



*Platform for European Medical Support
During Major Emergencies*

D4.2 IAT Tool





PULSE

Platform for European Medical Support during major emergencies

WP4 Tools

Deliverable D4.2-IAT Tool

30/11/2015



Contractual Delivery Date:	Actual Delivery Date:	Deliverable Type*-Security**:
30/11/2015	30/11/2015	R - PU
607799	PULSE	Platform for European Medical Support during major emergencies

*Type: P: Prototype; R: Report; D: Demonstrator; O: Other.

**Security Class: PU: Public; PP: Restricted to other programme participants (including the Commission); RE: Restricted to a group defined by the consortium (including the Commission); CO: Confidential, only for members of the consortium (including the Commission).

Responsible:	Organisation:	Contributing WP:
Malmignati Francesco	Selex ES	WP4

Authors (organisation)
Malmignati Francesco (Selex ES), Massimiliano Taglieri (Selex ES)

Abstract:
<p>This document is a report focused on the development of the Intelligent Analysis Tool (IAT). This the PULSE platform's component that is in charge of the generation of weak signals, such as the occurrences of an unusual biological event.</p> <p>This report provides a description of the objective that IAT aims to achieve and provides important details on functionalities, architecture, technologies and underlying libraries used for its implementation.</p>

Keywords:
Analysis, Signal, Crawling, NLP, CEP, UMLS



Revisions

Revision	Date	Description	Author (Organisation)
0.1	29/10/2015	First contribution on the component architecture and description of the relationship with the other components	Taglieri M. (Selex ES)
0.2	05/11/2015	First contribution regarding the component objective and functionalities	Francesco M. (Selex ES)
0.3	12/11/2015	Revision of chapters 5 and 6	Taglieri M. (Selex ES)
0.4	13/11/2015	Revision of chapters 3 and 4	Francesco M. (Selex ES)
1.0	16/11/2015	Editing of the document and definition of the final version	Francesco M. (Selex ES)
1.1	30/11/2015	Review of the document	Paul Kiernan (SKY)
1.2	30/11/2015	Addressing reviewer's comments	Massimiliano Taglieri (Selex ES) Francesco Malmignati (Selex ES)
1.3	30/11/2015	Final version of the document	Francesco Malmignati (Selex ES)



Table of contents

Revisions	3
List of figures	5
List of Tables	6
1 List of acronyms	7
2 Executive Summary	8
3 Introduction.....	9
3.1 Scope of the Document	9
3.2 Structure of the Document.....	9
3.3 Relations with other Deliverables	9
4 Tool description	10
4.1 Objective	10
4.2 Functionalities	10
4.2.1 Clinical record analysis	11
4.2.2 Twitter messages analysis	12
4.2.3 Web sites analysis.....	12
4.2.4 Weak signal generation.....	12
4.2.5 Scenario evolution.....	13
4.3 Relations with WP2 Use Cases.....	13
5 Architecture	15
5.1 Software Component Architecture.....	15
5.2 Component relation to overall PULSE architecture	17
6 Component Technologies	19
6.1 List of core technologies	19
6.2 3 rd Party libraries and licenses.....	20
References	21



List of figures

Figure 1: Clinical record - JSON example.....	11
Figure 2: Intelligence and Analysis Tool Architecture – Comp. diagram	15
Figure 3: Intelligence and Analysis Tool Architecture – Schema.....	16
Figure 4: PULSE Platform Architecture – Component diagram.....	18



List of Tables

Table 1 - 3 rd party libraries and licenses.....	20
---	----



1 List of acronyms

Acronym
CEP
DSVT
ECDC
EHR
GUI
IAT
JSON
NLP
REST
UMLS
WHO



2 Executive Summary

This document is a report focused on the development of the *Intelligent Analysis Tool* (IAT), one of the software components composing the PULSE Platform Architecture.

The tool's objective is to provide an early warning system that is able to alert decision makers to the occurrence of an unusual biological event. The IAT in fact is able to systematically gather and analyse incoming disease-related data and to notify the presence of possible epidemic's breeding grounds. In particular, the Intelligent Analysis Tool is able to extract information regarding the disease symptoms from (1) clinical records coming several selected hospitals, (2) geo-localized tweet messages generated from the Twitter platform and (3) web sites, specialized blogs, news containing disease-related information. If the number of persons suffering from these acquired disease symptoms is above a predefined threshold, then the IAT generates a weak signal.

