Software requirements specification [SRS]
Project – PCS26-18

1) SRS Overview

This Software Requirements Specification (SRS) document defines the functional and technical aspects of a decentralized online meeting platform aimed at enhancing security, privacy, and usability. The platform integrates blockchain technology for immutable logging, decentralized storage via IPFS, and blockchain-based role-based access control. Key features include end-to-end encryption, real-time video conferencing powered by WebRTC, and robust user authentication mechanisms. The system is designed to be scalable, user-friendly, and adaptable for diverse industries while ensuring compliance with privacy regulations and data integrity standards.

2) Purpose

The purpose of this SRS is to outline the requirements for developing a decentralized online meeting platform that ensures secure, private, and robust communication for stakeholders in institutes, organizations, and government bodies. By leveraging blockchain technology, the platform aims to protect sensitive data, ensure transparency through immutable logging, and enable secure decentralized storage. The document serves as a guide for developers, testers, and stakeholders to align on the system's features, functionality, and architecture.

3) Scope

The following software products will be identified and integrated into the decentralized online meeting platform:

- **WebRTC**: Enables real-time video and audio communication between participants, providing low-latency, high-quality conferencing capabilities.
- **IPFS (InterPlanetary File System)**: Facilitates decentralized storage for meeting recordings and shared files, ensuring data integrity and availability across distributed nodes.
- **Blockchain (Solana)**: Ensures secure, immutable logging of meeting activities, implementing role-based access control and decentralized identity verification for participants.
- **End-to-End Encryption**: Encrypts all communication (video, audio, file transfers) to ensure that meeting data remains confidential and protected from unauthorized access.
- **OAuth**: Handles secure user authentication and authorization, integrating with blockchain for secure, tamper-proof identity verification.
- **SRTP**: Handles secure transmission of media packets over network using AES-256 encryption.

- The platform will deliver:
- **Decentralized Security and Security**: Leverage blockchain technology for secure, immutable logging of meeting data and actions.
- **End-to-End Encryption**: Ensure all communications, including video, audio, and file sharing, are encrypted in transit and at rest.
- **Role-Based Access Control**: Utilize blockchain-based identity verification for secure authentication and controlled access to resources.
- **Scalable Architecture**: Build a system that is adaptable for varied use cases, including education, healthcare, and business environments.
- **User-Friendly Design**: Provide intuitive interfaces for scheduling, joining meetings, managing settings, and accessing recordings.

4) Product Perspective

The decentralized online meeting platform is a standalone system but integrates seamlessly with larger organizational IT ecosystems. It interacts with other related products such as user management systems, blockchain networks, and video conferencing tools, forming a cohesive and secure collaboration environment.

4.1) System Interfaces

- APIs for interaction with blockchain, IPFS, and external identity providers.
- Data exchange through REST APIs, ensuring secure and efficient communication.

4.2) User Interfaces

- Web and mobile applications with intuitive designs, accessible through browsers and native apps.

4.3) Hardware Interfaces

- Requires devices with a camera, microphone, and stable internet connectivity.

4.4) Software Interfaces

- Interacts with WebRTC, IPFS, blockchain SDKs, and authentication frameworks like OAuth. Use of MongoDB Atlas Database for storage of data.

4.5) Communication Interfaces

- Secure communication protocols (e.g., HTTPS, SRTP) for data transmission.
- Peer-to-peer communication for video and audio streaming via WebRTC.

4.6) Memory

- Minimal local storage, leveraging cloud and decentralized networks for scalability.

4.7) Operations

- Real-time meeting setup, scheduling, and logging.
- Blockchain ensures tamper-proof logs and secure operations.

4.8) Site Adaptation Requirements

- Adaptable to organizational environments with minor configuration for branding, user roles, and compliance needs.

5) Product Functions

The decentralized online meeting platform will perform the following major functions:

- **User Management**: User registration, authentication, and role-based access control using blockchain-based identity verification.
- **Meeting Scheduling and Management**: Schedule, start, and manage online meetings with role-based participation and access rights.
- **Real-Time Communication**: Enable high-quality video, audio, and text-based interactions using WebRTC.
- **File Sharing**: Allow secure, encrypted sharing of documents during meetings.
- **Immutable Logging**: Use blockchain to maintain a tamper-proof record of meeting actions and events.
- **Meeting Recordings**: Store recordings and files on decentralized storage systems like IPFS for secure, scalable access.
- **Security and Privacy**: Provide end-to-end encryption for communication and ensure compliance with privacy regulations.
- **User Interface**: Intuitive web and mobile interfaces for managing settings, accessing recordings, and participating in meetings.
- **Notifications and Alerts**: Send reminders and updates for scheduled meetings and changes.

