浙江大学

ZHEJIANG UNIVERSITY

基于分布式数字身份的电影院售票系统

课题名称	基于分布式数字身份的电影院售票系统
分　　营	数字身份分营
学生姓名	******
指导老师	******
日　　期	2024 年 07 月 15 日

目录

1 背景调研

1.1 元宇宙与数字身份

随着互联网技术的逐渐发展，元宇宙 (Metaverse) 的概念逐渐进入了人们的视野。元宇宙是一个通过虚拟现实（ VR ）、增强现实（ AR) 等技术构建的高度沉浸式的虚拟环境，在这个环境中，用户可以进行社交、娱乐、教育、商务等多种活动。在元宇宙中，用户可以以一个虚拟化身的形象参与各种互动，体验与现实世界不同的奇妙场景和丰富体验。

元宇宙的概念最早由科幻作家尼尔·斯蒂芬森 (Neal Stephenson) 在 1992 年的小说《雪崩》(Snow Crash) 中提出，近年来随着技术的进步，这一概念逐渐从科幻走向现实。元宇宙被认为是互联网发展的下一个阶段，它不仅仅是一个虚拟世界，更是一个包含物理世界和数字世界的融合体。

在元宇宙中，数字身份的管理变得尤为重要。用户需要在不同的虚拟环境中进行身份验证、资产管理和社交互动，这些都依赖于一个安全、可靠且易于管理的数字身份系统。然而，传统的集中式数字身份系统难以满足元宇宙对身份管理的高要求。这些系统通常存在单点故障、隐私泄露、数据篡改等问题，无法提供足够的安全性和隐私保护。

为了解决这些问题，分布式数字身份系统应运而生。分布式数字身份系统利用区块链技术，通过分布式的方式存储和管理身份数据，确保数据的安全性和不可篡改性。用户在这种系统中拥有对自己身份数据的完全控制权，能够自主决定何时、何地、向谁分享哪些数据，从而有效保护个人隐私。

1.2 数字身份技术的演化

1.2.1 传统的中心化、集中化的数字身份

传统的数字身份系统通常采用中心化和集中化的管理模式。在这种模式下，用户通过向后端服务提供用户名、密码等信息来进行身份验证和管理。这些信息由一个或多个服务器统一管理和存储，服务器通常由企业或服务提供商控制。

这种中心化的数字身份管理模式有以下问题：

- 单点故障：所有用户的身份信息存储在同一个或少数几个服务器上，若服务器出现故障或被攻击，整个系统的安全性和可用性将受到严重影响。
- 隐私风险：用户的身份信息集中存储在服务器上，容易成为攻击者的目标。一旦服务器被攻破，用户的个人信息可能会被泄露或滥用。
- 信任问题：用户必须完全信任身份提供者 (如社交媒体平台、电子商务网站等) 来保护其个人数据。然而，身份提供者可能会出于商业利益或其他原因滥用用户数据。
- 数据孤岛：每个服务提供商独立管理用户身份信息，导致用户在不同平台之间需要重复注册和认证，增加了用户的管理负担和使用不便。

1.2.2 联盟身份

为了克服传统中心化数字身份系统的缺点，联盟身份 (Consortium Identity) 应运而生。联盟身份是一种在区块链网络中由多个独立组织 (或实体) 组成的联盟 (Consortium) 中，每个成员组织的身份和角色管理模式。在这种模式下，用户只需注册一个数字身份即可参与和管理区块链网络中的活动，而不必在每个成员组织中重复注册。

联盟身份的具有诸多优点：

- 分布式管理：联盟身份系统通过区块链技术将身份数据分散存储在多个独立组织的节点上，消除了单点故障的风险，提高了系统的安全性和可靠性。
- 跨组织互操作性：在联盟身份系统中，用户的数字身份可以在联盟中的多个成员组织之间共享和验证，避免了重复注册和认证的问题，提高了用户体验。
- 增强的隐私保护：用户的身份信息在区块链网络中经过加密处理，只有在得到用户授权的情况下才会被共享和验证，增强了隐私保护。
- 透明和可追溯性：区块链的不可篡改性和透明性确保了用户身份数据的完整性和可追溯性，有助于防范身份欺诈和数据篡改。

然而，联盟身份系统也存在一些问题：

- 数据高度垄断：尽管联盟身份系统分散了数据存储，但联盟中的中心节点仍然掌握大量用户身份信息，可能导致数据垄断，增加隐私泄露的风险。
- 隐私风险：由于多个组织共享用户身份信息，一旦联盟中的任何一个节点被攻破，用户的隐私信息可能会被泄露。
- 数据过度集中：虽然联盟身份在一定程度上解决了互操作性差的问题，但数据过度集中于联盟成员中，如果中央管理机构出现问题，则整个系统的安全性和稳定性将面临巨大风险。

综上所述，联盟身份虽然在解决传统中心化身份系统的一些问题上具有显著优势，但仍需应对数据垄断和隐私保护等挑战。在未来的发展中，需要进一步探索和完善技术手段，以确保联盟身份系统的安全性、可靠性和隐私保护能力。

1.2.3 分布式数字身份

分布式数字身份 (Decentralized Digital Identity) 是一种基于区块链和分布式账本技术的新型身份管理模式。在这种模式下，身份数据不是集中存储在某一个或几个服务器上，而是分散在一个分布式的网络中。这种分布式的身份管理方式利用密码学技术和共识机制来确保数据的安全性和不可篡改性，使得用户对自己的身份数据拥有更大的控制权。

相比于传统的中心化数字身份和联盟身份，分布式数字身份简化了用户身份的管理和验证流程，提高了用户的隐私保护和数据安全性。用户无需与第三方机构共享自己的所有隐私信息，可以选择性地向其他用户或服务提供者提供部分身份数据，甚至不提供任何真实身份信息也能完成身份验证。这种身份自主管理的模式有效防止隐私泄露和数据滥用，增强了用户对自己身份数据的控制能力。

总的来说，分布式数字身份具有如下优势：

• 隐私安全：身份数据通过加密技术进行保护，只有在用户授权的情况下才能被访问和使用，并且可以实现选择性披露和零知识证明，从而有效防止隐私泄露和数据滥用。

• 身份自主管理：用户对自己的身份数据拥有完全的控制权，不需要依赖任何第三方机构来管理或验证其身份。这种自主性增强了用户对身份数据的掌控能力。

• 身份可移植性：用户的数字身份不依赖于特定的身份服务提供商，可以在不同的平台和应用之间自由迁移和使用。这种可移植性简化了用户身份管理的复杂性。

• 可信数据交换：区块链技术保证了数据的完整性和不可篡改性，使得身份数据在交换和验证过程中具有高度的可信度。用户和服务提供者可以在没有中介的情况下进行安全可靠的数据交换。

1.3 区块链技术

区块链技术是分布式数字身份系统的基础，为分布式的数据存储和管理提供了可靠的平台。区块链通过其分布式、不可篡改和透明的特性，确保了身份数据的安全性和完整性。

1.3.1 基本特性

• 分布式：区块链通过分布式网络中的节点共同维护账本，避免了单点故障。每个节点持有账本的完整副本，确保数据的高可用性和可靠性。

• 不可篡改：每个区块都包含前一个区块的哈希值，形成链式结构，使得任何篡改都需要同时修改所有后续区块，几乎不可能实现。通过加密技术和哈希函数，保证了数据的完整性和不可篡改性。

