Marie Skłodowska-Curie Actions (MSCA)

Research and Innovation Staff Exchange (RISE)

H2020-MSCA-RISE-2017

Intelligence-Driven Urban Internet-of-Things Ecosystems for Circular, SAfe and IncLusive Smart CITIES

D3.3 Software component and network monitoring

Abstract: This deliverable includes the specification and implementation of the basic monitoring mechanisms for networks and software components of the IDEAL-CITIES platform.

Contractual Date of Delivery	30/06/2020
Actual Date of Delivery	30/10/2020
Deliverable Security Class	Public
Editor	Andreas Miaoudakis
Contributors	NP, FORTH, CBN, BLS, BU, ENPC

The *IDEAL-CITIES* consortium consists of:

FOUNDATION FOR RESEARCH AND TECHNOLOGY -HELLAS	FORTH	GR
ECOLE NATIONALE DES PONTS ET CHAUSSEES	ENPC	FR
BOURNEMOUTH UNIVERSITY	BU	UK
BLUESOFT SPOLKA Z OGRANICZONA ODPOWIEDZIALNOSCIA	BLS	PL
CABLENET COMMUNICATION SYSTEMS LTD	CBN	CY
NODAL POINT SYSTEMS	NPS	GR

Document Revisions & Quality Assurance

Internal Reviewers

Andreas Miaoudakis, Anastasia Vayona, Kyriaki Constantinou, Jakub Rola, Rafal Szustkiewicz, Iraklis Tsatsoulis, Kostas Voglis.

Revisions

Version	Date	By	Overview
0.1	01/04/20	Vasilis Katos (BU)	ToC
0.2	05/05/20	Anastasia Vayona (ENPC)	CE parameter monitoring
0.3	20/07/20	Dimitris Mallis & Ioannis Chalkias (BU)	Monitoring concepts and tools
0.4	24/07/20	Jakub Rola (BLS)	Application profiling and Cloud services monitoring
0.5	12/08/20	Aristos Andreou (CBN)	Network Monitoring
0.6	14/08/20	Vasilis Katos (BU)	Introduction
0.7	16/09/20	Marios Angelopoulos (BU)	Network Management
0.8	20/09/20	Jakub Rola (BLS) and Vasilis Katos (BU)	Monitoring architecture
0.9	14/10/2020	Natalia Chechina (BU)	Application profiling & Database Monitoring
	14/10/2020	Kyriaki Constantinou (CBN)	Emergency Preparedness and Response Plans
1.0	19/10/20	Theodoros Kostoulas (BU)	Application Level Monitoring
1.1	27/10/20	Andreas Miaoudakis (FORTH)	Review and Editing

List of Abbreviations

CE	Circular Economy
CASB	Cloud Access Security Brokers
CTI	Cyber Threat Intelligence
DLP	Data Loss Prevention
DDoS	Distributed Denial of Service
EDR	Endpoint Detection and Response
ELK	Elasticsearch, Logstash, Kibana
EPP	Endpoint Protection Platforms
LaaS	Logging as a Service
HIPAA	Health Insurance Portability and Accountability Act
IAM	Identity and Access Management
ICMP	Internet Control Message Protocol
IoC	Indicator of Compromise
IPS	Intrusion Prevention System
LCA	Location, Condition, Availability
LEM	Log & Event Manager
LM	Log Management
ML	Machine Learning
NFV	Network Function Virtualization
NMS	Network Management System
PCI DSS	Payment Card Industry Data Security Standard
SaaS	Software as a Service
SDN	Software Defined Network
SEM	Security Event Manager
SIEM	Security Information and Event Monitoring
SIM	Security Information Management
SNMP	Simple Network Management Protocol
SOAR	Security Orchestration and Automated Response
SOC	Security Operations Centre
VM	Virtual Machine
UEBA	User and Entity Behaviour Analytics
UTM	Unified Threat Management

Table of Contents

1 Introduction

"You cannot control what you cannot measure" (p.3) [1]. Although this famous quote in the field of software engineering has been lately criticised, even by its author, it remains valid provided that the application domain and context is well understood and defined. By a high level of confidence, monitoring – the main process by which measurements are performed – is a key component of most systems, not just information, but also economic, societal, environmental and so forth. Monitoring provides the necessary information and intelligence to measure the performance of a system and more important allow troubleshooting and take corrective actions in the case of deviations. This is particularly evident in the cybersecurity domain where a significant effort in terms of research and funding is invested on Secure Information and Event Management Systems (SIEM).

At this point the difference between monitoring and audit will need to be stressed. Monitoring primarily refers to the continuous and near real-time measuring of the relevant system parameter; auditing is normally performed at a future point in time on historical information, and on events that took place in the not so immediate past.

In the context of IDEAL CITIES, monitoring refers to the near real-time and continuous measurement of the different layers and components of the IDEAL CITIES platform, to ensure that the system is performing at an optimal level and that is capable to meet the user demands. A substantial source of information is found in the logfiles of software services, operating systems and network appliances and a suitable log management framework is defined.

In addition to the monitoring specification of the developed platform, the business level monitoring is also considered. This introduces the data-driven circular economy concepts from the perspective of monitoring and describes how these can be mapped onto the IDEAL CITIES platform.

In the following sections a top-down approach is described. Starting from the monitoring concepts and tools, the services and network monitoring requirements are outlined and elicited. Then, following a document analysis approach the monitoring architecture for the IDEAL CITIES platform is presented.

2 Monitoring concepts and tools

A smart city is a system of systems that computerises the urban web and enables cyber-physical devices in order to increase its operative capabilities and performance [2]. Those devices are generators of overwhelming volumes of data that require to be sensed, captured, processed, communicated and (according to circumstances) stored, as long as the storage capabilities of the system allows it; the management and administration of a city depend on these data and along with them, the daily life of the citizens.

This new status quo of urbanisation requires constant and increased monitoring. Any possible abnormal incidents should be detected, analysed and amended in a prioritised manner, due to the number of possible incidents, with a response as close to real-time; as long as the severity of the incident allows it. In a complex combination of systems that also share data, monitoring is not a countermeasure limited to failing devices and networks. Failure in a smart city infrastructure might result to directly affecting human life or the environment.

The examples for that argument can be absolute. Monitoring in the medical industry could prevent a patient's pacemaker from shutting down or raise an alert for a patient with dementia and a forgotten treatment dosage. Sensors' alerts for detected smoke or increased temperature in an area could lead to the early detection of fire. Monitoring the smart grid could detect sudden voltage spikes that could damage a power station that is supplying electricity to a city or a hospital.

Beside the aforementioned examples, monitoring the smart city infrastructure shares the same scope with an IT department. This involves monitoring the operation (scheduling, data replication, accounting) and analysing the performance of networks, applications, edge devices, cloud infrastructures, hardware etc in order to optimise it. The success of the process maintains the operation of the supervised systems also from the cyber-security perspective. SIEM solutions, Intrusion Detection/Prevention Systems (IDS, IPS) are vital for a smart city, after considering the fact that there is an increasing number of devices that connect to the Internet.

2.1 Monitoring criteria

Monitoring is a process that includes the following four stages [3];

i. Generation of events – sensors perform sensing and propagate events following routing protocols and policies

ii. Processing of the events – the processing is application-specific and might include actions like data clustering, filtering etc

iii. Distribution of the events to the interested stakeholders – the processed events can be transferred to stakeholders coming from varied backgrounds with different objectives

iv. Presentation of monitoring results – the results need to be readable and conclusive in order to ease the decision-making process

The process of each step might differ for each of the industries that operate in the smart city paradigm, due to their respective and, specific requirements. In the diverse ecosystem of a

smart city, it would have been impossible to monitor a system, efficiently, without having to tackle a series of challenges like the factors mentioned below;

i. The performance of real-time monitoring is affected by the resources of the infrastructure and network conditions [4]

ii. Smart cities are comprised of heterogenous systems that are required to share data and resources [5]

iii. Automation of data processing is essential to successfully cope with the increased size of data [6]

iv. The sensors' readings should be properly analysed in order to understand the precise nature of the sensed event [7]

v. The numbers of false positives and false negatives describing security and performance events need to be reduced [8]

vi. Monitoring mechanisms require appropriate planning and careful deployment to increase their efficiency [9]

vii. Monitoring mechanisms should come up to speed with the advancements in the field of enabling technologies (e.g. 5G technologies and networks)[8]

Overcoming the challenges in the monitoring process is critical for the Quality of Services that a smart city has to offer. Its success depends on the level of achievement in the following goals [10], [11], [12];

i. Scalability – the monitoring system should be able to cope with increased amount of data, coming from an increased number of devices

ii. Robustness – the monitoring system should have increased resilience to overcome failure and adjust to changes while continuing its operation

iii. Non-intrusiveness – the monitoring system should add a lightweight overhead to the resources of the infrastructure

iv. Interoperability – the monitoring system should monitor multiple infrastructures by deploying commonly used protocols and file formats

v. Live-migration support – the monitoring system should migrate from one host to another without interruption of operations

vi. Privacy compliance – the monitoring systems should be compliant with the necessary data protection and privacy regulations, due to the nature of data generated in a smart city

vii. Passive monitoring – the monitoring system should not interfere with the operations of the monitored properties

viii. Situational awareness - the monitoring system should increase situational awareness for the system under observation

ix. Dynamic monitoring - monitoring should be dynamic and assist in predicting and preventing actions that are harmful to the monitored systems.

Accomplishing the goals mentioned above, is the key to build a monitoring system and a process that would provide an optimal level of situation awareness and reach a level of maturity where the smart city administrators could be making data-driven decisions. The prerequisite of this result is the compliance with the set of requirements that follow [13];

i. The monitoring service should be non-intrusive and should not degrade the performance of the monitoring system

ii. The monitoring service should give accurate estimate of large flows of incidents even if it misses some small flows

iii. The monitoring service must work across a heterogeneous mix of host platforms

iv. The monitoring service should export its data through rich interfaces and plug the easily into other systems

v. The monitoring service should ensure the integrity of the messages and provide trust checks, access control and support cyber-security services

vi. The monitoring service should not deplete the resources of the monitored property

2.2 Monitoring scope

As it has been analysed above, a smart city is a framework of vertical industries that employ a variety of devices and policies in order to provide services to users. This fact creates an extended range of devices that need to be constantly monitored in order to maintain their performance levels, raise alerts when abnormalities are detected and increase the situational awareness for the infrastructures and users. Regardless of the monitored system's industry, the monitoring scope that can be segmented in the following categories:

i. Network Monitoring – A network monitoring system monitors the components of a computer network (e.g. routers, switches, firewalls, servers etc) in order to monitor the network traffic, detect any possible faulty devices, improve network performance, configure network connections, create alerts and reports for the network management.

   Network monitoring systems can be deployed as firmware or software products. The process can also be outsourced to a third-party vendor. Successful network monitoring can optimise the use of the infrastructure's resources and provide compliance with the agreed Service Level Agreement (SLA).

   (Network monitoring is the subject of extensive analysis in chapter 4.)

ii. Cloud Monitoring – A cloud monitoring system is responsible for monitoring, reviewing and managing the operation of the cloud-based components of an infrastructure. With the majority of enterprises migrating their services to the cloud, the main types of cloud monitoring are the following [14];

   Database monitoring – Databases are an important source of data for most cloud applications. Database monitoring reviews processes, queries, availability, and

consumption of cloud database resources. This technique can also track queries and data integrity, monitoring connections to show real-time usage data. For security purposes, access requests can be tracked too.

Website monitoring - Website monitoring can check the availability, functionality and the performance of websites and their services. Their configuration and constant update can increase the user experience of the end users and also protect the websites from becoming hosts of malware.

Virtual network monitoring – It ensures that the virtualised components of a network are working properly and that the necessary virtual resources are distributed to the respective applications. Virtual network monitoring requires the collection of metrics relevant both to physical (hardware) and virtual (software services) performance of the hosts and virtual machines (VMs).

Cloud storage monitoring – Cloud storage monitoring provides metrics for the storage resources and processes acclaiformamed by virtual devices and services. Metrics such as CPU usage, disk I/O, memory, network traffic, uptime can be used to analyse the health and availability of the storage capacity.

Virtual machine monitoring – It enables the creation, management and governance of VMs. The VM monitor, also known as hypervisor, allocates the necessary computing, memory, storage etc to the built VMs and manages their operation, status and availability over a single or even multiple, interconnected hosts.

iii. Application Monitoring – Application monitoring is the tool that ensures and manages the performance of software applications in order to maintain the user experience at the desired level and produce alerts in case of issues, after collecting the relevant data.

Application monitoring collects a wide range of information that is important for the function of an application, with some of the most critical being the following metrics [15], [16];

- CPU usage – the restraint brought on the performance of the application
- Error rates – the frequency that the application degrades or fails
- Response times – the average time that the application requires to respond to a request
- Request rates – the traffic that the application receives
- Application uptime – the time that the application is available to a user
- Performance of dependencies – the effect of interacting elements on the application (e.g. databases, webservices etc)
- Code-Level performance – the analysis of the application at the code level to identify and improve its implementation

iv. Edge monitoring – Due to the increased numbers of edge devices in a smart city, monitoring the expanded edge of the network is another focal point for the monitoring process. The process should detect the devices that connect and disconnect to the network, monitor their interaction with it, their operation (usage, energy consumption, interaction with neighboring devices etc.), and their response times while maintaining the operation at the SLA.

