PROJECT DELIVERABLE REPORT

NAIADES

Greening the economy in line with the sustainable development goals

D7.6 NAIADES Security Mechanisms and Blockchain Component – Mid-term

Project Title:

A holistic water ecosystem for digitisation of urban water sector

SC5-11-2018

Digital solutions for water: linking the physical and digital world for water solutions

Document Information

Grant Agreement Number	820985	Acronym	NAIADES
Full Title	A holistic water ecosystem for digitization of urban water sector		
Topic	SC5-11-2018: Digital solutions for water: linking the physical and digital world for water solutions		
Funding scheme	and Innovation action		
Start Date	1stJUNE 2019	**Duration**	36 months
Project URL	www.NAIADES-project.eu		
EU Project Officer	Alexandre VACHER		
Project Coordinator	CENTER FOR RESEARCH AND TECHNOLOGY HELLAS - CERTH		
Deliverable	D7.6 NAIADES Security Mechanisms and Blockchain Component Mid-term		
Work Package	WP7 – Operational and Management tools		
Date of Delivery	**Contractual** M18	**Actual**	M18
Nature	R - Report	**Dissemination Level**	PU-PUBLIC
Lead Beneficiary	GT		
Responsible Authors	Kristo Klesment	Contact	kristo.klesment@guardtime.com +3725053587
	Eunah Kim	Contact	Eunah.kim@udgalliance.org
Reviewer(s):	Andreea Paunescu (SIMAVI)		
Keywords	Security, Blockchain, access control, identity management, IMS, NAIADES		

Revision History

Version	Date	Responsible	Description/Remarks/Reason for changes
0.1	06/10/2020	GT	Table of Contents
0.2	30/10/2020	GT, UDGA	Add chapter text
0.4	11/11/2020	GT, UDGA	Overall refinement
0.5	27/11/2020	SIMAVI	Internal Review
1.0	30/11/2020	GT	Review and Release
2.0	11/05/2021	GT	Applying PO suggestion

Abbreviations

KSI	Keyless Signature Infrastructure
SDK	Software Development Kit
MIDA	Machine Integrity, Defence and Awareness

1 Summary

1.1 Introduction

This task focuses on the definition, design and development of the core security mechanisms aiming to proactively ensure the secure exchange of information between NAIADES stakeholders. To guarantee confidentiality and integrity of the information transmitted, end-to-end security will be addressed across all layers of the system integrating in a seamless manner three major groups of security mechanisms: authentication, access control and transport security.

In this context, well established authentication mechanisms will be combined with a multi-stakeholder attribute-based access-control mechanism that will provide, based on a security token included within the submitted request and the evaluation of the security policies, fine-grained access control to data in the cloud, guaranteeing that only the specific group of receivers indicated by the data owner has access to the data and for a concrete purpose.

At transport layer, cryptographic mechanisms will be adapted to the water management needs and integrated in the NAIADES middleware to finally guarantee the confidentiality and integrity of data in motion.

The NAIADES architecture proposes to adapt and deploy a blockchain implementation as its central component. The architecture is adopted in order to support non-repudiation of data and decentralization of use in the NAIADES smart mobility ecosystem. In this context, the blockchain will be used to provide an audit trail of water consumers data, enabling both product data traceability and secure access for stakeholders. The work in the task will adapt a blockchain implementation considering architectural properties to record transactions between key platform modules and critical system event data. Units of sharing, event-ledger update and distribution, client model and secure access and immutable storage, and back bone replication principles are also considered.

1.2 Identity management and Access control

In the multi-stakeholder service, it is highly important to provide well established Identity management (Idm) and fine-grained access control for the submitted request to access the data in the platform. Idm controls the access of users to network, service and application. It manages secure and private authentication from users to devices, networks and services. It covers authorization and trust management, user profile management, privacy-preserving disposition of personal data, Single Sign-On (SSO) to service domains and Identity Federation towards applications.

As NAIADES aims to make FIWARE compatible data interoperability for the multiple pilots, the identity management and access control are provided by leveraging on powered by FIWARE components: Keyrock and Wilma PEP Proxy, which can be well integrated with the context data management. Keyrock is the FIWARE component responsible for Identity Management. Using Keyrock (in conjunction with PEP Proxy) enables to add OAuth2-based authentication and authorization security to your services and applications. The Wilma PEP Proxy in combination with Identity Management and Authorization PDP GEs, adds authentication and authorization security to the backend applications. Thus, only the allowed users will be able to access the network, data and services. The PEP Proxy allows to programmatically manage specific permissions and policies to NAIADES resources allowing different access levels to the multi-stakeholders.

The backend of Idm will generate an authentication token, at login, for NAIADES users, based on row credentials as username and password. It is provided as a container (dockerized) and can be run on any environment and can be used and accessed both from inside and outside NAIADES cloud platform.

1.3 A blockchain-based auditing mechanism

As part of the objective of realizing a holistic security and privacy toolkit for smart water management, a novel blockchain-backed architecture is to be designed that provides anti-tamper and early warning protection for critical system events and feeds. By registering events into the KSI Blockchain in real-time, an immutable, irrefutable audit trail will be created that can be independently verified at any point in the future. The solution will preserve the integrity of cloud platform devices and data traffic events with guaranteed immutability, which in turn, enables the detection of changes and promotes situational awareness through generated alerts.

Guardtime will utilise its KSI Blockchain technology to develop an innovative blockchain-based auditing mechanism for the project. The technology will be used primarily to sign and record events relating to water measurement and the transfer of data between key NAIADES' modules. In addition, Guardtime's MIDA technology will be used to provide auxiliary protection against cyber threats to cloud-platform-based devices, including those of the blockchain-based auditing mechanism.

The combined contributions of the technologies offer:

- Immutable state-captured data from distributed or cloud environments.
- Associated data-source identifiers, which allow data to be traceable to signers.
- Cryptographically time-stamped data (essential for auditing).
- Ensured long-term data integrity, measured in tens of years - even in light of the advances in quantum computing.
- Exportable, signed data-containers that are transferable without key management efforts.
- Data-containers suitable for audit and data forensics.
- Demonstrable and verifiable integrities of processes, such as data capture, transfer, security analysis and storage.
- Reduced effects of system key leaks; these should not have catastrophic effects on the system. Keyless infrastructure does not have PKI related risks, where a single key leak could compromise thousands of users.

Guardtime intends to design and implement a blockchain-based auditing mechanism using advanced cryptographic techniques with proprietary blockchain integration. The mechanism provides the capabilities to verify the integrities of recorded system events (relating to water measurement and system data traffic) against the distributed ledger of the KSI Blockchain. These guarantees are available over the long-term, where events can be independently verified many years later. Data consistency is proven mathematically against the KSI Blockchain (which is anchored in widely-witnessed newspaper publications) using standard cryptographic techniques. Moreover, the immutability of data enables detection of changes and promotes situational awareness through generated alerts – one of MIDA's numerous roles in the project. This means that even an insider with full access to all systems is unable to cover their tracks once the recorded events are signed.

Enterprise-grade data immutability is provided through registering events in sequence into the KSI Blockchain, enabling support for compliance across multiple regulatory standards. It is possible to detect and alert system administrators upon alteration, corruption or deletion of any underlying system data. The fingerprint of every signed event is registered into the KSI Blockchain through the use of cryptographic hashes and Merkle trees, thus providing a mathematically provable, long-term trust-anchor for all event data, without the need for key storage or maintenance of shared secrets (a major limitation of PKI and related encryption methods). Stored system events are continuously verified against the blockchain, based on risk policy, where changes to both recent and historical data can trigger system alerts.

List of definitions **Term**	**Definition**
Hash functions	Several KSI services make use of cryptographic one-way hash functions (such as SHA-256) to transform documents of any type and size into a non-reversible, fixed-size hash.
Aggregation	The aggregation process takes the individual hashes of the documents from all its users and creates a global, binary hash-tree. The input hashes are used to build the hash tree (also known as a Merkle hash tree), where each parent node is the hash of its two child hashes concatenated together. The process of constructing the tree is called aggregation.
Root hash	The final hash produced by the aggregation process (or the construction of the global, binary hash-tree) from the individual document hashes of all its users.
Hash chains	An extended KSI signature contains two types of hash chain: the aggregation hash chain and the calendar hash chain. The aggregation hash tree is formed every second but destroyed as soon as all users have received hash chains from their input hash to the global root hash; this root hash is registered in the global calendar blockchain. The calendar hash chain is used to provide a link to the trust anchor. Once extended to publication, the KSI signature contains both types of hash chain. In order to prove that a particular document (its hash) participated in the global hash tree, the entire tree is not needed - the root hash value can be recalculated from any particular input hash using the corresponding hash chain. Signature verification is essentially the process of checking the signature's hash chains.
Calendar blockchain	The calendar blockchain is a special hash tree containing an important time element. The root hash provided by the aggregation process is registered in the global calendar blockchain every second; the calendar blockchain is perpetual, and data is only appended to it. The calendar blockchain produces data-hashes suitable for publication and use as trust-anchors.

2 Overview of NAIADES Security mechanism

2.1 Context and scope

Figure 1 provides an overview of the architectural modules and layers which comprise the NAIADES platform. The grey boxes in the Figure indicate the locations of the security components which will be used to ensure platform protection. Additionally, interactions between the security components, and the modules and layers they are tasked to secure, are also illustrated.

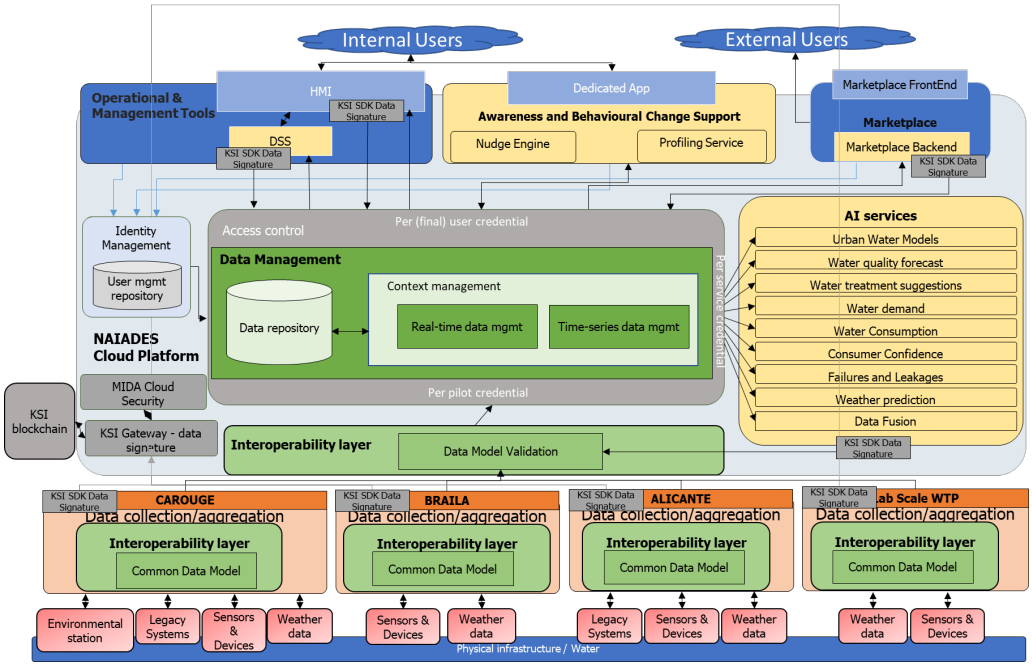

Figure 1. NAIADES architecture modules and layers overview.

