Computer Science Capstone Project

QuantumZero Digital Wallet

Jonathan Race

Jeanier Anderson

Carlos Palma

14 December 2025

National University

Table of Contents

Table of Tables

Table of Figures

Abstract

The QuantumZero project introduces a Zero-Trust Decentralized Identity Wallet designed to address long-standing security and privacy challenges associated with traditional, password-based authentication systems. Modern identity practices continue to rely on centralized credential storage and repeated disclosure of personal information, creating systemic vulnerabilities that lead to credential theft, data breaches, and user privacy erosion. QuantumZero provides a mobile-first solution that leverages Decentralized Identifiers, Verifiable Credentials, and hardware-backed cryptographic protections through technologies such as the iOS Secure Enclave and Android Keystore. The platform incorporates Zero-Knowledge Proofs to enable selective disclosure, allowing users to prove specific claims without revealing unnecessary information. A supporting backend service issues, verifies, and revokes credentials using a distributed ledger or trust registry to ensure tamper-resistant validation. The system was designed and developed employing software engineering principles, secure architectural patterns, and iterative prototype testing. The result is a functional, privacy-preserving identity solution that empowers users with control over their credentials while reducing risk exposure for organizations. QuantumZero demonstrates the practical integration of emerging decentralized identity standards into a cohesive, mobile-ready authentication platform, illustrating the potential for future passwordless ecosystems built on strong cryptography and Zero-Trust principles.

Acknowledgements

The project team would like to express sincere gratitude to the faculty and staff of National University for providing the guidance, curriculum structure, and academic support that made this capstone project possible. We especially thank our instructor for their feedback and direction throughout each phase of the development process. We also acknowledge the encouragement of our families, peers, and professional colleagues, whose support allowed us to balance academic responsibilities with full-time careers. Their patience and understanding were essential in bringing the QuantumZero project to completion. Finally, we extend appreciation to the broader cybersecurity and decentralized identity communities, whose published research, open standards, and collaborative ecosystems informed the technical foundations of this work.

Chapter 1: Introduction

1.1 Project Vision

The vision of the QuantumZero project is to create a secure, decentralized, mobile-first identity wallet that fundamentally reshapes how individuals authenticate and share personal information in the digital world. The system seeks to eliminate the long-standing dependence on passwords, centralized identity stores, and repetitive personal data disclosures by providing a user-controlled mechanism for issuing, storing, verifying, and selectively sharing digital credentials. QuantumZero will achieve this by integrating Decentralized Identifiers (DIDs), Verifiable Credentials (VCs), Zero-Knowledge Proofs (ZKPs), and hardware-backed cryptographic protections into a cohesive, user-friendly mobile platform. The overarching vision is clear:

"To move beyond the risks of centralized data silos and strictly return ownership of digital identity to the user."

The primary stakeholders of this system include end users, service providers, security teams, system architects, compliance officers, and future developers who may extend or maintain the platform. End users demand privacy, control, and simplicity in how they manage their identities; service providers (Verifiers in the DID model) and enterprises urgently require secure, tamper-resistant authentication flows that eliminate reduce the burdens of storing and protecting sensitive identity data; compliance and governance stakeholders require systems that strictly adhere to modern privacy regulations such as GDPR, NIST 800-63, and zero-trust directives. Developers and maintainers require a clear and well-documented architecture that supports extensibility, security patching, and integration with evolving decentralized identity standards.

Stakeholder interviews were conducted using a semi-structured format, focusing on identifying pain points, evaluating current identity-management shortcomings, and validating priorities such as usability, privacy, and interoperability. Questions explored challenges with password overload and fatigue, trust in centralized identity providers, regulatory risks, and concerns about user burden. Interview responses emphasized the following themes:

1. Users are suffering from password fatigue and have lost trust in large, centralized identity custodians.
2. Organizations demand strong authentication but refuse the continued liability of holding user data
3. Both users and organizations expect a seamless, mobile-first experience that does not compromise on security.
4. Stakeholders prioritize transparency in how cryptographic processes operate behind the scenes.

These insights shaped the system's emphasis on simplicity, selective disclosure, and strong device-level security.

QuantumZero answers this mandate by empowering users to store keys securely on their own mobile devices using secure hardware modules such as Secure Enclave or Android Keystore, ensuring that private keys never leave the device. Service providers (Verifiers in the DID model) benefit from authentication processes that rely on verifiable, tamper-resistant Verifiable Presentations (VPs) rather than shared secrets. This fundamentally shifts the custody of the Verifiable Credential (VC) entirely to the user, thereby minimizing data retention liability. Organizations gain a system that minimizes data retention, reduces attack surface, and aligns with zero-trust principles by verifying credentials without trusting any single centralized authority.

Addressing the major tension between security requirements and the user's right to privacy, the system leverages Zero Knowledge Proofs (ZKPs) to solve the problem of "over-disclosure", the risk of sharing too much sensitive information to prove a simple fact. This allows a user to prove a specific attribute (e.g., "I am over 18") without revealing the underlying sensitive data (e.g., their date of birth on the VC). This approach effectively decouples verification from data sharing, ensuring that the system verifies the fact rather than hoarding the user's data. This directly aligns with advanced privacy regulations and the principle of selective disclosure.

To ensure longevity and sustainability past deployment, the system rejects a static 'deploy and forget' model in favor of a clearly defined maintenance plan. Maintenance responsibilities will encompass full stack oversight, including ledger or trust-registry updates, mobile application updates, cryptographic library versioning, vulnerability remediation, documentation maintenance, and continuous verification driving the compliance to DID and VC standards. Because of this complexity, the individual responsible for maintenance must possess foundational knowledge of zero-trust architectures, mobile secure storage APIs, decentralized identity frameworks, and secure backend operations. These responsibilities ensure that QuantumZero remains resilient, secure, and aligned with emerging standards throughout its entire lifecycle.