• 共识机制：节点通过共识算法达成共识，确保数据一致性和安全性。这些共识机制确保了区块链网络的分布式和防篡改特性。

1.3.2 智能合约

智能合约 (Smart Contract) 是一种基于区块链技术的协议，旨在以数字化形式促进、验证或执行合约的条款。通过智能合约，参与方可以在没有中介机构的情况下，自动执行合约内容。

智能合约通过编程语言编写，并部署在区块链网络中。首先，开发者使用特定的编程语言 (如 Solidity、Go、JavaScript 等) 编写合约代码，定义合约的规则和行为。然后，将编写好的智能合约部署到区块链网络上。部署后，合约的代码和状态存储在区块链上，成为不可篡改的一部分。当预定义的条件满足时，智能合约会自动执行。例如，在某些事件发生时 (如支付、投票结果等) ，合约会自动进行相应的操作，并将执行结果记录在区块链上，确保透明和不可篡改。

1.3.3 Hyperledger Fabric 框架

Hyperledger Fabric 是一个模块化和可扩展的企业级区块链平台，由 Linux 基金会主持的 Hyperledger 项目开发。它旨在提供一个通用的、开源的分布式账本平台，特别适合复杂的企业用例。Hyperledger Fabric 与其他区块链平台的不同之处在于其高度的灵活性和可配置性，能够满足企业对隐私、安全性和性能的需求。

Hyperledger Fabric 的核心组件包括以下几个方面：

1. 链码 (Chaincode)：链码相当于智能合约，负责定义业务逻辑。链码可以用 Go、Java 或 JavaScript 编写，并在指定的通道上执行。它们可以被安装、实例化和调用以处理交易请求。

2. 通道 (Channel)：通道是一个私有的区块链子网络，允许特定的成员组进行机密交易。每个通道都有独立的账本，保证了交易的隐私性和隔离性。

3. 共识机制 (Consensus)：Hyperledger Fabric 采用了可插拔的共识机制，允许网络管理员根据需要选择适当的共识协议。常见的共识机制包括 Kafka、Raft 等。这种设计提供了灵活性，能够在性能和安全性之间进行权衡。

4. 排序服务 (Ordering Service)：排序服务负责对交易进行排序，并将其打包成区块。排序服务确保了交易的全局顺序，是 Fabric 架构中实现一致性的关键组件。

5. 节点类型：Hyperledger Fabric 包含不同类型的节点，如对等节点 (Peer Node) 和排序节点 (Orderer Node)。对等节点维护账本并执行链码，而排序节点则负责交易排序和区块生成。

6. 账本 (Ledger)：Fabric 的账本由两部分组成：世界状态和交易日志。世界状态存储区块链的最新状态，而交易日志记录了所有的交易历史。

1.4 W3C 数字身份标准

W3C (World Wide Web Consortium）提出了分布式身份（Decentralized Identifiers，DIDs）和可验证凭证（Verifiable Credentials，VCs) 两大标准，旨在为分布式数字身份系统提供统一的规范。DIDs 是一种新型的标识符，能够在不依赖中心化注册机构的情况下唯一标识个体或实体；VCs 则是一种标准化的数据格式，用于表达和验证各种声明。

1.4.1 分布式数字身份标志符[1]

分布式数字身份标志符 (以下简称 DID) 是一个唯一的、持久的、分布式的标识符，用于唯一标识个体、组织、设备或其他实体。

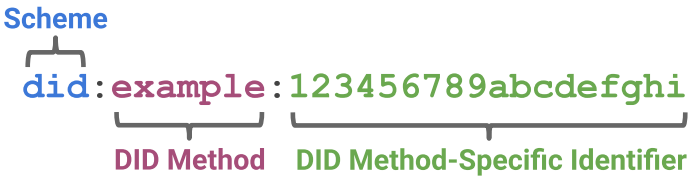


图 1.1 DID 的一个简单例子

如图 1.1，是一个简单的 DIDs 的例子，包含三个部分：

1.DID URI 方案标志符：即固定的前缀“did”，表明这个 URI 字符串是一个 DID。

2.DID 方法标志符：这个去 DID 是用哪一套方案（方法）来进行定义和操作的。

3.唯一标识符：这个部分是一个特定的字符串，用来唯一标识一个实体或个体。

1.4.2 分布式数字身份文档 (DID Document)[1]

分布式数字身份文档 (以下简称 DID Document) 是一个包含了 DIDs 的所有信息的文档，用于描述 DIDs 的所有者、公钥、服务端点等信息。每一个 DID Document 有一个唯一的 DID 作为标识符，用于标识文档的所有者。所有 DID Document 均存在区块链上，任何人都可以通过 DID 来查找和验证文档的信息。

一个符合 W3C 规范的 DID Document 包含信息如图 1.2：

属性名	必需性	类型	描述
id	必需	string	这是DID主体的唯一标识符，符合小节 1.4.1 中描述的DID语法规则，是在DID系统中标识主体的基本元素。
alsoKnownAs	可选	string[]	允许DID主体的附加标识符，便于在不同平台或系统中链接代表同一实体的不同标识符。
controller	可选	string | strig[]	指定控制DID文档的其他DID，为授权或控制的委派提供机制。
verificationMethod	可选	VerificationMethod | VerificationMethod[]	一组用于认证或验证与 DID 主体交互的方法（如加密公钥）。
assertionMethod	可选	VerificationMethod[] | string	分别指定不同的验证方法，用于断言声明、加密通信、调用能力和委派能力。
keyAgreement	可选		
capabilityInvocation	可选		
capabilityDelegation	可选		
authentication	可选		
service	可选	ServiceEndpoint[]	详细说明与DID主体相关的服务，提供如何通过特定服务端点与主体互动或通信的信息。

图 1.2　DID Document 的属性列表

其中验证方法 (VerificationMethod) 的属性列表如图 1.3 所示，用于描述一种验证该 DID Document 是否有效的方法。

服务端点 (ServiceEndpoint) 的属性列表如图 1.4 所示是一个包含了服务的终端信息的对象，用于描述与 DID 主体相关的服务，提供了如何通过特定服务端点与主体互动或通信的信息。

属性名	必需性	类型	描述
id	必需	string	验证方法在DID文档中的唯一标识符。

controller	必需	string	指定验证方法的控制器 DID，用于授权或控制的委派。
type	必需	string	指定验证方法的类型，确保方法在各种上下文中正确应用
publicKeyJwk	可选	Object	符合 [RFC7517]中定义的 JSON Web Key (JWK)格式[3] 的公钥
publicKeyMultibase	可选	string	符合 Multibase 编码格式[4] 的公钥

图 1.3 VerificationMethod 的属性列表

属性名	必需性	类型	描述
id	必需	string	符合 RFC3986 中定义的 URI 格式[5]，用于唯一标识服务端点。
type	必需	string	描述服务类型，有助于理解服务的预期用途或协议。
serviceEndpoint	必需	string	提供访问服务的联系点或机制，通常为 URI。