The increased number of devices, deployed on the edge of the network and their limited capabilities in terms of processing power, memory and battery, raises a need for carefully crafted monitoring processes that balance the need for efficient monitoring and device performance.

v. Hardware monitoring – Hardware monitoring collects, reads and displays data from the hardware sensors of an infrastructure regarding its physical components; their status and availability, in real-time. The main metrics of a hardware monitor are temperature, fan speed, power supply, battery data, voltage, processor clock speed, disk array health [17]. This visibility can allow users to identify and solve any potential issues that could lead to devices being offline and removed from service.

vi. Cyber-security monitoring – With the cyber-threat landscape constantly changing and the enlarged attack surface of the smart city, cyber security is a critical element for preserving the operation of smart cities. Continuous monitoring can assist in detecting and managing security related incidents that could lead to data breaches, damage to infrastructure or its operations, financial loss etc.

Monitoring tools should detect threats and attacks against a system, provide alerts and sufficient information regarding the incident. The monitoring policies should be based on strategic analysis of the architecture's assets and the possible risks, extracted by a risk assessment. Extensive monitoring signifies the monitoring of the networks, systems, services and user activities of an infrastructure by an up to date tool that is setup appropriately to collect events and raise alerts based on a set of designated rules.

Further analysis for a series of available monitoring tools, their capabilities and their operations follows on the chapter 2.4.1.

A list of indicative products that offer monitoring tools appears on the following table. Due to the importance of monitoring, not only at the smart city, but also at the enterprise level, there is a plethora of commercial tools. Despite that, the number of open-source tools or tools that are supported by communities is not negligible. The majority of tools can provide monitoring in multiple fields, a fact that is expected due to common properties and devices that can be found in them and the overlaps in their operations. Cyber-security monitoring tools are listed in chapter 2.4.1.

Product Name	Vendor	Source	Monitored fields
Appoptics	Solarwinds	https://www.appoptics.com	Application
Sysdig	Sysdig	https://sysdig.com	Application

New Relic	New Relic	https://newrelic.com	Application
Cloudwatch	Amazon	https://aws.amazon.com/cloudwatch/	Application
Appdynamics	Cisco	https://www.appdynamics.co	Application
Opsview	Opsview	https://www.opsview.com	Cloud and infrastructure
US Cloud	Us Cloud	https://www.uscloud.com	Network and application
Solarwinds Real-Time Bandwidth Monitor	Solarwinds (free procuct)	https://www.solarwinds.com/free-tools/real-time-bandwidth-monitor	Network
Solarwinds Server health monitor	Solarwinds	https://www.solarwinds.com/free-tools/server-health-monitor	Hardware
Solarwinds Real-Net Performance Monitor	Solarwinds	https://www.solarwinds.com/network-performance-monitor	Network
Sensu	Open-source	https://docs.sensu.io/sensu-core/1.7/overview/what-is-sensu/	Full-stack
Prometheus	Open-source	https://prometheus.io	Application and server
Nagios	Open-source	https://www.nagios.org	Application and server
Icinga	Open-source	https://icinga.com	Network and server
Cacti	Open-source	https://www.cacti.net	Edge device
LibreNMS	Open-source	https://www.librenms.org	Edge device
Obsvervium	Community(freemium)	https://www.observium.org	Edge device
Pandora FMS	Open-source	https://pandorafms.org	Network and server
Spiceworks	Open-source	https://www.spiceworks.com/free-network-monitoring-management-software/	Network and server
Wireshark	Open-source	https://www.wireshark.org	Network
Nmap	Open-source	https://nmap.org	Network
Traceroute	Open-source	https://www.ultratools.com/tools/traceRoute	Network
Zabbix	Open-source	https://www.zabbix.com	
Netdata	Open-source	https://www.netdata.cloud	Cloud, infrastructuree, and application

Paessler RPTG Net Monitor	Paessler	https://www.paessler.com	Hardware
ManageEngine Applications Manager	ManageEngine	https://www.manageengine.com	Hardware
Cpuid HWMonitor	CPUID	https://www.cpuid.com/softwares/hwmonitor.html	Hardware
Open Hardware Monitor	Open-source	https://openhardwaremonitor.org	Hardware

Table 1: Monitoring Tools

2.3 Log management

Most of the organisations that own an IT infrastructure are investing in Log Management solutions. Such systems play a key role in increasing or at least maintaining cyber security resilience levels.

"A log file is a file that keeps a registry of events, processes, messages and communication between various communicating software applications and the operating system." [18]

Logs fall under the following categories, namely event logs, auditing logs, security logs and access logs; and they can be found in operating systems running on web servers, database or file servers, workstations, in security software such as antivirus, firewalls or those in routers, switches etc., but also in commercial and government off-the-shelf applications (COTS/GOTS). Their analysis can generate information that could be used for different scenarios including troubleshooting, system's performance optimisation, and monitoring/investigating users' activities. However, log analysis is often evaluated as a low-priority duty; mainly because it is manual, time-consuming, and the potential benefit is considered to be inversely proportional. Moreover, system and network administrators rely on the analysis of logfiles after an incident occurs reducing in that way the value of collecting and storing the logs in the first place.

Collecting and gathering in one place many and diverse data from a variety of sources is what makes log management challenging. All this information coming from the infrastructure components is usually generated under the syslog standard, transferred and stored in a central syslog consolidation server that offers large amount of storage capacity [19]. However, finding the way to balance effectively the limited resources with the ever-increasing number of log data produced requires the identification and the establishment of practices by the organisation itself. According to the "Guide to Computer Security Log Management" provided by NIST in 2015, the practises are the following and they should all adhere to relative laws and regulations by design [20]:

- Appropriate prioritisation of log management by defining requirements and goals.
- Establishment of the related policies and procedures.
- Creation and maintenance of a log management infrastructure in a secure manner.

- Assure that well-trained staff is assigned to handle log management responsibilities.

Depending on the sector an organisation belongs to, it might need to comply with specific guidelines and standards, such as the period and the availability of stored logs, to retain an appropriate audit trail history. For example, one of the requirements defined by the Payment Card Industry Data Security Standard (PCI DSS) [21] for the respective organisations is that they should be able to provide analysts and investigators with access to log files that cover at least one-year period of time with three months of instant availability. Other compliant systems like the Health Insurance Portability and Accountability Act (HIPAA) require from organisations to keep audit logs for a minimum of six years [22].

As previously discussed, establishing a centralised scheme for log management can address the need for long-term logfiles' storage, log-monitoring and the potential need for analysis. This approach simplifies not only management procedures but also those related to the log data protection from theft and/or alteration. Preserving security and privacy of log data is strictly related to the overalls system's/network's security as logs might record sensitive information like usernames and passwords, email accounts, to name a few. The respective concerns reflect all stages of log management (i.e. data transmission and storage) entailing both authorised and unauthorised individuals.

2.4 SIEM and supportive tools

A phenomenon that is being observed quite often is that log management and SIEM are mentioned together. They support a few similar capabilities and functions, based on the same principles, achieving analogous but also different goals. The main difference between the two is that SIEM, in contrast with log management systems and, can be largely automated, featuring real-time log data for security purposes.

Security Information and Event Management (SIEM), also known as Security Incident and Event Management, is a system designed to be used by an organisation's Security Operations Centre (SOC). Although SIEM tools come with automated functionalities by design, it does not mean that once the tool is configured it does the job by itself. There is a need for specifically trained personnel to make sense of the results such systems offer. Under these circumstances an organisation can exploit SIEM's full capabilities regarding incident detection and response. Moreover, as previously discussed, compliance with specific standards and regulations is mandatory for every enterprise that performs log management operations and SIEM systems are thriving to that area as well; by organising the relevant logs it is possible for a SIEM tool to create compliant reports in an effective and efficient manner, saving resources (i.e. time, personnel, costs) and at the same time avoiding fees that might occur in case of incompliance.

SIEM systems are available for decades both as products and services, but for the organisations they used to be costly to implement. Nevertheless, the cost of modern SIEM integration has been reduced compared to the old ones, while the system's efficiency was improved by new technologies such as Cloud Computing, Machine Learning, Artificial Intelligence. Security Event Management (SEM), Security Information Management (SIM), and Security Event Correlation (SEC) the three main functionalities combined in a SIEM tool. All exist for years as separate tools, but it is the fusion of their functionalities that makes SIEM systems so special in terms of security control. In other words, the primary functionalities of a SIEM app are the collection, standardisation, correlation, aggregation, and detection of

malfunctions across the organisation's IT infrastructure, and the notification to appropriate parties when unusual activity is observed or alarms have been detected. SIEM platforms are able to perform the aforementioned functions both on historical log data and on real-time events. They can also create relationships in order to help security personnel to identify abnormalities, vulnerabilities and incidents. The related information, known as Indicators of Compromise (IoC), can reveal the time, the total number of successful or unsuccessful login attempts, IP addresses of offending hosts, HTTP requests and many more. Apart from identifying IoCs it is also of high importance to secure and maintain the integrity of this information as they also serve as evidence in digital forensics investigations.

Endpoint Protection Platforms (EPP) that typically include Antivirus, Firewalls, Intrusion Prevention Systems (IPS), etc., and Patch Management Software and Tools are typically leveraged by the enterprise in order to keep the infrastructure up-to-date and protected from threats. As EPP is more like a first-line defence mechanism used in order to block known threats, Endpoint Detection and Response (EDR) tools cover the next level of security by detecting and replying to threats. Besides these technologies, Data Loss Prevention (DLP) solutions are also very useful as they focus on preventing unentitled users from sharing sensitive data. These actions are complemented by Identity and Access Management (IAM) solutions as they connect disparate authentication services in order to compel the user request a single service each time he/she tries to access on premise or cloud systems and applications. In essence, EDR and IAM provide information that can be fed to SIEM, User and Entity Behaviour Analytics (UEBA), DLP and Cloud Access Security Brokers (CASB) systems. The latter is another key element in the enterprise's security mechanism as it keeps records of who is using cloud applications according to their access privileges. Moreover, Security Orchestration and automated response (SOAR), which has been recently introduced, are systems that detect an alert generated by the SIEM and automatically perform containment measures on the system concerned to reduce response times, improve consistency and increase incident-response team's productivity.

In conjunction with free or paid Cyber Threat Intelligence (CTI) feeds, advanced solutions like the ones described above can further upgrade SIEM system's performance as CTI feeds provide insights regarding existing and future security threats. This cooperation gives greater sensitivity to the SIEM allowing it to be more accurate in the identification of incidents while facilitating the formation of decisions and the implementation of strategies.

Ultimately, an appropriate endpoint and network monitoring systems consisting of such technologies would beneficially contribute the identification and mitigation of cyber threats faced by organisations.

2.5 Log Management and SIEM Tools

Log management and SIEM solutions can be either self-hosted (on-prem) or cloud-based, also known as Software as a Service (SaaS) cloud logging and/or Logging as a Service (LaaS). Each of the two options comes with pros and cons for the organisation; they are briefly discussed below.

On-premises solutions exist prior the cloud-based computing services (i.e. IaaS, Paas, SaaS), which in general, emerged in the late '90s. Cloud monitoring, in particular, belongs to the early 21st century innovations. As shown in the following figure, the blocks in cyan colour depict

elements of an IT infrastructure managed by the organisation owning the infrastructure whereas the red blocks refer to the IT components that belong to an organisation but they are managed by a third-party company offering cloud services.

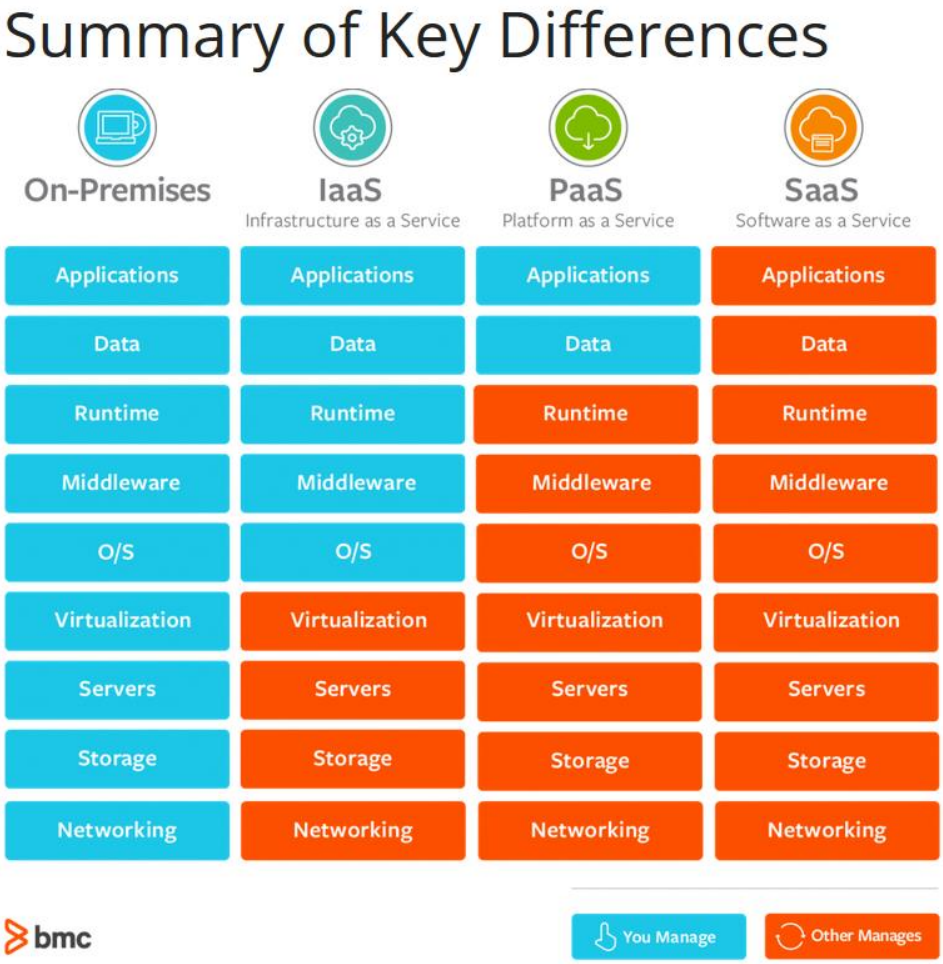

Figure 1 - Key differences between On-Prem, IaaS, PaaS, SaaS computing services [23]

Cloud logging solutions fall under the last category (i.e. SaaS) which, at a glance, shows that all the relative network components are managed by the cloud service provider. The opposite is true for the on-prem logging solutions. That is the main difference between the on-prem, and cloud-based logging solutions and it is also the root of the advantages and disadvantages deriving from each option.