The architecture of the tool is based on 3 main components:

- A *Language Text Analyser*, based on the Natural Language Processing (NLP) [7], which is able to analyse the text coming from Twitter and external web sites;
- A *Clinical Records Analyser*, based on NLP and the UMLS Metathesaurus [6], which is able to analyse the hospitals' clinical records;
- A *Complex Event Processing* [18] Engine, which is able to combine the incoming data streams and to infer patterns that could indicate the high probability of an epidemic's breeding ground.

The generated weak signal, represented as a JSON object, is then delivered to the Decision Support Validation tool, that visualize it on a GUI and allows the user to classify it as relevant or not relevant.



3 Introduction

3.1 Scope of the Document

This document is a covering document to support the software delivery of the *Intelligence Analysis Tool* (IAT).

As stated in [1], the development of the *IAT* prototype is part of WP4 activities and this software component has to be delivered as 'D4.2 – Intelligence Analysis Tool'.

This report summarizes the software component delivery and provides high-level details on the architecture, technologies and underlying libraries used for its implementation.

3.2 Structure of the Document

This document is structured into the following sections.

- Description of the *Intelligence Analysis Tool* objective.
- Main functionalities of the IAT.
- Description of the *Intelligence Analysis Tool* internal architecture and relation of this component with other elements of the PULSE platform architecture;
- List of technologies adopted for the implementation of *the Intelligence Analysis Tool*
- List of underlying 3rd party libraries used for *Intelligence Analysis Tool* implementation and summary of the corresponding licenses

3.3 Relations with other Deliverables

The work presented in this deliverable is related to the following WP2 and WP4 deliverables:

- D2.2 - Use case specification [2] – This document describes the use cases related to two PULSE reference scenarios: SARS-like epidemics and Stadium crush.
- D4.1 – Decision support Validation tool [3] – It is the tool that invokes the API provided by the IAT and provides a graphic front-end for the management of the generated weak signals.
- D4.7 – Event evaluation for biological event [4] – The ENSIR tool is invoked by the IAT to provide the scenario evolution functionality



4 Tool description

4.1 Objective

One of the objective of the PULSE platform is to provide an early warning system that is able to alert decision makers to the occurrence of an unusual biological event.

An early warning system is a surveillance system that continuously collects information on specific epidemic diseases in order to notify for a possible presence of epidemic signals in specific zones and then asking for a prompt public health intervention. Outbreaks of infectious disease can spread rapidly, causing enormous losses to health and livelihood. The best strategy to stop their spread is to intervene at an early stage, when an epidemic is still at an embryonal phase and it would be easier to prevent its spreading.

Currently, the existent surveillance systems rarely apply systematic methods to detect changes in trends. In most cases they just rely on an in-depth review done by epidemiologists of the incoming data, thing which is rarely done in a systematic way.

In PULSE we tried to innovate this approach and to design and develop a tool, called Intelligence Analysis Tool (IAT), that is able to systematically gather and analyse incoming disease-related data and to notify the presence of possible epidemic weak signals.

The incoming data that the IAT currently supports are:

- Clinical records coming several selected hospitals,
- Geo-localized tweet messages generated from the Twitter platform
- Web sites, specialized blogs, news containing disease-related information

The Intelligence Analysis Tool is then able to

1. analyse the above-mentioned incoming data
2. extract information related to the disease symptoms (e.g. for SARS, cough, high fever etc.)
3. generate a signal if e.g. the number of persons suffering from the selected disease symptoms is above a predefined threshold.

In the following section, we will go into details and we will provide an exhaustive description of these functionalities provided by the IAT.

4.2 Functionalities

IAT provides different functionalities that support the generation of weak signals that indicates the possible presence of epidemic's breeding grounds. All these functionalities are made available to other components of the PULSE architecture (e.g. DSVT) through a standard Web Service RESTful interface. As defined in the REST specification [5], all the messages that are actually exchanged between the Intelligence Analysis tool and the other platform's tools are HTTP-based requests and responses. They include a message body whose information content is represented in JavaScript Object Notation (JSON).



4.2.1 Clinical record analysis

As said in 4.1, one of the data taken as input by the IAT are the clinical records provided by selected hospital (e.g. hospitals that decided to provide their clinical records to the PULSE platform). These data are crucial to the IAT since they provide the essential mean for a deep understanding of the persons' health conditions in a specific zone.

