

# Functional and technical specifications of “social media based Emergency DSS platform

---

Final Version of 27/04/2020

Deliverable Number D.3.3.1.

<b>Project Acronym</b>	E-CITIJENS
<b>Project ID Number</b>	10044361
<b>Project Title</b>	Civil Protection Emergency DSS based on CITizen Journalism to ENhance Safety of Adriatic Basin
<b>Priority Axis</b>	2
<b>Specific objective</b>	2.2
<b>Work Package Number</b>	WP3
<b>Work Package Title</b>	MODELLING "SOCIAL MEDIA BASED" CIVIL PROTECTION EMERGENCY MANAGEMENT SYSTEM
<b>Activity Number</b>	3.3
<b>Activity Title</b>	Design of hardware and software specifications of "social media based" Emergency DSS platform
<b>Partner in Charge</b>	PP5
<b>Partners involved</b>	LP, PP3,PP4, PP5
<b>Status</b>	Final
<b>Distribution</b>	Public

## Summary

1. Services description and data flow .....	3
2. Core module services for data injection .....	3
3. Services exposed to the decision maker .....	4
4. Output to the clients .....	5
5. Flow description .....	5
6. EDSS Architecture and partner's competences .....	7

## 1. Services description and data flow

The process of the core module of EDSS in charge to the LP can be described by listing the following features:

- **Receives formatted data** from PP4, exposing a series of Web Services (described below).
- **Analyzes and aggregates** data using part of models designed by PP5 (ALMA MATER STUDIORUM - UNIVERSITY OF BOLOGNA) during WP3, to extract structured datasets from text, images, videos, etc.
- **Persists** raw and structured data on a specific persistence layer.
- **Exposes Web Services** used by PP4's Decision Maker to obtain large amount of aggregated data (described below).
- **Uses aggregated data** and outputs from Decision Maker to fill a **Web Dashboard** (data visualization).
- Implements a **shared platform** giving access to different user profiles and granting them rights to access data.

## 2. Core module services for data injection

Going into details of exposed services and API, we can identify the **APIs that start decisional processes**:

- **CrawlerSocialInputAPI**: it's a Rest API exposed by LP (server side), through it PP4 (client side) will send data related to social networks posts, individually or aggregated using parameters like: socialNetworkName, userName, dateTime, textContent, tags, mediaContent, localization, interactions, etc.. The service will return OK or KO to PP4 as result of the received data.
- **CrawlerSensorInputAPI**: it's an Rest API exposed by LP (server side); through it PP4 (client side) will send data collected from sensors and probes, individually or aggregated, sending parameters like: sersorId, sensorName, localization, measuredValue, dateTime, measureUnit, etc.. The service will return OK or KO to PP4 as the result of received data.
- **CrawlerInstitutionalDataInputAPI**: it's a Rest API exposed from LP (server side); through it PP4 (client side) will send data collected from institutional sources, individually or aggregated, sending parameters like: source, value, measureUnit, textContent, mediaContent, etc. (more detailed informations about parameters will be available after the institutional source will be identified). The service will return OK or KO to PP4 as the result of received data.

While receiving input data, the EDSS core module (in charge at LP) will be responsible of the following activities:

- Persistency of raw data to the DB.
- Semantic analysis on the text contents from the social posts, using models in charge at PP5 or

- other known models.
- Analysis on multimedia contents using AI, based on machine learning models in charge at PP5 or other known models.
- Analysis on data from sensors and probes based on models in charge at PP5 or other known models.
- Analysis on data from institutional sources based on models in charge at PP5 or other known models.
- Aggregation of data received from the different sources based on each analysis results.
- Persistency of aggregated and analyzed data on the DB.

### 3. Services exposed to the decision maker

The EDSS core module in charge at LP, will expose other two services to another module in charge at PP4 which represent the effective Decision Maker used by the system.

The following API's could be some kind of synchronous request splitted in two singular steps (it may take some time between the request for the data set and the response prepared by the decision maker):

- **GetBulkAggregatedOutputData:** it's a REST API exposed by LP (server side); through it PP4 (DecisionMaker - client side) will request a bulk of aggregated data which can be used to take decisions (check if a potential dangerous event could be emerge from data related to a particular area, individuation of the interested geographical area, etc.). The API will use some parameters like the date range, type of data, max rows and so on (more details could be explained when the model will be more accurately). The API will respond with the requested aggregated data that fit a known transport model, sending the same dataset ID generated when the DM made the request. This API will be used periodically by the DM to maintain updated the elaborated data.
- **SetDecisionOnBulk:** it's and REST API exposed by LP (server side); through it PP4 (DecisionMaker - client side) will send a response containing the results from DM decisions, emergency related data obtained from aggregated data, emergency level, etc., in addition to the ID of the previously requested bulk input data.