On-prem logging advantages:

- Data stay on-site
- Avoid risks associated with transferring and storing of data on the cloud
- Full control maintenance (of the platform, system, personnel, etc.)

On-prem logging disadvantages:

- Significantly high cost (for additional hardware/software, hiring/training specialised personnel)

- Delays (resulting from the need for tailored configuration to the context of the organisation, the long learning curve of the trainees, and the limited integrations)

Cloud-based logging advantages:

- More cost-efficient (by avoiding the additions required for the on-prem solutions)
- Reduced delays (by leveraging pre-trained cybersecurity experts along with a pre-configured platform)

Cloud-based logging disadvantages:

- Data transferred off-site (resulting in limited access/control of data and security/privacy risks as both the data stored in the cloud and those in-transit could be vulnerable to malicious actions)
- Alert fatigue (a phenomenon that occurs when the administrators from the organisation hiring the cloud-based logging services might get distracted from real threats when constantly dealing with multiple and repetitive alerts/notifications)

2.5.1 Kiwi Syslog Server (on-prem)

Kiwi Syslog Server is promoted as a powerful and cost-effective unified console for robust network monitoring. Its capabilities include management, filtering and forwarding of syslog messages, event logs, and SNMP traps coming from both Windows and Linux/Unix hosts. Reaction to syslog messages according to built-in actions, smart syslog alerting, and secure transmission of logs are also among the functionalities offered by Kiwi Syslog Server. In addition to those, this tool supports automated archive and clean-up tasks that help the organisation comply with log management regulations (e.g. PCI-DSS, FISMA, SOX, etc).

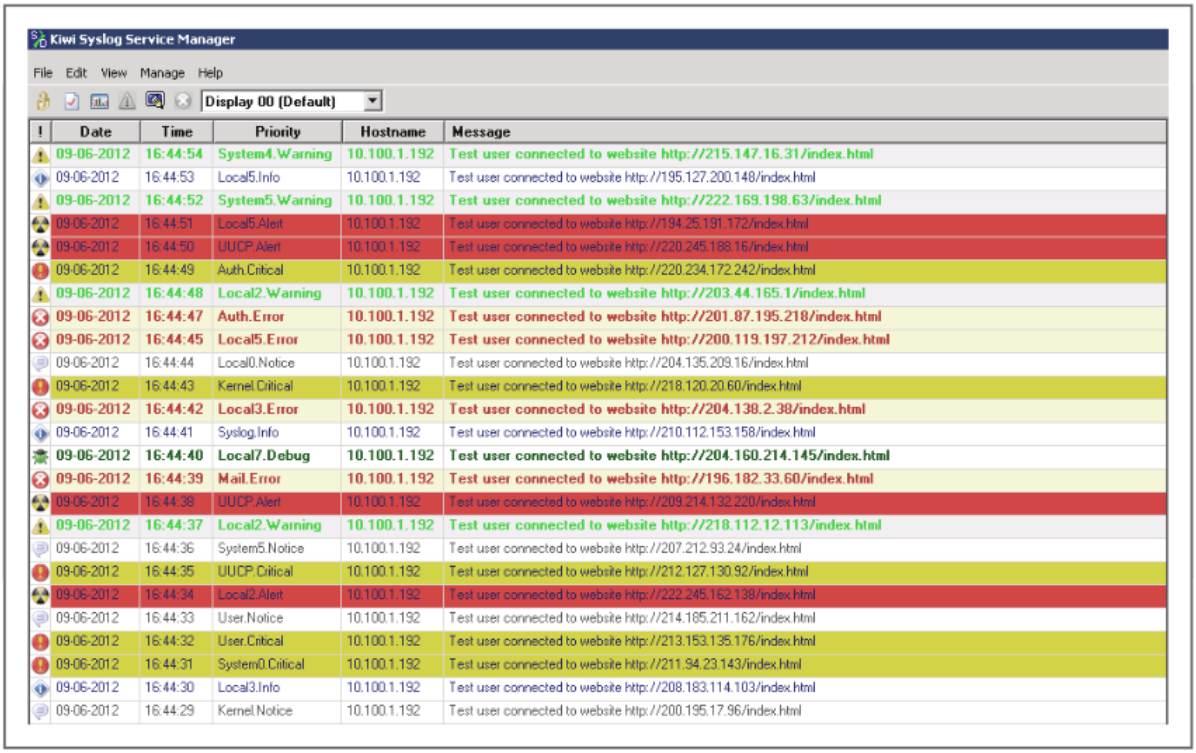

Figure 2 - Kiwi Syslog Service Manager [24]

2.5.2 SolarWinds Security Event Manager (SEM) (on-prem)

About a year ago (end of May 2019), SEM [25] was introduced as an upgraded version of the Log & Event Manager (LEM) which was the former SIEM product offered by SolarWinds. The upgrades refer to all the components (i.e. console, agents, and reports) and the features that were available in LEM. As a standard SIEM solution, SEM collects, normalises, and aggregates messages coming from several IT and ICT network components using a powerfull analytics machine. Advanced search and filtering capabilities help the organisation leveraging this tool to act proactively by tracing potential cyber threats. Real-time identification of security breaches and compliance validation are also possible thanks to dedicated reporting software that comes with SolarWinds SEM.

Figure 3 - SolaWinds Security Event Manager (SEM)

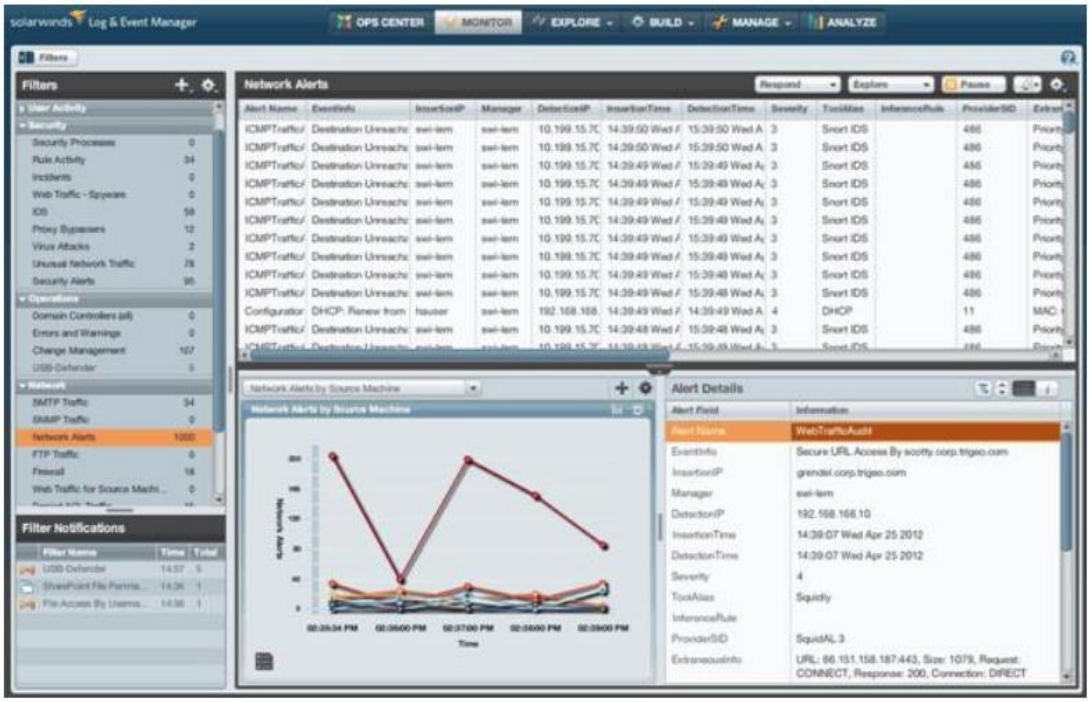

Figure 4 - SolarWings Log & Event Manager (LEM) [26]

2.5.3 Splunk Security Enterprise (SE) (on-prem and cloud-based)

It is common for log management and SIEM tools to offer a free trial version for a specific period (usually 2-4 weeks). Splunk, however, advertises three product/service alternatives regardless of timeframes. As can be seen in Figure 5, the first option is free but limited in terms of the availability of certain features while the other two options are almost identical (cloud-based services offer an extra feature: Dynamic Data).

Features	Splunk Free	Splunk Enterprise	Splunk Cloud
Maximum Daily Indexing Volume	500MB	Unlimited	Unlimited
Maximum Users	1	Unlimited	Unlimited
Universal Data Collection/ Indexing	✓	✓	✓
Metrics Store	✓	✓	✓
Data Collection Add-Ons	✓	✓	✓
Monitoring and Alerting		✓	✓
Dashboards and Reports	✓	✓	✓
Search and Analysis	✓	✓	✓
Event Annotation	✓	✓	✓
Automatic Data Enrichment	✓	✓	✓
Anomaly Detection	✓	✓	✓
Tables, Data Models and Pivot	✓	✓	✓
Splunkbase Apps	✓	✓	✓
Splunk Premium Solutions		✓	✓
High Availability		✓	✓
Disaster Recovery		✓	✓
Clustering		✓	✓
Distributed Search		✓	✓
Performance Acceleration		✓	✓
Access Control		Granular and Customizable	Granular and Customizable
Single Sign-On/LDAP		✓	✓
Developer Environment		Full access to APIs and SDKs	Full access to APIs and SDKs
Dynamic Data			✓
Support	Community	Standard or Premium	Standard or Premium

Figure 5 – Splunk pricelist of services

Splunk's Security Enterprise is one of the most popular and efficient SIEM solutions currently available in the market. However, its licencing pricing policy may not be applicable to some SMEs. Splunk ES is an analytics-driven SIEM and in addition to both basic and state-of-the-art SIEM capabilities, SE facilitates User Behaviour Analytics (UBA), Machine Learning (ML) and SOAR tools for anomaly and threat detection [27]. It can be deployed as a self-hosted or cloud-based solution and in some cases as a hybrid model. Depending on the licencing packet purchased, it offers a good level of scalability as there are no specific limitations (e.g. users, servers, capacity).

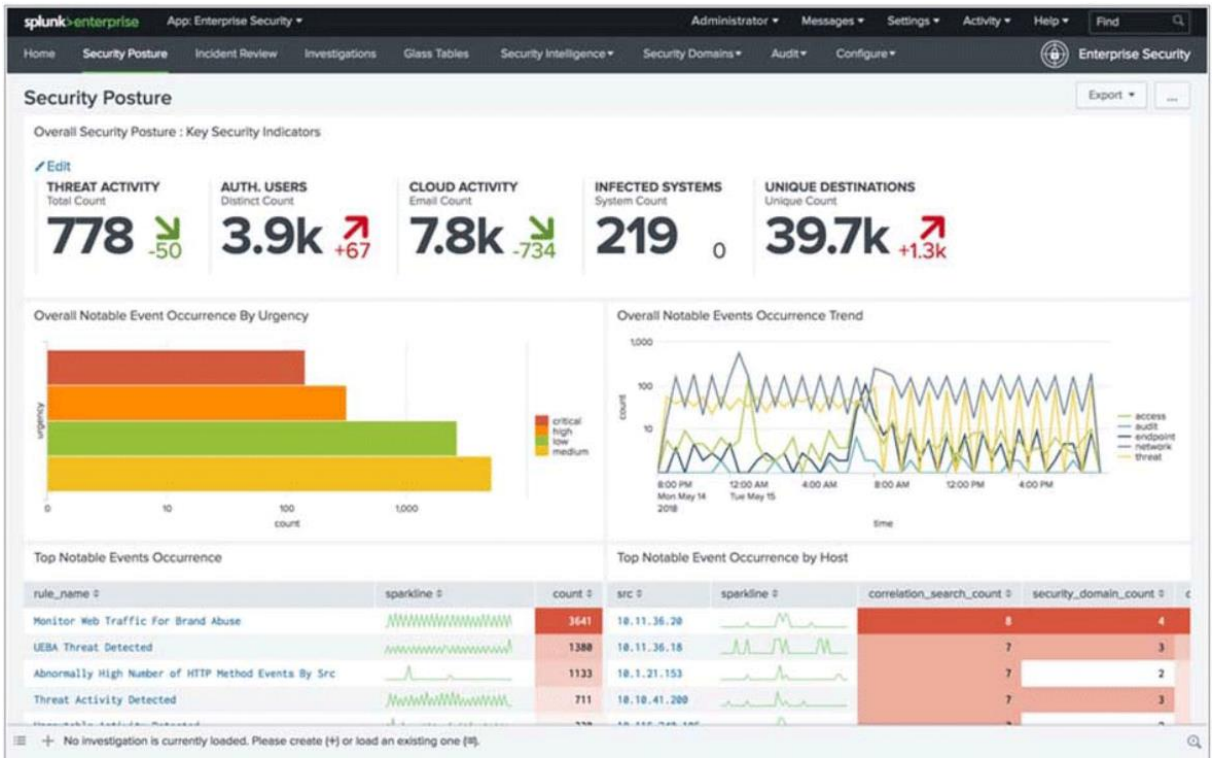

Figure 6 - Splunk Enterprise Security (SE) [28]

2.5.4 SolarWinds Loggly (cloud-based)

Loggly is a SaaS log management and analysis tool offered by SolarWinds. It comes in three packages (i.e. Lite, Standard, Enterprise) with each one of those followed by analogous limitations. Its architecture does not involve agents making, in this way, easier to collect logs from different sources and aggregate them. On top of that, its users would find Loggly quite handy as it comes with a very easy-to-use User Interface (UI), and also because it is compatible with a variety of systems (Windows, Linux, Mac) running on public/private clouds, private servers or in containers. This tool also guarantees interoperability with other tools a team might use in daily operations by offering a full REST-ful API. This feature allows the integration of tools like GitHub, Slack and Jira, among others, which will further support the creation of tickets and alerts more efficiently.