NAIADES' architecture is designed to be modular and scalable. Each module - or building block of the NAIADES architecture - provides a specific functionality and can be connected to other modules to change the operational characteristics of the system. Hence, depending on your use-case, certain modules may be replaced by other modules (to improve the platform's performance, for example), and new modules may be added to provide new services for NAIADES users (perhaps increasing the platform's versatility, for example). Despite this, as new iterations of the NAIADES' architecture are made, it is important to adhere to the following design principle when adding new modules (or blocks): new blocks should be adapted to the platform's formats, communication methods and security. In relation to this Deliverable, it is important to ensure a new module's compatibility with the overarching security mechanisms described in this document.

The security layer affects all the other layers of the architecture. Its main purpose is focused on data, users and system protection. It ensures secure exchange of information, guaranteeing end-to-end confidentiality and integrity by implementing authentication, access control and transport security. A well-established authentication mechanism is combined with a multi-stakeholder, attribute-based access control mechanism that will provide, based on a security token included within the submitted request, and the evaluation of the security policies, fine-grained access control to data in the cloud, guaranteeing that only a specific group of receivers (indicated by the data owner) has access to the data for a concrete purpose.

As part of the architecture's overall security mechanisms, KSI Blockchain will be used to provide an audit trail of water consumers' data, enabling both data traceability and secure access for stakeholders. Any data which is signed using KSI signatures will be shared between the data generators and the platform's administrative component - a form of KSI signature manager - in charge of blockchain-based data integrity

checking. The latter, a platform management layer, will be placed in charge of platform health and will monitor platform infrastructure to detect problems that hinder correct operation. The management layer will also use functionalities from the security layer, owing to the incorporation of MIDA, to add security information to the management of the platform and be able to trigger alarms when any type of problem arises.

Such an adaptation (and deployment) of this blockchain-based architecture will include consideration of the architectural properties of data transactions and event data, information sharing, record update and distribution, a client model for secure access and immutable storage, and back-bone replication principles.

In summary, a number of modified KSI Software Development Kits (SDKs) will be integrated with the NAIADES platform, as close to water measurement data sources as possible. These locations will help ensure the integrity of measurement data by securing it as quickly as possible. Furthermore, the devices on which SDKs are present will have access to KSI gateway. KSI gateway provides access to the signing and extending services of the KSI service network. The signing and extending services each provide integral functions for the long-term security of the water measurement data. KSI gateway, in turn, will have access to the KSI Blockchain. The KSI Blockchain provides the distributed ledger for independently-verifiable proofs of data integrity, signing time and signing entity.

In addition, KSI SDKs will be located at the outputs of several key modules of the NAIADES architecture. The SDKs can be used to secure any resulting output data, to provide a basis for maintaining the integrity of data traffic events. Such data may also be used for future audits (of a nature which depends on the modules concerned and the types of output data).

Moreover, as the NAIADES platform is heavily reliant on safe cloud use, MIDA will be deployed to monitor and secure the platform's key cloud resources. MIDA provides real-time and provable awareness to detect changes across cloud infrastructure. This reduces the time-to-detection of misconfigurations, unauthorized access and deployment of system assets. When combined with access to the KSI Blockchain (via KSI gateway) MIDA also contributes to the system's application as a governance and audit toolkit, to which it contributes several useful event-correlation and analysis features. It is proposed to deploy the majority of MIDA's sub-components - which are each explained in detail later in this Deliverable - on the same cloud platform as the NAIADES modules they are tasked to protect.

NAIADES authentication and access control mechanisms provide a specific set of security features such as identity management, access control, authentication, and authorization. It is based on the OAuth2 standard protocol that will delegate and generate access token and protect NAIADES from unauthorized access. The identity and authentication management component will issue different roles and permissions to each user and application, where the access tokens can be used to grant or deny access to the APIs exposed by the platform components based on provided roles and permissions.

The security mechanisms described above will be developed to provide strong data integrity guarantees, making the critical event data and feeds of the architecture immutable, and preventing changes to the architecture without detection. Such work is to be reported in this document. The Blockchain Auditing Mechanism produced will be a product & service licensed by Guardtime Ltd for stand-alone or cloud-based solutions.

2.2 Security requirements

Relevant guidance has been adapted from ENISA's *Appropriate security measures for smart grids: Guidelines to assess the sophistication of security measures implementation* document. The document describes a set of security measures considered appropriate for implementations of smart grids; however, these considerations can be reasonably applied to smart water projects, also. Furthermore, as data privacy issues are not covered in the ENISA document, considerations on the topic of security and privacy of data are provided in the *Security*

and privacy of data section of this document. A set of minimum security measures are outlined to improve the minimum level of cyber security services for NAIADES-like projects.

The parts of the guidance addressed by the security mechanisms detailed in this Deliverable cover the following areas:

- Secure lifecycle process for smart water project components/systems and operating procedures - this includes activities and procedures related to the secure operation, configuration, maintenance, and disposal of the smart water project components and systems. It is suggested that the security provider:
    - Identifies and defines beforehand the necessary security requirements for smart water project components and systems during the design and procurement - to address this, security requirements have been documented in this (and previous) texts, and secure coding practices have been established to reduce common security errors.
    - Ensures that the base security configuration of a smart water project's components/systems is identified, set and maintained for every instance of that component/system - in accordance with this, information on the restricted use of functions and the permitted ports, protocols and/or services is provided in this Deliverable.
    - Establishes and maintains activities for software/firmware upgrade on the components and smart water project's information systems - for this, it has been stated that the platform operator will be responsible for the implementation, and updating, of the security tools required for the project; in this sense, the security tools are the software of the KSI Blockchain and MIDA technologies.
    - Performs security testing activities on the smart water project's components/systems to verify its security - to address this, rigorous security testing is planned for a later stage of the NAIADES project (listed as a future step in the *Conclusions* section of this Deliverable).
- Audit and accountability - this domain recommends the implementation of an audit and accountability policy and associated controls in order to verify compliance with specific legal requirements and organisation policies; in particular, the security provider may:
    - Establish and maintain an audit and accountability process that enables sufficient logging capabilities in the smart water project's systems and components and provides valuable log data for analysis - two technologies (KSI Blockchain and MIDA) are implemented to provide logging capabilities for future analysis of key system events. These records contain valuable timing information, component IDs, entity information, system state information and additional metadata.
    - Establish and maintain monitoring activities on the smart water project's information systems and components - indeed, one of MIDA's primary roles is that of monitoring cloud platform activities and events; MIDA also has the capability to monitor physical system devices.
    - Protect the audit information generated - to address this, audit information will be safeguarded through the implementation of KSI signatures and a KSI Blockchain-based architecture, designed to uphold the integrity of key system events.
- Information systems security - this domain covers the definition of requirements to protect the information managed by the smart water project's information systems using different technologies like firewalls, antivirus, intrusion detection and etc.; the security provider should:
    - Implement security requirements in order to protect the information on the smart water project's information system - with respect to the technologies of this Deliverable, use of firewalls/iptables and secure networks is recommended when integrating KSI gateways,

and, furthermore, use of MIDA provides the necessary protection for key cloud platform devices, on which NAIADES is reliant.
    - Establish and maintain system/groups/user accounts on smart water project information systems - such information (on account types, access rights and privileges) is given in the *Identity and Access Management* section of this Deliverable.
    - Enforce logical access to authorized entities on smart water project information systems and security perimeters - methods are documented in relation to authentication type and authorization schema in this Deliverable. These precautions are taken in relation to the Identity and access management aspects of the project.
    - Establish and maintain secure remote access where applicable to smart water project information systems - in response to this, methods of remote access, authentication and encryption methods are detailed in this Deliverable.
- Network security - this domain highlights the design and implementation of requirements that protect the established communication channels among the smart water project information system and the segmentation between business and industrial networks; in particular, it is recommended to:
    - Establish and maintain a segregated network for the smart water project information system - due to the layered and modular design of the NAIADES platform, recommendations for the establishment of clear trust boundaries (which reflect its design) are provided in this Deliverable.
    - Establish and maintain secure communications across the segregated network - in answer to this, several communication standards and protocols are listed as technological requirements for the components in this Deliverable.

The remaining aspects of security governance & risk management; management of third parties, personnel security, awareness, and training; incident response & information knowledge sharing; continuity of operations; and physical security are not within the scope of the planned security mechanisms.

Furthermore, it should be noted that the adoption of many of the proposed security measures requires the consensus and cooperation of various stakeholders of the NAIADES ecosystem. This may require significant coordinated efforts to ensure the correct degree of acceptance and compliance with the above guidance.

2.3 Technology requirements (of KSI and MIDA)

2.3.1 Functional

Functional requirements for individual technologies are listed below.

KSI gateway:

- Gateway server (KSI gateway block) - a software component at the customer premises providing access to the Aggregation or Extender Service. A dedicated gateway or cluster of gateways is usually installed for an organization to provide the service to the users of that organization.
- The way KSI gateway is deployed in terms of network zones depends on the network architecture of the user organization and how the applications are going to use it for signing. A classic example is that an organization has a KSI gateway deployed in their network and it provides access to KSI services for the users of that organization.
- The server is not accessible by third parties without access control mechanisms.
- The gateway should be behind a firewall or have local iptables/firewall configured to allow traffic only from authorized IP addresses.

- The gateway should be accessible to applications that sign data and extend signatures. Corporate deployments usually expect access to the gateway through a secure network.
- Gateway must have access to upstream aggregators and extenders in the KSI service network. This communication is not required to be over secure networks.

Digital signatures:

- Application Integration (Digital signature block) - Guardtime provides fully featured SDK-s for C, Java, GO, JavaScript and .NET to facilitate KSI service integration to customer applications.

MIDA:

- Requires access to the gateway server to sign, extend and verify the KSI signatures of data dockets.
- Sub-components can reside together in a single cloud instance, each in its own instance, or any combination in-between.
- Several sub-components require the configuration of inbound listening ports for HTTP connections from client applications.
- Several sub-components require the configuration of outbound access to, for example:
    - KSI gateway (protocol: `TCP`; auth: `HMAC`).
    - Configuration Server (protocol: `TCP`; auth: `HTTP signature`).
    - Other SMS components (protocol: `TCP`; auth: `HMAC`).
    - The cloud (connections over HTTP and HTTP, typically).
    - A PostgreSQL database instance (protocol: `TCP`; auth: `md5`).
    - An SMTP server (default port: `25`; protocol: `SMTP`; auth: `SMTP AUTH`).
    - Token Provider Server (protocol: `TCP`; auth: `JWT`).
    - An LDAP server (default port: `389/636`; protocol: `LDAP/LDAPS`).
- If KSI gateway credentials are retrieved using Provisioning Service, outbound access to Provisioning Service is required (protocol: `TCP`; auth: `HMAC`).

2.3.2 Non-functional

Non-functional requirements for individual technologies are listed below.

Digital signatures and KSI gateway:

- Reliability, availability, maintainability: 24/7
- Performance: up to 1 000 000/1 sec.
- Scalability: depends on conditions.
- Usability: not critical, end user will not access this module
- Portability and compatibility: KSI SDK is provided for C, Java, GO, JavaScript and .NET to facilitate KSI service integration to applications.

MIDA:

- Reliability, availability, maintainability: 24/7.
- Performance: 1 000 000/1 sec.

- Scalability: depends on conditions, solution should scale with platform growth.
- Usability: not critical, end user will not access this module. Should be user friendly when configuring monitoring and alerting; and to give a complete overview of the cloud state.
- Portability and compatibility: monitoring services for cloud, and agents for Linux, MacOS and Windows operating systems.
- Security: server is not accessible by third parties without access control mechanisms.