6) User Characteristics

The platform targets the following user groups with varying levels of technical expertise:

- **Administrative Users**: System administrators, IT staff.
- **Organizational Users**: Faculty, managers, professionals.
- **General Participants**: Students, employees, stakeholders.

7) Limitations

- **Scalability**:
  - ➢ While the system is designed to be scalable, increased blockchain transaction volume and IPFS storage usage may lead to higher latency and costs.
  - ➢ Hosting a large number of concurrent users may require significant infrastructure scaling.
- **Internet Availability**:
  - ➢ The platform requires stable, high-speed internet for seamless video conferencing and real-time communication. Users in regions with poor connectivity may face disruptions.
- **IPFS Storage Costs**:
  - ➢ Decentralized storage via IPFS, while secure, can incur additional costs as storage requirements grow due to meeting recordings and file sharing.
- **Blockchain Transaction Latency**:
  - ➢ Blockchain operations, such as logging actions or verifying identities, may introduce delays due to network congestion or transaction confirmation times.
- **Resource Constraints**:
  - ➢ Devices with low processing power or outdated hardware may struggle to handle WebRTC-based real-time communication and blockchain operations effectively.

8) Assumptions and Dependencies

1. **Platform Availability**: Assumes access to blockchain networks, IPFS, and WebRTC-compatible devices.
2. **Third-Party Integrations**: Dependence on stable APIs for OAuth and decentralized storage.
3. **Infrastructure**: Availability of reliable internet and cloud services for seamless operations.
4. **Regulatory Stability**: Assumes no major changes in compliance requirements like GDPR or HIPAA.
5. **User Expertise**: Assumes users have basic familiarity with online meeting tools and devices.
6. **Device Compatibility**: Assumes hardware capabilities to support video conferencing and encryption mechanisms.

9) Apportioning of Requirements

The major requirements are allocated to the following software elements:

Requirement	**Software Element**
User Authentication	Blockchain, OAuth, JWT
Real-Time Communication	WebRTC, Socket.io
Decentralized Storage	IPFS
Meeting Scheduling & Management	Backend APIs, Calendly and Google Calendar
Role-Based Access Control	Blockchain, Middleware
End-to-End Encryption	WebRTC, Encryption Libraries such as SRTP
Logging and Auditing	Blockchain, Backend APIs, Winston-Morgan
User Interface	Frontend (Web and Mobile Apps)

10) Specified Requirements

Functional Requirements:

1. Authenticate users with login credentials, OAUTH and blockchain, output access tokens.
2. Schedule/manage meetings and store details in the database.
3. Enable real-time communication with WebRTC and support encrypted streams such as SRTP.
4. Allow file sharing via IPFS, Cloudinary and MongoDB with secure access.

Cross-References: OAuth for authentication, IPFS for storage, WebRTC for communication.

Each requirement is uniquely identifiable and traceable.

11) External Interfaces

1. Authentication Interface

- **Name of item:** Authentication API
- **Description of purpose:** Provides secure login and user verification using JWT.
- **Source of input or destination of output:**
  - Input: User credentials (email, password, or biometric data).
  - Output: Access tokens or error messages for invalid credentials.
- **Valid range, accuracy, and/or tolerance:**
  - Input must match predefined formats for emails or hashed passwords.
- **Units of measure:** N/A (digital strings).
- **Timing:** Real-time authentication (within 500ms).

- **Relationships to other inputs/outputs:** Access tokens link to session management APIs.
- **Data formats:**
  - Input: JSON with fields like {"email": "user@example.com", "password": "hashedPassword"}.
  - Output: JSON tokens (e.g., JWT).
- **Command formats:** RESTful API commands like POST /auth/login.
- **Data items/information included in the input and output:**
  - Input: Email, hashed password.
  - Output: User token, success status, or error messages.