With the widespread usage of electronic health record (EHR), several innovative technologies had been introduced to improve the understanding of the patients' conditions. We decided to use the *Natural language processing* (NLP) that allows us to (1) process sentences in a natural language (e.g. English), (2) to extract meanings from EHRs and (3) to convert free text narratives into structured, standardized formats.

The clinical records' texts are often ungrammatical, consist of short phrases usually without an extended context and make use of a lot of acronyms and abbreviations. For this reason we based our NLP analysis on the Unified Medical Language System (UMLS) Metathesaurus [6], which is a repository of over 100 biomedical vocabularies. Within the Metathesaurus, terms across vocabularies are grouped together based on meaning, forming concepts. Each individual concept is assigned multiple semantic type categories from the UMLS Semantic Network.

The NLP analysis performed by the IAT aims at extracting, from all the incoming data, a subset of clinical records where symptoms related to a specific disease have been notified. This stream will be then correlated by the IAT with the other incoming data streams (e.g. Twitter messages, crawled web sites), trying to find a possible connection within these data. This connection, if present, will be evaluated by the IAT in order to verify the presence of a rule that will send a weak signal message to the DSVT. More info about this evaluation process can be found in section 4.2.4.

The clinical record can be delivered to the IAT by simply sending a POST message to the following URL: <http://hostname:port/crt>

The message is composed of a JSON file containing the text of the clinical record and the GPS coordinates of the hospital that released the record. Figure 1 shows an example of this JSON message.

```
{
  "id": "AAS",
  "text": "Dr. Nutritious Medical Nutrition Therapy for Hyperlipidemia Referral from: Julie Tester, RD, LD, CNSD Phone contact: (555) 555-1212 Height: 144 cm Current Weight: 45 kg Date of current weight: 02-29-2001 Admit Weight: 53 kg BMI: 18 kg/m2 Diet: General Daily Calorie needs (kcal): 1500 calories, assessed as HB + 20% for activity. Daily Protein needs: 40 grams, assessed as 1.0 g/kg. Pt has been on a 3-day calorie count and has had an average intake of 1100 calories. She was instructed to drink 2-3 cans of liquid supplement to help promote weight gain. She agrees with the plan and has my number for further assessment. May want a Resting Metabolic Rate as well. She takes an aspirin a day for knee pain.",
  "latitude": 42.0,
  "longitude": 12.15
}
```

Figure 1: Clinical record - JSON example



4.2.2 Twitter messages analysis

Social networking sites as Twitter have become extremely powerful tools, used by people all around the globe to share every aspect of their life. It is used to express feelings, thoughts, images, links and a lot more. In the PULSE context, we thought that being able to acquire and analyse messages exchanged by the people through similar platforms is an important way to help understanding the actual health conditions of the people. The IAT is the tool in charge of this functionality since it is able to (1) gather messages coming from the Twitter platform, (2) filter these messages taking only those that have attached geo-localized information and (3) analyse them in order to check if they contain specific keywords related to a disease symptoms (e.g. cough, high fever for SARS disease).

As said in 4.2.1, this stream is correlated with other data streams (e.g. clinical data records, crawled web sites) trying to find a possible connection within these data and then to generate a possible weak signal.

4.2.3 Web sites analysis

The IAT foresees the analysis of another data stream concerning in this case the content of specific online web sites. The Intelligence Analysis actively and continuously crawls the web to extract pertinent content from predefined web sites URLs. This content is then internally analysed by the IAT in order to check if the document is “talking” about a monitored disease. As described in 4.2.1 for the clinical records, also in this case the IAT makes use of the Natural Language Processing, specifically the Text Classification methodology [7], to check whether the extracted text can be classified as “disease-related” or not.

The starting crawled web sites URLs can be manually defined by the user in order to give a complete control over the possible web sites that the IAT should monitor.

Example of web sites that could be monitored are:

- News web sites
- Blogs
- Specialized medical web sites
- WHO web site
- ECDC web site

The content classified as “disease-related” is the correlated with the other data streams (e.g. clinical data records, twitter messages) to find a possible connections and, if available, to generate a weak signal.

4.2.4 Weak signal generation

As explained in 4.2.1, 4.2.2 and 4.2.3, three different data streams are gathered by the IAT to acquire information regarding the status of the people’s health conditions.

These data streams are then tracked and analysed by the IAT through an internal *Complex Event Processing* (CEP) [18] whose goal is to combine these streams and to infer patterns that could indicate the high probability of an epidemic’s breeding ground.