2.3.3 Software

Software requirements for individual technologies are listed below.

KSI gateway server:

- Operating systems: RHEL/CentOS 7 for KSI gateway.
- Server: virtual or physical.

Data signatures:

- For application integration the following SDKs are available: C, java, GO, JS, .NET.

MIDA:

- Operating systems: CentOS/RHEL7.1 or later; Amazon Linux 2.
- Programming language: Java 8, update 161 or later.
- Server: virtual or physical.
- Database: PostgreSQL 11 or later AND Redis 5.0 or later.

2.3.4 Hardware

Hardware requirements for individual technologies are listed below.

KSI gateway:

- CPU: 2 cores*
- GPU: -
- RAM: 2GB*
- HDD: 40GB

(* Minimum requirements)

MIDA:

- CPU: 1 core*
- GPU: -
- RAM: 1GB*
- HDD: 1GB*; 1+TB for Database

(* If running all components on separate machines.)

3 NAIADES Security mechanisms

Detailed specifications and explanations of the key constituents of the system of NAIADES Security mechanisms are provided in this section. This includes KSI Blockchain, KSI infrastructure, KSI signatures and MIDA, and FIWARE based Identity management and Access control.

3.1 KSI Blockchain

3.1.1 Overview

Guardtime's KSI Blockchain is both a method and a globally distributed network infrastructure for the issuance and verification of KSI signatures. Its design overcomes two major weaknesses of traditional blockchains, namely their scalability and commitment time; KSI Blockchain-based systems are usable for large applications, at industrial scale.

Normally, a blockchain is a distributed public record of events - an append-only ledger where each new event is cryptographically linked to the previous one. For typical blockchains, a distributed consensus protocol is used to add new entries to the blockchain. This protocol ensures that all participants agree on a unified public record of events, without the input of a central authority. In contrast, KSI Blockchain is a permissioned scheme which relies on a proprietary consensus protocol. This protocol validates the blocks of a blockchain using only approved accounts, provided with authority by a central source; the process is automated, and it simply requires that validators' computer systems (termed nodes of authority in the Proof of Authority protocol) remain secure and uncompromised at all times. As such, to uphold the validity of the blockchain, the core nodes of the KSI service network (which compile the record of events for the blockchain) are strongly authenticated.

KSI Blockchain provides the following properties:

- Data integrity – a KSI signature links input data to a verifiable, distributed trust-anchor (a widely-witnessed event, published in a newspaper or via electronic media) using a one-way hash chain. In doing so, the integrity of the data can be verified by comparing the output hash of the hash chain (in the signature) with the hash of the distributed trust anchor (typically provided as a publication).
- Signing time – a KSI signature provides strong proof of signing time. Due to the design of the system's aggregators (which construct a global hash tree from the system's input data hashes every second) signing time is encoded into the "shape" of the resulting blockchain ledger. In other words, a hash chain is produced by a set of hierarchical aggregators to which an important time element is added; this is known as the calendar hash chain, and it enables the derivation of signing time for the original data.
- Signing entity – a KSI signature provides attestation of origin. When the system aggregates a set of input data hashes, as part of the formation of a new node (or block) for the blockchain, the identities of all participants (from customer applications to Blockchain core servers) are tied to it. As such, these identities are also embedded in the signature. The chain of identities is forward secure, meaning it cannot be modified after signature creation. In addition, the nature of the method means that signatures do not provide non-repudiation (see the Non-repudiation section for more information).

3.1.2 Role of KSI Blockchain

KSI Blockchain will be used to link input data to widely-witnessed evidence in the form of newspaper publications, or publications via online media. Thus, the role of KSI Blockchain is to provide a distributed ledger for independently-verifiable proofs of integrity, signing time and signing entity for NAIADES platform data - specifically, data relating to water measurements and the transfer of data between key NAIADES' modules. The process also guarantees end-user privacy; as, to ensure the integrity of

NAIADES platform source data, KSI Blockchain relies on the general use of cryptographic hash functions - no keys are involved in the process, and no systems need to be trusted for signature verification.

KSI Blockchain provides high-availability and scalability to ensure that any number of signatures can be requested worldwide, and the time to get the signature is only approximately one to two seconds. The characteristics of elasticity and scalability are essential to the operation of the NAIADES platform when dealing with large amounts of data, for example.

Lastly, as its design is both secure and decentralized, the inclusion of a KSI Blockchain-based technology enables water service providers to guarantee the state of the data exchange network without relying on trusted administrators or the procedures that define the security of the network.

3.1.3 Non-repudiation

KSI signatures do not provide non-repudiation. Lack of this characteristic means that the KSI Blockchain system (alone) cannot indisputably prove the actual entity (e.g. a person or a device) who requests the signature. For every use-case, it is necessary to supplement KSI Blockchain-based systems with an adequate set of legal framework, procedures, system engineering and application design, which may include additional cryptographic engineering to achieve the desired strength and granularity of the signer's identity.

It should be noted that the keys used for authentication within KSI infrastructure are symmetrical, meaning that a parent entity can, in theory, control the use of, and impersonate, its child entities. Hence, the meaning, strength and legal status of a signing entity must be explicitly agreed for each use-case. As a rule-of-thumb, using KSI Blockchain to manage the identities of IoT devices within a corporate network is feasible, whereas, identifying users in a public environment, without additional effort or precautions – if this is indeed necessary for a project - is not.

3.1.4 KSI infrastructure

KSI Blockchain is built and accessed through KSI infrastructure alone (i.e. it is not connected with other blockchain systems). This infrastructure is designed to be layered and hierarchical, a summary of which is available in Figure 1.1.

The key components are:

- A user application – controls the SDK and is responsible for initiating signing, verifying, and extending requests.
- KSI gateway server - the lowest layer of the hierarchy of Aggregators and Extenders, featuring a customer-facing Aggregator and Extender, packaged into a KSI gateway server. The gateway layer has the most servers, and the number of servers typically decreases with every higher level.
- Hierarchical aggregator servers – the global aggregation tree is built using hierarchically organized aggregators. A given aggregator provides the hash-chain from the tree it produces. These individual hash-chains, when linked together, form the full chain from the document hash to the root of the global hash tree. Higher aggregators are responsible for linking hash-chain fragments. The design optimizes the storage capabilities of the system.
- The core cluster – registers the root hash of the global aggregation tree in the calendar blockchain every second, thus constantly growing the blockchain. The Core consists of several independent servers that agree on the hash value to be added to the calendar blockchain. Periodically, the root hash of the calendar blockchain is published in electronic media and newspapers. The core is also responsible for producing PKI signatures for use in short-term signature verification.
- Extender servers - the calendar blockchain is distributed downstream using Extender servers. This allows end users, at the gateway-level, to use more rigorous trust anchors when verifying signatures.

The KSI Core, Aggregation and Extender networks are operated as a permissioned blockchain by Guardtime; however, branches of aggregation and distribution hierarchies can be operated by third parties.

In this project, it is proposed that the KSI infrastructure components related to aspects of operation, access and configuration are handled by the NAIADES cloud operator (SIMAVI); this includes the provision of user access to relevant preconfigured endpoints. It is Guardtime's responsibility to provide manuals, information and additional support to the operator as needed. In addition, the gateway layer of the network will be hosted on-premises, for the best possible security and service quality.

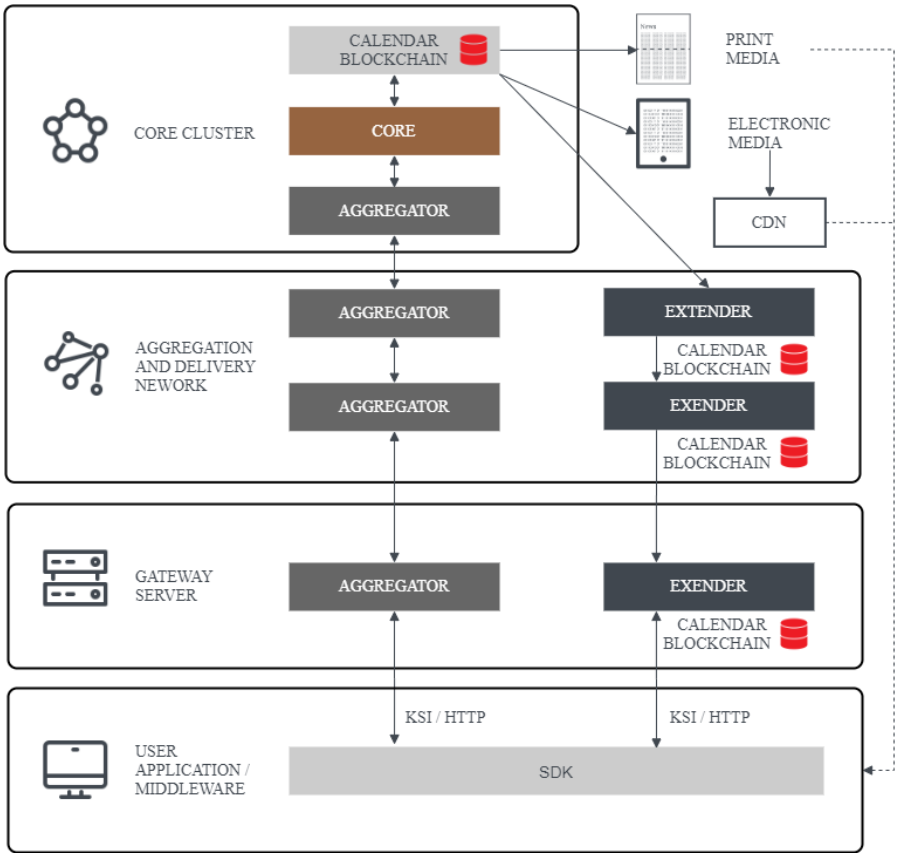

Figure 1.1. KSI infrastructure diagram.

3.1.5 KSI Blockchain limitations

The following is a list of the limitations of KSI Blockchain:

- Data encryption – the aggregation process makes use of cryptographic one-way hash functions (such as SHA-256) to transform data of any type and size into a non-reversible fixed size hash value; however, this is not the same as providing data encryption: the data on which the signing process operates is not, itself, encrypted.
- Validation of data/content - a KSI signature does not provide validation of business rules or distinctions between "good" and "bad" data. It operates on all data regardless of the content.
- Non-Repudiation – this means that the identity of an entity requesting KSI signatures cannot be proven indisputably (using KSI Blockchain technology alone); see the Non-repudiation section for more information.
- Prevention of double-spending - there is no verification mechanism for preventing the issue of double-spending. Double-spending is a potential flaw in digital cash schemes in which the same single digital token can be spent more than once. Unlike physical cash, a digital token consists of a digital file that can be duplicated or falsified.

- Data transport / storage - integrity checking functions are typically performed before and after data transit, and before and after the data storage process. The system does not provide secure channels for data transit or secure options for data storage.
- Integrity checks and audits – these will be carried out by third parties (for example, by an internal controller of the customer's, or an auditor).

3.2 KSI signatures

3.2.1 Overview

KSI signatures and their associated infrastructure rely on the sole use of cryptographic hash-functions for the verification of data integrity. When combined with the availability of a public ledger, KSI signatures can be used to validate the end-nodes of a blockchain which together constitute the final ledger's data. Such a model provides capabilities for the cryptographic proof of signing time, data integrity, and attribution of data origin for a blockchain.

KSI services are accessible via a KSI gateway. Each KSI gateway has two endpoints, to allow connection to: 1) the aggregation service, for the aggregation of data hashes to produce KSI signatures, or 2) the extender service, to provide access to a trust anchor for the extension of pre-existing signatures. In addition, during the verification process the extender service may be utilized to fetch part of the calendar blockchain to verify the signature; the signature is not permanently extended, however.

