



Peking University HSBC Business School (PHBS)

PAO-BU-LIAO

A Smart Contract-based Management System for
Advance Receipts and Accounts Receivable.

Instructor: Professor Zeng HaiYang

Topic: Project Type 3

Group 2 (names in no specific order)

Ren Kai (任凯)	2401212437
Zhang WeiJie (张伟杰)	2401212474
Jiang YiMing (姜一鸣)	2401212398
Hu Rui (胡瑞)	2401212392
Lan Yang (兰杨)	2401212401



Block Chain and Digital Currency

26th April 2025

1. Introduction

“Pay First, Use Later” and “Use First, Pay Later” are two common business models in everyday life, where there is a time lag between the receipt of cash and the occurrence of services. This creates problems in managing the buyer's **Advance Receipts** and the seller's **Accounts Receivable**, as shown in Figure 1.1. Under the traditional payment system, these problems are based on a plain trust between buyer and seller, which is often not sturdy and **runaways** are always occurring. As shown in Figures 1.2 and 1.3, the number and amount of runaway cases in gyms, coffee shops and milk tea stores, which are represented by rechargeable membership cards, **have been increasing yearly** in recent days. But the recovery rate of arrears averaged less than 20 percent!

The essential reason for this phenomenon is that under the traditional payment model, the flow of funds is untraceable and can be used arbitrarily. In this semester's course, we gained an in-depth understanding of how smart contracts work. Taking into account the advantages of their **non-tamperability**, they can therefore be used to develop management tools for advance receipts and accounts receivable that can truly record the flow of cash, which is the application that will ultimately be presented in this project - Paobuliao.

As the project framework in Figure 1.3, we will describe the business construction in detail in **Part 2**, show the technical implementation in **Part 3**, and draw a conclusion in **Part 4**.

2. Business Construction

In essence, Paobuliao is a **smart contract-based online transaction platform**. The overall framework is shown in Figure 2.1. It supports **2 transaction models**: Use First, Pay Later and Pay First, Use Later. The platform covers **4 core operational scenarios**: Verification, Contract Establishment, Daily Consumption, and Contract Termination. It is designed to effectively address real-world issues where **sellers abscond with prepaid funds** after buyers purchase stored-value cards.

To ensure logical clarity, this report will primarily focus on the Pay First, Use Later model. A brief explanation of the Use First, Pay Later model will be provided at the end.

2.1 User Registration and Identity Verification

Both buyers and sellers must register before using the platform. To ensure a **one-to-one correspondence** between users and accounts, buyers are required to submit **biometric information** (e.g., fingerprints or iris data) during registration, while sellers must provide valid **business qualifications** (Figure 2.2). These identity materials are encrypted using the user's **private key** and uploaded to the smart contract for account creation. After onboarding, sellers

may publish and update their available service offerings on the platform for buyers to browse and contract.

2.2 Contract Establishment and Daily Consumption

When a buyer decides to purchase a stored-value card for a particular service, they initiate a **Contract Establishment Request** to the smart contract. Upon approval, the designated funds are transferred from the buyer's account into a dedicated escrow account managed by the smart contract (Figure 2.3). This step corresponds to the real-world process of purchasing a **stored-value card**, but importantly, the funds are **not immediately transferred** to the seller.

Following this, the buyer may consume services based on the contract terms and their remaining balance. Payments to the seller are disbursed incrementally from the buyer's balance as services are rendered (Figure 2.4).

2.3 Contract Termination

If the buyer submits a **Contract Termination Request** to the smart contract. Upon receiving the request, the smart contract notifies the seller and initiates the refund process, **returning the remaining balance** to the buyer (Figure 2.5). On the other hand, if the **seller absconds**, the contract will be terminated automatically. The system identifies seller default through two mechanisms. **About Active Recognition**, the smart contract regularly verifies the validity of the seller's business qualifications. If the qualification is found to be revoked, the contract is immediately terminated and the remaining buyer balance is refunded. **About Passive Recognition**, buyers are allowed to report seller misconduct through the platform. If the number of complaints exceeds a defined threshold, the system determines that the seller has defaulted, and the buyer's remaining funds are refunded accordingly (Figure 2.6).

2.4 Use First, Pay Later Model

The Use First, Pay Later model differs mainly in that the buyer **does not deposit funds** into the smart contract in advance. Instead, service fees are settled periodically (e.g., monthly or annually), and there is a **risk of buyer default**. However, since each buyer account is bound to their biometric information, legal accountability can be more easily pursued in the case of non-payment. Additionally, unlike the Pay First, Use Later model, no refunds are involved upon contract termination in this case.

3. Technical Implementation

3.1 Implementation Logic

This system is designed around two payment models: “Pay First, User Later” and “Use First, Pay Later”. Corresponding smart contract frameworks have been developed for each model. A centralized Manager Contract is implemented to monitor and regulate user behavior and merchant operations, ensuring transaction security and compliance. The technical framework is illustrated in Figure 3.1. And we will explain them step by step.

3.1.1 Core Contract Structure

The system of Paobuliao consists of the following four main contract modules:

- Accounts Receivable Contract (Use First, Pay Later): Record the user's pending debt (`pendingDebt`) and enable settlement and seller balance updates by `payDebt()` function.
- Advance Receipts Contract (Pay First, User Later): Freeze user funds in advance via `deposit()` to ensure cash availability. The `consume()` function completes the consumption.
- Manager Contract: Executes user and merchant credit assessments and violation penalties using a combination of periodic checks and event-driven mechanisms.
- Record Contract: Serves as the system's database, storing key information such as user identity, public keys, balances, and merchant qualifications.

