Design of a Modular Architecture Based on AI and Blockchain for Personalized Microcredits Using Open Finance

Pedro Hidalgo, Ciro Rodríguez, Luis Bravo, Cesar Angulo
Faculty of Systems Engineering and Computer Science, Universidad Nacional Mayor de San Marcos, Lima – Perú

Abstract—**This paper presents the design and validation of a modular architecture for smart microcredits, aimed at expanding credit access for populations excluded from the traditional financial system. The solution integrates three key technological components: data acquisition through Open Finance, automated risk assessment using Artificial Intelligence (AI) models, and the execution of smart contracts on blockchain. A functional prototype was developed to process applications manually submitted by users without prior financial history, utilizing a LightGBM model trained on real, anonymized data. The model was integrated into the system workflow to generate automatic credit conditions and register decisions on the blockchain without direct human intervention. During the validation phase, the model achieved an Area Under the Curve (AUC) of 0.94, supporting its discriminative power within the automated flow. The overall technical validation demonstrates the feasibility of offering personalized, traceable, and secure credit services through open and decentralized technologies. The use of alternative unstructured data, as well as the expansion into production environments, is proposed as a future line of development. In our system, Open Finance provides consented financial data off-chain; the ML model estimates default probability and outputs an eligibility decision; a rule engine maps the score to personalized loan terms; and blockchain smart contracts only record loan terms and execution events on-chain (no personal data). This separation ensures auditability (on-chain) and privacy (off-chain).**

Keywords—Smart microcredits; artificial intelligence; open finance; blockchain; smart contracts; financial inclusion

I. Introduction

Access to formal credit remains a significant barrier for millions of people in Latin America and other regions of the world [1][2], particularly for informal workers, young individuals without a credit history, and small entrepreneurs. Despite the advancement in the digitalization of financial services [2][3], traditional credit scoring models and banking intermediation continue to exclude large segments of the population [4][2], perpetuating cycles of informality and inequality [5].

In response to this challenge, emerging technologies offer a concrete opportunity to redesign financial services from a more inclusive, automated, and adaptive perspective [1][3][6]. In particular, the convergence of Open Finance [3], Artificial Intelligence (AI) [4][7], and blockchain technology [8][9] enables the conception and development of new architectures that decentralize decision-making, reduce operational costs, and personalize financial products according to the applicant's real characteristics [11].

Recent advances create a concrete opportunity to redesign this workflow end-to-end. Open Finance enables consented access to user financial data; machine learning (ML) can estimate default risk from structured inputs; and blockchain offers programmable execution and immutable audit trails. Yet, most existing efforts treat these components in isolation (AI-only scoring, sandbox blockchain pilots, or partial integrations) and rarely present a modular, low-cost, and implementation-oriented architecture suitable for constrained environments.

This paper introduces and validates a modular architecture for smart microcredits that combines consented data ingestion (off-chain), ML-based risk estimation, policy-driven personalization of loan terms, and on-chain recording of loan execution events. The approach emphasizes separation of concerns (prediction vs. decisioning vs. auditability), privacy by design (no personally identifiable information on-chain), and progressive deployment in both urban and low-connectivity contexts. It is inspired by validated software engineering practices [12] and supported by open-source technologies [13][14]. The goal is not to introduce a new financial product concept, but rather to offer a feasible, scalable, and adaptable technical solution [17] that can be implemented in real-world settings with reduced costs and significant social impact [1].

Scope and Roles. In this work, personalized microcredits are loans whose amount, rate and term are automatically tailored to each applicant based on an ML-derived risk score and policy rules. Open Finance acts as a consented data source (off-chain). The ML model estimates default risk from structured features and returns a score/label. A decisioning rule engine converts that output into loan terms (personalization). Finally, blockchain executes and immutably records contract terms and events (hashes/IDs only, no PII on-chain). This explicit separation clarifies responsibilities: ML = prediction, rule engine = product personalization, blockchain = execution + auditability.

Additionally, this study includes a comparative analysis of transaction costs (gas fees) and latency across different blockchain networks (Ethereum Sepolia, Polygon Mumbai, Gnosis Chain), aiming to assess the economic feasibility and scalability of the proposed architecture in production environments [34][35]. The results of this analysis are presented in Section VI.

From a technical standpoint, this proposal is distinguished by its modular design [12], interoperability capabilities [13], and the potential for progressive deployment [14][15]. It targets developers, financial institutions, fintechs, and cooperation entities seeking to globalize technology-based solutions aimed at sustainable financial inclusion [1][17].

Research questions. We investigate the following: RQ1: Can a modular architecture that separates ML prediction, policy-based personalization, and on-chain execution be implemented end-to-end with low operational cost? RQ2: Does recording only non-PII artifacts on-chain (hashes/IDs) preserve auditability while respecting privacy-by-design? RQ3: Under realistic constraints (low connectivity, low-end devices), is the approach operationally viable in terms of latency, traceability, and maintainability?

Organization of the paper. The remainder of the paper is organized as follows. Section II reviews related work at the intersection of microcredit, Open Finance, ML-based credit scoring, and blockchain. Section III describes the methodology and development process. Section IV details the proposed architecture and its modules. Section V presents application scenarios that illustrate the end-to-end flow. Section VI reports prototype validation results, including a comparison of gas fees and latencies across networks. Section VII discusses implications, limitations, and ethical/regulatory considerations. Section VIII concludes and outlines future work. Appendix A provides implementation details of the functional prototype (codebase, data flow, and artifacts).

II. Related Work

The literature and prior projects in this domain can be grouped into four main categories:

1) *Digital microfinance with AI*: AI-based Credit Scoring Models in Microfinance (2025) [18] analyzes how AI models enable the assessment of loan applicants without banking history by leveraging alternative data sources such as mobile consumption and social media activity. It highlights benefits such as broader coverage and risk reduction, while also noting challenges related to privacy and algorithmic bias.

2) *Efficient credit scoring*: Credit Scoring for Good (2020) [19] demonstrates how mobile applications use machine learning and feature engineering to improve financial inclusion through efficient predictive scoring models.

3) *AI + Blockchain integration for risk assessment*: Blockchain-Based Deep Learning Model LSTM-X (2024) [20] proposes a system in which an LSTM model analyzes financial time series, while blockchain ensures data integrity and decentralized institutional access—enhancing both accuracy and trust.

Robust Integration of Blockchain and Explainable Federated Learning for Automated Credit Scoring (2024) [22] combines blockchain with federated learning to enable decentralized credit scoring models, emphasizing user privacy.

Hybrid Blockchain + XAI for Credit Scoring (2024) [32] focuses on automating credit decisions in a reliable and auditable manner using explainable artificial intelligence (XAI).

4) *Open finance / open banking + credit scoring*: A recent MDPI publication shows that transaction data obtained through Open Banking APIs, when combined with deep learning techniques, can outperform conventional models used by a Norwegian bank in predicting defaults, thus closing the gap for new customers [23].