These CEP patterns can be customized by the PULSE users through the definition of simple rules following the *if-then-else* pattern. In the following you can see an example



of rule:

- **if**
 - during the last day and in a certain location (described as a circumference, with the centre specified with some specific GPS coordinates and a radius specified in km), 10 Twitter messages containing the SARS symptoms keywords (e.g. cough, high fever) have been notified;
 - during the last day, one article that talks about that epidemic symptoms in that location has been notified;
 - during the last day and in a certain location, 10 clinical records containing the epidemic symptoms have been notified;
- **then**
 - trigger a weak signal and continue monitoring
- **else**
 - continue monitoring

Therefore, the generated weak signal is delivered to the DSVT as described in D4.1 [3].

The weak signal is described as a JSON object and contains:

- an id, to identify each signal;
- a description of the signal;
- the text of the web sites that generated the alert, if any;
- the URLs of the web sites, if any;
- the text of the clinical records that generated the alert, if any;
- the list of symptoms that have been observed in the clinical records, if any;
- the GPS coordinates of the hospitals that sent the clinical records, if any;
- the text of the Twitter messages that generated the alert, if any;
- the list of symptoms that have been observed in the Twitter messages, if any;
- the GPS coordinates of the Twitter messages, if any.

The IAT allows to retrieve the list of the rules that are currently configured in the system by simply doing a GET on the following url <http://hostname:port/cepcfg/rules>.

4.2.5 Scenario evolution

The IAT provides also support the decision making process by providing a prediction on the evolution of the scenario. In our SARS-related scenario, we considered the evolution prediction as the simulation of the possible spread of the disease. In the PULSE platform this functionality is provided by the ENSIR tool, and in such sense, the IAT exploits the ENSIR's service to provide the scenario evolution feature. The ENSIR tool functionalities are described in D4.7 [4].

4.3 Relations with WP2 Use Cases

The Intelligence Analysis Tool has a strong relation with just one of the use cases defined in D2.2 [2], specifically the UC-SARS LIKE – 01.

In this use case a weak signal is notified to the competent authorities that will take decisions based on the WHO and ECDC guidelines. As it is easy to see, the IAT is involved in the weak signal generation step, specifically in the point 1 of the Basic Path. This weak signal is generated by analysing news articles from specialised



official and unofficial medical sites, blogs and online newspapers as it has been described in section 4.2.

5 Architecture

5.1 Software Component Architecture

The component diagram and the schema in Figure 2 e Figure 3 show the internal architecture of *IAT*. This software component is composed of several sub-modules that interact to produce, at the final stage, weak signals about possible outbreaks of SARS-like epidemics. More details on this kind of signals can be found in section 4.2.