2. Meeting Creation Interface

- **Name of item:** Meeting Management API
- **Description of purpose:** Enables users to create, schedule, and invite participants to secure online meetings.
- **Source of input or destination of output:**
  - Input: Meeting details (title, participants, date/time).
  - Output: Confirmation and blockchain transaction ID.
- **Valid range, accuracy, and/or tolerance:**
  - Dates must follow the format YYYY-MM-DDTHH:MM:SS.
  - Participant count: 1–1000.
- **Units of measure:**
  - Time in UTC format.
- **Timing:** Response time within 1 second.
- **Relationships to other inputs/outputs:** Links to user authentication and blockchain validation.
- **Data formats:**
  - Input: JSON, e.g., {"title": "Team Meeting", "participants": ["user1@example.com"], "datetime": "2024-12-01T10:00:00"}.
  - Output: JSON confirmation: {"status": "success", "meetingID": "123456789", "blockchainID": "0xabc123..."}.
- **Command formats:** RESTful API commands like POST /meetings/create.
- **Data items/information included in the input and output:**
  - Input: Title, participant emails, date/time.
  - Output: Meeting ID, blockchain transaction ID.

3. Blockchain Verification Interface

- **Name of item:** Blockchain Verification API
- **Description of purpose:** Verifies meeting data integrity and participant authenticity using blockchain.
- **Source of input or destination of output:**
  - Input: Meeting ID or transaction ID.
  - Output: Verification status and associated metadata.

- **Valid range, accuracy, and/or tolerance:**
    - Blockchain hashes must match recorded data with 100% integrity.
- **Units of measure:** N/A (digital hashes).
- **Timing:** Verification complete within 2 seconds.
- **Relationships to other inputs/outputs:** Links to authentication and meeting management APIs.
- **Data formats:**
    - Input: JSON, e.g., {"meetingID": "123456789", "blockchainID": "0xabc123..."}.
    - Output: JSON verification: {"status": "verified", "metadata": {...}}.
- **Command formats:** RESTful API commands like GET /blockchain/verify.
- **Data items/information included in the input and output:**
    - Input: Meeting ID, blockchain ID.
    - Output: Verification status, metadata.

4. Data Streaming Interface

- **Name of item:** Secure Video/Audio Streaming API
- **Description of purpose:** Enables encrypted video/audio communication during meetings.
- **Source of input or destination of output:**
    - Input: Video/audio data streams from participants.
    - Output: Real-time encrypted video/audio streams.
- **Valid range, accuracy, and/or tolerance:**
    - Frame rate: 30–60 fps.
    - Audio bitrate: 32–320 kbps.
- **Units of measure:** Mbps for streaming bitrate.
- **Timing:** Sub-200ms latency for real-time communication.
- **Relationships to other inputs/outputs:** Requires active session tokens from authentication API.
- **Data formats:**
    - Input/Output: Encrypted media streams (e.g., WebRTC).
- **Command formats:** WebRTC signalling with SDP (Session Description Protocol).
- **Data items/information included in the input and output:**
    - Input: Raw audio/video streams.
    - Output: Encrypted streams.

12) Functions

a) Validity Checks on the Inputs:

- **Authentication Inputs:**
    - Ensure email format is valid.
    - Verify password hash complies with encryption standards.
- **Meeting Creation Inputs:**
    - Validate meeting title, participant emails, and date/time format.

    - Check participant count.
    - Ensure no overlapping schedules for hosts.
- **Blockchain Data:**
    - Verify blockchain transaction IDs are valid 64-character hexadecimal strings.

b) Exact Sequence of Operations:

1. **Authentication Workflow:**
    a. User submits credentials.
    b. System hashes the password and compares it to the stored hash.
    c. Blockchain verifies user authenticity.
    d. If verified, a session token is issued.
2. **Meeting Creation Workflow:**
    a. Host submits meeting details.
    b. System validates input data and reserves slots.
    c. Meeting data is stored and recorded on the blockchain.
    d. Confirmation, including transaction ID, is sent to the host.
3. **Data Streaming:**
    a. Participants join the meeting.
    b. System verifies tokens and initiates encrypted communication channels.
    c. Streams are securely transmitted using WebRTC.

c) Responses to Abnormal Situations:

1. **Overflow:**
    a. Prevent buffer overflows during data transmission by implementing input size restrictions.
    b. Automatically reject overly large media file uploads or streams.
2. **Communication Failures:**
    a. Detect connection loss and attempt automatic reconnection.
    b. Notify users of disruptions and provide an option to rejoin.
3. **Error Handling and Recovery:**
    a. For invalid inputs, return detailed error messages.
    b. Log errors for debugging while ensuring sensitive information is not exposed.

d) Effect of Parameters:

- **Meeting Duration:** Directly affects recording storage requirements.
- **Number of Participants:** Influences bandwidth usage for real-time communication.
- **Blockchain Validation Interval:** Impacts response time for verification processes.

