFINITION

A deep research was conducted of the state of the art of tracking goods in transport logistics; the emerging solutions; telecommunications technologies available, limitations and costs; assurance of the authenticity of the information; emerging low power wireless technologies; the need to use international distributed application solutions and identities; among other factors.

This research helped to establish the Global Connect Complete Product Definition. The following Complete Product Definition establishes not only the functionalities that the Global Connect will have, but also the technologies involved its development and deployment.

3.1 General System Architecture

Below you can see the general architecture of the final container management system.



3.2 System Description

3.2.1 Solution 6loWPAN with mesh topology with WI-SUN

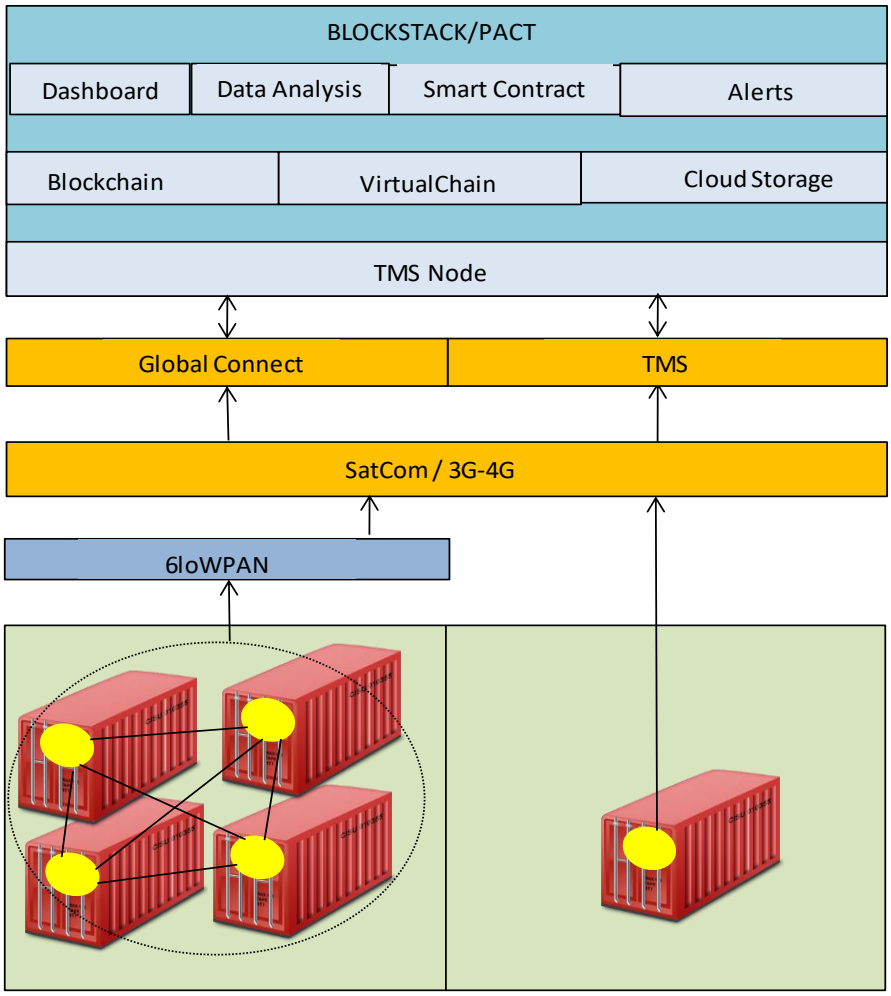
The first stage is the installation in containers of IoT devices with capacity to measure temperature, relative humidity, concentration of gases (CO2, O2, ethylene ...), vibration, impact, proximity, location (GPS), opening of the enclosure, luminosity, weight, or another. Each of the sensors have thresholds established by the IoT device administration service provided by Paradigma through the distributed web platform.

In addition to the minimum and maximum thresholds, it is considered to configure the frequency of capture/transmission of sensor data.

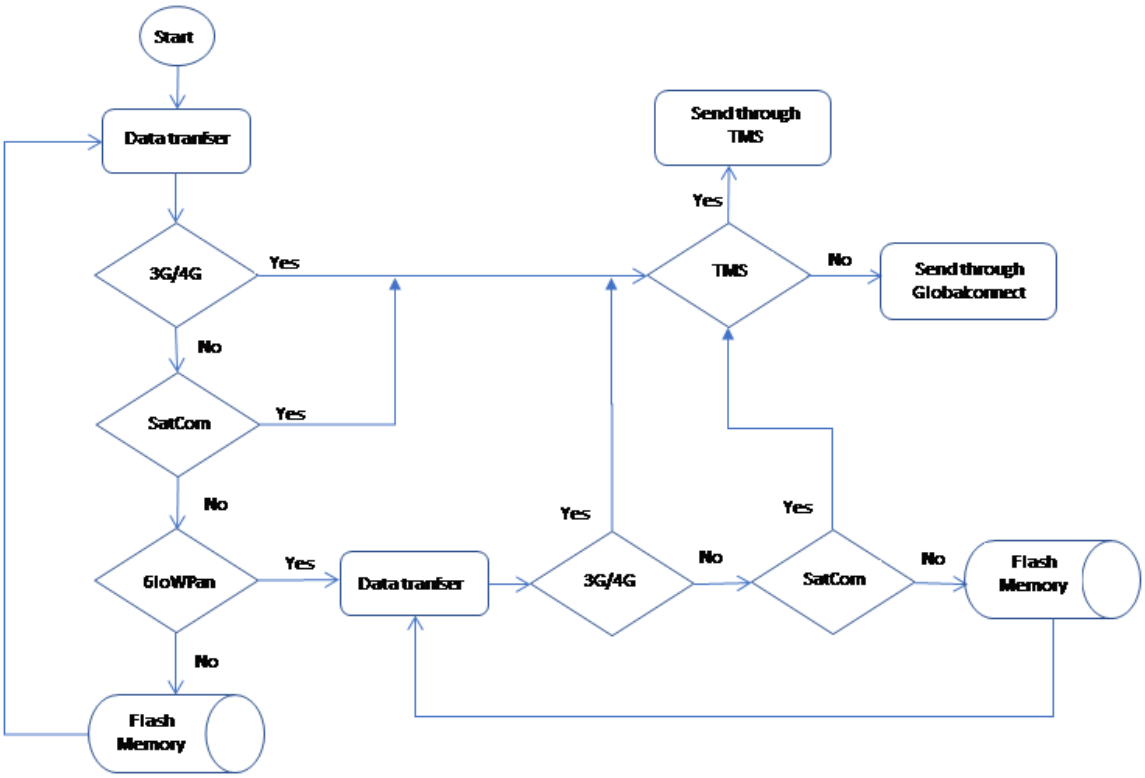
These IoT devices should have a built-in GPS module for localization, and communication modules for 3G/4G mobile operators, Satcom satellite telephony, Bluetooth BLE, and a wireless network module 6loWPAN. Additionally, a flash memory storage system will be implemented in case of no telecommunication coverage.