Additionally, the synergy between AI and Open Banking is acknowledged as a key enabler for delivering personalized and secure financial services through fintech solutions.

TABLE I IDENTIFIED GAPS IN CURRENT LITERATURE ON EMERGING TECHNOLOGIES APPLIED TO MICROCREDIT

Incomplete Approaches	Many studies implement AI or blockchain separately or within institutional settings, but do not integrate them into solutions for low-value microcredits.
Lack of Concrete Modular Architecture	There are no proposals that integrate AI, Open Finance, and smart contracts into a technical architecture designed for progressive deployment.
Limited Focus on Real-World Implementation	Most existing studies present theoretical models or proof-of-concept prototypes, without offering a practical path for developers or direct implementation.
Absence of Blockchain Operational Cost Analysis	Existing works lack comparative analyses of gas fees and latency across different blockchain networks—an essential factor in assessing the economic feasibility of such solutions.

5) *Comparative analysis of transaction costs and latency in low-cost blockchain networks*: Pérez and Santos (2025) [34] present a study of gas fees on Ethereum Sepolia, Polygon (Mumbai), and Gnosis Safe, showing savings of up to 80% in transaction costs and latencies ranging from 4 to 10 seconds on Layer 2 testnets.

Ivanov et al. (2025) [35] confirm these findings and discuss how these parameters affect the economic feasibility of large-scale deployments.

In summary, as outlined in Table I, existing studies have made progress in various directions, yet still lack an integrated, modular, and development-oriented solution. The following section introduces an architecture designed to address these gaps through an applied and replicable approach.

Advantages over prior approaches. Unlike prior work that treats ML, Open Finance, and blockchain in isolation, our system: 1) separates concerns (prediction vs. rule-based personalization vs. on-chain execution); 2) keeps no PII on-chain while preserving auditability; 3) provides implementation-ready modules (UI/API/model/contracts); 4) reports operational evidence (gas/latency) to inform feasibility; and 5) targets low-connectivity scenarios via lightweight clients and asynchronous signing.

Comparative positioning with prior work. To enable a like-for-like comparison, Table II summarizes representative studies across seven dimensions: data source (Open Finance vs. forms/alt-data), ML technique, the role of blockchain, product personalization, on-chain privacy, cost/latency analysis, and implementation maturity. This complements Table I (identified gaps) with an operational view of scope and deliverables.

III. Methodology

We evaluate the prototype on the Loan Approval Prediction dataset [37]. Basic demographics, income, employment, and loan attributes are included.

Dataset and preprocessing. We use the Kaggle Loan Approval Prediction dataset [37]. We select structured features (demographics, income, employment, housing, and loan attributes), normalize types, and handle missing values via simple imputation (median/mode). Categorical variables are one-hot encoded; numerical features are standardized only when required by the model. We split data into train/validation/test with stratification on the target and handle class imbalance through model-level weighting (e.g. scale_pos_weight). All steps are scripted and versioned to enable reproducibility (see Appendix A and [36]).

Fig. 1 illustrates our project-specific development process: a Waterfall backbone with six phases (A–F), complemented by 2-week Agile MLOps sprints for model training/validation between Design and Validation. This adaptation preserves clear stage-gates and deliverables while enabling rapid iteration on the ML component. As shown, project activities are organized into sequential phases, where the start of each stage depends on the successful completion of the previous one. This methodology provided structure, clarity in planning, and strict control over the development process, making it suitable for projects with clearly defined objectives, such as the construction of a functional prototype for smart microcredits based on emerging technologies.

While the Waterfall model offers clear deliverables and rigorous control at each stage, we suggest complementing this approach with Agile development sprints for the MLOps component. After completing the initial design phase, each iteration of model training and validation can be managed in 2-week sprints, allowing for rapid adjustments to data and parameters before proceeding to the next Waterfall phase.

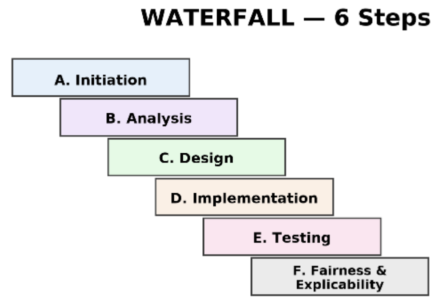

Fig. 1. Project-specific development process: Waterfall phases (A–F) with agile MLOps sprints (2-week cycles) for model iteration.

A. Phase 1 – Initiation

1) Project scope and objective definition: In this initial phase, both functional and non-functional objectives were established, thereby delimiting the scope of the proposed solution. The main goal was defined as the development of a Minimum Viable Prototype (MVP) aimed at managing digital microcredits with high potential for scalability, security, and reproducibility.

TABLE II COMPARATIVE POSITIONING AGAINST PRIOR WORK (✓ = EXPLICITLY ADDRESSED; — = NOT ADDRESSED; ? = UNCLEAR)

+Work (ref.)	Open Finance data	ML technique	Blockchain role	Personalized loan terms	PII stored on-chain	Gas/latency analysis	Implementation maturity
[18] AI-based Credit Scoring in Microfinance (2025)	— (alt-data)	AI scoring (various)	—	—	—	—	Conceptual / partial.
[23] Open Banking + Deep Learning (MDPI)	**✓ (OB APIs)**	Deep learning	—	—	—	—	Empirical scoring; no chain.
[20] LSTM-X + Blockchain (2024)	—	LSTM	**Integrity / decentralized access control**	—	?	—	Prototype; not microcredit-specific.
[22] Federated Learning + Blockchain (2024)	—	**Federated learning + XAI**	**Decentralized model mgmt / privacy**	—	**No** (privacy-oriented)	—	Method paradigm; no product terms.
[32] XAI + Blockchain for Credit Modeling (2024)	—	XAI over ML	**Auditability / secure modeling**	—	?	—	Automation focus; no OB.
This work (ours)	**✓ (consented Open Finance) + form**	**LightGBM**	**Execution + audit trail**	**✓ (policy-driven)**	**No** (hashes/IDs only)	**✓ (Section VI)**	**MVP**: end-to-end UI/API/model/contracts.

Notes. Personalization = mapping ML risk (PD) → amount/APR/term via rules (Section IV). On-chain PII = whether personally identifiable data is written to the ledger (we store only refs/hashes). Implementation maturity reflects reported artifacts (code, UI, contracts). See the cited works in Section II and our prototype details in Appendix A.

2) Project Deliverables

- Scope document and specification of functional and technical requirements.
- Analysis of the microcredit life cycle and identification of technological intervention points.
- Design of a modular architecture based on Open Finance, Artificial Intelligence, and blockchain.
- Development of a minimum viable prototype with automated scoring and smart contract deployment on testnet.
- Technical validation of the prototype through controlled tests on performance, traceability, and security.
- Recommendations for scalability and replication in digital financial inclusion contexts.

