Abstract

Nowadays, information is one of the main resources of any company and plays an important role in decision making. For top management teams, obtaining important information as fast as possible is a priority, which can become challenging when there is a lot of data to process. Complex Event Processing (CEP) engines are able to process incoming data stream and extract important information while filtering non-relevant data. Thus, CEP engines are capable of analyzing thousands of data records very quickly, reducing the latency between receiving data and processing it, making it appropriate to be used for decision making. This operational capability, we consider, could benefit any data management system built today. But, there are diverse types of information systems, applied to a variety of enterprise areas, operating in different environments as well as requiring numerous methods to ensure their correct operability. One type of those systems are the Safety-critical systems, responsible for infrastructures with great impact and that could possibly cause high damage to people, society or environment. Safety-critical systems are responsible for performing operations in critical environments, such as, water storage, petroleum and atomic stations, vehicle systems, avionics, medical devices, etc. Since those systems are required to generate response and alerts in real-time, it is clear that CEP engines can be a solution for their performance goals. But, safety-critical systems have other, and more important, quality attributes such as security, dependability and safety. In this work, we investigate if CEP engines can be used for safety-critical applications and are able to cope with the inevitable quality attributes of these systems. After describing CEP engines, safety-critical systems and their quality attributes we focus on safety and security and provide a solution for data authenticity as mechanism added to one of the most popular CEP engines, ESPER. We conclude that proposed solution provides data authenticity but also has considerable performance impact.

Keywords: CEP, Safety-critical, sensor networks

Resumo

Nos dias de hoje, a informação é um dos principais recursos de qualquer empresa e desempenha um papel importante na tomada de decisão. Para as equipas de gestão, a obtenção de informações importantes o mais rápido possível é uma prioridade, que pode se tornar desafiadora quando existem muitos dados a serem processados. Os mecanismos complexos de processamento de eventos (CEP) são capazes de processar o contínuo fluxo de dados, separando a informação e filtrando os dados de pouca relevância. Assim, os sistemas da CEP são capazes de analisar milhares de registros de dados de forma muito rápida, reduzindo o atraso entre a receção e processamento de dados, tornando-se conveniente para a tomada de decisão. Acreditamos que esta capacidade operacional, poderia beneficiar qualquer sistema atual de gestão de dados. Mas, existem diversos tipos de sistemas de informação, aplicados a uma variedade de áreas empresariais, que operam em diferentes ambientes, além de exigir inúmeros métodos para garantir sua correta operacionalidade. Um tipo desses sistemas, são sistemas críticos, responsáveis por infraestruturas com grande impacto e que podem causar danos elevados às pessoas, à sociedade ou ao meio ambiente. Os sistemas críticos são responsáveis pela realização de operações em ambientes críticos, tais como armazenamento de água, estações de petróleo e atómicas, sistemas de veículos e aviários, dispositivos médicos, etc. Uma vez que esses sistemas são necessários para gerar resposta e alertas em tempo real, é possível que os motores CEP podem ser uma solução para melhorar o desempenho desses sistemas. Mas, os sistemas críticos possuem outros atributos de qualidade, como a proteção, a confiabilidade e a segurança. Neste trabalho, investigamos se os motores CEP podem ser usados em sistemas críticos e se são capazes de lidar com os atributos de qualidade desses sistemas. Depois de descrever os motores CEP, os sistemas críticos e seus atributos de qualidade, nos concentramos em segurança e proteção e fornecemos uma solução para a autenticidade de dados como mecanismo adicionado a um dos motores CEP mais populares, o ESPER. Concluímos que a solução proposta fornece autenticidade de dados, mas também tem um impacto considerável no desempenho.

Palavras-chave: CEP, Safety-critical, sensor networks

Table of Contents

List of Figures

List of Tables

Acronyms and Definitions

AES - *Advanced Encryption Standard* – encryption method developed by U.S. government. Uses symmetric key algorithm to encrypt transferred data and decrypt on the destination.

CEP – *Complex Event Processing* – standard executable language for specifying actions within business processes with web services.

CoAP - *Constrained Application Protocol* – web application protocol used for communication between constrained devices, network components with low processing and power capabilities, that are part of the same system.

CRM – *Customer Relationship Management* – enterprise tools and processes focused on the client requirements, interaction and overall interaction between the corporate and client.

DTLS - *Datagram Transport Layer Security* – protocol used to provide communication security in UDP protocol communication.

HDFS - *Hadoop Distributed File System* – file system used in complex and distributed applications, providing fast and fault tolerance access to the data.

NASA - *National Aeronautics and Space Administration* – United States federal agency focused on technological research and space exploration.

RFID - *Radio-Frequency Identification* – identification and tracking of hardware objects using special tags that contain their id, using radio waves.

SH1 - *Secure Hash Algorithm 1* – cryptographic hash function used to compute message, thus making data secure, before it is transferred between network components.

SSL - *Secure Socket Layer* – standard for secure communication via TCP. The communication data is transferred via secure channel, where all the data is encrypted.

TCP – *Transmission Control Protocol* – standard that defines how to establish and maintain a network communication via which application programs can exchange data.

TLS - *Transport Layer Security* – protocol used for providing data security to communication between two applications.

UDP - *User Datagram Protocol* – communication protocol which send data packages between connected nodes.

WSN – *Wireless Sensor Network* - network consistent of geographically distributed sensor devices, used for data collection and system monitoring, connected through wireless.

6LoWPAN – *IPv6 over Low Power Wireless Personal Area Networks* – network where each node has its own IP address and system components can communicate and establish internet connection.

1 Introduction

In this chapter, we begin by describing the problem and the purpose of the realization of this work and then present our motivation as well as the main contributions of this project.

1.1 Problem statement

Complex Event Processing (CEP) can be defined as a set of tools and techniques for analyzing and controlling event streams consistent of data, gathered from different sources. While data is being collected and before it has been processed, this raw data is useless for management, decision making or performance analytics. CEP engines provide a fast processing and filtering of this data allowing understanding of what is happening within the system, business opportunity and problem identification, and more efficient usage of information for enhancing the operational performance and security (Luckham, 2008). Due to their popularity, CEP engines have been used in different systems, regardless of the enterprise area. Those systems can be considered a data processing solution to wide range of information systems in some of the following areas: business process automation, schedule and control processes, network monitoring and performance prediction, fraud and intrusion detection, risk management, military, power grid monitoring, etc. (Carvalho, 2008). Figure 1 presents the interaction of CEP engine with other system components, directly connected to it. While sensor values are constantly being streamed, CEP engine must process those and detect meaningful data, patterns, relations and provide this information to the end user as charts, data tables or alerts. Simultaneously, analyzed data is persisted so it could be used for historical analysis.

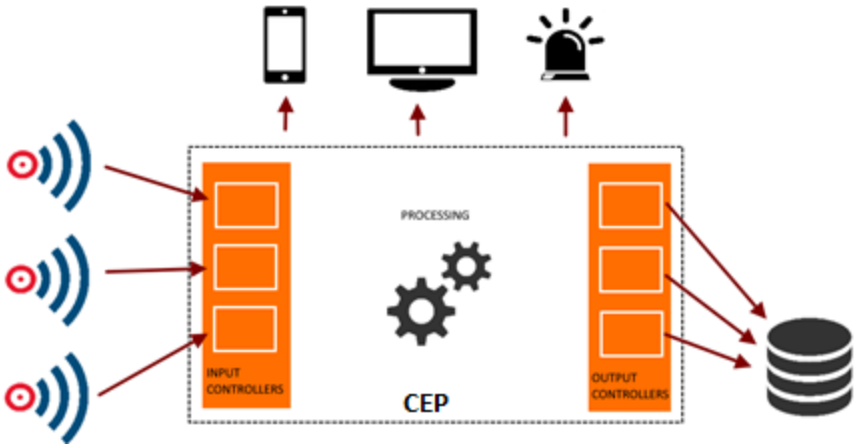


Figure 1 – Architecture example where data is collected from sensors and is received by CEP engine using input controllers. Then, the core of the system processes the received data and sends it to the end user. Parallelly, after being processed, data may be persisted.

Besides filtering data with simple queries, CEP can express rules that cannot be easily defined in other paradigms, such as, if-then paradigm, timestamp validation, timeouts, event patterns or even the absence of events (Adi, 2006). We still should notice that this data processing and filtering is not a totally disruptive technology, for many years' data warehouses have been responsible for analyzing and correlating multiple dimensions of the data. Main difference between those approaches is the processing timing. Since a data warehouse does not process data instantly it may not be considered the best solution for many current applications since there would exist a delay between data reception, persistence and analysis. Instead, data should be processed and events fired in real-time environment (Oberoi, 2007). Besides financial interest from companies, CEP engines are constantly being studied and improved by researchers and developers. This enables constant advance and growth of many available open-source project, developed and tested by the community. Also, major companies, such as, Oracle, IBM, Microsoft, etc., have been showing interest in event processing and provide their own solutions (Oracle, 2017), (Ali et al., 2009). However, most of the available engines are seen just as filters and do not provide some of the quality attributes that are required in most of the systems, starting from relatively simple single-instance system and up to large processing distributed solution. With the increase of the system size and its operational impact, there is a need to provide necessary mechanisms to guarantee its operability. There are many different quality attributes and, sometimes, it may be necessary to sacrifice some of those to provide others. For example, it may be necessary to slightly reduce performance to guarantee safety, using data encryption or backup management to ensure system's availability. Regardless, any computer system should be able to cover at least some of those quality attributes to be considered reliable. It is important to notice that some of the quality requirements may be considered indispensable by the industry standards and in this case, the final system should provide at least the required mechanisms. Those standards do vary from area to area but there are available basic guidelines, applicable to any system (NPES, 2005).

In this study, we focus on safety-critical systems, responsible for performing high impact operations and, therefore, those systems should be carefully designed since their incorrect operability may result in endangering human lives or property. Those systems should be able to function in different environments while providing fast response, high performance and availability, security, robustness, fault-tolerance and so on.

If we consider the performance, focusing on data processing, we believe that CEP engines would be highly beneficial for the safety-critical systems. Since they would reduce the required processing time as well as eliminate data noise, reducing the amount of data to be processed. This approach would allow faster transformation of data into information. But even while focusing on the performance, other quality attributes may not be neglected in critical systems and should covered either by the engine itself or by the remaining of the system. Our research question is: *Can CEP engines be used in safety-critical systems?* To answer this question, we begin by studying CEP engines and safety-critical systems, concentrating on their operational environments and required quality attributes. After identifying the attributes that some of the engines provide, we focus our attention on two attributes that have been neglected: safety and security. To better understand those attributes, we investigate what security mechanisms can be used and which ones are the most important in any system. Finally, we provide a solution for data authenticity, as mechanism added to a CEP engine. Besides our research question, by the end of our study we expect to also answer the following questions: What are the general and indispensable requirements of a safety-critical systems? What qualities current CEP systems offer? Do CEP engines provide any security mechanisms that are necessary in a safety-critical system? Which mechanism could be added to a CEP engine as a security measure? By answering these questions, we deliver information about security required in critical systems, specified by the industry standards versus security offered by existent CEP engines. In case those engines are not capable of covering the requirements that means that it will be necessary to either integrate other components or the system itself should validate the incoming data and guarantee privacy.

1.2 Methodology

We divided our study in three stages:

- Study of the applicability of CEP engines in critical systems while describing some of the existent solutions as well as the evaluation of their quality attributes. We describe some of the studies and quality attributes that were considered by the authors;
- Identify required quality attributes and consider two of the most neglected: safety and security. For that purpose, we study some of the most popular CEP engines and provide the list of the quality attributes that those engines offer;

- While focusing on one engine, ESPER, we propose a security mechanism for validation of data authenticity. By adding that mechanism to the engine, we ensure that only reliable data is processed.

1.3 Contributions

In practice, the main contributions of this work are:

- Definition of the quality attributes in safety-critical systems. The knowledge of required quality attributes allows not only better understanding of those systems but also provides the operation requirements. Those quality attributes have also been discussed in our published paper, presented in appendix B;
- Evaluation of complex event processing engines considering required quality attributes. The comparison of those quality attributes allows understanding the applicability of complex event processing in safety-critical systems;
- Development of a security mechanism, focused on data authenticity, for validation of data before it is processed. Depending on the architecture some of the existent solutions may not be the most appropriate or even usable.

1.4 Outline

The remainder of this document is organized as follows: section 2 presents the background for our work, describing existent studies of CEP engines and safety-critical systems. Then, section 3 describes the application of CEP engines in safety-critical systems as well as quality attributes of those systems. Section 4 describes the quality attributes that are important in safety-critical systems and evaluates CEP engines in terms of those attributes. Section 5 discusses the security characteristics and requirements in safety-critical systems and complex event processing engines, while section 6 describes the proposed solution that focuses on adding security, data authenticity mechanism to ESPER. Finally, section 7 presents our conclusions and future directions.

2 CEP Engines and Safety Critical Systems

Complex event processing engines have been studied over the past years. Due to their popularity, there are available many studies that evaluate their performance, operational characteristics as well as the quality attributes. Also, has been increasing the number of safety-critical systems and studies that investigate failure detection mechanisms, design specifications, system's quality and operability. Since besides CEP engines we study safety-critical systems, in this chapter we provide an overview of the existent solutions.

2.1 CEP engines

While providing a CEP overview, (Aidi, 2006) describes some of the most common areas where complex event processing systems are applied and environments where these engines can explore their full potential. Some of the examples are: bulk transaction monitoring and control, fraud detection, sales analysis, CRM, message tracking. The author presents some of the key benefits brought by those systems. Similarly, (Oracle, 2012) and (Aidi et al., 2006) emphasize their approach on financial area and propose solutions using CEP. All the authors state that CEP systems have a high potential for processing data when it arrives and can provide data of interest in a satisfactory amount of time. Since the main goal of CEP is to be agile and efficient in processing data so it can be delivered to the end user as soon as possible, (Cockburn, 2016), (Schmidt, Anicic & Stuhmer, 2012) and (Ammon et al., 2009) investigate the efficiency of those systems in alarm triggering and alert management. Were presented the advantages of the CEP systems and described how those can be tuned to provide all the required information. Those systems can be used for creating alarms in dangerous environments as well as noncritical businesses that completely rely on getting information on time. Therefore, (Daum, Götz & Domaschka, 2012) focused their study mainly on BPM - Business Process Management Systems and describe if CEP can satisfy all the requirements of those systems. The authors present other approaches that may be used to gather and filter data such as, message broadcasting, long, event management, etc. As conclusion was stated that CEP could be useful technology in supporting business processes in future when this technology evolves.

Concerned with performance, Oracle released a white paper with evaluation of their CEP engine (Oracle, 2008). In the study was used a single instance setup and evaluated the processing latency while constantly injecting data. Latency represents the time between sending event to the engine and receiving a response, while throughput was tested by loading more and more events into the system to be processed. Both number of connections as well as load where being scaled to provide

better latency understanding. System latency is directly affected by the number of clients that generate and send stream and the system should be able to efficiently handle many connections. The authors concluded that this engine is capable of efficiently dealing with up to 1 million events per second with multiple incoming streams. Also, in (Oracle, 2010) is described the system availability and fault tolerance. Were performed latency tests while discussing system's performance vs availability and concluded that there is a possibility of tradeoff configuration by the user, sacrificing one of those qualities. Data availability is also a requirement of sensor-based critical systems. Since the CEP engine itself can provide stream availability while ensuring that no data is lost or processed more than one time, the need of the rest of the system to handle those tasks is reduced. Also, CEP engines availability is discussed by (Saboor & Rengasamy, 2013). Were described the required CEP design factors, such as, scalability, availability, pattern matching and caching, and provided a set of guidelines for developing a CEP engine. Authors believe that their overview of the methods and mechanisms will help software developers while building their own solution. Wahl & Hollunder (2012) completed a performance evaluation of three CEP engines: Microsoft Stream Insight, Esper and Drools. Besides latency, researchers considered load tests and measured the consumed memory, CPU and latency/throughput. As the conclusion is stated that all those engines have very similar latency while system resources consumptions are different. CPU usage was similar for two of the systems while another one showed much higher processor usage. In contrast, the system that consumes more CPU requires much less RAM than other two engines. Baldoni, Montanari & Rizzuto (2015) proposed a solution that combines Complex Event Processing and Hidden Markov Models (HMM) to analyze system failures and their symptoms, using specifically defined metrics. The abnormal conditions are detected using the defined rules and alert events are fired. The CEP engine provides all the necessary data as well as the performance metrics while HMM are used for system state specification and recognition. The authors state that it is important to be able to detect failures in system components to be able to prevent the entire system to be compromised. Hence, this work proposes a failure prediction architecture, focused on the traffic control systems, named CASPER. In the study is stated that CASPER exhibits good accuracy and it can generate predictions with a margin of time that allows recovery actions to mitigate the upcoming failure occurrence. To ensure system's security, the system was developed as black box and as non-intrusive as possible. Each isolated module communicates with other and validate if the other component shows any failures. Exchanged data is validated by the pre-processing module which confirms each package headers. Also, as part of the system, exists failure prediction module, which categorizes the system as safe or unsafe.

Other study used ESPER and other two enterprise CEP engines to evaluate their performance (Mendes, Bizarro & Marques, 2009). The authors executed different types of queries, starting from simple operations, such as, selections and aggregations and up to join operations and multiple queries. It was possible to observe that CEP engines can achieve high throughput while performing basic operations while more complex operations require data to be mapped to memory to provide

higher throughput. In this study, while working with simple operations the bottleneck was not in the CEP system itself, rather the client API, that connected to the engine, was increasing the processing time. Also, it was observed that the window expiration mode (jumping/sliding window) had a significant performance impact. Benchmarking results, which show the latencies of a commercial CEP product, were presented in another study. The CEP engine was the WebLogic Event Server, an application server designed specifically for event processing applications that require high throughput and reduced latency, while handling large volumes of events (Alves & Rorke, 2008). Finally, performance tests, using Esper and StreamCruncher, were executed by other authors, who divided their evaluation in two parts: latency and throughput. It was concluded that both engines have their advantages and performance flaws but Esper is the most mature engine (Dekkers, 2007).

After considering described studies it is possible to understand that there are numerous CEP applications in different areas. While some of the authors describe available systems, others focus on performance evaluations and state that some of the engines may be more appropriate to use, depending on overall system purpose but, overall, complex event processing systems provide high operational throughput.

2.2 Safety-critical systems

While working with safety-critical systems, a study discussed that besides development design, documentation, testing and review that are necessary phases during the development of a critical system. System design must be carefully developed to guarantee that after the development the system will have the expected operability and that all of its components are correctly integrated. The authors state that testing may be challenging since to have a good average result it is always necessary to perform a lot of repeated executions. Thus, system trustworthiness can also be assured using rigorous mathematical techniques in the review process (Parnas, Schouwen & Kwan, 1990), (Collins, 2013). Similarly, Aftab & Nadeem (2013) address some techniques that should be used while developing a safety critical system. In the authors' paper are described formal and informal analysis techniques, such as, Fault Hazard Assessment (FHA), Fault Tree Analysis (FTA), Failure Mode Effect Analysis (FMEA) and Deductive cause-consequence analysis (DCCA). Different techniques may be used to ensure system safety and, although DCCA has high success rate out of all safety analysis techniques, it has a major drawback which is the inability to consider unintended behaviour and, therefore, fails in providing fault tolerance. While considering the safety of safety-critical systems, researchers identified several challenges, such as, encryption and authentication mechanisms, malicious and unsafe commands, lack of knowledge of the system vulnerabilities, etc. To address those challenges are presented security principles that could be integrated into safety-critical systems. Authors state that if systems do not have security mechanisms even small error occurrences may end up compromising the entire system. By using the proposed guidelines,

the system will be capable to recover from errors without having its performance affected. On the other hand, Axelsson, et al. (2016) describe an agile methodology while developing a safety-critical system. Due to the importance of critical systems agile methodology should be adapted to integrate hardware and software components and provide necessary early validation, e.g. by automation, continuous integration, feedback, transparency and flow of the system. As the conclusion of their study, the authors state that agile development methodology cannot guarantee all the requirements and, therefore, other methods should also be adopted. Since safety-critical systems should follow some of the available standards during the entire lifecycle, Youn & Yi (2014) discuss some of the standards that are more applicable to avionic systems. The authors state that in current systems both software and hardware components are equally important and required to process large amounts of data necessary to control the avionic system. Moreover, besides mutual influence between hardware and software, software malfunction may be related to its communication with the hardware. After describing some of the standards that should be applied and similarities between those, is declared that the performed study provides a better understanding not only in terms of software component of the systems but, also, the hardware requirements.

Any critical system, besides operability, requires bigger amount of security than any other system and Oedewald & Gotcheva (2015) state that those systems should be gifted with the ability to anticipate, monitor, respond and learn system's activities. This approach would guarantee expected system resilience. In the study are described some of the challenges of introducing security in a complex dynamic network of subcontractors, involved in the construction of a new nuclear power plant in Finland, Olkiluoto. Was concluded that it may be challenging to introduce safety culture in studied systems since subcontractor companies should prepare the part of the system, that they developed, to operate with other system components. Related to security, Zio (2016) discusses risks and vulnerabilities in critical infrastructures. After describing main concerns and actions, the author concludes that what usually causes system to be open to hazards, failures, malicious attacks and errors is the systems' high degree of inter- and intra-connectedness. Avila & Martínez (2015) describe safety-critical system monitoring and failure detection. Was provided a probabilistic approach for constant online monitoring of the system. Proposed solution may be used in diverse applications, such as, detection of unauthorized access, irregularities in vital signs and other variables in intensive care patients, fraud in financial services, and detection of path deviation in autonomous vehicles. Different fault types, activities, failures description and understanding are provided by Hamill & Goseva-Popstojanova (2014). The authors use NASA study as their base and describe possible system uncertainties, fault types and activities as well as possible relation in those. Was possible to detect fault associations and root causes of some of the failures.

Many studies of the safety-critical systems focus on the system's monitoring and system's failure detection (or even part of the system). Also, many authors discuss system security and importance of prevention while dealing not only with third-parties but, also, inside the system itself.

3 CEP in Safety-critical systems

There is a variety of quality attributes that should be provided by any information system. However, regardless of the system type and size, one of the main attributes is, of course, performance. In case of data processing it may be measured as the amount of data that engine processes per amount of time or by evaluating latency (necessary time to receive data, process stream and send the response). This is also applicable to safety-critical systems that require on-time alert generation. In this chapter, we describe some of the safety-critical systems that incorporate a CEP engine. This part of the study allows us to understand if there are already available safety-critical systems that use complex event processing and what system characteristics are considered.

3.1 System monitoring

CEP systems have been around for quite some time and have been constantly tested, improved and developed (Mendes, Bizarro & Marques, 2009). Due to the wide range of possible application domains, CEP capabilities have been constantly improving, over the past years. Currently they are used in a variety of software systems as a customizable tool for data processing and analysis. These engines can completely adapt to the desired system purpose and are able to process and present most important data, based on previously defined rules. Therefore, traditionally, CEP systems are part of more complex systems, inside which they are used to process incoming information, create alerts based on a collection of rules, and posteriorly provide filtered and summarized data to the final user (Eckert & Bry, 2009). Consequently, those systems are focused on information extraction and processing of the important data in real-time environments. Although our focus is on safety-critical applications, in this section, to correctly assess the state-of-the-art, we will also consider studies reporting on the application of complex event processing in other areas.

The design and the implementation of a lightweight and extensible Complex Event Processing engine, called LiSEP is described by Zappia, Paganelli & Parlanti (2012). Figure 2 shows the processing data flow in LiSEP. During the system design specification, the authors were driven by the principle of minimizing dependency on external software components and, therefore, LiSEP depends solely on the Java Standard Edition libraries, thus minimizing deployment requirements. Moreover, the LiSEP logic is strictly focused on core event processing, consequently resulting in a lightweight and minimal implementation.