The sensors must be able to communicate both on the high seas and on land, that is, the system is multimodal.

The following figure gives us a vision of the devices and their communication:



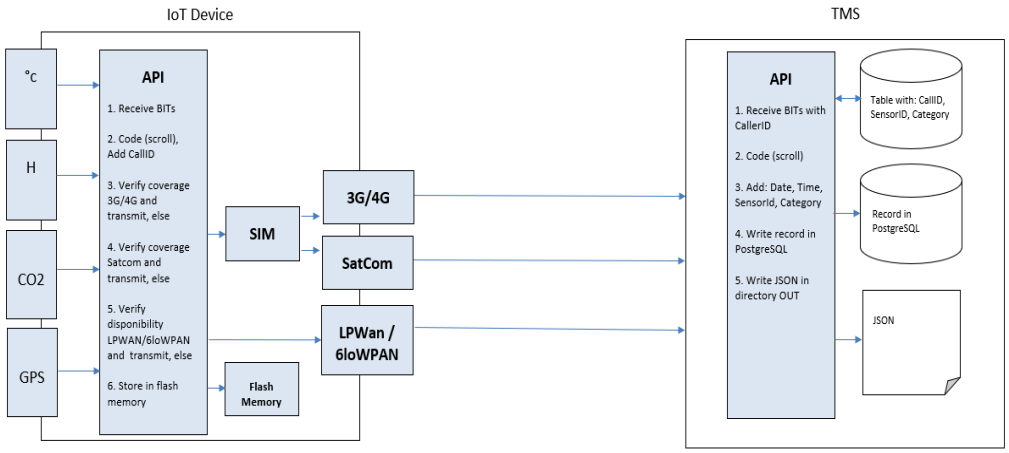
The IoT device will communicate in the first instance with the mobile operator to transmit the data, if it fails to do so it will be communicated by means of satellite telephony to transmit the data, if it does not succeed because of being without coverage or being very low on the stack of containers, which prevents communication, will transmit the data to the nearest device through the 6loWPAN protocol (wireless network), in case no nearby device can be found, the data will be stored in flash memory waiting for coverage to be transmitted.



3.2.3 The IoT devices and the TMS Gateway

The TMS technology takes in a new way advantage of the signaling channel in the mobile network system to set up a call / message for communicating between the container cloud. The container is encoded with its own ID and specified message is encoded in a way that a set of calls Will represent that information.

Each device has a CallID, an associated telephone number (SIM card) and Satellite Phone Number. It must also be considered that the devices can consist of one or more sensors, for this reason the devices must be categorized, since the TMS Server (central station) must understand what IoT device sends data, what is the value of this data and its category, the category will tell us how many measurements (sensors) the IoT device contains, in such a way that the TMS knows how many measurements it should receive from the device, and by means of a formula scroll will be received.



We assume that, on one hand, we have various IoT devices that can broadcast diverse information, that are equipped with phone numbers and can make phone calls.

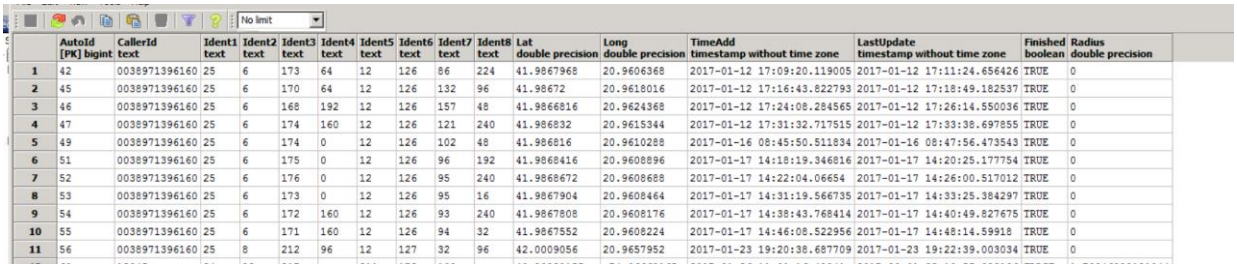
And, on the other hand, we have a centralized station (TMS) with a certain number of telephone numbers that is used to retrieve the information of telephone calls made to this station.

Therefore, on the side of the devices, you must somehow encode the information you want to transmit. This coding must be done through a list of subsequent telephone calls at the central station, which then uses this list of calls to decode the information obtained and sent by a device.

The central station has a database of all the IoT that can make calls, which includes their telephone numbers, along with the category to which they belong. In this way, the phone number (CallID) that calls automatically will give us information about the sensor identification (which sensor) and the category (type of data sent). It means that we do not need coding for these two elements. The only information that needs coding is the value of the data. As there are more calls to code the data, we must identify which call belongs to which part of the information.