KSI SDK integration, by performing the necessary computations and comparisons, allows the user to verify the integrity of KSI signatures in an automatic or manual fashion depending on use-case requirements. The configuration of a suitable trust anchor and verification policy also depends on use-case requirements.

The following sections discuss the signing, extending and verification procedures which underpin the use of KSI signatures.

3.2.2 Role of KSI signatures

The NAIADES platform utilizes KSI signatures as a means of providing independently-verifiable proofs of data integrity, signing time and signing entity. As such, these properties allow for the creation of a secure audit trail that enables both data traceability and secure access for stakeholders.

As data is signed, corresponding digital fingerprints are created that can be stored as attributes alongside the input data, or placed in separate storage. As KSI signatures do not provide non-repudiation of users' data, the auditing mechanism realized by the project will help ensure the principle of data privacy.

To generate a signature, a user interacts with a KSI gateway by submitting a hash-value of the data to be signed. And, once the data is aggregated, a KSI signature for the original data is returned to the user.

3.2.3 Signing data

KSI signatures make use of cryptographic one-way hash functions (such as SHA-256) to transform data of any type and size into a non-reversible fixed-size hash value. In this form, the hash value acts as a digital fingerprint for the original data, with a small footprint of only a few KB; it is feasible to store the signature with its associated data as an attribute, or separately in different types of data storage.

To generate a signature, a user interacts with a KSI gateway by submitting a hash-value of the data to be signed. Once the data is aggregated, a KSI signature is returned to the user. The process is depicted in Figure 2 below. KSI signatures are server-based, meaning that signing is only possible with online access to the KSI service network.

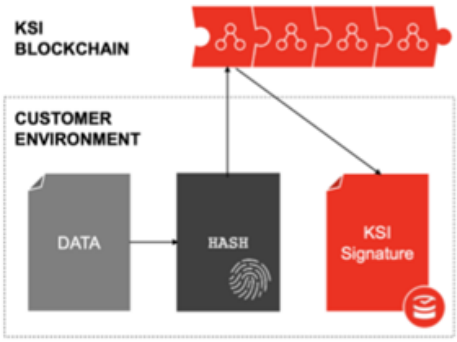

Figure 2. Schematic of the signing process.

KSI gateways also aggregate identity metadata into a signature's hash chain (in particular, the client ID received from the signing request). Once the client ID becomes part of the hash chain this information cannot be modified or removed from the signature.

Client-side aggregation is a feature of the KSI SDK, which allows you to provide multiple documents (hashes) for signing in one go. This means that only a single signing request is sent from the SDK to the gateway. Despite this, the SDK still provides the user with an individual KSI signature for each document. The benefits of client-side aggregation include exposing less information about a specific document to the gateway and reducing the number of signing requests sent over the network.

3.2.4 Extending signatures

Signatures are extended with the help of the extender service running in the KSI gateway. KSI signatures should be extended to allow them to be verified using the strongest possible trust anchor: a publication. Signatures extended to a publication can be verified over an indefinite period of time; whereas, it may only be possible to verify un-extended signatures up to 5-year-old after initial signing. The extension process typically combines the hash-chain of the original signature with the hash of the distributed trust anchor (found in a newly published newspaper, for example). The extender service requests the published hash from the KSI service network when the user submits a signature for extension - although the signature never actually leaves the user's device; the extenders of the network hold an up-to-date copy of the published hash. The published hash replaces the key-based authentication record of the original signature, making the signature truly keyless and the mathematical proofs of integrity and non-backdatability of the signed data applicable. The prerequisite for extension is that at least one publication has been issued after the time of signing.

To extend signatures, a user initiates the extension process using the KSI SDK. Indeed, the process is not carried out automatically. However, in the specific case where a user wishes to verify un-extended signatures against a publication (which requires the extension of signatures prior to verification), it is possible to perform temporary signature extension on-the-fly, if access to the extender service (which possibly holds an up-to-date copy of the published hash) is available.

3.2.5 Verifying signatures

Signature verification checks if the KSI signature's hash chain is trusted in accordance with an available trust anchor. This process establishes whether the integrity of the signature has been compromised. Specifically, a user initiates the verification process using the KSI SDK; the user provides data to the KSI SDK to be checked, and this data is hashed and compared with the hash of the data's corresponding KSI signature. The outcome of the verification process is the resulting positive or negative comparison of the hashes.

Signatures can be verified using a publication-based, calendar-based, or key-based verification policy, as detailed below. The verification process can be performed both online and offline. Further verification

depends on use-case (signature consistency is always verified, but possible business rules regarding signing time and entity are use-case specific).

Publication-based verification	The trust anchor is the publication of a root hash of the calendar blockchain; the root hash is periodically published in newspapers and via electronic media. This is suitable for long-term verification, and assumes that signatures are extended (for an exception to this rule, see the *Extending signatures* section).
Calendar-based verification	Uses a copy of the calendar blockchain as the trust anchor, available from the Extending service.
Key-based verification	The trust anchor is the PKI signature on the calendar hash chain. This is necessary for short-term verification when no new publication exists for a particular signature.

Note: KSI SDKs have a default verification policy that combines different forms of verification and takes care of the decision making on which policy should be used for verification. This is the recommended policy to use in most deployments.

3.3 Machine Integrity, Defence and Awareness (MIDA)

3.3.1 Overview

The Machine Integrity, Defense and Awareness (MIDA) technology is designed to auto-discover compliance deficiencies and cloud threats to provide value by controlling audit costs and reducing security risks of vulnerable cloud assets. MIDA enables system administrators to make data-driven, risk-based decisions by providing visibility and awareness in relation to its monitored cloud infrastructure and resources. In other words, MIDA provides capabilities for machine and environmental integrity state capture.

The primary constituents of the MIDA technology are:

- MIDA Service – captures state information directly from cloud infrastructure. It monitors resources and events such as console logins and the creation of security groups. The data are housed within secure data containers known as dockets and pushed to MIDA State Management Services for further processing.
- MIDA Agent – provides the function of distributed data capture for system state and configuration changes. MIDA Agents reside within actual sensors, or devices, to routinely monitor events such as file or directory changes. The resulting dockets, containing the captured data, are pushed to MIDA State Management Services for further processing.
- Guardtime's KSI Blockchain – provides the distributed trust anchor for data signing, data verification and extension of the KSI signatures used to secure the technology's data containers (dockets).
- MIDA State Management Services – provides docket exchange, storage, analysis and correlation management. It consists of the following three sub-components:
    - Sentry – validates the data within dockets in accordance with user-defined business logic, ensuring that only relevant system information is monitored, collected and processed. Sentry achieves this using data attributes, contextual information and by leveraging Guardtime's eXtensible Data Attribution Language (XDAL; see below). In addition, each time Sentry receives a docket it validates its KSI signatures to safeguard the security of docket data.

- Broker – provides storage and exchange of dockets between data stores and other State Management Services. Broker has configurable rules for persisting and routing dockets, and it is further tasked with persisting data to a relational database.
- Venture – orchestrates the flow of dockets throughout the configuration of MIDA components. It provides user-configurable workflow management, and event correlation and alerting, for streamlining valuable system processes.

- MIDA Dashboard – provides a web-based interface to clearly visualize and organize the key resource and event data processed by the system. This facility is key to providing value and meaningful insights for the owners and operators of the system.

A logical overview of the MIDA technology is depicted in Figure 3 below:

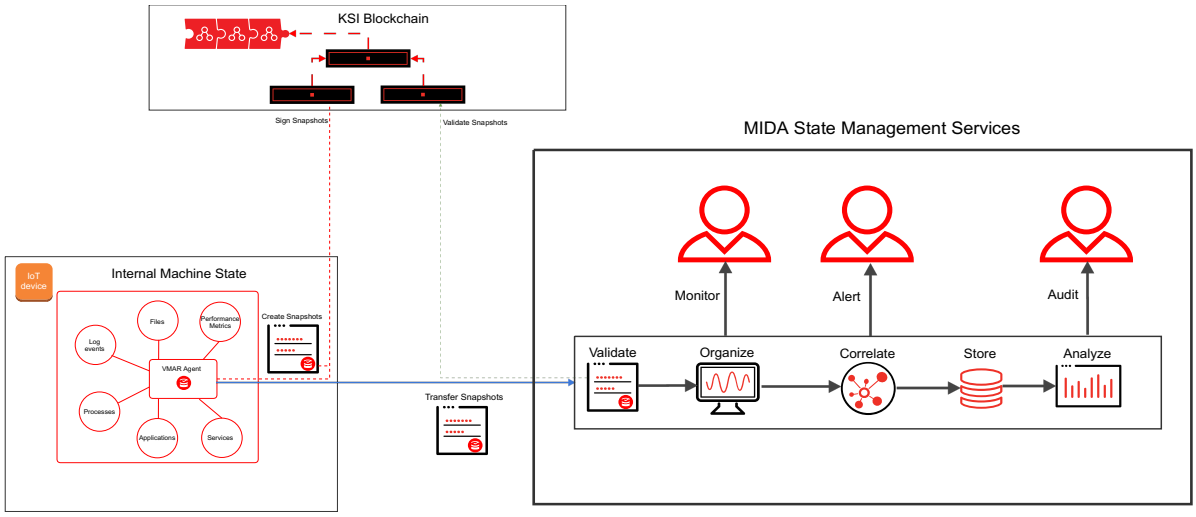

Figure 3. Logical overview of MIDA technology.

3.3.2 Role of MIDA

The main capabilities of MIDA include awareness and reporting, threat remediation, malicious or accidental state-change detection, and enhanced analytics. It is proposed to leverage the advantages of MIDA for several reasons, each of which is explained below.

The NAIADES system is heavily reliant on the safe use of the cloud for its operation. However, many data breach reports and industry analyses have pinpointed key issues with cloud security, revolving around misconfigurations, unauthorized asset creation, or misuse of credentials. This has contributed to a lack of visibility and situational awareness when running cloud infrastructures. In order to combat the issues causing these breaches, MIDA aims to:

- Provide real-time and provable awareness to detect changes across the infrastructure.
- Decrease time-to-detection of misconfigurations, unauthorized access to resources and deployment of assets.
- Provide cryptographically immutable inputs to event correlation and detailed analysis.
- Decrease cost of governance and audit via monitoring objects.

Indeed, MIDA provides adaptive state capture and event correlation allowing cloud environment owners and operators to significantly reduce the time from detection of accidental or malicious events to provable

and auditable remediation. The state capture of virtual computing environments (known as instances), security groups and other key assets can quickly indicate accidental or malicious misconfigurations.

Furthermore, as MIDA is modular by design, any number of its sub-components can be deployed to satisfy future use-case requirements of the project.

3.3.3 XDAL and dockets

Guardtime's eXtensible Data Attribution Language (XDAL) provides the basic data structure for housing the data collected by MIDA state captures. XDAL defines the syntax and semantics of the MIDA construct known as a docket. Dockets provide an interoperable and self-contained construct to cryptographically link data authenticity, identities, and contextual-based information using the KSI signature. Dockets, once sealed with KSI signatures, provide portability for state captures from MIDA Agents and Services to allow them to be verified across boundaries and in perpetuity.

Dockets are used to protect the authenticity of state-captured data and preserve the time at which state-captures occur. Dockets also encapsulate the identities of the MIDA Agents and Services responsible for their generation. The integrity of docket data can be verified by State Management Services (Sentry, Broker and Venture components), system owners, and auditors both manually and automatically.