Ultimately, the vision of QuantumZero is to provide a secure, privacy-preserving identity system that places users, not institutions, at the center of digital identity. Through decentralized identity standards, advanced cryptographic protections, and zero-trust design principles, the project aims to demonstrate how next-generation digital identity technologies do not have to compromise usability for security, they can be implemented practically, intuitively, and securely in a mobile environment. QuantumZero is not just a technology demonstration; it is a commitment to a future where users are no longer passive subjects of data collection, but active owners of their digital identity.

Table 1. ***QuantumZero Identity Provider Comparison***

Feature	QuantumZero	Microsoft Entra ID	Google Identity	Apple ID
Decentralized Credentials (DID)	Yes	Yes **Enterprise**	No	No
Verifiable Credentials	Yes	Yes	No	No
Zero-Knowledge Proof	Yes	No	No	No
Selective Disclosure	Cryptographic **Strong**	Policy based	Attribute based	Minimal
Hardware Secure Enclave	Required	Supported	Supported	Required
Zero-Trust Enforcement	Cryptographic **Math Based**	Policy-Based **Conditional Access**	Risk-Based **Context Aware**	Device-Based **Hardware**
Passwordless Authentication	Yes	Yes	Yes	Yes

Public Blockchain Dependency	No **Private Trust Registry**	Optional **Bitcoin/ION**	No	No
Biometric-Bound Identity	Strong	Optional	Optional	Strong
User Data Ownership	End-User	Corporate	Corporate	Corporate
Vendor Lock-In	Low **Open Standards**	Medium	High	High
Privacy by Design	Core Principle	Compliance driven	Business driven	Ecosystem driven

1.2 Problem Statement

In a world increasingly shaped by digital transformation, identity has become both a critical asset and a profound vulnerability. Modern authentication practices still rely heavily on passwords, centralized databases, and repetitive personal data disclosures, mechanisms that are outdated, easily compromised, and fundamentally misaligned with the privacy and security expectations of today's users. Each year, billions of credentials are leaked due to database breaches, credential stuffing attacks, and phishing schemes, highlighting a systemic flaw in how digital identity is created, stored, and verified.

At the same time, global momentum is building toward decentralized identity systems, models in which users, rather than corporations or institutions, maintain control over their digital credentials. These emerging standards, based on Decentralized Identifiers (DIDs) and Verifiable Credentials (VCs), create the opportunity for secure, interoperable, passwordless authentication while minimizing the exposure of personal information. However, despite strong support from standards bodies and industry leaders, practical, user-friendly implementations remain limited. Most solutions are confined to enterprise ecosystems, lack mobile-grade security, or fail to incorporate advanced privacy-preserving technologies such as Zero-Knowledge Proofs (ZKPs).

This gap presents a significant opportunity: a secure, decentralized, privacy-preserving identity wallet that empowers individuals to manage and use their digital credentials directly from their mobile device, without ever needing a password or exposing sensitive personal data.

The proposed system addresses this need by developing a Zero-Trust Decentralized Identity Wallet called QuantumZero, a mobile platform that allows users to store, manage, and present Verifiable Credentials using cryptographic proofs rather than traditional login methods. Instead of relying on passwords or centralized user databases, the application leverages the secure hardware modules embedded in modern smartphones, such as the iOS Secure Enclave and Android Keystore, to safely generate and store private keys and biometric authentication artifacts. The system's backend service will issue, verify, and revoke VCs using a distributed ledger or simulated blockchain, enabling tamper-resistant trust without requiring any central authority.

A defining capability of this platform is its integration of Zero-Knowledge Proofs, which allows a user to prove specific claims, such as employment status, age, or organizational membership, without revealing any unnecessary information. This enables a level of privacy protection far beyond conventional identity systems, aligning the platform with zero-trust principles by ensuring that no party receives more data than is strictly required.

The value of this solution extends to multiple stakeholders.

- For end users, it provides a seamless, passwordless login experience backed by strong cryptography and device-level security, reducing the risk of credential theft and eliminating the burden of password management.
- For service providers and enterprises, it offers dramatically reduced identity-related risk exposure, improved compliance with data-minimization regulations, and a scalable model for secure authentication without storing sensitive user information.
- For developers and the broader security community, the platform demonstrates a practical, working implementation of future-oriented identity standards, DIDs, VCs, distributed ledgers, and ZKPs, brought together into a coherent, mobile-ready solution.

Ultimately, this system aims to redefine digital identity by granting individuals true ownership of their credentials while providing organizations with a secure, tamper-resistant method of authentication. By merging decentralized identity principles with modern zero-trust architectures and hardware-backed key protection, the application represents a forward-looking approach to solving one of the most persistent problems in cybersecurity: how to authenticate users securely without compromising privacy, usability, or scalability.

1.3 Project Feasibility

Paragraph Example

1.4 Draft of Client Agreement

This draft client agreement establishes a high-level understanding between the QuantumZero Project Team and CHATEAUFORGE LLC, which serves as the internal client and system owner for the QuantumZero Digital Wallet project. This agreement is intentionally non-contractual and is provided to clarify expectations, responsibilities, and acceptance boundaries for the duration of the academic capstone.

CHATEAUFORGE LLC is a Wyoming-formed limited liability company and provides the governance, security boundaries, and ownership context for all project artifacts. All source code, documentation, and supporting materials are developed and stored within environments approved by the client, including GitHub repositories, Google Drive, Google Spaces, and approved AI-assisted tooling.

The project is conducted as an academic capstone initiative and follows a defined execution window from November 23, 2025 through February 28, 2026. During this period, the QuantumZero Project Team agrees to design, implement, and document a functional prototype demonstrating decentralized identity concepts, Zero-Trust security principles, and privacy-preserving authentication mechanisms.