The TMS when it recognizes a call, executes a script that has the telephone number and the extension as input arguments.

This script then decodes the value (extension) and places the decoded value in a database (in this case it is implemented in PostgreSql) that contains all the data emitted for different sensors:



	AutoId [PK] bigint	CallerId text	Ident1 text	Ident2 text	Ident3 text	Ident4 text	Ident5 text	Ident6 text	Ident7 text	Ident8 text	Lat double precision	Long double precision	TimeAdd timestamp without time zone	LastUpdate timestamp without time zone	Finished boolean	Radius double precision
1	42	0038971396160	25	6	173	64	12	126	86	224	41.9867968	20.9606368	2017-01-12 17:09:20.119005	2017-01-12 17:11:24.656426	TRUE	0
2	45	0038971396160	25	6	170	64	12	126	132	96	41.98672	20.9618016	2017-01-12 17:16:43.822793	2017-01-12 17:18:49.182537	TRUE	0
3	46	0038971396160	25	6	168	192	12	126	157	48	41.9866816	20.9624368	2017-01-12 17:24:08.284565	2017-01-12 17:26:14.550036	TRUE	0
4	47	0038971396160	25	6	174	160	12	126	121	240	41.986832	20.9615344	2017-01-12 17:31:32.717515	2017-01-12 17:33:38.697855	TRUE	0
5	49	0038971396160	25	6	174	0	12	126	102	48	41.986816	20.9610288	2017-01-16 08:45:50.511834	2017-01-16 08:47:56.473543	TRUE	0
6	51	0038971396160	25	6	175	0	12	126	96	192	41.9868416	20.9608896	2017-01-17 14:18:19.346816	2017-01-17 14:20:25.177754	TRUE	0
7	52	0038971396160	25	6	176	0	12	126	95	240	41.9868672	20.9608688	2017-01-17 14:22:04.06654	2017-01-17 14:26:00.517012	TRUE	0
8	53	0038971396160	25	6	173	0	12	126	95	16	41.9867904	20.9608464	2017-01-17 14:31:19.566735	2017-01-17 14:33:25.384297	TRUE	0
9	54	0038971396160	25	6	172	160	12	126	93	240	41.9867808	20.9608176	2017-01-17 14:38:43.768414	2017-01-17 14:40:49.827675	TRUE	0
10	55	0038971396160	25	6	171	160	12	126	94	32	41.9867552	20.9608224	2017-01-17 14:46:08.522956	2017-01-17 14:48:14.59918	TRUE	0
11	56	0038971396160	25	8	212	96	12	127	32	96	42.0009056	20.9657952	2017-01-23 19:20:38.687709	2017-01-23 19:22:39.003034	TRUE	0

The average time needed to establish call is approximately 5 seconds. Taking that in our case we have available the phone extension ranging from 0000 until 9999, with one call we can transfer approximately 13.3 bits. From this we can conclude that for this communication channel the capacity is 13.3 bits / 5 sec = 2.66 bits/sec.

In this way, with 11 calls we have the data of one container (4 calls for longitude, 4 calls for latitude, 1 call for temperature, 1 call for relative humidity and 1 call for carbon dioxide). Totalizing 55 seconds in the case of mobile telephony, 99 seconds in the case of satellite telephony (placing a delta of 4 seconds with respect to mobile telephony).

For each phone call, a new record is added, filling in one by one all the values for each identifier separately. The device is programmed in such a way that it initially sends the longitude and latitude values first.

For each phone call, a new record is added, filling in one by one all the values for each identifier separately. The device is programmed in such a way that it initially sends the longitude and latitude values first.

For each call, a JSON is generated, which must be saved in the OUT folder:

Schema example

```
{
  "DTEvent": string,
  "EventsContainer": [
     {
         "CallID": string,
         "EventsSensor": [
            {
               "ContainerSensorId": number,
               "ContainerCategorySensorId": number,
               "Value": string,
               "PositionsGPS": {
                   "coordinates": [
                     number,
                     number
                   ]
                   "type": Point
               },
               "DTregistry": string
            }
         ]
      }
   ]
}
```

Event example


```
{
  "data": [
    {
          "DTEvent": "2018-10-21T12:34:00:7992",
          "EventsContainer": [
             {
                 "CallID": "9835676546",
                 "EventsSensor": [
                    {
                       "ContainerSensorId": 4019,
                       "ContainerCategorySensorId": 34,
                       "Value": "1",
                       "PositionsGPS": {
                           "coordinates": [
                             -0.308595436,
                             39.43556368
                           ]
                           "type": Point
                       },
                       "DTregistry": "2018-10-21T10:30:44:00Z",
                    },
                    {
                       "ContainerSensorId": 4020,
                       "ContainerCategorySensorId": 34,
                       "Value": "14",
                       "PositionsGPS": {
                           "coordinates": [
                             -0.308595436,
                             39.43556368
                           ]
                           "type": Point
                       },
                       "DTregistry": "2018-10-21T10:33:59:00Z",
                    }

                ]
              }
           ]
        }

  ]
}
```