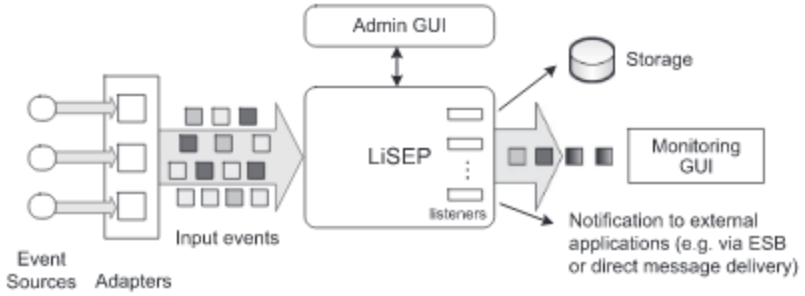


Figure 2 - LiSEP data flow (Zappia, Paganelli & Parlanti, 2012)

The proposed solution is complemented by the specification of the Event Processing Language, based on the SQL syntax. As a proof of the architecture, is proposed a case study on dangerous goods monitoring during maritime transport as a part of Italian Ministry for Economic Development research project, called SITMAR - Integrated system for goods maritime transport in multi-modal scenarios. More data-focused approach is presented by Evchina & Lastra (2016). This work aims aiding the end users of monitoring systems by delivering the selected information to each user, based on their role in the system. The described approach combines Semantic Web technologies and Complex Event Processing (CEP) for configuration purposes and run-time data processing and analyzing. The authors state that the final developed solution should be able to provide ways to deal with multiple devices and multiple users of the system; should be reconfigurable to reflect changes in the environment and/or user information needs; and finally, the device updates should be delivered to users within reasonable amount of time. Considering those requirements, the developed approach provides two major advantages. Firstly, the behavior of the system could be easily changed by only configuring the underlying ontology and, secondly, CEP usage at runtime makes system event-driven and reactive to frequent changes in the environment.

3.2 Critical infrastructures monitoring

Itria, Daidone & Ceccarelli (2014) present an approach for critical situation detection that uses CEP architecture for real time event analysis as well as the event correlation. Event analysis consists of data processing and event correlation corresponds to establishing a relation between input events. Those events are gathered from various sources and are necessary for detecting patterns and situations of interest in the emergency management context. This solution describes the engine, developed in the context of the Secure! Project (Secure, 2016). Figure 3 presents the proposed architecture.

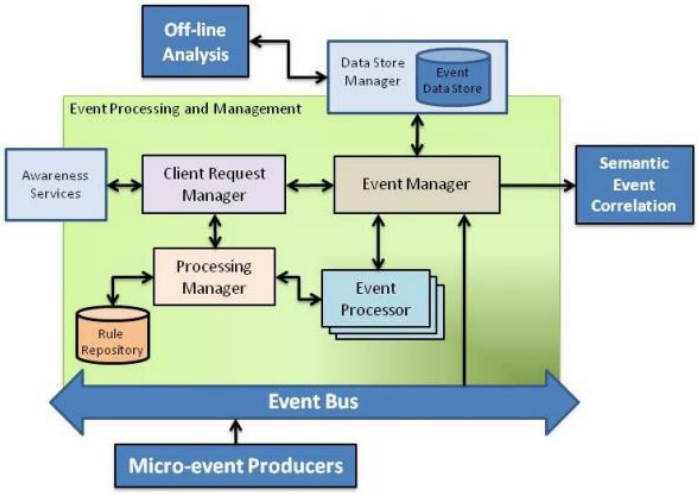


Figure 3 – Proposed architecture. System modules are integrated and are necessary for managing, storing, correlating and aggregating events. Presented module receives micro events and processes those against stored rules. Event processing module is based on ESPER correlation engine (Itria, Daidone & Ceccarelli, 2014).

That solution has two main requirements: the correlation module should be adaptable to the possible changes of the source environment, and it must process the available historical data to evaluate the actual events considering what has happened before. After submitting the system for testing, the authors concluded that their approach can be easily used and maintained. It is also extensible to other scenarios where the application requires nearly real-time correlation, like intrusion detection system (Ficco & Romano, 2011) or monitoring of the critical infrastructures.

3.3 Critical Systems management and predictions

Solution for power grid monitoring, using complex event processing, is presented by Cerullo et al. (2014). The authors claim to be able to provide a detailed treatment of the security issues resulting from the adoption of Wireless Sensor Networks (WSN) and QoS-enabled IP connections. The proposed solution attempts enhancing current information security and event management technology by improving its capability of detecting and mitigating attacks targeting the heterogeneous network infrastructure. Figure 4 presents the grid monitoring infrastructure.

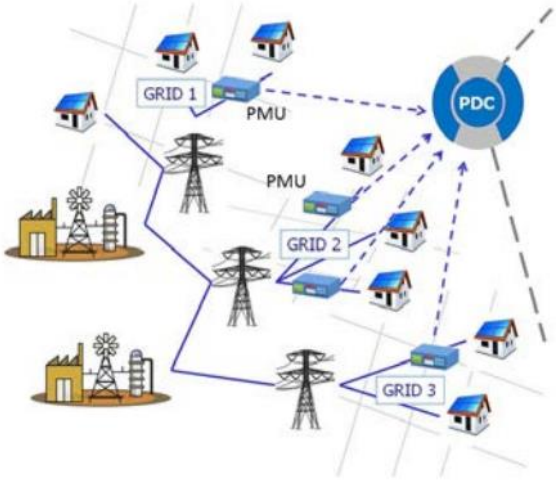


Figure 4 – Integration between system components allows understanding of the entire system at the same time. IF part of the system in compromised the rest of the system is affected and anomaly may be detected. PDC - Phasor Data Concentrator is responsible for collecting and measuring data from PMUs - Phasor Measurement Units that are devices that use signals to measure power quantity (Cerullo et al., 2014).

As an example, in power grid scenario, the attacker may prevent some of the nodes from sending events to the connected collector, thus hiding changes in the power grid conditions. The WSN security probe generates alarms based on the analysis of the network and periodically calculates the package generation rate at every node. The developed engine can correlate those alarms to protect the visualization server. Besides improving system performance, the authors introduced a mechanism that controls the number of exchanged data packages inside the system and attempts identifying malicious nodes. Also, communication monitoring allows detection of DDoS attacks and mitigate traffic attacks. Wang & Kuang (2015) propose a traffic prediction method based on Predictive Complex Event Processing (PreCEP) and Bayesian networks to improve the system's prediction accuracy. The prediction model is trained with historical data and it is used to predict future events based on the recent output of basic CEP engine. Was addressed the prediction problem for moving objects that can be vehicles or even pedestrian and concluded that the performance of the PreCEP still needs to be improved. Currently the parallel method only works when is learning the structure of models in one context and training models for different context.

3.4 Cloud data management

Cloud platform monitoring system, JTangCMS - JTang Cloud Monitoring System, is proposed by Lu et al. (2016). The authors claim that proposed solution can deal with the flexibility, scalability, efficiency and performance challenges of cloud monitoring. Below (see Figure 5) we present the system architecture.

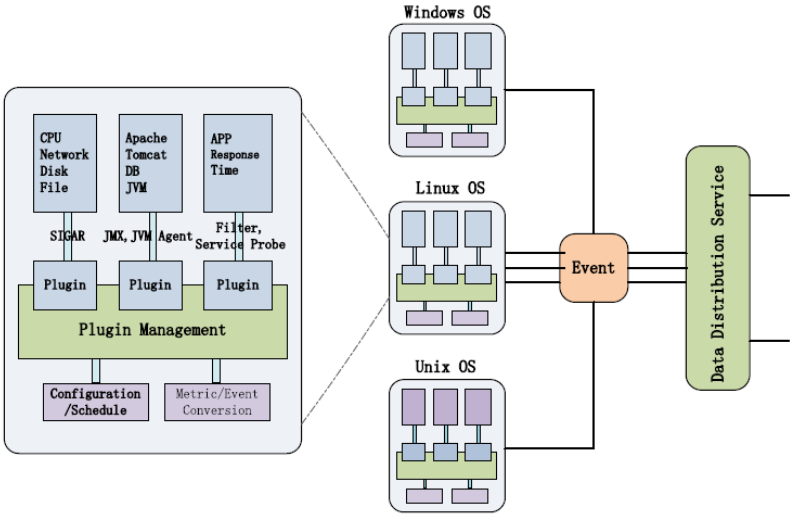


Figure 5 - JTangCMS architecture (Lu et al., 2016). Systems components may be considered configurable plugins which may be added or removed on demand. It is possible to use other available components or develop own modulus, depending on preferences and operational environments.

The system gathers all the data using dissemination framework that allows transferring huge amount of runtime information with high throughput and low latency. For that purpose, is used DDS - Data Distribution Service which partitions the input into smaller parts. After transferring all the data, it must be analyzed and, therefore, was authors developed a support system using CEP, named JTangCMS. The performed evaluations of the implemented algorithm and DDS-based data delivery system state that JTangCMS is an efficient solution and does support intelligent decision making. In Li, Cao & Liu (2013), the authors propose the overall structure and workflow for a CEP-based monitoring system, which can be applied to a private cloud to alert system failures.

3.5 Healthcare

Healthcare is another highly important area that could potentially take advantages of CEP systems. Wang et al. (2010) describe real-time healthcare applications and state that CEP engines can efficiently monitor patient behavior and control medical regulations. Similarly, (Foley & Churcher, 2009) and (Naqishbandi, Sheriff & Qazi, 2015) propose architecture solutions and required characteristics of the systems using complex event processing in healthcare domain. The authors state that CEP engines are highly useful for large and critical data processing and can improve medical systems. Bruns et al. (2014) proposed an ambulance coordination architecture that provides real-time data processing and delivers comprehensive data to the end users. The developed architecture consists of two core components: CEP and FSM- Finite State Machines. Figure 6 presents the ambulance coordination architecture.

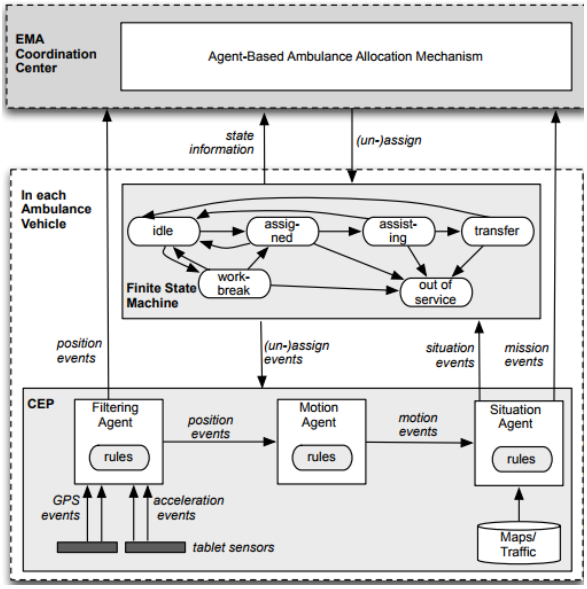


Figure 6 - System architecture. There are two components: Coordination Center and Ambulance module, deployed in each vehicle. Coordination center controls each vehicle state in the fleet. Based on the available information this component should assign vehicles. Vehicle component constantly analyzes and presents current vehicle state (Bruns et al., 2014).

CEP engine focuses on gathering and analysis of the sensor data streams, emitted by the ambulance, to automatically detect changes of the operational states defined in the FSM. The authors defend that efficient and fast patient care requires reliable and up to date information and, therefore, presented an approach that determines the actual state of all ambulances as well as possible relevant mission incidents. Described solution attempts completing research of (Brotcorne, Laporte & Semet, 2003), (Li et al., 2011), (Ciampolini, Mello & Storari, 2004).

3.6 RFID

Zappia et al. (2014) also present patient related data and usage of CEP systems in healthcare area. The goal is to use RFID technology to be able to support clinical management of the patients. The proposed solution is focused mainly on the following scenarios: patient identification and tracking as well as the drug administration. The proposed distributed system is based on event abstraction, event aggregation and event transformation, and uses those to offer a hierarchical and distributed data processing architecture where data is provided by different processing nodes. Overall, the projected solution, based on CEP and RFID technologies, is used to support clinical risk management by means of detecting possibly dangerous patient conditions as well as patient take care. A solution for critical situation detection in elderly daily life is proposed by Xu et al. (2014). The study emphases on the personal assistance as well as possible risks and identification of the required assistance situations. These situations are detected by analyzing the received data from the sensors. This study is focused on ALL – Ambient Assisted Living technologies and uses

sensors to monitor welfare parameters and environmental conditions as described by Wolf, Schmidt & Klein (2009).

Another RFID solution is proposed by Yao, Chu & Li (2011). Is described an RFID framework, using CEP, for managing hospital data, gathered from different sources, and the system focused on detecting patterns and medically significant events. Therefore, the authors created prototypes that attempt showing that CEP has the ability of providing alerts to the healthcare professionals as well as increasing quality of healthcare and patient safety (see Figure 7). One of the main goals consists of identifying the patient and tracking all the necessary procedures since there may be mistakes from part of the hospital staff.

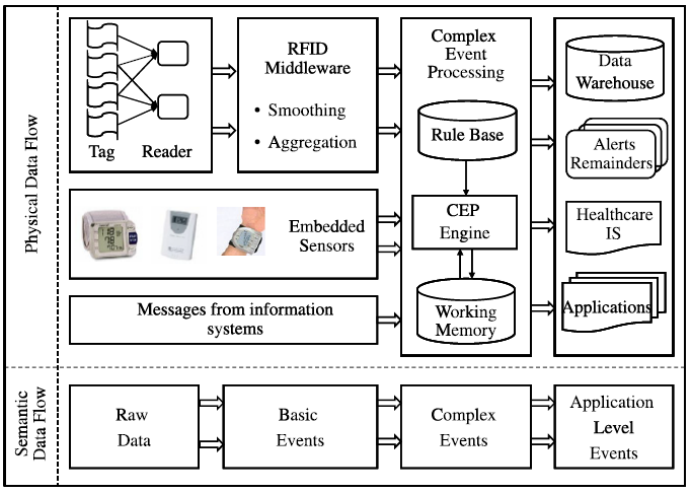


Figure 7 - Designed Framework System data flow may be divided in two parts: Semantic and physical data flows. Collected data is processed and filtered, to remove noisy data, generating basic events. Basic events are then processed again, using aggregation, etc., and create other events. In the physical data flow, data is received from the sensors, is processed and posteriorly is delivered to the end-user (Yao, Chu & Li, 2011).

Also, the system provides all the necessary historical data for emergency handling as well as for health problems, medication identification and response. All those characteristics show that this approach may reduce errors as well as provide faster and more efficient risk management.

3.7 Controlling system events based on rules

CEP scaling solution for processing CPS - Cyber-Physical System data is showed on elevator scenario to describe the challenges of CEP technology in a CPS context (Ollesch, 2016). Figure 8 shows the proposed architecture using Kinect sensor.

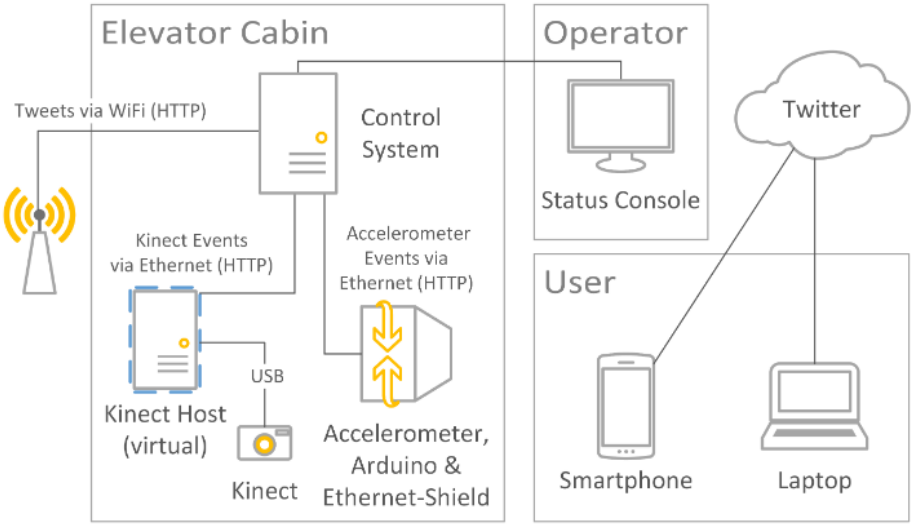


Figure 8 - Architecture of solution in CPS system (Ollesch, 2016).

One of the main challenges of this architecture is the calculation of the floor, since it may be calculated wrongly as well as the sensors themselves are subject to errors and can be disturbed since, for example, passengers could obstruct the Kinect sensor. CEP systems are based on rules and using this example was possible to proof that some parts of the system design should be adjusted. Those adjusts may not be easy to accomplish, depending on the engine that is used. For example, considering ESPER engine (Esper, 2016), its rules are embedded inside the program code and cannot be externalized. Consequently, each change affects the source code and it will require new system distribution. Wang et al. (2014) also study ESPER as an example and define the event process functions, like event attribution extraction and composition determination. The authors investigate the challenges in designing a CEP for Cyber-Physical System, and propose a semantic enhancement CPS event model. Balogh et al. (2016) propose a conceptual architecture for system monitoring that ensures the correct behavior of the system by using a set of different monitoring rules. It was developed a prototype using the VIATRA-CEP Event Processing Language (Viatra, 2016). Posteriorly monitoring rules are distributed over the nodes. The system evaluation was based on development and monitoring of safety-critical embedded systems in the railway domain.

Beer & Heindl (2007) overview the state of art in testing dependable event-based systems and identify the challenges that should be addressed in the future. Were considered two case studies: large-scale project for business-unit transportation systems and a small-scale telecommunications project for Airbus. The study was focused on testing and the authors state that it is an important topic and cost reduction of testing may be achieved by developing new and more efficient techniques of requirements tracing as well as test-case design and validation. Part of the study was on (Luckham, 2002) who proposed CEP usage for managing multiple events and event causality to elicit meaningful events in an event-based information system. This is because CEP can evaluate

incoming data and transform it into out coming events. Lang & Capík, (2014) present a procedure for performing predictive analysis of complex events occurrence in time critical complex event processing systems. Although the authors developed a system in publish-subscribe architecture, there were no considerations of safety nor security. But, it is possible tune the system to perform better or worth according to its data and environment. This system adaptability provides better operational characteristics and makes it more reliable. Similarly, in other study, was stated that using CEP it is possible to identify and apply business intelligence rules over the streams of events and this technology is critical in an environment where time plays an important role, such as, real time decision making (Tendick, Denby & Ju, 2016). In the study were investigated the possibilities and the use of the available methods and techniques for classification and prediction in complex event processing. For this purpose, it was designed and implemented an application CepPredictiveAnalysis, which uses defined methods for predictions and despite certain limitations, the system gives acceptable accuracy in financially-oriented applications.

4 Quality attributes

In the previous sections we described CEP engines and Safety-critical systems as well as different applicational areas of those. Quality attributes transcend system type and purpose and are necessary in any system to ensure its correct operability. There are different quality attributes that may be covered by the system design and implementation and, below, we describe some of those. Also, the software quality attributes determine the quality of the final product and define if the developed system is operating as expected. Different quality attributes may affect some specific layers of the software and it is important to find the most appropriate tradeoff between attributes, depending on the system purpose and operational environment (see Figure 9).

Figure 9 - Common software quality attributes (Elzinga, 2017).

While developing a safety-critical system there is a wide range of standards that must be followed and those standards define some of the system's requirements (Bowen & Stavridou, 2002). Those requirements may also be translated into system quality attributes and define the system characteristics without getting deep inside into its functionalities. Currently software systems present, in most cases, a long list of quality attributes (Laranjeiro, Soydemir & Bernardino, 2015) and, therefore, we decided to consider in this work the most general ones, which are fundamental in safety-critical systems. Also, we should mention that different works and authors consider different number of attributes, some being encapsulated by others. Therefore, we studied quality

attributes described by (Barbacci et al., 1995), (Sommerville, 2004), (Atoum & Bong, 2015). Hence, the considered quality attributes are:

- *Dependability*: can be directly translated into trustworthiness of the developed system and represents the confidence in the correct functioning of its operations. One of the dimensions of dependability is the fault tolerance that states that failure of a part of the system cannot compromise the whole system. Also, should be considered the repair capability of the designed system, which describes if it can recover without any intervention, as expected.
- *Security*: ensures that extracted, stored and processed by the system data is not easily intersected or corrupted. There should not be any intentional disruption by the third parties. Authentication mechanism is also a requisite since it allows user identification and extends up to the possibility of the system to recognize the configured devices and treat them as trusted. The collected data will be recognized as viable. System vulnerability should be reduced as much as possible to prevent possible harm by the attackers and important data exposure.
- *Reliability*: describes the probability of the system of performing designed operations in expected time. This attribute is tightened with the availability of the system that states that the system should have the ability to work with limited amount of data when it cannot be collected. There should be available backup data collectors that would gather at least part of the necessary data. Therefore, reliability considers not only software but hardware and firmware modules that are required for correct system functioning.
- *Safety*: this is another quality attribute highly related to the reliability. Thus, safety reflects the system's ability to operate, normally or abnormally, without danger of causing human injury or death and without damage to the system's environment. Some of related safety terms are: hazard, damage and risk. All of those are considered by the standards and usually are described as a prevention list of measures instead of qualities.

We believe that, among others, those are some of the most important requirements of critical software since those ensure the correct system operability and robustness. Also, chosen attributes enclose other vital software system characteristics, for example, performance, resilience, availability, etc.

There are different systems and it is important to understand what quality attributes authors often consider and which ones are mostly not covered. In the previous section we have described the safety-critical systems that are presented below. The summary of quality attributes analysis is presented in Table 1. It is important to notice that some of the solutions may have partially addressed some of the quality attributes, but we just consider overall approach, the authors focus and goals in the designed architecture.

Table 1 - Quality attributes

	Dependability	*Reliability*	*Security*	*Safety*
(Baldoni, Montanari & Rizzuto, 2015)	X		X	
(Zappia, Paganelli & Parlanti, 2012)	X			
(Evchina & Lastra, 2016)	X	X		
(Itria, Daidone & Ceccarelli, 2014)	X			
(Cerullo et al., 2014)	X		X	
(Wang & Kuang, 2015)		X		
(Lu, et al., 2016)	X	X		
(Li, Cao & Liu, 2013)	X	X		
(Bruns et al., 2014)		X		
(Zappia et al., 2014)	X			
(Xu et al., 2014)	X			
(Yao, Chu & Li, 2011)		X		
(Ollesch, 2016)	X			
(Wang et al., 2014)	X			
(Balogh et al., 2016)	X			
(Beer & Heindl, 2007)				X
(Lang & Capík, 2014)		X		

Dependability and reliability are quality attributes that are studied and considered in most of the existent solutions. After analyzing the results, we concluded that most of the systems are focused on their performance and operability and do not consider safety neither security. Those quality attributes are often treated as less important than assuring system's proper functioning or performance. Those are more functional characteristics and any designed and developed system is expected to work as intended. Some of the authors consider how their system is affected by security attributes and if their data is secure and only one study considers risk, hazardous situations management and system's awareness of those. It is important to notice that we performed our analysis of quality attributes based on authors' description and system specifications. Though, some of the described works may provide parts of considered by us as quality attributes, but those were not mentioned. Also, although we just have one level analysis, as previously stated, we investigated authors' focus and what their work was trying to achieve.

5 Security in CEP engines

We began the second part of our study by investigating some of the standards applied to safety-critical systems and, also, standards essential for basic data transferring and management. Since there are many available standards, our aim is to identify standards that apply to critical systems, such as powerplant management, and provide not only basic guidelines but specific mechanisms that should take part of any critical system. Therefore, we choose the following standards: NIST SP 800-53, SOC2 and IEEE 603-2009.

- NIST SP 800-53 standard is a standard for Security and Privacy Controls for Federal Information Systems and Organizations and it is based on other two standards: ISO/IEC 15408 and ISO/IEC 27001 (NIST, 2013). It is one of the obligatory standards for industrial systems as well as gasoline pipelines, water storage dams and other national security systems (NIST, 2010), (Stouffer & Katzke, 2008).
- IEEE 603-2009 is a specific standard for Safety Systems for Nuclear Power Generating Stations. It provides some of the operational criteria and minimum functional design principles for the power, instrumentation, and control portions of nuclear power generating station safety systems are established (IEEE,2009).
- SOC2 focuses on a business's non-financial reporting controls as they relate to security, availability, processing integrity, confidentiality, and privacy of a system. It is mostly used for cloud services and datacenters as a standard for control and assurance of the confidentiality and privacy of information that may be stored or processed by datacenters (Microsoft, 2016).

