ChatGPT

수탁형 지갑 백엔드 설계 (Session 2 학습 자료)

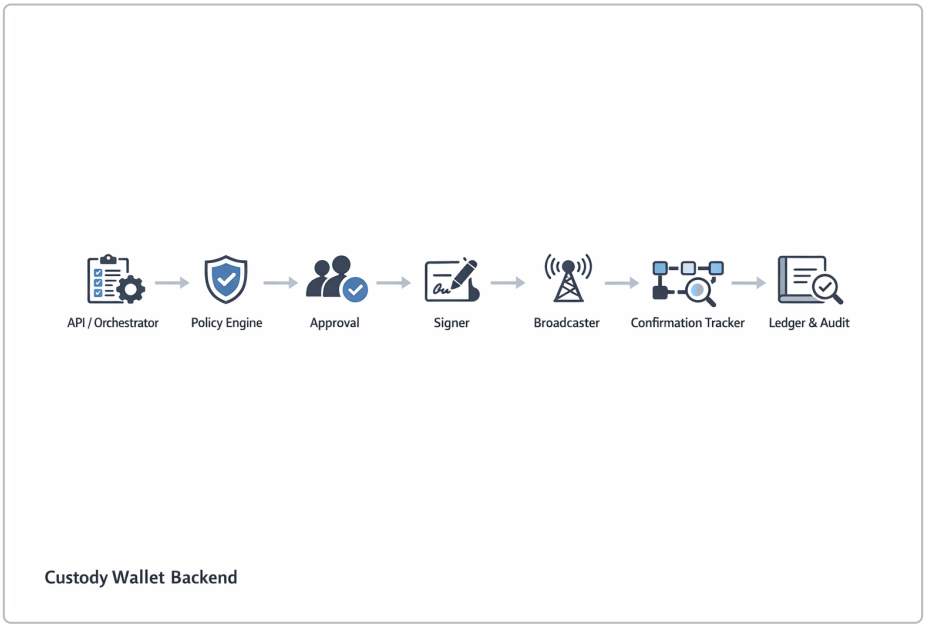
1. 서론 – 수탁 시스템의 본질

수탁형 지갑은 기업이나 서비스가 사용자 대신 디지털 자산을 보관하고 운영할 수 있게 해주는 **금융 IT 시스템**입니다.
탈중앙화된 블록체인과 달리, 수탁 시스템에서는 **권한·책임의 분리와 내부 원장**이 핵심입니다.
미국 뉴욕 금융감독청(NYDFS) 등 여러 규제 기관은 고객 자산을 온체인 지갑과 내부 원장에 별도 계정으로 관리하고, 계정별로 구분된 기록을 외부 블록체인과 정기적으로 대사(reconciliation)하도록 요구합니다 [1].
또한 Fireblocks 등 보안업체는 **다중 승인(approval)과 정책(policy) 엔진이 키 서명(signing) 기능과 분리된 구조**를 방어의 핵심 원칙으로 제시합니다 [2].

2. 표준 7 모듈 – 책임·경계·데이터

2.1 전체 아키텍처

수탁 시스템은 출금 요청이 들어온 시점부터 온체인 전파·확인·정산까지 여러 단계의 업무를 거칩니다. 아래 도식은 표준 7 모듈의 **입력–출력 경계**를 나타낸 것입니다.



도식 설명: API/Orchestrator는 사용자의 출금 요청을 업무 흐름으로 변환하고, Policy Engine과 Approval Module은 정책 판단과 사람의 승인을 분리한다. 승인된 요청만이 Signer로 전달되어 HSM 또는 MPC를 통해 서명이 생성되고, Broadcaster가 온체인에 전파한다. Confirmation Tracker는 블록체인에서 최종성을 확인하며, Ledger & Audit가 내부 잔고 변동과 감사 로그를 유지한다.