PP4 should expose a service (server side) to allow EDSS to request and obtain data at any time:

- **SendBulkAggregatedOutputData:** it's a REST API exposed by PP4 (server side), by which LP (Decision Maker - client side) will send a bulk of aggregated data that require to be analyzed by DM (check if the data may give rise to dangerous situations in a particular area, locate the involved area, etc.). The API will send the aggregated data through a dedicated shared model, which contains also the ID of the dataset requests by the DM before. That API could be called

by the EDSS to allow a more precise analysis on a particular subset of data, for example if there are some evidences that could predict some kind of emergency. That API could be also called **TakeDecision** since it explicitly invites the decision maker to “take a decision” on input passed data.

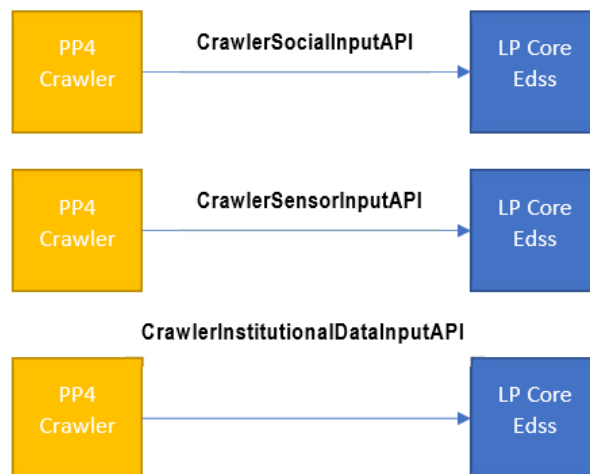
## 4. Output to the clients

With the data received as input, directly from social media, from sensors and institutional sources, or from the decision maker module, will be possible to view all the informations obtained from the system elaborations directly on a dashboard properly configured and profiled for every group or single user.

## 5. Flow description

The sequence of the communications between the modules, are described below:

1. The crawler implemented by PP4, can invoke whenever, in any order and in async mode any of the EDSS ingestion APIs.



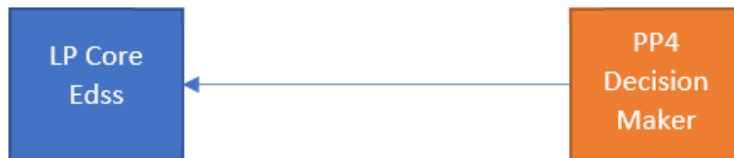
2. The Decision Maker for asks aggregated data to the core

**GetBulkAggregatedOutputData**



3. The EDSS decides autonomously to send data to the Decision Maker to request it's elaboration

**SendBulkAggregatedOutputData**



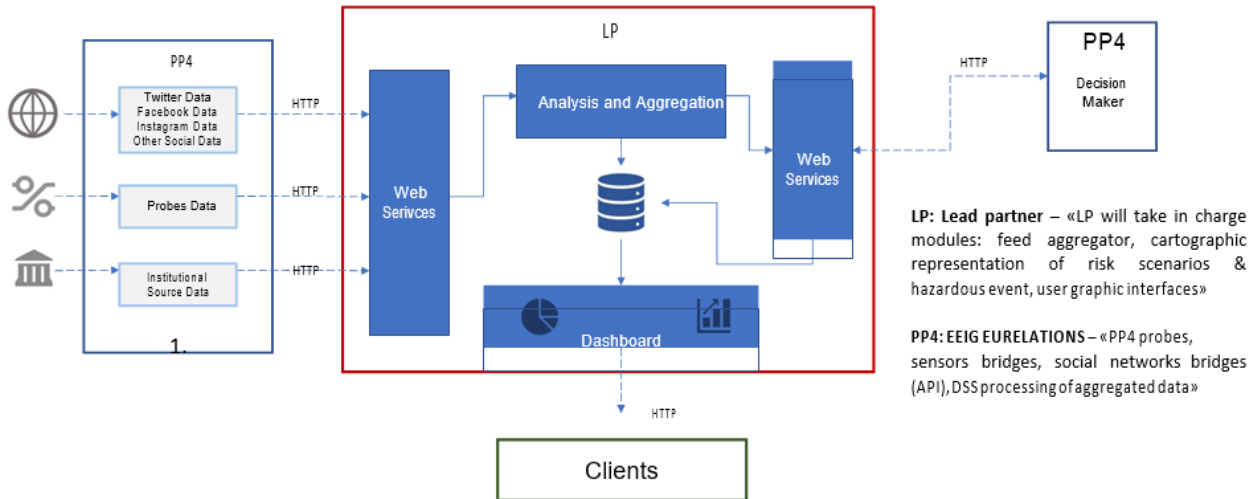
4. The Decision Maker sends the data to the EDSS core with the requested data relative response, including the ID of the bulk data used to make the decision