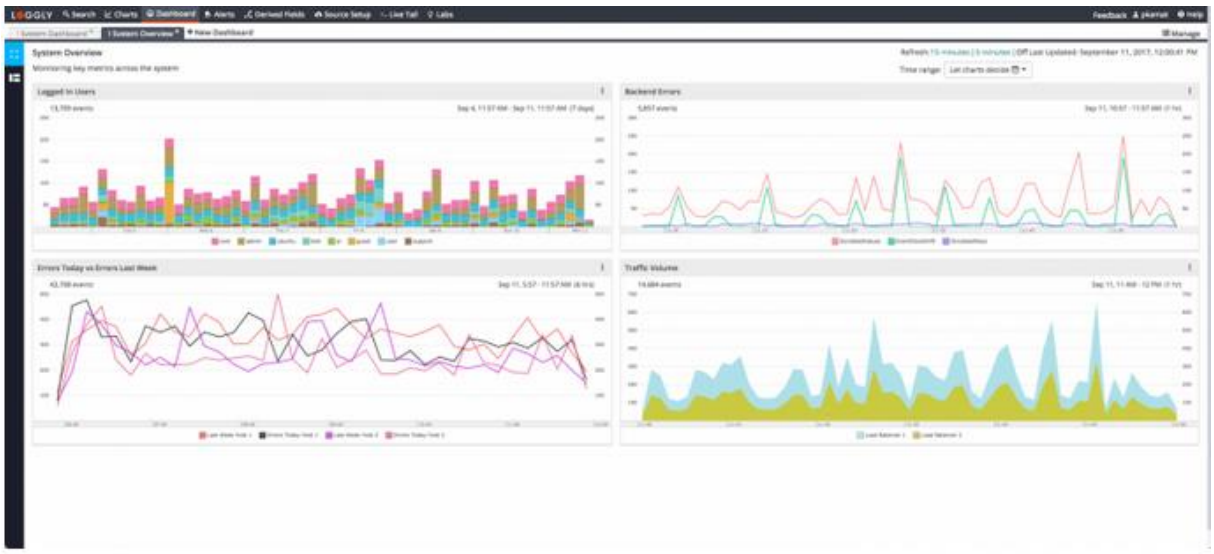

Figure 7 - SolarWinds Loggly [29]

2.5.5 Elastic (ELK) Stack (cloud-based)

Elastic Stack [30], or else known as ELK Stack, is a suite comprised of three tools (Elasticsearch, Logstash, Kibana) that targets to provide SMEs with a no/low-cost monitoring solution for their logfiles. The ELK stack community promotes three main reasons for an organisation to adopt this tool as a log management solution and those are the following:

- It is open source

  Avoid upfront costs as deployment could only require time to learn about the tool and how to use it.

- It is interoperable

  Elastic Stack can replace Logstash, Kibana, and maybe some more of its complementary tools with similar ones (e.g. Fluentd [31] for logging or Grafana [32] for visualisations) that are already part of the user's infrastructure.

- It offers managed services

  In cases where time is more valuable compare to the relevant financial costs of subscriptions, Elastic Stack vendors are offering the aforementioned tools along with managed services. Some examples are the following: Elasticsearch and Kibana hosted on AWS, Azure, and GCP.

On top of all the arguments described above, Elastic Stack is also able to handle logging in a microservices architecture where logs are created on every instance of each application. All the logs coming from different sources are imported, compiled and sent by Logstash to Elasticsearch. Instead of Logstash, logs can also be collected via Beats [33] which comes as a lightweight option as it consumes fewer resources than Logstash. Once the data from Logstash has been aggregated and transformed (if necessary), Elasticsearch indexes everything for further analysis. Kibana can be used to pull data from Elasticsearch for the visualisation, dashboarding and much more by using a query language (KQL).

Figure 8 - Elastic (ELK) Stack [34]

2.6 Dashboards

User and Entity Behaviour Analytics (UEBA) platforms are SIEM visualisation and dashboarding tools that can model insights which can be discovered by security analysts or sent to them as notifications/alerts. Most of the SIEM products and services, currently available, offer these functionalities in both standard and premium packages as they play a significant role in the overall performance and objectives each organisation is targeting to achieve while using these tools.

For a business, the idea of visualisation and dashboarding serves the need for an effective and efficient summary of *what is happening now or in the past*, and *how the business is affected*, which will further indicate actions or at least support decision-makers decide *what can be done to mitigate/ improve the current situation and/or plan strategies and actions to meet future goals*.

"A **data dashboard** is an information management tool that visually tracks, analyses and displays key performance indicators (KPI), metrics and key data points to monitor the health of a business, department or specific process." [35]

Three main types of dashboards that add value to the Business Intelligence are the following:

- **Operational**
  Provide the user with a quick overview of performance in a specific period of time (e.g. per day)
- **Analytical**
  Provide the user with trends and/or abnormalities based on real-time or historical data that reflect the current situation.

- **Strategic**
  A correlation between the KPIs which define the long-term strategy and the performance tracks the organisation is holding can formulate dashboards from a strategic point of view.

For whatever the objectives may be, all types of dashboards should follow standards in the design process. Effective Dashboard Design includes practices that are well-described by Andrea Janes et al. [36]; however, it is worth mentioning the fundamentals. There are two pillars upon which the practices are developed and those are the *Data* to be presented and the *Graphics* to be used.

As for the Data, it needs to be clear which information is relevant to the specific objectives the dashboard is aiming to serve. This might relate to the:

- Context of the data
- Amount of the data
- Timeframe of the data

Although a clear definition of the parameters above would clarify the relevant data to be used it is the graphical designing and presentation processes, which act as the means of information communication, that will create and deliver the knowledge/intelligence contained in the dashboard. So, the Graphics are also very important as *colour*, *position, texture, shape and size* are the design characteristics adhering to the best user's experience (UX) when properly used. The following should be considered when designing the graphics of a dashboard.

- It should be effortless for the user to absorb the given information from the dashboard.
- Any interaction between the user and the visualisations of the dashboard in order to reveal the actual information should be avoided.
- The position of data should be arranged in such way to minimise the necessary time for the user to consult the dashboard.
- Positioning and gradual revelation of data should guide the user's attention to the most important data.
- The design language and colour scheme should be consistent for the creation of a UX-friendly dashboards.

Dashboards can visualise and surface virtually any type of information. They are meant to provide the user with the most relevant details inside a clear screen that is easy to understand, and the structure should represent a logical hierarchy of knowledge, which would provide the user with the possibility of diving into the details if necessary.

3 Services monitoring

By following a top-down exploration of the different services engaging in the delivery of a CE aware platform, we start by the business logic level, then to application profiling which is also closely coupled with database monitoring and finally the cloud services. The network monitoring is explored in detail later on in Section 4.

3.1 Business logic level monitoring

Monitoring capabilities are quintessential to data-driven circular economy. Monitoring on all levels (from operational to management) is performed with varying degrees of granularity, to ensure that the assets are utilised and interventions are performed whenever needed. Moreover, both real-time decision-making and mid to long term planning, can be efficient if the right type of information is available to the right party.

In order for monitoring to be deployed, there is a requirement to reach consensus and adopt appropriate circularity indicators. This is highlighted by the EC [37], stating that "to assess progress towards a more circular economy and the effectiveness of action at EU and national level, it is important to have a set of reliable indicators". In general, and from a business and policy perspective, the scope of an indicator can be on one of three levels, namely micro (products, companies and consumers), meso (eco-industrial parks) and macro (city, region, nation and beyond) [38]. In IDEAL-CITIES, the focus is placed on the indicators and the monitoring the IT infrastructure and smart assets, as an attempt to study and explore the CE and IoT value drivers [39]. This approach translates to monitoring the use of cases on an operational and management level. In principle, the higher level of monitoring, one departs from the real-time requirements and experiences an aggregation of information, where trends can be observed and long-term decision making can take place.

At an **operational layer**, fine grained monitoring is performed in real time. The circularity-enabling properties of Location, Condition and Availability (LCA) are used to specify the circularity state of any asset, infrastructure related (e.g. networked device), or IoT. Although the LCA triad is primarily used to deliver a CE centric business model (by making sure the asset is utilised in the most effective manner throughout its entire lifecycle), for a use case and scenario like the emergency evacuation of an area or building this information is of high value as it will provide the means to make real-time decisions taking into consideration device reliability and trustworthiness aspects. Such assessments are not normally performed in non-CE environments, or if they are they are part of the risk assessment and business continuity processes, which are not dynamic and are performed in a "paper type" exercises, and in a particular point in time in the past. That is, the LCA CE-enabling properties, extend and improve the concepts of Mean Time to Recover and Mean Time Between Failure, as they provide a more tailored and hence of higher integrity and reliability measurements, rather than having to rely on the generic and device manufacturer's information sheets.

At the **management and strategy layer**, the utilisation of the smart assets is monitored and this can be performed through reporting activities that produce aggregate reports on a daily, weekly and monthly basis. These reports can summarise the utilisation of the assets as well as their condition, in order to alert and warn when intervention is needed to refurbish, repair, recycle or remove assets. The historical information on availability captures the demand levels allowing the calculation of demand trends and predictions. From a CE standpoint is a business

critical function as not only underutilisation of a resource can be observed, but also overutilization due to elastic demand can be measured, which is also a situation to be avoided as it will be detrimental to the overall CE asset ecosystem. As such, striking the right balance can only be performed with accurate and continuous monitoring.

3.2 Application profiling

When the production-ready application is deployed on the cloud provider platform like AWS, AZURE, IBM Cloud, etc. Cloud providers offer market-ready tools for monitoring the state of an application. Most of the cloud platforms offer pre-configured tools for monitoring the health of the deployed application. The options for an administrator of a cluster are diverse and range from the simplest logs with basics events, the summary periodical reports, graphical dashboards, and analysis of events through the use of ML algorithms for failure prediction.

For a platform that implements the microservices architecture the most important metrics are service availability and response time of each component. Time of response and traffic to each component are the main indicators for the horizontal scaling of a given component. The availability is crucial for the proper execution of a functionality provided by a given component.

Being able to quickly identify bottlenecks and understand the relationships between different components in the cloud-native system is a huge differentiator. The ability to monitor the response time of each component for transactions allows identifying the possible malfunction autoscaling mechanism.

The main Key Performance Indicators used in profiling the performance of cloud applications are % of CPU and RAM usage. During the lifetime of the application, some of its components may become the bottlenecks of the platform performance. It is crucial to indicate the possible weak links in the microservice environment.

The main indicator of user experience is the response time of the user request. This parameter needs to be monitored to ensure the best possible performance of an application. In a microservice architecture, the end-user request is forwarded between many components before the final response is sent back. Monitoring the response time of each component provides a metric that indicates the bottleneck component of the application. Collecting metrics of CPU utilization, memory utilization, TCP/IP connection, and response time can be statistically analysed and correlated. Statistics analysis provides a better understanding of the dependencies of microservice architecture.

Collecting the above-mentioned metrics may lead to storing large amounts of data on a cloud disk space which need to be paid by operators. This issue has been already identified and studied, to minimise the amounts of collected data that can be implemented by monitoring policies. Those policies set the polling frequency depending on the current variance of resource utilisation. More information can be found in the conference paper “Autonomy of Cloud Monitoring and Metering” [40].

From the above, we have contradicting demands. On the one hand, we need to collect information as frequent as possible to have full information about trends and then develop

accurate predictions. On the other hand, as the number of devices grows (a) it becomes more expensive to collect and store information on the cloud and (b) the amount of generated traffic also grows which leads to jamming the networks. To address these demands we can utilise edge technologies [41] which will (1) enable us to maintain a high frequency of collection of monitoring information at scale, (2) keep cloud costs and communication relatively low, (3) enable resilience and timely responses in cases of failures and attacks (both cyber and physical). The monitoring will be performed on three levels: on the devices themselves, on the intermediate stations, and on the cloud.