图 1.4 ServiceEndpoint 的属性列表

1.4.3 可验证凭证 (VC)[2]

可验证凭证 (Verifiable Credentials，以下简称 VC) 是一种用于表达和验证声明的标准化数据格式，是一个实体给另一个实体的某些属性做背书而发出的描述性声明，并附加自己的数字签名，用以证明这些属性的真实性，可以认为是一种数字证书。发行 VC 的实体的 DID Document 也存在于公开的区块链网络中，验证机构可以通过验证签名来确认 VC 的真实性。

用户可以将 VC 打包为可验证表达（Verifiable Presentation，以下简称 VP），用于在不同的场景中展示和验证自己的身份信息。验证机构 VP 的签名以确保该 VP 是由该用户自主生成且未被篡改的。

一个符合 W3C 规范的 VC、VP 包含信息如图 1.5、图 1.6 所示：

属性名	必需性	类型	描述
id	可选	string	VC 的唯一标识符
type	必需	string[]	VC 的类型，用于描述 VC 的类别或用途
issuer	必需	string	VC 的发行者 DID
issuanceDate	必需	string	VC 的发行日期
expirationDate	可选	string	VC 的过期日期
credentialSubject	必需	Object	VC 的验证主体，包含声明的所有实体信息
proof	可选	Object	VC 的证明，包含了 VC 的签名信息和验证方法

图 1.5　VC 的属性列表

属性名	必需性	类型	描述
id	必需	string	VP 的唯一标识符
type	必需	string[]	VP 的类型，用于描述 VP 的类别或用途
verifiableCredential	必需	VerifiableCredential[]	包含的 VC 列表，用于展示和验证 VC 的信息。
proof	必需	Object	VP 的证明，包含了 VP 的签名信息和验证方法

图 1.6　VP 的属性列表

属性名	必需性	类型	描述
type	必需	string	证明的类型，用于描述证明的类别或用途
created	可选	string	证明的创建日期
verificationMethod	可选	string	用于验证证明的方法
proofPurpose	可选	string	证明的目的
proofValue	可选	string	证明的值，通常是签名或哈希值

图 1.7　证明 (Proof) 的属性列表

2 场景分析

本项目旨在设计并实现一个基于 Hyperledger Fabric 的分布式数字身份系统，应用于电影院售票系统。该系统利用分布式数字身份验证和 NFT 技术，实现用户身份验证和电影票的购买、验证流程。

2.1 应用场景选择

A. 行业背景和需求分析

电影行业的售票系统面临身份验证和年龄限制的问题，特别是对于受年龄限制的电影。现有的系统主要依赖于传统的身份验证方法，如身份证检查，这不仅耗时，而且容易泄露用户隐私。

B. 现有问题和痛点

当前的身份验证方式存在多个问题：

隐私泄露：用户在验证身份时，需要提供身份证等个人信息，容易造成隐私泄露。

身份造假：传统身份验证方式容易被伪造，难以保证真实性。

管理成本高：人工验证身份和年龄限制，增加了电影院的管理成本。

C. 分布式数字身份的优势

分布式数字身份系统通过区块链技术和零知识证明技术，能够提供更加安全、隐私保护的身份验证方式，降低管理成本，提升用户体验。

2.2 场景需求定义

本小节将从用户需求和功能需求两个方面，定义电影院售票系统的场景需求。

2.2.1 用户需求

普通用户：能够方便快捷地购买电影票，同时确保个人隐私不被泄露。

电影院：能够准确验证用户身份和年龄，防止未成年用户购买限制级电影票。

公安局：作为权威机构，能够提供可信的出生证明。

2.2.2 功能需求

1. 数字身份创建：用户能够自由创建数字身份，并在本地保管私钥。

2. 出生证明申请：用户能够向公安局申请出生证明 (VC)。

3. 匿名化验证：用户能够使用零知识证明技术构造匿名证明 (VP)，电影院可以验证用户的年龄而不泄露真实信息。

4. 电影票购买：用户能够使用数字身份购买 NFT 票据。

5. NFT 票据创建：电影院能够通过验证用户的 DID 和 VP，为用户创建 NFT 票据。

6. 电影票验证：电影院能够通过验证用户提供的 NFT 票据，实现身份和票据的验证。

2.2.3 用例图

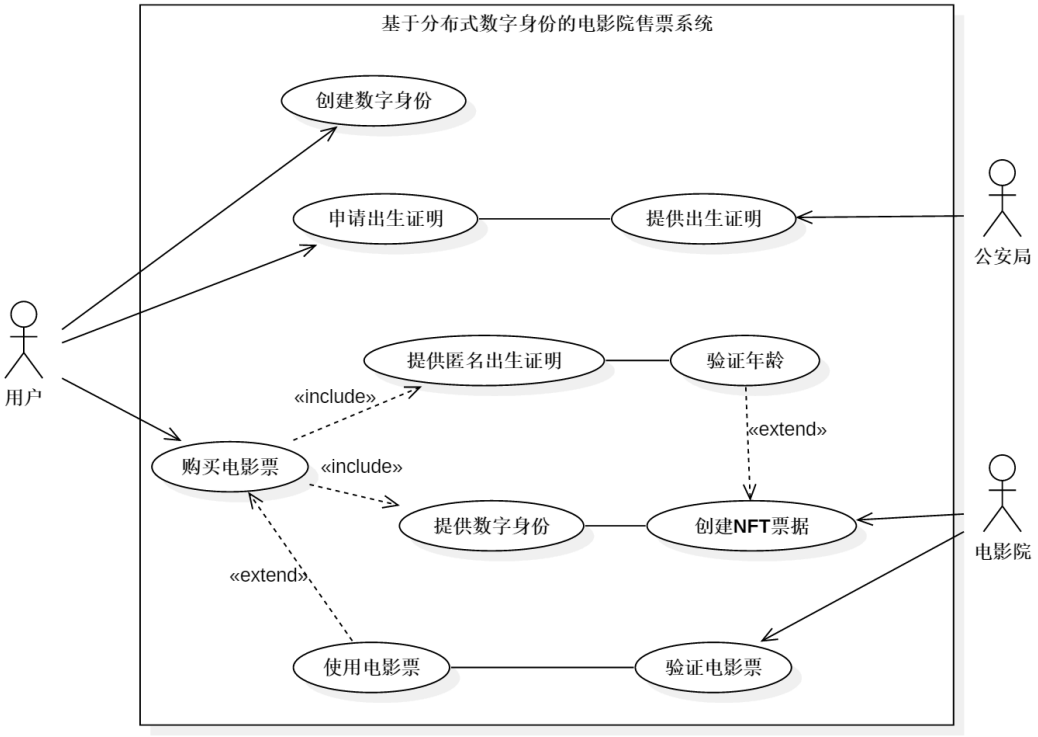


图 2.1 基于分布式数字身份的电影院售票系统用例图

2.3 场景实例描述

1. 在实际应用中，用户 Alice 希望观看一部限制级电影。

2. Alice 使用本地生成的密钥对创建数字身份，并向公安局申请出生证明 VC。

3. 在购买电影票时，Alice 使用零知识证明技术构造匿名证明 VP，提供给电影院。

4. 电影院验证 Alice 的证明属实，为 Alice 创建 NFT 票据，并把 NFT 票据 ID 发送给 Alice。

5. 到达电影院后，Alice 提供 NFT 票据的 ID 和自己的签名，验证自己对电影票的所有权。

6. 电影院验证通过后，Alice 可以进入电影院观影。

在此过程中，Alice 并没有向电影院提供自己的真实年龄和身份信息，电影院也成功验证了 Alice 的年龄和身份。这种基于分布式数字身份的电影院售票系统，不仅提高了用户的隐私保护，同时也提升了电影院的管理效率。