3.2.4 TMS central station and the TMS Node API

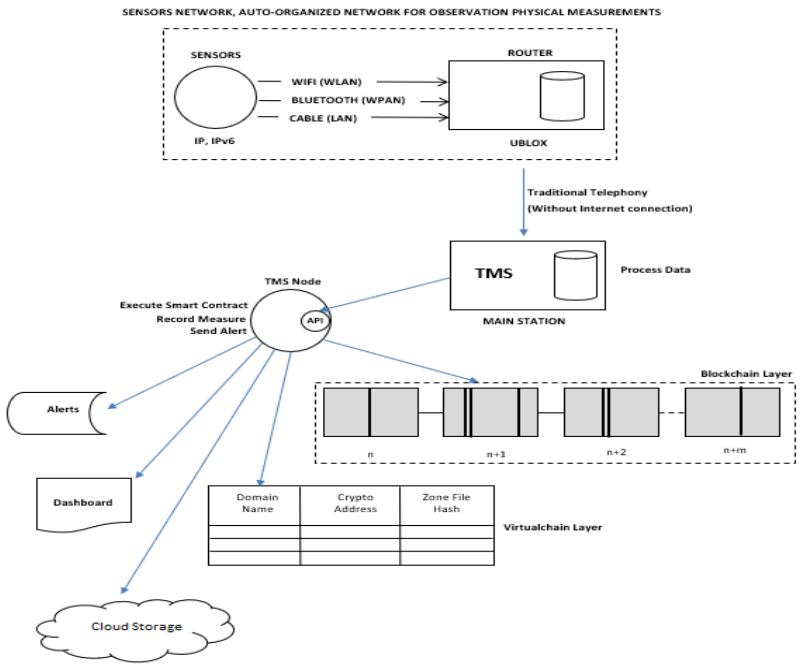
The application has the ability to communicate between the TMS Gateway and Blockstack using the HTTP protocol and JSON files (web services).

The TMS server processes and sends the received data from the IoT devices through a specialized Peer to Peer Network based in Atlas, but modified for this solution, renamed as “Globalconnect network”. The components of this network are the TMS Server and several TMS nodes that could be deployed to cover different territories. The TMS server decodes and sends the data of the measurements in XML or JSON format to the *TMS Node* that is in the Atlas Network of the BlockStack of the *TMS Node* ID.

The TMS node receives the data from the TMS Server in XML or JSON format and performs the following actions:

1. Parser the JSON file
2. Relate CallID with ID Container and Bill of lading number
   Note 1*: Each container is identified by a unique identifier. This identifier has a specific format: three uppercase letters followed by a U, to end with seven digits (for example, BMOU8710633).*
   Note 2*: You must have previously related the CallID of the IoT device with the ID of the container and with the Bill of Lading.*

3. With BL number, look for Smart contract associated to BL and retrieve and execute the *Smart Contract* using the received information.
4. Compare the received measurement with the *Smart Contract* rules.
   a. If does NOT comply then:
      - Records the issue in Blockstack (Virtualchain layer and user Storage layer) and in the underlying Blockchain.
      - Send specific user alerts, to be determined (SMS, eMAIL).
   b. If it complies then:
      - Records only in Blockstack (virtualchain layer and user storage layer).



The *TMS Node* API contains classes and methods that allow functionalities such as: search user on which the data is stored, apply the rules of the Smart Contract (associated with the BL), determine if it complies or if it does not comply, write measures in Blockchain and/or Blockstack, send alerts via SMS and/or Email.

The parameters needed are:

- Date and time of shipment (by TMS)
- Date and time received by the *TMS Node*
- Caller ID (from where it is sent, to look for user)
- Record type (if temperature, humidity, location, etc ...)
- The actual measurement received.

The CallID allows to know the ID of the container and the folio of the Bill of Lading.

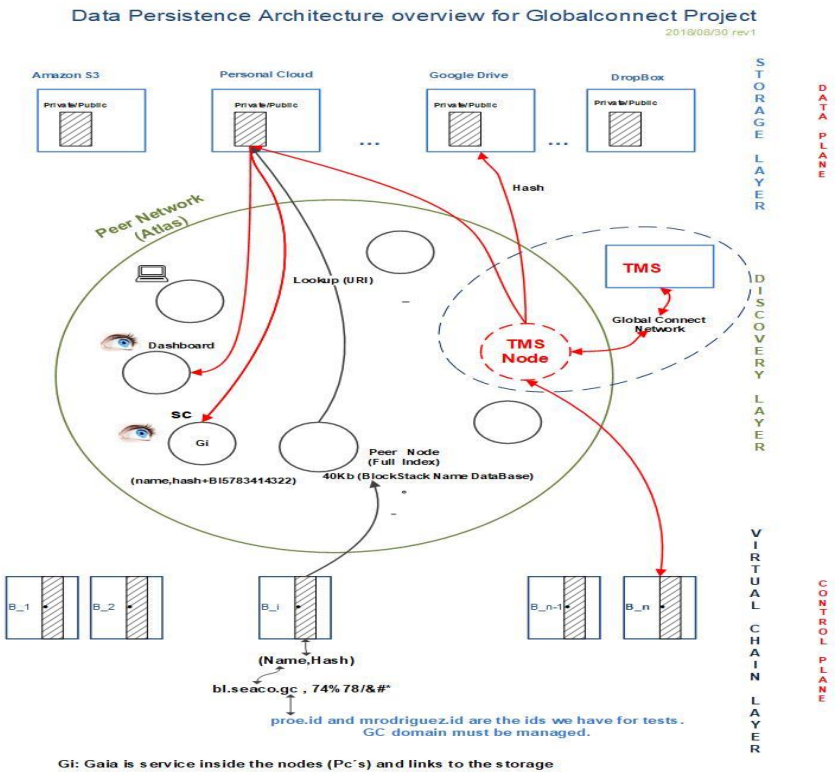
3.2.3 Blockchain with Blockstack

It is a closed ecosystem, a private and permissioned network. The members are invited to join and keep a copy of the ledger Blockchain technology.

The access to the respective network is be restricted and security is thus heightened.

The architecture is based on the Blockstack framework that implements the decentralized authorization through a DNS and PKI, can be used to share IoT device node addresses and data without a central authority in a trustless manner, and distributed storage in the cloud, which allows us to have scalability.

Blockstack domains (called BNS) are not registered on the traditional DNS run by an organized called ICANN. Instead they're registered on a blockchain in a fully decentralized way. This means that Blockstack domains are truly owned by their owners and cannot be taken away. All Blockstack domains have public keys by default (public keys are required to own the domains).



The implementation has three components:

- A *blockchain*, implemented using *virtualchains*, is used to bind digital property, like domain names, to public keys. Blockstack's blockchain solves the problem of bootstrapping trust in a decentralized way i.e., a new node on the network can independently verify all data bindings.