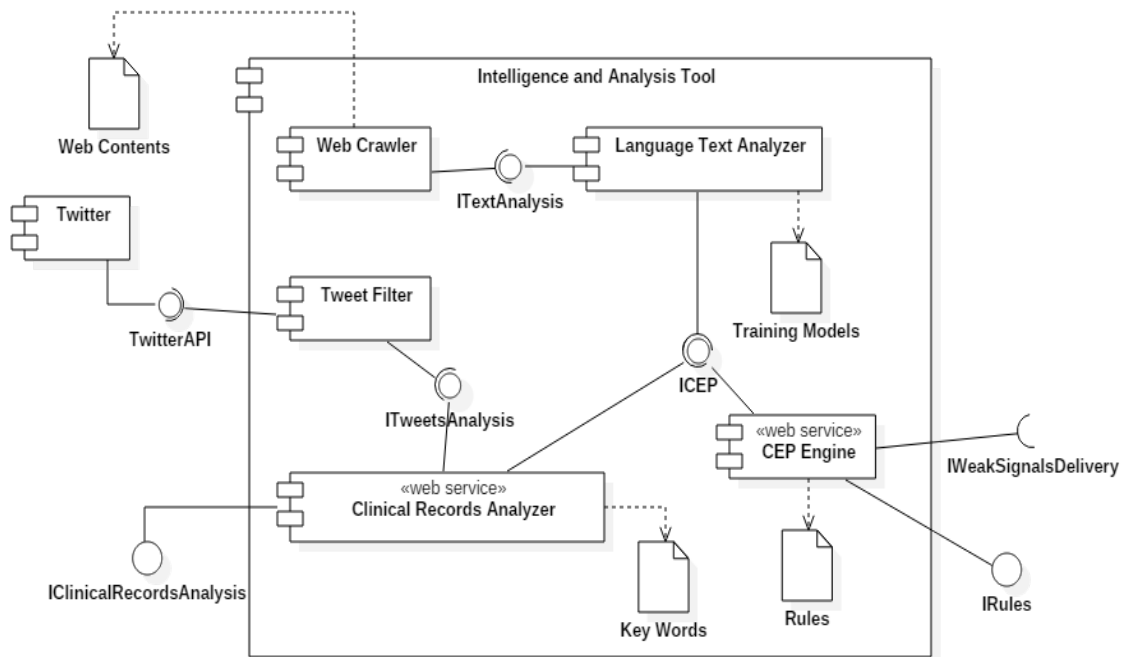


Figure 2: Intelligence and Analysis Tool Architecture – Comp. diagram

The *Web Crawler* is the first of these internal sub-modules; it implements a search engine that explores web contents on the Internet and extracts text documents according to pre-configured guidelines. The *Web Crawler* configuration consists in setting up two types of information that, in turn, are used to guide the exploration on the Internet. The first information is represented by the list of websites whence the crawler has to start the search for relevant documents. This list is used to define the starting points of the search on known sites whose contents are focused on SARS topics. The second information is instead the list of web domains that the *Web Crawler* should consider as valid during the exploration process. These parameters allow, in some way, to limit the extraction of text documents only to those sites whose domains are related to SARS epidemics.

After this short explanation, let's see how the *Web Crawler* works. It begins the searching process starting from the list of pre-configured web sites and continues the exploration in a recursive manner (i.e., site by site), by following the outgoing hyperlinks if and only if they fall in one of the pre-configured web domains. The process ends when all the available hyperlinks, fulfilling the condition on the web domain, have been explored and the relative text documents have been sent to the *Language Text Analyzer*.

The *Language Text Analyzer* is an intermediate sub-module that implements a tool for classifying text by using the *Natural Language Processing (NLP)*. It exposes the *ITextAnalysis* interface that is used by the *Web Crawler* to send streams of text documents.