Acceptance of project outcomes is based on functional demonstration and documentation review rather than production readiness or commercial certification. The client acknowledges that the system is delivered as a prototype intended for educational and architectural demonstration purposes.

Regular project coordination is maintained through scheduled team meetings held on Sundays and Wednesdays, with additional communication conducted as needed through Google Spaces. Project progress, task tracking, and change management are managed using GitHub Projects and Issues.This draft agreement defines the intent and relationship between the client and project team, while detailed execution terms, deliverables, acceptance criteria, and governance processes are formally defined in Appendix F: Internal Statement of Work (SOW).

1.5 Team Structure and Capabilities

The QuantumZero project team consists of three graduate students enrolled in the Master of Science in Computer Science (MSCS) program at National University: Jonathan Race, Jeanier Anderson, and Carlos Palma. Each member contributes a complementary set of skills, perspectives, and technical expertise that align with the complexity and security requirements of a Zero-Trust Decentralized Identity Wallet. The following subsections describe the team's structure, capabilities, skill inventories, role assignments, and development plan to address identified skill gaps.

1.5.1 Team Composition and Assigned Roles

1.5.1.1 Project Manager and Lead Architect

Jonathan Race

Jonathan serves as both the Project Manager and Lead Architect. His broad background in cybersecurity, enterprise architecture, incident response, cloud engineering, and DevSecOps reflects over thirteen years of government, military, and industry experience . As Project Manager, Jonathan is responsible for scheduling, documentation quality, requirements coordination, and risk management. As Lead

Architect, he ensures that the system design adheres to Zero-Trust principles and supports core project goals, including secure key management, decentralized identity workflows, and cryptographic integrity.

Core Competencies

- Zero-Trust architecture, enterprise cybersecurity, governance
- DevSecOps automation, CI/CD engineering
- Incident response, threat analysis, vulnerability remediation
- Programming (Python, Bash, PowerShell, SQL, C, JavaScript)
- Cloud integration, containerization, and systems engineering
- Technical leadership and documentation

These skills directly support architectural design, hardware-backed cryptographic integration, backend services, and overall system direction.

1.5.1.2 Quality Assurance Manager and Cloud Security Engineer

Jeanier Anderson

Jeanier brings extensive expertise in cloud security with hands-on experience across AWS, Azure, and Google Cloud, supported by advanced certifications including CISSP and multiple Google Cloud professional credentials . In her role as Quality Assurance Manager and Cloud Security Engineer, she oversees test planning, threat modeling, vulnerability assessment, and cloud-security architecture design. She also validates the system's alignment with Zero-Trust principles and ensures that the Verifiable Credential (VC) issuance and verification services follow secure cloud and IAM best practices.

Core Competencies

- Cloud security engineering across AWS, Azure, and GCP
- IAM architecture, secrets management, security baselining
- Vulnerability assessment, penetration resistance planning
- Infrastructure as Code and secure CI/CD pipeline practices
- Threat modeling, risk assessment, and standards compliance
- Security training and technical documentation

These competencies support secure backend development, Zero-Trust validation, QA processes, and cloud-driven architecture for VC issuance and verification.

1.5.1.3 Systems Integration Engineer and Client Advocate

Carlos Palma

Carlos brings a strong engineering background specializing in automation control systems, SCADA architectures, HMI development, communication protocols, and real-time system diagnostics. His professional experience includes programming in C++, Java, and Python and designing operational interfaces for monitoring and control applications in industrial environments. As Systems Integration Engineer, Carlos contributes expertise in system logic design, interface behavior modeling, troubleshooting methodology, and workflow validation. His experience with sensor/actuator calibration, communication stacks such as Modbus and RS485/TCP, and structured diagnostics supports the project’s need for reliable data flow modeling, API integration, and robust system testing.

In his role as Client Advocate, Carlos ensures the platform’s usability by translating operational requirements into intuitive interface behavior and validating the system from an end-user perspective. His dual background in engineering and interface design positions him to analyze user workflows, identify friction points, and ensure that credential presentation and verification processes are seamless and clear.

Core Competencies

- Automation system programming and troubleshooting
- HMI design, interface logic modeling, and real-time workflows
- Programming in C++, Java, Python
- Networking, diagnostics, and communication protocols (Modbus/RS485/TCP)
- Systems integration, equipment commissioning, and operational testing
- User workflow analysis and interface evaluation

These skills enable Carlos to support system architecture translation, API-level integration, practical workflow validation, and user-centered evaluation of the QuantumZero identity wallet.

1.5.2 Constraints and Team Considerations

The team must complete the project within the three-month capstone program while balancing full-time professional obligations. Key limitations include:

- Limited prior experience with decentralized identity frameworks (DIDs, VCs, DIDComm).
- Varied familiarity with mobile secure-hardware APIs (Secure Enclave, Android Keystore).
- Need for hands-on development with Zero-Knowledge Proof (ZKP) libraries.
- Requirement to coordinate asynchronous schedules and distributed collaboration.

These constraints inform the team's structured learning plan and division of responsibilities.

1.5.3 Skills Gaps and Development Plan

1. Decentralized Identity (DIDs and Verifiable Credentials)
   - Lead: Jonathan
   - Approach: Review W3C DID/VC specifications; prototype DID creation; evaluate DIDKit or Hyperledger Aries SDKs.
   - Timeline: Weeks 1–3.
2. Zero-Knowledge Proof Implementation
   - Lead: Jeanier
   - Approach: Study ZKP frameworks (e.g., zk-SNARKs); build minimal-viable ZKP workflows; validate privacy-preserving claim proofs.
   - Timeline: Weeks 2–5.
3. Secure Enclave / Android Keystore Integration
   - Lead: Jonathan (cryptographic design), Carlos (systems integration and implementation)
   - Approach: Carlos will translate architectural requirements into functional mobile integration workflows, focusing on key storage, interface logic, and operational reliability.
   - Timeline: Weeks 3–6.
4. Distributed Ledger / Trust Registry Simulation
   - Lead: Jonathan & Carlos
   - Approach: Evaluate ledger simulation methods (hash-chains, lightweight Indy sandbox, or custom trust registry).
   - Timeline: Weeks 4–7.

