DESIGNING A SELF-SOVEREIGN IDENTITY (SSI) MANAGEMENT SYSTEM USING DISTRIBUTED LEDGER TECHNOLOGY (DLT) AND ARTIFICIAL INTELLIGENCE (AI)

Lappeenranta–Lahti University of Technology LUT

CT30A8912 Software and system architectures

2024

Trieu Huynh Ba Nguyen

Nguyen Bao Quan

TABLE OF CONTENT

1. Introduction

Our project focuses largely on providing users with the smoothest, most comfortable and safest experience when using online public services through e-ID verification. E-ID is one of the modern trends to support users to quickly verify their identity when using their personal smart devices. In essence, it is a form of virtual token, using biometric data as well as identity documents, and other types of data to help users quickly confirm their identity, and be redirected to the pages of the public services they need.

In the project, we use the DLT paradigm - distributed ledger technology - to store information securely in cyberspace. It comes with a Blockchain system to support the system, providing a decentralized, secure storage method to record transactions as well as sensitive information of users. Each operation or service is represented as a block in the blockchain and the very nature of avoiding recording all data in the same place but dividing it into many places like this will help reduce the risk of errors, as well as increase the security of the entire system. Users always own a private key - on their personal devices, this key will be used to match the public key stored in the block chain, thereby increasing accuracy, reducing waiting time and other procedures that users have to do when confirming their identity, as well as ensuring their privacy. Combining these two technologies together is necessary, and we have studied a lot for this issue, this will ensure that the system will operate most smoothly, bring comfortable experience to users, as well as ensure transparency and integrity of their data.

The main goal we want to achieve through this project is to provide secure and convenient access to online public service portals through user identity verification. Our system includes the use of biometric data (face) as well as identification documents (ID, driving license, passport, etc.) provided by users, combined with well-trained AI modules to verify and compare the provided data with each other.

We assessed that the project was of medium complexity, requiring the cooperation of many different parties, so the integrity of the system was extremely important and had to be ensured at all times. Errors or failures could lead to significant inconvenience and disadvantages for users, more seriously, data loss and financial impact on the project. Therefore, we agreed that annual maintenance and testing of the system was essential to ensure that no functions were broken, and we paid special attention to the security function, as it needed to be continuously improved and adapted to technological advances, as well as to minimize the possibility of serious errors occurring.

We use Agile methodology throughout the project development process, as we believe its incremental and iterative nature is a perfect fit for our current needs. The Large Scale Scrum Framework (LeSS) enables us to deliver the workflow while bringing multiple teams together to work towards larger project goals. The project budget is expected to come mainly from external funding sources, from government and community organizations, which will ensure that the project development process runs smoothly.

Overall, our system is always user-oriented, we hope to provide them with the safest, most convenient and comfortable experience possible throughout the usage process. The system has been designed to be user-friendly as well as provide the fastest response speed, the least errors, as well as the highest safety and security thanks to the advanced modules that we will include in the project.

2. Architectural Analysis

2.1. Stakeholders

Stakeholder	Role	Needs	Concerns
Users (Citizens)	End-users who manage their identities. e.g., A Finnish citizen using SSI for healthcare services via Kela	- Easy and secure identity management - Control over personal data - Seamless access to services - Compliance with eIDAS for cross-border recognition	- Privacy and data protection - User-friendliness and accessibility - Trust in the system and technology - Potential misuse or unauthorized access
Service Providers	Organizations offering services e.g., Nordea Bank offering online account opening	- Reliable and verifiable identity information - Quick and secure identity verification - Enhanced customer experience - Reduced fraud and identity theft	- Integration complexity with existing systems - Compliance with regulatory standards (e.g., eIDAS) - Data security and privacy issues - Trust in the SSI management system
Identity Providers	Entities that issue and validate identities e.g., Finnish Digital Agency (DVV)	- Secure and efficient credential issuance - Compliance with legal and regulatory frameworks (e.g., eIDAS) - Trustworthy and tamper-proof system	- Maintaining high security standards - Handling large volumes of verification requests - Protecting against fraudulent activities
Regulatory Bodies	Government and regulatory authorities e.g., European Data Protection Supervisor (EDPS)	- Enforcement of compliance with eIDAS and other regulations - Oversight and monitoring of identity management systems - Facilitation of cross-border recognition and interoperability	- Ensuring privacy and protection of citizens' data - Addressing potential legal and ethical issues - Ensuring technological neutrality and inclusivity
Developers	System and software developers e.g., Developers at TietoEVRY building an SSI platform	- Clear guidelines and standards for implementation - Tools and resources for building secure and compliant systems - Ongoing support and updates for system improvements	- Keeping up with regulatory changes - Balancing security, usability, and performance - Managing potential technical vulnerabilities
Businesses	Companies using the SSI system for operations e.g., Verkkokauppa.com verifying customer identities	- Streamlined KYC processes and reduced verification costs - Improved customer trust and satisfaction - Assurance of authenticity and reduced fraud	- Data protection and compliance with regulations - Adapting to new technologies and systems - Integrating with existing business processes

Academia and Researchers	Researchers studying SSI systems e.g., Aalto University conducting research on digital identities	- Access to data (with consent) for research and development - Collaboration opportunities with industry and government	- Ensuring ethical use of data and privacy protection - Maintaining transparency and trust in research activities
Legal Entities	Legal professionals and advisors e.g., Roschier Attorneys Ltd advising on data protection laws	- Clear legal framework and guidelines for SSI implementation - Assurance of compliance with eIDAS and other relevant laws	- Addressing legal liabilities and responsibilities - Ensuring the system adheres to privacy laws and regulations
Public Institutions	Government services utilizing SSI e.g., Kansaneläkelaitos (Kela) providing social security services	- Efficient and secure identity verification for public services - Reduction in administrative burdens - Enhanced trust and transparency in government services	- Ensuring inclusivity and accessibility for all citizens - Protecting against identity fraud and misuse - Ensuring the system's scalability and reliability

2.2. Constraints

a) Technical Constraints

- Scalability: Can automatically scale its computing capability to handle a high volume of traffic.
- Interoperability: Integrating with existing identity management systems and DLTs, smoothly working with different platforms and technologies.
- Data Storage and Management: Storing large amounts of biometric data securely while ensuring quick access for verification processes using blockchain.
- Reliability: Consistent and reliable performance, particularly for real-time identity verification processes.

b) Security Constraints

- Privacy: Protecting user data (especially biometric data) from unauthorized access and ensuring compliance with data protection regulations like GDPR.
- Cybersecurity: Resilient against cyber-attacks, including hacking, phishing, and denial-of-service (DoS) attacks.
- Decentralization Security: Maintaining the integrity and security of the decentralized network, preventing attacks such as the 51% attack on blockchain networks.

c) Legal and Regulatory Constraints

- Legal Compliance: Adhering to regional and international regulations, and other local laws.
- Legal Liability: Defining legal liability in cases of identity fraud, data breaches, and misuse of the system.
- Data Sovereignty: Ensuring that data stored and processed by the system adheres to the data sovereignty laws of different countries.

d) Ethical and Social Constraints

- User Consent and Control: Ensuring users have full control over their data and provide informed consent for any data sharing.

- Bias and Fairness in AI/ML: Avoiding biases in AI/ML models to ensure fair and unbiased identity verification across different demographics.
- Accessibility: Accessible to all users, including those with disabilities and those without access to advanced technology.

e) Operational Constraints

- Cost: Managing the costs associated with developing, deploying, and maintaining the SSI system, including the costs of computational resources, network infrastructure, and regulatory compliance.
- Maintenance and Updates: Regular updates and maintenance of the system to address security vulnerabilities and improve functionality.
- User Training and Support: Providing adequate instructions and support for users to effectively use the system.

f) Performance Constraints

- Latency: Minimizing the latency in, enabling quick and seamless user experiences.
- Throughput: Handling a high number of concurrent requests without performance drop.
- Accuracy and Reliability: High accuracy in the verification processes to minimize false positives and negatives.