3.1.2 Account and Identity Authentication Mechanism

To ensure secure and reliable participation in the system, both buyers and merchants must undergo a standardized identity authentication process.

Buyer Registration Process: Buyers complete identity verification either through biometric data or by generating a cryptographic key pair. Upon successful verification, the system stores relevant user information, including `userId`, `public key`, and initial balance.

Merchant Registration Process: Merchants are required to upload qualification documents and generate a key pair. After verification, the system stores their information in the `SellerInfo` struct, which includes the `merchantId`, a `hash` of the qualification documents, the merchant's `public key`, and `current status`.

3.1.3 Consumption and Payment Logic

The system supports two distinct transaction models — Use First, Pay Later and Pay First, Use Later — each with tailored contract logic to ensure smooth and secure payment execution.

Use First, Pay Later Flow

- a. **`consume(userId, amount)`:** The consumption amount is added to the buyer's `pendingDebt`.

- b. **payDebt(userId, amount)**: The function verifies that the repayment amount does not exceed the pending debt. Upon successful payment, both the pendingDebt and the corresponding sellerBalance are updated accordingly.

Pay First, Use Later Flow

- a. **deposit(userId, amount)**: Before consumption, the user deposits funds, which are deducted from their available balance and frozen to guarantee future payment.
- b. **consume(userId, amount)**: The system checks if the frozen funds are sufficient; if validated, the amount is transferred to the seller to complete the transaction.

3.1.4 Credit Scoring Mechanism

To encourage responsible behavior and maintain the integrity of the platform, a dynamic credit scoring system is applied to both buyers and merchants.

- a. Buyer Score: If a buyer fails to repay their pendingDebt within the designated timeframe, their credit score will be reduced accordingly.
- b. Merchant Score: Negative reviews or frequent complaints will negatively impact a merchant's credit score.

Should any participant's credit score fall below a defined threshold, the system may trigger smart contract restrictions, such as account suspension or temporary trading bans. This scoring mechanism serves as a deterrent against misconduct and promotes trust and fairness.

3.1.5 Contract Termination Mechanism

To ensure contract lifecycle management and mitigate risk, the system incorporates multiple termination scenarios:

- a. Voluntary Termination by User: Users may choose to terminate their contracts, but must accept penalties as outlined in the contract terms. The Manager Contract will automatically execute the termination and enforcement process.
- b. Automatic Termination upon Contract Expiry: Contracts will be automatically closed once they reach their expiration date.
- c. Merchant Violation or Exit: If a merchant is found to be acting maliciously, such as receiving a complaint rate over 60% or being flagged for abnormal behavior, the Manager Contract will deactivate or revoke their account to protect users.

This multi-faceted termination mechanism ensures that the system remains orderly, secure, and responsive to both voluntary exits and misconduct.

3.2 Demonstration

The mobile interface was built using Vue.js, Vite, and CSS, ensuring fast performance and a responsive, user-friendly layout. As illustrated in Fig 3.2, the registration interface supports

both users and merchants (via biometric data and business license uploads). Fig 3.3 shows the main page, offering quick access to contract-related features. Users can initiate and confirm smart contracts through Fig 3.4, with successful creation displayed in Fig 3.5, Fig 3.6 presents a daily consumption subpage that showcases real-time transaction progress, payment statuses, and smart contract execution stages, while Fig 3.7 handles contract termination, indicating user or merchant-initiated exits. Finally, Fig 3.8 displays transaction data observed from the backend, ensuring transparency and traceability.

4. Conclusion & Extension

This project proposes Paobuliao, a decentralized service platform that addresses trust issues in prepaid and postpaid contractual agreements through the integration of cryptographic technologies, smart contracts, and community governance. To solve common market failures, such as merchant fraud, user default, and lack of legal recourse in low-value service transactions, Paobuliao introduces a novel framework with both technological depth and practical usability. The key technical innovations include:

- **Biometric Identity + Public Key Encryption**
- **Dual Contract Modes** (“Pay Before” & “Pay After”)
- **On-Chain Penalty Enforcement**
- **Community-Powered Merchant Exit Detection**

The technological design of Paobuliao not only resolves long-standing trust issues in service-based economies but also introduces a **scalable, programmable trust protocol** that can generalize across industries. It demonstrates how smart contracts can enforce nuanced behaviors like conditional payments, penalty rules, and consensus-based contract termination. Furthermore, by embedding identity verification and governance logic into the protocol itself, the system reduces reliance on third-party arbitration and opens possibilities for **jurisdiction-light, self-enforcing agreements**.

In a market often plagued by unreliable prepayment practices and enforcement asymmetry, *Paobuliao* offers a blueprint for restoring balance. It redefines how trust is built, enforced, and visualized — particularly in sectors underserved by traditional legal frameworks. From both a design and systems perspective, this project showcases how combining biometric identity, smart contracts, and community mechanisms can lead to **more equitable digital service environments**.