2.2 각 모듈의 역할과 경계

모듈	역할 및 책임	입력/출력	상태 저장소와 불변규칙
API/Orchestrator	외부 시스템 또는 사용자로부터 출금·조회 요청을 받아 **워크플로우로 오케스트레이션**한다. 같은 요청을 여러 번 받아도 중복 실행되지 않도록 idempotent하게 설계한다 [3] .	요청 (request) → 워크플로우 상태 (event)	워크플로우 DB/큐. **중복 실행 금지**가 불변 조건이다.
Policy Engine	금액, 주소, 속도, 화이트리스트 등 규칙에 따라 **출금이 가능한지 판단**한다. Fireblocks TAP(Transaction Authorization Policy)은 트랜잭션 이동에 대한 한도와 승인을 규정하는 룰 세트이며, 보관 솔루션보다 위에 존재하는 거버넌스 레이어이다 [4] .	출금 초안 → 허용/거절/추가검증	정책 룰 DB, 화이트리스트, 리스크 데이터. **승인 없이 Signer로 직접 전달하지 못한다**는 불변 규칙이다.
Approval	사람이 승인하는 단계(4-eyes, 한도 등). 정책을 통과한 건에 대해 승인·반려·보류를 결정한다. Fireblocks의 방어 시나리오에서 **다중 독립 디바이스 승인**과 승인 쿼럼이 내부자나 악성 애플리케이션의 공격을 방지하는 핵심 통제 포인트로 설명된다 [5] .	정책 통과 건 → 승인 여부	승인 워크플로우 DB. **승인 없이는 서명 금지**가 불변 규칙이다.
Signer	HSM, KMS 또는 MPC를 통해 **키로 트랜잭션을 서명**하는 기능만 수행한다. 서명 요청은 구조화된 포맷과 정책 승인을 증명하는 메타데이터를 포함해야 하며, signer는 승인 또는 정책을 스스로 판단하지 않는다 [6] .	서명 요청 → 서명 결과	키 저장소 (HSM/KMS/MPC)와 서명 로그. **정책·승인 로직을 모르면 서명만** 하는 것이 불변이다.
Broadcaster	서명된 트랜잭션을 블록체인 네트워크에 전파한다. 신뢰할 수 없는 네트워크와 멀티캐스트 환경에서 **idempotent 전파**가 중요하다—아티클에서는 신뢰할 수 있는 결제 시스템은 트랜잭션 상태를 명확히 모델링하고 각 단계의 전환을 idempotent하게 처리해야 함을 강조한다 [3] .	서명된 트랜잭션 → 전파 결과 (tx hash)	전파 로그/큐. **전파는 중복 전송해도 결과가 동일해야 함**이 불변 규칙이다.
Confirmation Tracker	블록체인에서 트랜잭션이 포함되고 특정 블록 깊이에서 확정되는지를 모니터링한다. ChainScore Labs는 신뢰할 수 있는 트랜잭션 플로우를 위해 **마이닝 / 확정 / 최종성 상태를 구분하고, 충분한 확인 수(N confirmations)에 도달하기 전에는 원장에 반영하지 말아야 한다**고 제안한다 [7] . 또한 nonce 관리를 통해 tx hash만이 아니라 발신자와 nonce로 트랜잭션을 추적해야 dropped/replaced 트랜잭션을 감지할 수 있다 [8] .	tx hash (+ from/nonce) → 포함/실패/교체 이벤트	체인 관측 DB. **tx hash 하나만 믿지 않음**—nonce 기반으로 추적하며, 여러 블록 확인 후 원장에 반영한다는 규칙이다.

모듈	역할 및 책임	입력/출력	상태 저장소와 불변규칙
Ledger & Audit	내부 **잔고·정산·대사·감사 기록**을 유지하는 핵심 모듈이다. CMTA 디지털 자산 수탁 표준은 "내부 원장"이 오프체인 데이터베이스로서, 고객 계정과 연결된 디지털 자산을 추적하지만 온체인 주소와 일대일 관계는 없으며, 자산 배정은 내부 원장에만 존재한다고 설명한다 [9].	모든 상태 이벤트 → 회계적 원장과 감사 로그	원장 DB(저널), 감사 로그. **체인≠원장**이며 내부 원장이 단일 진실의 근거이다.

핵심 원칙: "Policy/Approval은 '허용 여부'를 결정하고, Signer는 '서명만' 한다. **권한과 키를 같은 곳에 두지 않는다**."