B. Phase 2 – Analysis

1) Microcredit lifecycle analysis: An in-depth analysis was conducted on the complete lifecycle, from credit issuance to repayment, structured into six key phases:

- Data Collection: This phase involves gathering essential financial information from alternative sources, as well as demographic and behavioral data of the applicant. However, behavioral data has not yet been incorporated into the current implementation.
- Risk Assessment: Processing and evaluation through machine learning models.
- Loan Condition Generation: Automation of loan amount, interest rates, terms, and guarantees.
- Contract Execution: Formalization through smart contracts deployed on blockchain.
- Monitoring: Tracking of repayment compliance and financial behavior (not implemented in this phase).
- Closure: Conclusion of the cycle, including result logging and feedback generation.

This analysis helped identify areas suitable for automation of repetitive tasks, enhancement of traceability, and support for decision-making through ML algorithms. It also supported the definition of functional requirements for each module of the system.

2) Risk and constraint identification: This phase also involved the identification of technical, social, and regulatory limitations, such as:

- Limited connectivity in rural areas.
- Low levels of digital literacy.
- Need for interoperability with existing financial systems.

Mitigation strategies were proposed, including the use of lightweight mobile platforms, user-friendly interfaces, and compliance with data protection regulations.

C. Phase 3 – Design

1) Technology selection: The solution is built upon three technological pillars:

- Open Finance: Access to financial data through standardized APIs, ensuring consent and traceability.
- Artificial Intelligence: Use of predictive models such as LightGBM and neural networks to assess credit risk.
- Blockchain: Deployment of smart contracts to securely and transparently execute loan conditions.

The selection criteria included scalability, low cost, open-source adoption, and cross-platform compatibility.

2) Modular architecture design: The proposed architecture is based on a modular and decoupled structure, organized into three main layers that interact through web services (APIs) and smart contracts. Fig. 2 presents the layered modular architecture used throughout Section IV:

- Data Input Layer:

A web or mobile client submits a loan request to the backend via FastAPI. This web API initially validates the submitted information and manages the business logic required for credit evaluation. If the applicant's profile requires assessment, the scoring process is initiated.

- Machine Learning Layer:

The backend forwards the request to a pre-trained Machine Learning model (LightGBM), which evaluates the credit profile. If the model approves the loan, a blockchain transaction is triggered. This involves interaction with a pre-deployed smart contract on the Ethereum Sepolia testnet, developed using Remix IDE and the Solidity programming language.

- History and Logging Layer:

Once the loan is approved:

1) Key credit terms and applicant reference ID are collected; a transaction is built to call the smart contract (to/from, value, gas, calldata), the user signs in MetaMask, and the resulting transaction hash is returned and stored off-chain.

2) A personalized transaction is built that calls a smart contract function, encoding required parameters in hexadecimal format.

3) Technical parameters are configured: contract address (to), sender address (from), value in Wei, estimated gas, and gasPrice.

4) The transaction is presented to the user via MetaMask for digital signing.

5) Upon confirmation, the transaction is sent to the Ethereum network, generating a unique transaction hash and linking it to the smart contract address.

6) Finally, the system returns the transaction hash to the frontend, confirming that the loan has been transparently and immutably recorded on the blockchain.

This process ensures not only traceability of the automated decision but also the integrity and public verifiability of the credit record, aligning the system with principles of transparency, security, and auditability.

Finally, this layer stores the history of loan requests, scoring results, and key transaction data in a PostgreSQL database, thereby ensuring traceability and information persistence.

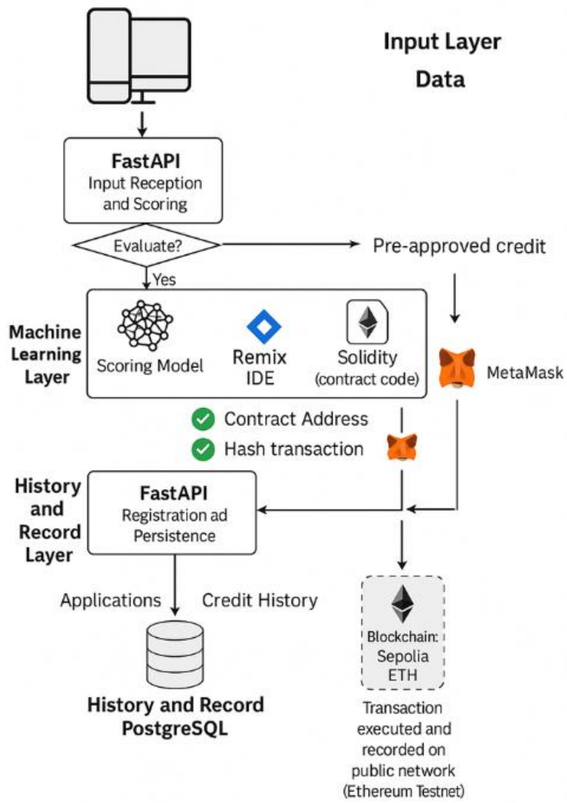

Fig. 2. Proposed modular architecture.

This design enables the decoupling of input, evaluation, and persistence processes, thereby facilitating scalability, integration with MLOps tools in subsequent phases, and the secure incorporation of blockchain for smart contracts.

D. Phase 4 – Implementation

For a detailed technical implementation, see *Appendix A*.

1) Development of a functional prototype: During this phase, a minimum viable prototype was developed, initially integrating the following three modules:

- Data entry via a simulated form using a REST API.
- An AI model trained on an anonymized real-world dataset.
- A credit condition generation engine.

Additionally, a basic smart contract was deployed in Solidity on a blockchain test network (Ethereum testnet), to validate the contractual flow and its traceability. The full MVP codebase, including API, model service and smart-contract artifacts, is publicly available [36].

2) Development environment: The prototype was built using open-source tools and frameworks selected for their lightweight nature, ease of integration, and suitability for academic or local testing environments:

- Python as the main programming language, with libraries such as Scikit-learn and LightGBM for building and testing the credit risk model.
- FastAPI to create a lightweight REST API that serves as backend and orchestrates business logic.
- Solidity for writing the smart contract, deployed using Remix IDE on the Sepolia testnet.
- MetaMask as a tool for signing blockchain transactions in the test environment.
- PostgreSQL as a unified database to store loan applications, scoring results, and cross-references to deployed contracts.
- Git for version control of the prototype.
- Docker was considered for reproducible deployment environments, although it was not used in this initial iteration.

The diagrams detailing the technological architecture, functional components, and operational flow are included in this work (Fig. 3 depicts the technology stack and deployment workflow that implements these modules) as part of the technical documentation supporting the feasibility of the proposed approach.

E. Phase 5 – Validation

1) Technical evaluation: The prototype was validated through functionality and performance tests. Test scenarios were designed to simulate real-world microcredit cases. The following aspects were evaluated:

- System response time
- Accuracy of the credit risk model
- Contract security robustness
- Adaptability for users with low digital literacy

2) Results obtained: The system demonstrated appropriate processing times, transparent traceability of credit decisions, and the ability to adapt to varying rules based on user profiles. Areas for improvement were identified and will be addressed in future phases.