Monitoring on the devices. Some straightforward and basic monitoring can be implemented on the devices.. This monitoring should neither include complex computations nor require storage of large amounts of data. The frequency of the information collection (F1, Figure 9) should be defined by the purpose of the device and the required response time on a failure or abnormality. The devices would collect state information for some time and then send a "summary" to one of the intermediate stations with frequency F2, where $F1 \gg F2$. When an abnormality is detected, depending on the type, the device may send an immediate warning to its intermediate station, or provide the information as part of the regular summary.

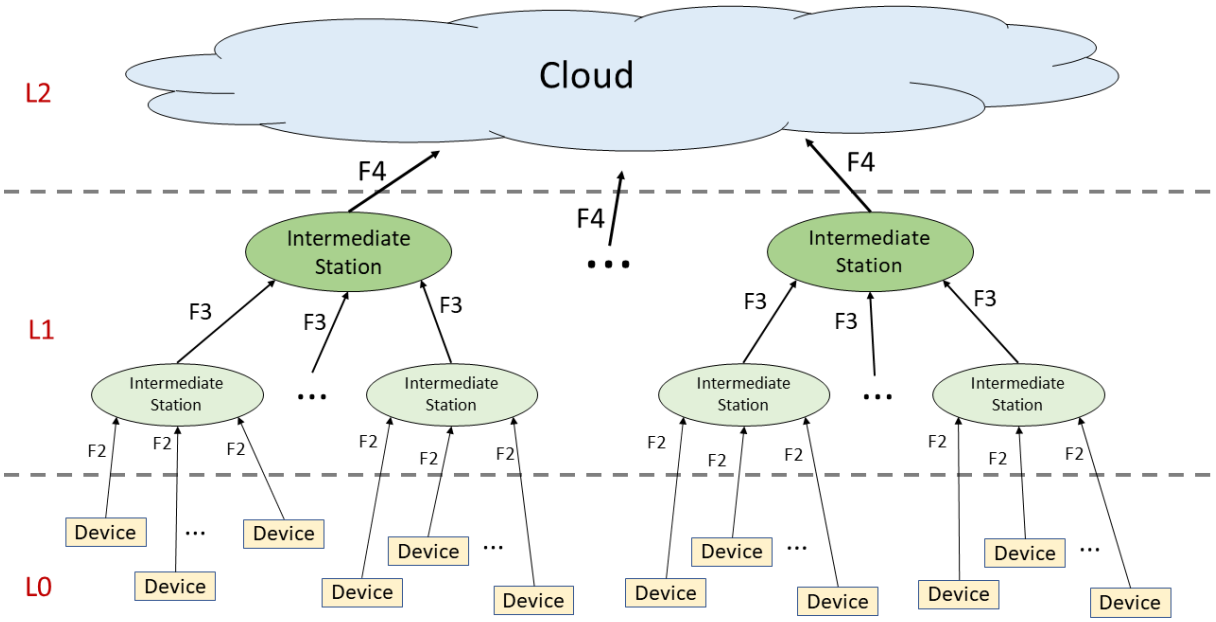

Figure 9 - Resilient monitoring at scale

Monitoring by the intermediate stations. The intermediate stations can be of different computation power and depending on the scale may consist of multiple layers. Their function is four-fold.

1) to collect information from the level below and forward the summary to the next level significantly reducing the amount of forwarded information and its frequency, i.e. $F3 \ll F2$.
2) to conduct some limited analysis on the state of the monitored devices in case some need either urgent actions or alerts.

3) to ensure scalability -- by introducing an intermediate station for a set of devices we reduce a load (demand) on a single point and eliminate a single point of failure (going for scalable multi-layer decentralised approach, rather than non-scalable centralised approach).

4) to provide resilience -- by introducing an opportunity for the devices to join another station while their station is recovering from a failure, we ensure that we keep receiving monitoring information from the devices and are aware of the state of the whole system.

Monitoring on the cloud. The cloud would receive information from intermediate stations. Its main purpose is to have full information about the whole system, conduct full-scale analysis, and develop predictions.

3.3 Database monitoring

To be able to handle and maintain information at the scale of a smart city the database should be resilient to failures and cyber-attacks. Distributed NoSQL databases are widely acknowledged as effective and state-of-the-art tools to ensure resilience to failures at scale. Following the CAP theorem [42] a database can only satisfy two of the following three criteria:

- Consistency -- defines the level of synchronisation between different parts of a distributed database. Strong consistency ensures that all reading requests receive the most recent information.
- Availability -- defines the database responsiveness independently of the failures of some of its components.
- Partition tolerance -- defines responsiveness of the database despite networks and communication failures, and the ability of the database to find a consensus regarding the latest version after the recovery.

Since the cloud collects information concerning the whole system for analysis and prediction, and it is neither practical nor vital to have 100% accurate information from all sensors at all times, the most suitable model for the distributed database is AP+EC (Availability, Partition tolerance, and Eventual Consistency). This means that despite communication, synchronisation, and other types of failures the database should be responsive at all times by providing reading request of available information and accepting writing requests. In addition, there should be mechanisms in place that enable smooth and seamless reconciliation of the database after a partition; while the consistency between various parts of the database is not a guarantee but is the state the databases constantly strives for.

Some other important aspects that should be considered when choosing an architectural approach for the database include the following [43]:

- Functional requirements e.g. scan queries, sorting, full-text search. These define techniques on the bases of which the database is developed, such as sharing, replication, storage management, and query processing.
- Geographical distribution of database components and backups to ensure the database availability despite either component, or server, or datacentre failures.

- Type of replication, i.e. eager (synchronous) or lazy (asynchronous).
- Dependencies between replications, e.g. master-slave, multi-muster.

As in any monitoring of a large-scale database, the monitoring function should be lightweight collecting only essential information to analyse its state, (e.g. level of partitioning, the proportion of dropped writing requests, network failures, and database internal failures). At the same time, it should be ensured that the monitoring neither slows down nor takes resources from the database. The monitoring can be implemented as part of the database, or a separate monitoring tool, or a mixture of two. The analysis of the information can be conducted either online or offline. Online monitoring is essential to understand the state of the database and an ability to take swift actions to address the problems. It should only include easily accessed information that does not require significant computational resources. The purpose of offline monitoring is to provide insights into database performance over a period of time and enable planning for improvements, reengineering, and significant changes.

3.4 Cloud services monitoring

Commercial cloud providers charge their clients by occupying the CPU, RAM, and disk space. Those indicators are crucial for companies that are using cloud providers due to the billing amount. Most of the Cloud providers provide (?) their clients the tools for monitoring the utilization of physical resources.

One of the main concerns and tasks of the cloud administrator is to configure the deployment of each component in a way that it utilises the physical resources, balancing the good end-user experience whilst maintaining low operation costs. The main advantage of cloud solutions is the flexibility of resource utilisation. The cloud allows the vertical and horizontal scaling of the system. In the cloud solutions, vertical scaling adds extra capability or power to a single component by increasing memory and computing power. At the same time, horizontal scaling adds more replicas of service to the cluster.

The goal for DevOps who administrate the cloud application is to optimise the utilization of physical resources for good performance of a deployed application. Administrators can manage the settings of each component and how many replicas of each component are deployed on the cloud. Despite the high complexity of the microservice architecture and the increase of physical resources, the multi-component structure ensures better scalability of the application and separates the functionality of the system. Flexibility in scaling the applications increases physical resources by increasing the billing amount, but at the same time allows delivering the high-performance platform. Operators of cloud applications need to find the middle ground between high performance and the cost of running it. This balancing can be achieved through limited scale up and down of the number of offered replicas as required. To accomplish the needed level of service the microservice horizontal scaling need to be implemented. Horizontal autoscaling mechanisms need to be triggered by a metric that best indicates the performance of each microservice. Administrators need to define the best set of metrics that can trigger the autoscaling mechanism of each microservice. The indicators used as metrics are CPU utilization, RAM availability, and free disk space. DevOps can change the

frequency of checking the metrics, scaling algorithm and strategy, behaviour when the metrics are unavailable, and period between the scaling action.

4 Network Monitoring

Network monitoring is a critical IT process where all networking components like routers, switches, firewalls, servers, and VMs are monitored for fault and performance and evaluated continuously to maintain and optimize their availability. One important aspect of network monitoring is that it should be proactive. Finding performance issues and bottlenecks proactively help in identifying issues at the initial stage. Efficient proactive monitoring can prevent network downtime or failures [44].

As it is easily understood, network monitoring and its various components, are an important part of any network setup. For this reason, most network equipment vendors (Cisco, Juniper, Huawei, HPE, Fortinet etc) are developing their own monitoring tools (a widespread term for these tools is Network Management System – NMS) so that customers can manage, configure and monitor their networks. Nevertheless, there are also a number of open source applications which can act as an independent 3rd party NMS for any type of network setup either single or multi-vendor. A list of applications that provide monitoring tools are presented in paragraph 2.2.

Network monitoring is based on a number of protocols which enable networking equipment to communicate with any compatible NMS in order to exchange relevant information. Most popular monitoring protocols are Simple Network Management Protocol (SNMP), Internet Control Message Protocol (ICMP), CDP, jFlow, sFlow, IPFIX, HTTP, FTP etc. Some are industry standard and others are vendor proprietary. The most common protocol for network monitoring is SNMP.

With the introduction of IoT the importance of network monitoring has been increased. New devices are introduced into IoT networks as well as things that until now did not have the need of connectivity are becoming “smart” and connected thus need to be configured and monitored.

Moreover, the threat of security risk is also becoming bigger as more and more devices and sensors are connected online. In fact, an HP study revealed that vulnerabilities appear in more than 70% of devices that are connected to IoT networks [45].

4.1 Configuration management

Network configuration is the process of defining the network operation, flow, controls and parameters.

Configuration Management is defined as the set of functions which manage and organize network parameters and attributes, prepare configuration changes scripts and track network configuration changes. At the core of this process is the network configuration management database. When something goes wrong and the network needs repair, changes, or upgrading, the network admins will look at the database for the necessary information, so they can make an informed choice as to the best course of action. The database stores the IP address and location of every hardware device, in addition to data pertaining to programs, versions, updates, and default settings.

In general, when a network admin is looking into deploying a configuration management tool in a network, the below key features/tasks should be taken into consideration:

- Maintain baseline device configuration

- Ability to rollback to previous configurations
- Archiving network configuration changes
- Firmware updates/upgrades distribution
- Device backups
- Bulk configuration updates

As already mentioned in [46], the introduction of IoT networks increased the need of Configuration Management which now becomes a high priority process. IoT environments are characterized by a high degree of device heterogeneity and network protocols stack, where each object may have different processing capabilities or different communication patterns, resulting in a fragmented technology landscape. This demands a dynamic and context-aware configuration management system. One of the most important challenges in the management of IoT networks is the heterogeneity of the devices that belong to the network, using many different technologies at each layer. Thus, it is important that a management platform enables the discovery of a device in the considered environment dynamically, to meet the requirements of the applications. The network management should be based on the following five functional components: configuration, failure, member, report, and state:

- The configuration (self-configuration) is responsible for performing system configuration initialization functions, such as collecting and storing the configurations of the other functional components and devices.
- The failure (self-aware) aims to identify, isolate, correct, and record failures that occur in the IoT system.
- The member is responsible for handling member associations of the IoT system and important information from any relevant entity (IoT service, device, applications, user).
- The report allows the information refining provided by other management functions, generating reports or retrieving reports from a history.
- The state (self-monitoring) aims to monitor and provide the past, present, and future states of the IoT system that are required by the Failure function. It has a function to change or apply a particular state in the system.

This already heterogeneous and highly diverse landscape is becoming increasingly more complicated with the introduction of additional technologies (e.g. mmWave and sub-GHz radio access in 5G networks), applications (e.g. ultra-high capacity networks and vehicular networks) and stakeholders (e.g. local authorities, neutral network hosts, etc). These ongoing advancements aggravate the challenges related to network monitoring and configuration management.

Up until very recently, network management has been addressed in an ad-hoc way (mainly due to the distributed way of Internet development), which has led to a fragmented landscape of protocols and methods. For instance, protocols dealing with access control and traffic engineering have been developed independently from one another and assume a different approach (localized management vs centralized computations) thus hindering a unified and scalable method of overall network management. While so far pertaining issues have been addressed efficiently, the problem quickly grows towards not being manageable.

In particular, with respect to the *control plane*, new network applications introduce new network topologies that carry distinctive qualities; for instance, in MANETS [47] and VANETS [48] the network topology can be highly dynamic and prone to frequent changes over time. Also, the increased capacity of the emerging 5G networks will support the concurrent connection of several devices, significantly increasing the scale of the network. With respect to the *data plane,* the support of novel application areas by emerging networks also poses significant challenges related to data attributes such as their volume, volatility and veracity.

This is nicely demonstrated by the challenges faced by network operators that are struggling to efficiently manage large datacenters (typically consisting of hundreds of thousands of machines and tens of thousands of switches) offering multi-tenancy services to thousands of customers. This rise in complexity highlights the need to *extract simplicity* with respect to network management. Extracting simplicity necessitates extracting abstractions that obfuscate underlying dependencies. Similarly to the approach taken in areas such as Operating Systems and Databases, this comprises mitigating dependencies on hardware (e.g. programming switches in machine languages) by defining commonly understood interfaces that will allow for modular network design and management.