Appendix

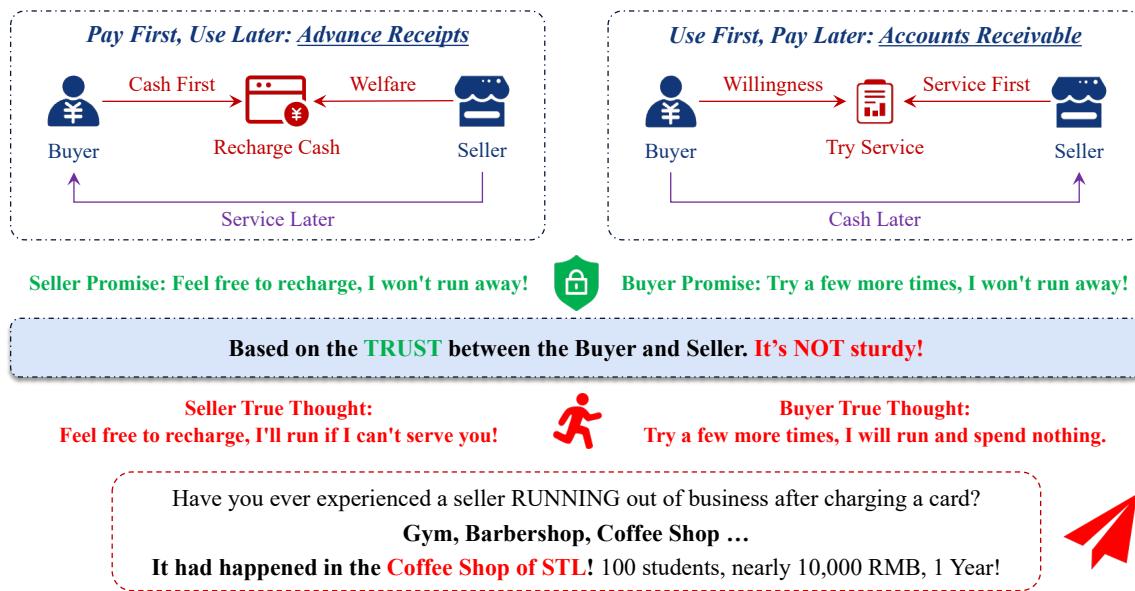


Figure 1.1 Runaway dilemma of “Pay First, Use Later” and “Use First, Pay Later” business models.

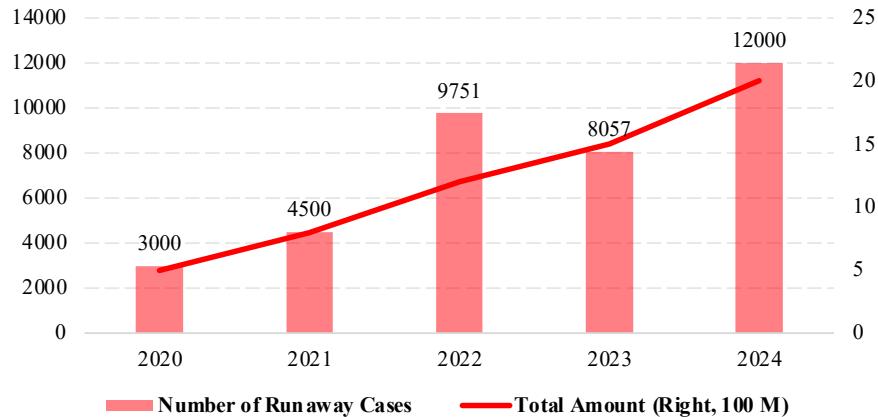


Fig 1.2 Statistics on Prepaid GYM Run Away.