3.3.4 Data pollers

Pollers are utilized by MIDA Agents and Services to monitor resources and events taking place within local file systems or cloud infrastructure. The data is placed inside secure dockets and sent to State Management Services for further processing. In general, pollers carry out atomic or composite tasks when querying information:

- Atomic tasks - run a single poller to capture information about a system resource or event.
- Composite tasks - run multiple pollers where the set of collected data may be used to create composite dockets, containing individual dockets of information.

The pollers also carry out tasks with or without state:

- Tasks with state - make use of timestamp information to monitor changes in queried data.
- Stateless tasks - perform the same data collection in every cycle, regardless of historical measurements.

Monitoring tasks available to MIDA Agent pollers (which physically reside within sensors, or devices) include disk monitoring, capturing SSH login events, detecting file or directory changes, and identifying ownership or permission changes. The functions of MIDA Service pollers, however, which capture information directly from cloud infrastructure, depend largely on the cloud platform chosen.

3.3.5 Provisioning Service

Provisioning Service is designed to simplify the process of providing large-scale access to KSI gateways. Access is required for certain MIDA components (Agents and Services, and all State Management Services) to sign, verify and extend KSI signatures - which secure the data within dockets. Provisioning Service facilitates the enrolment of MIDA components to KSI gateways by automatically provisioning them KSI gateway access credentials. This has the potential to greatly reduce the burden of work placed on system administrators. Provisioning Service also has a UI which handles API calls to the service. The UI is designed to facilitate the enrolment of entities (essentially a group of signers which share a common location, department or role) and automate KSI gateway credential provisioning.

3.3.6 Token Provider Server

Token Provider Server grants token-based authentication for MIDA Dashboard, Provisioning Service and Configuration Server UI users. Combined with message signing, the system validates a requestor's identity and upholds the integrity of end-to-end communications.

3.4 Accreditation

The KSI service received certification as an eIDAS qualified trust service in Q1 2020.

eIDAS is the Regulation (EU) No 910/2014 of the European Parliament and of the Council of 23 July 2014 on electronic identification and trust services for electronic transactions in the internal market and repealing Directive 1999/93/EC.

- Full-text available

https://eur-lex.europa.eu/legal-content/EN/TXT/?uri=uriserv%3AOJ.L_.2014.257.01.0073.01.ENG.

- For KSI service, Chapter III, Section 6 of Electronic time stamps is most applicable.

As Guardtime is an eIDAS qualified trust service provider in the field of electronic time stamps, KSI services must comply with the following standards:

- ETSI EN 319 401 (General Policy Requirements for Trust Service Providers).
- EN 319 421 (Policy and Security Requirements for Trust Service Providers Issuing Electronic Time-Stamps; the basis for Guardtime Timestamping Policy).

To maintain acceptance, Estonian and international legislation obliges us to check potential customers and partners against laws and public embargo lists.

A summary of the regulatory, publicly available policies and documents used can be found at https://guardtime.com/library/tsp; particular attention should be given to the Guardtime KSI service Practice Statement and the Guardtime KSI Service Disclosure Statement.

3.5 Access control and identity management

3.5.1 Identity management

The Identity and Authentication Management is the first step for accessing data, services and applications. Identity Management is a component of NAIADES platform that stores and handles users, both internal (pilots' users) and external users (Marketplace users). Identity Management is the first layer for getting the authentication of a user (external of internal). Based on the token Data signature, it will check the user role and authorize or reject some internally processes. It provides secure and private identification and authentication for users, trust management, and Identity Federation towards applications. It provides management of user life-cycle functions by providing account creation and management, and enforcement of policies and procedures for user registration, identification and authentication. It supports the enforcement of policies and procedures for user registration, secure and private authentication and user profile management. In addition, it allows to link an application with the user account, in order to enable that application to authenticate on the behalf of users, by interacting with its Oauth2 APIs. This module interacts with the Authorization and Accounting component to exchange information on access request to assets.

NAIADES project enables identity management in IoT Platform using one of the key stones (generic enablers) of FIWARE called Keyrock. Keyrock provides OAuth2-based authentication and authorization

security to NAIADES services and applications. Identity Manager APIs comply with existing standards for authentication and user and provide access information.

The main identity management concepts within Keyrock are:

- Users:
  - have a registered account in Keyrock.
  - can manage organizations and register applications.
- Organizations:
  - are group of users that share resources of an application (roles and permissions).
  - Users can be members or owners (manage the organization).
- Applications:
  - has the client role in the OAuth 2.0 architecture and will request protected user data.
  - are able to authenticate users using their Oauth credentials (ID and secret) which unequivocally identify the application
  - define roles and permissions to manage authorization of users and organizations
  - can register Pep Proxy (Access control components) to protect backends.

Keyrock component in NAIADES provides both a GUI and an API interface. Here below a summary of what these interfaces look like:

Keyrock GUI login:

This interface authenticates project operators to access management dashboard of IdM.

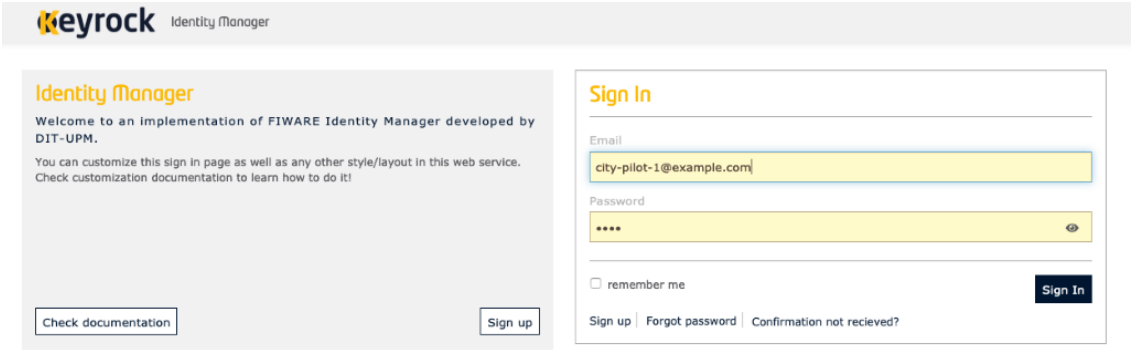

Figure 4 Keyrock GUI login

Organzations (grouping of users):

This dashboad provides project operators capability to group set of users sharing common access rights, e.g. group together NAIADES data collectors into a same category.

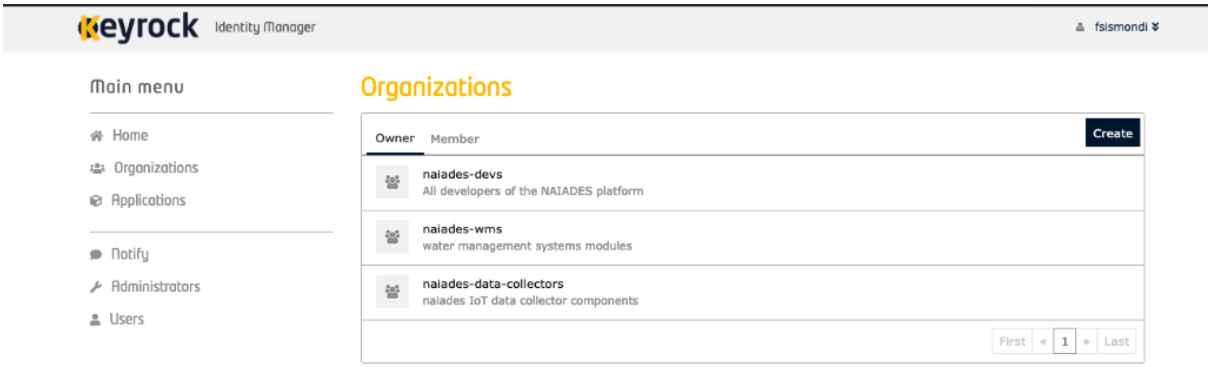

Figure 5 Dashboard of Organizations

Roles (access rights) by type of organizations:

This dashboard provides the capability to project operators to set permissions and policies per role. As an example, data-collector role includes rights to READ any entity, and WRITE WeatherObserved type of resources as shown in the Figure 6.

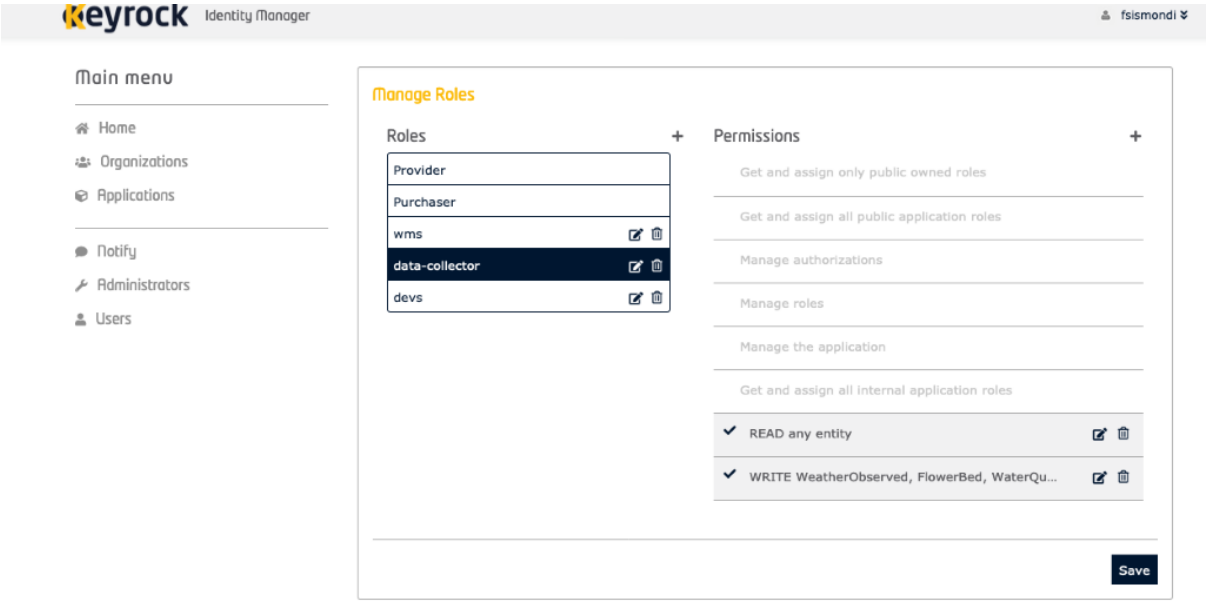

Figure 6 Manage the role

Permissions:

The following form is to describe types of permissions. Here we define permission name, and HTTP action and type of resource (using regular expressions).