- A peer network, called *Atlas*, gives a global index for discovery information and

- A decentralized storage system, called *Gaia*, provides high-performance storage backends without introducing central trusted parties.

The architecture decouples the security of name registration and name ownership from the availability of data associated with names by separating the control and data planes.

Introduces four layers, with two layers (*blockchain* layer and *virtualchains* layer) in the control plane, and two layers (routing layer and data storage layer) in the data plane.

The control plane consists of a *blockchain* and a logically separate called a *virtualchains*. In the *blockchain* and in the *virtualchains* (Control Plane) is defines the protocol for registering human-readable names, creating (name,hash) bindings (by example: bl.seaco.gc, x7@#34.....) and creating bindings to owning cryptographic keypairs.

Virtualchains are like virtual machines, where a specific VM like Debian 8.7 can run on top of a specific physical machine. Different types of *virtualchains* can be defined and they run on top of the specific underlying blockchain.

Virtualchains operations are encoded in valid *blockchain* transactions as additional meta-data.

Blockchain nodes do see the raw transactions, but the logic to process virtualchain operations only exists at the virtualchain level.

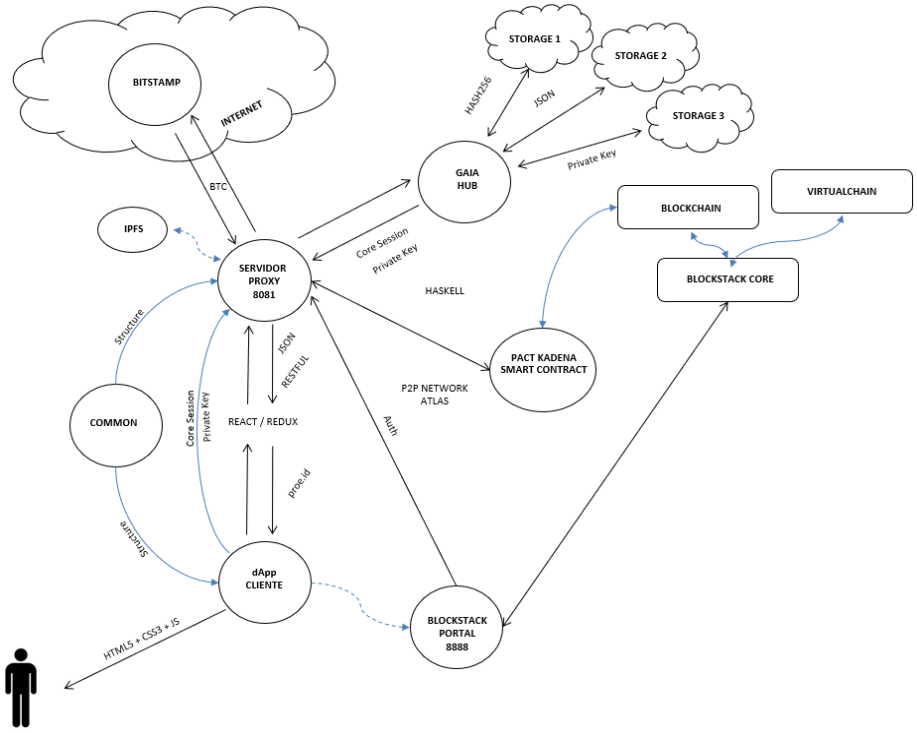
The data plane is responsible for data storage and availability. It consists of:

a) zone files for discovering data by hash or URL, and
b) external storage systems for storing data.

Data values are signed by the public keys of the respective name owners. Clients read data values from the data plane and verify their authenticity by checking that either the data's hash is in the zone file, or the data includes a signature with the name owner's public key.

This hides the individual APIs for storage backends and exposes a simple PUT/GET interface to Blockstack users. Looking up data for a name, like werner.id, works as follows:

- Lookup the name in the virtualchain to get the (name, hash) pair.
- Lookup the hash (name) in the Atlas network to get the respective zone file (all peers in the Atlas network have the full replica of all zonefiles).
- Get the storage backend URI from the zonefile and lookup the URI to connect to the storage backend.
- Read the data (decrypt it if needed and if you have the access rights) and verify the respective signature or hash.



3.2.5 Smart Contract with PACT

The traditional Bill of Lading (BL) will be complemented with a *Smart Contract* on the BlockStack and KadenaIO Pact. Each BL could have a completely different *Smart Contract* as the container cargo could be different. This method will provide a high level of security, assurance of compliance of quality measures, traceability, flexibility and a more efficient process to archive past transactions – all key issues in the logistics industry.

The Bill of Lading is the core of the system and is associated with a Blockstack user (for example proe.id) and a *Smart Contract* that evaluates service compliance. The *Smart Contract* gets executed whenever new measurement information is received evaluating if that information is complying with the rules inicially defined.

In addition, the BL is associated with a container and associated with several communication possibilities such as satellite telephone, mobile telephone, among others (PSTN, GSM, etc ...).

Finally, a BL will also be associated with the container enrollment and the IoT devices and sensors.

The system before the association, must verify that the appropiate IoT device is available or occupied.

The user node that assigns the BL with the associations described above (for example proe.id) must top-up BTC in its own application BTC account and in order to operate the system will transfer funds (BTC) to the *Smart Contract* associated with the service. The service requieres funds to operate.

The *Smart Contract* is used to record information. In this way, each time a measure is received, the *Smart Contract* is executed.

The Smart Contract will store the information in the underlying Blockchain or VirtualChain:

- Assignment of BL.
- Non-compliance with Smart Contract.
- Termination of the service associated with the BL.

The user node that assigns a IoT device to the BL (for example, proe.id) will have to pay also to the *TMS Node* account a guarantee for returning the of the IoT device, so that it registers that event in the Blockchain.