2.3. Development and Architecture Principles

- **Security by Design**: Security should be integrated at every stage of development. This includes employing end-to-end encryption for data transmission and storage, implementing multi-factor authentication (MFA), and conducting regular security audits. Such measures reduce the risk of data breaches, ensure the integrity of user identities, and build trust among users.
- **Privacy by Design**: Privacy must be a fundamental consideration in the design and operation of the system. This involves minimizing data collection to only what is necessary, using pseudonymization and anonymization techniques, and ensuring users have control over their data. These practices enhance compliance with data protection regulations like GDPR and foster user trust.
- **Decentralization**: Utilizing DLT achieves a decentralized system where no single entity controls the entire network. Identity proofs and verifiable credentials should be stored on a blockchain, with smart contracts automating processes. Decentralization increases system resilience against single points of failure and reduces the risk of centralized data breaches.
- **Scalability**: The system must be designed to handle a growing number of users and transactions efficiently. This can be achieved by using scalable blockchain technologies (e.g., sidechains, sharding) and optimizing AI/ML algorithms for performance. Ensuring scalability accommodates increasing demand without compromising performance.
- **Interoperability**: The system should interact seamlessly with other identity management systems and services. Adopting open standards and protocols (e.g., DID, Verifiable Credentials, OAuth 2.0) and ensuring API compatibility facilitates cross-border recognition and integration with existing systems, enhancing the system's utility.
- **User-Centric Design**: Prioritizing the needs and experience of end-users is crucial. This involves conducting user research, designing intuitive interfaces, and providing comprehensive user support and education. A user-centric approach increases user adoption, satisfaction, and engagement.

- **Regulatory Compliance**: Ensuring the system complies with relevant legal and regulatory requirements is essential. Regularly reviewing and updating the system to adhere to laws such as eIDAS, GDPR, and other standards avoids legal penalties and ensures the system is legally sound.
- **Ethical AI/ML Practices**: AI/ML solutions should be implemented ethically and responsibly. This includes using unbiased datasets, ensuring transparency in AI decision-making, and regularly auditing AI/ML models for fairness. Ethical practices ensure fair treatment of all users and maintain public trust in the technology.
- **Modular Architecture**: A modular architecture allows for flexibility and easy updates. Using microservices architecture, containerization (e.g., Docker), and ensuring components can be independently updated or replaced enhances the system's adaptability and facilitates maintenance and scalability.
- **Continuous Improvement**: Adopting practices that support ongoing improvement and innovation is crucial. Utilizing agile development methodologies, implementing continuous integration and continuous deployment (CI/CD) pipelines, and encouraging feedback loops keep the system up-to-date with the latest technological advancements and user needs.
- **Robust Governance**: Establishing clear governance structures and policies is essential. Defining roles and responsibilities, creating policies for data management and user rights, and setting up a governance board ensures accountability, transparency, and proper management of the system.

3. Architecturally Significant Requirements

3.1. Functional Requirements

Functional requirements	Description	Stakeholders
FR-001	User registration with biometric data (facial recognition) is supported.	Citizens
FR-002	Users can manage and update their personal information.	Citizens
FR-003	Multi-factor authentication (MFA) is supported for enhanced security.	Citizens, Service Providers
FR-004	Identity verification using AI/ML-based facial recognition is supported.	Service Providers
FR-005	Verifiable credentials are stored on a distributed ledger.	Identity Providers
FR-006	Service providers can request and verify user identities.	Service Providers
FR-007	The system ensures compliance with GDPR and other relevant data protection regulations.	Regulatory Authorities
FR-008	An API for interoperability with other identity management systems is supported.	Developers, Service Providers
FR-009	Users can change their consent or revoke access to their data.	Citizens
FR-010	A user-friendly interface accessible to individuals with disabilities is supported.	Citizens

3.2. Non-functional Requirements (Quality Attributes)

Non-functional requirements	Description	Stakeholders
QA-001	Average annual uptime of 98%.	All stakeholders

QA-002	User requests are responded within 3 seconds under normal operating conditions.	Citizens
QA-003	The system should scale to support up to 10000 concurrent users.	Developers, Service Providers
QA-004	All data in transit and at rest has end-to-end encryption.	Citizens, Service Providers
QA-005	Integration with at least 3 external third-party identity management systems.	Developers, Service Providers
QA-006	24/7 customer support with a maximum response time of 1 hour is provided.	Citizens, Service Providers
QA-007	The system should be accessible to users with disabilities, adhering to WCAG 2.1 standards.	Citizens
QA-008	Identity verification is processed within 3 seconds.	Service Providers
QA-009	The system should have a modular architecture to make updates and maintenance easy.	Developers
QA-010	AI/ML models are regularly audited for bias and fairness.	Citizens, Regulatory Authorities

3.3. Architecturally Significant Requirements (ASR)

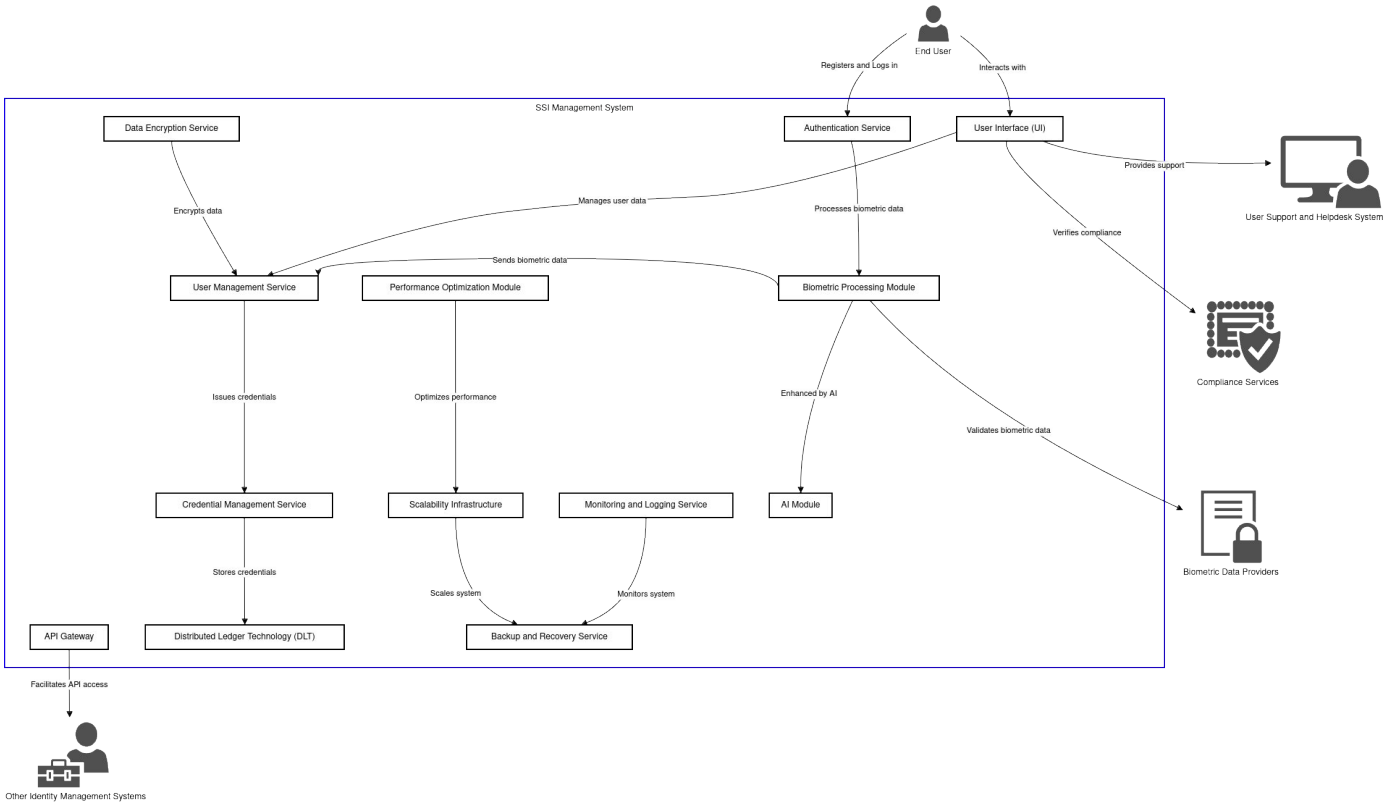
ASR	Description	Reasoning
ASR-001	User registration with biometric data (facial recognition) is supported.	Impacting the choice of biometric technologies and the integration of AI/ML for facial recognition.
ASR-002	Multi-factor authentication (MFA) is supported for enhanced security.	Affecting the security architecture and the implementation of various authentication mechanisms.
ASR-003	Verifiable credentials are stored on a distributed ledger.	Affecting the choice of DLT technology and impacts how data integrity and decentralization are managed.
ASR-004	The system ensures compliance with GDPR and other relevant data protection regulations.	Significant implications on data storage, processing, and user privacy measures.
ASR-005	An API for interoperability with other identity management systems is supported.	Impacting the system's integration capabilities and the architecture of external interfaces.
ASR-006	Average annual uptime of 98%.	Impacting the overall system architecture, including redundancy, failover mechanisms, and hosting infrastructure.
ASR-007	User requests are responded within 3 seconds under normal operating conditions.	Affecting the performance tuning, load balancing, and system optimization strategies.
ASR-008	The system should scale to support up to 10000 concurrent users.	Affecting the design for scalability, including database sharding, microservices, and horizontal scaling.
ASR-009	All data in transit and at rest has end-to-end encryption.	Critical implications for the security architecture, affecting how data encryption and key management are implemented.