These targeted learning objectives ensure adequate preparation for implementation during phase two and three.

Table 2. ***Team Role-Responsibility Summary***

Role	Team Member	Primary Responsibilities
Project Manager	Jonathan	Planning, scheduling, documentation, coordination
Lead Architect	Jonathan	System design, cryptographic architecture, identity workflows
Quality Assurance Manager	Jeanier	Security testing, threat modeling, validation
Cloud Security Engineer	Jeanier	Secure backend design, IAM, cloud controls
Client Advocate	Carlos	User-centric design, requirements alignment
Systems Integration Engineer	Carlos	API integration, interface logic modeling, systems validation, distributed ledger co-development
Research Leads	Assigned	Learning DIDs, ZKPs, secure key modules, ledger design

The QuantumZero team's combined experience across cybersecurity, cloud engineering, software development, and user-centered design provides a robust foundation for constructing a secure, decentralized identity solution. By clearly defining roles, understanding constraints, and establishing a structured skills-development plan, the team is positioned to execute a project that meets both the academic rigor of the MSCS program and the technical expectations of a modern Zero-Trust digital identity platform.

Chapter 2: Project Planning and Management

2.1 Project Plan

The QuantumZero Digital Wallet is being developed using an iterative, milestone-driven project plan aligned with the three-month capstone schedule. The work is organized into three primary phases Analysis & Design, Construction & Implementation, and Testing & Validation followed by documentation and final delivery. Each phase includes defined tasks, owners, and time estimates that take into account the team's full-time professional responsibilities and the learning curve associated with decentralized identity, Zero-Trust security, and mobile secure-hardware APIs. Project coordination and tracking are managed using GitHub Projects and Issues, with recurring team meetings held twice per week and ad-hoc collaboration through online messaging and shared documentation. The goal of this plan is to ensure delivery of a functional, well-documented prototype while minimizing schedule risk and technical debt.

2.1.1 Project Overview and Assumptions

QuantumZero is being developed as a capstone project with CHATEAUFORGE LLC acting as both the sponsoring client and security boundary owner. All source code, documentation, artifacts, and data are stored within environments controlled or explicitly approved by the client, including GitHub repositories, Google Drive, Google Spaces, and approved AI-assisted tooling. No third-party environments outside these boundaries are assumed for production data storage.

The project timeline spans November 23, 2025 through February 28, 2026, with the team currently entering Week 4. The project follows an iterative, milestone-based execution model, emphasizing analysis and design completion early in the term, followed by prototype construction, validation, and documentation finalization. Key non-obvious assumptions informing estimates include:

- Team members are balancing full-time professional obligations alongside capstone work.
- Several technologies (DIDs, ZKPs, mobile secure enclaves) require structured learning time.
- The system is delivered as a functional prototype, not a commercial production deployment.
- The client and development team are effectively the same organizational entity, reducing contractual friction but increasing the importance of clear internal acceptance criteria.

2.1.2 Task List and Checklist

To structure execution, the project work is organized into five major task areas: **analysis and design, learning and research, implementation, testing and validation, and documentation and delivery**. This phased structure allows the team to progress in a disciplined and logical sequence, ensuring that foundational architectural decisions are made early, technical knowledge gaps are addressed before development begins, and the prototype evolves in a controlled, predictable manner. Each task area is further broken down into detailed subtasks that guide day-to-day work activities and prevent scope drift. These subtasks are continuously tracked using GitHub Projects, Issues, and Milestones, enabling the team to visualize workflow, identify bottlenecks, reassign work when needed, and maintain transparency throughout the capstone timeline. The checklist format also supports incremental progress reviews during weekly meetings, ensuring that every requirement whether technical, design-related, or documentation-driven can be monitored, updated, and marked complete as the project advances from concept to prototype implementation.

2.1.2.1 Analysis and Design Tasks

- Finalize functional and nonfunctional requirements
- Complete system architecture diagrams (DFDs, class diagrams)
- Define trust registry / ledger simulation model
- Define mobile wallet architecture and secure key workflows
- Define acceptance criteria per functional capability

2.1.2.2 Learning and Research Tasks

- Study W3C DID and VC specifications
- Evaluate DID libraries and SDKs
- Research Zero-Knowledge Proof frameworks
- Review Secure Enclave and Android Keystore integration patterns

2.1.2.3 Implementation Tasks

- Mobile wallet prototype (DID generation, VC storage, VP creation)
- Backend issuance and verification services
- Trust registry or ledger simulation
- Zero-Knowledge Proof integration (selective disclosure)
- Administrative dashboard prototype

2.1.2.4 Testing and Validation Tasks

- Unit testing for DID, VC, and ZKP workflows
- Security validation and threat modeling

- Integration testing between mobile and backend components
- Offline verification and QR-based workflows

2.1.2.5 Documentation and Delivery Tasks

- User Manual and Maintenance Manual
- System architecture documentation
- Test suite documentation
- Final report refinement and formatting
- Appendix completion and review

2.1.3 Task Assignments and Contingencies

Table 3. ***Task Assignments Summary***

Task Area	Primary Owner	Secondary
Project management & scheduling	Jonathan Race	Jeanier Anderson
Architecture & cryptographic design	Jonathan Race	Carlos Palma
Cloud security & QA	Jeanier Anderson	Jonathan Race
Systems integration & workflows	Carlos Palma	Jonathan Race
ZKP research & validation	Jeanier Anderson	Carlos Palma
Documentation & report integration	Jonathan Race	All team members