Once the user who has previously assigned, the service is completed and the IoT device is returned, the *TMS Node* returns the guarantee to the original sender account.

The user who assigns and requests the traceability service authorizes the *TMS Node* to write the storage layer associated with the service. And this same authorizes other users to read the same data.

❖ **Smart Contract Events**

- Born/ Smartcontract starts and price determination
- Link Sensor with container (Subscription)
- Initial Tracking / Location Data
- Voyage Tracking
- Arrived to Destination
- Cargo Accepted/Rejected (proof of delivery)
- Return of IoT Device/ Smartcontract ends

3.2.6 dApp

The applications of the ecosystem are "serverless" (Serverless Application) and are totally descentralized. There are no intermediaries or servers.

They are SPA applications (single-page) in Javascript and Haskell. They contain a library called " blockstack.js. "

The ecosystem of paradigm applications facilitates the exchange of transactions between an institution and its end users, and allows the persistence of data in the cloud, in JSON format, so that they can be imported into the organization's own management systems.

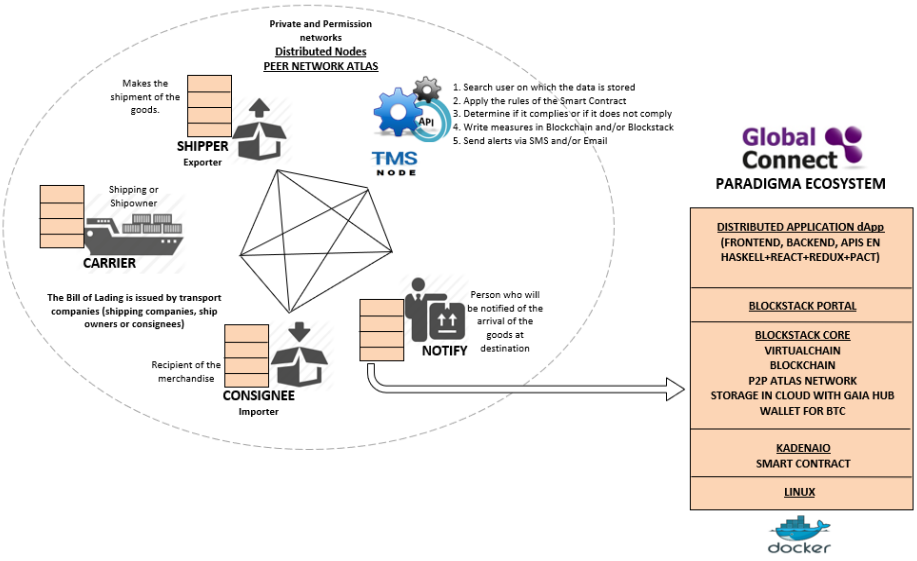
Paradigm applications facilitate, through "Blockstack", manage and authenticate the user, to have access to their resources, perform transactions and allow the storage of data in the cloud. The storage of data is simple and reliable and uses the existing infrastructure in the cloud. Users connect with their Dropbox, Google Drive, AWS Amazon, One Drive, among others, and the data is synchronized from their local device to the cloud, managed by Gaia Hub.

The identity is controlled by the user and uses the "Blockchain" for the secure management of keys, devices and user names.

The connection of the users with the applications, are anonymous by default and use a specific application key, but their full identity can be revealed at any time. The keys are for signing and encrypting and can be changed.

Through "Blockstack" a decentralized domain name system (DNS) is provided, which allows creating a distributed but closed environment under a domain for a financial organization, that is, where all the actors under the domain participate.

It also allows a decentralized public key distribution system, a registry of applications and also allows the control of user identities.

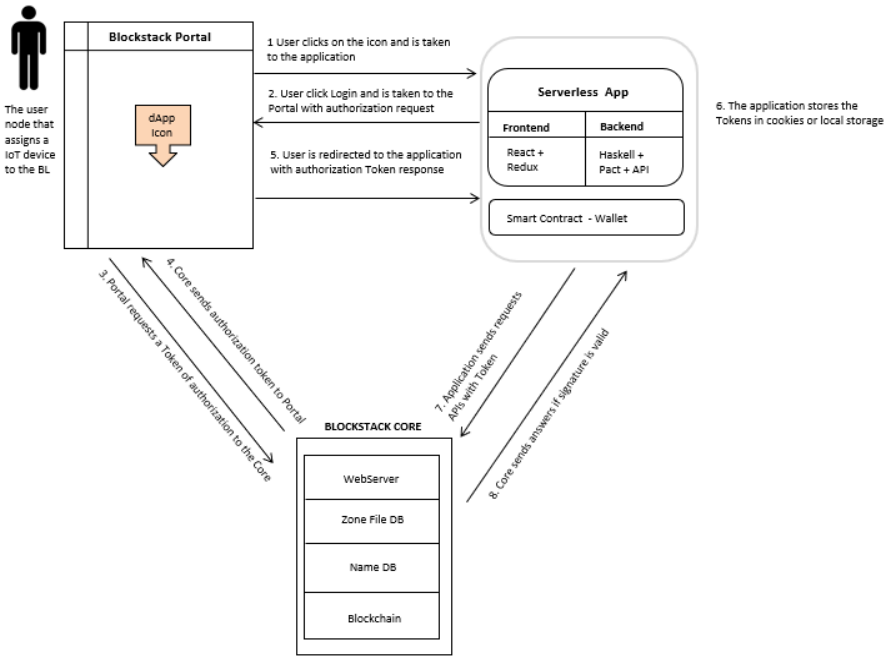


Users store and share all the transactions made in the domain, the scripts contain a high level of encryption, the transactions are propagated through the P2P protocol (with the exception of the Smartphone and tablets that must be synchronized -PUSH SYNCHRONIZATION IN BACKGROUND).