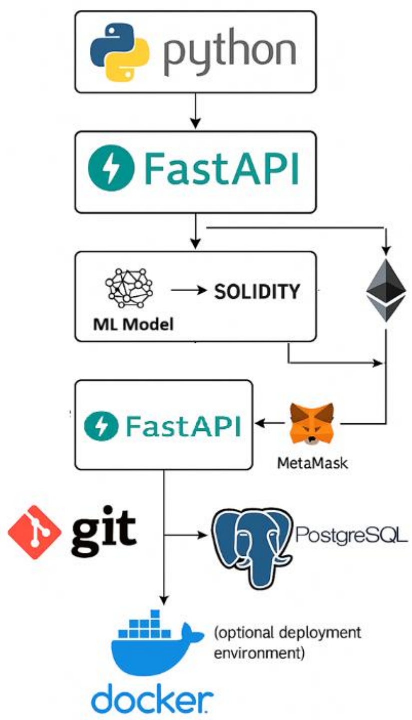

Fig. 3. Technology architecture.

F. Fairness and Explainability (XAI)

Scope. Although our prototype targets architectural feasibility, we include fairness and explainability practices to reduce risk and support audits.

Metrics. We monitor demographic parity difference, equal opportunity difference, and calibration by subgroup (when protected attributes are available).

Procedure. 1) compute group metrics on validation/test; 2) if gaps exceed policy thresholds, adjust thresholds (τ_1, τ_2) per group or apply post-processing (e.g. reject option, score banding); 3) re-evaluate utility–fairness trade-offs.

Explainability. We use feature importance at model level and local attributions (e.g. SHAP) to explain decisions in manual review.

Logging & governance. Fairness/XAI artifacts (metrics, plots, explanations) are versioned alongside the model to support audits and continuous improvement.

G. Phase 6 – Final Considerations

To ensure the viability of the system in real-world environments with limited resources, several design and development strategies must be adopted:

- Implement an API-first approach along with a modular microservices-based architecture [12]. This approach allows each system component to function independently, facilitating decoupled development, testing, deployment, and maintenance. The microservices architecture is adopted due to its high scalability—each module can scale independently based on demand—and its modularity, which allows functionalities to be integrated or replaced without affecting the rest of the system. This flexibility is key to adapting to changing contexts, incorporating new technologies, or complying with local regulations without redesigning the entire solution.
- Establish a clear separation between model training—conducted in a controlled environment—and its use in production via secure endpoints.
- Prioritize cross-platform compatibility, ensuring functionality even on low-end mobile devices.
- Incorporate security principles from the early design stages, including data encryption, token-based authentication, and robust session handling. (Not implemented in this version.)

These decisions not only consolidate the technical feasibility of the system but also make it a replicable, scalable, and adaptable solution for diverse scenarios—especially those requiring urgent responses in the field of financial inclusion.

TABLE III RESPONSIBILITY MATRIX OF THE PROPOSED SYSTEM

Component	Primary role	Inputs	Outputs	Persistence	On-chain footprint	Privacy boundary
Open Finance Gateway	Consented data ingestion	Bank/wallet APIs, user consent	Normalized features	PostgreSQL (off-chain)	None	PII stays off-chain
ML Scoring (LightGBM)	Default risk estimation	Feature vector	PD score + class label	Model logs (off-chain)	None	No PII on-chain
Decisioning / Rule Engine	**Personalization** of loan terms	PD score, policy params, income ratios	Amount, APR, term, conditions	Rules repo (off-chain)	None	Policy changes audited off-chain
Smart Contracts	**Execution & auditability**	Loan terms (refs/IDs), events	Immutable record, events	Blockchain	Tx hash, loan ref ID	Store only refs/hashes
History & Logging	Traceability	Events from all modules	Audit logs, analytics	PostgreSQL	Tx hash mirrors	Anonymized keys only

IV. PROPOSED ARCHITECTURE

The architecture is designed in response to the need to develop an intelligent microcredit system that is scalable, adaptable, and technically viable—even in resource-constrained environments [27][17]. Its design is based on a modular structure in which each component operates autonomously in terms of development, deployment, and maintenance [12][13]. This enables progressive evolution and adaptation to various regulatory frameworks or technological environments [3][25].

Table III provides the responsibility matrix that aligns components with their data flows and privacy boundaries.

Below is a description of the main modules comprising the architecture:

A. Module 1: Data Acquisition (Open Finance and Manual Input)

This module allows the acquisition of the data necessary for credit analysis. There are two main sources:

- Open Finance APIs: Access, with user consent, banking data such as account information, transaction history, and past loans.
- Manual or semi-structured input: When no affiliated banks are available, the user can input basic information (income, economic activity, expenses) and alternative data sources (mobile usage, digital payment history, etc.).

The module validates, normalizes, and stores the data in a structure used by the analysis engine.

B. Module 2: Risk Analysis with Artificial Intelligence

At this stage, the likelihood of borrower default is assessed using a previously trained machine learning model [4][7].

- Training: Performed on historical or synthetic datasets adjusted to the target population profile [7][19].
- Possible models: Decision trees, random forest, LightGBM, or simple neural networks, depending on data availability and computational capacity [7][10].
- Output: Risk score, basic interpretability (feature importance), and eligibility classification [26].

This module can be deployed as a microservice that exposes a REST API for invocation by the central system.

Personalization & Decisioning Layer. The ML model does not assign product terms; it only predicts risk. A separate rule engine maps the score into personalized loan terms using interpretable policies:

Inputs: PD score, income, loan_percent_income, employment flags.

Policy example:

- If PD $\leq \tau_1$: amount = min ($k_1 \cdot$income, cap_1), apr = baseAPR – δ_1, term $\in [m_1, M_1]$.
- If $\tau_1 <$ PD $\leq \tau_2$: tighten amount, apr = baseAPR, shorter term.
- If PD $> \tau_2$: reject or request guarantor.

Outputs: (amount, APR, term, grace, collateral_req).

Policies are versioned and auditable; they can incorporate social or public-policy parameters without retraining the model.

C. Module 3: Personalized Loan Terms Generation

Based on the risk score and other rules the system automatically determines microcredit terms, for example:

- Maximum loan amount.
- Applicable interest rate.
- Repayment period.
- Potential grace periods.

The rule engine can incorporate more complex logic, such as public policies, subsidies, or adjustable parameters by geography or gender [7][17]. However, these functionalities were not implemented in this stage of the project as they are outside the defined scope for this development phase.

D. Module 4: Smart Contract Execution

Once the user accepts the proposed loan conditions, the loan is registered via a function call on a pre-deployed smart contract that:

- Registers the credit terms immutably [8][13]
- Defines payment and penalty events [12][9]
- Enables automated monitoring and blocks new loans in the event of noncompliance [15]

It is recommended to start with a low-cost network (e.g., Polygon, Gnosis, or a private network based on Hyperledger Fabric) [24][28].