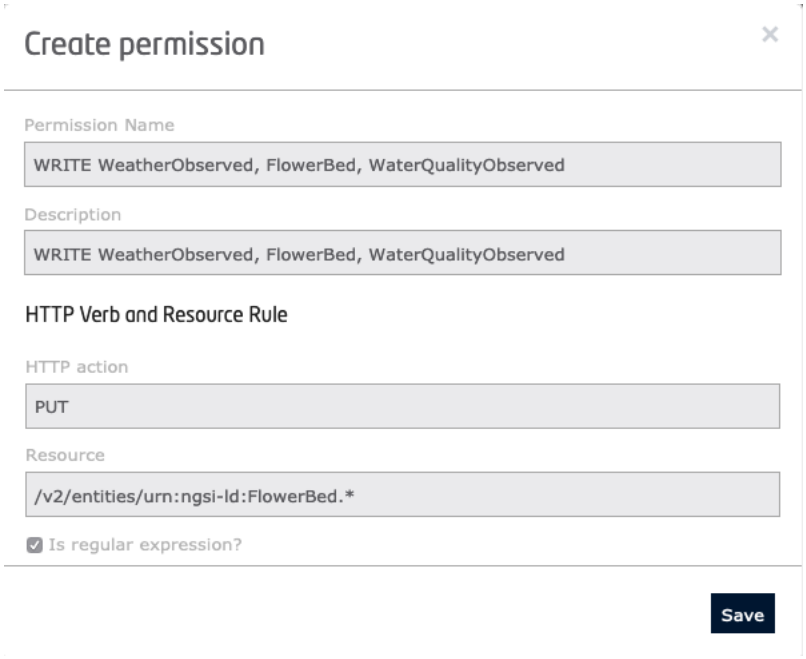

Figure 7 Description of Permissions

Once a set of users, organization (groups of users) and roles have been set up, we can use IdM to respond to external apps to query this service to ask if a certain user is valid and if it has the rights to run a certain operation on a specific resource.

3.5.2 **Access control**

The Authorization Management provides authorization and accounting capabilities which are critical aspects to support NAIADES services and applications. It enforces a set of conditions defining whether users have granted on the access to a specific resource.

Access control is implemented using a Policy Enforcement Point (PEP), which intercepts resource access requests, makes access control decision requests, and enforces access control decisions. It also provides a Policy Decision Point (PDP) that evaluates access request by checking authorization policies for rendering an access control decision. It provides a policy retrieval point that connects to the policy management component and a policy information point to obtain applicable authorization policies according to an access control decision request and attributes that are needed for evaluating authorization policies, for example the IP address of the requester, creation time of the resource, current time or location information of the requester. This information is combined in order to get a finial access control decision.

The PEP proxy Wilma is a http proxy component which is deployed for every service in the platform exposing data to the outside world. It acts as a http proxy, which forwards request to the service being protected only when policies are respected.

It heavily depends on the Identity Management, which allows us to manage specific permissions and policies to resources allowing different access levels for the users. The following diagram shows the role of this component.

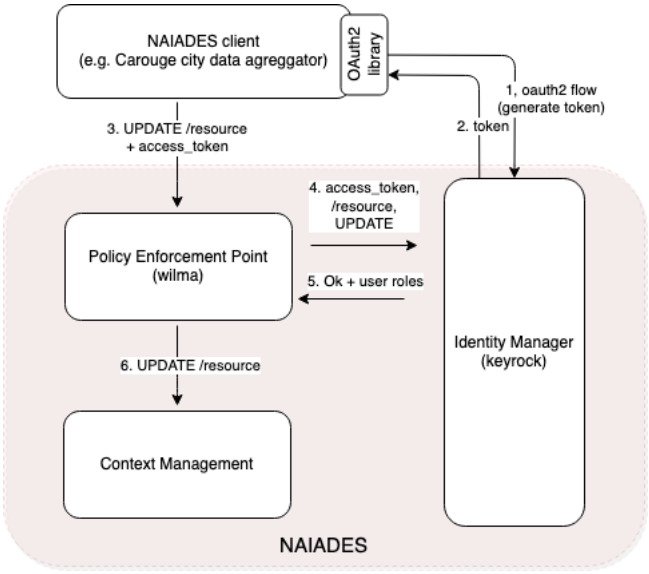

Figure 8 Flow diagram of identity management and access control

This diagram describes the solution for protecting /resource from the Context Manager service. In this case, the policy defined in IdM needs to define `carouge-city-data-aggregator` as a user with rights to run a UPDATE operation on /resource. This flow can be applied for any service implementing a REST API such as Context Management.

4 Integration and adaptation

4.1 Security and privacy of data

Traditional solutions for data authentication rely on centralized trust authorities (Public Key Infrastructures – PKI), and they often suffer from problems of scalability and resilience (providing single points of failure, for example). Furthermore, data integrity relies traditionally on the 'hardened box' concept where perimeter security keeps 'bad' actors out and 'good' actors in. Moreover, data transferability is facilitated by checksums and key-based digital signatures which rely on several systems of trust, like key management, certification infrastructure and providing roots of trust. KSI Blockchain, however, operates using different principles.

KSI Blockchain enables massive scale data authentication without the reliance on centralized trust authorities. KSI Blockchain can be used to ensure data integrity, traceability, provenance and auditability throughout the lifecycle of data. The history of data modifications and event integrity can be retrieved from the blockchain security solution at any time, and the data's validity, time of change and signing entity can be verified in a way that third-party validation, independent from the system, is possible.

The NAIADES system aims to increase the privacy and security of ICT–based smart-water management systems. In-part, this is achievable through the exploitation of a blockchain-backed architecture that provides anti-tamper and early warning protection for critical system events and feeds. Indeed, a form of decentralised, data-protection infrastructure is expected to result from the project.

In practice, data being pushed to the NAIADES platform (from real-world water sensors and devices, for example) through the API shall not contain any information which enables identification of data subjects. With respect to the blockchain-based auditing mechanism, this will be due to the operational nature of KSI Blockchain. Furthermore, any changes in cloud platform configurations and local machine states will be monitored by MIDA, which will use KSI signatures to preserve the integrity of the detected information; monitoring tools will be in place to verify that data has not been tampered with; attacker vectors and moves will be analyzable (based on the events monitored); and data in transit will contain the data owner's signature, allowing administrators of the platform to identify who created it, and if the data has been changed or diverted between components.

In the following list, an example data flow is provided to illustrate the steps taken by the NAIADES cloud platform when it receives input water quality data. Importantly, this data is signed and validated within the NAIADES system using KSI services; similar processes also occur at the outputs of the AI Services Module.

The example is taken from the Carouge pilot study, and it relates to the processing of water quality parameters collected via a number of water fountain sensors. The goal of the pilot was to improve water quality monitoring and provide warnings in relation to pH, bacteria and chlorine content. The data flow provides the basis for an illustration of the interactions between data and the security components relevant to this document.

1. The data is collected by specific water quality sensors.
2. The data is sent to the Data Collection and Aggregation (DCA), which receives environmental information from sensors and devices.
3. The data is converted to NGSI compatible in the module of Common Data Model.
4. The data is signed using a KSI SDK with access to the KSI gateway hosted on the NAIADES cloud platform. Records of the data are entered into the KSI Blockchain, and a set of KSI signatures are produced and stored alongside the original data as metadata (or other storage options may be provided for the signatures). This process is initiated by the user.

5. DCA requests a token and receives one from Keyrock (Identity management). The water measurement data are sent to the NAIADES cloud platform with the token.
6. After Data Validation Module, the data is sent to Context Management Module where, after being granted access, it will distribute the collected data to the data repository.
7. NAIADES internal users will connect through the HMI.
8. The HMI will connect with the user management API (which checks the user repository) to grant access to the users. A KSI SDK is located here to secure the integrity of Operational & Management Tools data; this will most likely be the data received (to be typically processed) from the data repository via the Context Management Module. Processed data can be signed using a KSI SDK, and a process similar to the one described in 3. will take place, if initiated by the user.
9. The user management will inform the Data Management module.
10. The user will use the HMI to request some information (data they can access with their credentials); the request is sent to the Context Management Module.
11. The Context Management Module will collect the data from the data repository and send it to the HMI when the token information (received from Keyrock, the identity management module) contains the access right.
12. The data's destination device has access to a KSI SDK, as described in 10., so modified or processed data can be signed if initiated by the user; if so, a process similar to that described in 3. will takes place.
13. NAIADES water quality services will collect the data they require from the Context Manager (with their service credentials); the Context Manager will get that data from the data repository.
14. The services will generate and output that which will be sent to the Common Data Model module, which may in-turn transform the outputs (if they are found to be incorrect). The outputs of the services can be signed using a KSI SDK, and a process similar to the one described in 3. takes place, if initiated by the user.
15. The Data validation module will send the data to the Context Management module; the Context Manager will send it to the data repository.

To provide further clarification, a diagram of interconnections of security components and other system components is provided in Figure 1 (above). In addition, MIDA will be deployed in a more general sense to monitor and secure the platform's key cloud resources; or, more specifically, it will periodically encapsulate the configuration information of the cloud resources - depended upon by the cloud-based Modules depicted in Figure 1 - in data containers (known as dockets), sign them with KSI signatures and ingest them for further analysis, providing visualization of the data if necessary.

4.2 Demo/try-out options

It is proposed to deploy an on-site KSI gateway server to provide access to the aggregation and extender services for signing, extending and verifying data. For demonstration purposes, access credentials for KSI gateway try-out servers (hosted by Guardtime Ltd) can be requested from guardtime.com/technology/blockchain-developers. The access credentials also permit you to use the KSI command-line tool, which performs various functions including the signing of data in files or other sources, extending existing KSI signatures and verifying them using different trust anchors.

KSI signatures can also be verified at https://guardtime.com/verify, which calculates a document's hash (in your local browser) and compares it with that of the document's corresponding KSI signature.

4.3 Audit trail

In the context of NAIADES, the blockchain-based auditing and situational awareness mechanism will allow all the relevant parties to be mutually assured that their signed system events have not been tampered with. A Proof of Authority scheme will be used to verify the authenticity of these record. Proof of Authority works differently than the more common Proof of Work scheme used by several cryptocurrency blockchain implementations. Proof of Authority relies on a trusted authority, or set of trusted authorities, to provide the definitive version of the data to which all parties' copies are compared. This acts as the source of truth required to show that system participants, at any given time, hold an unaltered copy of the same log data. In this project, until a future development of a central knowledgebase for the definitive version of the data takes place, the central database will be in the hands of the platform operator.

Blockchain-based auditing will not interfere with traditional log management techniques and protocols. Checks that signed data have not been tampered with, forged or deleted will be carried out by data owner's when required; whereas, current applications and systems may simply store more traditional logs into other forms of auditing system, without any necessity to extend or change their access interface. This approach will combine the highest security with the easiest deployment, as the Proof of Authority mechanism provides support for the authenticity of the logs.

The generality of the approach followed by NAIADES could also be exploited in other different scenarios where secure, distributed storage of log entries is of paramount importance (e.g. public health, consumption, public administration repositories, and notaries).

4.4 Platform integration of KSI Blockchain

KSI signing software will be located next to the platform's data generators to provide data integrity as early as possible. Data generators include the NAIADES' Collector/Aggregator, which receives environmental information from sensors and devices, and AI services. The platform operator will therefore benefit from an increased transparency which helps to prevent system misuse. Users will be clear that data is not manipulated after the signing process takes place and that the service providers are not responsible any such modifications. KSI signing software will also be located at the outputs of key architectural modules of the NAIADES platform. This data (processed, transferred or otherwise) will also be imbued with a transparency to help prevent system misuse.

Guardtime Ltd provides fully-featured Software Development Kits (SDKs) for C, Java, Go and .NET, to facilitate the integration of a KSI Blockchain-based system into the NAIADES platform. The KSI SDK provides the lowest level of integration, enabling "full access" to the signing, extending and verification functions, as well as to their fine-tuning.

The following table highlights the features of the KSI SDK:

	KSI SDK
API type	Provides a native (Java, JavaScript, C, C#, Go, .NET) interface. The user application will use the interface directly by embedding the KSI SDK in the application.
Authentication	Connects directly to KSI gateway using KSIAP/KSIEP protocols which use symmetric key HMAC authentication.
Signature persistence	Does not provide KSI signature persistence. User decides where to store KSI signatures and when to extend them.