When eliciting network abstractions, one needs to consider that the data plane and the control plane serve different functionalities and therefore necessitate different abstractions. In this context, challenges pertaining to the data plane are being addressed with the introduction of novel networking paradigms, such as Multi-access Edge Computing, that allow for data to be curated and processed closer to their source and consumption points. On the other hand, challenges pertaining to the control plane of the network are being addressed by paradigms, such as Software Defined Networks (SDN) and Network Function Virtualization (NFV).

SDN and NFV provide abstractions that directly address the complexity and lack of modularity of the control plane arising from the fact that it serves a variety of goals (routing, isolation, traffic engineering). Decomposing the task at hand allows defining sub-problems that can be studied independently (thus lowering complexity) and then define corresponding abstractions. Indicative examples include abstractions for a general *forwarding model* (compatibility with low level hardware and software), *network state* (make decisions on the entire network) and *simplified configuration* (computing the configuration of each physical device). In current SDN frameworks, indicative abstractions for the forwarding model are provided by OpenFlow [49] and abstractions for network state are provided by Network Operating System [50].

It is worth noting that while the use of abstractions extracts simplicity, it does not remove complexity; it allows, however, for a radically different approach in network management. For instance, configuring a network routing protocol necessitated the exact knowledge of the topology of the network, of the networking hardware and the traffic policing protocols. With the introduction of SDV/NFV, this task is now reduced to a routing protocol operating over a network graph. The corresponding algorithm takes as input the abstract topology of the network, computes the best forwarding routes which are then pushed to the SDN platform and then the network switches. While the computation is centralized logically, the actual algorithm can be run distributively and locally in network segments (thus allowing for great scalability).

4.2 Security management

Network Security can be defined as a set of processes, policies and practices that are adopted to prevent unauthorized access, misuse, modification, or denial of a computer network and network-accessible resources. Network security involves the authorization of access to data in a network, which is controlled by the network administrator [51].

Network security is usually implemented by introducing relevant devices in a network. These can be firewalls, antivirus scanning devices, content filtering devices, intrusion detection appliances and preventive devices such as penetration testing devices, vulnerability assessment appliances or anti-DDoS systems. The most common and popular security devices are firewalls, network security devices which establish a border between an internal network and the internet and manage the exchanged traffic based on some protocols. These can be either hardware or software appliances.

Lately new types of devices like UTM (Unified Threat Management) appliances show significant uptake in the IT industry. UTM appliances perform different functions; they help in averting data leaks and perform functions like gateway anti-spam, gateway antivirus protection, network load balancing, appliance reporting, network intrusion detection and prevention, URL filtering, email security, content filtering, web application firewalls, wireless security, VPN termination, continual automatic updates, DDOS mitigation, reduced compliance burden, accelerated performance etc. In the case of UTM appliances, all data is centralized and can be viewed holistically. Thus, users can get a better, real-time overview of threat detection statuses [52].

Network security management allows an administrator to manage a network consisting of physical and virtual firewalls or other network security appliances from one central location. Administrators need network security management solutions to get a high level of visibility into network behaviour, automate device configuration, enforce global policies, view firewall traffic, generate reports, and provide a single management interface for physical and virtual systems. Network security management helps reduce manual tasks and human errors by simplifying administration with security policy and workflow tools through a centralized management interface. It can reduce risk across the network and protect data by leveraging the information on threats, network vulnerabilities and their criticality, evaluating potential options to block an attack, and providing intelligence for decision support [53].

IoT networks pose additional challenges in security management. An IoT system spans from the device via different network interfaces to the cloud that hosts the platform and applications that provide services that are consumed by IoT service users. Each element of the chain must be considered when designing a proper approach to security and identity in the IoT. This requires a holistic end to end approach for the design and implementation of security management. In 2017 telecommunications vendor Ericsson issued a paper [54] that describes in high level its vision for security management in IoT networks. Figure 10 describes this approach.

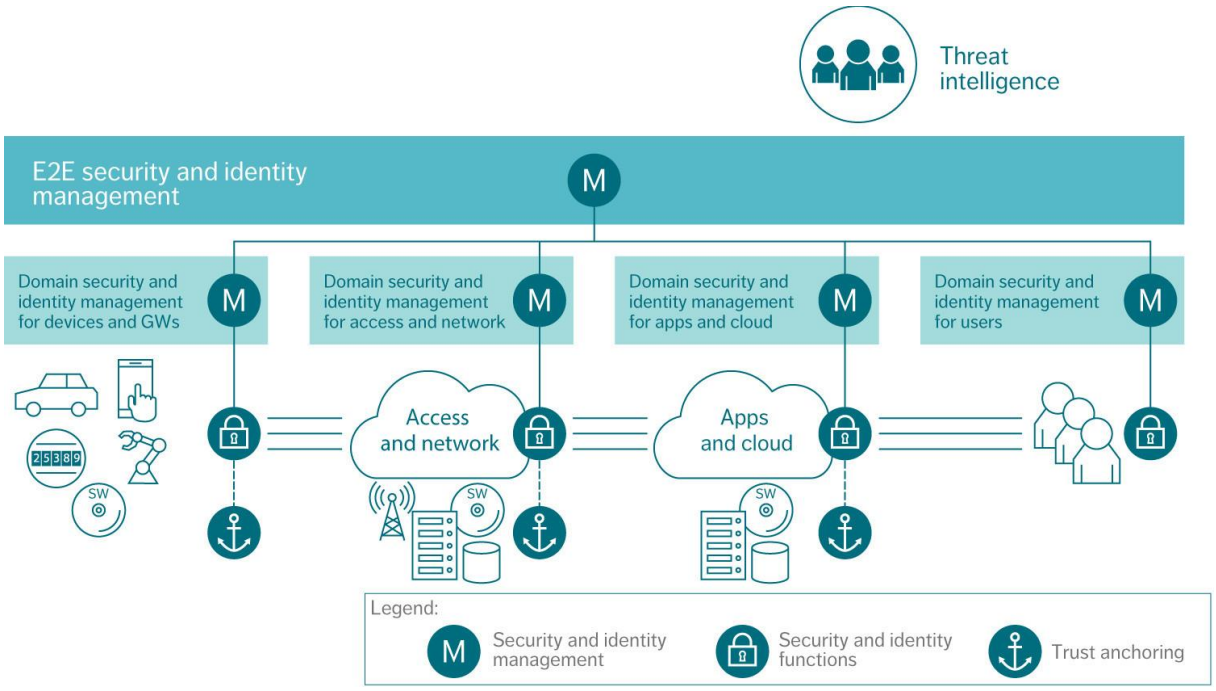

Figure 10: A high level vision for Sequrity management in IoT Networks

Key aspects in this type of security management implementation are security and identity management, security and identity functions and trust anchoring. Being a holistic approach, Ericsson suggests that all building blocks of an IoT network i.e. users, applications, cloud infrastructure, access & backbone networks and devices & gateways should be taken into consideration when designing its security architecture. Management of security functions within these domains ensures that security and identities are properly managed, configured and monitored within the domain according to policies, regulations, and agreements. The domains are managed both horizontally and vertically. Horizontal security (cross-domain) is required at connectivity and application levels. Vertical security from hardware to application can be used in every domain to provide a hardware-based root of trust, ensuring the integrity of the domain. The domains are built on trusted hardware and software. When required by the industry and the use case, trust is anchored to hardware.

With regards to SDN environments, security monitoring in the different planes is evident, as the scope for the respective plane is different. Figure 11 illustrates the scope of the threats on the three planes, data, control and application. An SDN-aware monitoring approach would need to deploy vantage points across all planes and then offer additional correlation between them.

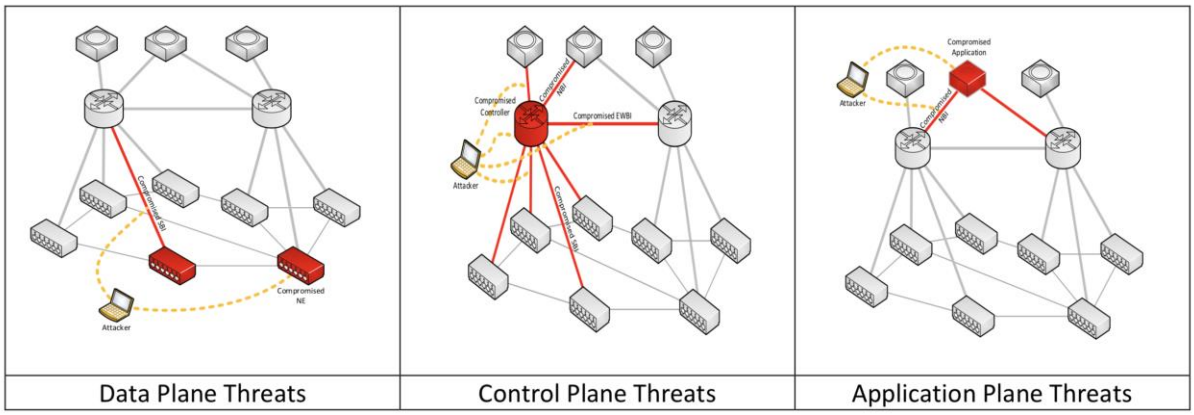

Figure 11 Overview of scope of SDN threats (source: [55])

4.3 Performance monitoring

Performance monitoring can be defined as a number of techniques that enable, manage and ensure optimal performance levels for a network. This requires for the performance and quality level for each network device and component to be continuously monitored in a granular manner.

Network performance monitoring processes help in identifying performance bottlenecks, whereas network performance management ensures that problems are mitigated and the network is restored to the required performance level. In addition to internal metrics, network performance management also reviews, analyses, maintains and manages performance from a user perspective [56].

Key factors in network performance management include:

- Network delays
- Packet losses
- Throughout
- Packet transmission
- Error rates

IoT devices will increase network congestion, and this will only grow with time. Network performance monitoring solutions are rapidly maturing alongside IoT developments. They will be able to accommodate the demands for IoT within any network. Having the solutions before the change will make it easier for any network administrator. Instead of letting congestion slow them down, network management tools will free up time usually spent searching for network issues. Network administrators should be keeping up with the technology in order to prepare for its quick implementation.

Investing in more bandwidth helps prepare for the IoT revolution. The technology is coming faster than some may realize, so having the right bandwidth ahead of time will make this transition more manageable. Network performance monitoring solutions will provide the knowledge they need to build the proper network size. Establishing the most efficient plan is only possible with appropriate network information.

5 Monitoring Architecture

The monitoring and alerting system for the IDEAL CITIES platform will comprise of monitoring of the assets (IoT and end user applications), the network components and the microservices architecture based on Kubernetes. More specifically, for the project's platform proof of concept purposes, minikube is selected, as this is a lightweight, Kubernetes single node cluster.

5.1 Microservices level monitoring

The monitoring of the minikube instance(s) will be delivered by the Prometheus open source monitoring tool [57]. As such, the data model for all data collected for monitoring purposes will be stored as time series. That is, all data will be accompanied by a timestamp when stored. Monitoring will be possible through predefined metrics. The latter requires as a minimum to define metric names and optional labels. The labels introduce data dimensionality. The IDEAL CITIES platform specification (D4.1) includes the detailed name-labels that will enable monitoring on both performance and business logic level (i.e. the circularity parameters). Naturally, a metric name is expected to have more than one labels, as expressed with the following notation:

```
<metric_name>{<label_name1>=<value>, …}
```

A reference architecture of the Prometheus monitoring framework is shown in Figure 12 below.

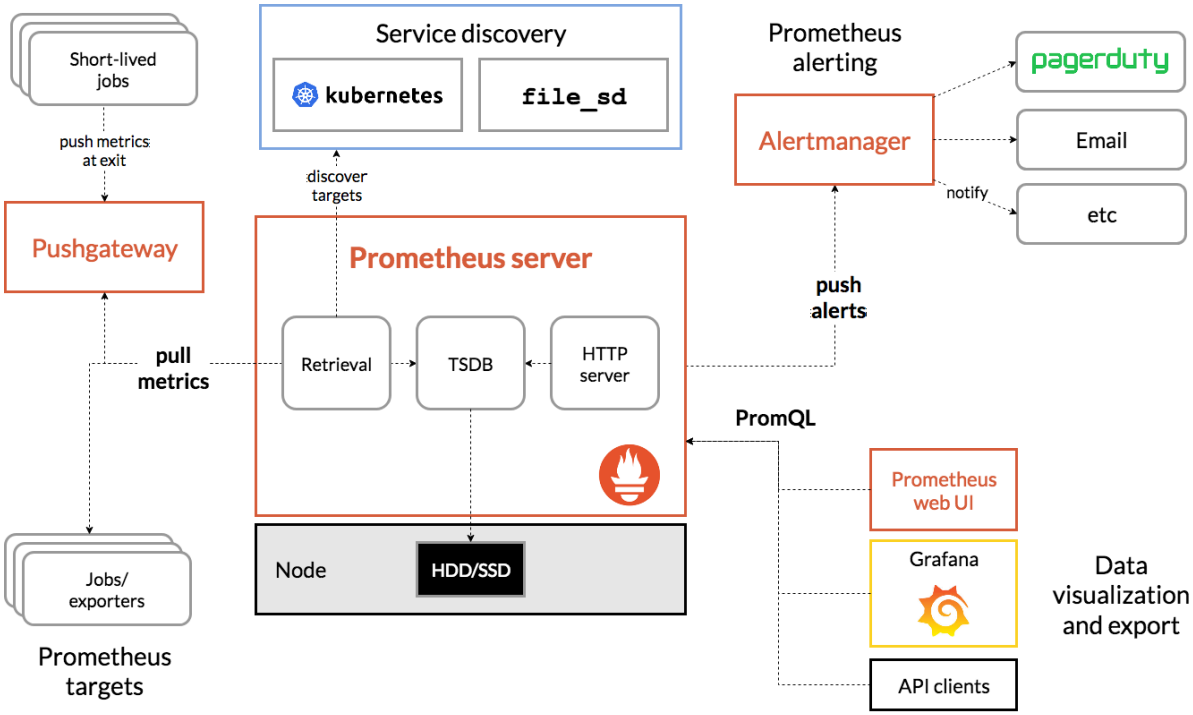

Figure 12. Example Prometheus monitoring architecture (source: [58])

5.2 Network monitoring architecture

Network data collection will be primarily performed near the minikube. Network monitoring will be delivered through the collection and analysis of network flows. It should be noted that this option is appropriate and sufficient for proof of concept approaches where there is limited

SDN deployment. In a scenario where extensive network virtualisation is delivered, more substantive monitoring will need to be considered (such as monitoring the data and control planes, dynamic configuration changes, network slicing etc.)