Each user has a series of "dockerized" applications installed locally, including:

- A set (one or several) of Web applications (Frontend, Backend) in Javascript and Haskell developed by Paradigma (Dashboard, IoT Devices Initialization Service to link them to the container, container tracking, reports, ...)
- Blockstack Portal

- Blockstack Core
- Wallet
- S.O. Linux



The environment consists of a "Blockstack Portal" as a container for Paradigma applications. The user by pressing on the icon of the application contained in the Portal, is taken to the application. Next, you must authenticate yourself through the login and the application will take you to the Portal to request an authorization token to the Core. The Core sends the token to the Portal, the application stores the token in cookies or localstorage. Then, with the token the application can send API requests to the Core and the latter returns the answers if the signature is valid.

4. MINIMUM VIABLE PRODUCT DEFINITION

The purpose of having a Minimum Viable Product is to serve as pivot to test a new fundamental hypothesis about the product, strategy, and engine of growth.

It is about aligning our efforts with a business and product that are working to create value and drive growth, in other words to put us on a path toward growing a sustainable business.

There are several hypotheses that have emerged from the feedback with the potential customers presented in the conclusions in 2.

1. There is an essential need to have the information of the conditions of transportation of their products from the origin to the destination closest to realtime.
2. There is a need to receive the information of the conditions of transportation of their products for each and all their containers.
3. Total costs will be put into consideration, but it is safe to say that it depends on the losses the customer is having.
4. It is also safe to say that a delay in hours of vital information (the refrigeration unit can give a sign that the container is too hot) can reduce cost.

Based on these hypotheses it was defined a set of Minimum Viable Product (MVP) characteristics to solve the problem.

MVP Characteristics Name: A. Global Connect Basic Capture
Hypotheses of existing client problems: 1. There is an essential need to have the information of the conditions of transportation of their products from the origin to the destination closest to realtime for a on-time decision (logistics) and commercialization of goods. 2. It is also safe to say that a delay in hours of vital information (the refrigeration unit can give a sign that the container is too hot) can reduce cost.
Characteristics to solve the problem Installation in containers of IoT devices with capacity to measure temperature, relative humidity, concentration of gases (CO2, O2, ethylene ...), location (GPS), Each of the sensors have thresholds established by the IoT device administration service provided by Paradigma through the distributed web platform. In addition to the minimum and maximum thresholds, it is considered to configure the frequency of capture/transmission of sensor data. These IoT devices should have a built-in GPS module for localization, and communication modules for 3G/4G mobile operators, Satcom satellite telephony, Bluetooth BLE. Additionally, a flash memory storage system will be implemented in case of no telecommunication coverage. The sensors must be able to communicate both on the high seas and on land, that is, the system is multimodal. The IoT devices must send the captured data to the Global Connect Cloud in the Internet through, GPRS, 3G, or 4G data communications using the mobile or satellite operators. The IoT device will communicate in the first instance with the mobile operator to transmit the data, if it fails to do so it will be communicated by means of satellite telephony to transmit the data, if it does not succeed because of being without coverage or being very low on the stack of containers, the data will be stored in flash memory waiting for coverage to be transmitted.

The TMS node receives the data from the Global Connect Cloud in XML or JSON format and performs the storage in the appropriate user accounts. The *Smart Contract* is used to record this information. In this way, each time a measure is received, the *Smart Contract* is executed.

From the above it follows that the following information will be stored in the underlying Blockchain:

- Assignment of BL.
- Non-compliance with Smart Contract.
- Termination of the service associated with the BL.

The minimun battery autonomy for the IoT device must be 30 days before recharging.

Test date: March 2019

MVP Characteristics Name: B. Global Connect Mesh Capture

Hypotheses:

1. Not all the containers will be able to transmit using mobile or satellite operators as there are piles of containers on top the others.

2. There is a need to receive the information of the conditions of transportation of their products for each and all their containers. Knowing that specific container is not meeting the rules of acceptance for a certain period of time can help to define an intervention or to check a specific container during the voyage in order to mitigate the situation. Also, could help to define during the journey the final use or commercialization of the products of the container avoiding therefore important losses.

Characteristics to solve the problem

Same characteristics as A. Global Connect Basic Capture, but additionally:

The IoT devices must send the captured data to the Global Connect Cloud in the Internet through, GPRS, 3G, or 4G data communications and a wireless network module 6loWPAN using the mobile or satellite operators.

The IoT device will communicate in the first instance with the mobile operator to transmit the data, if it fails to do so it will be communicated by means of satellite telephony to transmit the data, if it does not succeed because of being without coverage or being very low on the stack of containers, which prevents communication, will transmit the data to the nearest device through the 6loWPAN protocol (wireless network) using a mesh network mode, in case no nearby device can be found, the data will be stored in flash memory waiting for coverage to be transmitted.

Test date: April 2019

MVP Characteristics Name: C. Global Connect TMS Mobile Capture
Hypotheses: 1. There was no indication of costs considerations. Probably when the costs for the tracking service are know it Will be put into consideration.
Characteristics to solve the problem Same characteristics as B. Global Connect Mesh Capture, but additionally: The IoT devices must send the captured data to the Global Connect Cloud in the Internet through TMS, GPRS, 3G, or 4G data communications and a wireless network module 6loWPAN using the mobile operators. The TMS technology takes in a new way advantage of the signaling channel in the mobile network system to set up a call / message for communicating between the container cloud. The container is encoded with its own ID and specified message is encoded in a way that a set of calls Will represent that information. Each device has a CallID, an associated telephone number (SIM card. It must also be considered that the devices can consist of one or more sensors, for this reason the devices must be categorized, since the TMS Server (central station) must understand what IoT device sends data, what is the value of this data and its category, the category will tell us how many measurements (sensors) the IoT device contains, in such a way that the TMS knows how many measurements it should receive from the device, and by means of a formula scroll will be received.
Test date: June 2019