Client-side aggregation	Provides client-side aggregation, which allows you to provide multiple documents (hashes) for signing in one go. This means that only a single signing request is sent from the SDK to the gateway.

The way a KSI gateway is deployed in terms of network zones depends on the network architecture of the user organization and how the applications are used for signing. A typical deployment consists of an organization with a KSI gateway deployed in their network, which provides access to the KSI service network, for the users of that organization.

The following must be considered when assessing deployment options:

- A KSI gateway must be behind a firewall or have the local iptables/firewall configured to allow traffic only from authorized IP addresses.
- A KSI gateway must be accessible to applications that sign and verify data, and extend KSI signatures. Corporate deployments usually expect access to the KSI gateway to be through a secure network.
- A KSI gateway must have access to upstream aggregators and extenders in the KSI service network. This communication is not required to be over secure networks.

4.5 Platform integration of MIDA

The NAIADES platform requires additional security measures to prevent the exploitation of vulnerabilities associated with blockchain use. In order to achieve the goal of producing a holistic resilience framework - robust against outside threats and detecting anomalies in internal data flows - modules like deep packet inspection, intrusion detection and firewalls are required. It is therefore feasible to add MIDA's awareness and reporting, threat remediation, malicious or accidental state change detection, and enhanced analytics capabilities to the resilience framework. Indeed, MIDA is designed to process large amounts of data, with visualization tools and a convenient search facility, offering notifications, alerts and a messaging system for distributed use.

MIDA will be integrated with the NAIADES platform's Security Mechanisms Module. MIDA will be used to encapsulate (unmodified and modified) cloud configuration information in data containers (named dockets) and ingest them for further processing. A KSI gateway will be deployed in the NAIADES' cloud for access to the signing, extending and verification of docket signatures.

4.6 Platform integration of Identity management and Access control

It is important to establish a framework of policies (and technologies) to ensure users are assigned the appropriate access to the NAIADES resources by Identity management and Access control systems.

There are several good practices to take into account when using these systems, to help uphold the security of the system. These include maintaining the principle of least privilege, defining appropriate trust boundaries, understanding hierarchical inheritance of access, as well as others.

NAIADES enables these functionalities using FIWARE components as explained in the Section 3.5. Identity Management has the role of authenticating all applications integrated with NAIADES Cloud platform, mainly from external NAIADES applications like Data Collection Aggregation, Marketplace, Operational and Management tools, external public applications via Marketplace, etc. Currently Both IdM and PEP have been integrated into the server side (development and production servers) using virtualization tools: docker and docker-compose. Please refer to D3.9 for more details on this.

In order to ease the integration of clients (apps requesting resource access), UDGA provides the following WIKI document to the consortium partners. It details the process for generating and using OAuth tokens for accessing NAIADES' services resources using authentication and authorization:

https://gitlab.distantaccess.com/naiades/naiades-platform-poc/-/wikis/securing-API-with-PEP-proxy

As a summary, this demonstrates:

1. un-identified clients try to access resources.
2. Putting a new value for a /resource into context manager.

The above two figures show the example of the behaviors of IdM and Access control. Figure 9 shows the example of 'GET" operation without the designated token that indicates the access right. As shown in the figure, it responds with an error message. Figure 109 illustrates the same operation with a correct token. As the figure depicts, the request is accepted and the value is returned.

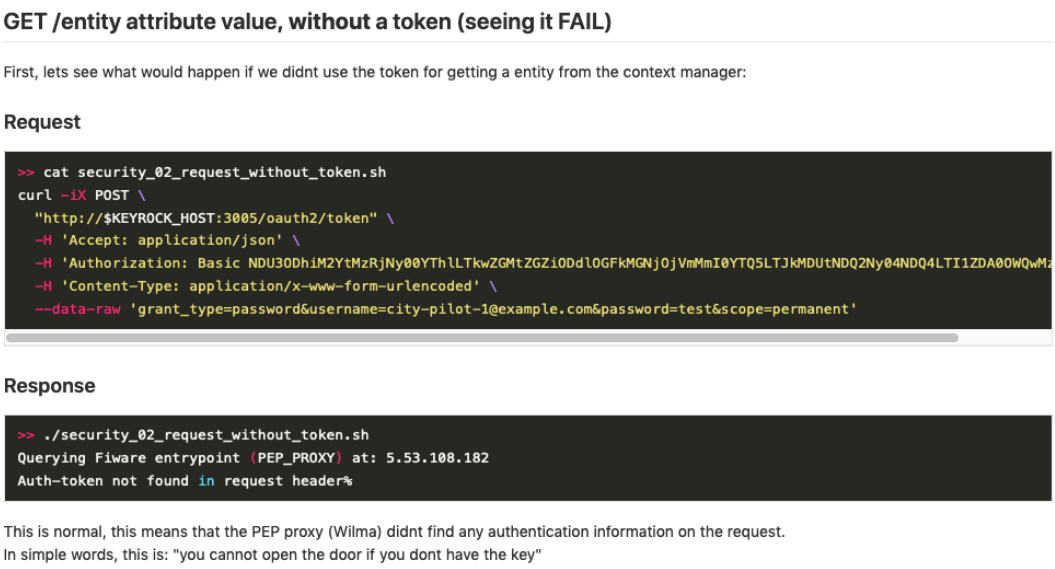

Figure 9 Example of the access the resource without designated token

PUT /entity attribute value, with a token

Lets update moisture value to a random number:

Request

```
curl --location --request PUT \
    "http://$PROXY_HOST:1027/v2/entities/urn:ngsi-ld:FlowerBed:FlowerBed-1/attrs/soilMoistureVwc/value"\
    --header "Fiware-Service: carouge" \
    --header "Fiware-ServicePath: /Watering" \
    --header "X-Auth-Token: $KEYROCK_TOKEN" \
    --header "Content-Type: text/plain" \
    -d $(( RANDOM % 10 ))
```

Response

```
>> ./security_04_update_entity_with_token.sh
Querying Fiware entrypoint (PEP_PROXY) at: 5.53.108.182


If you didnt get any error message, then you can re-run script <security_04...>.sh you will get a different value for the
```

PUT request was accepted, now lets see what inside of that value again:

```
>>./security_03_request_with_token.sh

Querying Fiware entrypoint (PEP_PROXY) at: 5.53.108.182
2%
```

Woo-hoo! we managed to update an entity attribute with our token! We have been granted to do this cause our token was generate with a user with privileges to do so (access control).

Figure 10 Example of access the resource with correct token

As an example of the usage of identify management and access control, the following Figures illustrate a series of sequence diagram with Carouge use case. Figure 11 shows a sequence diagram of obtaining a token before access NAIADES service, and Figure 1212 illustrates sequence diagram to update an entity using the received token. Figure 13 depicts a sequence diagram that an AI module subscribes to an entity with a token.