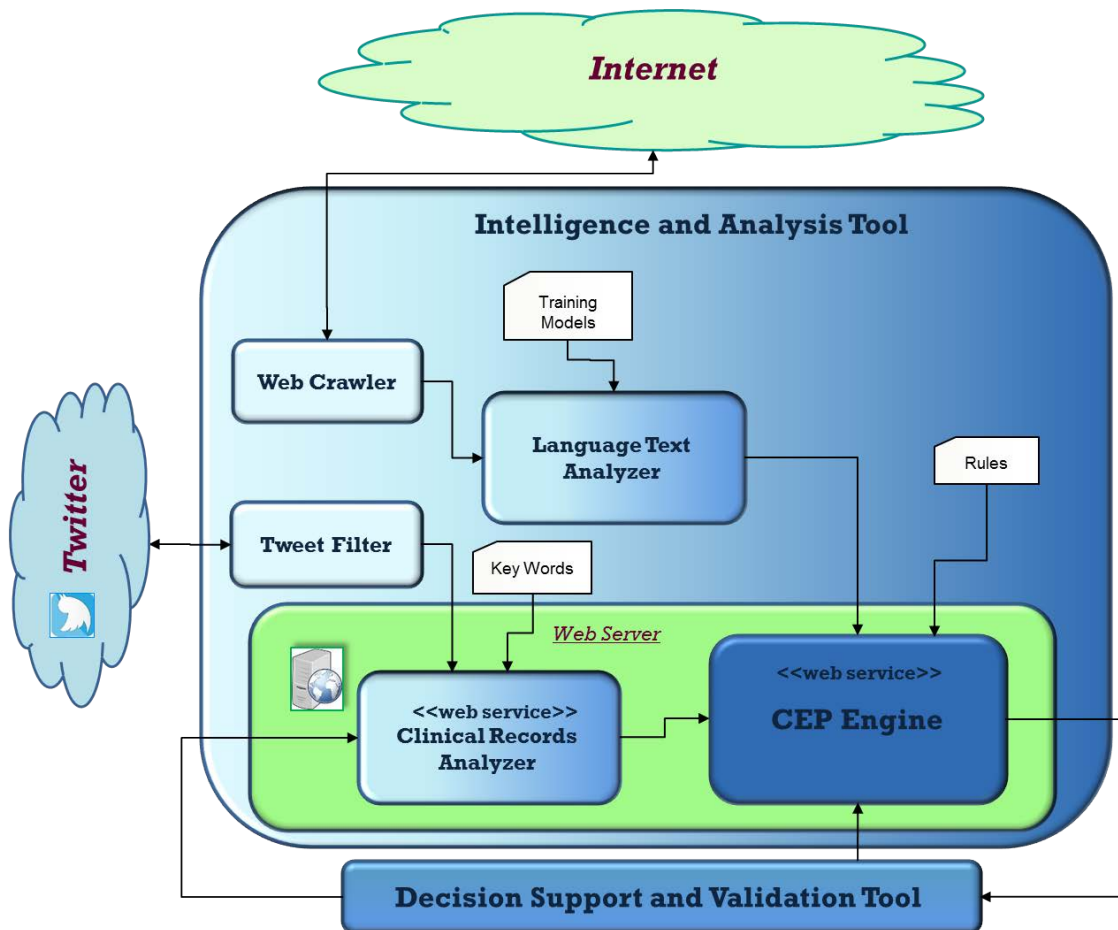


Figure 3: Intelligence and Analysis Tool Architecture – Schema

The linguistic analysis carried out by the *Language Text Analyzer* depends on particular *Trainings Models* which are required to configure the module for recognizing the text documents containing information pertaining to SARS symptoms.

The aforementioned input streams are processed according to *Training Models*, so that output streams only consist of text documents containing information about SARS symptoms. These output streams are sent to the *CEP Engine* for further elaborations.

The *Tweet Filter* is another *IAT* internal sub-module; it implements a Twitter client using the REST API [8] made available by *Twitter*. The main objective of this sub-module is to search tweets including geolocation information. Once collected, this information is sent to the *Clinical Records Analyzer* through the *ITweetsAnalysis* interface.

The *Clinical Records Analyzer* is an internal sub-module performing two core tasks: (i) a linguistic analysis on streams of tweets and (ii) the analysis of electronic medical records to recognize those reporting data related to symptoms caused by SARS infection.



The first process depends on *Key Words* which are used to filter tweets coming from the *Tweet Filter*, by selecting only those related to SARS symptoms. The generated output stream contains just tweets with information about SARS symptoms. These data are sent to the *CEP Engine*.

The second analysis allows to recognise the presence of clinical SARS symptoms in the incoming medical records. This is performed by exploiting the *Unified Medical Language System* (UMLS) [6] and its internal dictionary. If SARS symptoms are observed, the medical record is classified as probable SARS record and therefore it is sent to the *CEP Engine* for further elaborations.

The *Clinical Records Analyzer* implements a Web Service equipped with a RESTful interface (indicated as *IClinicalRecordsAnalysis* in Figure 2) that allows the hospitals to send the owned medical records through the *Decision Support and Validation Tool* (DSVT). As highlighted in Figure 3, the *Clinical Records Analyzer's* Web Service is made available online by means of a dedicated Web Server.

The *CEP Engine* is the sub-module that closes the data stream's processing chain crossing the *IAT*. It is a complex event processing and correlation engine that analyses the input streams coming from *Language Text Analyser* and *Clinical Records Analyzer*. The purpose of this elaboration process is to derive weak signals about possible outbreaks of SARS-like epidemics.

As stated in 4.2, the data streams provided by the previous sub-modules to the *CEP Engine* are series of:

1. text documents reporting SARS related information collected on the Internet;
2. tweets containing key words on SARS symptoms (each tweet is also accompanied with geolocation information);
3. electronic medical records concerning probable SARS cases.

These series are filtered, aggregated, and joined, possibly over temporal windows or sliding windows of multiple data. This elaboration can be configured by defining apposite rules that allows to specify the logic to derive weak signals from the input series. More details on these rules can be found in section 4.2.4.

Similarly to the *Clinical Records Analyzer*, the *CEP Engine* implements a Web Service with a RESTful interface (indicated as *IRules* in Figure 2) that is used to setup the configuration rules via *DSVT*. This Web Service is made available online with the same Web Server used for the *Clinical Records Analyzer*.

The weak signals resulting from the elaboration of the input streams are sent to the *DSVT* through the *IWeakSignalsDelivery* interface.