ASR-010	The system should have a modular architecture to make updates and maintenance easy.	Impacting the overall design approach, promoting modularity, microservices, and containerization for ease of updates and maintenance.

4.Element Catalogue

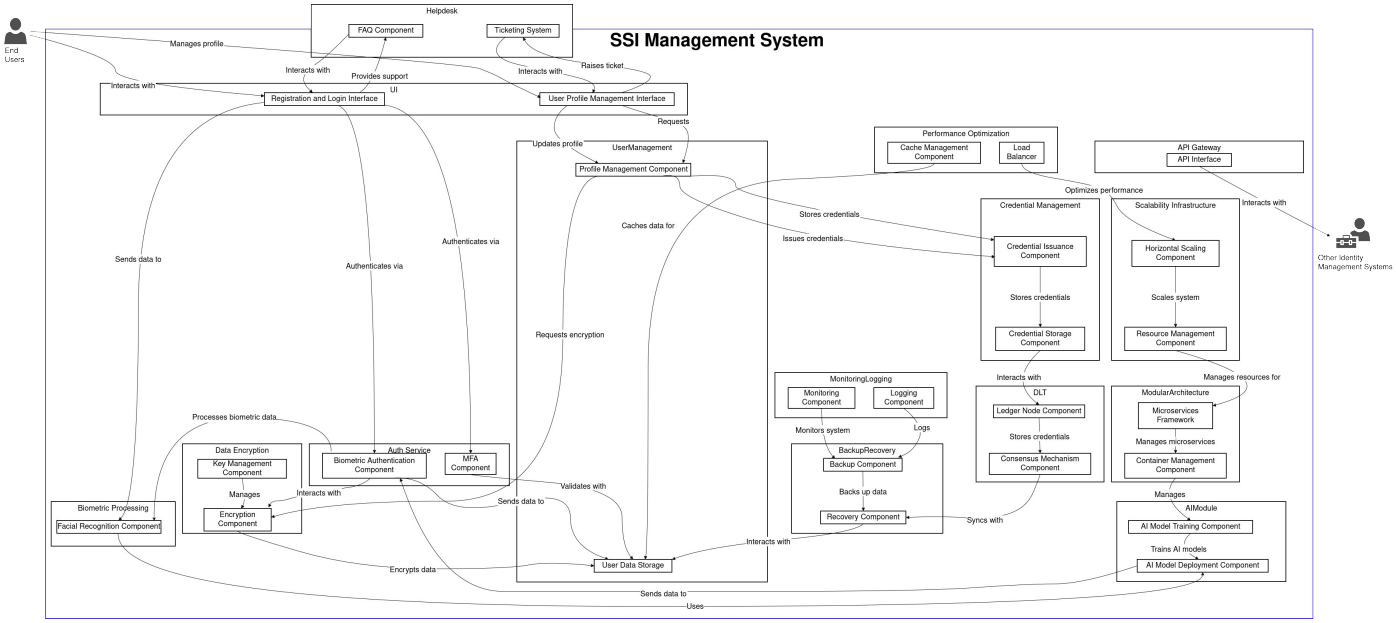
Name	Technology	Description
User Interface (UI)	React, HTML/CSS, JavaScript	The front-end component allowing users to interact with the SSI system, including registration and login.
Authentication Service	Node.js, Express	Backend service handling user authentication, including biometric and MFA.
Biometric Processing Module	Python, TensorFlow, OpenCV	AI/ML component for processing and verifying biometric data (facial recognition).
User Management Service	Node.js, Express	Manages user data, registration, and profile updates.
Credential Management Service	Node.js, Express, Blockchain (e.g., Hyperledger Fabric)	Handles issuance, storage, and verification of verifiable credentials on a distributed ledger.
API Gateway	Kong, Node.js	Facilitates communication between different system components and external systems.
Data Encryption Service	AES, RSA, TLS/SSL	Ensures all data in transit and at rest is encrypted for security and compliance.
Compliance Management Module	Node.js, Express	Ensures the system adheres to GDPR and other relevant data protection regulations.
Monitoring and Logging Service	ELK Stack (Elasticsearch, Logstash, Kibana), Prometheus, Grafana	Monitors system performance, logs activities, and provides analytics and alerts for uptime and security.
Performance Optimization Module	Nginx, Redis, Load Balancer	Optimizes response times and ensures the system can handle high concurrency.
Scalability Infrastructure	Kubernetes, Docker, Microservices	Ensures the system can scale to support up to 10,000 concurrent users.
Modular Architecture Framework	Microservices, Docker, Kubernetes	Facilitates easy updates and maintenance by promoting modularity and containerization.
Distributed Ledger Technology (DLT)	Hyperledger Fabric, Ethereum	Underlying technology for storing verifiable credentials in a decentralized manner.
Backup and Recovery Service	AWS S3, Azure Blob Storage, On-prem Backup Solutions	Ensures data is backed up regularly and can be recovered in case of system failures.
User Support and Helpdesk System	Zendesk, Freshdesk	Provides users with support and assistance, including FAQ, live chat, and ticketing.