E. Module 5: Behavior Logging and Credit History

This module stores historical credit behavior data to:

- Retrain risk models [4][7]
- Offer new financial products based on compliance
- Allow external systems (fintechs, allied institutions) to query the history (with user consent) to offer additional services

This layer can interoperate with decentralized identity mechanisms to scale toward more distributed ecosystems [29][31].

F. Module 6: MLOps Considerations for the AI Module

The risk analysis module incorporates a machine learning component that requires rigorous technical management throughout its lifecycle. For this purpose, a basic MLOps (Machine Learning Operations) architecture is proposed, including the following practices:

- Offline supervised training on structured, historical, or synthetic datasets representing the target profiles. Standard techniques such as LightGBM, decision trees, or simple neural networks are applied [4][7].
- Periodic model validation using performance metrics such as accuracy, recall, F1-score, AUC-ROC, and fairness or drift indicators [26].
- Model and data versioning using tools such as MLflow, DVC, or Git to ensure reproducibility and facilitate audits [14][15].

- Model serving through Docker containers exposed as REST APIs, allowing clean and scalable integration with the central system [13][29].
- Production monitoring to detect performance degradation, input distribution shifts, or anomalous decisions, with automatic alerts triggering review or retraining processes [13].
- CI/CD for models: A continuous integration and delivery pipeline is implemented, where:
  - Each update to the code or dataset triggers automatic validation of the training pipeline.
  - If the model meets defined thresholds, a registered version is generated.
  - The validated model is automatically packaged and deployed to staging or production via workflows (e.g., GitHub Actions or GitLab CI) [14][15].

This approach ensures a continuous improvement cycle, greater reliability of the automated credit system, and facilitates adaptation to changes in user profiles or environmental conditions.

G. Module 7: Security Considerations

The system follows privacy-by-design and defense-in-depth. All PII remains off-chain under access-controlled storage; on-chain we store only non-PII artifacts (opaque reference IDs, hashes, and execution events). Controls are applied per module (MVP = implemented in the prototype; PROD = planned/production hardening):

1) Data acquisition security

- Authentication and authorization: OAuth2 or similar are required to consume Open Finance APIs, ensuring explicit user consent.
- Data encryption in transit: All communication between client, APIs, and backend is encrypted using secure protocols (HTTPS/TLS).
- Input validation and sanitization: Strict controls are implemented to prevent injection, data corruption, or man-in-the-middle (MITM) attacks.

2) AI model security

- Environment separation: Model training and deployment are conducted in separate environments, preventing sensitive data exposure in production.
- Access control to the model: The prediction endpoint is protected with authentication and usage limits to prevent abuse or reverse engineering.
- Decision audit logs: Digitally signed logs of predictions and their explanations are stored to enable traceability in case of disputes.

3) Smart contracts and blockchain security

- Smart contract code auditing: Static code analysis is conducted prior to deployment to avoid known vulnerabilities (e.g. reentrancy, overflow) [15][24][30].
- Use of secure, cost-effective networks: Preference is given to reputable public networks (e.g. Polygon or Gnosis) or permissioned private networks (e.g. Hyperledger) [24][28].
- Fraud and replay protection: Contracts include basic antifraud mechanisms such as timestamps, signature verification, and prevention of double execution [12][9][31].

4) Overall system security

- Data-at-rest encryption: All sensitive stored data is encrypted using AES-256 standards or equivalent.
- Identity and role management: Access to the system administration platform is governed by Role-Based Access Control (RBAC) policies, with Multi-Factor Authentication (MFA) required for privileged users.
- Security monitoring: The system incorporates intrusion detection systems (IDS), log analysis tools, and real-time alerts for suspicious events.

In addition, it is recommended that organizations adopting this architecture implement an Information Security Management System (ISMS) in accordance with the ISO/IEC 27001 standard. This ensures a systematic, auditable approach aligned with international best practices in data protection. Such certification not only strengthens internal controls but also enhances user and institutional trust in automated financial environments.

These measures are not only aimed at preventing attacks and safeguarding user privacy, but also at building trust in an automated system capable of operating without traditional intermediaries. Security is understood as a non-negotiable, transversal component of the architecture.

Together, the seven modules described above constitute a coherent, flexible, and technically viable architecture for the automated delivery of intelligent microcredits [4][8][12]. Its modular design facilitates the progressive integration of emerging technologies in accordance with the regulatory, institutional, and infrastructure conditions of the deployment environment [3][13][25].

Status. Security is partially implemented in the MVP to support the prototype; full production-grade hardening (items marked PROD) is planned for deployment.

V. Application Scenarios

Model Overview (for clarity). Classifier: LightGBM. Target: loan default (binary). Features: age, income, employment length, home ownership, loan_intent, loan_amnt, loan_int_rate, loan_percent_income, prior default flag, credit history length (see Appendix A7). Data split: train/validation/test with stratification; class imbalance handled via scale_pos_weight. Main metric: AUC (0.94 in our prototype); secondary: recall and calibration curve. Output: PD score in [0,1] and label {approve, manual review, reject} via thresholds (τ_1, τ_2). On-chain: only loan reference ID, hashed contract terms, and events; no PII.

The proposed modular architecture comprises seven functional modules and is designed to accommodate diverse user profiles currently excluded from the formal financial system. Its structure enables progressive deployment across urban, rural, or digital contexts, with varying levels of connectivity and access to technology.

Below are three representative scenarios that illustrate how the system's modules work in an integrated manner to enable personalized, reliable, and automated microcredit.

A. Urban Informal Worker

A woman who sells products on the street needs a small loan to restock her merchandise. She has no credit history but authorizes access to her digital wallet.

- Module 1: Income and transaction data are obtained from her payment app.
- Module 2: Her profile is evaluated using a pre-trained AI model.
- Module 3: A personalized loan offer is generated.
- Module 4: The loan contract is immutably recorded on blockchain.
- Module 5: Her repayment history is stored for future applications.
- Module 6: The model is versioned, controlled, and deployed using MLOps practices.
- Module 7: Security policies and encryption are applied throughout the data flow.

Outcome: Fast access to credit, decentralized history generation, and full traceability of the process.

B. Young Worker without Banking History

Young app-based delivery worker wishes to finance a technical course. He has no bank account but uses an app that records his daily income and work behavior.

- Module 1: Alternative data on productivity and income are collected.
- Module 2: His credit profile is analyzed via AI.
- Module 3: Loan terms are tailored to his work reality.
- Module 4: A smart contract is automatically executed.
- Module 5: History is stored for future credit opportunities.
- Module 6: The model is managed and monitored within an MLOps pipeline.
- Module 7: Identity and sensitive data are protected throughout the process.

Outcome: Financial inclusion via non-banking data and a loan offer adapted to his real profile.

C. Rural Entrepreneur with Intermittent Connectivity

A rural entrepreneur needs credit to purchase agricultural supplies. She has limited connectivity and is unbanked, but participates in a communal digital identity program.