3 技术选型

技术选型是构建系统的关键步骤，通过选择合适的技术框架和工具，可以确保系统的高效运行和稳定性。本节将详细阐述基于分布式数字身份的电影院售票系统的技术选型。

3.1 区块链框架

Hyperledger Fabric 是一个开源的企业级许可分布式账本技术平台，由 Linux 基金会下的 Hyperledger 项目管理。其设计目的是在企业环境中提供高度模块化和可配置的架构，以满足多种行业用例的需求。[6]

在本项目中，选择 Hyperledger Fabric 作为开发区块链应用的框架，主要基于以下几个原因：

1. **支持多种编程语言编写智能合约**：Hyperledger Fabric 支持使用 Java、Go 和 Node.js 等通用编程语言编写智能合约。在本项目中，选择了 TypeScript 来编写智能合约。这不仅降低了开发人员的学习成本，还提高了开发效率和代码的可维护性。

2. **模块化架构**：Hyperledger Fabric 的模块化架构允许用户根据需求定制区块链网络。这种灵活性使得能够针对分布式数字身份系统的特定需求进行优化和调整，包括共识机制、成员服务和数据存储等模块。

3. **数据隔离**：Hyperledger Fabric 的通道机制允许不同的参与方在同一网络中创建隔离的账本数据，从而保护数据隐私和实现数据隔离。该系统可以利用这一特性，将 NFT 票据与用户 DID Document 进行隔离。

4. **社区支持**：作为一个广泛使用的开源项目，Hyperledger Fabric 拥有强大的社区支持和丰富的资源。这为开发和维护分布式数字身份系统提供了有力的保障。

5. **用户交互便捷**：Hyperledger Fabric SDK 提供了丰富的 API，使得应用可以方便地与区块链网络进行通信和操作，而 Gateway 则简化了与网络的连接过程，提高了系统的易用性。

3.2 前端框架

本项目选择 Vue3 和 Vite 作为前端技术栈。Vue3 是一个渐进式 JavaScript 框架，具有高性能、易用性和灵活性，适合构建现代 Web 应用程序。Vite 是一个新的前端构建工具，专为现代前端开发工作流程设计，具有快速的开发环境和高效的构建性能。[7,8] 选择 Vue3 和 Vite 的原因如下：

1. **高性能和响应性**：Vue3 的虚拟 DOM 和高效的 diff 算法使得应用具有高性能和快速响应的特点，适合需要频繁更新和交互的分布式数字身份系统前端。

2. **易用性和灵活性**：Vue3 具有简洁的语法和易用的 API，使得开发人员能够快速上手并高效地开发复杂的用户界面。同时，Vue3 的组件化设计允许将 UI 分解成可复用的小组件，提高代码的可维护性和可扩展性。

3. **强大的生态系统**：Vue3 拥有丰富的生态系统，包括 Vue Router 等官方支持的库和工具。这些工具可以帮助实现路由管理等常见功能，从而简化开发过程。

4. **Vite 的快速开发环境**：Vite 使用原生 ES 模块进行开发，提供了即时的模块热替换（HMR）功能，使得开发过程更加高效和顺畅。同时，Vite 的构建速度极快，适合大型应用的开发和部署。

3.3 后端框架

本项目选择 Node.js 和 Express 作为后端技术栈，并使用 TypeScript 进行开发。Node.js 是一个基于 Chrome V8 引擎的 JavaScript 运行环境，适合构建高性能的网络应用。Express 是一个简洁且灵活的 Node.js Web 应用框架，提供了一系列强大的功能用于 Web 和移动应用开发。[9] 选择 Node.js 和 Express 的原因如下：

1. **高性能和异步 I/O**：Node.js 采用事件驱动和非阻塞 I/O 模型，具有高并发处理能力，适合处理大量用户请求和复杂的后台逻辑。这对于分布式数字身份系统中的数据处理和网络通信尤为重要。

2. **TypeScript 的类型安全**：使用 TypeScript 可以提高代码的可读性和可维护性，提供类型检查和自动补全功能，减少开发过程中的错误。

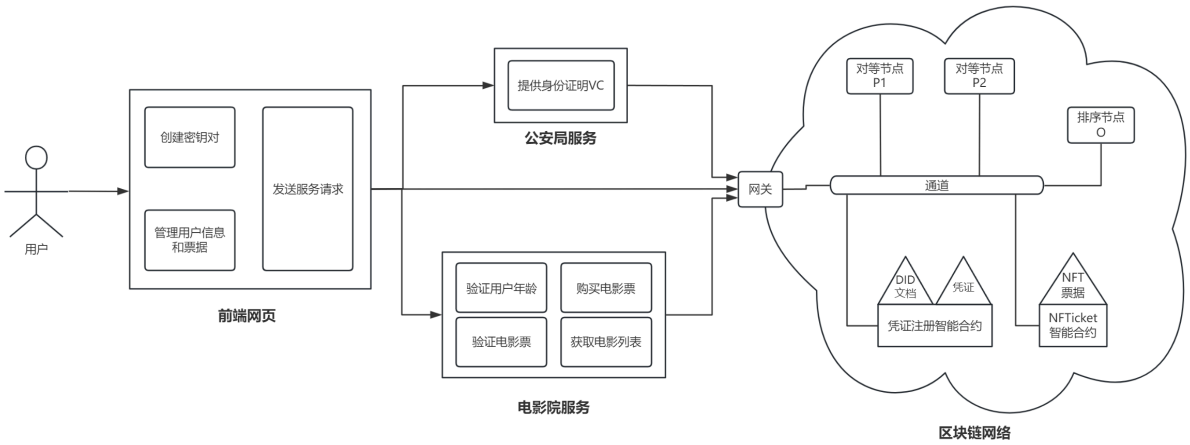
3. **丰富的中间件支持**：Express 具有丰富的中间件生态系统，可以轻松集成各种功能，如身份验证、日志记录、错误处理等。这使得能够快速搭建和扩展后端服务，满足分布式数字身份系统的多样化需求。

4. **与前端的无缝集成**：由于 Node.js 和前端都使用 JavaScript/TypeScript，开发人员可以在前后端共享代码和工具，提高开发效率。同时，Node.js 还可以方便地与 Vue3 和 Vite 进行集成，构建高效的前后端协作环境。