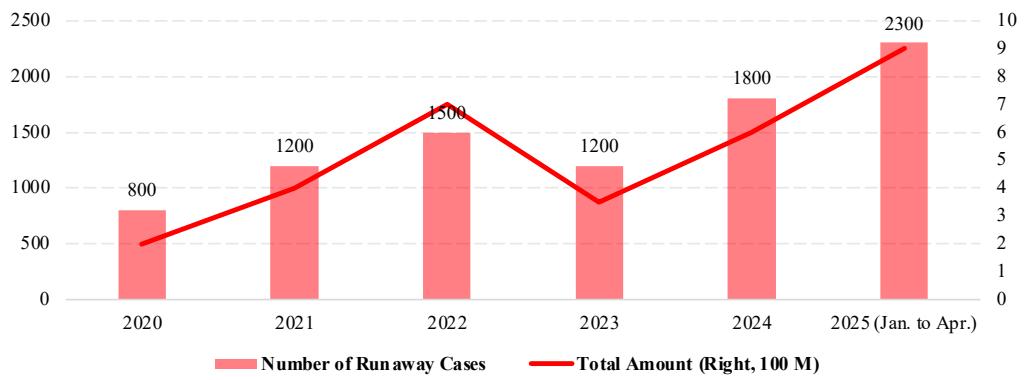
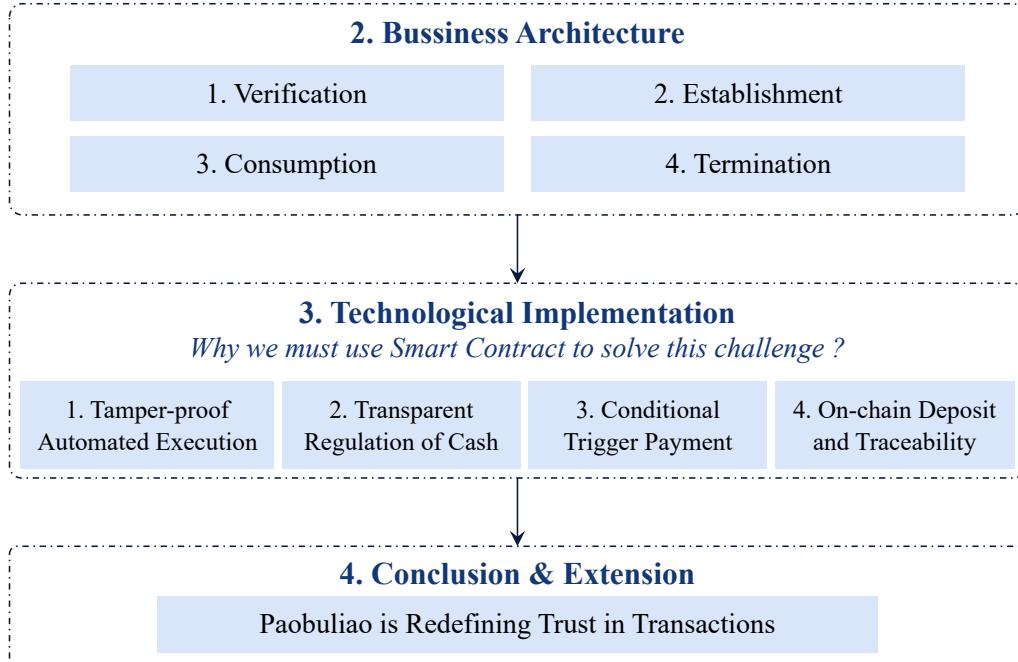
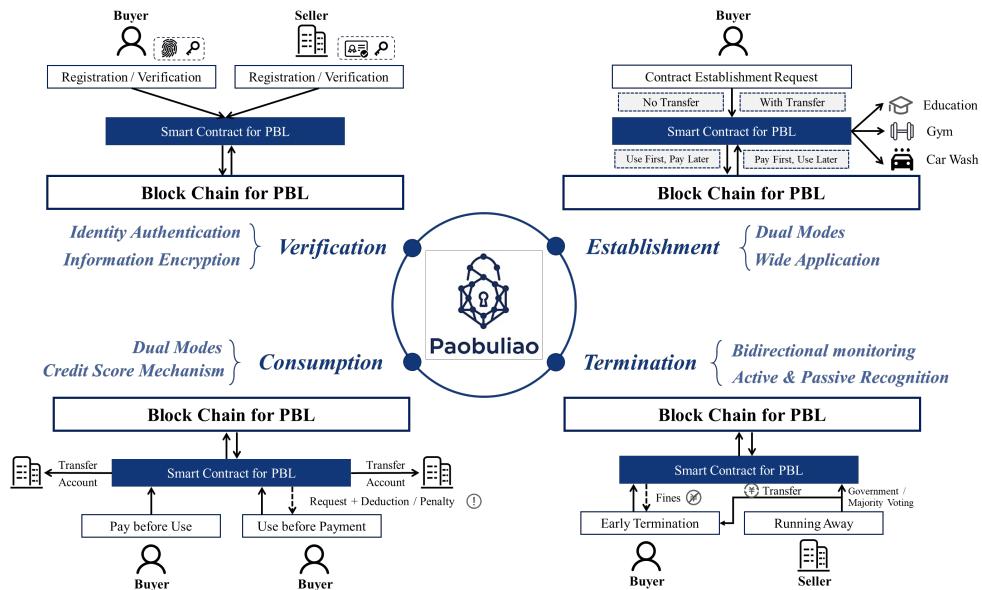
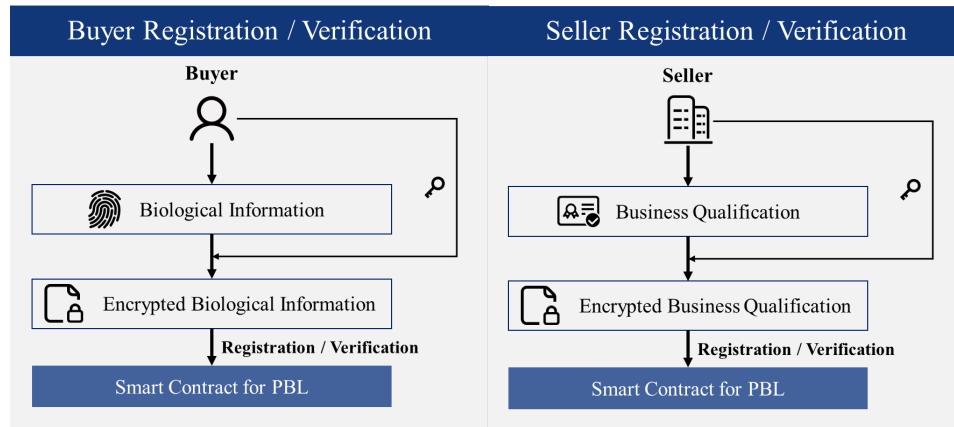
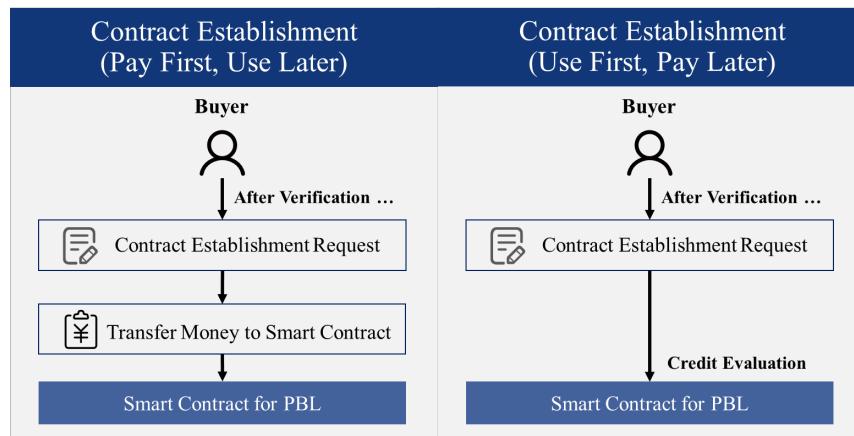
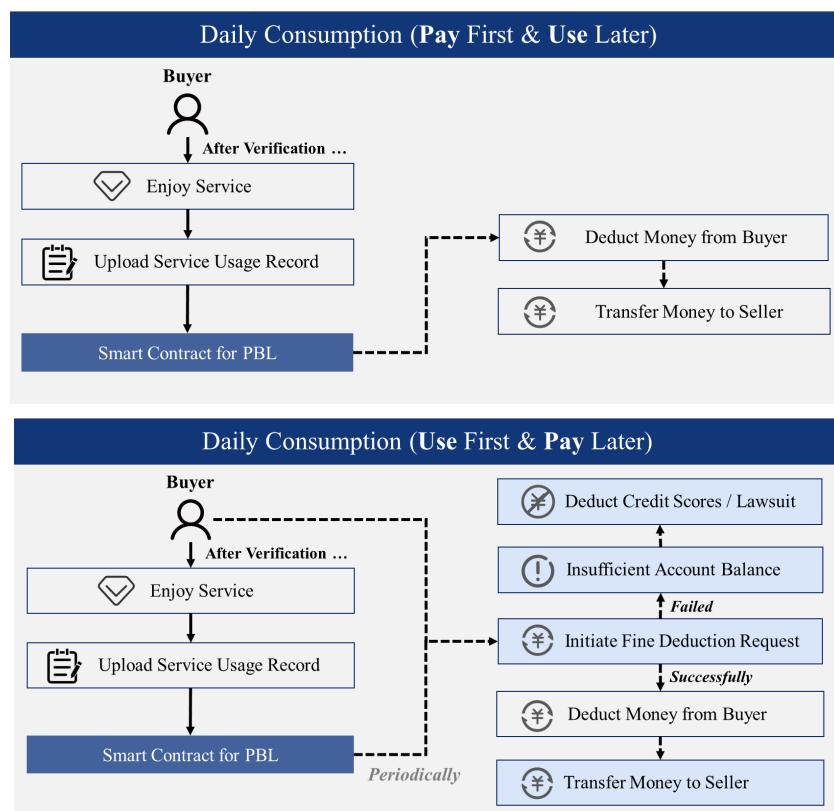