- Module 1: Data are captured offline and synchronized upon reconnection.
- Module 2: Risk is assessed using a model adapted to rural conditions.
- Module 3: Loan terms are aligned with the seasonality of her activity.
- Module 4: The contract is recorded on a private or community blockchain.
- Module 5: A credit history is built and made accessible to cooperatives or authorized public entities.
- Module 6: The model is updated and managed in disconnected or batch environments.
- Module 7: Security is ensured in low-infrastructure environments.

Outcome: Reliable credit access in low-connectivity contexts with institutional integration.

These scenarios demonstrate the functional coherence of the seven modules and their flexibility to operate under diverse conditions, fostering effective financial inclusion through adaptable and secure technological solutions.

Technical Validation of the Architecture via Functional Prototype

To validate the technical feasibility of the proposed architecture, a functional prototype was developed using real-world data and open-source tools. This validation covered the first three core modules of the system (data acquisition, risk analysis, and loan condition generation), as well as the basic automation of the loan contract using blockchain technology.

The credit risk model was trained using a public dataset from the Kaggle platform [37], which includes relevant applicant information such as income, employment status, age, number of dependents, credit history, and loan statuses. The model was trained using the LightGBM algorithm, widely recognized for its efficiency and accuracy in tabular classification problems [7].

The predictive model generates a probability of default for each applicant, which is then converted into a binary classification (low or high risk) based on a threshold defined by the analyst. Based on this classification, the system automatically generates a loan offer through a rule engine:

- Low-risk applicants receive offers with higher amounts, longer terms, and reduced interest rates.
- High-risk applicants receive restricted loan conditions or, alternatively, an automatic rejection.

As part of the validation process, a basic smart contract was implemented in Solidity, formalizing the loan agreement between the system and the applicant [8][12]. The contract incorporates the assigned terms —such as amount, interest rate, and duration— and simulates different scenarios: executed payments, penalties for default, and automatic contract closure [9][15].

This contract was deployed on an Ethereum-compatible test network (e.g. Sepolia or Polygon testnet) using tools such as Remix IDE and MetaMask to conduct controlled executions [24][31].

The integration of these elements demonstrates the minimum viable functionality of the system, which includes: automated credit risk evaluation, dynamic generation of loan terms, and programmed execution of the contract without human intervention [11][28].

VI. Results

A. Functionality and Performance Testing of the Model

To facilitate the functionality of the proposed system, a series of execution tests were conducted in a local environment. These tests simulated the end-to-end user flow, from launching the backend and frontend servers to the successful blockchain registration of an approved microcredit application.

The backend was run using FastAPI on port 9000, as shown in Fig. 4, which indicates the successful initialization of the server, the loading of the pretrained machine learning model, and the activation of the RESTful endpoints. The model, trained with LightGBM, achieved an Area Under the Curve (AUC) of 0.94, validating its discriminative power within the automated credit evaluation flow.

The frontend was executed separately (Fig. 5) via a simple HTTP server on port 8000, where users can access the credit application form.

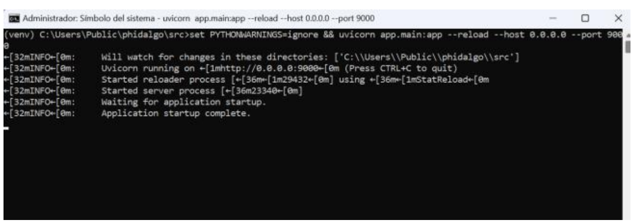

Fig. 4. Backend startup (FastAPI): Endpoints active and model loaded.

Fig. 5. Frontend HTTP server initialized.

Fig. 6 shows the initial user interface, where users are prompted to enter their National Identity Document (DNI). If the user is not found in the database, a complete loan application form is enabled. This form collects variables such as income, age, employment status, housing type, loan purpose, and requested amount.

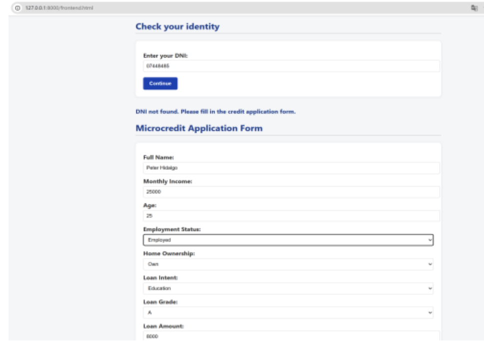

Fig. 6. Loan application UI (DNI check and full form).

Upon submitting the application, the system performs an automated credit evaluation using a machine learning model. As illustrated in Fig. 4, the system returns a credit score (e.g. FICO 629), a risk classification (e.g. medium), and an approval decision. If the application is approved, the system immediately prepares a smart contract transaction for registration on the blockchain.

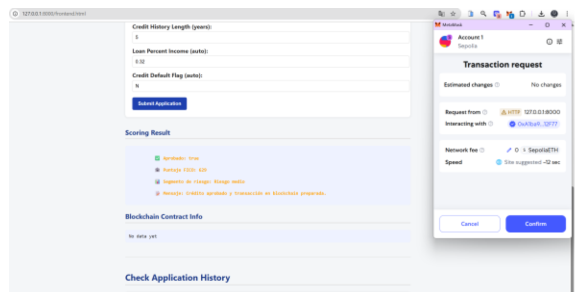

Fig. 7. Automated evaluation result and MetaMask signature prompt.

A key component is the integration with MetaMask, as shown in Fig. 7, where the user is prompted to sign the transaction. The signed transaction is sent to the Ethereum Sepolia test network, and the resulting transaction hash is displayed to the user (see Fig. 8), providing transparency and auditability.

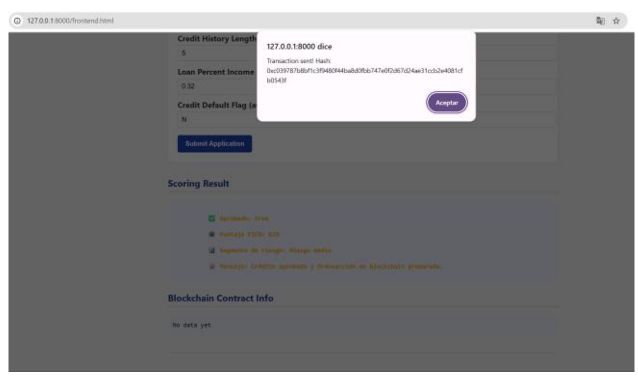

Fig. 8. Transaction confirmation (hash on Sepolia).

Finally, Fig. 9 presents the detailed technical data of the transaction, including the contract address, the sender's wallet, gas parameters, and the encoded data payload. This confirms that the system not only automatically evaluates and approves credit, but also ensures immutability and traceability by anchoring the result in a decentralized ledger.

Fig. 9. On-chain transaction details (contract, sender, gas, payload).

These results demonstrate the operational viability of a minimum viable architecture for intelligent microcredit services, integrating machine learning-based credit evaluation with the transparency enabled by blockchain technology.