2.1.4 Estimation of Remaining Work

Analysis & Design Phase (Remaining):

- Effort: ~40–50 total hours
- Activities: Requirements finalization, diagrams, acceptance criteria

Construction & Implementation Phase:

- Effort: ~120–150 total hours

- Activities: Mobile prototype, backend services, ledger simulation, ZKP integration

Testing & Validation Phase:

- Effort: ~40–50 total hours
- Activities: Unit, integration, and security testing

2.1.5 Overhead Estimation

Table 4. ***Estimates Times for Overhead***

Activity	**Estimated Effort**
Project management & coordination	20–25 hours
Client/team meetings	25–30 hours
Learning new technologies	30–40 hours
Code reviews & design reviews	15–20 hours
Report writing & revision	30–35 hours
Integration & system testing	20–25 hours

2.1.6 Loss of Available Effort

Predictable reductions in available effort include:

- Federal and personal holidays
- End-of-year scheduling constraints
- Professional work travel or on-call duties
- Estimated loss: 10–15% of total available project time.

2.1.7 Realistic Commitments (Next Two Months)

The team commits to delivering:

- A fully documented functional prototype of the QuantumZero Digital Wallet
- Working DID, VC, and VP flows with selective disclosure
- Secure mobile key storage using platform hardware APIs

- Backend issuance, verification, and revocation services
- Administrative monitoring interfaces
- Complete system documentation and test artifacts

The project does not commit to:

- Production-grade scalability guarantees
- Public blockchain deployment
- Formal third-party security certification

2.2 Risk Analysis

Paragraph Example

2.3 Problem Analysis

Paragraph Example

Chapter 3: Requirements and Design

3.1 Functional and Non-Functional Requirements

3.1.1 Statement of Nonfunctional Requirements

The QuantumZero system must meet a set of operational constraints and nonfunctional requirements that ensure the platform performs reliably, securely, and consistently across mobile devices and backend services. These requirements address performance benchmarks, security safeguards, backup and recovery expectations, system support needs, OS and platform limitations, and network dependencies. Because the system integrates decentralized identity standards, Zero-Trust principles, and hardware-backed cryptography, these nonfunctional constraints establish the necessary foundation for secure and privacy-preserving identity operations.

3.1.1.1 Nonfunctional Requirements Table

Category	ID	Requirement
Operation & Performance	NF-OP-01	The system must respond to all standard mobile app actions (VC storage, VP creation, DID generation) within less than 1 second under normal conditions.
Operation & Performance	NF-OP-02	Backend issuance and verification APIs must respond within 500 ms under normal conditions.
Operation & Performance	NF-OP-03	Backend system availability must meet or exceed 99% uptime.
Security	NF-SEC-01	All private keys must be generated and stored within hardware-backed secure storage (iOS Secure

		Enclave, Android Keystore StrongBox).
Security	NF-SEC-02	All communication between mobile app, backend, and external services must use TLS 1.3 or higher.
Security	NF-SEC-03	The platform must enforce Zero-Trust security principles by authenticating and authorizing every backend request.
Security	NF-SEC-04	DID and VC operations must comply with W3C specifications for decentralized identifiers and verifiable credentials.
Security	NF-SEC-05	Zero-Knowledge Proof workflows must ensure selective disclosure and prevent exposure of unnecessary user data.
Compliance	NF-SEC-06	The system must minimize data collection to comply with GDPR data minimization principles.
Compliance	NF-SEC-07	Authentication strength must align with NIST 800-63 identity assurance guidelines.
Backup/Restore	NF-BR-01	User credentials must never be backed up to QuantumZero servers.
Backup/Restore	NF-BR-02	Users must be able to export an encrypted backup of credentials to personal cloud providers.
Backup/Restore	NF-BR-03	Backend trust-registry and issuance metadata must

		be backed up daily with a retention period of 30 days.
Backup/Restore	NF-BR-04	Restore procedures must include cryptographic integrity verification through hashing.
Support	NF-SUP-01	System logs, metrics, and diagnostics must be accessible through an administrative dashboard.
Support	NF-SUP-02	System documentation must include user guides, admin manuals, and maintenance manuals.
Support	NF-SUP-03	Backend services must include health endpoints for uptime and availability monitoring.
Platform	NF-PLT-01	The mobile application must support iOS 16+ and Android 13+ to ensure secure hardware availability.
Platform	NF-PLT-02	Backend services must be deployable using containerized infrastructure (Docker/Kubernetes).
Platform	NF-PLT-03	All system components must be modular, allowing future adoption of emerging decentralized wallet technologies.
Network	NF-NW-01	The mobile application must support offline operation for VP generation when internet connectivity is unavailable.
Network	NF-NW-02	DID resolution must support both remote queries

		and local caching.
Network	NF-NW-03	API rate-limiting must be enforced to prevent denial-of-service indicators.
Extensibility	NF-EXT-01	The system must support future identity standards and extensible DID methods.

3.1.1.2 Strategies to Meet Nonfunctional Requirements

To ensure QuantumZero meets the strict performance and security benchmarks defined above, the following technical strategies will be implemented:

- **Performance Optimization Strategy:** To meet the <1 second response time (NF-OP-01), the mobile application will utilize local-first architecture. Verifiable Credentials will be stored in the native filesystem (SQLite/CoreData) rather than fetched from the network for every presentation. Backend latency (NF-OP-02) will be minimized by deploying services via Docker containers on auto-scaling infrastructure, ensuring resource availability during high-load events.
- **Security & Zero-Trust Implementation:** We will satisfy NF-SEC-01 by strictly enforcing the use of iOS CryptoKit (Secure Enclave) and Android Jetpack Security (Keystore) for all key generation. Private keys will be flagged as non-exportable at the hardware level. To meet NF-SEC-02, all network traffic will be pinned to TLS 1.3 certificates to prevent man-in-the-middle attacks.
- **Backup & Compliance Strategy:** To adhere to GDPR data minimization (NF-SEC-06), the backend will not store raw PII. Instead, it will only store cryptographic hashes and DID documents. User-side backups (NF-BR-02) will be

implemented using AES-256 encryption derived from a user-held recovery phrase, ensuring that even if the backup is intercepted, the data remains inaccessible to the QuantumZero team.