3. Approval과 Signer 분리의 이유 – 사고 시나리오

3.1 분리하지 않을 때 발생하는 사고

1. **내부자 또는 키 탈취 사고**: Signer가 정책 판단까지 수행한다면, 키를 탈취한 공격자가 즉시 정책을 우회하고 대량 출금을 실행할 수 있다. NCC Group은 서명 모듈이 승인 없는 메시지 큐를 그대로 처리하는 경우 악성 요청이 승인 없이 실행될 수 있다고 경고한다 [6].
2. **피싱/오용**: 서명 요청에 승인 절차가 없다면 정상처럼 보이는 악성 요청을 막기 어렵다. 예컨대 공격자가 DeFi 사이트를 위조해 무제한 token approval을 요청하면, 정책 엔진과 승인 절차가 이를 차단하지 않는 한 키 보유자가 서명할 수 있다 [10].
3. **책임 소재 붕괴**: 정책 결정과 서명이 한 모듈에서 이루어질 경우, 사고 시 누가 어떤 이유로 승인했는지 로그를 분리해 추적하기 어렵다. 감사 추적과 포렌식 분석을 위해서는 정책 판단·사람 승인·서명 로그를 명확히 분리해야 한다 [11].

3.2 분리했을 때 얻는 통제 포인트

- **정책·승인 워크플로우**: Policy Engine과 Approval Module은 출금 한도, 속도 제한, 화이트리스트 등을 통해 1차적으로 위험을 차단한다. Fireblocks 보고서는 다중 기기 승인을 이용해 각 승인자가 독립된 하드웨어 디바이스에서 확인하도록 설계하면 공격자가 단일 워크스테이션을 장악해도 승인 프로세스를 통과할 수 없다고 강조한다 [2].
- **Signer의 단순화**: Signer는 입력 포맷 검증과 서명만 수행하고, 정책을 모른다. 정책이 부착된 메타데이터를 검증하거나, 승인자의 서명 또는 토큰을 체크하면 서명 권한이 있는지 알 수 있다. 이렇게 하면 서명 모듈 자체를 심플하게 하여 공격면을 줄일 수 있다 [6].
- **포렌식 가능한 로그 체인**: 정책 판단 로그, 승인 로그, 서명 로그가 분리되어 있으면, 사고 발생 시 trace ID를 따라 "누가 요청했고, 어떤 정책이 적용됐으며, 누가 승인했고, 어떤 키로 서명했는지, 블록체인 결과가 무엇인지"를 재구성할 수 있다. 이는 감사 요건의 핵심이다 [11].

3.3 MPC·다중 승인 구현 사례

- **다중 승인 / 다중 디바이스**: Fireblocks defense-in-depth 모델은 다중 독립 디바이스 승인과 정책 엔진을 결합해 악성 내부자나 피싱 공격을 차단한다. 승인 쿼럼이 요구되며 각 승인자는 본인의 하드웨어-보안 장치에서 생체 인증을 통해 승인해야 한다 [5]. 이를 통해 한 사람이 단독으로 출금을 허용할 수 없도록 한다.
- **MPC의 역할**: MPC 지갑은 키를 여러 조각으로 나누어 분산 저장하며, 정책 기반 트랜잭션 승인을 지원한다. Safeheron 가이드에 따르면 MPC 지갑은 **다중 승인과 정책 기반(Transaction approvals)**을 지원해 컴플라이언스와 제어력을 향상한다 [12]. 또한 Scalable Solutions blog는 MPC 기반 시스템이 **정책 기반 승인, 지리적 분산, 역할 분리**를 적용해 MiCA 및 NYDFS의 분리 요건을 충족한다고 설명한다 [13].

4. 내부 원장(Internal Ledger) 설계 포인트

초보 설계자가 가장 많이 실수하는 부분은 **"체인 잔고 = 우리 잔고"라는 오해**이다. 실제로는 내부 원장이 단일 진실이며 블록체인은 외부 증빙 역할을 한다 [9]. 내부 원장은 계정·잔고·거래 이벤트를 기록하는 저널 형태의 데이터베이스로 설계해야 한다. 아래에서 잔고, 정산, 대사, 감사 측면을 살펴본다.