5.3 Application level monitoring

Monitoring at the application level is highly coupled with the monitoring of intelligent assets. In this case, monitoring data will reside in the application access logfiles and the database. Through the situational awareness dashboard UI, the two use cases (lifelogging and supporting the visually impaired users) will be realised under the fire / emergency evacuation scenario. Specifically, the two use cases combined can compose the overall fire evacuation scenario through the following aspects:

1. Lifelogging and participatory sensing: This refers to the initial registration and ongoing capturing of assets and spatiotemporal objects and events. This is needed in order to:
    a. Bootstrap the application by populating it with the initial training data and annotations
    b. enable the dynamic and adaptable system responses that accompany real-time decision making
2. Supporting the mobility and visually impaired citizens: this use case, with the help of the lifelogging functionalities, focuses on the execution of the fire evacuation scenario. For the showcased scenario the four monitored groups are as follows:
    a. Impaired users
    b. Specially skilled users such as fire wardens and first aiders
    c. All users potentially affected by the threat
    d. Evolution of threat (in this case the location and progress of fire)

As such, monitoring is part of the business logic of the individual use cases and it will be assisted by utilising the appropriate modalities that allow us to interpret human behaviour. The developed unobtrusive, adaptive interface will rely on everyday devices (the very same mobile phones of the participants) to detect behaviours and actions, to sense and understand the person's environment and extract the location of the person.

In the majority of the work conducted in behaviour understanding, the results are hard to be transferred to real life interactions, since genuine formulations and spontaneous actions characterise real life solutions. In this context, a wide variety of modalities convey valuable information about the behaviour and intentions of a user, such as behavioural like acceleration and images or physiological like heart rate. These will allow tracking the user behavior in the scenario of interest.

In the area of human-computer interaction the sensing and rendering of human actions and emotions are performed through the appropriate physiological and behavioural modalities, having as an ultimate target to facilitate the sensing and rendering of intentions and behaviours, considering different context and environmental factors. Extracting the existing patterns among the involved modalities would enable the enhancement of behaviour recognition and the overall interaction between the human and the environment or with

other humans. In this sense, the fire evacuation scenario is expected to elicit a variety of actions and emotions to its participants, allowing to capture close to real-life spontaneous reactions.

With multiple persons considered, interpersonal correlations will be examined by employing algorithms directly on the captured signals, on their feature map, or based on the corresponding probabilistic models. The conducted multi-user experiments in the framework of the two use cases would enable better studying of the synergies among the involved modalities and their association with higher level behavioural phenomena such as emotion contagion / emotion sharing. This will allow us to support the citizens, especially those that are most in need.

5.3.1 Lifelogging

We assume that inside a smart circular city environment, the communication and the interaction between citizens, IoT devices and smart cities infrastructures can be achieved in many ways. An example of these ways could be through the development and the adoption of an application that will be used by all citizens.

In a smart, circular city environment, an application is designed to be used by the citizens within this environment. The functionalities of this application are specific and include GPS, acceleration, audio and video, and user's input.

However, this application can be used when an incident takes place and it is necessary to be reported. This incident may be a terrorist attack, or an incident, like a fire or a car accident, etc. Furthermore, the application is equipped with interaction interfaces for users and with dashboards for administrators who will receive feedback from the user. In this way, the application will reform its functionality depending on the needs of users in an emergency case.

In such cases, citizens will be able to report every incident through the user input functionality of the application, when it takes place in order for the incident to be reported. Moreover, the application will provide helpful information about the incident and how citizens must handle the situation. For instance, in case of fire, the application will be able to provide details about this incident and directions to the users regarding how they must act and where they should be gathered in order to be safe.

Lifelogging is by definition a monitoring activity. A recent example of a focused scope lifelogging is the recently developed COVID19 track and trace applications that continuously, periodically or on demand register a user's location. The lifelogging data when enriched with external contextual information (through COVID19 tests in this case), offers a potentially substantial added value offering health and safety to citizens.

With regards to the fire evacuation scenario, the current state is that fire evacuation plans are static. A fire evacuation plan on a particular location is informed by fire prevention documented guidelines, a risk assessment and a risk register. The latter is a living document that needs to be kept up to date in order for the evacuation plan to be of value. At the same time, the plan is relatively static and during a fire incident there is no information (or at least in a form that can be instantly accessed) on the number of people and their actual location. A lifelogging approach would introduce a dynamic dimension through monitoring the people's

locations within a defined area and as such would provide in principle a better decision making in real-time.

To illustrate the above, an alternative real-life fire scenario was documented and related to the scope of the research, so that to theoretically describe the application functionality in a case of a real event. It was stated that, as every fire scenario requires, initial research and study must be conducted prior to the evacuation, in order to determine the infrastructure (fire routes, doors, elevators, hazardous areas, flammable areas etc), the people (number, physical condition) and the pathetic/ energetic fire meanings (detectors, extinguisher etc).

Consider the enriched floor plan as shown in Figure 13. For the registration phase the following assets are captured:

- Core infrastructures in the building e.g. data rooms (red polygons)
- Fire exit doors (green exit sign)
- Fire extinguishers (fire ext. icon).
- First aid boxes locations (green first aid icon)
- Fire routes (green arrows).
- Safety team members (orange nodes).

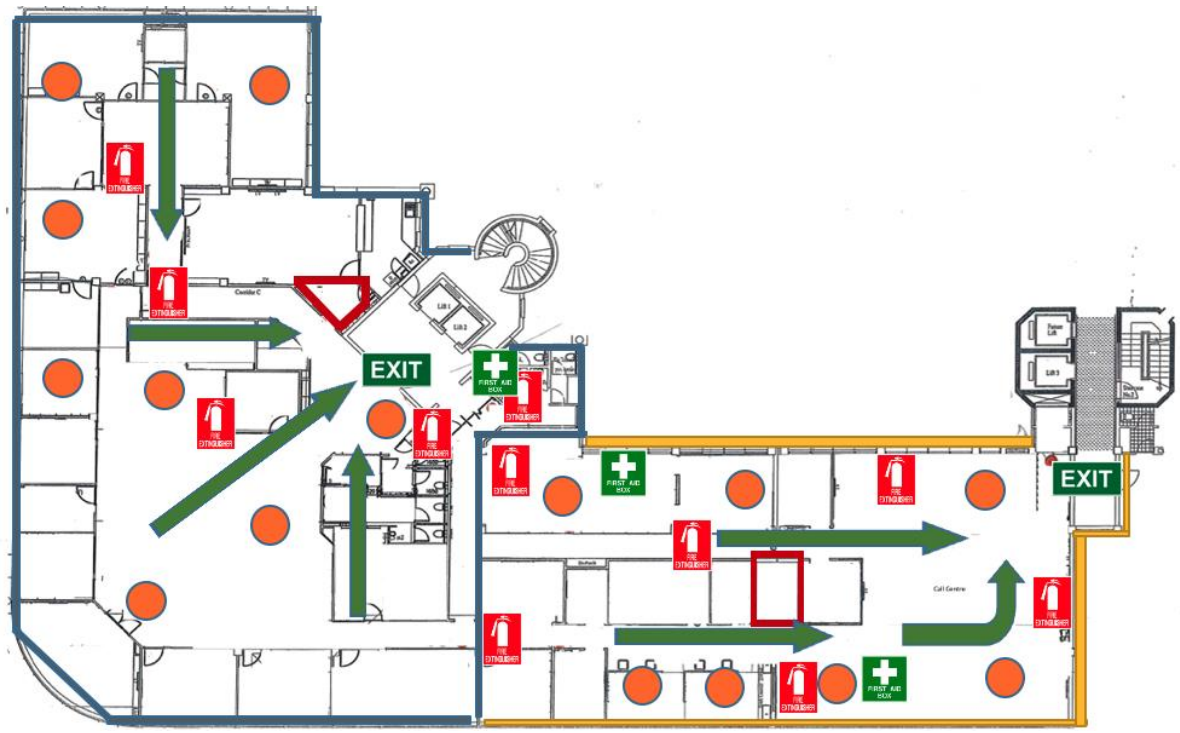

Figure 13 Fire evacuation plan

The core infrastructure and fire exit doors are fixed and static information. The fire extinguishers' properties conforming to the circularity pattern of LCA (location, condition, availability) are captured. Initially this will need to be done manually through lifelogging of a user with the appropriate role and access permissions. For IoT-enabled fire extinguishers, the

LCA can be monitored automatically and periodically. In this case any relocation or dislocation of an extinguisher would not adversely affect the opportunity to use them as the system will maintain accurately and up to date information. Moreover, the Condition property of the extinguisher will give information on the asset's potential to contribute to the extinguishing of the fire.

The green arrows designate all alternatives of the exit routes. In the event of a reported fire, some routes may/will become unavailable; this information can be reported by any user who is located on a particular path. Upon reporting the unavailability of a particular route, the system would recalculate and suggest an alternative route.

As a legal obligation of every employer, an organization must designate an evacuation team, who will guide and safely exit all occupants out of the building. In the below figure it was best described that, when a fire is identified, an ad hoc risk assessment is conducted by the fire marshals who are in a position to state the type and the risk level of the fire. In a high-risk level of fire, event reporting is then activated and fire action and the evacuation plan is implemented.

From the above it is evident that the location and time data types constitute a minimum amount of information that needs to be fundamentally captured, which can be contextualised with additional feeds.

5.3.2 Supporting the visually and mobility impaired users

In Figure 15, a fire event is reported by a person (Figure 14) (or sensor) and an evacuation procedure is triggered.

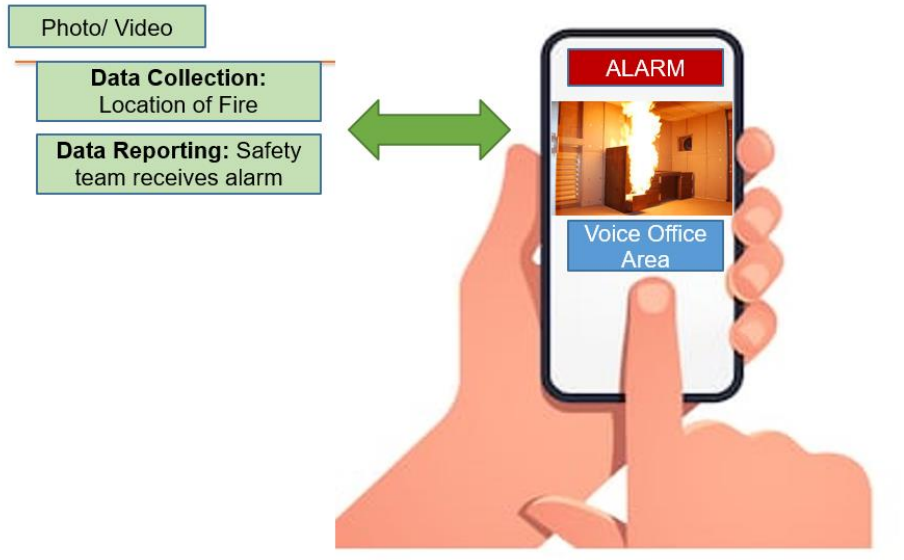

Figure 14 Manual reporting of fire

Based on the location of the incident, the closest safety team members are identified and are asked to assess the severity of the situation. For this scenario we assume that they report the firs to be a risk and this would trigger a global/building evacuation procedure. All people's locations would be identified and depending on their positions they will receive advise on the

evacuation route to follow. The continuous monitoring would allow to update and revise the evacuation route if needed, as both the fire progresses and crowd density changes.

The safety and first aid members who are not handling the fire would be located and summoned to the locations of any impaired people (disabled, pregnant, etc.). This function is particularly critical as it shows the opportunity to offer safety to more vulnerable citizens while allocating and making the best use of a resource. The whole emergency preparedness and response procedure is depicted in Figure 16.