Fig 1.3 Statistics on Prepaid Coffee & Milk Tea Shop Run Away.

**Fig 1.4** The framework of our project.**Fig 2.1** The overall framework business construction.

**Fig 2.2** The logic of **registration** process of buyer and seller.**Fig 2.3** The logic of **contract establishment** process.**Fig 2.4** The logic of **daily consumption** process.

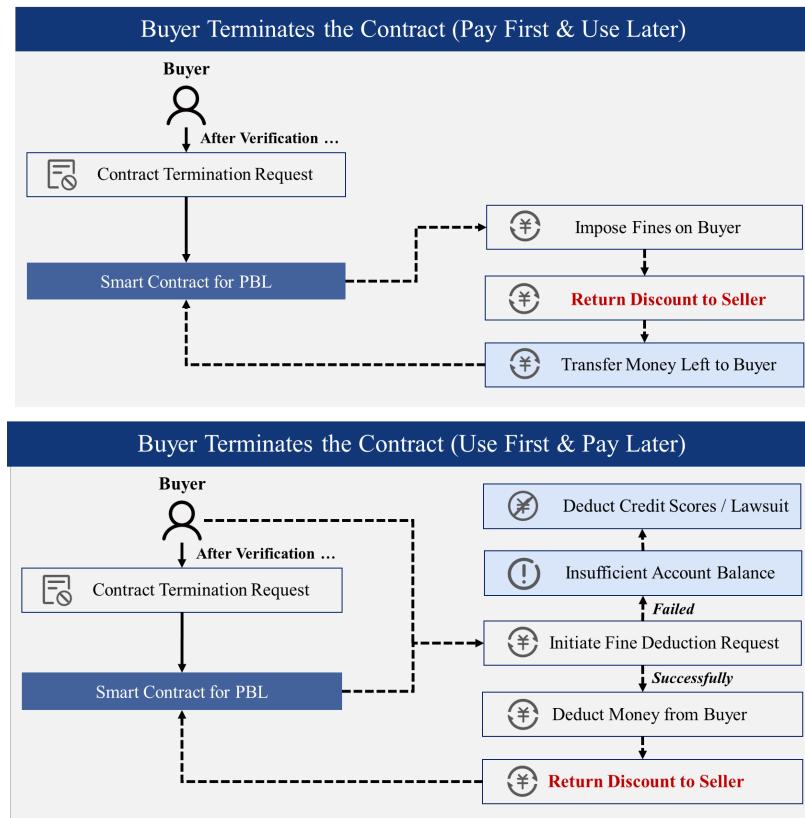


Fig 2.5 The logic of **contract termination** process (initiated by **buyer**).