4.1 잔고(Balance)

블록체인 주소 잔고는 회사 전체의 총 보유량만을 보여주며, 고객별 잔고를 알 수 없다. 따라서 수탁 시스템에서는 **계정별 내부 잔고**를 유지해야 한다 [9]. 잔고는 "스냅샷 값"보다 **저널 기록(변동 이벤트)** 중심으로 설계해야 나중에 모든 상태 변화를 재생할 수 있다.

잔고 상태 모델 – 여러 핀테크 사례와 결제 용어집에서 잔고를 여러 상태로 구분한다:

- **Posted/Settled Balance**: Modern Treasury는 **posted balance**가 완전히 정산된 거래의 합계로 내부 원장 대사(reconciliation)의 기준이며, 최종 상태의 자금이라고 설명한다 [14]. 이 상태는 블록체인에서 충분한 컨펌 후 '정산 완료'에 해당한다.
- **Pending Balance**: 같은 기사에서 **pending balance**는 결제망을 통해 이동 중인 자금(미정산/미취소 포함)을 의미하며 [15], 일부는 실패할 수 있어 신뢰도가 낮다. Nomupay 용어집도 **pending balance**를 "유입된 자금이 리스크 평가 중인 상태"로 정의한다 [16].
- **Available Balance**: Modern Treasury는 **available balance**가 사용자가 즉시 사용할 수 있는 금액으로, 출금 중(pending outflows)인 금액을 미리 차감해 표시한다고 설명한다 [17]. Nomupay는 **available balance**가 제재 스크리닝을 통과한 자금으로 즉시 지불에 사용할 수 있는 잔고라고 정의한다 [18].
- **Reserved Balance**: Nomupay와 Rapyd Docs는 **reserved balance**를 차지백·환불 등 법적 책임을 위해 따로 보유하는 잔고로 설명한다 [19] [20].
- **Received Balance/Under Review**: Rapyd는 **received balance**를 아직 사용 가능하지 않은 입금액으로 정의하고 [21], Fipto는 별도의 "Under Review balance"에 유입된 자금을 보류한 후 검증이 완료되면 고객 계정으로 이동시킨다고 설명한다 [22].

따라서 수탁 시스템에서는 **Available / Reserved / Pending / Settled** 네 가지 상태로 잔고를 구분하는 것이 권장된다. 각 상태는 저널에 기록되고, 상태 변환은 불변 규칙과 함께 관리해야 한다.

4.2 정산(Settlement)

정산 정책은 출금 요청 시점에 잔고를 즉시 차감할지(예약/홀드) 아니면 체인에서 컨펌 후 차감할지를 정의한다.

- **예약/홀드**: 사용자 출금 요청을 받으면 우선 `Reserved` 상태에서 잔고를 확보하고 이후 체인에서 성공 시 `Pending → Settled`로 이동한다. 이는 오버스펜딩을 방지한다.

- **후정산**: 일부 서비스는 체인 컨펌 후 잔고를 차감하는데, 이 경우 체인 전파 실패 시 롤백 로직이 필요하다.

정산 정책은 자금 이탈 리스크와 고객 경험 사이에서 균형을 잡아야 한다.

4.3 대사(Reconciliation)

대사는 내부 원장과 블록체인 상태, 거래소·은행 계좌를 매칭하는 작업이다. Lightspark는 **대사가 두 개 이상의 원장을 맞추는 과정이며**, 블록체인에서는 내부 기록과 퍼블릭 블록체인을 일치시키는 것이라고 설명한다 [23]. TRES Finance는 NYDFS, MAS, FSA 등 여러 규제기관이 **일일 또는 근접한 주기의 대사**를 요구하고, 여러 원천(지갑, 거래소, 은행)을 아우르는 **다중 소스 대사**가 표준이 되고 있다고 강조한다 [24]. 대사 과정에서는 타이밍 지연·재편

(reorg)·삭제된 트랜잭션에 대한 예외 큐와 사람이 개입하는 프로세스가 필요하다. ChainScore Labs는 **충분한 블록 확인 후에만 잔고를 갱신하고, 재편(reorg) 발생 시 로그와 영수증을 재검증**해야 한다고 지적한다 [7].