3.1.1.3 Maintenance and Support

Post-deployment maintenance will include mobile app updates, backend security patches, dependency version updates, cryptographic library updates, and ongoing monitoring of ledger integrity. The support process must also include periodic vulnerability assessments, documentation reviews, and the upkeep of trust-registry entries. Routine activities include verifying DID method compatibility, reviewing revocation registries, and ensuring backward compatibility for updated mobile OS versions.

3.1.1.4 Client-Side Responsibility for Support

In a production environment, ongoing system support would be managed by a designated Client-Side System Administrator or Technical Owner responsible for overseeing operational continuity and system governance. This role would include managing configuration updates, reviewing system logs, maintaining credential templates, monitoring trust-registry integrity, and coordinating security updates. Because this project does not specify an external client organization, the responsibility is described in terms of the required functional role rather than a specific individual, ensuring that the system can be adopted by any entity capable of maintaining decentralized identity infrastructure.

3.1.1.5 Capabilities of the Responsible Person

The individual fulfilling the System Administrator or Technical Owner role should possess the following capabilities:

- Foundational understanding of decentralized identity concepts, including Decentralized Identifiers (DIDs), Verifiable Credentials (VCs), and trust registries.
- Working knowledge of Zero-Trust security principles and modern secure authentication models.
- Familiarity with mobile secure storage mechanisms such as Secure Enclave and Android Keystore.
- Ability to monitor and interpret backend system logs, metrics, and diagnostic outputs.
- Basic competence in cloud-based infrastructure, containerized services, and API operation.
- Understanding of cryptographic workflows, including key lifecycle management, signature verification, and credential revocation.

These capabilities ensure the administrator can support the system effectively, troubleshoot operational issues, and uphold the integrity of decentralized identity processes.

3.1.1.6 Training Requirements

To prepare the System Administrator or Technical Owner for effective long-term system maintenance, the following training sessions would be required:

1. System Architecture and Workflow Overview

   - DID lifecycle and VC issuance/verification flows.
   - Interactions between the mobile wallet, secure hardware modules, backend services, and the trust registry.

2. Administrative Dashboard and Monitoring Tools
   - Viewing and interpreting logs.
   - Tracking revocation events.
   - Monitoring API health and system diagnostics.
3. Backup and Restore Procedures
   - Managing server-side ledger/registry backups.
   - Understanding user-side encrypted backup/export mechanisms for credentials
4. Security and Compliance Training
   - Proper handling of cryptographic material.
   - Privacy expectations and selective disclosure using Zero-Knowledge Proofs (ZKPs).
   - Understanding common threat models and mitigation strategies.
5. Update, Patch, and Version Maintenance
   - Applying backend service patches and updates.
   - Tracking changes to cryptographic libraries and evolving identity standards.
   - Understanding mobile OS updates that may affect secure hardware APIs.

Completion of this training ensures that the support role can maintain system stability, uphold identity integrity, and ensure compliance with evolving decentralized identity standards.

3.1.2 Statement of Functional Requirements

The functional requirements define the operational behaviors and capabilities that the QuantumZero system must deliver to support decentralized identity workflows. These requirements specify the essential functions for generating decentralized identifiers, issuing and verifying credentials, producing verifiable presentations, enforcing security controls, supporting user workflows, and enabling administrative oversight. Together, these requirements establish the complete functional scope of the system and ensure that QuantumZero provides a secure, reliable, and user-controlled decentralized identity solution.

3.1.2.1 Functional Requirements Table

Category	ID	Requirement
Operation & Performance	F-OP-01	The system must allow users to generate decentralized identifiers (DIDs).
Operation & Performance	F-OP-02	Users must be able to receive, store, and view verifiable credentials (VCs).
Operation & Performance	F-OP-03	Users must be able to generate verifiable presentations (VPs).
Operation & Performance	F-OP-04	Users must be able to share VPs via QR code or secure data transfer.

Operation & Performance	F-OP-05	The app must display visual progress indicators (spinners/bars) for any operations taking longer than 500ms.
Security	F-SEC-01	Users must authenticate using biometrics or device PIN to access the wallet.
Security	F-SEC-02	Private keys must be generated using native secure hardware APIs. ● Keys must never leave the secure module in plaintext. ● Keys must be non-exportable and hardware-bound.
Security	F-SEC-03	The backend must cryptographically verify signatures on VCs using W3C methods.
Security	F-SEC-04	The system must validate VC revocation status during every verification request.
Security	F-SEC-05	The system must support zero-knowledge proof generation for selective disclosure.
Security	F-SEC-06	The system must enforce nonce-based or timestamp-based replay protection for all verification requests.
Backup/Restore	F-BR-01	Users must be able to export an encrypted backup

		package of their credentials and keys.
Backup/Restore	F-BR-02	Users must be able to restore credentials on a new device following re-authentication.
Backup/Restore	F-BR-03	Administrators must be able to restore trust-registry and system configuration backups.
Backup/Restore	F-BR-04	Backup packages must include integrity verification (checksum or signature) to detect tampering
Support	F-SUP-01	Administrators must be able to manage issuers, credential templates, and revocation lists.
Support	F-SUP-02	The system must log issuance and verification events for auditing.
Support	F-SUP-03	The system must provide automated test suites for DIDs, VCs, and ZKPs.
Support	F-SUP-04	The system must expose health-check endpoints for monitoring component status.
OS & Platform	F-PLT-01	The mobile app must integrate with platform-specific secure storage APIs.
Network	F-NW-01	Backend services must expose standardized REST or DIDComm-style interfaces.
Network	F-NW-02	The system must support online credential