Our evaluation parameters are represented by the necessary mechanisms which are described by the standards. Not only at least some of those mechanisms should be present in a safety-critical system but the CEP engines should provide at least basic security. Standards, described above, allowed us to identify specific security mechanisms that are necessary in any safety-critical system. Moreover, we believe that at least some of those mechanisms should be integrated in CEP engines. Since there are many available CEP systems, we chose some of the most popular, more referenced, that are large projects and provide enough official documentation. The considered CEP engines are, as follows:

- Apache Flink;
- ESPER;
- Drools;
- Opacle CEP;

- Sybase Aleri.

All those engines are available in both standard and enterprise versions. There is one specific characteristic that distinguish CEP engines, their operational language. There are two language specification possibilities: *stream-oriented* (transforming) languages and *rule-oriented* (detecting) languages (Vincent, 2016), (Won, 2016). The stream-oriented languages provide operations for processing the input streams such as filtering, joining or aggregating to obtain some other output streams. The rule-oriented languages use rules for processing streams, clearly separating the triggering conditions and the actions to be taken when the conditions are met. The main component of the stream-oriented systems is the sliding window (Figure 10). There are three types of sliding windows: a) time based windows that captures the last segment of an ordered stream for a given time interval. For example, showing the relation between current data record and records processed specified number of seconds ago; b) tuple based windows, which capture the last number (parameter) of elements of a stream; c) partitioned windows, which operate similarly to a tuple based window, with an additional constraint over the data attributes.

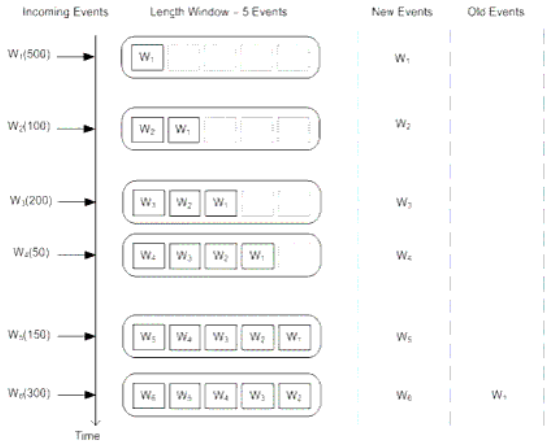


Figure 10 – Example of window that considers and analyzes 5 events which are considered the new events and other are old events (Oracle, 2017).

The received stream is partitioned into sub streams and only specific number of the records is processed per stream and their processing results are combined into the final output. Rule-oriented languages use Event-Condition-Action (ECA) rules as a formalism for defining the actions that should be executed when specific conditions are met (Flink, 2017). That means that a specific action is performed if the condition is satisfied. Therefore, a rule is evaluated only when triggered by a specific event, which can be a primitive event (database operations, temporal events or external notifications) or a combination of primitive events using logic operators (Moraru, 2012).

Since all the considered systems are large projects, there is substantial amount of available information that is constantly being updated. Therefore, table data is based on the white papers, documentation provided by the companies, email lists and forums. As stated above, were chosen

the security mechanisms that we considered the most important (basic), required in any safety-critical system and were specified in standards:

- Access control – defines different levels of system access and operations usage;
- Backups – data persistence;
- Encryption – may be encryption of the data itself, parts of the data or even communications;
- Fail-safe procedures: represent the system recovery capabilities and the failure management;
- Reporting information security events – security monitoring and reports;
- User identification and authentication – user management and authentication for system access;
- Wireless communication policy – methods that consider communication outside of the system.

Below we present the results of our study and describe what system's characteristics are not available and which ones are present and how those operate. Table 2 presents the summary of security mechanisms, described in safety-critical standards, and their availability in the studied CEP engines (presented in alphabetic order and nothing else).

Table 2 - CEP engines security attributes.

	Apache Flink	***Drools***	***Esper***	***Oracle CEP***	***Sybase Aleri***
Access control	NO	YES	NO	YES	YES
Backups	YES[5]	NO	NO	YES[4]	YES
Encryption	NO[1]	NO	NO	YES[1]	YES
Fail-safe procedures	YES	NO[3]	NO	YES	YES
Reporting information security events	NO	NO	NO	YES	NO
System status identification	YES[2]	NO	NO	YES[2]	YES

[1] Encrypted connections and data streams between nodes will be available soon (Flink, 2017). We consider that communication requirement will also be covered by those mechanisms.
[2] Requires usage of web client to monitor the system.
[3] There is request for the implementation of a possibility to persist the state of the used by the system memory after each state change and related to that the possibility to system recovery and posterior state restore in case of a crash (Jboss, 2017).
[4] Uses specific persistent event store.
[5] Apache Kaffka is a messaging system that could be used with Flink and persists incoming data in logs. Output data, after being processed by Flink, can be stored in any database (ex. Cassandra, MongoDB, HBase, MySQL, etc.) (Ewen, 2016).

User identification and authentication	YES	YES	YES	YES	YES
Wireless communication policy	NO[1]	NO	YES	YES	YES

Evaluated CEP engines are presented in the table by the alphabetic order, and below we describe our conclusions and study, starting with Apache Flink.

5.1 Apache Flink

Apache Flink is the engine that provides standard authentication mechanisms but there is no possibility for detailed data management access. Ideally, a Flink application may use a variety of different connectors (Kafka, HDFS, Cassandra, Flume, Kinesis etc.) by means of arbitrary authentication methods (Kerberos, SSL/TLS, username/password, etc.). Those connectors are responsible for managing the security requirements (identification and authentication). While satisfying the security requirements for all the connectors, Flink provides first-class support for Kerberos authentication only. Also, for example, it is possible to enable Hadoop security without providing security authentication for the ZooKeeper, or vice versa. Thus, each component (connection) may use separated authentication (Flink, 2017). While working in a cluster, it is necessary to setup public key authentication on the master node as the user who will later execute all the Flink operations. User credentials that are used by the master node should be also existent on each other node, for master to be able to manage available worker nodes (Flink, 2017). Encryption feature will be soon developed and will be available in the future versions. Apache Flink offers reliable execution with strict exactly-once-processing consistency guarantees and deals with failures via checkpointing and partial re-execution. The checkpointing mechanism constantly (at regular intervals) creates consistent snapshots of the state of operators, including the current position of the input streams and if it is necessary (in case of system failure) can replay parts of the job or the entire job. Job execution is managed by the JobManager and it is responsible for coordinating the distributed execution of the dataflow. It tracks the state and progress of each operator and stream, schedules new operators, and coordinates checkpoints for recovery. In a high-availability setup, the JobManager persists a minimal set of metadata at each checkpoint to a fault-tolerant storage and it is possible reconstruct the checkpoint and recover the dataflow execution (Carbone et al., 2015), (Carbone et al., 2015). However, there are no event mechanisms or logs regarding system security status. We can say that Flink itself does not necessary considers security, it is designed to always take part of a system where other components would be responsible for data protection. Therefore, excluding cluster management and component/node connection using user identification.

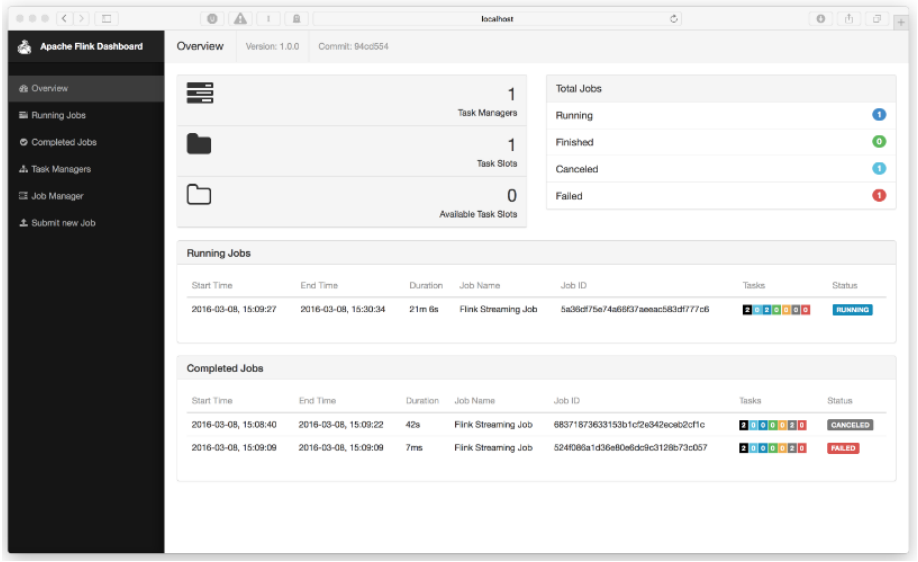


Figure 11 - Apache Flinx web console dashboard. Provides overview of the running and last completed jobs (Flink, 2017)

For overall system status overview, may be used web console (Flink, 2017). This user interface provides summary of the executed jobs and processed results.

5.2 Drools

Similarly, Drools is a CEP engine developed as a module responsible for adding event processing capabilities into the platform (JBoss, 2017). This engine provides both user authentication as well as access control. Under the directories of the system it is possible to find the login-config.xml file which contains all the necessary system configurations as well as the user-related information (users, passwords and roles). This file should only be maintainable for a fixed and small number of users (Mollenkopf & Tirelli, 2009), (JBoss, 2017). To manage considerable quantity of users and their roles more efficiently Drools may be integrated with Guvnor (Drools, 2017). This application allows editing permissions for different stored rules (JBoss, 2017). This per-rule permissions allow restricting users that may manage specific system resources and rules, per user's role. Roles represent diverse user groups, providing different permissions as well (Bhochhibhoya, 2015). Drools, as most of the CEP engines, does not provide own data persistence. Since this engine is written in Java, may be used Java Persistence API (JPA) as described by (Žáková, 2013). Since this engine does not physically store any memory-mapped data, there is no data encryption or backups. Fail safe procedures are pendent for implementation and will be available in the future versions. System status identification and security monitoring are also not considered by Drools. There are no reports or logs performed regarding system status. Figure 12 presents the Drools Workbench UI, used to manage user's projects and receive notifications about processing errors.

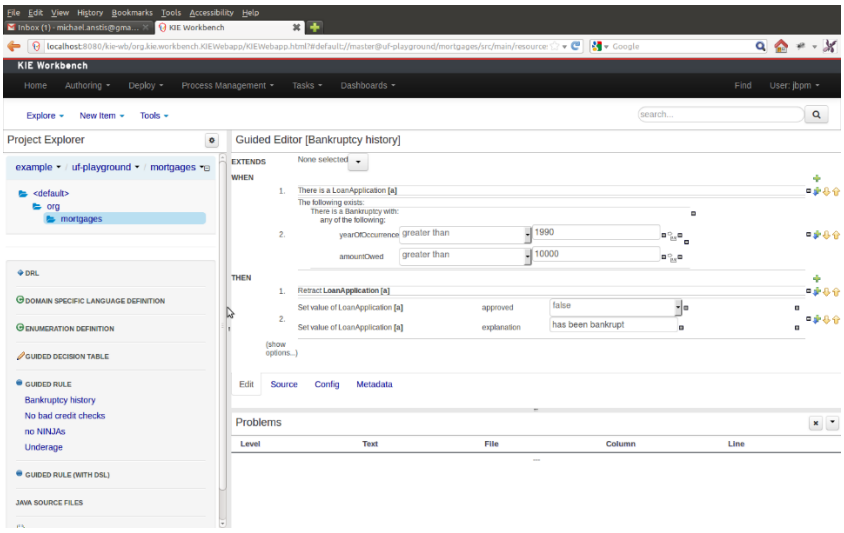


Figure 12 - Drools Web UI. Presents the project rules and events that are triggered when meaningful events are detected. Also in the bottom section, there is error and problems report, generated while running a specific project (Jboss, 2017).

5.3 ESPER

Esper is probably one of the most known and used CEP engines. However, it is important to notice that, as many others, the developers of this system focus mainly on its performance and throughput. Out of all considered security mechanisms, ESPER still provides user authentication as stated in (EsperTech, 2017). Similarly, to Drools, user authentication data is written in xml file and it describes the authentication used for the connections. Since this authentication is used for low level management there is no user access control. Since ESPER does not perform any I/O operations, no data persistence or failure recovery are available (EsperTech, 2017). Although ESPER itself is simply processing system it can provide horizontal scalability, load distribution as well as replication in enterprise version, while built on Apache Kafka and Apache Zookeeper. The fault management and recovery are achieved by combining ESPER with EsperHA module (EsperTech, 2017). There are no security metrics or system constant monitoring. More complex processing rules must be written in EPL – Event Processing language which is very like SQL - Structured Query Language, used by relational databases for data manipulation. However, differently from databases, ESPER does not stores any data. Instead, the user may define a set of rules thought which the received data is run. This means that when a specific patter is detected the event is triggered and some action is performed.

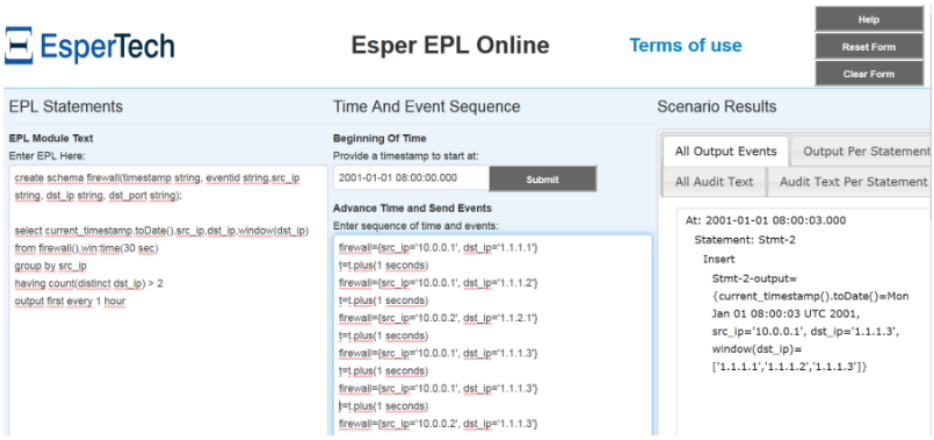


Figure 13 - Esper web GUI. On the right side are defined EPL statements, one with data schema and other one with desired rule. In the center are presented events and on the right side are shown the results (Klendar, 2017).

Since this system has standard and enterprise versions, while working with enterprise version there is available Rich Multi-Window GUI, presented on Figure 13. It may be used as EPL editor, debugger and presents the detailed memory use and metrics reporting for existent rules, job monitoring and historical job data (with tables, charts and gauges) (EsperTech, 2017).

5.4 Oracle CEP

Oracle CEP also provides basic security and authentication. Although the authentication is per user (username + password), there are standard user roles/groups that are available: Operator, Monitor, ApplicationAdmin, Deployer, BusinessUser and Admin. This engine supports various security providers for authentication, authorization, role and credential mapping. Oracle CEP is configured to use the file-based providers for both authentication and authorization but there is always a possibility to configure the system to use an LDAP or DBMS as authentication info provider (Oracle, 2017). For data persistence, the engine offers persistence of the output events when, for example, those are written in the database. By default, Oracle CEP stores recorded events in a database that may be posteriorly queried by the application (by specifying the connection and credentials). Default database used by this engine is single Berkeley DB instance, bundled with the Oracle CEP server. However, there is always the option to change storage system. For example, it can be a Relational Database Management System such as Oracle Database or Derby (Oracle, 2017). As we have described above, in the future versions Oracle CEP will support encrypted connections and data streams between cluster nodes (Flink, 2017). Fault tolerance is achieved by high availability of the system in active-active architecture (Prassinos, 2009). The active-active approach requires for a cluster to have at least two nodes that are up at all time and, therefore, this approach can shorten the failover time. Oracle CEP will choose one server in the cluster to be the primary node and the remaining nodes are considered secondary. High availability allows users to make the most appropriate performance vs. quality-of-service tradeoff for their environment,

including a precise recovery option that guarantees that no events are lost (Oracle, 2010). System monitoring and management is achieved using web console, Oracle CEP Visualizer (Sybase, 2010).

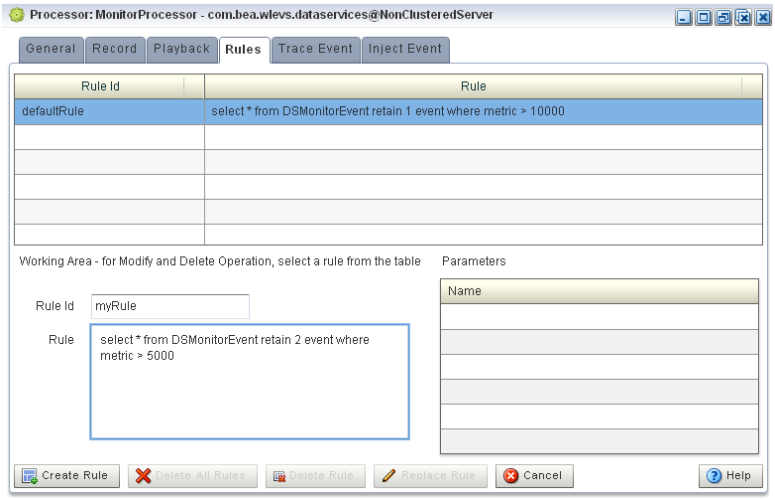


Figure 14 - Oracle CEP rules tab. Allows adding system rules that will be used while analyzing incoming data. This statement defines that each time window will retain two events with metric value higher than 5000 (Oracle, 2017).

This web user interface allows not only manage running jobs and rules, but also perform any administrator tasks, such as, view the structure of a system domain, manage security, configure server instances, etc. Besides overall engine management it is possible to create processing rules, using rules tab, as presented on Figure 14.

5.5 Sybase Aleri

Starting with user management (access controls and authentication), Aleri can use the Pluggable Authentication Module (PAM) package found in Linux and Solaris. In PAM authentication, client applications should connect to the core engine either using command interface, gateway interface or SQL query interface and must supply a user name and password before they can execute any commands. In this case, it is the responsibility of the system administrator to specify the type of the authentication that will be performed by the users (Sybase, 2010). It is important to notice that it is possible to restrict user access even if the authentication is successful. This type of operation complements basic access control and the streaming processor server may be configured to deny the authenticated user the ability to query certain streams through the SQL interface or subscribe to streams through the subscription interface, or to stop the server (Sybase, 2010). Data backup is an integral part of a data management and protection strategy. It is recommended to backup Sybase Aleri streamed data regularly. Should be backed up two types of data: XML files that define the data streams and how they interact (Data Models) as well as generated operational log. If both of those are backed up, it is possible to restore the system after any failure. In terms of backup

creation, there are two possibilities: “off-line” backup (back up the data model and log files while the streaming platform is down) or in an “on-line” backup mode (the data models and log stores are generated while the platform is running). However, it is important to notice that although in “on-line” mode the system will be running, its current operation will be suspended while the backup files are created (Sybase, 2010). Besides backups and restores, system failure management is performed using system availability. While working in a cluster, if the primary server fails, the high availability configuration detects the failure and promotes the secondary server to primary status with minimal interruption to client requests. Also, since this system is designed to be operating in a cluster mode, one of the possible identifications of system status is weather the specific node is down or not. This can be easily achieved by the status command.

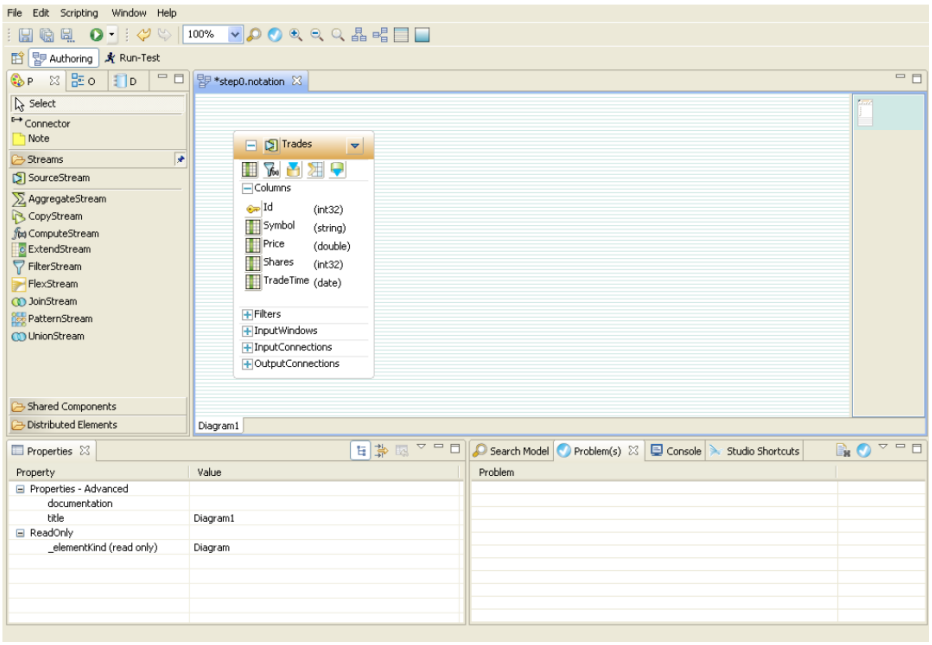


Figure 15 - Aleri Studio UI with source stream, responsible for receiving data which posteriorly is sent to other stream components (filter/aggregate/join/... streams). It is possible to identify data fields that are considered inside the stream: Id, Symbol, Price, Shared and Trade Time. (Sybase, 2010).

Overall system monitoring may be performed either using commands or by accessing Aleri Studio Monitor (Gadgil, 2012). Finally, Aleri provides encryption, using AES, which protects the data sent over a network (transferred between clients and the streaming platform), using secure sockets – SSL (Sybase, 2010).

In summary, we state that although all the studied CEP engines provide basic security mechanism (authentication) there are many other mechanisms that are not considered. Out of all the engines ESPER is the least prepared system to provide data security while Oracle CEP offers many approaches to secure not only the data but the system itself. That means that if ESPER is integrated in a critical system it will be necessary for the other system components to provide security and failover mechanisms.

6 An approach for handling security inside CEP engines

When safety-critical systems have geographically distributed components that communicate using WSN (Wireless Sensor Network), one of the most important requirements is data authenticity. It can be achieved by using secure communication with transferred data encryption, such as SSL. But, while working with sensors with limited processing capabilities, this is not the most appropriate and lightweight. Moreover, we believe that the CEP engine itself should be able of evaluating the streamed data before processing it. This approach would remove additional mechanisms, outside of CEP, and reduce the data management process. Therefore, in this section, we propose own security solution. It consists of adding a security mechanism that ensures data authenticity and integrity, to one of the most popular CEP engines, ESPER.

6.1 Security in WSN

There are different system architectures and it is important to understand their security requirements. For example, it is necessary to provide secure data transferring in a system where all its' components communicate using Internet while in private network, this type of security and authenticity validation may not be required. But since some of the safety-critical systems process incoming data from the sensors, while focusing on security, it is important to investigate some of the available solutions. While dealing with distributed systems, working outside of the internal network, there are different threats, starting from communication channel and up to the data source identification. One of the most common approaches is the use of SSL. Currently most of the systems communicate using SSL/TLS secure connection, that uses certificates to provide encrypted channel between one point and other after initial handshake. This approach in widely used in TCP network communication and has proven secure enough. But, while working with sensors with low processing and memory capabilities, this protocol may not be used because the sensor does not have enough memory to be able to map the certificate to perform the handshake and authenticate itself (IBM, 2012). An overview of sensor network security mechanisms is provided by Granjal, Monteiro & Silva (2015). The authors provide a survey about mechanisms that could be used to guarantee security in internet-integrated low-power networks. Were described different protocols and communication technologies (6LoWPAN, DTLS, CoAP and RPL), suited to specific application levels, and presented some of the security challenges, requirements and available mechanisms. Another security survey was performed by Islam, Shen & Wang (2012).

Were defined some of the security requirements, such as, confidentiality, integrity, authenticity, availability, as well as different attacks that can happen in WSN. After describing some of the industry standards, the authors state that although those provide basic guidelines and some security solutions, the in-depth specification is done by the developer.