4 系统设计

4.1 总体系统架构

系统的总体架构如下图所示：



该系统主要包括三个模块：前端网页 (客户端) 、服务器 (公安局服务和电影院服务) 、区块网络。

用户直接操作前端网页，前端网页可以在本地创建密钥对，并保存私钥。前端网页提供一系列服务请求接口，用户可以通过这些接口创建数字身份、申请出生证明、购买电影票等。在前端页面可以形象地展现这个流程。

公安局服务用于给用户提供出生证明。该系统中只有公安局唯一一个权威的 VC 发行机构。

电影院服务提供购票、检票等服务，可以验证用户年龄、验证 NFT 票据的所属权。

区块链网络用于存储用户的数字身份文档和 NFT 票据。该网络中有以下几个组成部分：

1.2 个对等节点 P1、P2，每个对等节点上都有所有资产的副本。

2.一个排序节点 O，用于对交易进行排序。

3.一个通道，连接 P1、P2 和 O。该通道上部署了一个链码。

4.链码中存在两个智能合约，一个用于存储用户的数字身份文档和凭证，一个用于存储 NFT 票据。

服务和网页可以通过网关访问区块链网络进行交易。

4.2 数据结构设计

这一小节将介绍系统中的数据结构设计。所有数据结构均在符合 W3C 标准的前提下进行了一些简化。

4.2.1 DID Document

属性名	必需性	类型	描述
id	必需	string	这是 DID 主体的唯一标识符，符合小节 1.4.1 中描述的 DID 语法规则，是在 DID 系统中标识主体的基本元素
version	必需	string	这是 DID 文档的版本号，用于标识文档的版本
updated	必需	string	这是 DID 文档的最后更新时间，用于标识文档的更新时间
created	必需	string	这是 DID 文档的创建时间，用于标识文档的创建时间
controller	必须	string	指定控制 DID 文档的其他 DID，为授权或控制的委派提供机制
verificationMethod	可选	VerificationMethod	一个用于认证或验证与 DID 主体交互的方法（如加密公钥）
authentication	可选	string	一个 URI，指向一个 VerificationMethod，用于验证 DID 主体的身份

图 4.2　DID Document 数据结构

属性名	必需性	类型	描述
id	必需	string	这是 VerificationMethod 的唯一标识符
type	必需	string	指定公钥的加密算法
controller	必需	string	指定控制 VerificationMethod 的其他 DID，为授权或控制的委派提供机制
publicKeyMultibase	必需	string	这是 VerificationMethod 的公钥，用于验证 DID 主体的身份；公钥以 Multibase 编码表示

图 4.3　VerificationMethod 数据结构

本项目中，VerificationMethod 的加密算法均为 `secp256k1`。

4.2.2　零知识证明的 VC 凭证主体

属性名	类型	描述
id	string	凭证声明的用户 DID
min	number	凭证声明的最小出生年份

max	number	凭证声明的最大出生年份
birthYearIndex	number	凭证声明的出生年份索引，例如 0 表示用户在 min 年出生，1 表示用户在 min+1 年出生，以此类推
seed	string	随机种子，用于生成盐值序列
merkleTreeRoot	string	Merkle 树的根哈希值
rootSignature	string	VC 发行机构对 Merkle 树根的签名
signer	string	VC 发行机构的 DID
description	stirng	凭证声明的描述

图 4.4　零知识证明的 VC 凭证主体

4.2.3　零知识证明的 VP 凭证主体

属性名	类型	描述
id	string	凭证声明的用户 DID
assert	string	凭证声明的断言，格式为 年份:[0|1]，例如 1990:1 表示用户在 1990 年已经出生，1980:0 表示用户在 1980 年还未出生
dataIndex	number	表示该断言在 Merkle 树中的索引
salt	string	表示该断言数据添加的盐值
merklesibling	string	默克尔树验证路径，哈希值之间通过空格分隔
rootSignature	string	VC 发行机构对 Merkle 树根的签名
signer	string	VC 发行机构的 DID

图 4.5　零知识证明的 VP 凭证主体

4.2.4　NFT 票据

属性名	类型	描述
id	string	NFT 票据的唯一标识符
ownerDid	string	NFT 票据的所有者 DID
movie	Movie	电影详细信息
expirationDate	string	NFT 票据的过期日期

图 4.6　NFT 票据数据结构

4.3 核心算法设计

隐私保护是分布式数字身份的一个重要功能。在传统的中心化数字身份系统中，用户必须将自己的个人信息集中存储在服务器上，容易成为攻击者的目标。而在分布式数字身份系统中，用户可以选择性地共享自己的身份信息，只在必要的情况下提供必要的信息，从而保护个人隐私。

4.3.1 选择性披露

选择性披露是一种重要的隐私保护机制，它允许用户在验证身份时，只披露必要的部分信息，而不需要暴露全部的个人数据。这种方法基于密码学技术，确保披露的信息真实可信，同时保护未披露信息的隐私。

下面介绍一种基于默克尔树 (Merkle Tree) 的选择性披露方法[10]。

默克尔树是一种二叉树，其中每个叶子结点的值是一个数据的哈希值，每个非叶子节点的值是其子节点值的哈希值。为了防止哈希碰撞，在对原始数据哈希前进行加盐处理。由一个固定的随机种子生成一系列的随机序列作为盐，然后将盐和原始数据拼接后进行哈希。

图 4.7 是一个例子，将用户的身份信息作为数据值，构建默克尔树：

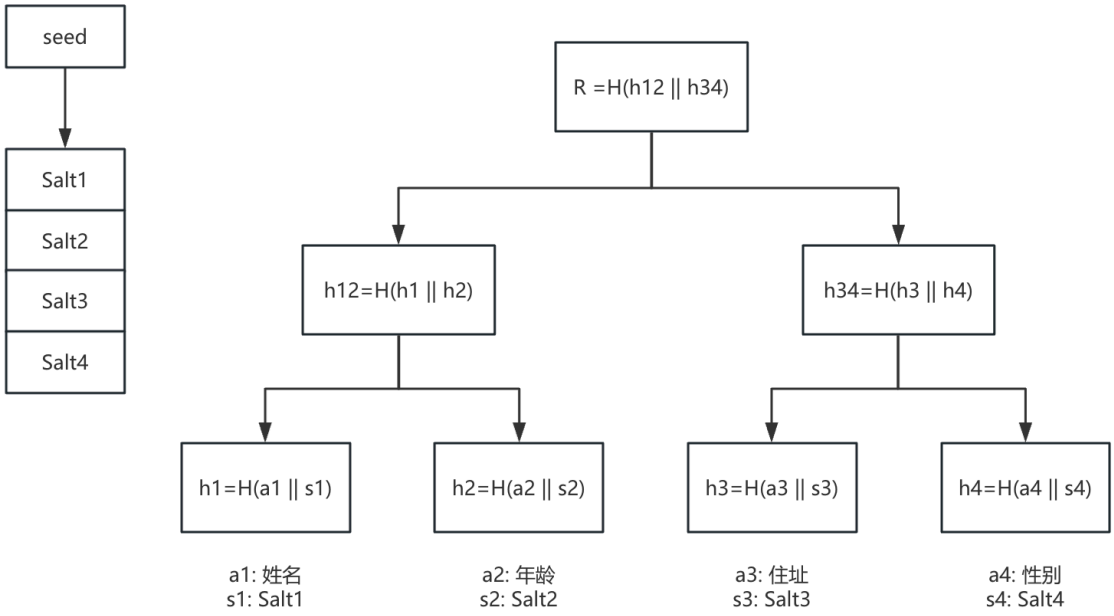


图 4.7 用户身份信息的默克尔树构建

首先，发行 VC 的机构生成一个随机种子 `seed`，根据某种固定的算法生成一个盐值的序列。其次，将用户的身份信息作为数据与对应的盐值拼接后，进行哈希运算，得到的结果作为默克尔树的根节点。最后，将所有根节点两两拼接，再进行哈希运算，得到的结果作为他们的父节点；重复这一个过程直到只剩下一个节点，即为默克尔树的根节点 `merkleRoot`。