		verification through the backend.
Network	F-NW-03	The system must support offline verification using QR-encoded VPs where possible.
Network	F-NW-04	The system must support trust-registry lookups to verify issuer legitimacy.
Network	F-NW-05	The mobile app must maintain a cached copy of the trusted ledger, as applicable to the end-users DID for offline validation.
Usability	F-USR-01	The mobile UI must provide intuitive workflows requiring no more than three user actions per credential operation.
Usability	F-USR-02	Users must receive clear warnings for revoked, expired, or invalid credentials.
Usability	F-USR-03	The UI must meet basic ADA accessibility standards (contrast, readable text, screen-reader compatibility).
Other Relevant Aspects	F-ORA-01	Users must be able to securely delete stored credentials.
Other Relevant Aspects:	F-ORA-02	The system must provide notifications when credentials are revoked or updated.
Other Relevant	F-ORA-03	The system must allow users to review and manage

Aspects		app permissions and data-sharing preferences.

3.1.2.2 Strategies to Meet Functional Requirements

The following mechanisms describe how the functional capabilities of the system will be technically realized:

- Identity Management Mechanisms: The generation of DIDs (F-OP-01) will be handled using W3C DID Core compliant libraries (e.g., Hyperledger Aries components) and cryptographic primitives. Verifiable Credentials (VCs) will be verified (F-SEC-03) using a cryptographic library that supports zk-SNARK validation and W3C VC signature methods. This ensures a high security-to-performance ratio suitable for mobile and backend processing.
- Zero-Knowledge Proof Integration (F-SEC-05): The system will implement ZKPs using the Mopro toolkit for cross-platform proving and the Noir DSL for circuit definition. The verification logic will be executed efficiently on the Go/Rust backend using Gnark, ensuring selective disclosure and compliance with privacy requirements.
- User Workflow Support: To support offline verification (F-NW-03), the system will implement QR code data transfer using the ISO 18013-5 (mDL) engagement model. This allows a Verifier to scan a User's device directly to receive a Verifiable Presentation (VP) without requiring an internet connection.
- Platform & OS Compatibility: Platform constraints (F-PLT-01) are addressed by building the application in React Native, allowing a single codebase to interface

with native modules (Swift for iOS, Kotlin for Android) where hardware-specific security features (NF-SEC-01) are required.

- Operational Support Features: To satisfy support requirements (F-SUP-02), the backend will integrate structured logging (JSON format). This allows administrators to query issuance and verification failures via the dashboard without accessing user Personally Identifiable Information (PII).
- Automated Testing (F-SUP-03): Automated test suites will be employed to ensure functional consistency across the DID lifecycle, VC format validation, and ZKP integrity checks. This testing will specifically include ZKP soundness and completeness checks to validate cryptographic integrity.

3.2 Data Flow Diagrams

Paragraph Example

Figure 1. ***Example figure diagram***

3.3 Class Diagrams and Database Design

Paragraph Example

Chapter 4: Project Implementation

4.1 User interface Design

Paragraph Example

4.2 System Implementation (or Prototype)

Paragraph Example

4.3 Defect Tracking

Paragraph Example

4.4 Version and Change Control

Paragraph Example

Chapter 5: Conclusions and Recommendations

5.1 Summary of the Project

Paragraph Example

5.2 Project History and Lessons Learned

Paragraph Example

5.3 Statement of Results

Paragraph Example

5.4 Future Work

Paragraph Example

References

Apple. (n.d.). *Protecting keys with the Secure Enclave*. Apple Developer Documentation. https://developer.apple.com/documentation/security/protecting-keys-with-the-secure-enclave

D. Benarroch, L. Brandão, M. Maller, and E. Tromer (Ed.). (2022, July 17). ZKProof Community Reference. *ZKProof*, *Version 0.3*. https://docs.zkproof.org/reference

Decentralized Identifiers (DIDs) v1.0: Core architecture, data model, and representation. (n.d.). W3C. https://www.w3.org/TR/did-core/

Goldwassser, S., Micali, S., & Rackoff, C. (1989, February). The Knowledge Complexity of Interactive Proof Systems. *1989 Society for Industrial and Applied Mathematics*, *18*(1), 186-208. http://people.csail.mit.edu/silvio/Selected%20Scientific%20Papers/Proof%20Systems/The_Knowledge_Complexity_Of_Interactive_Proof_Systems.pdf

Google. (2025, April 17). *Android Keystore system | Security*. Android Developers. https://developer.android.com/privacy-and-security/keystore

Hyperledger Aries. (n.d.). *hyperledger-aries/aries: allows trusted online peer-to-peer interactions based on decentralized identities and verifiable credentials*. GitHub. https://github.com/hyperledger-aries/aries

Hyperledger Indy. (2024). *Hyperledger Indy: Hyperledger Indy provides tools, libraries, and reusable components for providing digital identities rooted on blockchains or other distributed ledgers so that they are interoperable across administrative domains, applications, and any ot*. Read the Docs. https://hyperledger-indy.readthedocs.io/en/latest/

OpenWallet Foundation. (n.d.). *openwallet-foundation/bifold-wallet: designed to enhance the way we interact with digital identities, making the process both secure and user-friendly*. GitHub. https://github.com/openwallet-foundation/bifold-wallet

SpruceID. (n.d.). *spruceid/didkit: A cross-platform toolkit for decentralized identity*. GitHub. https://github.com/spruceid/didkit

SpruceID. (n.d.). *spruceid/sprucekit-mobile: Libraries and examples for integrating verifiable credentials (VC) and mobile driver's licenses (mDL) into mobile apps.* GitHub. https://github.com/spruceid/sprucekit-mobile