Applied towards critical systems, Zhang & Zhang (2016) deliver security analysis of general Cyber-Physical Systems, security issues in smart grid and power systems (using SCADA systems), secure algorithms, secure state estimation and control. The authors state that attacks in CPSs can take various forms, such as inadvertent infiltration through infected devices and network-based intrusion by exploiting poorly-configured firewalls. If the attacker can access system network it is possible to spread malware, compromise the communication equipment and inject false information in measured or computed data. The resulting consequences can be severe, e.g., wide area power blackout in a national power grid or crash and failure of safety-critical infrastructure (nuclear power plants and military facilities). Jung et al. (2016) focused their study on approach presented by Chang et al. (2015), concerning security challenges while dealing with user biometrical information. Some of the described vulnerabilities are:

- Off-Line Password Guessing Attack (attempts to input a password until the correct password is discovered because many users tend to employ simple, brief passwords for the sake of convenience);
- User Impersonation Attack (if an attacker obtains a password, the attacker can pretend to be a legal user);
- Session Key Compromise (if attacker knows shared ids and passwords, can establish trustworthy connection);
- Scalability Problem (the use of the verifier table is inefficient in terms of the computation time since the changed values at each phase need to be updated in the verifier table);
- Absence of a Session Key Verification Process (there necessary additional procedures that would verify authentication key and its distribution).

After proposing their own, enhanced scheme, the authors conclude that their solution has enough efficiency and security to be used in WSN.

In the communication environment, authentication and key distribution among trusted nodes are some of the most important phases. Chen & Chao (2011) provide a survey on key-distribution techniques as well as metrics that are traditionally used to analyze the proposed solutions. Some of the common metrics are: node resilience – confidentiality of the data exchanged between nodes; resource efficiency – resource consumption; connectivity – while using shared key, the probability of nodes determining their shared key; adaptability – adaptability of distribution scheme to different environments, ignoring some of the assumptions. Authors overview some of the key

distribution techniques. Node authentication scheme is proposed by Fouchal et al. (2016). The proposed solution is backed by TPM - Trusted Platform Module which is a specialized chip on a device that stores RSA encryption keys specific to the host system for hardware authentication. During the initialization step each node is configured with its own public and private keys. When the node sends a message, it should send its signature and only after the server receives the message and it is validated it is considered authentic. The authors evaluated power consumption generated by the approach and concluded that energy cost may be considered low. Other solutions of key distributions are provided by (Javanbakht et al., 2014) and (Gandino, Ferrero & Rebaudengo, 2017). The study describes own methodology of key deployment and communication between nodes and clusters. Both solutions can be tuned to provide better system resilience and neither of them overpowers the systems and increases its operational requirements. Finally, applied to a critical system, Li et al. (2015) review protocol designed for the healthcare area, proposed by He et al. (2013). The studied solution is designed to be composed of medical sensors (such as ECG electrodes, blood pressure and temperature sensors) with limited capacity in storage, computing and bandwidth. The authors describe some of the available authentication methodologies and posteriorly focus on the solution proposed by He et al., (2013) and provide their enhancement to solve authentication and session agreement key phase. Also, the authors claim solving possible DoS problem by adding to the available mechanisms new authentication protocol for healthcare application, using Wireless Multimedia Sensor Networks - WMSNs with user anonymity. One of the requirements in a distributed system is the secure communication between its components. While considering smaller, private systems, security may not be the main requirement since all the components can communicate using private network which is expected to be secure enough. However, while working with more distributed, geographically, components that communicate using Internet, this is a different case. As our case study, we choose sensor networks and security systems that use sensors for collecting the required data. Before being able to provide a secure solution, it is important to understand not only the challenges of a WSN but also the different types of the sensors that exist. Those types may influence the security requirements and mechanisms' limitations. (Malan, Welsh & Smith, 2008).

It is important to consider that there are architectures working with sensors that may not perform operationally and energy expensive operations. One study is focused on MICA2, processing module used in WSNs (Crossbow Technology, 2004). Although the authors successfully proposed public key infrastructure that could be handled by this sensor, as conclusion, was stated that the proposed solution could be improved to reduce the memory consumption. Wander, et al. (2005) also present other low power controllers, such as, Texas Instruments MSP430 (Bierl, 2000) and Atmel ATmega128L (Atmel, 2011). Was evaluated the usage of different authentication and encryption mechanisms in those systems and concluded that some of the hashing algorithms, specifically ECC, can be used in low power components. While data encryption requires more

computational power, hashing functions can offer security with small key sizes. The data exchange itself is performed over protocol based on SSL but which was simplified to reduce the costs.

As case study, we choose sensor networks and security systems that use sensors for collecting the required data. Before being able to create a solution that could ensure data security (authenticity and integrity), it is important to understand not only the challenges of a WSN but also the different types of the sensors that exist. That would allow us to understand how hardware affects the security mechanisms since by using low power components it is possible to reduce the overall system cost. For example, MSP-EXP430G2 controller can be bought for around 10 dollars and ESP8266, designed by Espressif Systems, for around 6 dollars. More powerful controllers provide more processing capabilities but their cost is higher. For example, RaspberryPi costs around 30 dollars while Intel Galileo board costs around 80 dollars. It is important to mention that Arduino components can be acquired separately and it is possible to design own board starting from scratch but that would require additional time and production costs (effort to combine different components and guarantee that the final product is operating as expected).

6.2 Proposed security mechanism

Currently most of the systems communicate using SSL/TLS secure connection, which uses certificates to provide encrypted channel between one point and other after initial handshake. This approach in widely used in TCP network communication and has proven secure enough. But, while working with sensors with low processing and memory capabilities, this protocol may not be used because the sensor does not have enough memory to be able to map the certificate to perform the handshake and authenticate itself. Therefore, it is necessary other approach, with lower hardware requirements. One popular method is random key predistribution [52, 53]. As we described in previous sections, this method distributes random public keys or cluster keys between nodes that would allow them to communicate with each other. Keys are generated randomly and are distributed pair wise between nodes. Server stores the matrix which is used for generation of the keys as well as for the identification of nodes that share the same key. This approach also requires periodically key renewal that would ensure system's security even if an attacker can get one of the existent keys. In this case, server should occasionally distribute the keys and there is always, no matter how small, the possibility of this information being caught. One of the possibilities to eliminate this part of communication is the usage of pre-distributed configuration. This means that all the sensors are manually configured, when system is deployed, with public and private keys and the server can identify each one of them. One of the main flaws of this approach is that the system is update or the keys renewal would require some trusted third-party entity to perform the deploy and manually configure all the system components. When the system is highly geographically distributed, this becomes even more of an exhausting task but, on the other hand, no secret data is transferred at any point.

6.2.1 Methodology

As we have stated above, our goal was to add a security mechanism to ESPER that would ensure data authenticity. Since ESPER is available in java and .Net programming languages, we chose .Net source code. The basic idea of ESPER is to “subscribe” to the specific data conditions/events that appear in the collected streams and it is the part of the engine where we added data validation. After receiving the data, the engine must be capable to determine if this data is trustworthy and if it is, it will be processed, otherwise it is discarded and logged. As mechanisms we used AES encryption, with 128 bits per key, using symmetric key and SH1 hashing. In Figure 16 we present the scheme of implemented approach.

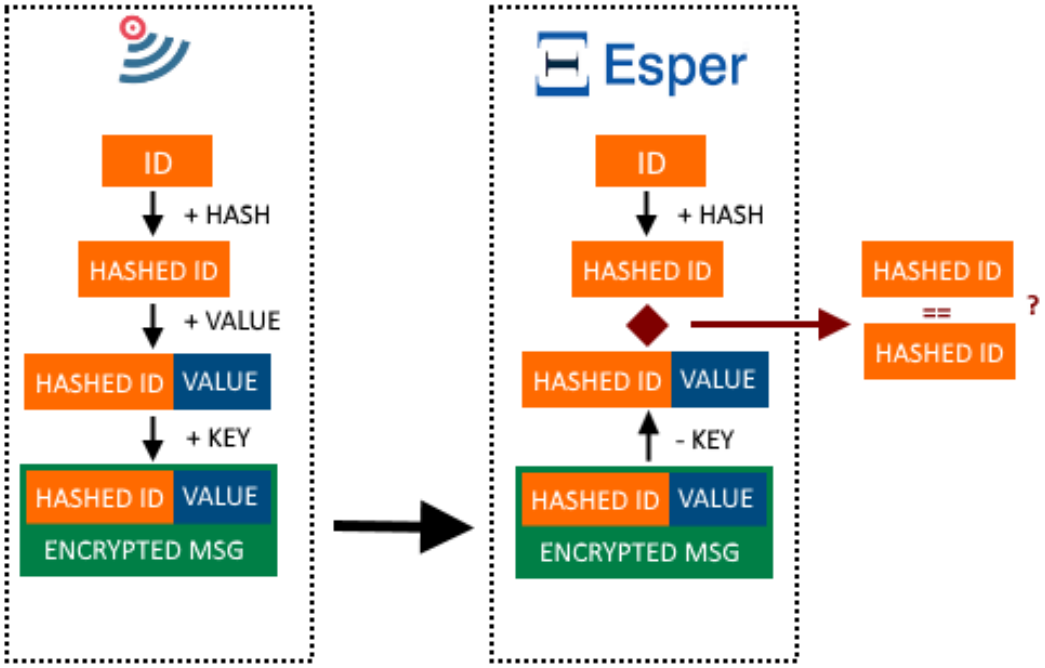


Figure 16 - Proposed solution for validation of data authenticity. Operational flow is represented using arrows. Orange boxes represent string values, such as, IDs. Blue ones represent numerical long value which is the sensor value. Green is the final message that is transferred from the sensor to the ESPER and it consists of hashed sensor ID and value. Decision component (dark red) compares the IDs and validates the sensor authenticity.

Each system component is configured only with its personal ID and key that is used to encrypt transferred data. Server, on the other hand, besides the key has the knowledge of the expected communication devices (their IDs). When the sensor wants to communicate the current value, it must hash its ID and create a message by combining the hashed key and value. After the message string has been created, it is encrypted using the key and the sent to the engine. Then, when the system receives the message, it must be decrypted, using symmetric key, so it is possible to validate the received message. After decryption, the message is constructed by device ID and the value. Since this device ID is hashed and it is not possible to remove the hash, ESPER must hash known

clients' IDs and compare if any of the strings match. If the ID is valid, the engine accepts the received value otherwise it is discarded and this event is logged.

After testing the developed solution, we concluded that the engine is capable of successfully provide data authenticity and process only the data from trustworthy source. Moreover, the proposed solution does not require the usage of certificates, reducing the hardware requirements of the sensors. Since the proposed methodology provides data authenticity and the ESPER became capable of detecting trustworthy information, we decided that it is important to evaluate the performance cost of the added security. Therefore, we performed execution tests that would demonstrate how the execution time is affected, comparing the throughput with and without the data authenticity mechanism. The time measurement was performed on the server side, reducing the possible overhead from the client. That means that sent messages are previously prepared and the engine is constantly receiving the incoming data. As additional comparison, we implemented communication and data transfer using SSL, with authentication and certificates on the server as well as the client sides. All the measurements are performed on the server side, starting after receiving the first record and finished when expected number of operations has been performed.

6.2.2 Setup

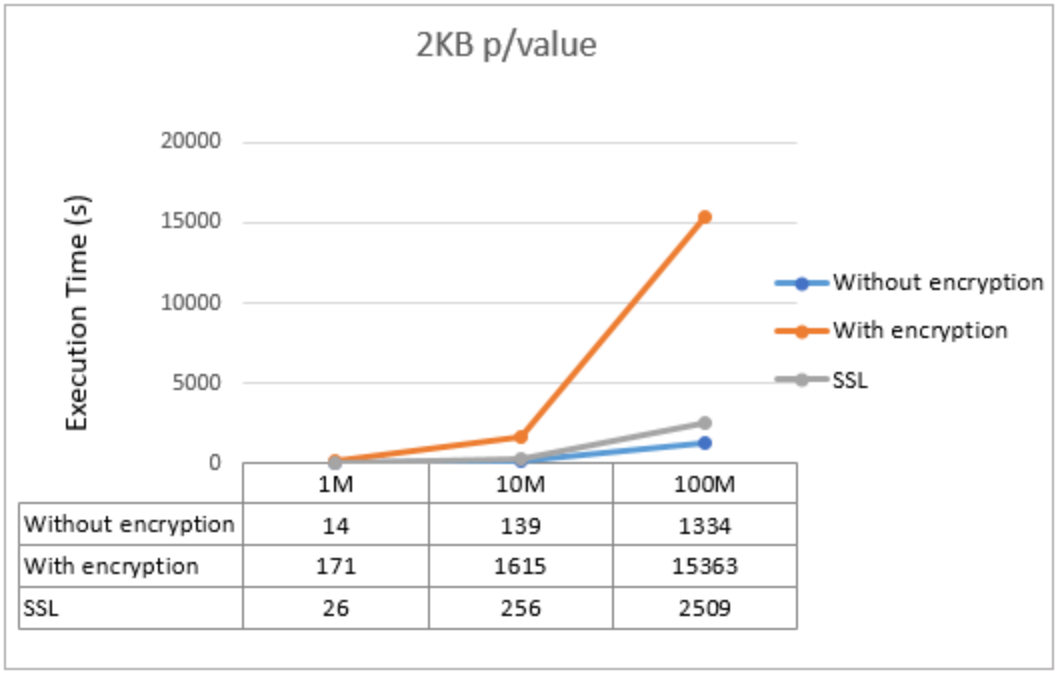
The machine processing characteristics are, as follows:

- Intel Core i5-6600K @ 3.5GHz
- 16GB RAM
- 4 cores / 4 logical processors

It is important to notice that all the tests were performed in the same environment, always using all the 4 available cores. This means that there were no environment variations regardless of the test case (no encryption / with encryption / SSL).

6.2.3 Results

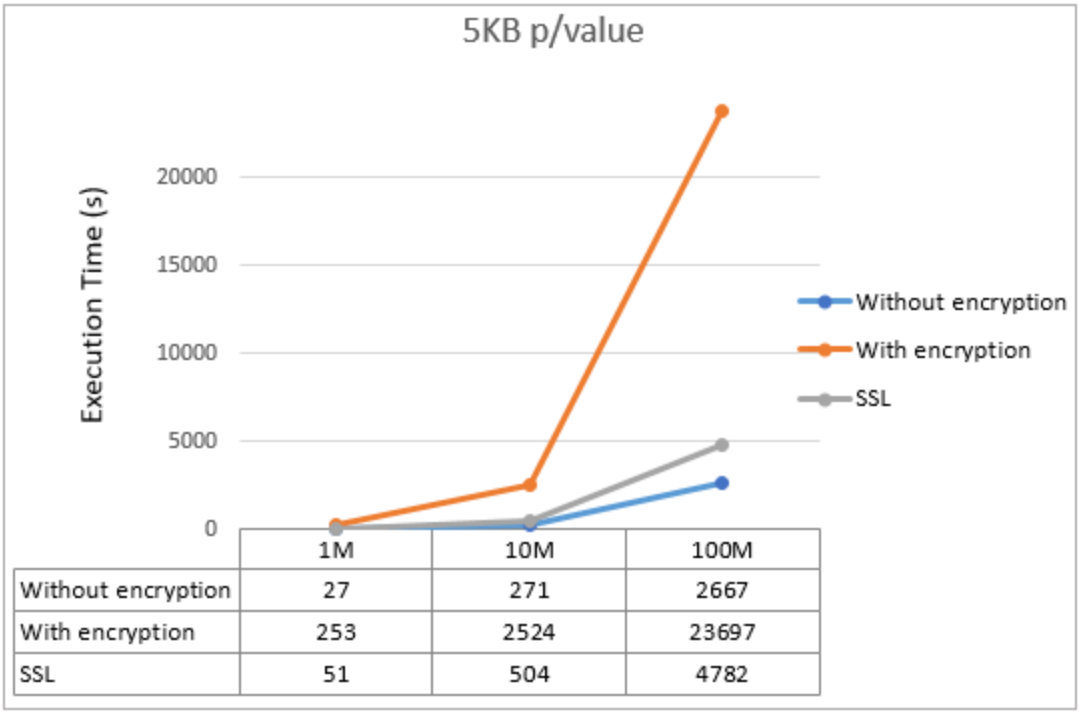
Below we present the obtained results, complete table of test execution is available in Appendix A. To better understand that impact, we varied the number of processed records as well as record size. First, we considered 2KB per value and then increased it to 5KB. Also, were performed 1 000 000, 10 000 000 and 100 000 000 operations.



	1M	10M	100M
Without encryption	14	139	1334
With encryption	171	1615	15363
SSL	26	256	2509

Figure 17 - Execution time, in seconds, with sensor value of 2KB. Each point represents different number of records, starting with 1000000, up to 10000000 and 100000000. Presented numbers are the average of 10 executions

While comparing the results, we concluded that, as expected, lowest execution time is obtained while working without any security mechanisms. Secondly, 1.8 times slower, is the execution time using SSL. But, using AES encryption and SH1 hashing to guarantee data authenticity, data processing demonstrated to be between 11 and 12 times slower when compared to the performance without encryption and between 6 and 6.5 while comparing with SSL. Then, we increased the value size to 5KB (5KB + 3 bytes for header per message), more than twice the size of the previous value and size that is bigger than one page of text. As we expected, the overall execution time, regardless of the security layer, has increased significantly.



	1M	10M	100M
Without encryption	27	271	2667
With encryption	253	2524	23697
SSL	51	504	4782

Figure 18 - Execution time with 5KB per sensor value. Were performed 1000000, 10000000 and 100000000 operations. Each value is average from 10 executions.

Similarly, to the previous results, SLL has average performance impact (in average, requires twice the time to process received data) while usage of encryption and hashing results in more than 8 times higher processing time. The obtained results show the overhead that is added by using data encryption and hashing. As expected, SLL does increases the processing time but still has much better performance than encryption. As previously stated, all the measurements were performed on the ESPER side and the initial handshake between client and server was not considered. However, if the handshake would be performed periodically (to renew the connection) it could possibly affect the continuous operational results. To better understand the processing capabilities, we compared the obtained throughput.

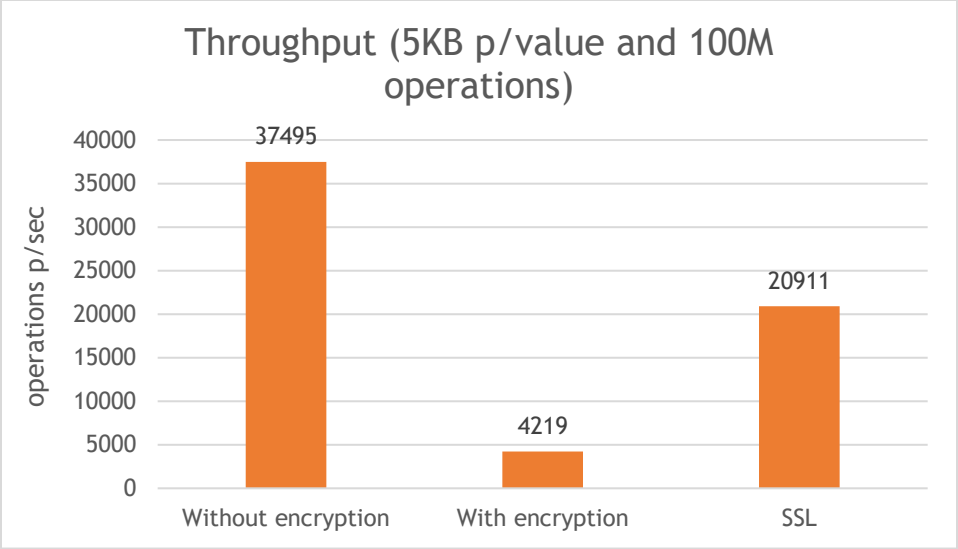


Figure 19 - Throughput obtained during the execution of tests using 5KB of data per value and 100 million of operations

While analyzing the throughput, we observed that the difference in the number of performed operations per second with and without any security mechanisms is noticeable. Due to the processing capabilities, it was possible to achieve a processing speed of over 37K of messages per second. Although low power sensors are not capable of achieving this throughput, the comparison allowed us to understand how the performance is affected by using encryption. When using hashing and AES protocol the throughput reduced more than eight times.

6.2.4 Discussion of results

Overall, we observed that the engine is capable of successfully validate data authenticity and process only the data from trustworthy source. Moreover, the proposed solution does not require any certificates, reducing the hardware requirements of the sensors. While comparing the obtained results, it is possible to conclude that added security mechanisms for validation of data integrity and authenticity creates big performance overhead, especially while comparing with the security that is already provided by SSL. In some tests, the system required up to ten times more time to process all the messages. The main restriction not to use SSL would be the type of the system since while working with components that have more capabilities, SSL would be a better approach that would not affect the performance as much. As expected, without any security ESPER can provide high throughput and processes the incoming messages efficiently. But that means that it would be necessary to add other security mechanisms as a part of the system but outside of the CEP engine. That may be unreasonable since it would be more efficient to validate the data before processing it. Thus, reducing the amount of data to be processed by the CEP (not trusted data is discarded) and overall reducing the processed data by the rest of the system (processed data is filtered against the CEP rules).

7 Conclusions and future work

In this work, we described the quality attributes of the CEP engines and characteristics that are required in safety-critical systems. Currently there are some solutions and application areas, where complex event processing systems are already used and provide better system operability. While describing some of the main attributes, we focused our attention on safety and security. This attention to security characteristics is not considered by most of the authors and available solutions. While, also, considering security mechanisms of CEP engines we compared those to understand the level of security, per engine. Currently there are different available systems and performed overview describes some of the main mechanisms of those engines and compares them between each other. Finally, while focusing on adding security to ESPER, we provided evaluation results that show that data encryption and hashing reduce the engine performance, resulting in up to 12 times higher execution time. This scenario (execution pattern) is noticeable regardless of the message size as well as number of operations, increasing accordingly. As additional tests, we used really small amount of data, 32bytes per value. The observation of those results ensured already detected behavior, proportional increase in execution time to the number of records and use of secure mechanisms.

7.1 Main conclusions

After performing our research, we conclude that safety-critical systems are expected to be able to safely, correctly and efficiently perform critical operations in specific environment, while ensuring that will not endanger any life or property. Thus, those systems should be carefully designed and have a set of qualities and characteristics that would attempt ensuring their proper functioning. Currently there are many available standards that determine quality requirements of a critical system and characteristics that are mandatory in specific areas. Besides robustness and performance, one of the main requirements of a safety-critical system is the security. Managed data as well as the data collection mechanisms, should be protected against outside entities. While focusing on operability, we studied the applicability of the CEP engines in safety-critical systems, separating authentic and relevant data on input. Therefore, we presented some of the most important studies that incorporate the use of CEP engines in software-critical systems. We described performed studies and some of the authors' main goals and conclusions. After concluding our research and while considering the results, we

report that although the authors consider some of the system characteristics, there are other, which are highly important, and are not considered in many studies. We believe that security and safety are equally important quality attributes and have been neglected by some of the available solutions. Those may have high impact while dealing with critical data as well when processing the received information. Both safety and security are focused on providing correct system data by regarding authenticity of the received information, system components safekeeping.

It was determined that most of the studied systems do not provide required security mechanisms that would be strictly necessary in any safety-critical system. Besides understandable communication requirements, some of the engines don't even provide overview of the system (CEP) itself nor security events reporting. Even while not considering third-party access, it is necessary to understand current system status. At present, there are some protection mechanisms that are already used by most of the developed systems, such as, authentication, data encryption, safe connections, hardware authentication and register, etc.

It is possible to conclude that one of the basic mechanisms, authentication, is present in all the studied engines. However, it is not nearly enough security for a complete safety-critical system. Mostly, CEP engines are expected to provide only data processing, focusing mainly on the operational performance, while the remaining of the system should provide all the required data security. That would be acceptable if there would be no processing overhead. For example, during data stream reading from the outside of the system, if the CEP engine is not capable of hardware identification as well as the connection encryption, adding those mechanisms to the system after the data has been received would increase the amount of data that would be unnecessary processed. It is more efficient to detect not trustworthy data on the early stages, reducing the processing time.