4.4 감사(Audit)

감사 기록은 규제 요구사항과 사고 대응에 필수이다.

- **누가 무엇을 요청했는지**: API / Orchestrator는 요청자 정보, 목적, 시간 등을 기록해야 한다.
- **정책 근거**: Policy Engine은 적용된 규칙과 화이트리스트 정보, 리스크 평가 데이터를 로그로 남긴다.
- **누가 승인했는지**: Approval Module은 승인자 ID, 순서, 시간, 코멘트 등을 저장한다. Fireblocks는 다중 디바이스 승인과 생체 인증을 통해 각 승인자가 독립적으로 인증하도록 하여 감사를 돕는다 [5].
- **어떤 키로 서명했는지**: Signer 모듈은 HSM/MPC ID, 서명 요청 ID, 서명 결과를 로그에 남긴다. NCC Group은 승인 토큰을 포함한 서명 요청을 검증해야 함을 강조한다 [6].
- **체인 결과**: Confirmation Tracker는 tx hash, 블록 높이, nonce, 재편 여부 등의 정보를 저장해 외부 증빙으로 삼는다. ChainScore Labs는 nonce 기반 추적과 충분한 컨펌 후 갱신을 권장한다 [7].

이 모든 로그는 **trace ID**로 연결되어야 하며, 하나의 사고에 대한 포렌식을 가능하게 해야 한다.

5. 실전 워크시트 질문

강의 도중 수강생이 스스로 서비스 아키텍처를 설계해 볼 수 있도록 다음 질문을 워크시트로 제공한다. 각 질문은 7 모듈과 내부 원장 설계를 자신의 시스템에 맞게 적용하는 데 도움이 된다.

1. **우리 서비스에서 Policy Engine은 무엇을 기준으로 출금 요청을 통과/거절/추가 검증하는가?** – 화이트리스트, 한도, 속도 제한, 리스크 점수 등.
2. **Approval은 몇 단계로 구성되는가?** – 1인 승인, 2인(4-eyes) 승인, 긴급 해제 프로세스 등.
3. **Signer는 어떤 형식의 요청만 받아들이는가?** – verifyingContract, chainId, policy approval proof가 포함된 정형화된 서명 요청 포맷.
4. **Ledger의 상태는 어떤 네 단계로 나뉘는가?** – Available / Reserved / Pending / Settled 모델 적용 여부.
5. **대사 불일치 발생 시 어디로 라우팅하고 어떻게 해결하는가?** – 자동 재시도 큐 vs. 사람 개입, 예외 대기열.

6. 강의 운영 제안 (1시간)

시간	내용	설명
0–10 분	**수탁 시스템의 본질**	권한·책임 분리의 중요성, 규제 배경. 예: NYDFS의 분리·대사 요구사항 [1].
10–35 분	**7 모듈 설명**	각 모듈의 경계·책임·입출력·불변 규칙을 도식과 함께 설명. 정책 엔진과 승인, 서명자 역할 분리 강조.
35–50 분	**Approval vs Signer 분리**	사고 시나리오(내부자 탈취, 피싱, 책임소재 붕괴)를 소개하고, Fireblocks와 NCC Group 등의 사례로 설득 [2] [6].
50–60 분	**내부 원장 설계 & 워크시트 작성**	잔고 상태 모델(Available/Reserved/Pending/Settled)을 설명하고, 정산·대사·감사 설계 포인트를 논의 [14] [18] [7]. 워크시트 질문에 대해 조별로 논의.