13) Usability Requirements

1. **Effectiveness:**

    a. **Requirement:** 95% task success rate (e.g., scheduling meetings, joining sessions) without assistance.
    b. **Metric:** Task completion rate and average time to complete tasks.

2. **Efficiency:**
    a. **Requirement:** Average response time ≤ 2 seconds, real-time streaming latency ≤ 200ms.
    b. **Metric:** System response times for key operations.
3. **Satisfaction:**
    a. **Requirement:** Intuitive, accessible interface (WCAG 2.1 AA compliance).
    b. **Metric:** ≥ 85% user satisfaction in surveys and minimal complaints.
4. **Avoidance of Harm:**
    a. **Requirement:** Zero critical security incidents; robust data protection.
    b. **Metric:** Compliance with GDPR/ISO standards and no breaches.
5. **Context of Use:**
    a. **Requirement:** Seamless functionality on mobile/low bandwidth (≥ 512 kbps).
    b. **Metric:** Usability ratings across devices and environments.

14) Performance Requirements

1. Support many concurrent users (e.g., 95% of transactions processed in less than 1 second).

2. Handle a high volume of data and transactions within defined time periods for normal and peak workloads.

3. Provide secure end-to-end encryption for all audio, video, and shared content.

4. Implement robust blockchain-based authentication and authorization mechanisms.

5. Ensure low latency and high reliability for real-time video/audio communication.

6. Maintain data integrity and prevent tampering or unauthorized access to meeting recordings and transcripts.

7. Scalable to support growing user base and increased usage over time.

8. Provide detailed usage metrics and analytics for administrators.

15) Logical Database Requirements

a) Types of information: user accounts, authentication credentials, meeting metadata, session recordings, transcripts, shared content, user activity logs.

b) Frequency of use: High for user authentication and meeting metadata; moderate for session recordings and transcripts; low for user activity logs.

c) Accessing capabilities: Secure and role-based access control for administrators, hosts, and participants. Support for granular permissions.

d) Data entities and relationships: Users, meetings, recordings, shared content, activities. Relationships between users, meetings, and associated data.

e) Integrity constraints: Data immutability and tamper-resistance for critical records like meeting transcripts. Referential integrity between related entities.

f) Security: End-to-end encryption for all data. Secure storage and access controls. Audit logging of all database activities.

g) Data retention: Meeting recordings and transcripts retained per compliance requirements. User activity logs kept for a defined period for analytics and investigations.

16)Design Constraints

1. **Compliance with Industry Standards:**

   - Adhere to global security standards such as ISO/IEC 27001 for information security.
   - Ensure GDPR compliance for data privacy in the EU.

2. **Regulatory Compliance:**

   - Encryption methods must meet legal standards (e.g., FIPS 140-2 in the United States).
   - Recording storage must comply with country-specific data residency laws.

3. **Performance Limitations:**

   - The system must maintain low latency (<300ms) for real-time audio and video communication.
   - Scalability must be achievable without compromising security or performance.

4. **Hardware and Network Constraints:**

   - Support devices with limited computational resources (e.g., mobile phones, low-end laptops).
   - Operate effectively in low-bandwidth environments while maintaining security.

5. **Project-Specific Limitations:**

   - Development must fit within predefined budgets and timelines.
   - Use only open-source or pre-approved third-party libraries for cost efficiency and transparency.

17) Standards Compliance

1. **Report Format:**

- Generate detailed activity reports for meetings, including participant logs, file-sharing events, and chat histories.
- Reports must adhere to specified formats (e.g., PDF or CSV) for interoperability.

2. **Data Naming:**

- Follow consistent naming conventions for stored data such as user credentials, meeting IDs, and recording files.
- Use unique identifiers to avoid duplication or conflicts.

3. **Accounting Procedures:**

- Implement auditable financial tracking for subscription payments, meeting hosting costs, and storage usage.
- Support integration with accounting software to ensure compliance with financial regulations.