While considering studied engines, we determined that the CEP engine that could potentially be used as a part of safety-critical system is the Oracle CEP. This engine provides some of the basic safety and security mechanisms and does not creates an additional load on the rest of the system architecture. This engine is followed by Sybase Aleri, Apache Flink and Drools, ordered by the quantity of provided security. Finally, ESPER is the CEP engine that focuses mainly of the processing. This engine could be an alternative due to its performance, but for it to be used in safety-critical system, it would be necessary adding security mechanisms outside of the CEP and that could compromise the overall system reliability.

7.2 Limitations

The main limitation of performed study is the usage of synthetic workload. Used sensor values are not real and although those can represent the message itself and allowed us to test the execution time of operations, real values could benefit the study even more. Also, the performed experiments were not conducted in deployed and working system with ESPER being part of it. In this case the simulated tests were run over ESPER as standalone instance and not as part of a bigger and more complete system. This testing environment does not completely represent a fully functional system, having all the necessary processing and UI components.

7.3 Directions for future research

As future work, we will develop a complete system and add all the necessary hardware components that would fully represent communication between real sensor and central system. This environment modification would allow us to fully test, in a real environment, the developed solution. While working in real environment it is possible to compare different available sensors and understand how their hardware characteristic compare between each other and which of those have better performance. While working with security and encryption algorithms, not only there are different configurations per algorithm, such as, key size, but also there is a variety of available encryption solutions. It would be important to understand the impact of algorithms on processing capabilities and detect sensors with lower performance. Finally, it would be important to investigate a security solution that could be integrated as a part of ESPER, as a final product. That mechanism could provide security certifications to this CEP engine.

References

Aftab, A.H, Nadeem, A. (2013). A Survey of Safety Analysis Techniques for Safety Critical Systems. *IJFCC 2013 Vol.2(2): 134-137 ISSN: 2010-3751*.

Aidi, A. (2006). Complex Event Processing. *IBM Haifa Labs*

Aidi, A., Botzer, D., Nechushtai, D., Sharon, G. (2006). Complex Event Processing for Financial *Services. Services Computing Workshops, 2006. SCW '06. IEEE*.

Ali, M., Gerea, C., Sezgin, B., Tarnavski, T., Verona, T., Wang, P., Zabback, P., Kirilov, A., Ananthanarayan, A., Lu, M., Raizman, A., Krishnan, R., Schindlauer, R., Grabs, T., Bjeletich, S., Chandramouli, B., Goldstein, J., Bhat, S., Li, Y., Di Nicola, V., Wang, X., Maier, D., Santos, I., Nano, O., Grell, S., Raman, B. S. (2009). Microsoft CEP Server and Online Behavioral Targeting. *International Conference on Very Large Data Bases (VLDB).*

Atmel Corporation. (2011). 8-bit Atmel Microcontroller with 128Kbytes In-System Programmable Flash

Atoum, I., Bong, C. (2015). Measuring Software Quality in Use: State-of-the-Art and Research Challenges. Measuring Software Quality in Use: State-of-the-Art and Research Challenges.

Avila, L. and Martínez, E. (2015). An active inference approach to on-line agent monitoring in safety–critical systems. *Advanced Engineering Informatics 29 (2015) 1083–1095*.

Axelsson, J., Nyfjord, J., Papatheocharous, E., Törngren, M. (2016). Notes On Agile and Safety-Critical Development. *ACM SIGSOFT Software Engineering Notes Page 23 March 2016 Volume 41 Number 2.*

Baldoni, R., Montanari, L., Rizzuto, M. (2015). On-line failure prediction in safety-critical systems. *Future Generation Computer Systems 45 (2015) 123–132*.

Balogh, L., István, D., István, R., Dániel, V., András, V. (2016). Distributed and Heterogeneous Event-based Monitoring in Smart Cyber-Physical Systems.

Barbacci, M., Longstaff, T., Klein, M., Weinstock, C. (1995). Quality Attributes. *Software Engineering Institute Carnegie Mellon University. Technical Report CMU/SEI-95-TR-021; ESC-TR-95-021*

Beer, A., Heindl., M. (2007). Issues in Testing Dependable Event-Based Systems at a Systems Integration Company. *Second International Conference on Availability, Reliability and Security (ARES'07).*

Bhochhibhoya, R. (2015). https://github.com/bhochhi/drools-guide/wiki/how-to-integrate-LDAP-with-drools-workbench-for-authentication-and-authorization%3F.

Bierl, L. (2000). MSP430 Family Mixed-Signal Microcontroller Application Reports. *Texas Instruments*

Bowen, J., Stavridou, V. (2002). Safety-Critical Systems, Formal Methods and Standards. *Software Engineering Journal 8(4).*

Brotcorne, L., Laporte, G., Semet, F. (2003). Ambulance location and relocation models. *European journal of operational research, vol. 147, no. 3, pp. 451–463.*

Bruns, R., Dunkel, J., Billhardt, H., Lujak, M., Ossowski, S. (2014). Using Complex Event Processing to Support Data Fusion for Ambulance Coordination. *Information Fusion (FUSION), 2014 17th International Conference.*

Carbone, P., Ewen, S., Haridi, S., Katsifodimos, A., Mark, V., Tzoumas, K. (2015). Apache Flink™: Stream and Batch Processing in a Single Engine. *Bulletin of the IEEE Computer Society Technical Committee on Data Engineering*.

Carbone, P., G. Fora, S. Ewen, S. Haridi, and K. Tzoumas. (2015). Lightweight asynchronous snapshots for distributed ´ dataflows. *arXiv:1506.08603.*

Carvalho, J. (2008). Use of Complex Event Processing for Business Optimization. *TIBCO Software. Banca & Seguros'08.*

Cerullo, G., Formicola, V., Iamiglio, P., Sgaglione, L. (2014). Critical Infrastructure Protection: having SIEM technology cope with network heterogeneity.

Chang, I.P., Lee, T.F., Lin, T.H., Liu, C.M. (2015). Enhanced Two-Factor Authentication and Key Agreement Using Dynamic Identities in Wireless Sensor Networks. *Sensors 2015, 15, 29841–29854.*

Chen, C. and Chao, H. (2011). A survey of key distribution in wireless sensor networks. Security snd Communication Networks Security Comm. *Networks 2014; 7:2495–2508.*

Ciampolini, A., Mello, P., Storari, S. (2004). A multi-agent system for medical services synergy and coordination. *International ECAI 2004 workshop on Agents applied in health care, J. Nealon, U. Cortes, J. Fox, and A. Moreno, Eds., p. 3846.*

Cockburn, D. (2016). Dealing with Quality Defects and Rapid Alerts. *Available at:* http://www.emAmmona.europa.eu/docs/en_GB/document_library/Presentation/2009/12/WC500017884.pdf.

Collins, I. (2013). Chapter 1: Safety-Critical Computer System Design and Evaluation. *Howard University Department of Electrical and Computer Engineering.*

Elzinga, T. (2015). Assessing IT solutions with CORA. *http://www.coramodel.com/assessing-it-solutions/*

Crossbow Technology. (2004). MICA2: Wireless Measurement System. http://www.xbow.com/Products/Product pdf files/Wireless pdf/6020-0042-04 A MICA2.pdf.

Daum, M., Götz, M., Domaschka, J. (2012). Integrating CEP and BPM: how CEP realizes functional requirements of BPM applications. *DEBS '12 Proceedings of the 6th ACM International Conference on Distributed Event-Based Systems. Pages 157-166.*

Dekkers, P. (2007). Master Thesis Computer Science. Complex Event Processing. *Radboud University Nijmegen, Thesis number 574, October 2007.*

Drools. (2017). http://www.drools.org/.

Eckert, M. and Bry, F. (2009). Complex Event Processing. German language in Informatik Spektrum. *Springer 2009.*

Esper. (2017). http://www.espertech.com/products/esper.php

EsperTech. (2017). http://www.espertech.com/esper/release-5.2.0/esper-reference/html/configuration.html.

EsperTech. (2017). http://www.espertech.com/esper/faq_esper.php

EsperTech. (2017). http://www.espertech.com/products/esperhq.php

Evchina, Y., Lastra, J. L. (2016). Hybrid approach for selective delivery of information streams in data intensive monitoring systems. *Advanced Engineering Informatics 30 (2016) 537–552.*

Ewen, S. (2016). The Stream Processor as a Database. *Hadoop Summit.*

Felder, R., & Silverman, R. (1988). Learning and Teaching Styles in Engineering Education. *Journal of Engneering Education 78 (7)*, 674-681.

Ficco, M., Romano, L. (2011). A generic intrusion detection and diagnose system based on complex event processing. *Proceedings on the 1st International Conference on Data Compression, Communication, and Processing, p. 275-284.*

Flink. (2017). https://ci.apache.org/projects/flink/flink-docs-release-1.0/quickstart/run_example_quickstart.html.

Flink. (2017). http://apache-flink-user-mailing-list-archive.2336050.n4.nabble.com/Security-in-Flink-td4202.html.

Flink. (2017). https://ci.apache.org/projects/flink/flink-docs-release-1.2/ops/security-kerberos.html.

Flink. (2017). https://ci.apache.org/projects/flink/flink-docs-release-0.8/cluster_setup.html.

Flink. (2017) https://ci.apache.org/projects/flink/flink-docs-release-1.0/quickstart/run_example_quickstart.html.

Foley, J., Churcher, G.E. (2009). Applying Complex Event Processing and Extending Sensor Web Enablement to a Health Care Sensor Network Architecture. *Conference: Proceedings of the 4th International Conference on Communication System Software and Middleware*.

Fouchal, H., Biesa, J., Romero, E., Araujo, A., Taladrez, O.N. (2016). A Security Scheme for Wireless Sensor Networks. *Global Communications Conference (GLOBECOM), 2016 IEEE.*

Gadgil, M. (2012). Aleri – Complex Event Processing. https://www.javacodegeeks.com/2012/04/aleri-complex-event-processing.html

Gandino, F., Ferrero, R., Rebaudengo, M. (2017). A Key Distribution Scheme for Mobile Wireless Sensor Networks: q-s-Composite. *IEEE Transactions on Information Forensics and Security, Vol. 12, No. 1, January 2017*.

Granjal, J., Monteiro, E., Silva, J. (2015). Security in the integration of low-power Wireless Sensor Networks with the Internet: A survey. *Ad Hoc Networks 24 (2015) 264–287.*

Hamill, M. and Goseva-Popstojanova, K. (2014). Exploring fault types, detection activities, and failure severity in an evolving safety-critical software system. *Software Qual J (2015) 23:229–265.*

He, D.B., Kumar, N., Chen, J.H., Lee, C.C., Chilamkurti, N., Yeo, S.S. (2013). Robust anonymous authentication protocol for health-care applications using wireless medical sensor networks. *Multimedia Systems: In press, 10 Dec 2013.*

Heck, E. v., & Vervest, P. (2007). Smart business networks: how the network wins. Communications of the ACM, Vol. 50 No. 6, 28-37.

IBM. (2012). Security Secure Sockets Layer (SSL). *Version 5 Release 4.*

IEEE. (2009). 603-2009 - IEEE Standard Criteria for Safety Systems for Nuclear Power Generating Stations. Revision of IEEE Std 603-1998.

Islam, K., Shen, W. and Wang, X. (2012). Wireless Sensor Network Reliability and Security in Factory Automation: A Survey. *IEEE Transactions On Systems, Man, And Cybernetics—Part C: Applications And Reviews, Vol. 42, No. 6, November 2012.*

Itria, M.L., Daidone, A., Ceccarelli, A. (2014). A Complex Event Processing Approach for Crisis-Management Systems. *EDCC Workshop Big4CIP*.

Javanbakht, M., Erfani, H., Javadi, H., Daneshjoo, P. (2014). Key Predistribution Scheme for Clustered Hierarchical Wireless Sensor Networks based on Combinatorial Designs. *Security Comm. Networks 2014; 7:2003–2014.*

JBoss. (2017). https://docs.jboss.org/drools/release/5.5.0.Final/drools-guvnor-docs/html_single/#d0e42.

JBoss. (2017). https://www.jboss.org/dms/judcon/2012india/presentations/day1track3session2.pdf.

JBoss. (2017). https://issues.jboss.org/browse/DROOLS-1221.

JBoss. (2017). https://docs.jboss.org/drools/release/6.2.0.CR3/drools-docs/html/DroolsComplexEventProcessingChapter.html.

Jung, J., Moon, J., Lee, D., Won, D. (2016). Efficient and Security Enhanced Anonymous Authentication with Key Agreement Scheme in Wireless Sensor Networks. *Sensors*.

Klendar, N. (2017). https://xakep.ru/2014/12/24/esper/#toc04.

Lang, J., Capík, Z. (2014). Prediction based on hybrid method in complex event processing. SAMI 2014. *IEEE 12th International Symposium on Applied Machine Intelligence and Informatics, January 23-25, 2014. Herl'any, Slovakia.*

Laranjeiro, N., Soydemir, Seyma., Bernardino, J. (2015) A Survey on Data Quality: Classifying Poor Data. *21st IEEE Pacific Rim International Symposium on Dependable Computing, PRDC 2015, Zhangjiajie, China, pp. 179-188*.

Li, L., Cao, B., Liu, Y. (2013). A study on CEP-based system status monitoring in cloud computing systems. *6th International Conference on Information Management, Innovation Management and Industrial Engineering, 1, pp. 300–303.*

Li, X., Niu, J., Kumari, S., Liao, J., Liang, W., Khan M. (2015). A new authentication protocol for healthcare applications using wireless medical sensor networks with user anonymity. *Security Comm. Networks 2016; 9:2643–2655.*

Li, X., Zhao, Z., Zhu, X., Wyatt, T. (2011). Covering models and optimization techniques for emergency response facility location and planning: a review. *Mathematical Methods of Operations Research, vol. 74, no. 3, pp. 281–310.*

Lu, X., Yin, J., Xiong, N., Deng, S., He, G., Yu, H. 2016. JTangCMS: An efficient monitoring system for cloud platforms. *Information Sciences 370–371 (2016) 402– 423.*

Luckham, D. (2008). The Power of Events: An Introduction to Complex Event Processing in Distributed Enterprise Systems. *International Workshop on Rules and Rule Markup Languages for the Semantic Web, pp 3-3*

Luckham, D. (2002). The Power of Event. *Addison Wesley.*

Malan, D. J., Welsh, M., Smith, M. D. (2008). Implementing public-key infrastructure for sensor networks. *ACM Transactions on Sensor Networks (TOSN) 4.4 (2008): 22*.

Mendes, M., Bizarro, P., Marques, P. A Performance Study of Event Processing Systems. (2009). Performance Evaluation and Benchmarking. *Volume 5895 of the series Lecture Notes in Computer Science pp 221-236.*

Microsoft. (2017). https://www.microsoft.com/en-us/trustcenter/Compliance/SOC.

Mollenkopf, A., Tirelli, E. (2009). Applying Drools Fusion Complex Event Processing (CEP) for Real-Time Intelligence. *JBoss World Chicago 2009.*

Moraru, A. (2012). Methods for Complex Event Processing. Doctoral degree.

Naqishbandi, T., Sheriff, I., Qazi, S. (2015). Big Data, CEP and IoT: Redefining Holistic Healthcare Information Systems and Analytics. *International Journal of Engineering Research & Technology (IJERT). Vol. 4 Issue 01*.

NIST. (2013). Security and Privacy Controls for Federal Information Systems and Organizations. *NIST Special Publication 800-53, Revision 4.*

NIST. (2010). Guide for Applying the Risk Management Framework to Federal Information Systems. *Computer Security Division Information Technology Laboratory National Institute of Standards and Technology Gaithersburg, MD 20899-8930.*

NPES. (2005). Standards: What Are They and Why Are They Important? *NPES Standards Bluebook*

Oberoi, S. (2007). Introduction to Complex Event Processing & Data Streams. *A white-paper by Real-Time innovations.*

Oedewald, P. and Gotcheva, N. (2015). Safety culture and subcontractor network governance in a complex safety critical project. *Reliability Engineering and System Safety 141 (2015) 106–114.*

Ollesch, J. (2016). Doctoral Symposium: Adaptive Steering of Cyber-Physical Systems with Atomic Complex Event Processing Services. *DEBS '16 June 20 - 24, 2016, Irvine, California, USA.*

Oracle. (2012). Financial Services Data Management: Big Data Technology in Financial Services. An Oracle White Paper.

Oracle. (2008). Oracle Complex Event Processing Performance. An Oracle White Paper.

Oracle. (2010). Oracle Complex Event Processing High Availability. An Oracle White Paper.

Oracle. (2017). https://docs.oracle.com/cd/E13213_01/wlevs/docs30/config_server/security.html

Oracle. (2017). https://docs.oracle.com/cd/E13157_01/wlevs/docs30/config_server/security.html

Oracle. (2017). https://docs.oracle.com/cd/E12839_01/doc.1111/e14302/server_tasks.html

Oracle. (2010). Oracle Complex Event Processing High Availability. An Oracle White Paper.

Parnas, D.L., Schouwen, A.P., Kwan, S.P. (1990). Evaluation of SafetyCritical Software. *Communications of the ACM Volume 33 Issue 6, June 1990, Pages 636-648.*

Prassinos A. (2009). Active / Active Configurations with Oracle Active Data Guard. *Distinguished Member of Technical Staff MorphoTrak, SAFRAN Group. Oracle Open World 2009.*

Saboor, M., Rengasamy, R. (2013). Designing and Developing Complex Event Processing Applications. *Sapient corporation.*

Schmidt, K., Anicic, D., Stuhmer, R. (2012). Event-driven Reactivity: A Survey and Requirements Analysis.

Secure. (2016). Secure! Project, http://secure.eng.it/

Sommerville, I. (2004). Software Engineering (7th Edition). *Pearson Addison Wesley.*

Stouffer, K. and Katzke, S. (2008). Industrial Control System Security and NIST SP 800-53 Overview. *National Institute of Standards and Technology.*

Sybase. (2010). Administrators Guide, Sybase Aleri Streaming Platform 3.1. *Sybase, Inc.*

Sybase. (2010). Frequently Asked Questions (FAQ), Sybase Aleri Streaming Platform 3.1. http://infocenter.sybase.com/help/topic/com.sybase.infocenter.dc01297.0311/pdf/FAQ.pdf%3Fnoframes%3Dtrue. *Sybase, Inc.*

Sybase. (2010). Product Overview, Sybase Aleri Streaming Platform 3.1. http://infocenter.sybase.com/help/topic/com.sybase.infocenter.dc01286.0311/pdf/ProductOverview.pdf?noframes=true. *Sybase, Inc.*

Sybase. (2010). Introduction to Data Modeling and the Aleri Studio. Sybase Aleri Streaming Platform 3.1. *Sybase, Inc.*

Tendick, P., Denby, L., Ju, W. (2016). Statistical methods for complex event processing and real time decision making. *WIREs Comput Stat 2016, 8:5–26.*

Viatra. (2016). VIATRA-CEP Documentation. https://wiki.eclipse.org/VIATRA/CEP.

Vincent, P. (2016). http://www.complexevents.com/2016/05/12/cep-tooling-market-survey-2016/.

Wahl, A. and Hollunder, B. Performance Measurement for CEP Systems. (2012). *Service Computation: The Fourth International Conferences on Advanced Service Computing.*

Wander, Arvinderpal S., et al. "Energy analysis of public-key cryptography for wireless sensor networks." Pervasive Computing and Communications, 2005. PerCom 2005. Third IEEE International Conference on. IEEE, 2005.

Wang, D., Rundensteiner, E., Wang, H., Ellison, R. (2010). Active complex event processing: applications in realtime health care. *Proceedings of the VLDB. Volume 3 Issue 1-2, September 2010, Pages 1545-1548.*

Wang, Y., Kuang, L. (2015). Traffic prediction method based on complex event processing and adaptive bayesian networks. *The 2015 International Academic Research Conference, 3-6 August, University of London.*

Wang, Y., Zhou, X., Shan, L., Miao, K. (2014). Study on Complex Event Processing for CPS: An Event Model Perspective.

White, S., Alves, A., Rorke, D. (2008). WebLogic event server: a lightweight, modular application server for event processing. *In Proc. of DEBS 2008.*

Wolf, P., Schmidt, A., Klein, M. (2009). Applying Semantic Technologies for Context-Aware AAL Services: What we can learn from SOPRANO. *Workshop on Applications of Semantic Technologies 09.*

Won, T. (2016). https://tedwon.atlassian.net/wiki/display/SE/CEP#CEP-CEPProducts.

Yao, W., Chu, C., Li, Z. (2011). Leveraging complex event processing for smart hospitals using RFID. *Journal of Network and Computer Applications, 34, 799– 810.*

Youn, W. and Yi, B. (2014). Software and hardware certification of safety-critical avionic systems: A comparison study. *Computer Standards & Interfaces 36 (2014) 889–898.*

Žáková, I. (2013). Drools Fusion and Utilization of Complex Event Processing in Web Applications. Master's Thesis.

Zappia, I., Paganelli, F., Parlanti, D. (2012). A lightweight and extensible Complex Event Processing system for sense and respond applications. *Expert Systems with Applications 39, 10408–10419.*

Zappia, I., Ciofi, L., Paganelli, F., Iadanza, E., Gherardelli, M., Giuli, D. (2014). A distributed approach to Complex Event Processing in RFID-enabled hospitals. *Euro Med Telco Conference (EMTC).*

Zhang, L. and Zhang, H. (2016). A Survey on Security and Privacy in Emerging Sensor Networks: From Viewpoint of Close-Lo Won op. *Sensors.*

Zio, E. (2016). Challenges in the vulnerability and risk analysis of critical infrastructures. *Reliability Engineering and System Safety 152 (2016) 137–150.*

Appendix A – Test executions

Below we present the results of executed test, per run. Time values are in seconds.

2048 bytes p/sensor value

Nº of records	*Without encryption*	*With encryption*	*With SSL*
1000000	15	168	24
	14	165	26
	14	170	26
	14	173	27
	13	169	26
	14	168	26
	14	174	24
	15	178	24
	14	177	26
	14	167	27

Nº of records	*Without encryption*	*With encryption*	*With SSL*
10000000	137	1458	247
	140	1698	254
	126	1630	267
	142	1638	250
	139	1668	267
	139	1444	257
	139	1685	248
	140	1682	244
	142	1638	267
	142	1605	255

Nº of records	*Without encryption*	*With encryption*	*With SSL*
100000000	1390	15426	2527
	1341	15182	2585
	1350	15191	2568
	1207	15894	2467
	1319	15334	2490
	1380	15187	2416

1367	15361	2437
1277	15369	2550
1370	15192	2562
1343	15490	2488

5120 bytes p/sensor value

Nº of records	*Without encryption*	*With encryption*	*With SSL*
1000000	27	250	52
	27	255	52
	27	253	52
	28	255	50
	27	252	50
	27	248	51
	27	258	52
	27	252	51
	26	255	51
	27	255	51

Nº of records	*Without encryption*	*With encryption*	*With SSL*
10000000	274	2563	512
	273	2517	513
	274	2500	509
	274	2515	511
	273	2541	505
	253	2540	489
	255	2521	512
	282	2497	490
	274	2541	498
	276	2507	505

Nº of records	*Without encryption*	*With encryption*	*With SSL*
100000000	2645	23569	4657
	2678	23515	4825
	2677	23374	4784
	2643	24016	4635
	2684	24001	4913
	2658	23560	4893
	2676	23781	4860
	2648	23566	4877
	2692	23570	4690

	2671	24020	4682

32 bytes p/sensor value

Nº of records	*Without encryption*	*With encryption*	*With SSL*
1000000	4	89	10
	4	88	11
	4	88	12
	4	89	11
	6	87	12
	3	88	11
	6	89	11
	4	89	11
	4	89	10
	6	89	12

Nº of records	*Without encryption*	*With encryption*	*With SSL*
10000000	55	888	97
	55	826	101
	50	854	102
	38	912	97
	46	895	102
	45	927	102
	45	846	100
	51	851	97
	50	900	101
	41	924	100

Nº of records	*Without encryption*	*With encryption*	*With SSL*
100000000	423	8128	1021
	409	8120	971
	507	9249	1031
	507	8266	1008
	433	8203	964
	426	8105	1000
	430	8140	1024
	440	8341	980
	433	9342	1021
	500	8196	1016

Appendix B – Published paper – COMPLEXIS'17

ON THE USE OF CEP IN SAFETY-CRITICAL SYSTEMS

Veronika Abramova[1], Bruno Cabral[2] and Jorge Bernardino [1,2]
[1] *Polytechnic Institute of Coimbra, ISEC - Coimbra Institute of Engineering, Coimbra, Portugal*
[2] *University of Coimbra – CISUC, Centre for Informatics and Systems of University Coimbra, Coimbra, Portugal*
a21190319@alunos.isec.pt, bcabral@dei.uc.pt, jorge@isec.pt

Keywords: Safety-critical, CEP, Systems, Event processing.