**SendBulkAggregatedOutputData**

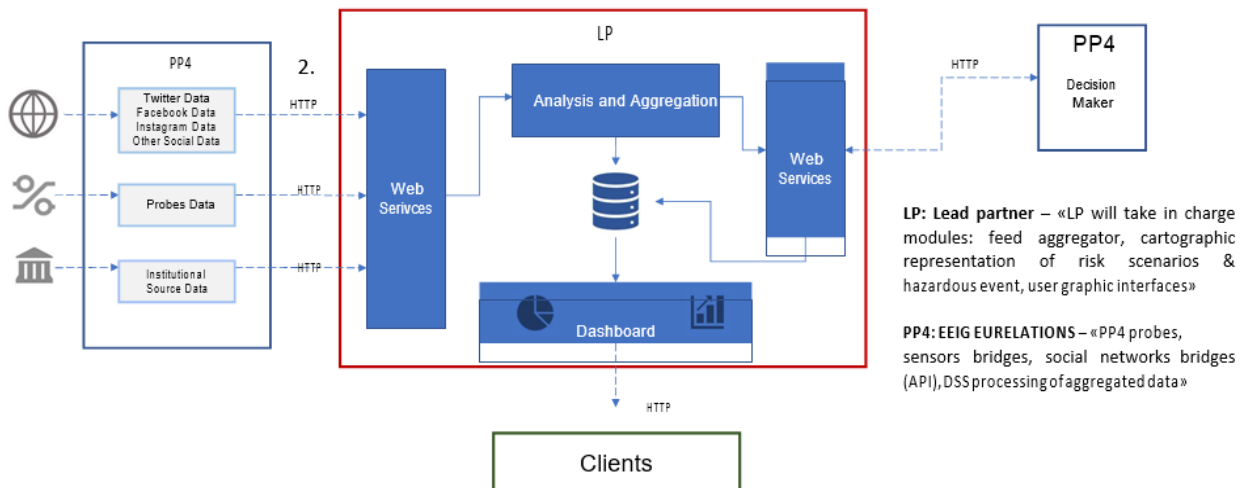


## 6. EDSS Architecture and partner's competences

### 1. PP4 Extracts data from social networks, probes and other institutional sources

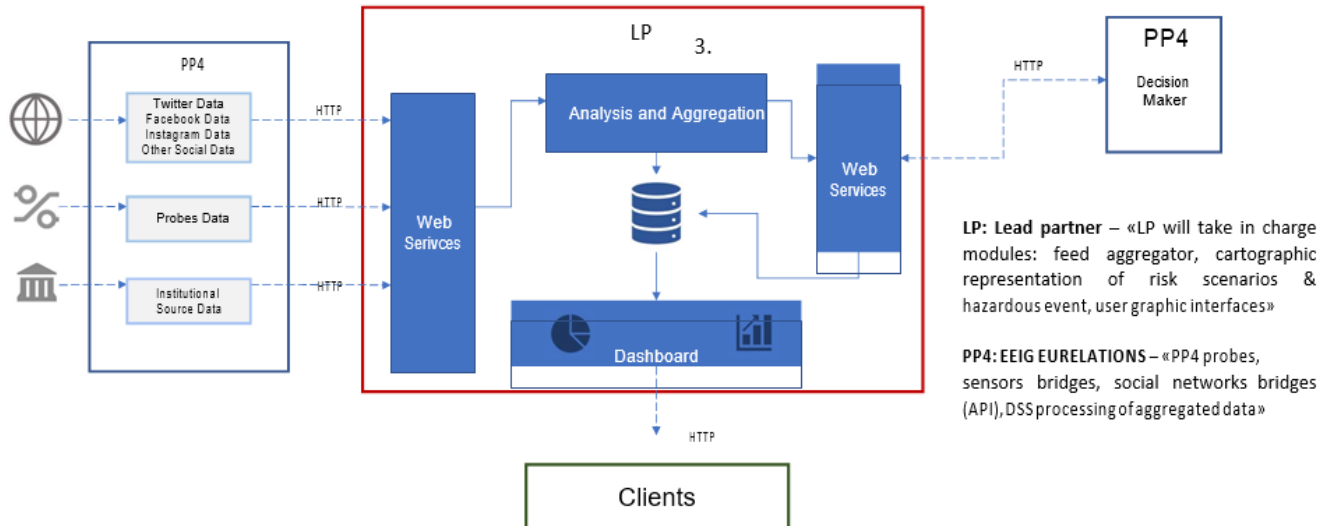