MVP Characteristics Name: D. Global Connect TMS Mobile and Satellite Capture
Hypotheses: 2. There was no indication of costs considerations. Probably when the costs for the tracking service are know it will be put into consideration.
Characteristics to solve the problem Same characteristics as C. Global Connect TMS Mobile Capture, but additionally: The TMS technology will take also the advantage of the signaling channel in the satellite mobile network system to set up a call / message for communicating between the container cloud.
Test date: July 2019

5. GLOSSARY OF TÉRMS

6loWPAN. IPv6 over Low Power Wireless Personal Area Networks
API. Application Programming Interface
ATA. Actual Time of Arrival (at Port of Destination)
ATD. Actual Time of Departure (from Port of Origin)
BLE. Bluetooh Low Energy
CEP. Complex Event Processing
CO2. Carbon Dioxide
CORS. Cross-Origin Resource Sharing
CPS. Cyber Physical System
CTS. Container Tracking Service
DAG. Directed Acyclic Graph
DLT. Distributed Ledger Technologies
DSNS. Domain Sensor Name Server
EPL. Event Processing Language
ETA. Estimated Time of Arrival (at Port of Destination)
ETD. Estimated Time of Departure (from Port of Origin)
GPS. Global Positioning System
HMI. Human Machine Interface
HTTP. Hypertext Transfer Protocol
HTML. HyperText Markup Language
IANA. Internet Assigned Numbers Authority
ICO. Initial Coin Offering
IDE. Integrated Development Environment
IEEE. Institute of Electrical and Electronics Engineers
IETF. Internet Engineering Task Force
IoE. Internet of Everythings
IoT. Internet of Things
ISO. International Organization for Standardization
JSON. JavaScript Object Notation
LPWAN. Low-Power Wide-Area Network
M2M. Machine to Machine
MNO. Mobile Network Operator
O2. Oxygen
OTT. Over The Top
REST. Representational State Transfer
RFC. Request for Comments
Router ABC. Always Best Connected
SATDR. Group of Distributed Real Time Systems and Applications
STH. Short Time Historic
TEU. Twenty-foot Equivalent Unit
TMS. Traveling Messaging System
WI-SUN. Wireless Smart Ubiquitous Network

6. BIBLIOGRAPHY

[1] https://blogthinkbig.com/8-400-millones-de-dispositivos-estaran-conectados-a-internet-a-finales-de-2017
[2] Government of the United States, Official information regarding the Global Positioning System, https://www.gps.gov/
[3] Adafruit Inudstries, Adafruit Ultimate GPS HAT for Raspberry Pi https://www.adafruit.com/product/2324
[4] Omega engineering, ¿Qué es un sensor termopar? https://es.omega.com/prodinfo/termopares.html
[5] K-type thermocouple reference tables: https://es.omega.com/temperature/pdf/Type_K_Thermocouple_Reference_Table.pdf
[6] Sensirion sensors, Digital Humidity Sensor SHT7x, https://www.sensirion.com/en/environmental-sensors/humidity-sensors/pintype-digital-humidity-sensors/
[7] Omega engineering, ¿Qué es un acelerómetro? https://es.omega.com/prodinfo/acelerometro.html
[8] Rs store, Sensor ultrasónico RS Pro, https://es.rs-online.com/web/p/sensores-de-proximidad-ultrasonicos/2370799/
[9] LoRa Alliance, What is the LoRaWAN Specification? https://www.lora-alliance.org/about-lorawan
[10] LoRa Alliance, A technical overview of LoRa and LoRaWAN, November 2015
[11] LoRa Alliance, LoRaWAN 101A Technical Introduction, 2015
[12] Sigfox, Sigfox technology overview https://www.sigfox.com/en/sigfox-iot-technology-overview
[13] Carles Antón-Haro and Mischa Doler, Machine-to-machine (M2M) Communications, 2015
[14] Pilar Andrés-Maldonado y PabloAmeigeras, NarrowBand IoT Data Transmission Procedures for Massive Machine Type Communications¸ Universidad de Granada
[15] Rohde&Schwarz, NarrowBand Internet of Things Whitepaper, https://www.rohde-schwarz.com/es/aplicaciones/internet-de-las-cosas-en-banda-estrecha-white-paper_230854-314242.html
[16] Activage Project, Report on IoT European Platforms, September 2017, https://iot-epi.eu/
[17] Documentación oficial de FIWARE Orion: https://fiware-orion.readthedocs.io
[18] Official documentation of FIWARE STH Comet https://fiware-sth-comet.readthedocs.io
[19] Official documentation of FIWARE Perseo (CEP) http://fiware-iot-stack.readthedocs.io/en/latest/cep/index.html
[20] MongoDB Inc, What is MongoDB? https://www.mongodb.com/what-is-mongodb
[21] University of Berkeley, Cyber-physical Systems https://ptolemy.berkeley.edu/projects/cps/
[22] Docker Inc., What is Docker, https://www.docker.com/what-docker
[23] Docker Inc., What is a container, https://www.docker.com/what-container
[24] Canonical Group, LXD Introduction, https://linuxcontainers.org/lxd/#
[25] Canonical Group, The no-nonsense way to accelerate your business with containers, February 2017
[26] Maersk Line, What is Remote Container Management, https://www.maerskline.com/shipping/remote-container-management
[27] Loginno, Smart container https://loginno.com/
[28] Traxens, TRAXENS Technology, http://www.traxens.com/en/technology
[29] Pointer Telocation LTD, Cellocator, https://www.cellocator.com/products/cellotrack/
[30] Tritón of GlobeArea, http://www.globearea.com/triton
[31] Weightless, http://www.weightless.org/
[32] Weight Sensor of Nanolike https://www.nanolike.com/fill-level-monitoring-solution/
[33] https://www.u-blox.com/en
[34] Blockstack https://bitcoinexchangeguide.com/kadena/
[35] Kadena https://kadena.io/
[36] https://www.plugintoiot.com/sub-1ghz-modules/#CC1310-module
[37] https://www.wi-sun.org/