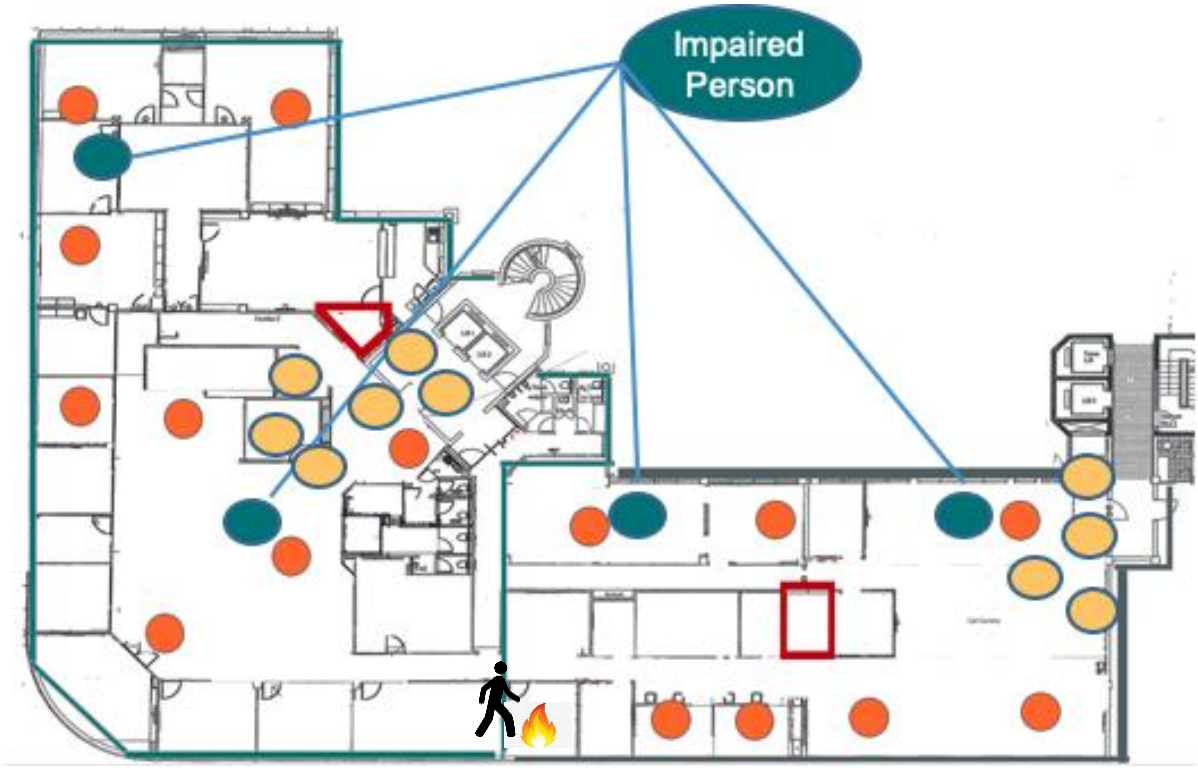

Figure 15 Execution of fire evacuation supporting impaired citizens

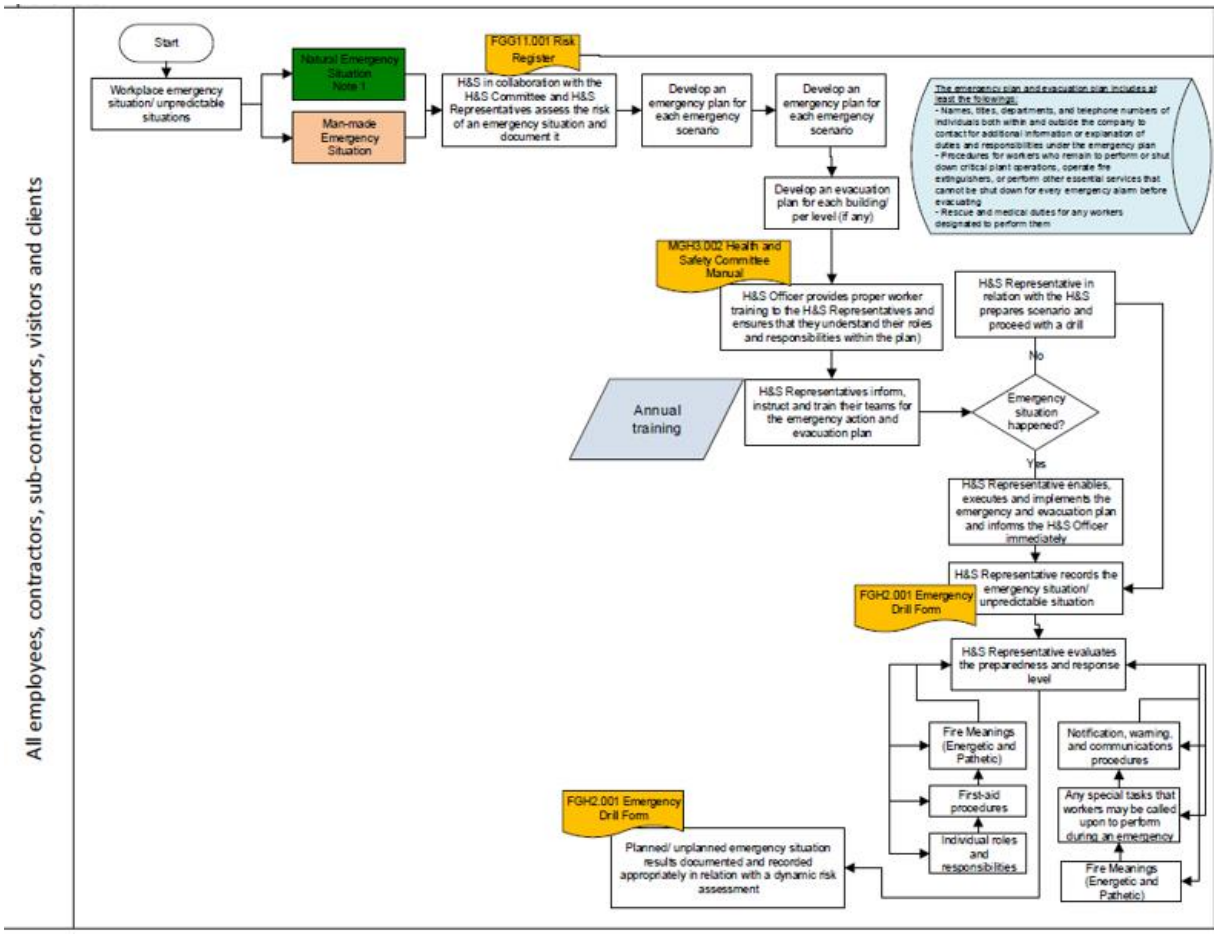

Figure 16 Emergency preparedness and response procedure.

6 Conclusions

This deliverable describes the monitoring approach for the software components and network infrastructure of the IDEAL CITIES platform. Monitoring is an integral and essential part of this project, not only from a system administrator's viewpoint, but needed in order to promote the data-driven aspects of the CE paradigm.

The deliverable initially reviewed the major approaches and theory in monitoring, including main concepts, scope and tools, across all layers (infrastructure/network, services, application and business logic level). The report concluded by recommending an appropriate architecture of the IDEAL CITIES platform and factored in the particular requirements needed to showcase the selected scenario (fire evacuation) which is selected for the evaluation of the technical objectives of the project. In addition, an analysis of reuse of the functionalities of components of the application was conducted and it was demonstrated that the application could deal with initially unplanned for and newly introduced requirements such as social distancing for indoor locations, as maybe required due to the COVID19 pandemic.

7 References

[1] DeMarco, T., 1982. Controlling software projects

[2] The anatomy of the data-driven smart sustainable city: instrumentation, datafication, computerization and related applications (Bibri, 2019)

[3] A taxonomy of grid monitoring systems (Zanikolas, Sakellariou, 2005)

[4] Design of disaster management system using IoT based interconnected network with smart city monitoring (Sakhardande et al, 2016)

[5] The cyber physical implementation of cloud manufacturing monitoring systems (Morgan, O'Donnell, 2015)

[6] A stealth monitoring mechanism for cyber-physical systems (Graveto et al, 2019)

[7] A Cyber-Physical System for Environmental Monitoring (Mois et al, 2016)

[8] Smart health monitoring systems: An overview of design and modelling (Baig, Gholamhosseini, 2013)

[9] The Sensable City: A Survey on the Deployment and Management for Smart City Monitoring (Du et al, 2019)

[10] A Network Edge Monitoring Approach for Real-Time Data Streaming Applications (Taherizadeh et al, 2016)

[11] Enhancing security and privacy in traffic-monitoring systems (Hoh et al, 2006)

[12] Opportunities and challenges in monitoring cyber-physical systems security (Bonakdarpour et al, 2018)

[13] Living on the edge: Monitoring network flows at the edge in cloud data centers (Mann et al, 2013)

[14] https://www.netapp.com/us/info/what-is-cloud-monitoring.aspx#:~:text=Cloud%20monitoring%20is%20a%20method,applications%2C%20and%20other%20cloud%20infrastructure.

[15] https://www.appdynamics.com/blog/product/application-performance-monitoring/

[16] https://dzone.com/articles/application-performance-monitoring-apm-tool-why-yo

[17] https://www.manageengine.com/network-monitoring/hardware-monitoring.html

[18] https://www.techopedia.com/definition/5445/log-file

[19] https://www.networkmanagementsoftware.com/what-is-syslog/

[20] https://nvlpubs.nist.gov/nistpubs/Legacy/SP/nistspecialpublication800-92.pdf

[21] https://www.pcisecuritystandards.org/documents/Effective-Daily-Log-Monitoring-Guidance.pdf

[22] https://www.totalhipaa.com/stay-hipaa-compliant-audit-logs/

[23] https://www.bmc.com/blogs/saas-vs-paas-vs-iaas-whats-the-difference-and-how-to-choose/

[24] https://www.solarwinds.com/-/media/solarwinds/swdcv2/licensed-products/kiwi-syslog-server/resources/datasheets/kiwi-syslog-server-datasheet.ashx?rev=df295fb836ae489eabbd21219d248986

[25] https://www.solarwinds.com/security-event-manager/use-cases/log-event-manager-software?cmp=org-blg-dns

[26] https://www.netadmintools.com/event-log-analysis-software

[27] https://www.infosecurityeurope.com/__novadocuments/254098?v=636052152332900000

[28] https://www.splunk.com/content/dam/splunk2/images/screenshots/conf19/enterprise-security/splunk-enterprise-security-posture-dashboard-overall.png

[29] https://www.loggly.com/product/

[30] https://www.elastic.co/log-monitoring

[31] https://www.fluentd.org/

[32] https://grafana.com/

[33] https://www.elastic.co/beats/

[34] https://www.elastic.co/log-monitoring

[35] https://www.klipfolio.com/resources/articles/what-is-data-dashboard

[36] https://www.researchgate.net/publication/286996830_Effective_dashboard_design

[37] EC-European Commission. 2015a. Closing the loop –An EU action plan for the Circular Economy. The Circular Economy Package Proposal, Brussels, Belgium

[38] https://arxiv.org/ftp/arxiv/papers/1901/1901.02709.pdf

[39] DOI 10.1109/MVT.2020.2991788

[40] Anatomy of Cloud Monitoring and Metering Ali Anwar, Anca Sailer, Andrzej Knochut Conference: the 6th Asia-Pacific Workshop DOI: 0.1145/2797022.2797039 https://www.researchgate.net/publication/299907309_Anatomy_of_Cloud_Monitoring_and_Metering

[41] Shi, W., Cao, J., Zhang, Q., Li, Y. and Xu, L., 2016. Edge computing: Vision and challenges. *IEEE internet of things journal*, *3*(5), pp.637-646.

[42] Brewer, E., 2010, July. A certain freedom: thoughts on the CAP theorem. In *Proceedings of the 29th ACM SIGACT-SIGOPS symposium on Principles of distributed computing* (pp. 335-335).

[43] Gessert, F., Wingerath, W., Friedrich, S. and Ritter, N., 2017. NoSQL database systems: a survey and decision guidance. *Computer Science-Research and Development*, *32*(3-4), pp.353-365.

[44] https://www.manageengine.com/network-monitoring/basics-of-network-monitoring.html

[45] https://www8.hp.com/us/en/hp-news/press-release.html?id=1744676

[46] IoT Network Management: Content and Analysis (Jonathan de Carvalho Silva, Joel J. P. C. Rodrigues, and Mario Lemes Proenc¸a Jr.)

[47] Mobile Ad-hoc Networks

[48] Vehicular Ad-hoc Networks

[49] *McKeown, Nick; et al. (April 2008). "OpenFlow: Enabling innovation in campus networks". ACM IGCOMM Computer Communication Review. doi:10.1145/1355734.1355746*

[50] Dean, Tamara (2009). "Network Operating Systems", *Network+ Guide to Networks*, 421(483)

[51] https://en.wikipedia.org/wiki/Network_security

[52] https://hackercombat.com/utm-vs-firewall-know-the-difference-to-choose-the-right-protection/

[53] https://www.juniper.net/us/en/products-services/what-is/network-security-management/

[54] https://www.ericsson.com/en/reports-and-papers/ericsson-technology-review/articles/end-to-end-security-management-for-the-iot

[55] ENISA, 2016. Threat Landscape and Good Practice Guide for Software Defined Networks/5G. https://www.enisa.europa.eu/publications/sdn-threat-landscape

[56] https://www.techopedia.com/definition/29972/network-performance-management

[57] https://prometheus.io/

[58] https://prometheus.io/docs/introduction/overview/