Figure 11 Obtain a token

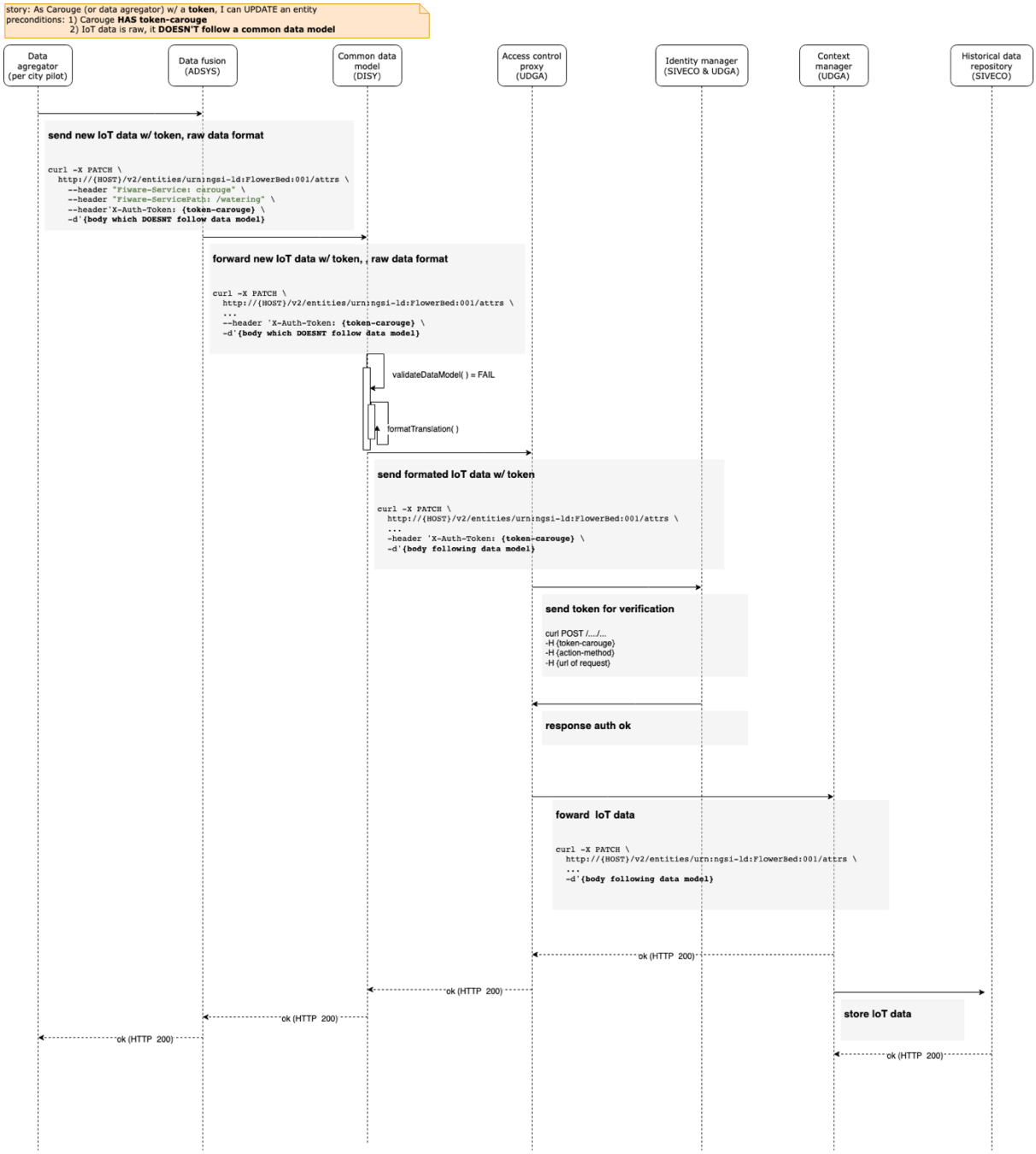

Figure 12 Update an entity

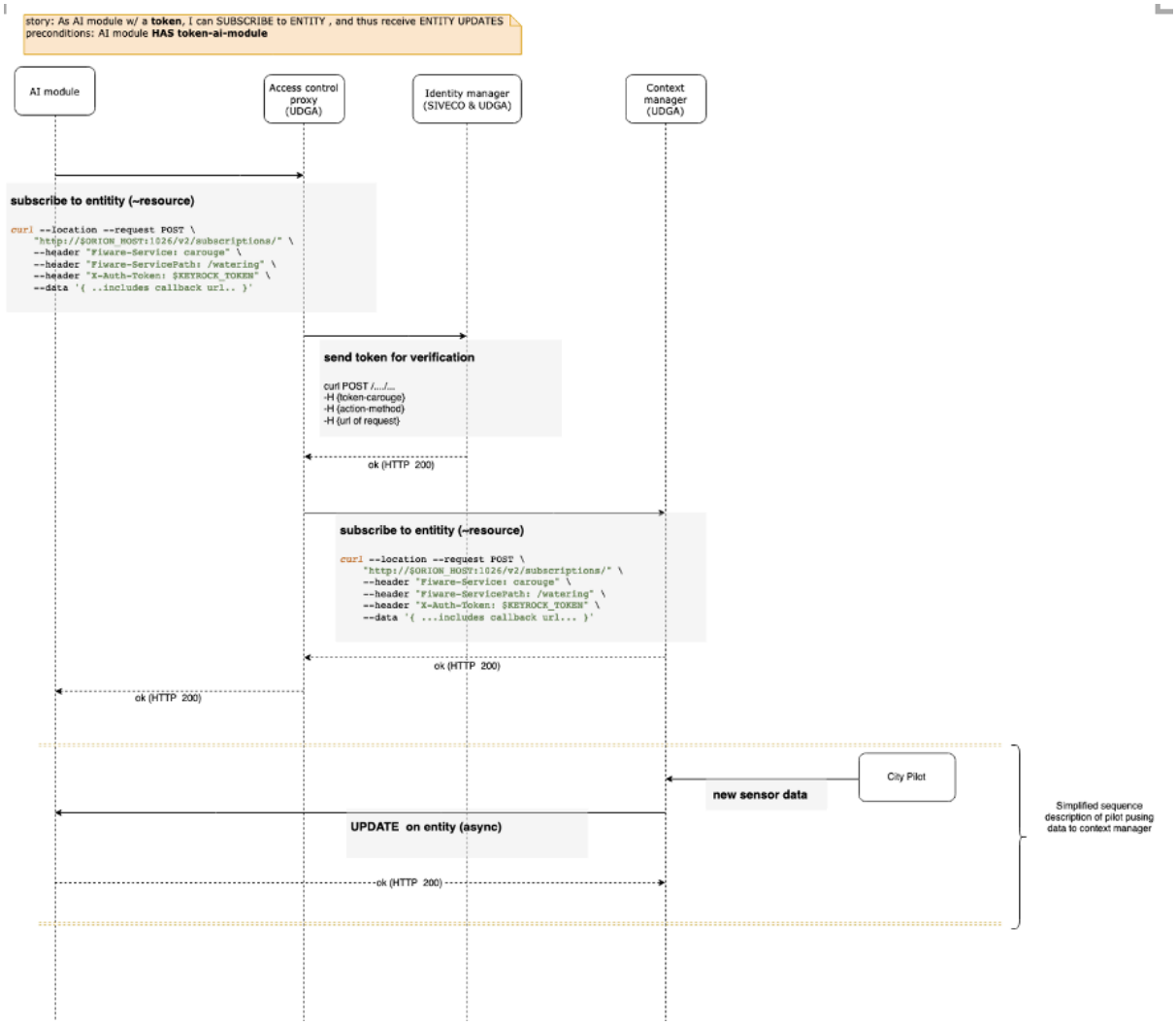

Figure 13 Subscription with a Token

5 Pilot test

5.1 Overview

Pilot testing of the NAIADES ecosystem took place under real operating conditions to both demonstrate its capability and to validate it. Three implementations of the NAIADES architecture were piloted, each of them adapted to meet the situation and requirements of their alternative, precise implementation. The three pilot sites were Alicante, Brăila and Carouge, and the Data Signature module was integrated with the NAIADES platform for each of the pilot locations.

Questionnaires were sent to the hosts of the pilot projects in each location. The information is included in the following section, which is in-turn followed by an analysis of the content of the questionnaires, with respect to the information that can be gained to improve the design of the security components relevant to this deliverable. The Data Signature module was integrated with the NAIADES platform for each of the pilot locations.

5.2 Questionnaires

Questionnaires were provided to each of the pilot sites, with each site trialling a different version of the NAIADES system. Detailed information concerning the deployment circumstances, as well as the technologies implemented at each site, can be found in the previously issued deliverable, D2.9 NAIADES Architecture Mid-term; summary information is also given below.

Currently, a completed questionnaire is only available for the Brăila site. To provide context, the city of Brăila has no major concerns with the production of water. Its major issue relates to water distribution networks, in particular water losses. Although the city has a water loss strategy in place, with the purpose of decreasing water losses, leakage is an ongoing problem whose detection is seen as a priority. At the moment, average water losses amount to about 750 l/h/km, and officials aim to reduce this value to 50 l/h/k. Indeed, in 2018 the city had 41% of losses/non-revenue water. They have a street network with low pressure, and the network for apartment blocks is served by a station that increases the level of pressure. Two use cases were defined for the Brăila site implementation, namely 1) a water consumption forecast and 2) leak detection. The completed questionnaire is provided in Appendix A.

Other sites have not yet returned their completed questionnaires.

6 Conclusions and future steps

The NAIADES project envisions the positive transformation of the urban water environment. This will be achieved through automated and smart-water resource management and environmental monitoring, attaining a high level of water services for both residential and commercial consumers. Central to this project is the efficient use of physical and digital components of the water ecosystem to gather and process environmental information. This data, which can at times comprise sensitive aspects of personal data, must be appropriately secured.

The deliverable D7.6 gathers all the specifications about NAIADES platform security measures. The main points of this document can be summarized as follows:

- KSI Blockchain provides a method, and a globally distributed network infrastructure, for the issuance and verification of specialist KSI signatures.
- KSI signatures provide independently verifiable proofs of data-integrity, signing-time and signing-entity.
- KSI signatures can be used to sign documents of any type and size; the resulting "digital fingerprints" of the data have footprints of only a few KB.
- KSI signatures are produced by an aggregation service which is only accessible via a KSI gateway.
- KSI gateways also provide access to the trust anchors used for the extension of pre-existing signatures, for added, long-term security.
- KSI signatures do not provide non-repudiation, meaning that the identity of an entity requesting a signature cannot be proven indisputably using KSI Blockchain-based technology alone; this is advantageous for end-user privacy.
- The KSI gateway layer should be hosted on-site for the best possible security and service quality. The remainder of the KSI service network should be hosted off-site, by a third party (e.g. Guardtime).
- The KSI SDK provides an opportunity for the integration of signing and extending features with the platform.
- The KSI service received certification as an eIDAS qualified trust service in Q1 2020.
- Guardtime's MIDA technology provides capabilities for machine- and environmental-integrity state-capture, including awareness and reporting, threat remediation, malicious or accidental state-change detection, and enhanced analytics.
- The main security implications of the flow of data through the platform have been identified and addressed (by the planned security mechanisms of the platform).
- The points of interaction of the blockchain-based audit trail with the existing architectural design of the platform have been specified.
- Functional, non-functional, hardware and software requirements for the implementation of security mechanisms have been specified.
- An analysis of architecture has provided recommendations to uphold system security in case of the project's future development.
- For secure, efficient Identity management and access control for NAIADES, FIWARE components of Keyrock and PEP Proxy are utilized in conjunction with FIWARE based context management.

- Using Keyrock (in conjunction with PEP Proxy) enables to add OAuth2-based authentication and authorization security to your services and applications. Only the allowed users will be able to access the network, data and services.
- The PEP Proxy allows to programmatically manage specific permissions and policies to NAIADES resources allowing different access levels to the multi-stakeholders.
- The backend of Idm will generate an authentication token, at login, for NAIADES users, based on row credentials as username and password.
- The services are provided as dockerized, and can be run on any environment and can be used and accessed both from inside and outside NAIADES cloud platform.
- Results of the pilot test questionnaires showed that there is minimal readiness for cyber threats management in the water utility and each company has its own approach.

The following list outlines future steps for the project:

- Functional testing (performed by UAT).
- Intentional misuse of platform data and its analysis to strengthen platform security.
- Interviews with platform users.
- Feedback collection.
- Design and implementation of a central knowledgebase, storage a definitive log record for use with the Proof of Authority scheme.
- Migration from Development platform to Production platform
- End to End test and validation with pilot use cases in Production platform.

Appendix A: Questionnaires

The following questionnaire is for the Brăila site:

Questionnaire - learning from different water systems

This questionnaire is intended to be filled-out by utility companies, and it addresses the following questions:

- Data formats and communication protocols.
- Cyber security and data protection activities and scope.

1) Data formats and communication protocols

- Please provide information regarding the models, data formats and communication protocols used:

Country	Meter reading Data available	Information Model	File format
Romania	Indexes, meter registers	Siemens MAG 8000	CSV

2) Cyber security and data protection activities and scope

- Does your system generate PII (personally identifiable information)? If yes, can you specify the type of PII you will handle?
    a. For SCADA systems there are no PII
    b. For ERP system – Siveco Applications – we have PII (name, address for clients)
- GDPR came into force on 25 May 2018. Are there any country specific regulations that are stricter than GDPR and should be followed alongside the GDPR? Can you share what regulations these are? Have you assigned a GDPR data officer or are planning to do so in near future? Can you provide his/her contact to us? Considering the GDPR what are the main challenges that you have to solve in coming 6-12 months?

1. Law no. 190/2018 on measures to implement Regulation (EU) 2016/679. This law establishes the measures necessary to implement at national level, mainly, the provisions of art. 6 paragraph (2), art. 9 paragraph (4), art. 37-39, 42, 43, art. 83 paragraph (7), art. 85 and of art. 87-89 of the General Regulation on data protection.

2. Our company has a DPO since May 2018.

Email address: cpdp@apabraila.com

3. The main challenges identified are the complexity of the problems related to the mechanisms of protection of personal data, as well as the involvement in the process of the different functions, that require the familiarization of the personnel with the new regulatory framework.

- Other challenges identified are the difficulties in interpreting the notions of operator and authorized person, the excessive use of consent, the content of the register of processing activities.
- The Romanian authorities have not yet issued sufficient guidelines or guidelines regarding the problematic or interpretable aspects regarding the application of the GDPR.

- What new cyber security risks can you see in your systems? What methods are used to be prepared for potential cyber security threats?
    1. MAKING AN INVENTORY OF DATA
    2. ORGANIZING THE DATA IN SUITABLE LOCATIONS
    3. DEVELOPMENT OF CONFIDENTIALITY POLICIES
    4. PROTECTION OF DATA COLLECTED ON THE INTERNET
    5. CREATING SECURITY STRATEGIES
    6. PLAN FOR LOSS OR THEFT OF DATA
    7. EMPLOYEE TRAINING

- What kind of technology you use for the data exchange to external systems? Authentication, Public Key Infrastructure, Certification Authorities and other?
    a. For SCADA system data are transmitted using FTP transmission with Authentication
    b. For Client Portal communication is based on SFTP protocol
- Are the cyber security requirements coming from internal documents and procedures or from national or water standards? Can you provide the list of the publicly accessible ones?
    a. We are in the process of complying with the EU Regulation 1148/2016
- Do you have any data that you need to hold for long term?
    - YES
- Do you have any data that's integrity needs to verified to 3rd parties?
    a. NO

PS! Can you provide the contacts of the person responsible for the cyber security topic when exchanging information to other platforms?

a. We don't have a responsible with cyber security yet

3) Future

- Provide the vision how could the system evolve in the future, incl related to: new business use cases; new functionalities; types and resolution of data; data availability; GDPR compliance.

1. Password policies should encourage employees to use the strongest possible passwords, without creating the need or temptation to reuse or write passwords.

2. Internal access, non-public WLAN devices should be restricted to specific users, specific to the greatest extent possible, while meeting the needs of the business of the organization.

3. Encryption should be used to protect sensitive data, in addition to meeting regulatory requirements applicable to the Protection of information.

4. Establishing a policy for the use of internet by the employees.

5. We are in the process of complying with the EU regulation 1148/2016.