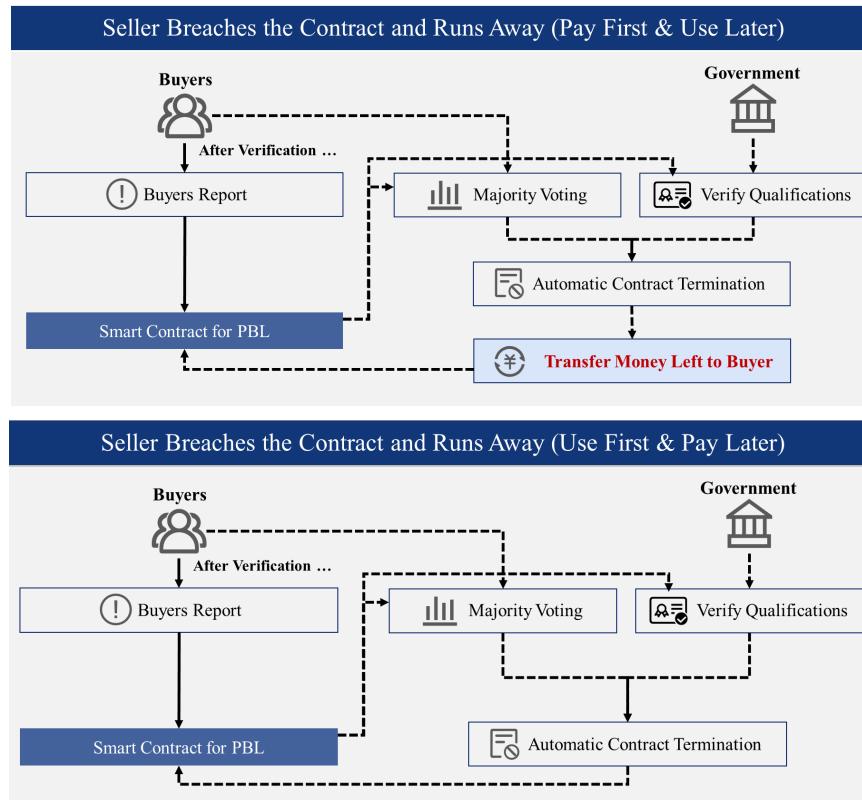


Fig 2.6 The logic of **contract termination** process (initiated by **seller**).

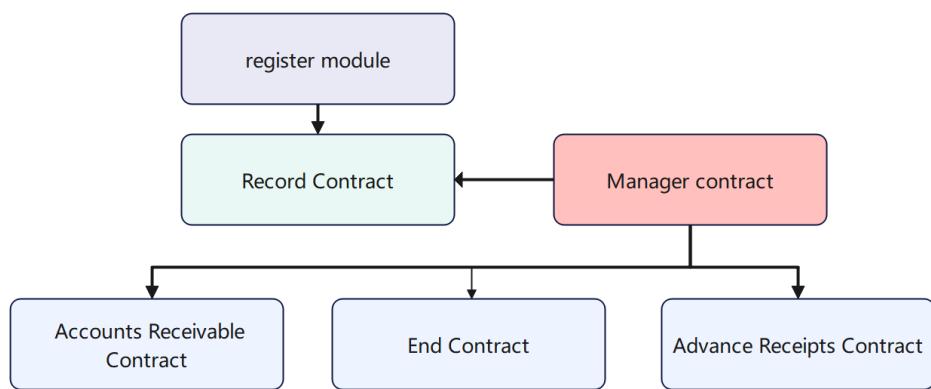


Fig 3.1 The framework of implementation logic.

The image shows the Paobuliao DApp interface, which is a DeFi Smart Contract Platform. The top navigation bar includes the platform name, balance (\$2,450.00), credit score (850), and a 'My Account' button.

Secure Prepayments with Blockchain Technology

Paobuliao DApp prevents default on prepaid services using smart contracts for education, fitness, automotive services and more.

[Get Started](#) [Learn More](#)

Why Choose Paobuliao?

- Secure Payments**: Smart contracts ensure your prepaid services are protected and funds are released only upon service delivery.
- Transparent Tracking**: Monitor your service usage and payment progress in real-time through our blockchain-based platform.
- Trust & Reliability**: Build trust between service providers and customers with our automated contract enforcement system.

User Registration

User registration fields include First Name (yang), Last Name (lan), Email Address (ddd@pku.edu.cn), Phone Number (1829203900). Below these are Biometric Verification and Document Verification sections.

Merchant Registration

Merchant registration fields include Business Name, Business Type (Education), Business Email, Business Phone, Business Credentials (Upload Business Documents), Business Profile (Upload Business Images). Below these are Biometric Verification and Document Verification sections.

Footer

Paobuliao is a decentralized application based on smart contracts to protect prepaid services and prevent defaults.

Quick Links: How It Works, For Merchants, For Users, Smart Contracts.

Support: Help Center, Contact Us, FAQ, Privacy Policy.

Connect With Us: Social media links (Twitter, Facebook, LinkedIn, GitHub), newsletter subscription form (Your email, Subscribe).

Payment methods: Visa, Mastercard, American Express, Discover, Apple Pay.

Fig 3.2 The interface for registration(Buyers/Sellers).

The screenshot displays the Paobuliao DApp's main interface. At the top, a navigation bar includes links for "Contract Agreement", "Daily Consumption", "Contract Termination", and "My Account". It also shows the platform name "Paobuliao", its description as a "DeFi Smart Contract Platform", a balance of "\$2,450.00", a credit score of "850", and a green upward arrow icon.

The main content area features a title "Secure Prepayments with Blockchain Technology" and a subtitle explaining that Paobuliao DApp prevents default on prepaid services using smart contracts for education, fitness, automotive services and more. Below this are two buttons: "Get Started" and "Learn More". To the right is a graphic illustrating a blockchain network with interconnected hexagonal nodes.

A section titled "Why Choose Paobuliao?" follows, containing two options: "Pay After Use" and "Pay Before Use".