Abstract: Safety-critical systems have to continuously manage risks, in order to handle hazardous situations and still be able to fulfil their purpose. While being composed by a variety of software, as well as hardware components, it is necessary for each part of these systems, alone and as a whole, to exhibit a required set of characteristics, necessary to ensure the correct system functioning. Complex Event Processing (CEP) systems have been used in a diversity of applications and, while they focus on fast data gathering and processing as well as in providing intelligence to their users, there is incomplete information about how they are adequate to integrate safety-critical systems. In this paper we investigate if the mainstream off-the-shelf CEP systems are suitable for safety-critical applications. We describe the use of complex event processing engines in safety-critical systems and how some authors enhance those to better correspond to the critical system requirements. We demonstrate that, although dependability is well handled in most CEP systems, the same cannot be assumed about security and safety attributes.

1 INTRODUCTION

Safety-critical systems are projected and designed to be used in avionics, medical devices, automotive braking systems, nuclear power plant management, flight management systems, chemical processes, nuclear power plants and other potentially life threatening systems, when failure in the system endangers human lives directly or indirectly. Therefore, these systems should be able to provide the required safety functions and overall system integrity (Knight, 2002). In order to be able to mitigate the potential impact of hazardous situations, safety-critical systems must be certified by a regulatory agency to ensure their correct operation. Consequently, it is important that systems in use are reliable while it is possible to detect design and functional problems. Also, those should be able to meet some of the most important quality attributes, such as dependability, reliability, safety and security (Malm et al., 2011). It is important to mention that those are complex systems, which are mostly build using different operational modules and systems combined together, such as Complex Event Processing (CEP) systems. CEP systems have been around for quite some time and have been constantly tested, improved and developed (Mendes, Bizarro & Marques, 2009). Due to the wide range of possible application domains, CEP system capabilities have been constantly improving over the past years. Figure 1 presents an example of a CEP system architecture – ESPER.

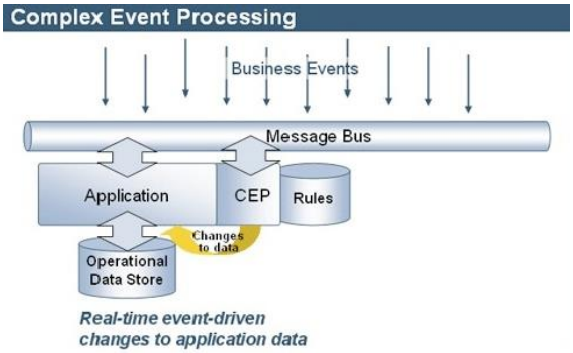


Figure 1: ESPER architecture (Esper, 2016).

Currently they are used in a variety of software systems as a customizable tool for data processing and analysis. These systems have the ability to completely adapt to the desired system purpose and be able to process and present most important data, based on previously defined rules. Therefore, traditionally, CEP systems are part of more complex systems, inside which they are used to process incoming information, create alerts based on a collection of rules, and posteriorly provide filtered and summarized data to the final user (Eckert & Bry, 2009). Consequently, those systems are focused on information extraction and processing of important data in real-time environments.

We investigate if current off-the-shelf CEP systems are suitable for usage in safety-critical applications, and are capable of performing designed operations in the required (acceptable) time. As our main contribution, we study some critical systems that use CEP engines, as well the improvements that were made to these CEP engines by some of the authors. We start by identifying the key quality attributes of safety-critical systems and use these to study how some of the available systems manage those requirements. This analysis allowed us to identify different characteristics of CEP systems that require more research.

The remainder of the paper is structured as follows: section 2 presents the state of the art and some of the application areas of CEP systems. Section 3 describes some of the most important quality attributes for safety-critical systems. Section 4 studies the usage of CEP in safety-critical applications and Section 5 evaluates if those quality attributes were considered by the authors and describes possible gaps of usage of CEP in safety-critical systems in some of the existent studies. Finally, Section 6 presents our conclusions and proposes some future research directions.

2 STATE OF THE ART

CEP systems have been used in a large number of systems for data processing and alarm triggering. Although our focus is on safety-critical applications, in this section, to correctly assess the state-of-the-art, we will also consider works reporting on the application of complex event processing in other areas.

For example, (Aidi, 2006) describes some of the most common areas of application of complex event processing systems, where these engines are capable of exploring their full potential. The author presents some of the key benefits brought by those systems. A more focused study is performed by (Oracle, 2012) and (Aidi et al., 2006), who emphasize their approach on financial area and propose solutions using CEP engines. All of the authors state that CEP systems have a high potential for processing data when it arrives and are capable of providing data of interest in a satisfactory amount of time. Since the main goal of CEP is to be agile and efficient in data presentation to the end user, (Cockburn, 2016), (Schmidt, Anicic & Stuhmer, 2012) and (Ammon et al., 2009) investigate the efficiency of those systems in alarm triggering and alert management. These authors present the advantages of CEP systems in those cases and describe

how those can be tuned to be able to provide all the required information. Those systems may be focused on alarms in dangerous environments as well as noncritical businesses that completely rely on getting information in time. Therefore, (Daum, Götz & Domaschka, 2012) focused mainly on BPM - Business Process Management Systems and describe if CEP are capable of satisfying all the requirements of those systems.

Healthcare is another highly important area that could potentially take advantages of CEP systems. (Wang et al., 2010) describe real-time healthcare applications and state that CEP engines can efficiently monitor patient behaviour and control medical regulations. Similarly, (Foley & Churcher, 2009) and (Naqishbandi, Sheriff & Qazi, 2015) propose architecture solutions and required characteristics of the systems using complex event processing in healthcare domain. Authors state that CEP engines are highly useful for large and critical data processing and are capable of improving medical systems.

Our work presents some of the available solutions and studies, performed by other authors, which focus on application of CEP engines in safety-critical systems. We attempt describing some of the systems developed by them and some of the authors' conclusions.

3 SAFETY-CRITICAL REQUIREMENTS

While developing a safety-critical system there is a wide range of standards that must be followed as well as system requirements (Bowen & Stavridou, 2002). Those requirements may be also translated into system quality attributes and define the system characteristics without getting deep inside into its functionalities. Currently software systems present, in most cases, a long list of quality attributes (Laranjeiro, Soydemir & Bernardino, 2015) and, therefore, we decided to consider in this work the most general ones, which are fundamental in safety-critical systems (see Figure 2).

Figure 2: Quality attributes.

We based our choice on the available standards and attributes presented by (Barbacci et al., 1995), (Sommerville, 2004), (Atoum & Bong, 2015). Therefore, the considered quality attributes are:

- **Dependability.** Can be directly translated into trustworthiness of the developed system and represents the confidence in the correct functioning of its functions. One of the dimensions of dependability is fault tolerance that states that failure in a part of the system cannot compromise the whole system. Also, should be considered the repair capability of the designed system, which describes if the system is capable of recovering without any intervention, as expected.
- **Security.** Ensure that extracted, stored and processed by the system data is not easily intersected or corrupted. There should not be intentional disruption by the third parties. Authentication mechanism is also a requisite since it allows user identification and extends up to the possibility of the system to recognize the configured devices and treat them as trusted. The collected data will be recognized as viable modifications and not by others. System vulnerability should be reduced as much as possible in order to prevent possible harm by attackers and important data exposure.
- **Reliability.** Describes the probability of the system of performing designed operations in expected time. This attribute is tightened with the availability of the system that states that the system should have the ability to work with limited amount of data when it cannot be collected. There should be available backup data collectors that would gather at least part of the necessary data. Therefore, reliability considers not only software but hardware and firmware modules that are required for correct system functioning.
- **Safety.** This is another quality attribute highly related to the reliability. Thus, safety reflects the system's ability to operate, normally or abnormally, without danger of causing human injury or death and without damage to the system's environment. Some of related safety terms are: hazard, damage and risk. All of those are considered by the standards and usually are described as a prevention list of measures instead of qualities.

We believe that, among others, those are some of the most important requirements of critical software. Also, chosen attributes enclose other vital software system characteristics, for example, performance, resilience, availability, etc.

4 CEP IN SAFETY-CRITICAL SYSTEMS

(Baldoni, Montanari & Rizzuto, 2015) proposed a solution that combines Complex Event Processing (CEP) and Hidden Markov Models (HMM) to analyse system failures and their symptoms considering specifically defined metrics. The anomalies conditions are detected using the defined rules and creating alert events. CEP provides all the necessary data as well as performance metrics while HMM are used for system state specification and recognition. Authors state that it is important to be able to detect faults in system components in order to be able to prevent the entire system to be compromised. Hence, this work proposes a failure prediction architecture, focused on the traffic control systems, named CASPER. Authors state that CASPER exhibits pretty good accuracy and it is able to generate predictions with a margin of time that allows recovery actions to mitigate the upcoming occurrence of a failure of the system.

Considering usage of CEP in different domains and with increased interest in CEP systems, (Zappia, Paganelli & Parlanti, 2012) describe the design and the implementation of a lightweight and extensible Complex Event Processing engine, called LiSEP. During the system design specification, authors were driven by the principle of minimizing dependency on external software components and, therefore, LiSEP depends solely on the Java Standard Edition libraries, thus minimizing deployment requirements. Moreover, the LiSEP logic is strictly focused on core event processing, consequently resulting in a lightweight and minimal implementation. The proposed solution is complemented by the specification of the Event Processing Language, based on the SQL syntax. As a proof of the architecture, authors propose experimenting the use of the LiSEP engine in a case study on dangerous goods monitoring during maritime transport as a part of Italian Ministry for Economic Development research project, called SITMAR - Integrated system for goods maritime transport in multi-modal scenarios. More data-focused approach is presented by (Evchina & Lastra, 2016). This work aims aiding end users of monitoring systems by delivering selected information to each user based on their role in the system. The proposed approach combines Semantic Web (SW) technologies and Complex Event Processing (CEP) for configuration purposes and run-time data processing and analysing. Authors state that final developed solution should be able to provide ways to deal with multiple devices and multiple users of the system; should be reconfigurable to reflect changes in the environment and/or user information needs; and finally, the device updates should be delivered to users within reasonable amount of time. Considering those requirements, the developed approach provides two major advantages. Firstly, the behaviour of the system could be easily changed by configuring only underlying ontology and, secondly, CEP usage at runtime makes system event-driven and reactive to frequent changes in the environment.

(Itria, Daidone & Ceccarelli, 2014) present an approach for critical situation detection that uses CEP architecture for real time event analysis as well as event correlation. While event analysis consists of data processing, event correlation corresponds to establishing a relation between input events that were gathered from various sources, for detecting patterns and situations of interest in the emergency management context. This solution describes the engine, developed in the context of the Secure! Project (Secure, 2016). That solution has two main requirements: the correlation module has to be adaptable to the possible changes of the source environment, and it has to process available historical data in order to evaluate the actual events considering also what has happened before. After submitting the system for testing, authors state that their approach can be easily used and maintained. It is also extensible to other scenarios where the application requires nearly real-time correlation, like intrusion detection system (Ficco & Romano, 2011) and monitoring of critical infrastructures.

Solution for power grid monitoring using CEP is presented by (Cerullo et al., 2014). Authors claim to be able to provide a detailed treatment of the security issues resulting from the adoption of Wireless Sensor Networks and QoS-enabled IP connections. The proposed solution attempts enhancing current information security and event management technology, by improving its capability of detecting and mitigating attacks targeting the heterogeneous network infrastructure. As an example, in power grid scenario, the attacker may prevent some nodes from sending events to the connected collector, thus hiding changes in the power grid conditions. The WSN security probe generates alarms based on the analysis of the network and periodically calculates the package generation rate at every node and the developed engine is capable of correlating those alarms to protect the visualization server. (Wang & Kuang, 2015) propose a traffic prediction method based on Predictive Complex Event Processing (PreCEP) and Bayesian networks to improve the prediction accuracy. The prediction model is trained with historical data and it is used to predict future events based on the recent output of basic CEP engine. Authors address the prediction problem for moving objects that can be vehicles or even pedestrian and conclude that the

performance of PreCEP still needs to be improved. Currently the parallel method only works when learning the structure of models and training models for different context.

Cloud platform monitoring system, JTangCMS - JTang Cloud Monitoring System, is proposed by (Xingjian et al, 2016). Authors claim that proposed solution can deal with the flexibility, scalability, efficiency and performance challenges of cloud monitoring. The system gathers all the data using dissemination framework that allows transferring huge amount of runtime information with high throughput and low latency using DDS - Data Distribution Service which partitions the input into smaller parts. After transferring all the data, it has to be analysed and, therefore, was developed a decision-making support system using CEP, named JTangCMS. Performed evaluations of the implemented algorithm and DDS-based data delivery system state that JTangCMS is efficient solution and does support intelligent decision making. In (Li, Cao & Liu, 2013) authors propose the overall structure and workflow for a CEP-based monitoring system, which can be applied to a private cloud to alert system failures.

(Bruns et al., 2014) proposed ambulance coordination architecture that provides real-time data processing and deliver comprehensive data to the end users. The developed architecture consists of two core components: CEP and FSM- Finite State Machines. CEP focuses on gathering and analysis of the sensor data streams, emitted by the ambulance in order to automatically detect changes of the operational states defined in the FSM. Authors defend that efficient and fast patient care requires reliable and up to date information and, therefore, present an approach that determines the actual state of all ambulances as well as possible relevant mission incidents. Presented solution attempts completing research of (Brotcorne, Laporte & Semet, 2003), (Li et al., 2011), (Ciampolini, Mello & Storari, 2004). (Zappia et al., 2014) also present patient related data and usage of CEP systems in healthcare area. The goal is to use RFID technology to be able to support clinical management of the patients. Proposed solution is focused mainly on the following scenarios: patient identification and tracking, and drug administration. The proposed distributed system is based on event abstraction, event aggregation and event transformation, and uses those to offer a hierarchical and distributed data processing architecture where CEP used data is provided by different processing nodes. Overall, the projected solution, based on CEP and RFID technologies, is used to support clinical risk management by means of detecting possibly dangerous patient conditions as well as patient take care. Solution for critical situation detection in elderly daily life is proposed by (Xu et al., 2014). Authors focus their study on personal assistance as well as possible risks and identification of the required assistance situations. These situations are detected by analysing data received from the sensors. This study is focused on ALL – Ambient Assisted Living technologies and uses sensors to monitor welfare parameters and environmental conditions as described by (Wolf, Schmidt & Klein, 2009).

Another RFID solution is proposed by (Yao, Chu & Li, 2011). Authors propose an RFID framework, using CEP, for managing hospital data, gathered from different sources, and try detecting patterns and medically significant events. Therefore, authors created prototypes that attempt showing that CEP has the ability of providing alerts to the healthcare professionals as well as increasing quality of healthcare and patient safety. One of the main goals consists of identifying the patient and tracking all the necessary procedures since there may be mistakes from part of the hospital staff. Also, the use of the system provides all the necessary historical data for emergency handling as well as for health problems and medication identification and response. All of those characteristics show that this approach may reduce errors as well as provide faster and more efficient risk management.

CEP scaling solution for processing CPS - Cyber-Physical System data (Ollesch, 2016) uses elevator example to describe the challenges of CEP technology in a CPS context. One of the main aspects is the calculation of the floor, since it may be calculated wrongly as well as the sensors themselves are subject to errors and can be disturbed (in the example was considered the use of Kinect sensor) since, for example, passengers would obstruct the Kinect sensor. CEP systems are based on rules and using this example was possible to proof that some parts of the system design should be adjusted. Considering Esper system (Esper, 2016) as example, its rules are embedded inside program code and cannot be externalized. Consequently, each change affects the source code of the CEP application. In particular, (Wang et al., 2014) also study Esper as example and define the event process functions, like event attribution extraction and composition determination. Authors investigate the challenges in designing a CEP method for Cyber-Physical System, and propose a semantic enhancement CPS event model. (László et al., 2016) propose a conceptual architecture for system monitoring that ensures the correct behaviour of the system considering a set of different monitoring rules. It was developed a prototype using the VIATRA-CEP Event Processing Language (Viatra, 2016). Posteriorly monitoring rules are distributed over the nodes. System evaluation was based on development and

monitoring of safety-critical embedded systems in the railway domain.

(Beer & Heindl, 2007) overview the state of art in testing dependable event-based systems and identify the challenges that have to be addressed in the future. Authors consider two case studies during their testing: large-scale project for business-unit transportation systems and a small-scale telecommunications project for AIRBUS. Authors' main focus is on testing and they state that it is an important topic and cost reduction of testing may be achieved by developing new and more efficient techniques of requirements tracing as well as test-case design and validation. Authors based part of their work on (Luckham, 2002) who proposed CEP usage for managing multiple events and event causality in order to elicit meaningful events in an event-based information system. This is because CEP can analyse incoming data and transform it into out coming events. (Lang & Capík, 2014) present a procedure for performing predictive analysis of complex events occurrence in time critical complex event processing systems. Similarly, to (Tendick, Denby & Ju, 2016), authors state that using CEP it is possible to identify and apply business intelligence rules over the streams of events and this technology is critical in an environment where time plays an important role, such as, real time decision making. Authors analyse the possibilities and use of the available methods and techniques for classification and prediction in complex event processing. For this purpose, it was designed and implemented an application CepPredictiveAnalysis, which uses defined methods for predictions and despite certain limitations, the system gives acceptable accuracy in financially-oriented applications.

Over the next section we present our analysis of the described solutions, concerning previously stated quality attributes.

5 REQUIREMENTS ANALYSIS

After briefly describing the considered quality attributes in Section 2, we consider that it is important to confirm whether described systems and solutions of CEP systems in safety-critical applications consider any of those.

Table 1: Quality attributes

	Dependability	Reliability	Security	Safety
(Baldoni, Montanari & Rizzuto, 2015	X		X	
(Zappia, Paganelli & Parlanti, 2012)	X			
(Evchina & Lastra, 2016)	X	X		
(Itria, Daidone & Ceccarelli, 2014)	X			
(Cerullo et al., 2014)	X		X	
(Wang & Kuang, 2015)		X		
(Lu, et al., 2016)	X	X		
(Li, Cao & Liu, 2013)	X	X		
(Bruns et al., 2014)		X		
(Zappia et al., 2014)	X			
(Xu et al., 2014)	X			
(Yao, Chu & Li, 2011)		X		
(Ollesch, 2016)	X			
(Wang et al., 2014)	X			
(László et al., 2016)	X			
(Beer & Heindl, 2007)				X
(Lang & Capík, 2014)		X		

The summary of quality attributes analysis is presented in Table 1. It is important to notice that some of the solutions may have partially addressed some of the quality attributes, but we just consider overall approach and the authors focus and goals in the designed architecture.

After analysing the results, we concluded that most of the systems are focused on their performance and operability and do not consider safety neither security. Those quality attributes are often considered less important than assuring system proper functioning or performance. Most of the authors, as expected, consider the proper functioning of their systems and focus on Dependability and Reliability attributes. Those are more functional characteristics and any designed and developed system is expected to work as intended. Some of the authors consider how their

system is affected by security attributes and if their data is secure and only one study considers risk and hazardous situations management and system awareness of those. It is important to notice that we performed our analysis of quality attributes based on authors' description and designed system specifications.

Though, some of the described works may provide parts of considered by us quality attributes, but those were not mentioned. Also, although we just have one level analysis, as previously stated, we investigated authors' main focus and what their work was trying to achieve.

6 CONCLUSIONS AND FUTURE DIRECTIONS

Safety-critical systems are necessary to perform critical operations if safe environment that will not endanger any life or property. Thus, those systems should be carefully designed and have a set of qualities and characteristics that would attempt ensuring their proper functioning. In this paper we present some of the most important studies that incorporate the use of CEP engines in software-critical systems. We describe performed studies and some of the authors main goals and conclusions. After considering the main system characteristics and quality attributes we attempt investigate which of those are considered in the selected investigations.

We conclude that there are some characteristics, which are highly important, that are not considered by all the authors. Those may have high impact while dealing with critical data as well when processing received information. Both safety and security are focused on providing correct system data by concerning authenticity of the received information, system components safekeeping. Our goal was to analyse some of the available solutions of application of CEP engines in safety-critical environments and describe some of the most common areas where those are used. We believe that there are still some system characteristics that have not been fully covered by existing works and we presented two of those, namely, safety and security.

After concluding our research and while considering the results, we believe that security and safety are equally important quality attributes and have been neglected by some of the available solutions. Therefore, as our future research direction, we will investigate some of the mechanisms that should be considered in safety-critical systems and that would provide necessary system safety. Currently, there are some protection mechanisms that are already used by most of the developed systems, such as, authentication, data encryption, safe connections, hardware authentication and register, etc. As future work we consider investigating methods from which safety-critical systems could benefit and that would not compromise systems performance.

REFERENCES

Aidi, A. 2006. Complex Event Processing. IBM Haifa Labs

Aidi, A., Botzer, D., Nechushtai, D., Sharon, G. 2006. Complex Event Processing for Financial Services. Services Computing Workshops, 2006. SCW '06. IEEE

Ammon, R., Emmersberger, C., Springer, F., Wolff, C. 2009. Event-Driven Business Process Management and its Practical Application Taking the Example of DHL

Atoum, I., Bong, C. 2015. Measuring Software Quality in Use: State-of-the-Art and Research Challenges. Measuring Software Quality in Use: State-of-the-Art and Research Challenges

Baldoni, R., Montanari, L., Rizzuto, M. 2015. On-line failure prediction in safety-critical systems. Future Generation Computer Systems 45 (2015) 123–132

Balogh, L., István, D., István, R., Dániel, V., András, V. 2016. Distributed and Heterogeneous Event-based Monitoring in Smart Cyber-Physical Systems

Barbacci, M., Longstaff, T., Klein, M., Weinstock, C. 1995. Quality Attributes. Software Engineering Institute Carnegie Mellon University. Technical Report CMU/SEI-95-TR-021; ESC-TR-95-021

Beer, A., Heindl., M. 2007. Issues in Testing Dependable Event-Based Systems at a Systems Integration Company. Second International Conference on Availability, Reliability and Security (ARES'07)

Bowen, J., Stavridou, V. 2002. Safety-Critical Systems, Formal Methods and Standards. Software Engineering Journal 8(4)

Brotcorne, L., Laporte, G., Semet, F. 2003. Ambulance location and relocation models. European journal of operational research, vol. 147, no. 3, pp. 451–463

Bruns, R., Dunkel, J., Billhardt, H., Lujak, M., Ossowski, S. 2014. Using Complex Event Processing to Support Data Fusion for Ambulance Coordination. Information Fusion (FUSION), 2014 17th International Conference

Cerullo, G., Formicola, V., Iamiglio, P., Sgaglione, L. 2014. Critical Infrastructure Protection: having SIEM technology cope with network heterogeneity.