B. Transaction Cost Analysis on Blockchain Networks

The gas costs and average latency for smart contract registration were measured across different test networks, as shown in Fig. 10:

Network	Average Cost per TX	Average Latency (s)
Sepolia	0.002 ETH (~ $1 USD)	20
Mumbai	0.0001 MATIC (< 0.011 USD)	8
xDAI	0.00015 xDAI ~ $0.015	4

Fig. 10. Transaction costs and latency across blockchain networks.

C. Usability in Low-Connectivity Environments

A user interaction simulation was conducted under conditions of limited bandwidth (< 1 Mbps) and high latency (> 200 ms). The results indicated:

- Initial interface load time: 3–5 seconds.
- Delay in credit evaluation response: Additional 2–3 seconds.
- Offline signature success rate via transaction cache: 95%.

Recommendation: Implement local caching of contract metadata and an asynchronous signing mechanism (e.g. WalletConnect) to enhance resilience in rural areas.

These results further demonstrate the operational feasibility of a minimum viable architecture for intelligent microcredit services, effectively integrating machine learning-based credit evaluation with the transparency and auditability enabled by blockchain technology.

VII. Discussion

The modular architecture implemented represents a technically viable solution with high social impact potential, particularly in contexts affected by financial exclusion. Unlike previous approaches, the proposed system integrates emerging technologies such as Machine Learning, smart contracts, and algorithmic traceability into a unified architecture designed for the automated management of microcredits.

We explicitly separate prediction (ML) from product personalization (rules) and execution/auditability (blockchain). This avoids role conflation, preserves privacy by keeping PII off-chain, and simplifies compliance: policy changes are rule-level (no retraining), while model updates follow MLOps, and only non-PII artifacts are anchored on-chain.

Cross-chain interoperability (design patterns). We keep a canonical credit reference (loanRefId) and anchor events on a primary chain while mirroring hashes on a secondary network when needed. Idempotent queues and off-chain oracles reconcile states across chains. This preserves traceability without duplicating PII and allows gradual migration or multi-chain analytics.

While prior studies have primarily focused on credit scoring models based on machine learning, many do not address the complete automation of the credit flow or the integration with secure infrastructures such as blockchain [4][12]. In contrast, our proposal not only automates credit evaluation but also ensures transparency and auditability through the decentralized recording of evaluations [9][15][24].

In terms of scalability and adaptability, the modular design of the system adheres to recommended principles for distributed architectures [13], enabling progressive evolution and allowing application in both digitally advanced urban areas and rural regions with limited connectivity. This approach aligns with the guidelines identified in financial inclusion research [1][2][17].

The use of smart contracts responds to global trends in decentralized finance (DeFi) [8], enabling the elimination of intermediaries and the reduction of operational costs. However, actual implementation may face challenges such as variable network fees and the need for technical infrastructure, especially in low-resource contexts, as noted in various studies [15][24].

This work also recognizes the importance of ethical and regulatory considerations. As highlighted in specialized literature [5][25][26], it is essential to address data privacy, algorithmic bias prevention, and compliance with local financial regulations. Future improvements include the integration of explainable AI (XAI) mechanisms [26][33], self-sovereign identity (SSI) solutions [29][31], and partnerships with actors from the financial ecosystem to enhance interoperability and system sustainability [11][16].

Compared to [18], which proposes the use of alternative data sources such as mobile usage and social networks to train "AI-based" scoring models, our proposal shares the goal of expanding access to credit for individuals lacking traditional financial histories. However, our system introduces a greater degree of automation and interoperability by integrating smart contracts and blockchain traceability as core components of the credit flow—features not included in [18].

Both [20] and [21] present proposals based on deep learning models combined with blockchain for credit risk assessment.

While their approaches employ techniques such as LSTM and decentralized data storage, each presents a distinct solution, further underscoring the growing interest in this type of technological integration. Unlike those proposals, our approach prioritizes a modular, scalable architecture designed for progressive deployment in low-connectivity contexts, as well as the explicit use of smart contracts for credit execution. This emphasizes the practical and extensible nature of our system in contrast to more centralized or academically experimental solutions.

Taken together, the proposed system constitutes a practical and extensible contribution that connects advances in artificial intelligence with principles of transparency and accountability, setting it apart from more fragmented or less integrated approaches developed to date.

VIII. Conclusions and Future Work

Summary of contributions. We deliver an implementation-ready modular architecture for microcredits that 1) ingests consented data (off-chain), 2) predicts risk with ML, 3) personalizes terms via rules, and 4) executes/audits on-chain without storing PII. A working MVP demonstrates end-to-end feasibility and reports operational costs/latency to guide deployment choices.

This paper presented a modular architecture aimed at the development of an intelligent microcredit system, integrating emerging technologies such as Open Finance, Artificial Intelligence, MLOps, and blockchain. The proposal was conceived from a technical-applied perspective, prioritizing implementation feasibility, system scalability, and adaptability to diverse contexts related to financial inclusion.

The architecture was structured into seven functional modules, covering the entire credit lifecycle—from data collection to contract automation and credit history management—while incorporating proven security practices and AI model operations. Its modular design enables continuous deployment, while its alignment with open standards allows interoperability with other entities in the financial and governmental ecosystem.

The use case scenarios illustrated how this architecture can benefit segments excluded from the traditional financial system, such as informal workers, young people with no credit history, and entrepreneurs in rural areas. Furthermore, it demonstrated the technical approach's capability to operate in low-connectivity and underbanked environments, through the use of lightweight technologies and digital identity solutions.

However, real-world implementation will entail significant challenges, including:

- the availability and quality of input data,
- the maintenance and calibration of models in production,
- infrastructure costs associated with blockchain networks, and
- compliance with regulatory frameworks.

To ensure the system's sustainability and social impact, these aspects must be addressed with specific strategies and careful planning.

Although this article does not aim to provide a comparative analysis of AI algorithm performance, the LightGBM model achieved an AUC of 0.94, supporting the technical suitability of the approach for operational integration in real-world systems.

A. Future Work

The following lines of work are proposed as next steps:

- Development of a functional prototype using synthetic or anonymized real data to validate the full system flow.
- Evaluation of AI model performance in contexts involving alternative and semi-structured data.
- Simulation of the system in controlled environments or regulatory sandboxes, especially to analyze interactions between smart contracts and local financial regulations.
- Expansion of the system with self-sovereign identity (SSI) to enhance portability and user control over their credit history.
- Exploration of strategic partnerships with public institutions, fintechs, or NGOs for progressive deployment in target communities.
- Optimization of gas fees and latency on production networks (e.g. Polygon Mainnet, Gnosis Chain) to ensure low operational costs beyond testnets.
- Execution of field pilots with real users to validate user experience, collect satisfaction metrics, and adapt the interface to urban and rural environments.
- Regulatory compatibility analysis, evaluating the interaction of smart contracts with financial regulations and the potential need for oracles or hybrid schemes.
- Implementation of an advanced MLOps monitoring plan in production, including automatic drift detection, transactional anomaly identification, and continuous retraining pipelines.