发行机构提供的 VC 中需要包含：随机种子 `seed` 和默克尔树的根节点 `merkleRoot`，以及根节点的签名 `rootSignature` (保证树根未被篡改)。

用户想要选择性披露某个数据，只需要提供以下几个字段：

1. 要披露的原始数据内容
2. 该字段在默克尔树中的索引
3. 该字段的盐值
4. 该字段的默克尔验证路径
5. 默克尔树的根节点 `merkleRoot`、发行机构的 DID、根节点的签名 `rootSignature`

验证机构在收到用户提交的 VP 后，首先通过签名验证根节点是否被篡改。如果根节点未被篡改，则将用户提供的原始数据和盐值拼接后进行哈希运算，得到的结果与默克尔验证路径中的哈希值依次拼接后进行哈希运算，最终得到的结果与默克尔树的根节点进行比对，如果相等则说明提供的原始数据可信。

验证过程如图 4.8 所示，图中红色的部分是用户 VP 中提供的数据：

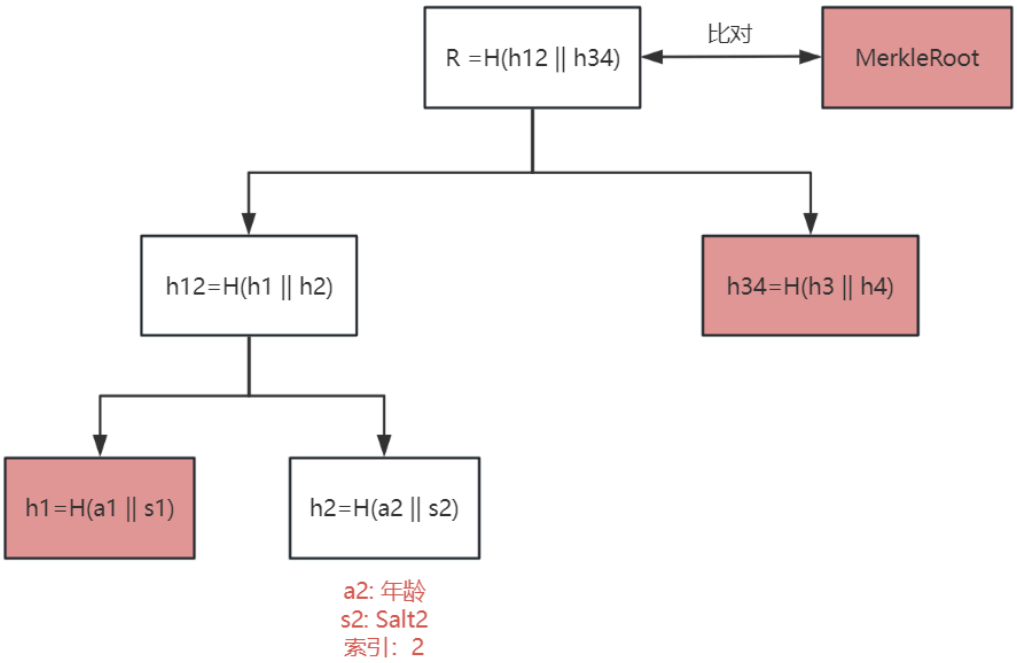


图 4.8　选择性披露示意图

装 订 线

4.3.2 零知识证明

在数字身份系统中，零知识证明 (Zero-Knowledge Proof，ZKP) 是一种可以验证某个声明是真实的技术，而不需要透露具体的身份信息。例如，用户可以证明自己的年龄大于 18 岁，而不需要透露具体的出生日期；用户可以证明自己的资产超过一定金额，而不需要透露具体的财务信息。通过这种方式，用户能够在保护隐私的前提下进行身份验证。

这里采用一种根据采样粒度与采样范围进行数据的断言构建的零知识证明方法[11]。下面以年龄证明为例，介绍该方法的具体实现。

发证机构首先在数据所在的作用域内进行采样。采样包括两个方面：范围和粒度。例如，对于用户的出生年份，可以将采样范围定义为 1900 年到 2020 年，采样粒度为一年。然后，为每个年份生成一个断言数组，数组中仅包含 0 和 1 两个值，0 表示当年用户未出生，1 表示当年用户已出生。

假设某用户出生于 1981 年，则他的断言数组为：

$$[0, 0, 0, ..., 1, 1, 1, ..., 1, 1, 1] \tag{4.1}$$

该数组中中有前面有 81 个 0，表示用户 1900 至 1980 年未出生；有 40 个 1，表示用户 1981 至 2020 年已出生。

将这个数组中的所有元素作为默克尔树的叶子节点，然后按照小节 4.3.1 中提到的方法构建默克尔树和 VC。用户可以选择性披露某个年份的断言，表明自己在某年是否出生。