SpruceID. (n.d.). *spruceid/ssi: The SSI library provides a simple and modular API to sign and verify claims exchanged between applications using Decentralized Identifiers (DIDs)*. GitHub. https://github.com/spruceid/ssi/

Verifiable Credentials Data Model v1.1. (2022, March 3). W3C. https://www.w3.org/TR/vc-data-model-1.1/

ZoKrates. (n.d.). *zokrates/zokrates: A toolbox for zkSNARKs on Ethereum*. GitHub. https://zokrates.github.io/

Appendix A: User’s Manual

Paragraph Example

Appendix B: Maintenance Manual

Paragraph Example

Appendix C: Additional Design Documentation

Paragraph Example

Appendix D: Sample Source Code

Paragraph Example

Appendix E: Test Suite

Paragraph Example

Appendix F: Internal Statement of Work (SOW)

F.1 Purpose

This Internal Statement of Work (SOW) defines the scope, responsibilities, deliverables, acceptance criteria, and governance structure for the **QuantumZero Digital Wallet** project. This SOW establishes a formal internal agreement between **CHATEAUFORGE LLC**, acting as the Client and system owner, and the QuantumZero Project Team, acting as the Development Team.

This document is intended to promote transparency, accountability, and structured execution while recognizing that the system is developed as an academic capstone prototype and not a commercial production deployment.

F.2 Parties

Client	**Development Team**
CHATEAUFORGE LLC	QuantumZero Capstone Project Team
Wyoming Limited Liability Company	Master of Science in Computer Science
	National University

F.3 Project Duration

- Start Date: November 23, 2025
- End Date: February 28, 2026
- Current Phase: Week 4 (Analysis and Design nearing completion)

F.4 Scope of Work

The Development Team shall design, implement, and document a Zero-Trust Decentralized Identity Wallet prototype named QuantumZero.

F.4.1 In-Scope Work

The following activities and components are included within the scope of this SOW:

- Design and development of a mobile identity wallet prototype supporting:
  - Decentralized Identifier (DID) generation
  - Verifiable Credential (VC) storage and management
  - Verifiable Presentation (VP) creation and sharing
- Integration of hardware-backed key storage using:
  - iOS Secure Enclave
  - Android Keystore
- Development of backend services for:
  - Credential issuance
  - Credential verification
  - Credential revocation
- Implementation of a trust registry or distributed ledger simulation
- Integration of Zero-Knowledge Proof (ZKP) selective disclosure workflows
- Development of administrative and monitoring interfaces
- Creation of system documentation and test artifacts

F.4.2 Out-of-Scope Work

The following items are explicitly excluded from this SOW:

- Production-grade scalability guarantees
- Deployment to a public blockchain network
- Formal third-party security certification or audits
- Commercial support agreements or service-level agreements (SLAs)

F.5 Deliverables

The Development Team shall provide the following deliverables by the conclusion of the project term:

1. Functional QuantumZero system prototype (mobile and backend components)
2. Source code repositories hosted within client-approved environments
3. System architecture and design documentation
4. User Manual
5. Maintenance Manual
6. Test suite and validation artifacts
7. Final capstone project report and supporting appendices

F.6 Acceptance Criteria

Deliverables shall be considered accepted when the following conditions are met:

- Core functional requirements operate as documented
- Security requirements are demonstrably enforced, including:
    - Hardware-backed cryptographic key storage
    - Encrypted communications using modern TLS
    - Zero-Knowledge Proof–based selective disclosure
- Primary use cases can be demonstrated end-to-end
- Documentation accurately reflects system design, behavior, and limitations
- Known constraints and prototype limitations are clearly disclosed

Acceptance is based on functional demonstration and documentation review, not on production readiness or certification.

F.7 Roles and Responsibilities

F.7.1 Client Responsibilities

The Client shall be responsible for:

- Providing architectural direction and project governance
- Defining priorities and acceptance expectations
- Maintaining ownership of all intellectual property produced
- Providing approved tooling, storage, and collaboration environments

F.7.2 Development Team Responsibilities

The Development Team shall be responsible for:

- Designing and implementing all system components
- Maintaining secure handling of code, data, and artifacts
- Producing accurate and complete documentation
- Tracking progress, tasks, and changes using GitHub Projects and Issues
- Communicating status and risks through scheduled and ad-hoc meetings

F.8 Data Ownership and Security Boundaries

- All source code, documentation, and artifacts produced during the project are the intellectual property of **CHATEAUFORGE LLC**, unless otherwise specified. Applicable rights are retained by the development team for work created.
- All project materials must remain within client-approved security boundaries, including GitHub, Google Drive, Google Spaces, and approved AI tooling.
- Private cryptographic keys and sensitive credential data must never be stored outside approved environments or transmitted insecurely.

F.9 Assumptions and Constraints

This SOW is based on the following assumptions and constraints:

- The system is developed as an academic capstone prototype.
- Team members are balancing project work with full-time professional obligations.
- Certain technologies (e.g., decentralized identity frameworks, Zero-Knowledge Proofs, secure hardware APIs) require learning and research time.
- Project schedules may be adjusted due to holidays, professional commitments, or unforeseen technical complexity.

F.10 Change Management

Changes to scope, schedule, or deliverables shall be documented and tracked using GitHub Issues and Project boards. Significant changes will be reflected in updated documentation and communicated during scheduled project meetings.

F.11 Termination

This Internal Statement of Work shall terminate automatically upon completion of the project term on February 28, 2026, or earlier by mutual agreement of the Client and Development Team.

CHATEAUFORGE LLC		QuantumZero Project Team	
Signatures:	______________	Signatures:	______________
	______________		______________
	______________		______________
Names:	______________	Names:	______________
	______________		______________
	______________		______________
Date:	______________	Date:	______________