7. 마무리 – 핵심 요약

- **경계가 곧 보안이다**. 각 모듈은 명확한 책임과 입력·출력을 갖고, 다른 모듈의 내부 상태를 알 필요가 없다. idempotency와 상태 기계(State Machine)가 모든 단계에서 중요하다 [3].
- **권한과 키를 분리**. Policy/Approval은 "허용 여부", Signer는 "서명만" 한다. 다중 승인, 정책 기반 엔진, 독립된 디바이스 인증을 통해 내부자·피싱·악성 코드를 방어한다 [5] [6].
- **내부 원장이 단일 진실**. 고객별 잔고는 블록체인 잔고와 동일하지 않으며, 네 가지 잔고 상태를 관리해야 한다 [9] [14] [25]. 대사는 다중 원천을 일치시키는 과정이며, 충분한 블록 컨펌과 재편 처리 로직이 필요하다 [7].
- **감사 가능성을 내재화**. 모든 단계의 로그를 trace ID로 연결하고, 누가 무엇을 했는지 기록해 규제·감사 요구에 대응한다.

이 자료를 통해 백엔드 개발자는 수탁형 지갑의 핵심 아키텍처와 설계 포인트를 이해하고, 실전에서 발생할 수 있는 질문과 돌발 상황에 대응할 수 있는 준비를 할 수 있을 것이다.

---

[1] Industry Letter - January 23, 2023: Guidance on Custodial Structures for Customer Protection in the Event of Insolvency | Department of Financial Services
https://www.dfs.ny.gov/industry_guidance/industry_letters/il20230123_guidance_custodial_structures

[2] [5] [10] [11] Securing Digital Assets in an Evolving Threat Landscape: The Fireblocks Defense-in-Depth Approach to Security | Fireblocks
https://www.fireblocks.com/report/the-fireblocks-defense-in-depth-approach-to-security

[3] The Key Elements of a Reliable Crypto Payment Gateway: By Kevin S
https://www.finextra.com/blogposting/30833/the-key-elements-of-a-reliable-crypto-payment-gateway

[4] Digital Asset Custody and Transaction Processing Leading Practices Using Fireblocks' MPC solution | Fireblocks
https://www.fireblocks.com/report/digital-asset-custody-and-transaction-processing-leading-practices-using-fireblocks-mpc-solution

[6] State of the Art of Private Key Security in Blockchain Ops - 3. Private Key Storage and Signing Module | NCC Group
https://www.nccgroup.com/research-blog/state-of-the-art-of-private-key-security-in-blockchain-ops-3-private-key-storage-and-signing-module/

[7] [8] How to Design High-Reliability Transaction Flows | ChainScore Guides | ChainScore Labs
https://www.chainscorelabs.com/en/guides/core-blockchain-concepts-and-architecture/transaction-lifecycle/how-to-design-high-reliability-transaction-flows

[9] cmta-digital-assets-custody-standard-dacs-en-approved-1st-may-2025.pdf
https://cmta.ch/content/9b489294d229f53521369a7f9e6387d5/cmta-digital-assets-custody-standard-dacs-en-approved-1st-may-2025.pdf

[12] Exploring the Basics of MPC Digital Wallet Infrastructure
https://safeheron.com/blog/mpc-digital-wallet-infrastructure-basics-and-how-it-works/

[13] Secure Multiparty Computation: the new standard for institutional digital asset security | Blog | Trends, Insights & Innovations in Digital Assets
https://scalablesolutions.io/blog/posts/mpc-asset-security

[14] [15] [17] Fintech Eng Challenges, Part I: Different Balance Types in a Wallet
https://www.moderntreasury.com/journal/fintech-eng-challenges-part-i-different-balance-types-in-a-wallet

[16] [18] [19] [25] Glossary – Nomupay
https://support.nomupay.com/hc/en-gb/articles/8572968740753-Glossary

20 21 Viewing Client Wallet Balances
https://docs.rapyd.net/en/viewing-client-wallet-balances.html

22 Fipto | Inside Fipto's Ledger System
https://www.fipto.com/articles/inside-fiptos-ledger-system

23 Reconciliation Explained: Financial Accuracy via Lightspark Grid | Lightspark
https://www.lightspark.com/glossary/reconciliation

24 Daily Digital Asset Reconciliation Is No Longer Optional | Crypto Accounting and Web3 Treasury | TRES Finance
https://tres.finance/daily-digital-asset-reconciliation-is-no-longer-optional/