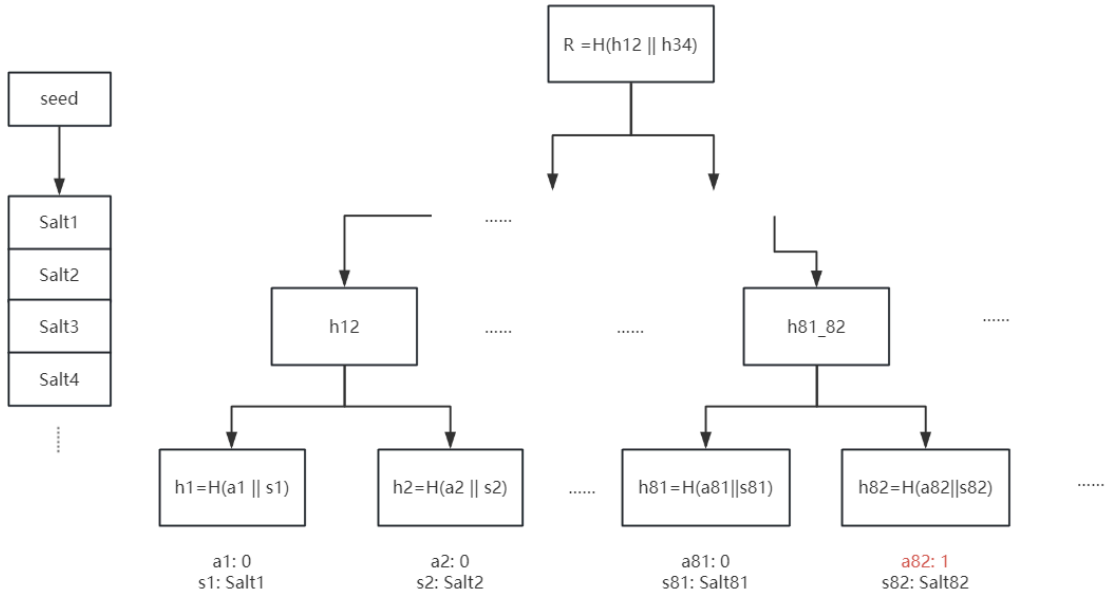


图 4.9 零知识证明的默克尔树构建

4.4 功能模块设计

4.4.1 数字身份创建

A. 业务流程描述

1. 用户在本地生成一个密钥对，私钥保存在本地。
2. 用户将公钥发送给凭证注册 (Credential Registry) 智能合约
3. 凭证注册智能合约根据用户公钥生成 DID Document，并将 DID 返回给用户。
4. 用户可以根据 DID 在区块链上查询自己的 DID Document。

B. 时序图

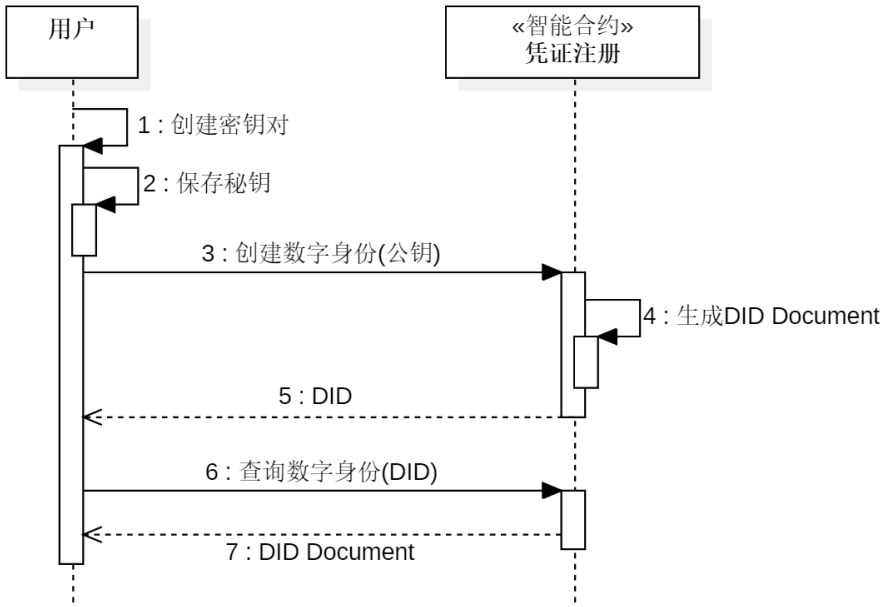


图 4.10 数字身份创建时序图

4.4.2 出生证明申请

A. 业务流程描述

1. 用户向公安局发出申请出生证明的请求，并提供 DID 和年龄证明。
2. 公安局 (VC Issuer) 验证年龄证明。
3. 公安局使用零知识证明方法，构造默克尔树，生成出生证明 (VC)。
4. 公安局将默克尔树树根注册到区块链上。
5. 公安局将出生证明发送给用户。