### 2. PP4 Sends data to LP Platform via Web Services

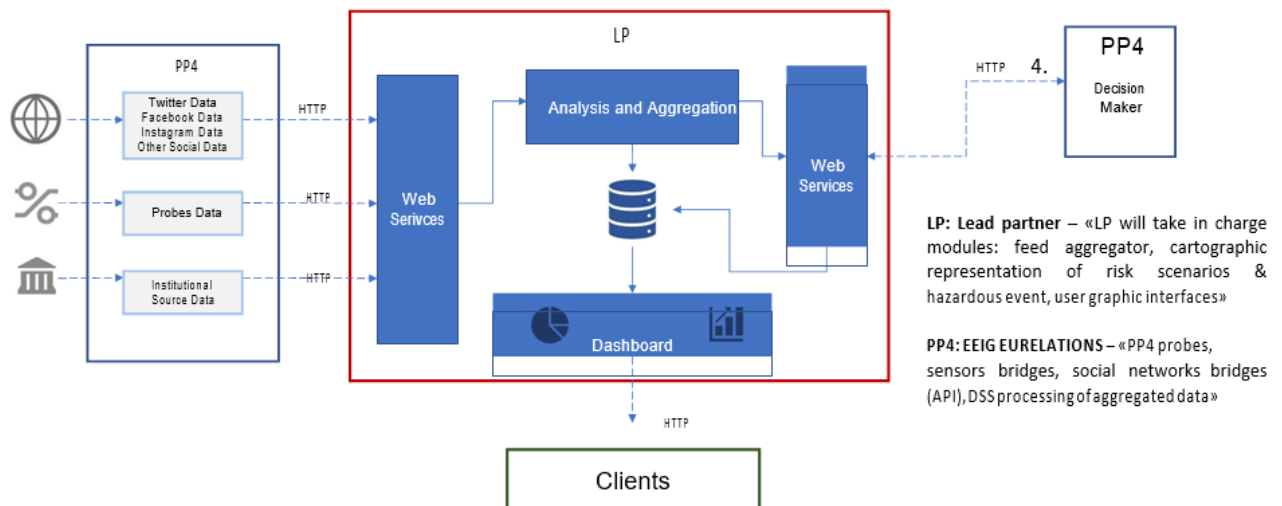




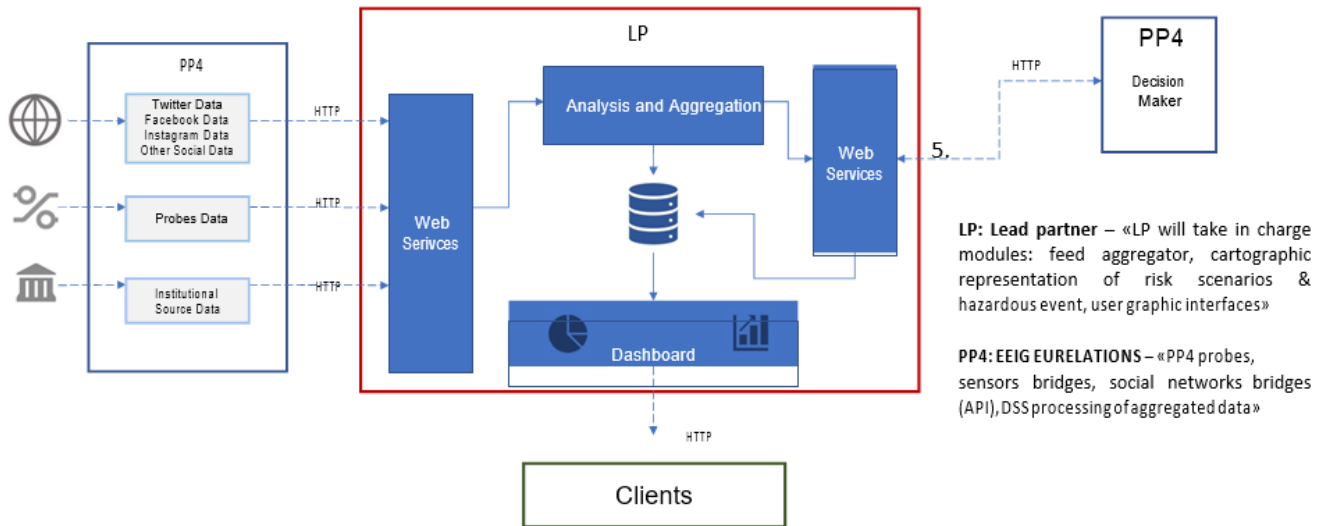
### 3. LP Platform analyzes, aggregates and persists data