5.2 Component relation to overall PULSE architecture

Figure 4 shows a component diagram of the PULSE Platform Architecture where the relation between *IAT* and *DSVT* is highlighted. As already mentioned in section 5.1, these two components interact each other through three interfaces: *IRules*, *IClinicalRecordsAnalysis* and *IWeakSignalsDelivery*. In particular, *DSVT* invokes the *IAT* Web Services in the following cases:

- every time a user wants to customize the elaboration of the data streams crossing the *IAT*, with the purpose of obtaining weak signals with specified features; in this case, the user sets up the *CEP Engine's* configuration rules via *DSVT* that, in turn, invokes the *IRules* interface;



- whenever a hospital needs to send electronic medical records to *IAT*; in this case the owned data are also sent via *DSVT*, but this time *DSVT* invokes another Web Service interface i.e. *IClinicalRecordsAnalysis*.

In addition to these situations, there exists the case in which data flow from *IAT* to *DSVT*. In fact, the *IAT* can invoke the *IWeakSignalsDelivery* interface every time a new weak signals is generated. The *DSVT* then classifies the received signals either as ‘relevant’ or ‘non-relevant’. For more detail on how this classification process works, refer to D4.1 [9].

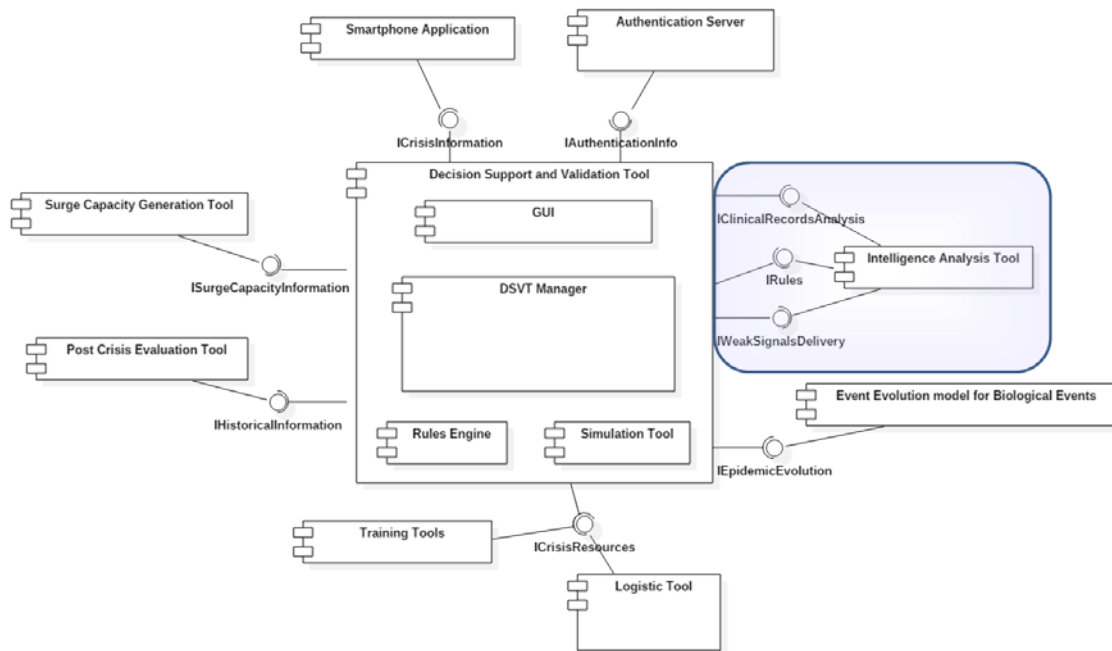


Figure 4: PULSE Platform Architecture – Component diagram



6 Component Technologies

6.1 List of core technologies

As stated in section 5.1, *IAT* is composed of several sub-modules: *Web Crawler*, *Language Text Analyzer*, *Tweet Filter*, *Clinical Records Analyzer* and *CEP Engine*. The core technologies adopted for the implementation of these sub-modules are:

- *Norconex HTTP Collector* – It is an open source enterprise web crawler that collects web sites contents for search engines or any other data repository [11]. This tool is used to implement the *Web Crawler* and, in particular, the functionalities that allow to explore the Internet to retrieve SARS related text documents.
- *Twitter REST API* – It is a REST API that provides programmatic access to read and write Twitter data [11]. This API is used by the *Tweet Filter* to search tweets including geolocation information.
- *LingPipe* – It is a kit for processing text using computational linguistics. *LingPipe* provides a classification facility that is able to analyse models of text classifications and then learns how to classify further documents by using what it can deduce from the aforementioned models [12]. This technology has been chosen for the implementation of the *Language Text Analyzer*. *Training Models* are the models used to instruct the analyzer to recognize text documents reporting SARS symptoms.
- *CTakes* – It is a natural language processing system for extraction of information from electronic medical records [13]. *CTakes* processes clinical notes by identifying types of medical entities such as drugs, diseases or disorders, signs or symptoms, anatomical sites and procedures. This technology has been adopted for implementing the *Clinical Records Analyzer* as it simply allows to recognize the presence of clinical SARS symptoms in the incoming tweets and medical records.
- *Esper* – It is a component for complex event processing (CEP) and event series analysis, available for Java applications. *Esper* enables the elaboration of large volumes of incoming messages or events, regardless of whether incoming messages are historical or real-time in nature. It filters and analyzes events in various ways and is able to fast derive aggregated conclusions from them [14]. For this reason, it represents the core technology adopted for the implementation of the *CEP Engine*.
- *Jetty* – It is a Java-based Web Server used to make available online the Web Services of *Clinical Records Analyzer* and *CEP Engine*. It implements an HTTP server and servlet container capable of serving static and dynamic content either from a standalone or embedded instantiations. This technology is developed as a free and open source project as part of the Eclipse Foundation [15].
- *Jersey* – It is an open source framework [16] that provides an API extending the JAX-RS toolkit [17] with additional functionalities, with the purpose of simplifying the implementation of RESTful services. This technology is used to build the Web Service RESTful interface offered by *Clinical Records Analyzer* and *CEP Engine*.



6.2 3rd Party libraries and licenses

Table 1 reports a list of third party libraries/frameworks used for the IAT implementation and the licenses under which they are distributed.

Table 1 - 3rd party libraries and licenses

Product	Version	Vendor	License
Norconex HTTP Collector	2.3.0	Norconex	Apache 2.0
Twitter4J	4.0.4	Twitter4J	Apache 2.0
LingPipe	4.1.0	Alias-i	GNU AGPL 3
cTAKES	3.2.2	Apache Software Foundation	Apache 2.0
Esper	5.3.0	EsperTech Inc.	GNU GPL 2
Jetty	9.2.10	Eclipse	Apache 2.0
Jersey	2.17	Oracle Corporation	Dual License: - CDDL 1.1 - GNU GPL 2



References

- [1] PULSE Project – Description of Work, version 1.0, October 2013.
- [2] PULSE Project Deliverable – D2.2 Use case specification
- [3] PULSE Project Deliverable – D4.1 Decision Support Validation tool
- [4] PULSE Project Deliverable – D4.7 Event evaluation for biological event
- [5] Fielding, Roy Thomas (2000). "Chapter 5: Representational State Transfer (REST)". Architectural Styles and the Design of Network-based Software Architectures (Ph.D.). University of California, Irvine
- [6] Unified Medical Language System (UMLS), <https://www.nlm.nih.gov/research/umls/>
- [7] F. Sebastiani. 2002. Machine learning in automated text categorization. ACM Computing Surveys 34(1):1--47.
- [8] REST APIs by Twitter Inc. – Homepage – <https://dev.twitter.com/rest/public>.
- [9] PULSE Project Deliverable – D4.1 Decision Support Validation Tool.
- [10] Norconex HTTP Collector by Norconex – Homepage – <https://www.norconex.com/collectors/collector-http>.
- [11] Twitter4J, Unofficial Java Library for Twitter API by Twitter4J – Homepage – <http://twitter4j.org/en/index.html>.
- [12] LingPipe by Alias-i – Homepage – <http://alias-i.com/lingpipe>.
- [13] cTakes by Apache Software Foundation – Homepage – <http://ctakes.apache.org>.
- [14] Esper CEP for Java by EsperTech Inc. – Homepage – <http://www.espertech.com/products/esper.php>.
- [15] Jetty Project by Eclipse Foundation – Homepage – <http://www.eclipse.org/jetty>.
- [16] Jersey by Oracle Corporation – Homepage – <https://jersey.java.net>.
- [17] JAX-RS Project by Oracle Corporation, Project Kenai and Cognisync – Homepage – <https://jax-rs-spec.java.net>.
- [18] D. Luckham, "The Power of Events: An Introduction to Complex Event Processing in Distributed Enterprise Systems", Addison-Wesley, 2002.