Ciampolini, A., Mello, P., Storari, S. 2004. A multi-agent system for medical services synergy and coordination. in International ECAI 2004 workshop on Agents applied in health care, J. Nealon, U. Cortes, J. Fox, and A. Moreno, Eds., p. 3846

Cockburn, D. 2016. Dealing with Quality Defects and Rapid Alerts. Available at: http://www.emAmmona.europa.eu/docs/en_GB/document_library/Presentation/2009/12/WC500017884.pdf

Daum, M., Götz, M., Domaschka, J. 2012. Integrating CEP and BPM: how CEP realizes functional requirements of BPM applications. DEBS '12 Proceedings of the 6th

ACM International Conference on Distributed Event-Based Systems. Pages 157-166

Eckert, M. and Bry, F. 2009. Complex Event Processing. German language in Informatik Spektrum, Springer 2009

Esper. 2016. http://www.espertech.com/products/esper.php

Evchina, Y., Lastra, J. L. 2016. Hybrid approach for selective delivery of information streams in data intensive monitoring systems. Advanced Engineering Informatics 30 (2016) 537–552

Ficco, M., Romano, L. 2011. A generic intrusion detection and diagnose system based on complex event processing. Proceedings on the 1st International Conference on Data Compression, Communication, and Processing, p. 275-284

Foley, J., Churcher, G.E. 2009. Applying Complex Event Processing and Extending Sensor Web Enablement to a Health Care Sensor Network Architecture. Conference: Proceedings of the 4th International Conference on COMmunication System softWAre and MiddlewaRE (COMSWARE 2009)

Knight, J. C. 2002. Safety critical systems: challenges and directions. Proceedings of the 24th International Conference on Software Engineering, ICSE '02. Pages 547-550

Lang, J., Capík, Z. 2014. Prediction based on hybrid method in complex event processing. SAMI 2014. IEEE 12th International Symposium on Applied Machine Intelligence and Informatics, January 23-25, 2014. Herl'any, Slovakia

Nuno Laranjeiro, Seyma Nur Soydemir, Jorge Bernardino: A Survey on Data Quality: Classifying Poor Data. 21st IEEE Pacific Rim International Symposium on Dependable Computing, PRDC 2015, Zhangjiajie, China, pp. 179-188.

Li, L., Cao, B., Liu, Y. 2013. A study on CEP-based system status monitoring in cloud computing systems, in: 2013 6th International Conference on Information Management, Innovation Management and Industrial Engineering, 1, pp. 300–303

Li, X., Zhao, Z., Zhu, X., Wyatt, T. 2011. Covering models and optimization techniques for emergency response facility location and planning: a review. Mathematical Methods of Operations Research, vol. 74, no. 3, pp. 281–310

Lu, X., Yin, J., Xiong, N., Deng, S., He, G., Yu, H. 2016. JTangCMS: An efficient monitoring system for cloud platforms. Information Sciences 370–371 (2016) 402–423

Luckham, D. 2002. The Power of Event, Addison Wesley

Malm, T., Vuori, M., Rauhamäki, J., Vepsäläinen, T., Koskinen, J., Seppälä, J., Virtanen, H., Hietikko, M., Katara, M. 2011. Safety-critical software in machinery applications. VTT, Espoo, Finland

Massimiliano, L. I., Daidone, A., Ceccarelli, A. 2014. A Complex Event Processing Approach for Crisis-Management Systems

Mendes, M., Bizarro, P. and Marques, P. 2009. A Performance Study of Event Processing Systems. Volume 5895 of the series Lecture Notes in Computer Science pp 221-236

Naqishbandi, T., Sheriff, I., Qazi, S. 2015. Big Data, CEP and IoT: Redefining Holistic Healthcare Information Systems and Analytics. International Journal of Engineering Research & Technology (IJERT). Vol. 4 Issue 01

Ollesch, J. 2016. Doctoral Symposium: Adaptive Steering of Cyber-Physical Systems with Atomic Complex Event Processing Services. DEBS '16 June 20 - 24, 2016, Irvine, California, USA

Oracle. 2012. Financial Services Data Management: Big Data Technology in Financial Services. An Oracle White Paper

Secure. 2016. Secure! Project, http://secure.eng.it/

Schmidt, K., Anicic, D., Stuhmer, R. 2012. Event-driven Reactivity: A Survey and Requirements Analysis

Sommerville, I. 2004. Software Engineering (7th Edition). Pearson Addison Wesley ©

Tendick, P., Denby, L., Ju, W. 2016. Statistical methods for complex event processing and real time decision making. WIREs Comput Stat 2016, 8:5–26

Viatra. 2016. VIATRA-CEP Documentation. https://wiki.eclipse.org/VIATRA/CEP

Wang, D., Rundensteiner, E., Wang, H., Ellison, R. 2010. Active complex event processing: applications in real-time health care. Proceedings of the VLDB. Volume 3 Issue 1-2, September 2010, Pages 1545-1548

Wang, Y., Kuang, L. 2015. Traffic prediction method based on complex event processing and adaptive bayesian networks. The 2015 International Academic Research Conference, 3-6 August, University of London

Wang, Y., Zhou, X., Shan, L., Miao, K. 2014. Study on Complex Event Processing for CPS: An Event Model Perspective.

Wolf, P., Schmidt, A., Klein, M. 2009. Applying Semantic Technologies for Context-Aware AAL Services: What we can learn from SOPRANO In: Workshop on Applications of Semantic Technologies 09

Xu, Y., Wolf, P., Stojanovic, N., Happel, H. 2014. Semantic-based Complex Event Processing in the AAL Domain. Mechatronic and Embedded Systems and Applications (MESA), IEEE/ASME 10th International Conference

Yao, W., Chu, C., Li, Z. 2011. Leveraging complex event processing for smart hospitals using RFID. Journal of Network and Computer Applications, 34 (2011) 799–810

Zappia, I., Paganelli, F., Parlanti, D. 2012. lightweight and extensible Complex Event Processing system for sense and respond applications. Expert Systems with Applications 39 (2012) 10408–10419

Zappia, I., Ciofi, L., Paganelli, F., Iadanza, E., Gherardelli, M., Giuli, D. 2014. A distributed approach to Complex Event Processing in RFID-enabled hospitals. Euro Med Telco Conference (EMTC)

Appendix C – Draft paper

To be submitted to International Journal of Critical Infrastructure Protection

CEP and safety requirements in safety-critical systems

First Author[a], Second Author[b], Third Author[a,b,]3

[a]*First affiliation, Address, City and Postcode, Country*
[b]*Second affiliation, Address, City and Postcode, Country*

Abstract

CEP engines have been gaining interest over the past years and are widely used as part of more complex systems. Since these engines are useful for processing data streams, their top-quality attributes consist of providing high performance (throughput) and low latency. Any working system and, especially, a critical system, should be capable of efficiently handling incoming data streams and produce correct outputs. With the growing ubiquity of CEP, there is an emerging uncertainty if these engines should be used as part of safety-critical systems, even if they are one of the best solutions for serving nearly real-time alerts based on complex data analysis. This doubt arises from the fact that in safety-critical systems, not only performance and functional characteristics should be addressed but, also, safety and security, among other quality attributes. These quality attributes can be neglected in many systems but are of the utmost importance in critical environments. In this paper, we investigate how CEP engines, such as Apache Flink, Drools, Esper, Oracle CEP and Sybase Aleri are being used as part of critical systems and determine if they provide the safety mechanisms required by industry standards. Also, we propose a security mechanism for assuring data authenticity and integrity inside one of the most used CEP engines today, ESPER

Keywords: Type your keywords here, separated by semicolons ;

1. Introduction

Data is being constantly produced, and devices used for implementing de “Internet of Things” (IoT), such as sensors, RFID readers, mobile and personal apparels, or even web and mobile apps play an important role in the massive production of data that we are witnessing today. Thus, despite the growing volume of data, it is necessary to continue to efficiently process, filter and extract important information from data on time. CEP engines are designed to process large amounts of streaming data in the lowest possible execution time. This operational design can provide important information in nearly real-time, while data is being constantly streamed, CEP engines process it against previously specified rules and direct any important information to the final user as quickly as possible. The quicker this information reaches the top management and decision makers, the better it is for the business intelligence. Thus,

* Corresponding author. Tel.: +0-000-000-0000 ; fax: +0-000-000-0000 .
E-mail address: author@institute.xxx .

CEP engines have been a popular solution in financial, management and other decision-support areas. For example, in the financial area, trading opportunities must be evaluated in very limited amount of time, reducing the possibility of income loss. Also, CEP filtering capabilities allow to distinguish important data from noise or any non-relevant data. Decision makers should be informed on time and CEP systems can ensure that only important information will be delivered to them. Same requirements apply to any Business Process Management system as well to the sensor-reliant systems. Those systems are designed to continuously receive and process incoming data and provide the possibility of generating a variety of alerts and messages to the end user.

Performance and robustness of the system are always the focus but we believe that, these days, security is a quality attribute of software systems of the utmost importance. Cybercrimes and unauthorized data access are ever important threats that any company should be protected from. While dealing with security and information safety there are different mechanisms and techniques that may be used, depending on the system purpose and design. Safety-critical systems are responsible for performing critical operations with high impact on human lives, as well as on society, and material losses in case of failure. Safety-critical systems should respect and follow industry standards regarding design, implementation, operation and security issues. Since safety-critical systems such as, water storage, petroleum stations, atomic stations, among others, usually must deal with streaming sensor data, we believe that the usage of CEP engines would provide performance and time-saving benefits. However, as previously mentioned, we must still address system security issues since a safety-critical system should not be compromised to the point of endangering anything or anyone.

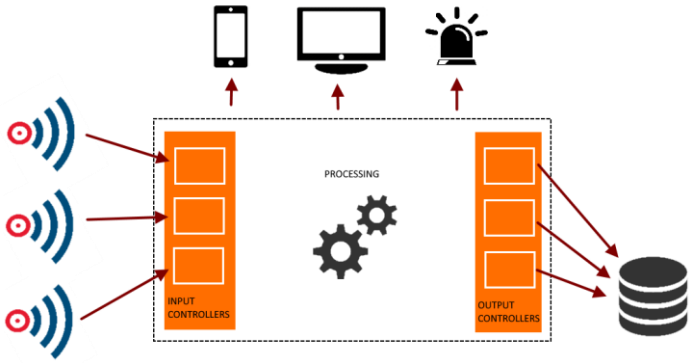


Fig. 1. Architecture of the system that uses CEP engine to receive and process data and posteriorly this data is sent to the rest of the system to be additionally processed and displayed to the user, as well as persisted

Over the past years, many studies have been done on CEP engines, focused on evaluating their processing capabilities and throughput [43-45] (Moazafari et al., 2013), (Mozafi et al., 2012), (Karakostas, 2013). Other even attempted to enhance their results by integrating other mechanisms and, therefore, increasing the amount of processed data while reducing streaming latency [46], [47] (Ray et al., 2013), (Grabs & Lu, 2011). However, we believe that high operational throughput should not be considered the only requirement for CEP systems. So far most of the existent studies do not cover other quality attributes, such as, safety and security, while focusing mainly on performance characteristics. Therefore, our study describes safety and security mechanisms of CEP engines and investigates if nowadays CEP engines can be used in safety-critical systems, where even a small software specification error, design flaw, or the lack of full clarification and specification of the system components can contribute to, or cause, a system failure or the injury, and even death, of people. For example, nuclear power plants are systems that besides processing large amounts of incoming sensor data must provide security status and alerts on time. The system must not report incorrect warnings or fail to raise alarms when necessary. Safety-critical systems must be compliant with both industry and legal standards, such as we will describe later, which are responsible for optimizing system safety in the design, development, use and maintenance, as well as specify their integration with hardware components in the operational environment. Some of the standards are general and could be applied to any system but, depending on the area, there are also specific requirements.

In this paper, we focus on system security and provide security mechanisms that should be used in safety-critical systems, based on some of the available standards. Since our main goal is to evaluate if CEP engines are suitable to be used in a safety-critical system, we study five of most used CEP engines: Apache Flink; Drools; Esper; Oracle CEP, and Sybase Aleri and evaluate if these systems provide any security or just focus on system performance. Finally, we propose a security mechanism for validation of data authenticity and integrity, as part of the ESPER CEP engine, thus moving security concerns inside ESPER. Having security by design within CEP engines is essential to make them compliant with SCS standards without relying on third-party dependencies. We evaluate the impact of our mechanism in the performance of the ESPER engine and compare it with other alternatives.

The remainder of paper is structured as follows: Section 2 describes the state of the art and CEP engines characteristics. Section 3 describes the proposed security mechanism while Section 4 presents the results of the execution tests. Finally, Section 5 are the conclusions and future work.

2. State of the art

Due to the popularity of CEP engines and their broad use in different areas, over the past years, several studies have been done to evaluate their quality attributes [49-51]. The most commonly evaluated attribute has been performance, in detail, the quantity of data that an engine can process per amount of time or latency (necessary time to receive data, process stream and send the response). Oracle released a white paper with the performance evaluation of their CEP engine [35]. In that study, a single instance setup was used and the processing latency measured, while the system was constantly under data injection. Both number of connections and load size where being scaled to provide a better understanding of the latency of the system. The authors concluded that the engine is capable of efficiently dealing with up to 1 million events per second. In other evaluation, the authors measured Oracle's CEP availability and perform latency to provide an overview of the system operability. While discussing the system's performance vs availability, it was concluded that it is possible to sacrifice one of those qualities while working in cluster environment [30].

CEP engines availability has also been addressed and guidelines for the development of a highly available CEP engines have been proposed in [40]. After describing CEP design considerations are presented considerations for development of an engine, such as development language, component isolation, data management, testing and debugging, etc.

A performance evaluation of three CEP engines: Microsoft StreamInsight, Esper and Drools is described in other study. Besides latency, researchers considered load tests and measured the consumed memory, CPU and latency/throughput [36]. Was concluded that all those engines have very similar latency while system resources consumptions are different. CPU usage was similar for two of the systems while another one showed much higher processor usage. In contrast, the system that consumes more CPU requires much less RAM than other two engines. Similarly, in [37], the authors performed tests using Esper and other two enterprise CEP engines (that were not disclosed). The authors executed different types of queries, starting from simple operations, such as, selections and aggregations and up to join operations and multiple queries. It was possible to observe that CEP engines can achieve high throughput while performing basic operations while more complex operations require data to be mapped to memory to provide higher throughput. In this study, while working with simple operations the bottleneck was not in the CEP system itself, rather the client API, that connected to the engine, was increasing the processing time. Also, it was observed that the window expiration mode (jumping/sliding window) had a significant performance impact. Benchmarking results, which show the latencies of a commercial CEP product, were presented in another study. The CEP engine was the WebLogic Event Server, an application server designed specifically for event processing applications that require high throughput and reduced latency, while handling large volumes of events [38]. Finally, performance tests, using Esper and StreamCruncher, were executed by other authors, who divided their evaluation in two parts: latency and throughput. It was concluded that both engines have their advantages and performance flaws but Esper is the most mature engine [39].

While working with safety-critical systems, was discussed that besides development design, documentation, testing and review that are necessary phases during the development of a critical system. System design must be carefully developed to guarantee that after the development the system will have the expected operability and that all of its components are correctly integrated. The authors state that testing may be challenging since to have a good average result it is always necessary to perform a lot of repeated executions. Thus, system trustworthiness can also be assured using rigorous mathematical techniques in the review process [41, 42]. Similarly, in [43], are presented different techniques that should be used while developing a safety critical system. In the authors' paper are described formal

and informal analysis techniques, such as, Fault Hazard Assessment (FHA), Fault Tree Analysis (FTA), Failure Mode Effect Analysis (FMEA) and Deductive cause-consequence analysis (DCCA). Different techniques may be used to ensure system safety and, although DCCA has high success rate out of all safety analysis techniques, it has a major drawback which is the inability to consider unintended behaviour and, therefore, fails in providing fault tolerance. While considering the safety of safety-critical systems, researchers identified several challenges, such as, encryption and authentication mechanisms, malicious and unsafe commands, lack of knowledge of the system vulnerabilities, etc. To address those challenges are presented security principles that could be integrated into safety-critical systems. Authors state that if systems do not have security mechanisms even small error occurrences may end up compromising the entire system. By using the proposed guidelines, the system will be capable to recover from errors without having its performance affected [55].

The previous works either focus on CEP engines or on safety-critical systems, addressing either processing capabilities or security in critical systems. It is important to understand that safety-critical systems also rely on performance and operability, to be able to detect hazardous events as fast as possible. Taking that into consideration, we believe that safety-critical systems could benefit from using CEP engines. But, although complex-event processing systems guarantee performance, there has not been enough efforts in assuring the security side of these systems. Therefore, we study some of the security mechanisms that should be used in critical environments and investigate if those are provided by the most popular CEP engines.

Since there are many available standards, our aim is to identify standards that apply to critical systems, such as powerplant management, and provide not only basic guidelines but specific mechanisms that should take part of any critical system. For our study were chosen the following standards: NIST SP 800-53, SOC2 and IEEE 603-2009. NIST SP 800-53 standard is a standard for Security and Privacy Controls for Federal Information Systems and Organizations and it is based on other two standards: ISO/IEC 15408 and ISO/IEC 27001 [1]. NIST SP 800-53 is one of obligatory standards for industrial systems as well as gasoline pipelines, water storage dams and other national security systems [2, 3]. IEEE 603-2009 is a specific standard for Safety Systems for Nuclear Power Generating Stations. It establishes some of the operational criteria and minimum functional design principles for the power, instrumentation, and control portions of nuclear power generating stations [4]. On the other hand, SOC2 focuses on a business's non-financial reporting controls as they relate to security, availability, processing integrity, confidentiality, and privacy of a system. This standard is mostly used for cloud services and datacentres as a standard for control and assurance of the confidentiality and privacy of the information that is stored or processed [5].

Standards, described above, allowed us to identify specific security mechanisms that are necessary in any safety-critical system. Moreover, we believe that at least some of those mechanisms should be integrated in the CEP engines. Since there are many available CEP systems, we chose some of the most popular, more referenced, that are large projects and provide enough official documentation. Considered CEP engines are, as follows: Apache Flink; Drools; Esper; Oracle CEP, and Sybase Aleri. All those engines are available in both standard and enterprise versions. There is one specific characteristic that distinguish CEP engines, their operational language. There are two language specifications: stream-oriented (transforming) languages and rule-oriented (detecting) languages [6,7]. The stream-oriented languages provide operations for processing the input streams such as filtering, joining or aggregating to obtain some other output streams. The rule-oriented languages use rules for processing streams, clearly separating the triggering conditions and the actions to be taken when the conditions are met. The main component of the stream-oriented systems is the sliding window. There are three types of sliding widows:

time based windows: that captures the last segment of an ordered stream for a given time interval. For example, returning the relation between current item and items from some (parameter) seconds ago;

tuple based windows: which captures the last number (parameter) elements of a stream;

partitioned windows: which operates similarly to a tuple based window but has a condition over one of the data record attributes.

The received stream is partitioned into sub streams and only specific number of the records is processed per stream and their processing results are combined into the final output. Rule-oriented languages use Event-Condition-Action (ECA) rules as a formalism for defining the actions that should be executed when specific conditions are met [16]. That means that a specific action is performed if the condition is satisfied. Therefore, a rule is evaluated only when triggered by a specific event, which can be a primitive event (database operations, temporal events or external notifications) or a combination of primitive events using logic operators [8].

Table 1 presents the summary of security mechanisms, described in safety-critical standards, and their availability in the studied CEP engines (presented in alphabetic order and nothing else). Since all the considered systems are large projects, there is substantial amount of available information that is constantly being updated. Therefore, table data is

based on the white papers, documentation provided by the companies, email lists and forums. As stated above, were chosen the security mechanisms that we considered the most important (basic), required in any safety-critical system and were specified in standards: Access control – defines different levels of system access and operations usage; Backups – data persistence; Encryption – may be encryption of the data itself, parts of the data or even communication channels; Fail-safe procedures: represent the system recovery capabilities and failure management; Reporting information security events – security monitoring and reports; User identification and authentication – user management and authentication for system access; Wireless communication policy – methods that focus on communications outside of the system.

Table 1. CEP engines and important security mechanisms.

	Apache Flink	Drools	Esper	Oracle CEP	Sybase Aleri
Access control	NO	YES	NO	YES	YES
Backups	YES[5]	NO	NO	YES[4]	YES
Encryption	NO[1]	NO	NO	YES[1]	YES
Fail-safe procedures	YES	NO[3]	NO	YES	YES
Reporting information security events	NO	NO	NO	YES	NO
System status identification	YES[2]	NO	NO	YES[2]	YES
User identification and authentication	YES	YES	YES	YES	YES
Wireless communication policy	NO[1]	NO	YES	YES	YES

1 Encrypted connections and data streams between nodes will be available soon [9]. We consider that communication requirement will also be covered by those mechanisms.

2 Requires usage of web client to monitor the system.

3 There is request for the implementation of a possibility to persist the state of the used by the system memory after each state change and related to that the possibility to system recovery and posterior state restore in case of a crash [10].

4 Uses specific persistent event store

5 Apache Kaffka is a messaging system that could be used with Flink and persists incoming data in logs. Output data, after being processed by Flink, can be stored in any database (ex. Cassandra, MongoDB, HBase, MySQL, etc.) [15].

2.1. Apache Flink

Apache Flink is the engine that provides standard authentication mechanisms but there is no possibility for detailed data management access. Ideally, a Flink application may use a variety of different connectors (Kafka, HDFS, Cassandra, Flume, Kinesis etc.) by means of arbitrary authentication methods (Kerberos, SSL/TLS, username/password, etc.). Those connectors are responsible for managing the security requirements (identification and authentication). While satisfying the security requirements for all the connectors, Flink provides first-class support for Kerberos authentication only. Also, for example, it is possible to enable Hadoop security without providing security authentication for the ZooKeeper, or vice versa. Thus, each component (connection) may use separated authentication [11]. While working in a cluster, it is necessary to setup public key authentication on the master node as the user who will later execute all the Flink operations. User credentials that are used by the master node should be also existent on each other node, for master to be able to manage available worker nodes [12]. Encryption feature will be soon developed and will be available in the future versions. Apache Flink offers reliable execution with strict exactly-once-processing consistency guarantees and deals with failures via checkpointing and partial re-execution. The checkpointing mechanism constantly (at regular intervals) creates consistent snapshots of the state of operators, including the current position of the input streams and if it is necessary (in case of system failure) can replay parts of

the job or the entire job. Job execution is managed by the JobManager and it is responsible for coordinating the distributed execution of the dataflow. It tracks the state and progress of each operator and stream, schedules new operators, and coordinates checkpoints for recovery. In a high-availability setup, the JobManager persists a minimal set of metadata at each checkpoint to a fault-tolerant storage and it is possible reconstruct the checkpoint and recover the dataflow execution [13, 14]. However, there are no event mechanisms or logs regarding system security status. For overall system status overview, may be used web console [16]. This user interface provides summary of the executed jobs and processed results.

2.2. Drools

Similarly, Drools is a CEP engine developed as a module responsible for adding event processing capabilities into the platform [17]. This engine provides both user authentication as well as access control. Under the directories of the system it is possible to find the login-config.xml file which contains all the necessary system configurations as well as the user-related information (users, passwords and roles). This file should only be maintainable for a fixed and small number of users [18,19]. To manage considerable quantity of users and their roles more efficiently Drools may be integrated with Guvnor [22]. This application allows editing permissions for different stored rules [23]. This per-rule permissions allow restricting users that may manage specific system resources and rules, per user's role. Roles represent diverse user groups, providing different permissions as well [20]. Drools, as most of the CEP engines, does not provide own data persistence. Since this engine is written in Java, may be used Java Persistence API (JPA) as described by [21]. Since this engine does not physically store any memory-mapped data, there is no data encryption or backups. Fail safe procedures are pendent for implementation and will be available in the future versions. System status identification and security monitoring are also not considered by Drools. There are no reports or logs performed regarding system status.

2.3. ESPER

Esper is probably one of the most known and used CEP engines. However, it is important to notice that, as many others, the developers of this system focus mainly on its performance and throughput. Out of all considered security mechanisms, ESPER still provides user authentication as stated in [24]. Similarly, to Drools, user authentication data is written in xml file and it describes the authentication used for the connections. Since this authentication is used for low level management there is no user access control. Since ESPER does not perform any I/O operations, no data persistence or failure recovery are available [25]. There is no security metrics or system constant monitoring. However, with enterprise version is accessible Rich Multi-Window GUI that may be used as editor, debugger and metrics reporting tool for existent rules, job monitoring and historical job data (with tables, charts and gauges) [26].