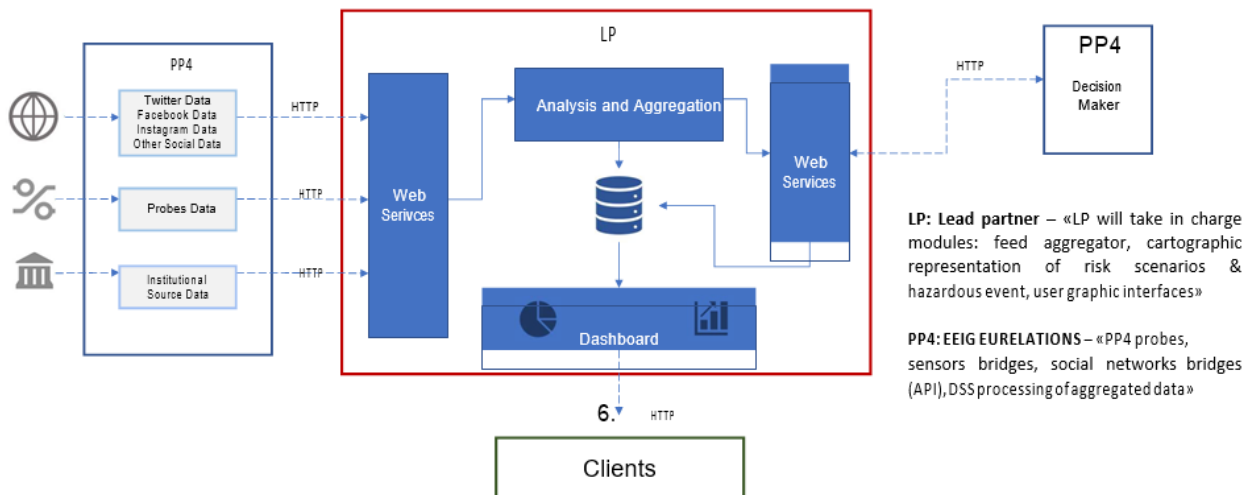
### 4. PP4 Decision Maker asks bulk of data to LP Platform



5. PP4 Decision Maker sends back the results of its elaboration to LP Platform



6. LP shows elaborated data on the dashboard to all clients connected



## PP4 Features

- *Crawler:*
  - **Extracts** data (user, content, media, datetime, localization, likes, etc.) from various social networks based on some keywords, hashtags or other configurable settings
  - **Format data** into a common, well known data model share with LP Platform
  - **Sends data** to LP Platform via Web Service
- *Decision Maker:*
  - **Interacts with LP Platform** masking bulk data, aggregated and formatted using a common, well known data model
  - **Analyze** these bulk data using complex model designed during WP3 by PP5 (ALMAMATERSTUDIORUM – UNIVERSITY OF BOLOGNA)
  - **Sends its analysis** to LP Platform via Web Services using a common, well known data model

## Lead Partner (Regione Molise) Features

- Implement a **common platform** accessed by different profiled users.
- Receives formatted data from PP4 exposing a **WebService**.
- **Analyzes and aggregates** these data using parts of model designed during WP3 by PP5 (ALMAMATERSTUDIORUM – UNIVERSITY OF BOLOGNA) trying to **extract structured information** from texts, images, video, etc.
- **Persists** raw and structured data into a persistence layer
- Exposes **Web Services** used by PP4 Decision Maker sending bulks of aggregated data
- Exposes **Web Services** used by PP4 Decision Maker to send elaborated data including suggestions and outputs produced by PP4 Decision Maker
- Used Aggregated data and PP4 Decision Maker output to populate a **web based dashboard** (data visualization feature).