5. Context View



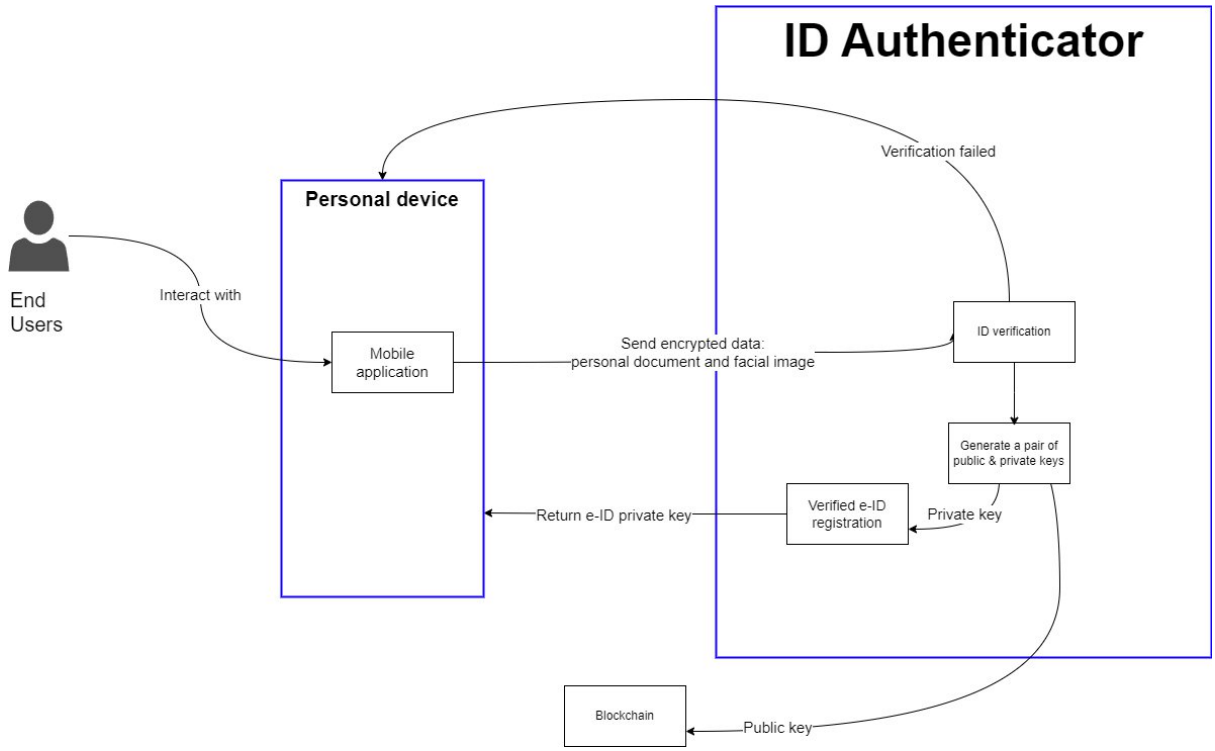
The context view diagram of the SSI Management System shows its interactions with external entities. The system facilitates user registration, authentication, and credential management, leveraging an AI module to enhance biometric data processing. It interacts with biometric data providers for authentication accuracy and other identity management systems for interoperability. The system ensures secure and compliant operations with data encryption and continuous monitoring. High reliability and performance are maintained through optimization and scalability infrastructure, while a dedicated helpdesk system provides robust user support.

6. Functional View

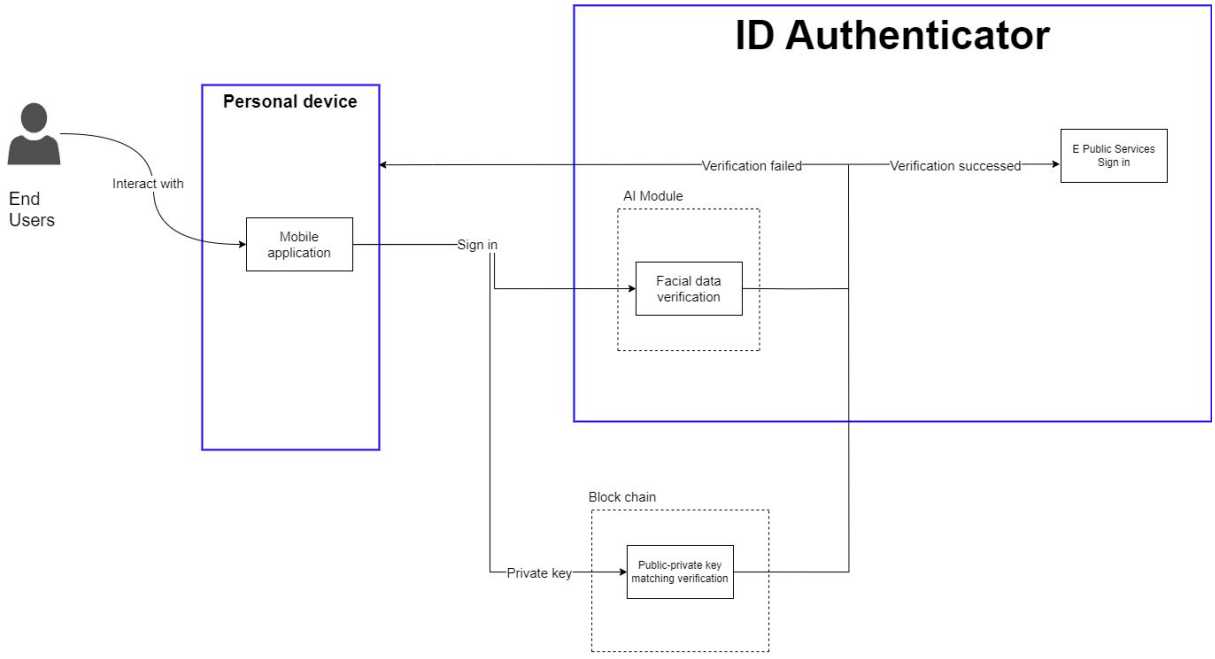


The functional view diagram presents the internal architecture of the SSI Management System, depicting how various components interact with each other and external entities to fulfill system requirements. The system is divided into several subsystems, each containing components responsible for specific functions.

7. Information View



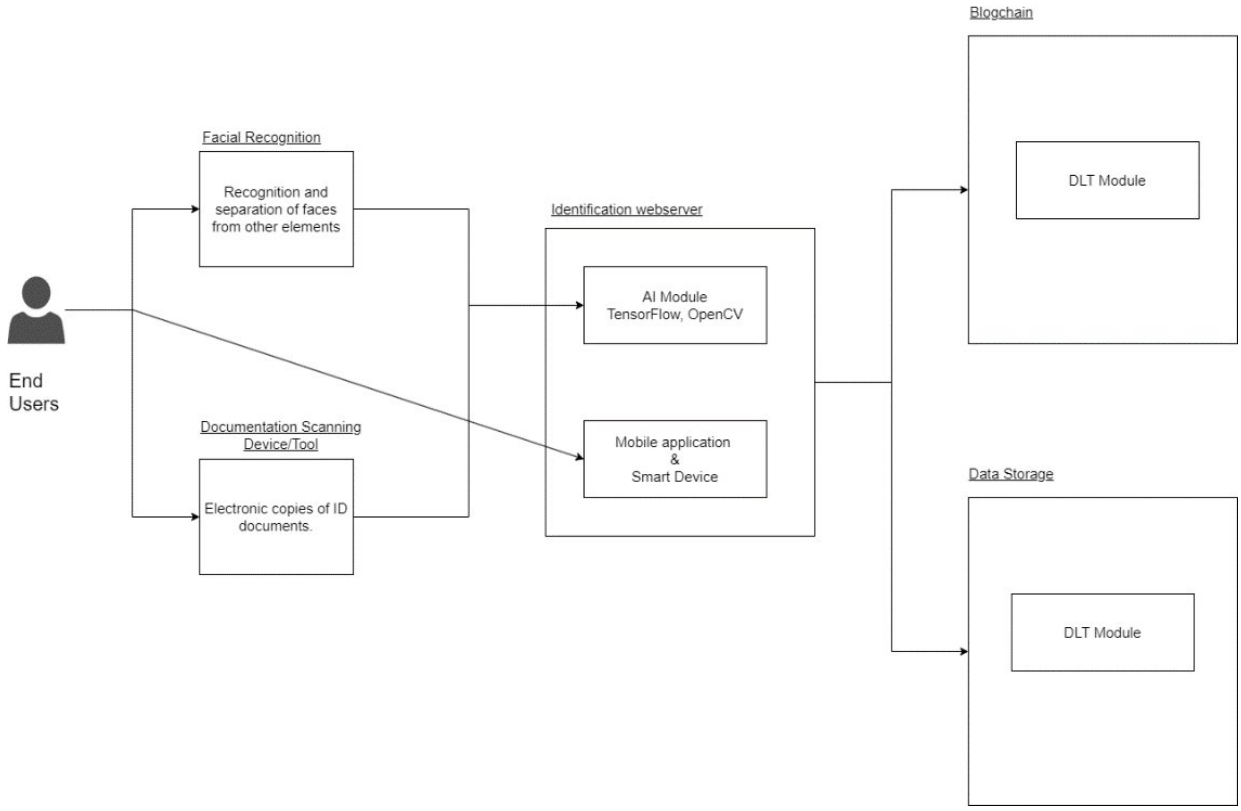
The first information view shows the integration process of e-ID. The process begins with the user going through certain applications on their smart devices, which allow them to scan their face and an electronic copy of their ID document. This information is then sent to the ID verification service for verification. If the verification is successful, a private-public key pair is generated, the private key is sent back to the user's device and stored there, while the public key is sent to the blockchain system for future verification. In case the verification fails, a message is sent back to the user's device to inform them of the status. Once the user receives an e-ID token containing some encrypted information as well as their private key, they can use it to log in and use public services later, the authentication is complete.