2.4. Oracle CEP

Oracle CEP also provides basic security and authentication. Although the authentication is per user (username + password), there are standard user roles/groups that are available: Operator, Monitor, ApplicationAdmin, Deployer, BusinessUser and Admin. This engine supports various security providers for authentication, authorization, role and credential mapping. Oracle CEP is configured to use the file-based providers for both authentication and authorization but there is always a possibility to configure the system to use an LDAP or DBMS as authentication info provider [27, 28]. For data persistence, the engine offers persistence of the output events when, for example, those are written in the database. By default, Oracle CEP stores recorded events in a database that may be posteriorly queried by the application (by specifying the connection and credentials). Default database used by this engine is single Berkeley DB instance, bundled with the Oracle CEP server. However, there is always the option to change storage system. For example, it can be a Relational Database Management System such as Oracle Database or Derby [29]. As we have described above, in the future versions Oracle CEP will support encrypted connections and data streams between cluster nodes [9]. Fault tolerance is achieved by high availability of the system in active-active architecture. The active-active approach requires for a cluster to have at least two nodes that are up at all time and, therefore, this approach can shorten the failover time. Oracle CEP will choose one server in the cluster to be the primary node and the remaining nodes are considered secondary. High availability allows users to make the most appropriate performance vs. quality-of-service tradeoff for their environment, including a precise recovery option that guarantees that no events are lost [30]. System monitoring and management is achieved using web console, Oracle CEP Visualizer

[31]. This web user interface allows not only manage running jobs and rules, but also perform any administrator tasks, such as, view the structure of a system domain, manage security, configure server instances, etc.

2.5. Sybase Aleri

Starting with user management (access controls and authentication), Aleri can use the Pluggable Authentication Module (PAM) package found in Linux and Solaris. In PAM authentication, client applications should connect to the core engine either using command interface, gateway interface or SQL query interface and must supply a user name and password before they can execute any commands. In this case, it is the responsibility of the system administrator to specify the type of the authentication that will be performed by the users [31]. It is important to notice that it is possible to restrict user access even if the authentication is successful. This type of operation complements basic access control and the streaming processor server may be configured to deny the authenticated user the ability to query certain streams through the SQL interface or subscribe to streams through the subscription interface, or to stop the server [31]. Data backup is an integral part of a data management and protection strategy. It is recommended to backup Sybase Aleri streamed data regularly. Should be backed up two types of data: XML files that define the data streams and how they interact (Data Models) as well as generated operational log. If both of those are backed up, it is possible to restore the system after any failure. In terms of backup creation, there are two possibilities: “off-line” backup (back up the data model and log files while the streaming platform is down) or in an “on-line” backup mode (the data models and log stores are generated while the platform is running). However, it is important to notice that although in “on-line” mode the system will be running, its current operation will be suspended while the backup files are created [31]. Besides backups and restores, system failure management is performed using system availability. While working in a cluster, if the primary server fails, the high availability configuration detects the failure and promotes the secondary server to primary status with minimal interruption to client requests. Overall system monitoring may be performed either using commands or by accessing Aleri Studio Monitor [32]. Finally, Aleri provides encryption, using AES, which protects the data sent over a network (transferred between clients and the streaming platform), using secure sockets – SSL [33, 34].

In summary, we state that although all the studied CEP engines provide basic security mechanism (authentication) there are many other mechanisms that are not considered. Out of all the engines ESPER is the least prepared system to provide data security while Oracle CEP offers many approaches to secure not only the data but the system itself. That means that if ESPER is integrated in a critical system it will be necessary for the other system components to provide security and failover mechanisms.

3. Proposed security mechanism

When safety-critical systems have geographically distributed components that communicate using WSN (Wireless Sensor Network), one of the most important requirements is data authenticity. It can be achieved by using secure communication with transferred data encryption, such as SSL. But, while working with sensors with limited processing capabilities, this is not the most appropriate and lightweight. Moreover, we believe that the CEP engine itself should be able of evaluating the streamed data before processing it. This approach would remove additional mechanisms, outside of CEP, and reduce the data management process. Therefore, in this section, we propose our own security solution. It consists of adding a security mechanism that ensures data authenticity and integrity, to one of the most popular CEP engines, ESPER.

There are different system architectures and it is important to understand their security requirements. For example, it is necessary to provide secure data transferring in a system where all its components communicate using the Internet while in private network, this type of security and authenticity validation may not be required. As case study, we choose sensor networks and security systems that use sensors for collecting the required data. Before being able to create a solution that could ensure data security (authenticity and integrity), it is important to understand not only the challenges of a WSN but also the different types of the sensors that exist. That overview allows us to understand how hardware affects the security mechanisms since by using low power components it is possible to reduce the overall system cost. For example, MSP-EXP430G2 controller can be bought for around 10 dollars and ESP8266, designed by Espressif Systems, for around 6 dollars. More powerful controllers provide more processing capabilities but their cost is higher. For example, RaspberryPi costs around 30 dollars while Intel Galileo board costs around 80 dollars. It is important to mention that Arduino components can be acquired separately and it is possible to design own board

starting from scratch but that would require additional time and production costs (effort to combine different components and guarantee that the final product is operating as expected).

Currently most of the systems communicate using SSL/TLS secure connection, that uses certificates to provide encrypted channel between one point and other after initial handshake. This approach in widely used in TCP network communication and has proven secure enough. But, while working with sensors with low processing and memory capabilities, this protocol may not be used because the sensor does not have enough memory to be able to map the certificate to perform the handshake and authenticate itself. Therefore, it is necessary other approach, with lower hardware requirements. One popular method is random key predistribution [52, 53]. As we described in previous sections, this method distributes random public keys or cluster keys between nodes that would allow them to communicate with each other. Keys are generated randomly and are distributed pair wise between nodes. Server stores the matrix which is used for generation of the keys as well as for the identification of nodes that share the same key. This approach also requires periodically key renewal that would ensure system's security even if an attacker can get one of the existent keys. In this case, server should occasionally distribute the keys and there is always, no matter how small, the possibility of this information being caught. One of the possibilities to eliminate this part of communication is the usage of pre-distributed configuration. This means that all the sensors are manually configured, when system is deployed, with public and private keys and the server can identify each one of them. One of the main flaws of this approach is that the system is update or the keys renewal would require some trusted third-party entity to perform the deploy and manually configure all the system components. When the system is highly geographically distributed, this becomes even more of an exhausting task but, on the other hand, no secret data is transferred at any point.

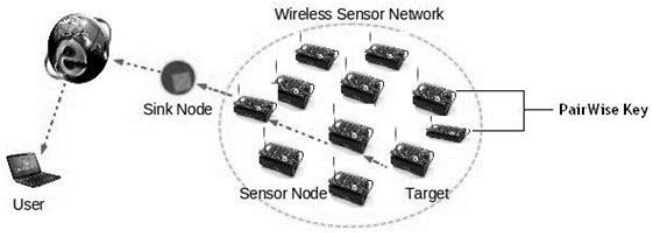


Fig. 2. Example of network that uses random key predistribution mechanism, proposed by [54]. All the nodes are connected to the same network and share pairwise key. Then, there is a mater node, server, which communicates outside of the network.

As we have stated above, our goal was to add a security mechanism to ESPER that would ensure data authenticity. Since ESPER is available in java and .Net programming languages, we chose .Net source code. The basic idea of ESPER is to "subscribe" to the specific data conditions/events that appear in the collected streams and it is the part of the engine where we added data validation. After receiving the data, the engine must be capable to determine if this data is trustworthy and if it is, it will be processed, otherwise it is discarded and logged. As mechanisms we used AES encryption, with 128 bits per key, using symmetric key and SH1 hashing. In Figure 3, we present the implemented approach.

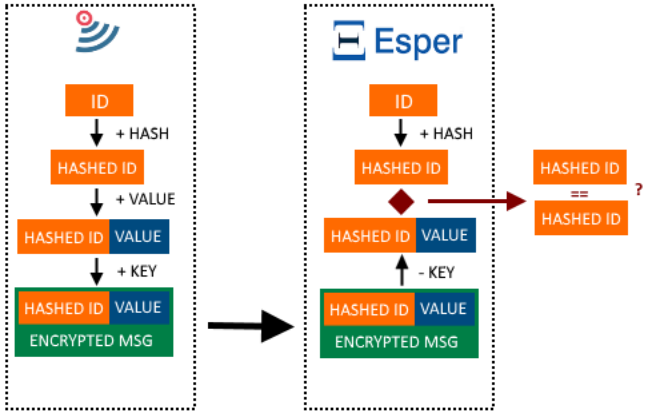


Fig. 3. Proposed solution for validation of data authenticity. Operational flow is represented using arrows. Orange boxes represent string values, such as, IDs. Blue ones represent numerical long value which is the sensor value. Green is the final message that is transferred from the sensor to the ESPER and it consists of hashed sensor ID and value. Decision component (dark red) compares the IDs and validates the sensor authenticity.

Each system component is configured only with its personal ID and key that is used to encrypt transferred data. Server, on the other hand, besides the key has the knowledge of the expected communication devices (their IDs). When the sensor wants to communicate the current value, it must hash its ID and create a message by combining the hashed key and value. After the message string has been created, it is encrypted using the key and the sent to the engine. Then, when the system receives the message, it must be decrypted, using symmetric key, so it is possible to validate the received message. After decryption, the message is constructed by device ID and the value. Since this device ID is hashed and it is not possible to remove the hash, ESPER must hash known clients' IDs and compare if any of the strings match. If the ID is valid, the engine accepts the received value otherwise it is discarded and this event is logged.

4. Experimental results

To evaluate the performance impact of proposed solution were executed different tests, while measuring the processing time and throughput. Machine processing characteristics used to evaluate proposed solutions are, as follows:

- Intel Core i5-6600K @ 3.5GHz
- 16GB RAM
- 4 cores / 4 logical processors
- Kingston Savage 250GB SSD

To better understand the performance impact, we varied the number of processed records as well as the record size. First, we considered 2KB per value (meaning that the total message had 2KB + 3 bytes for the header) and then increased it to 5KB. Also, were performed 1,000,000, 10,000,000 and 100,000,000 total operations (messages sent to ESPER). The time measurement was done on the server side, reducing the possible overhead of the client. That means that sent messages are previously prepared and the engine is constantly receiving the incoming data. As additional comparison, we implemented communication and data transfer using SSL, with authentication and certificates on server and client sides. All the measurements are performed on the server side, starting after receiving the first record and finished when expected number of operations has been performed. Also, the initial handshake time was also excluded.

It is important to notice that all the tests were executed in the same environment, always using all the 4 available processor's cores. This means that there were no environment variations regardless of the test case (no encryption / with encryption / SSL). Figure 4 presents the obtained execution time while using 2KB per record value.

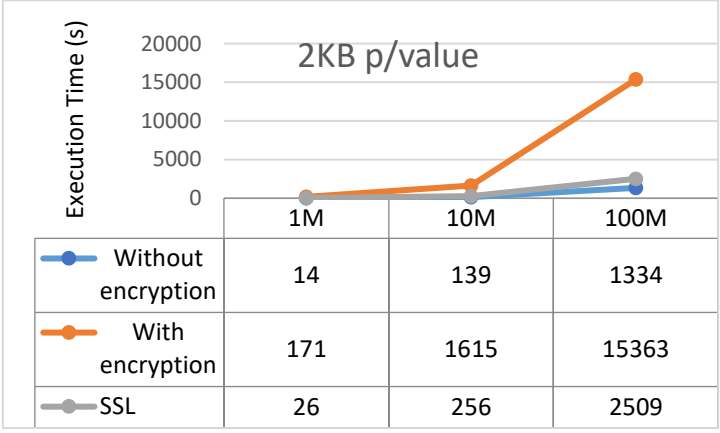
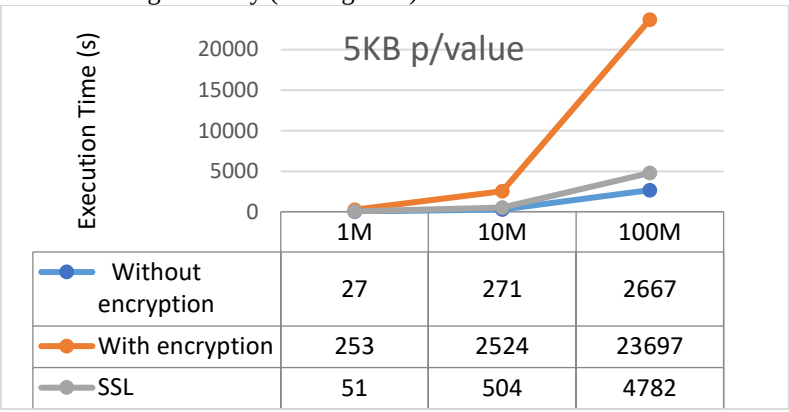


Fig. 4. Execution time, in seconds, with sensor value of 2KB. Each point represents different number of records, starting with 1,000,000, up to 10,000,000 and 100,000,000. Presented numbers are the average of 10 executions.

While comparing the results, we concluded that, as expected, lowest execution time is obtained while working without any security mechanisms. Secondly, 1.8 times slower, is the execution time using SSL. But, using AES encryption and SH1 hashing to guarantee data authenticity, data processing demonstrated to be between 11 and 12 times slower when compared to the performance without encryption and between 6 and 6.5 while comparing with SSL. Then, we increased the value size to 5KB (5KB + 3 bytes for header per message), more than twice the size of the previous value and size that is bigger than one page of text. As we expected, the overall execution time, regardless of the security layer, has increased significantly (see Figure 5).



	1M	10M	100M
Without encryption	27	271	2667
With encryption	253	2524	23697
SSL	51	504	4782

Fig. 5. Execution time with 5KB per sensor value. Were performed 1,000,000, 10,000,000 and 100,000,000 operations. Each value is average from 10 executions.

Similarly, to the previous results, SLL has average performance impact (in average, requires twice the time to process received data) while usage of encryption and hashing results in more than 8 times higher processing time. The obtained results show the overhead that is added by using data encryption and hashing. As expected, SLL does increases the processing time but still has much better performance than encryption. As previously stated, all the measurements were performed on the ESPER side and the initial handshake between client and server was not considered. However, if the handshake would be performed periodically (to renew the connection) it could possibly affect the continuous operational results. To better understand the processing capabilities, we compared the obtained throughput (see Figure 6).

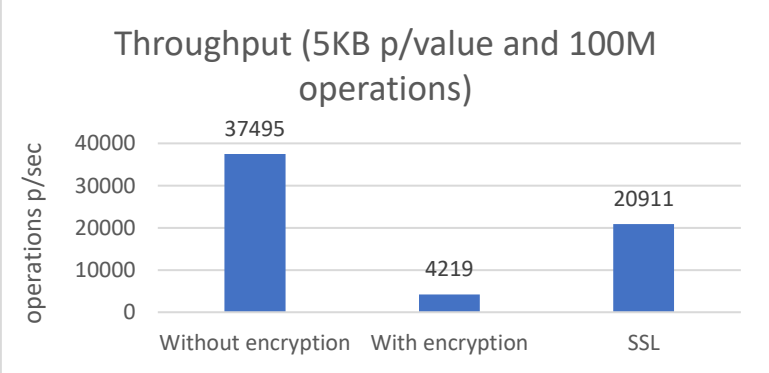


Fig. 6. Throughput obtained during the execution of tests using 5KB of data per value and 100 million of operations.

While analyzing the throughput, we observed that the difference in the number of performed operations per second with and without any security mechanisms is noticeable. Due to the processing capabilities, it was possible to achieve a processing speed of over 37K of messages per second. Although low power sensors are not capable of achieving this throughput, the comparison allowed us to understand how the performance is affected by using encryption. When using hashing and AES protocol the throughput reduced more than eight times.

Overall, we observed that the engine is capable of successfully validate data authenticity and process only the data from trustworthy source. Moreover, the proposed solution does not require any certificates, reducing the hardware requirements of the sensors. While comparing the obtained results, it is possible to conclude that added security mechanisms for validation of data integrity and authenticity creates big performance overhead, especially while comparing with the security that is already provided by SSL. In some tests, the system required up to ten times more time to process all the messages. The main restriction not to use SSL would be the type of the system since while working with components that have more capabilities, SSL would be a better approach that would not affect the performance as much. As expected, without any security ESPER can provide high throughput and processes the incoming messages efficiently. But that means that it would be necessary to add other security mechanisms as a part of the system but outside of the CEP engine. That may be unreasonable since it would be more efficient to validate the data before processing it. Thus, reducing the amount of data to be processed by the CEP (not trusted data is discarded) and overall reducing the processed data by the rest of the system (processed data is filtered against the CEP rules).

5. Conclusions and Future work

In this paper, we described some of the most popular CEP engines and addressed the problems in their security mechanisms. We can conclude that one of the basic mechanisms, authentication, is present in every of the studied engines. However, it is not nearly enough security for deployed safety-critical system. Mostly, CEP engines are expected to provide only data processing, focusing mainly on the operational performance, while the remaining of the system should provide all the required data security. That would be acceptable if there would be no processing overhead, resulted after processing the unnecessary data. For example, during data stream reading from the outside of the system, if the CEP engine is not capable of hardware identification as well as connection encryption, adding those mechanisms outside of CEP engine would require additional queue processing of events, resulting in at least 1.5x processing time.

Most of the studied systems do not provide required security mechanisms that would be strictly necessary in any safety-critical system. Besides understandable communication requirements, some of the engines don't even provide overview of the system (CEP) itself nor security events reporting. Even while not considering third-party access, it is necessary to understand current system status. Therefore, we concluded that the CEP engine that could potentially be used as a part of safety-critical system is Oracle CEP. This engine provides some of the basic safety and security mechanisms and does not creates an additional load on the rest of the system's architecture. This engine is followed by Sybase Aleri, Apache Flink and Drools, ordered by the quantity of provided security. Finally, ESPER is the CEP engine that focuses mainly on the processing and achieving best throughput. This engine could be an alternative due to its performance, but for it to be used in safety-critical system, it would be necessary adding security mechanisms outside of the CEP and that could compromise the overall system reliability.

The proposed solution, using data encryption and hashing for guaranteeing data integrity, can solve part of the security issues of a CEP engine while as a part of a more complex distributed system. One of our wain constrains were the processing and memory characteristics of the system sensors. To better understand how system performance is affected by usage of encryption, we compared the execution time of 1,000,000, 10,000,000 and 100,000,000 operations with encryption, without and with SSL. While comparing the results, we concluded that data hashing and encryption create huge system overhead and reduce overall system performance. On this case SSL, would be more efficient approach but only when used in environments where sensors can store certificates, perform handshake and use the encrypted channel for communication. As future work, we will evaluate the performance of ESPER while using other encryption algorithms to see how the encryption key size affects the throughput.

References

[1] NIST. Security and Privacy Controls for Federal Information Systems and Organizations. 2013. NIST Special Publication 800-53, Revision 4.

[2] NIST. Guide for Applying the Risk Management Framework to Federal Information Systems. 2010. Computer Security Division Information Technology Laboratory National Institute of Standards and Technology Gaithersburg, MD 20899-8930.

[3] Stouffer, K. and Katzke, S. Industrial Control System Security and NIST SP 800-53 Overview. 2008. National Institute of Standards and Technology.

[4] 603-2009 - IEEE Standard Criteria for Safety Systems for Nuclear Power Generating Stations. 2009. Revision of IEEE Std 603-1998.

[5] Microsoft. https://www.microsoft.com/en-us/trustcenter/Compliance/SOC.

[6] Vincent, P. http://www.complexevents.com/2016/05/12/cep-tooling-market-survey-2016/.

[7] https://tedwon.atlassian.net/wiki/display/SE/CEP#CEP-CEPProducts.

[8] Moraru, A. Methods for Complex Event Processing. 2012. Doctoral degree.

[9] http://apache-flink-user-mailing-list-archive.2336050.n4.nabble.com/Security-in-Flink-td4202.html.

[10] https://issues.jboss.org/browse/DROOLS-1221.

[11] Flink. https://ci.apache.org/projects/flink/flink-docs-release-1.2/ops/security-kerberos.html.

[12] Flink. https://ci.apache.org/projects/flink/flink-docs-release-0.8/cluster_setup.html.

[13] Carbone, P., Ewen, S., Haridi, S., Katsifodimos, A., Mark, V., Tzoumas, K. Apache Flink™: Stream and Batch Processing in a Single Engine. 2015. Bulletin of the IEEE Computer Society Technical Committee on Data Engineering.

[14] P. Carbone, G. Fora, S. Ewen, S. Haridi, and K. Tzoumas. Lightweight asynchronous snapshots for distributed ´ dataflows. arXiv:1506.08603, 2015.

[15] Ewen, S. The Stream Processor as a Database. 2016. Hadoop Summit.

[16] Flink. https://ci.apache.org/projects/flink/flink-docs-release-1.0/quickstart/run_example_quickstart.html.

[17] JBoss. https://docs.jboss.org/drools/release/6.2.0.CR3/drools-docs/html/DroolsComplexEventProcessingChapter.html.

[18] Mollenkopf, A., Tirelli, E. Applying Drools Fusion Complex Event Processing (CEP) for Real-Time Intelligence. 2009. JBoss World Chicago 2009.

[19] JBoss. https://docs.jboss.org/drools/release/5.5.0.Final/drools-guvnor-docs/html_single/#d0e42

[20] JBoss. https://github.com/bhochhi/drools-guide/wiki/how-to-integrate-LDAP-with-drools-workbench-for-authentication-and-authorization%3F

[21] Žáková, I. Drools Fusion and Utilization of Complex Event Processing in Web Applications. 2013. Master's Thesis.

[22] Drools. http://www.drools.org/.

[23] JBoss. https://www.jboss.org/dms/judcon/2012india/presentations/day1track3session2.pdf

[24] EsperTech. http://www.espertech.com/esper/release-5.2.0/esper-reference/html/configuration.html.

[25] EsperTech. http://www.espertech.com/esper/faq_esper.php

[26] EsperTech. http://www.espertech.com/products/esperhq.php

[27] Oracle. https://docs.oracle.com/cd/E13213_01/wlevs/docs30/config_server/security.html

[28] Oracle. https://docs.oracle.com/cd/E13157_01/wlevs/docs30/config_server/security.html

[29] Oracle. https://docs.oracle.com/cd/E12839_01/doc.1111/e14302/server_tasks.html

[30] Oracle. Oracle Complex Event Processing High Availability. 2010. An Oracle White Paper.

[31] Sybase. Administrators Guide, Sybase Aleri Streaming Platform 3.1. 2010. Sybase, Inc.

[32] Gadgil, M. Aleri – Complex Event Processing. 2012. https://www.javacodegeeks.com/2012/04/aleri-complex-event-processing.html

[33] Sybase. Frequently Asked Questions (FAQ), Sybase Aleri Streaming Platform 3.1. 2010. http://infocenter.sybase.com/help/topic/com.sybase.infocenter.dc01297.0311/pdf/FAQ.pdf%3Fnoframes%3Dtrue

[34] Sybase. Product Overview, Sybase Aleri Streaming Platform 3.1. 2010. http://infocenter.sybase.com/help/topic/com.sybase.infocenter.dc01286.0311/pdf/ProductOverview.pdf?noframes=true

[35] Oracle. Oracle Complex Event Processing Performance. 2008. An Oracle White Paper.

[36] Wahl, A. and Hollunder, B. Performance Measurement for CEP Systems. 2012. SERVICE COMPUTATION 2012: The Fourth International Conferences on Advanced Service Computing.

[37] Mendes, M., Bizarro, P., Marques, P. A Performance Study of Event Processing Systems. 2009. Performance Evaluation and Benchmarking. Volume 5895 of the series Lecture Notes in Computer Science pp 221-236.

[38] White, S., Alves, A., Rorke, D. WebLogic event server: a lightweight, modular application server for event processing. In Proc. of DEBS 2008.

[39] Dekkers, P. Master Thesis Computer Science. Complex Event Processing. Radboud University Nijmegen, Thesis number 574, October 2007.

[40] Saboor, M., Rengasamy, R. DESIGNING AND DEVELOPING COMPLEX EVENT PROCESSING APPLICATIONS. 2013. Sapient corporation.

[41] Parnas, D.L., Schouwen, A.P., Kwan, S.P. Evaluation of SafetyCritical Software. 1990. Communications of the ACM Volume 33 Issue 6, June 1990, Pages 636-648.

[42] Collins, I. Chapter 1: Safety-Critical Computer System Design and Evaluation. 2013. Howard University Department of Electrical and Computer Engineering.

[43] Aftab, A.H, Nadeem, A. A Survey of Safety Analysis Techniques for Safety Critical Systems. 2013. IJFCC 2013 Vol.2(2): 134-137 ISSN: 2010-3751.