- Pay After Use:** Pay only after you've received the service, secured by smart contract.
 - No upfront payment required
 - Automatic payment release after service completion
 - Protected by credit score system
- Pay Before Use:** Traditional prepayment with smart contract protection against defaults.
 - Protected prepayment with escrow
 - Automatic refund if service not delivered
 - Potential discounts from merchants

The interface then transitions to a "Select Service Category" section, which lists eight categories with corresponding icons:

- Education (Schools, courses, tutoring services)
- Fitness & Gym (Gym memberships, fitness classes)
- Car Services (Car wash, maintenance, repair)
- Food & Beverage (Restaurants, cafes, meal plans)
- Beauty & Spa (Salons, spas, beauty treatments)
- Housing (Rent, utilities, maintenance)
- Entertainment (Streaming, events, subscriptions)
- Other Services (Custom service agreements)

Below these categories is a "Construct Contract" button.

The footer contains a "Paobuliao" section describing the platform as a decentralized application based on smart contracts to protect prepaid services and prevent defaults. It also includes "Quick Links" for How It Works, For Merchants, For Users, and Smart Contracts, as well as "Support" links for Help Center, Contact Us, FAQ, and Privacy Policy. A "Connect With Us" section features social media icons and a newsletter subscription form. The footer also includes payment method icons (Visa, Mastercard, American Express, Discover) and a copyright notice: "© 2025 Paobuliao. All rights reserved."

Fig 3.3 The interface for main page.

The screenshot displays the Paobuliao DeFi Smart Contract Platform interface, divided into two main sections: 'Contract Construction' and 'Contract Confirmation'.

Contract Construction (Top Section):

- Header:** Back to Selection, Balance: \$2,450.00, Credit Score: 850 ↑, My Account.
- Progress Bar:** Select Model (checkmark), Select Category (checkmark), Define Terms (circle with '\$'), Generate Contract (circle with '4').
- Contract Construction Section:**
 - Merchant Selection:** Search Merchants, EduTech Academy (Education, 4.8), FitLife Gym (Fitness & Gym, 4.7).
 - Contract Duration:** Start Date: 2025/05/01, End Date: 2026/04/30, Renewal Options: Notify before renewal, Early Termination Terms: Fee penalty (10% of remaining value).
 - Special Terms:** Service quality guarantee with partial refund for substandard delivery (checked), Confidentiality agreement for all shared information, Dispute resolution through platform arbitration (checked), Force majeure clause for unforeseen circumstances.
 - Custom Terms:** Add any additional terms or conditions.
 - Attachments:** Drag files here or click to upload PDF, DOC, DOCX up to 5MB, Select Files.
 - Contract Summary:** No merchant selected, Service Category: Education, Payment Model: Pay After Use, Payment Frequency: Monthly, First Payment Due: 2025/05/01, Total Contract Value: \$99.99.
 - Smart Contract Preview:** Show Code (disabled).
- Buttons:** Save Draft, Generate Smart Contract.

Contract Confirmation (Bottom Section):

- Header:** Paobuliao, Quick Links (How It Works, For Merchants, For Users, Smart Contracts), Support (Help Center, Contact Us, FAQ, Privacy Policy), Connect With Us (Twitter, Facebook, LinkedIn, YouTube, Subscribe, Newsletter Sign-up).
- Content:**
 - Confirm Contract Generation:** Please review the contract details before proceeding. Once generated, the smart contract will be deployed to the blockchain.
 - Contract Parties:** Merchant: Not selected, User: John Doe.
 - Service Details:** Category: Education, Quantity: 1 months, Price: \$99.99 per months.
 - Payment Schedule:** Model: Pay After Use, Frequency: Monthly, First Payment: May 1, 2025.
 - Contract Duration:** Start Date: May 1, 2025, End Date: April 30, 2026, Renewal: Notify before renewal.
 - Important Information:** By generating this contract, you agree to the terms and conditions specified. The contract will be deployed to the blockchain and cannot be altered once finalized.
- Buttons:** Cancel, Confirm & Generate.

Fig 3.4 The Interface for contract creation & confirmation.

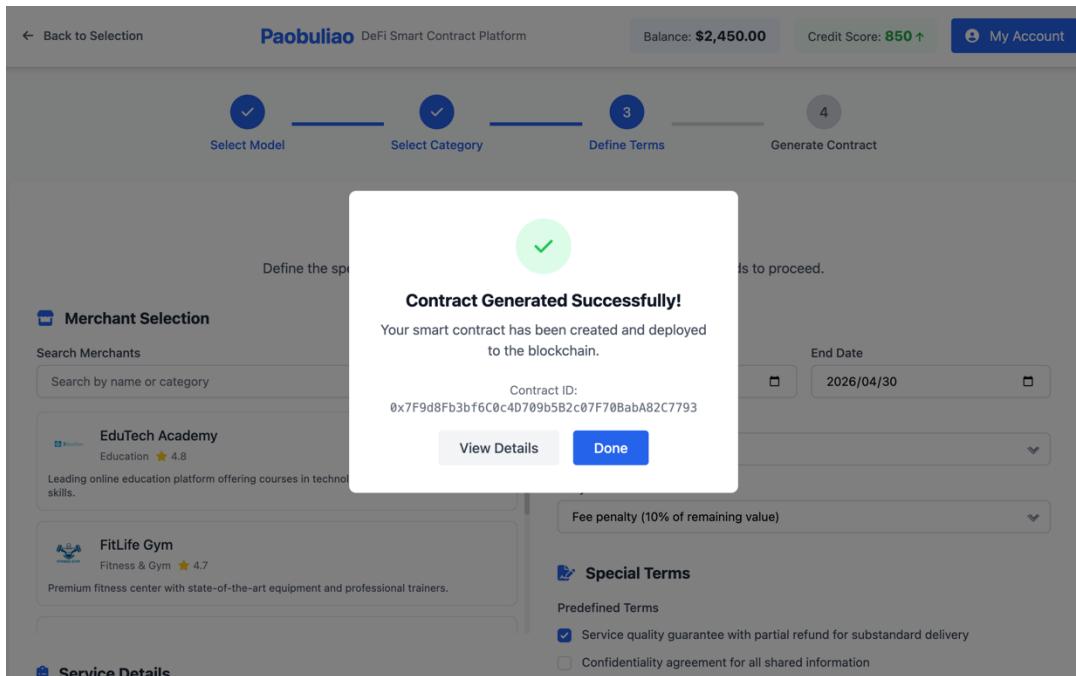


Fig 3.5 The Interface for contract success generation.