The next information view shows the process by which users log in to public services using their e-ID. The process starts when the user sends their login request along with their e-ID token to the ID Authenticator via some apps on the smart device. First, the app will match the public and private keys together to ensure they

are a pair. In addition, the trained AI module will also be used to analyze and evaluate the user's facial features to ensure they match the facial data that is being authenticated. If both factors are passed, the user will be logged in and use the public services they want. If only one of the two factors is passed, or both factors are not passed, the user's login request will be rejected.

8. Deployment View



The diagram above shows the physical interaction of the hardware on the device with the software components. Users can use their personal ID (personal ID, driving license, etc.) as well as their facial image to authenticate themselves. In the identification web server system, a well-trained AI module will be used to recognize and record the unique features of the user's face through deep learning algorithms such as TensorFlow or OpenCV. Once the user is authenticated, the blockchain system will be used to decide the gateway to the service portal.

9. Design Decisions

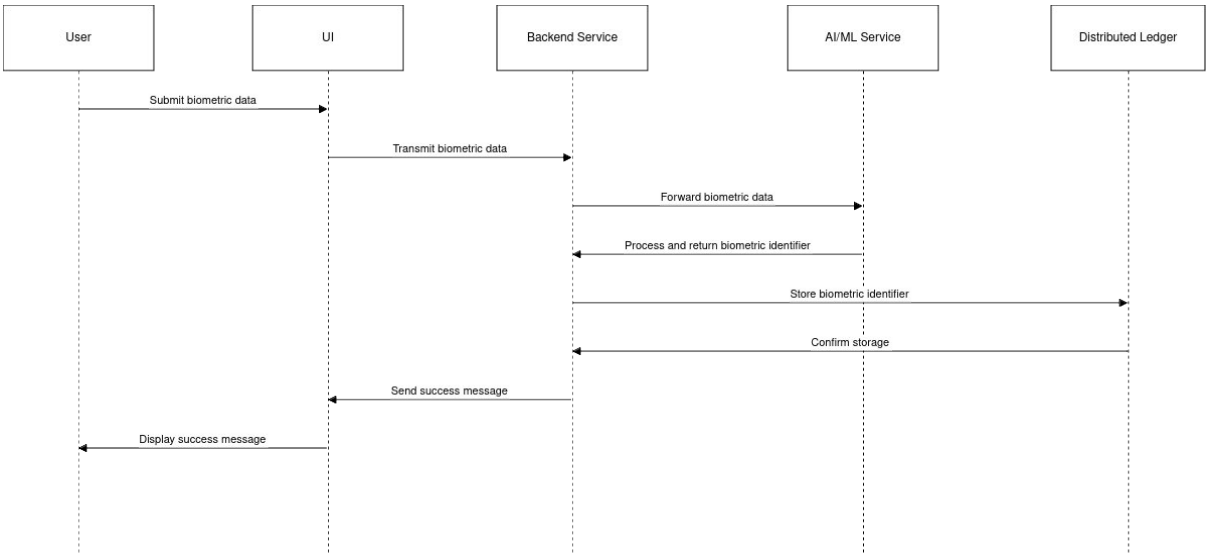
Design Decision	ASR Addressed	Alternative Considered	Rationale
Use of Biometric Authentication with Facial Recognition	ASR-001 (User registration with biometric data)	Other biometric methods (fingerprint, iris)	Facial recognition balances user convenience and security, with wide acceptance and ease of use.
Implementation of Multi-Factor Authentication (MFA)	ASR-002 (Multi-factor authentication)	Single-factor authentication (password only)	MFA enhances security significantly, mitigating risks associated with password-only systems.
Use of Distributed Ledger Technology (DLT) for Credential Storage	ASR-003 (Verifiable credentials stored on a distributed ledger)	Centralized database storage	DLT offers decentralized, tamper-proof storage, ensuring high integrity and security.
Compliance with GDPR and Other Regulations	ASR-004 (Compliance with GDPR and other regulations)	Minimal compliance focus	Strict compliance is critical for user trust and legal adherence, avoiding penalties and trust loss.
API Gateway for Interoperability	ASR-005 (API for interoperability)	Direct integration without an API gateway	An API gateway provides scalable, flexible integration with various external systems.
High Availability and Performance Optimization	ASR-006 (98% uptime), ASR-007 (3 seconds response time), ASR-008 (Support for 10,000 concurrent users)	Standard server infrastructure without advanced optimization	Ensuring high availability and performance is crucial for user satisfaction and system reliability.
End-to-End Data Encryption	ASR-009 (End-to-end encryption)	Partial or no encryption	Full encryption ensures data privacy and security, essential for protecting sensitive information.
Modular Architecture for Maintainability	ASR-010 (Modular architecture)	Monolithic architecture	Modular architecture allows easier maintenance, updates, and scalability, promoting flexibility.

10. Architecture Evaluation

10.1. Scenario-based evaluation

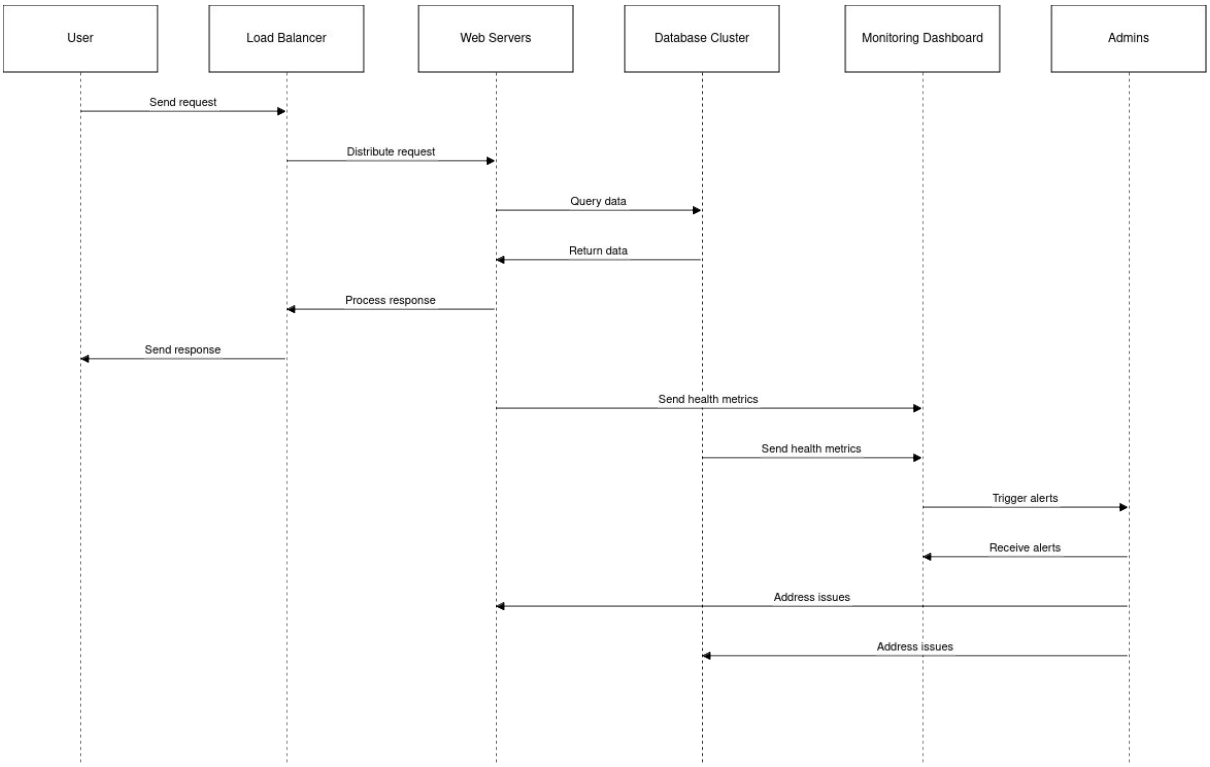
a) Scenario 1: User Registration with Biometric Data (FR-001)