4. **Audit Tracing:**

- Maintain a comprehensive audit trail of system activities, including:
    - Login/logout timestamps.
    - Changes to meeting settings (e.g., enabling/disabling encryption).
    - All file uploads, downloads, and deletions.
- Audit trails should include before-and-after values for any critical data changes to support forensic investigations.

18) Software System Attributes

a) Reliability

- Define factors to ensure the system meets the required reliability standards:
    - The online meeting platform must function correctly under normal and stress conditions (e.g., peak users).
    - Fault tolerance must be implemented:
        - Automatic reconnection for dropped participants.
        - Mechanisms for session recovery after network failures.
    - Ensure message delivery (chat, files, etc.) is consistent and accurate.

b) Availability

- Ensure availability by implementing the following:
    - The system must provide **99.9% uptime** to support critical business operations.
    - Implement mechanisms like:
        - **Checkpointing**: Save meeting progress to allow for seamless recovery in case of system failures.

        - **Auto-restart**: Allow servers to restart without affecting ongoing meetings.
    - Backup servers should be available for immediate failover during outages.

c) Security

Specify security measures to protect the platform from accidental or malicious access, modification, destruction, or disclosure:

1. **Cryptographic Techniques**:
    a. Use end-to-end encryption (e.g., AES-256) for meetings, chats, and file sharing.
    b. Secure storage of user data and meeting recordings.
2. **Activity Logging**:
    a. Maintain logs of access, changes, and meeting history while ensuring user privacy (GDPR compliance).
3. **Access Control**:
    a. Assign specific roles and permissions (e.g., host, co-host, participant).
    b. Limit file-sharing or screen-sharing to authorized participants.
4. **Restricted Communication**:
    a. Isolate different system modules (e.g., authentication, storage) to reduce the attack surface.
5. **Data Privacy**:
    a. Use secure hashing techniques for user credentials.
    b. Prevent unauthorized access by using robust authentication protocols (e.g., multi-factor authentication).

d) Maintainability

Attributes to simplify maintenance of the online meeting platform:

1. Design for modularity:
    a. Isolate video, audio, and chat systems into separate, maintainable modules.
2. Use standardized APIs and interfaces to integrate with external tools.
3. Keep system complexity low for easy debugging and updates.
4. Provide detailed error logs and automated diagnostic reports for issue resolution.
5. Ensure backward compatibility with older versions when introducing updates.

e) Portability

Attributes to ensure portability across different platforms:

1. Minimize dependency on platform-specific code (<10%).
2. Use portable programming languages like JavaScript or Python to ensure compatibility across operating systems (Windows, macOS, Linux).
3. Support both web-based (browser) and app-based access for ease of use.
4. Utilize platform-independent libraries for audio and video processing.

5. Design a responsive user interface to support various devices, including desktops, tablets, and mobile phones.

19)Verifications

1. **Functional Testing:**
   - Verify all essential features (e.g., video/audio streaming, chat, file sharing) function as per specifications.
   - Confirm proper role-based access control (host, participant, etc.) is implemented.

2. **Performance Testing:**
   - Conduct stress testing to ensure the platform handles the maximum number of participants without failure.
   - Verify that latency remains below the acceptable threshold for real-time communication (<300ms).

3. **Security Testing:**
   - Test encryption protocols for end-to-end security during meetings.
   - Perform penetration testing to ensure the platform is protected against potential threats (e.g., unauthorized access, data leaks).

4. **Data Integrity Testing:**
   - Validate that all user data (meeting logs, chat history, files) is stored and retrieved accurately.
   - Check audit trails to ensure no data tampering has occurred.

5. **Usability Testing:**
   - Verify the user interface is intuitive and functional across devices (desktop, mobile, tablet).
   - Ensure accessibility features are available and meet standards such as WCAG.

20)Supporting Information

1. **Sample Input/Output Formats:**
   - Example formats for meeting schedules (e.g., .ics files for calendar integration).
   - Formats for chat logs and meeting recordings (e.g., .txt for chat, .mp4 for video recordings).
   - Results of cost analysis for cloud storage of recordings.

2. **Background Information:**
   - Provide references to user survey results indicating the demand for features like enhanced security and multi-device compatibility.

- Include studies on user preferences for features like real-time transcription and recording storage.

3. **Description of the Problem:**

- Define the problems addressed by the system:
    - Insecure online meetings prone to unauthorized access.
    - Lack of user-friendly features in existing platforms for large-scale business meetings.
    - Insufficient compliance with data privacy regulations in competing platforms.