The screenshot shows the 'Active Contracts' section of the platform. It displays two active contracts: 'EduTech Academy' (Online Course Subscription) and 'FitLife Gym' (Annual Membership). Both contracts show payment progress bars and details like total amount, next payment date, and contract end date. Below this, the 'Recent Transactions' section lists five completed payments from different merchants: EduTech Academy, FitLife Gym, SparkleWash, and another EduTech Academy entry. The table includes columns for Merchant, Date, Amount, Status, and Type.

MERCHANT	DATE	AMOUNT	STATUS	TYPE
EduTech Academy Online Course	Apr 20, 2025	\$120.00	Completed	Automatic Payment
FitLife Gym Membership	Apr 15, 2025	\$70.00	Completed	Automatic Payment
SparkleWash Car Wash	Apr 10, 2025	\$35.00	Completed	One-time Payment
EduTech Academy	Apr 07, 2025	\$120.00	Pending	One-time Payment

Fig 3.6 The Interface for subpage-daily consumption.

Contract Termination

Contract Status: Termination Requested

You have requested to terminate your contract with EduTech Academy

Termination Details

Merchant	Service Type
EduTech Academy	Online Course Subscription
Termination Requested	Effective Date
April 22, 2025	May 1, 2025

Financial Settlement

Total Contract Value	Amount Paid
\$1,200.00	\$800.00
Refund Amount	Final Payment Due
\$200.00	\$0.00

Confirm Termination **Cancel Request**

Termination Timeline

- 🕒 **Termination Request Submitted**
 April 22, 2025
 You submitted a request to terminate your contract with EduTech Academy.
- 🔔 **Merchant Notified**
 April 22, 2025
 EduTech Academy has been notified of your termination request.
- 📅 **Refund Calculation**
 April 23, 2025
 Smart contract calculated the refund amount based on usage and terms.
- ⌚ **Pending Confirmation**
 Current Stage
 Waiting for your confirmation to process the termination and refund.
- ✅ **Contract Termination**
 Pending
 Once confirmed, the contract will be terminated and refund processed.

ⓘ **Need Assistance?**
 If you have questions about your contract termination, please contact our support team at support@paobuliao.com or call 1-800-123-4567.

Contact Support

Fig 3.7 The Interface for subpage-contract termination.

Transaction Details < >

Overview Advanced TxInfo State

TRANSACTION ACTION
Call | 0xb3ba4a2 | Method by 0x5C9feA9E...2cc59C589 on 0x5876144a...1Cf1AE575

① Transaction Hash: 0x3e96547c631dcb31fd225423257458a554f41802f5870c24e3d7c1a9510947a4 ⓘ
② Status: Success
③ Block: 330076300 | 71 L1 Block Confirmations
④ Timestamp: 14 mins ago (Apr-25-2025 01:16:44 PM +UTC)
⑤ From: 0x5C9feA9E6a590EeB3CD158181F22ffC2cc59C589 ⓘ
⑥ To: 0x5876144a769C19463B65916cF05209c1Cf1AE575 ⓘ ✓
⑦ Value: ₿ 0 ETH (\$0.00)
⑧ Transaction Fee: 0.00000500691 ETH (\$0.008854)
⑨ Gas Price Bid: 0.01 Gwei (0.0000000001 ETH)
⑩ Gas Price Paid: 0.01 Gwei (0.0000000001 ETH)

More Details: + Click to show more

⑪ Private Note: To access the Private Note feature, you must be Logged In

ⓘ A transaction is a cryptographically signed instruction that changes the blockchain state. Block explorers track the details of all transactions in the network. Learn more about transactions in our Knowledge Base.

The screenshot shows a detailed view of a blockchain transaction. At the top, there's a navigation bar with tabs for Overview, Advanced TxInfo, and State. Below that, a section titled 'TRANSACTION ACTION' shows a call to address 0xb3ba4a2 using method 0x5C9feA9E...2cc59C589 on contract 0x5876144a...1Cf1AE575. The transaction hash is 0x3e96547c631dcb31fd225423257458a554f41802f5870c24e3d7c1a9510947a4. The status is 'Success'. It occurred in block 330076300, which has 71 L1 Block Confirmations. The timestamp is 14 mins ago (Apr-25-2025 01:16:44 PM +UTC). The transaction originated from address 0x5C9feA9E6a590EeB3CD158181F22ffC2cc59C589 and was sent to address 0x5876144a769C19463B65916cF05209c1Cf1AE575. The value transferred was 0 ETH (\$0.00), and the transaction fee was 0.00000500691 ETH (\$0.008854). The gas price bid and paid were both 0.01 Gwei (0.0000000001 ETH). There is also a note about the Private Note feature, which requires logging in.

Fig 3.8 The transaction details of a contract.