- **Stimulus**: A new user initiates the registration process by providing biometric data (facial recognition).
- **Response**: The system captures the biometric data, processes it using AI/ML algorithms for facial recognition, and securely stores the identity proof on a distributed ledger.



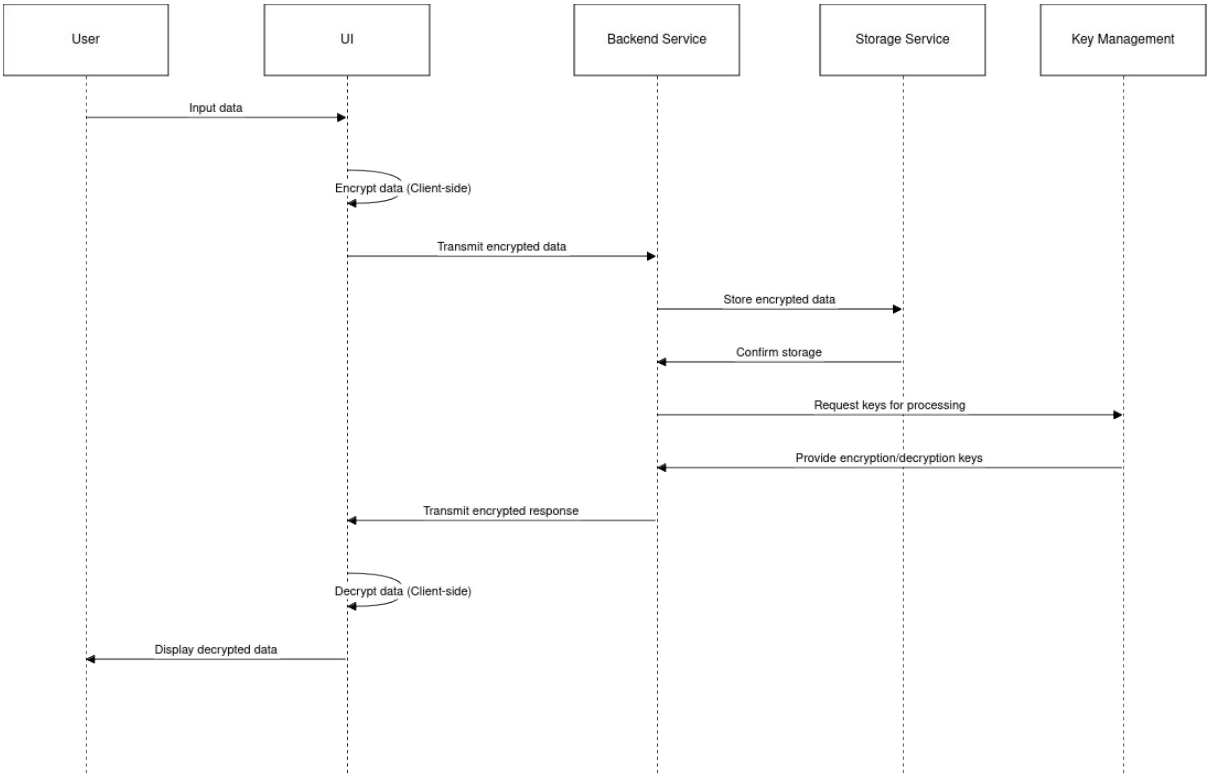
b) Scenario 2: System Uptime (QA-001)

- **Stimulus**: The system receives a continuous stream of user requests and operations over time.
- **Response**: The system maintains an average uptime of 98%, ensuring high availability and reliability for users.



c) Scenario 3: End-to-End Encryption (QA-004)

- **Stimulus**: Any data (personal information, credentials) is transmitted between the user and the system.
- **Response**: The system uses end-to-end encryption for all data in transit and at rest, ensuring data privacy and security.



10.2. Prototyping

a) Working with Prototypes

- **FR-001 - User Registration with Biometric Data**
  - Prototype: Functional Prototype with UI Mockup.
  - Purpose: Validate the effectiveness and user experience of the biometric data capture and registration process.
  - Testing Methods: Develop a prototype that simulates user registration, including capturing facial recognition data and processing it using AI/ML algorithms.
  - Stakeholders: Citizens, Identity Providers.
  - Goals: Collect feedback on ease of use, accuracy, reliability, and user comfort with the biometric capture process.
  - Decisions: Determine the best facial recognition technology, refine the user interface, identify usability issues.
- **FR-003 - Multi-Factor Authentication (MFA)**
  - Prototype: Interface and Workflow Prototype.
  - Purpose: Test the implementation and user experience of multi-factor authentication mechanisms.
  - Testing Methods: Create a prototype that includes various MFA options (e.g., SMS, email, authenticator app) and integrates them into the login process.
  - Stakeholders: Citizens, Service Providers.

- Goals: Evaluate security, usability, and user preference for different MFA methods; gather feedback on the authentication process.
- Decisions: Choose the most effective and user-friendly MFA methods, adjust the authentication flow.

- **FR-008 - API for Interoperability with Other Systems**
  - Prototype: API Skeleton.
  - Purpose: Test the interoperability and integration capabilities of the system with external identity management systems.
  - Testing Methods: Develop an API prototype that allows integration with at least three different identity management systems, simulating data exchange and identity verification processes.
  - Stakeholders: Developers, Service Providers.
  - Goals: Assess ease of integration, compatibility, and performance of the API; collect feedback on integration issues.
  - Decisions: Refine API design, improve documentation, ensure robust error handling and compatibility.
- **QA-001 - System Uptime**
  - Prototype: Skeleton Prototype with Monitoring.
  - Purpose: Validate the system's ability to maintain high availability and uptime.
  - Testing Methods: Set up a prototype with key system components (e.g., load balancer, web servers, database cluster) and monitoring tools to simulate continuous operations.
  - Stakeholders: System Administrators, Service Providers.
  - Goals: Monitor system performance, identify potential bottlenecks or points of failure, gather data on system reliability.
  - Decisions: Optimize system architecture for reliability, implement redundancy and failover mechanisms, improve monitoring and alerting processes.
- **QA-004 - End-to-End Encryption**
  - Prototype: Functional Prototype.
  - Purpose: Ensure data security and privacy through end-to-end encryption.
  - Testing Methods: Develop a prototype that implements encryption for data in transit and at rest, simulating data transmission and storage processes.
  - Stakeholders: Citizens, Service Providers, Regulatory Authorities.
  - Goals: Validate effectiveness of encryption methods, ensure compliance with security standards, assess impact on system performance.
  - Decisions: Confirm encryption techniques, enhance security protocols, address performance issues related to encryption.

b) Running the Tests

1. Setup Prototypes: Develop prototypes for each of the selected ASRs, ensuring they are functional and representative of the final system.