B. 时序图

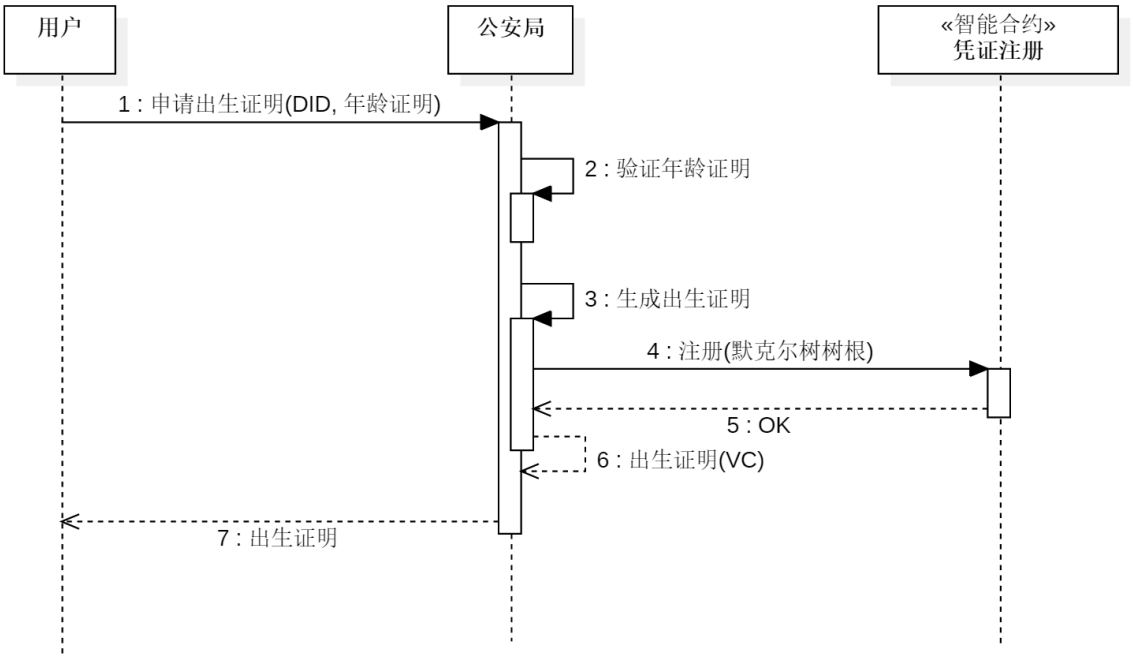


图 4.11　出生证明申请时序图

4.4.3　电影票购买与票据验证

A. 业务流程描述

1. 用户匿名化出生证明。

2. 在电影院网站上选择电影，点击购买，提供自己的 DID、匿名出生证明 (VP) 和电影 ID。

3. 电影院校验用户提供的的证明，判断是否符合电影的年龄限制。

4. 电影院向 NFTicket 合约请求创建 NFT 票据。创建的 NFT 票据包含唯一 ID、电影信息、过期时间和所属用户 DID。

5. 电影院将 NFT 票据的 ID 发送给用户。

6. 用户提供 NFT 票据的 ID、对电影票 ID 的签名给电影院验证。提供签名用于证明用户对电影票的所属权。

7. 电影院查询 NFTicket 合约，获取该 NFT 票据的电影信息、是否过期、所属用户。

8. 电影院查询凭证注册合约，获取用户的公钥，验证用户的签名。

9. 电影院验证通过后，用户可以进入电影院观影。

B. 时序图

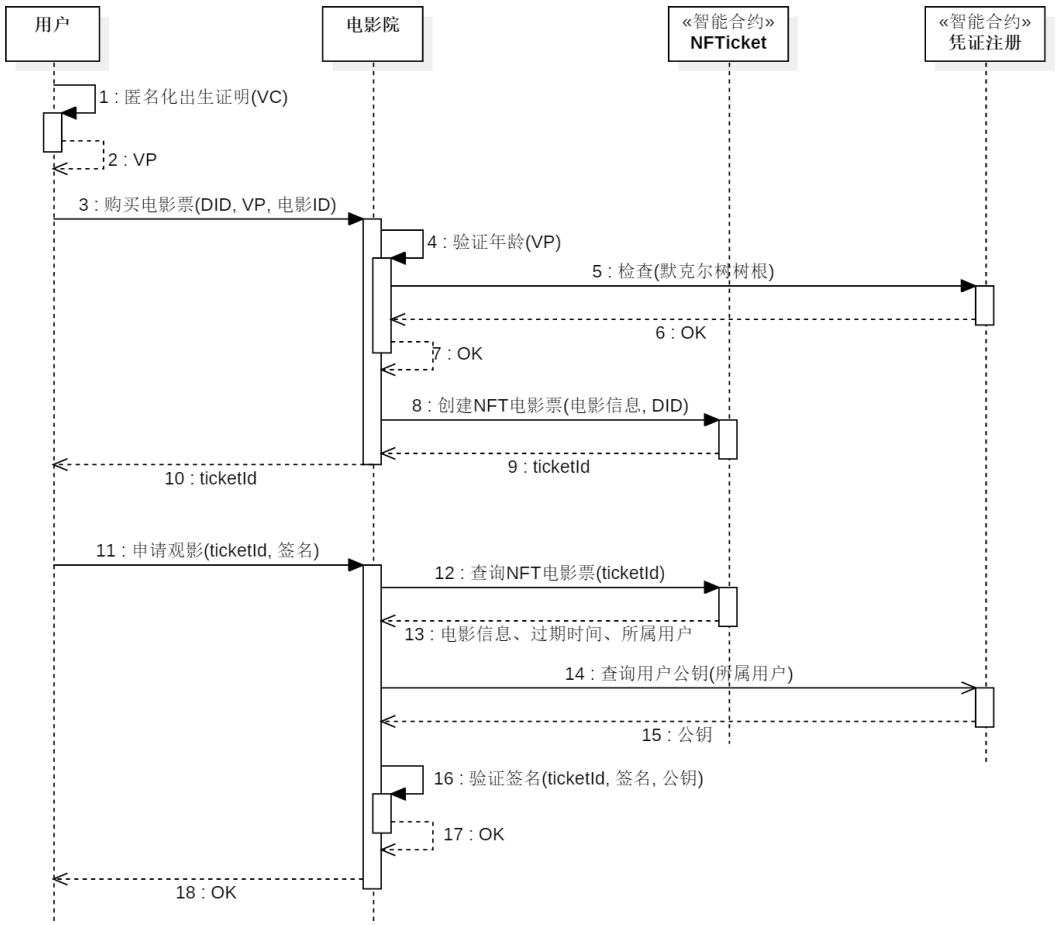


图 4.12　电影票购买与票据验证时序图

4.5　智能合约设计

4.5.1　凭证注册智能合约

函数名	参数	返回值	描述
createDID	did: string, didDocument: string	void	创建 DID Document
updateDID	did: string	void	更新 DID Document
getDIDDocument	did: string	DIDDocument	获取 DID Document
getDIDDocument	did: string	boolean	判断 DID Document 是否存在
registerCredential	issuerDid: string, credential: string	void	在区块链中注册某凭证

装 订 线

checkCredential	issuerDid: string, credential: string	boolean	检查某凭证是否在区块链中注册

图 4.13　凭证注册智能合约接口

4.5.2　NFT 票据智能合约

函数名	参数	返回值	描述
createNFTicket	movie: string, expirationDate: string, ownerDid: string	string	创建 NFT 票据，返回 NFT 票据的 ID
queryNFTicket	ticketId: string	NFTicket	获取 NFT 票据
nfticketExists	ticketId: string	boolean	判断 NFT 票据是否存在
verifyNFTicketOwner	ticketId: string, signedTickedId: string	boolean	验证 NFT 票据的所有者
getNFTicketsByOwner	ownerDid: string	NFTicket[]	获取用户的所有 NFT 票据

图 4.14　NFT 票据智能合约接口

5 总结

本文详细探讨了基于分布式数字身份电影院售票系统的设计与实现。

在背景调研小节，分析了数字身份技术在元宇宙以及未来的生活存在重要意义，并探究了传统的数字身份系统和存在的问题，包括隐私泄露、身份造假和高管理成本等。通过对区块链技术、W3C 分布式数字身份标准和零知识证明技术的研究，为后续的系统设计提供了理论基础。

在场景分析小节，提出了一个基于分布式数字身份的电影院售票系统的场景，该系统包括用户、公安局、电影院和区块链网络四个参与方。用户可以通过该系统实现购票、验证年龄、验证 NFT 票据的所属权等功能。公安局是唯一的 VC 发行机构，负责为用户提供出生证明。电影院提供购票、检票等服务。区块链网络用于存储用户的数字身份文档和 NFT 票据。

在技术选型小节，列举了选择的区块链技术框架、前端框架、后端框架等技术选型，并详细分析了这些技术的特点和优势。

在系统设计小节，详细介绍了系统的总体架构、数据结构设计和功能模块设计，并给出了各个功能模块的时序图。

通过本次的研究和实现，成功验证了该系统的可行性，不仅提高了安全性和隐私保护水平，还显著提升了用户体验和管理效率。

参考文献

[1] W3C. Decentralized Identifiers (DIDs) v1.0: Core architecture, data model, and representations[EB/OL]. (2022-07-19). https://www.w3.org/TR/did-core/.

[2] W3C. Verifiable Credentials Data Model v1.1[EB/OL]. (2022-03-03). https://www.w3.org/TR/vc-data-model/.

[3] JONES M. JSON Web Key (JWK)[EB/OL]. (2015-05-01). https://www.rfc-editor.org/rfc/rfc7517.

[4] BENET J, SPORNY M. The Multibase Encoding Scheme[EB/OL]. (2021-02-01). https://datatracker.ietf.org/doc/html/draft-multiformats-multibase-03.

[5] BERNERS-LEE T, FIELDING R, MASINTER L. Uniform Resource Identifier (URI): Generic Syntax[EB/OL]. (2005-01-01). https://www.rfc-editor.org/rfc/rfc3986.

[6] HYPERLEDGER. Hyperledger Fabric[EB/OL]. https://github.com/hyperledger/fabric#releases.

[7] VUE.JS. Vue.js 3[EB/OL]. https://cn.vuejs.org/.

[8] VITE. Vite[EB/OL]. https://vitejs.cn/vite3-cn/.

[9] EXPRESS. Express[EB/OL]. https://express.nodejs.cn/.

[10] RAMIĆ Š B, PRAZINA I, POZDERAC D, 等. Selective disclosure of claims from multiple digital credentials[EB/OL]. (2024). https://arxiv.org/abs/2402.15447.

[11] 曾毅. 去中心化数字身份DID简介——四、用户属性的零知识证明[EB/OL]. [2022-07-28]. https://www.cnblogs.com/studyzy/p/14211181.html.

装 订 线