2. Invite Stakeholders: Involve key stakeholders in testing sessions, providing them with clear instructions and scenarios to follow.
3. Collect Feedback: Use surveys, interviews, and direct observation to gather feedback on usability, performance, and overall satisfaction.
4. Analyze Results: Review feedback to identify common issues, preferences, and areas for improvement.
5. Iterate and Improve: Refine the prototypes based on feedback and repeat the testing process if necessary to ensure the final design meets stakeholder expectations.

c) Decisions Based on Results

- Technology Selection: Choose the best technologies and methods based on prototype performance and stakeholder feedback.
- User Experience Enhancements: Improve the user interface and workflows to ensure a seamless and intuitive experience.
- Security and Compliance: Ensure the system meets all security and regulatory requirements, addressing any identified vulnerabilities.
- Performance Optimization: Optimize system architecture and components to meet performance and scalability goals.

11. Architectural Roadmap

11.1. Reviewing and organizing the backlog

Architecturally Significant Requirements or ASRs can be defined/categorized as follows:

a) New Features:

- ASR-001: User registration with biometric data (facial recognition) is supported.
- ASR-002: Multi-factor authentication (MFA) is supported for enhanced security.
- ASR-003: Verifiable credentials are stored on a distributed ledger.
- ASR-005: An API for interoperability with other identity management systems is supported.

b) Architectural Improvements:

- ASR-004: The system ensures compliance with GDPR and other relevant data protection regulations.
- ASR-008: The system should scale to support up to 10000 concurrent users.
- ASR-010: The system should have a modular architecture to make updates and maintenance easy.

c) Bugs or Defects:

There is no ASRs that can be categorized in this section.

d) Technical Debt:

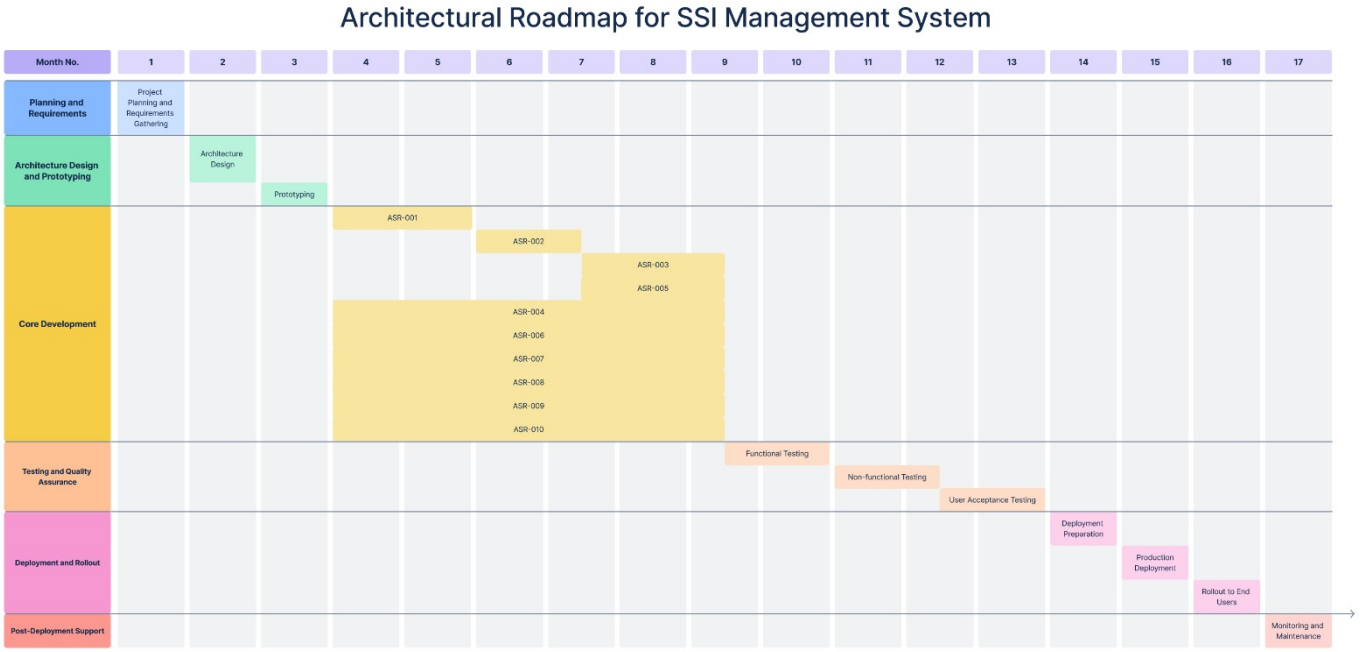
- ASR-006: Average annual uptime of 98%.
- ASR-007: User requests are responded within 3 seconds under normal operating conditions.
- ASR-009: All data in transit and at rest has end-to-end encryption.

The classification of Architecturally Significant Requirements (ASR) helps us understand the nature of these requirements and their impact on the project. We can completely rely on it to prioritize parts such as new features, architectural improvements, or bugs addressing and fixing.

The priority of these sections can be based on the needs of the stakeholders as well as the business context at that time. For example, the focus on ASR-009 is due to the sensitive nature of user data and the need to protect their biometric data for government services. This also comes with full compliance with GDPR and other data protection regulations (ASR-004), which helps users feel secure in granting permission for the app to use their data. sensitive, as well as limit future legal risks. Besides, ensuring that the app must provide seamless services, and meet user needs in the shortest time (ASR-007) as well as the ability to maintain interaction with public services through API ports (ASR-005) are very important. On the other hand, non-functional requirements such as ensuring the system can serve a large number of users at the same time (ASR-008), as well as making sure the software architecture is easy to maintain for upgrading and repairing (ASR-010) is also one of the must-have items when deploying this software solution.

Understanding the order in which ASRs are needed will give us a better overview of what we should prioritize for implementation. This will help a lot in planning as well as contribute to promoting the project to run smoothly.

The Architectural Roadmap



11.2. Risk assessment

There are some risks that could remarkably prevent the delivery of the software on time to the product owner:

a) Data disclosure or unauthorized access to the system

- Mitigation: Use a variety of encryption and storage methods to prevent attackers from finding important information all in one place. Tighten access to suspicious devices, as well as deploy the most modern and advanced security plans. Security practices may include increasing the frequency of scheduled system checks as well as increasing the frequency of unscheduled system checks.
- Contingency: In case there is an attack on the database, or data is leaked, it is necessary to prepare an action plan in advance to promptly identify vulnerabilities and respond to the attack. In addition,

warning relevant parties as well as taking appropriate remedial measures should also be implemented as soon as possible.

b) Comply with current regulations

- Mitigation: Ensure that the software solution has been vetted and approved, and that it fully meets the requirements of applicable laws, including GDPR. This can be done through users' confirmation of the use of their personal data, as well as ensuring that users and relevant parties fully understand and consent to what data will be used. Besides, users' voices also need to be respected as they have the full right to refuse, allow, access, edit and delete their personal information when needed.
- Contingency: There should be plans prepared in advance to deal with legal issues if necessary. As well as always being ready to adjust the software to suit current requirements.

c) Time and budget constraints

- Mitigation: A specific plan for using the budget so that spending is most effective and stable. The plan should be meticulously prepared to ensure that the software solution will be delivered to the customer on time and on budget without being exceeded.
- Contingency: In cases where budgets are exceeded or problems arise during development that prevent the product from being delivered to the customer on time, there should be a contingency plan for this problem. The plan may list contingencies that can be implemented as sources of funding or negotiate additional funding from partners, optimize the use of funds - this comes with cutting source of money for unnecessary areas, or negotiate for an extension of time to complete the project.

d) Scalability issues

- Mitigation: There are tests to ensure that the app will still work well when there are more than 10,000 users at the same time. Ensure that the response speed to users is within the allowed threshold and that no serious errors occur as well as take appropriate measures for each error that occurs due to too many users at the same time.
- Contingency: In case an error occurs when there are too many users, there should be measures to restore the system, bandwidth and connection to ensure the app can be used as soon as possible. Send notifications to relevant parties about the issue as well as notify users about progress and when the app can be operational again.

12. Integration Plan

12.1. Dependency on External Systems or Platforms

Our SSI management system depends on several external systems and platforms to function effectively. The following ASRs and scenarios show the need for inter-organizational integration:

- **ASR-005: API for Interoperability**
  - Scenario: External identity management systems need to access and verify user credentials stored on our platform.
  - Rationale: To ensure seamless integration with other identity management solutions, enabling users to utilize their SSI across different platforms.
- **ASR-004: Compliance with GDPR and Other Regulations**
  - Scenario: The system must exchange data with compliance monitoring and reporting services.
  - Rationale: Ensuring that the system adheres to regulatory requirements and can provide necessary compliance reports.

- **ASR-001: User Registration with Biometric Data**
  - Scenario: Biometric data providers supply facial recognition data.
  - Rationale: Enhancing the biometric authentication process with accurate and verified data from trusted providers.

12.2. Data Sharing Requirements

a) Data Shared by External Stakeholders

- Biometric Data: Biometric data providers supply facial recognition data. This is essential for user authentication and registration processes.
- Compliance Data: Government authorities and compliance services provide regulatory guidelines and compliance status. We need to ensure the system meets legal and regulatory requirements.
- User Data: Other identity management systems share user credentials and profile information. This is needed for interoperability and seamless user experience across platforms.

b) Data Shared by Our System

- Credential Data: User credentials stored on the DLT. They allows other systems to verify and authenticate users.
- Compliance Reports: Generated compliance reports and status. They provide necessary documentation for regulatory bodies and audits.
- User Profile Data: User profiles for interoperability. This is needed for seamless user experiences across integrated systems.

12.3. Standard Requirements

- W3C Verifiable Credentials (VC) Standard: For structuring and sharing user credentials.
- OAuth 2.0 / OpenID Connect: For secure authentication and authorization.
- RESTful API Standards: For API communication between systems.
- GDPR Compliance Standards: For data protection and privacy regulations.
- JSON Web Tokens (JWT): For secure data exchange and user authentication.

12.4. Regulatory Compliance and Audits

- GDPR: Ensuring that all data handling practices comply with the General Data Protection Regulation for user privacy and data protection.
- eIDAS: Compliance with the Electronic Identification, Authentication, and Trust Services regulation for secure electronic transactions.
- Regular Audits: Conducted to ensure ongoing compliance with GDPR and other relevant regulations.
- Compliance Reporting: Generating reports to demonstrate adherence to regulatory requirements, which may be reviewed during audits.

13. Deployment Plan

13.1. Testing Integrations

1. Unit Testing: Individual components, such as the API Gateway and Biometric Processing, will be unit tested to ensure they function as expected in isolation.
2. Integration Testing:
    - API Gateway: Test integration with external identity management systems and biometric data providers to ensure seamless data exchange.
    - Compliance Management: Test integration with compliance services to verify data exchange for regulatory reporting.
3. End-to-End Testing: Conduct comprehensive testing scenarios that simulate real-world use cases, involving multiple components and external systems to validate the entire workflow.
4. User Acceptance Testing (UAT): Engage end users and stakeholders to test the system in a real-world environment, gathering feedback and ensuring the system meets their needs and expectations.
5. Security Testing: Perform vulnerability assessments and penetration testing to ensure data security during integration, especially for sensitive user and biometric data.

13.2. Implementing Integrations

1. Incremental Integration: Gradually integrate external systems to minimize disruptions and allow for iterative testing and improvements.
2. API Gateway Configuration: Set up and configure the API Gateway to manage and secure API requests, implementing necessary authentication and authorization mechanisms.
3. Data Mapping and Transformation: Ensure data formats from external systems are correctly mapped and transformed to the required formats for internal processing.
4. Monitoring and Logging: Implement monitoring and logging mechanisms to track data flow, detect issues early, and ensure smooth integration.

13.3. Elements Hosted at External Stakeholders' Premises

- Biometric Data Providers: Host and manage the hardware and software required for capturing and processing biometric data. They will also handle secure data transmission to our system.
- Compliance Services: Host regulatory compliance monitoring tools and databases to receive and process compliance data from our system.
- Other Identity Management Systems: Host user data, authentication services, and APIs that will interact with our API Gateway to verify and exchange user credentials.
- Government Authorities: Host compliance databases and reporting tools to receive data for regulatory audits and compliance verification.

13.4. Responsibilities for Deployment and Updates

a) Development Team

- Responsible for the initial deployment of the entire SSI system, including setting up all internal components and initial integrations.

- Handle updates and maintenance of the internal system components, including UI, Authentication Service, Biometric Processing, User Management, Credential Management, Compliance Management, and Monitoring and Logging.

b) External Stakeholders

- Biometric Data Providers: Responsible for maintaining and updating the biometric capture and processing hardware/software.
- Compliance Services: Manage their compliance monitoring tools and ensure they can handle data from our system.
- Other Identity Management Systems: Ensure their systems and APIs are compatible with our integration requirements and perform necessary updates to maintain interoperability.
- Government Authorities: Maintain their compliance databases and reporting tools, ensuring they can receive and process compliance data.

c) DevOps Team

- Oversee the deployment pipeline, ensuring continuous integration and continuous deployment (CI/CD) practices are followed.
- Manage containerization and orchestration using tools like Docker and Kubernetes to ensure scalability and modularity.
- Monitor system performance, handle load balancing, and manage failover mechanisms to ensure high availability and performance.

Version History

Version	**Week No.**	**Changes**
1.0	Week 29	Document layout and headers created
1.1	Week 29	Part 1 added
1.2	Week 30	Part 2-4 added
1.3	Week 30	Part 5-8 added
1.4	Week 31	Part 9-11 added
1.5	Week 31	Part 12-13 added
2.0	Week 31	Diagrams redrawn to unified format and styling
2.1	Week 32	Version History and Work Hour Estimation added
2.2	Week 32	References added
3.0	Week 32	Document finalized

Work Hour Estimation

Members	**Group meetings/Individual hours**				**Total hours**	
	Week 29	**Week 30**	**Week 31**	**Week 32**	**Group**	**Individual**
Trieu Huynh Ba Nguyen	1/4	2/10	2/8	1/4	6	26
Nguyen Bao Quan	1/4	2/12	2/6	1/4	6	26
Farzana Tamanna	1/4	-	-	-	1	4

References

Web-EID. (2016). *Web eID: electronic ID smart cards on the Web*. [online] Available at: https://web-eid.eu/ [Accessed 20 Jul. 2024].

Open-EID. (2024). *Architecture of ID-software*. [online] Available at: https://open-eid.github.io/ [Accessed 20 Jul. 2024].

Mojeid.cz. (2024). *MojeID Documentation*. [online] Available at: https://www.mojeid.cz/documentation/singlehtml/index.html [Accessed 21 Jul. 2024].

Nets. (2019). *eIDs*. [online] Available at: https://www.nets.eu/developer/e-ident/eids/Pages/default.aspx [Accessed 22 Jul. 2024].

ID.ee. (2024). *Arendajale - ID.ee*. [online] Available at: https://www.id.ee/rubriik/arendajale/ [Accessed 30 Jul. 2024].