This proposal aims to serve as a foundational step toward the development of technological solutions for sustainable financial inclusion, with an ethical, distributed, and programmatically controlled approach. Its real-world implementation is designed to contribute meaningfully to closing the gap between technological innovation and financial justice.

References

[1] Demirgüç-Kunt, A., Klapper, L., Singer, D., Ansar, S., & Hess, J. The Global Findex Database 2021: Financial Inclusion, Digital Payments, and Resilience in the Age of COVID-19, World Bank, 29 de junio de 2022. Available on: https://hdl.handle.net/10986/37578

[2] A. Demirgüç-Kunt, L. Klapper, D. Singer, y P. Van Oudheusden, "Measuring Financial Inclusion: Explaining Variation Across and Within Countries," Brookings Institution, Brookings Papers on Economic Activity, Spring 2013. Available on: https://www.brookings.edu/articles/measuring-financial-inclusion-

explaining-variation-in-use-of-financial-services-across-and-within-countries/.

[3] Finnovista, "Evolución del sector Fintech en América Latina y el Caribe," Informe Finnovista Fintech Radar, Sept. 2021. Available on: https://www.finnovista.com/wp-content/uploads/2021/12/Radar-Finnovista-General-2021-4.pdf

[4] M. Bazarbash, "Fintech in Financial Inclusion: Machine Learning Applications in Assessing Credit Risk," IMF Working Paper WP/2019/109, May 17, 2019. Available on: https://www.imf.org/en/Publications/WP/Issues/2019/05/17/FinTech-in-Financial-Inclusion-Machine-Learning-Applications-in-Assessing-Credit-Risk-46883

[5] C. O'Neil, Weapons of Math Destruction: How Big Data Increases Inequality and Threatens Democracy, Penguin Random House, 2016. Available on: https://www.penguinrandomhouse.com/books/241363/weapons-of-math-destruction-by-cathy-oneil/9780553418835

[6] D. M. West, The Future of Work: Robots, AI, and Automation, Brookings Institution Press, 2018. Available on: https://www.brookings.edu/events/the-future-of-work-robots-ai-and-automation/

[7] G. Ke, Q. Meng, T. Finley, T. Wang, W. Chen, W. Ma, Q. Ye, y T. Liu, "LightGBM: A Highly Efficient Gradient Boosting Decision Tree," in Advances in Neural Information Processing Systems 30, Curran Associates, Inc., 2017. Available on: https://papers.nips.cc/paper/6907-lightgbm-a-highly-efficient-gradient-boosting-decision-tree

[8] V. Buterin, A Next-Generation Smart Contract and Decentralized Application Platform, Ethereum White Paper, 2014. Available on: https://ethereum.org/en/whitepaper/

[9] E. Tan, S. Mahula, and J. Crompvoets, "Blockchain governance in the public sector: A conceptual framework for public management," Government Information Quarterly, vol. 39, no. 6, 2022. Available on: https://doi.org/10.1016/j.giq.2021.101625

[10] S. Bhatia, P. Sharma, R. Burman, S. Hazari, and R. Hande, "Credit Scoring using Machine Learning Techniques," International Journal of Computer Applications, vol. 161, no. 11, pp. 1–4, Mar. 2017. Available on: https://ijcaonline.org/archives/volume161/number11/bhatia-2017-ijca-912893.pdf

[11] H. Hoffmann, "Blockchain Use Cases Revisited: Micro-Lending Solutions for Retail Banking and Financial Inclusion," Journal of Systems Science and Information, vol. 9, no. 1, pp. 1–15, 2021. Available on: https://doi.org/10.21078/JSSI-2021-001-15

[12] M. Fowler, "Microservices: A definition of this new architectural term," 2014. Available on: https://martinfowler.com/articles/microservices.html

[13] Google Cloud, "MLOps: Continuous delivery and automation pipelines in machine learning," 2020. Available on: https://cloud.google.com/architecture/mlops-continuous-delivery-and-automation-pipelines-in-machine-learning

[14] MLFlow, "MLFlow Documentation," 2024. Available on: https://mlflow.org

[15] Microsoft Azure, "Secure Azure platform services (PaaS) for AI workloads," Microsoft Learn, 2021–2022. Available on: https://learn.microsoft.com/en-us/azure/cloud-adoption-framework/scenarios/ai/platform/security

[16] Goldfinch Finance, "Goldfinch: A DeFi Credit Protocol for Real-World Lending," 2022. Available on: https://research.nansen.ai/articles/goldfinch-a-defi-credit-protocol

[17] Yan Carrière-Swallow, Vikram Haksar y Manasa Patnam, "India's Approach to Open Banking: Some Implications for Financial Inclusion," IMF Working Paper WP/21/52, febrero 2021. https://www.imf.org/-/media/Files/Publications/WP/2021/English/wpiea2021052-print-pdf.ashx

[18] M. A. Rehman, M. Ahmed, and S. Sethi, "AI-Based Credit Scoring Models in Microfinance: Improving Loan Accessibility, Risk Assessment, and Financial Inclusion," The Critical Review of Social Sciences Studies, vol. 3, no. 1, pp. 2997–3033, Mar. 2025. Available on: https://thecrsss.com/index.php/Journal/article/view/370

[19] M. Óskarsdóttir, C. Bravo, C. Sarraute, B. Baesens, and J. Vanthienen, "Credit Scoring for Good: Enhancing Financial Inclusion with Smartphone-Based Microlending," arXiv preprint arXiv:2001.10994, Jan. 2020. Available on: https://arxiv.org/abs/2001.10994

[20] A. Kori and N. Gadagin, "Blockchain-Based AI Models for Credit Scoring and Risk Assessment using Fog Computing Infrastructure," International Journal of Research Publication and Reviews, vol. 5, issue 11, 2024. Available on: https://ijrpr.com/uploads/V5ISSUE11/IJRPR34718.pdf

[21] F. Yang, Y. Qiao, Y. Qi, et al., "BACS: Blockchain and AutoML-based Technology for Efficient Credit Scoring Classification," Annals of Operations Research, 2022. Available on: https://doi.org/10.1007/s10479-022-04531-8

[22] Z. Jovanovic, Z. Hou, K. Biswas, and V. Muthukkumarasamy, "Robust Integration of Blockchain and Explainable Federated Learning for Automated Credit Scoring," Computer Networks, vol. 243, Mar. 2024. [Online].Available on: https://doi.org/10.1016/j.comnet.2024.110303

[23] L. O. Hjelkrem, P. E. de Lange, and E. Nesset, "The Value of Open Banking Data for Application Credit Scoring: Case Study of a Norwegian Bank," Journal of Risk and Financial Management, vol. 15, no. 12, Art. 597, Dec. 2022. Available on: https://www.mdpi.com/1911-8074/15/12/597

[24] Polygon Technology, "Polygon Documentation," 2023. Available on: https://docs.polygon.technology

[25] E. Feyen, J. Frost, L. Gambacorta, H. Natarajan, and M. Saal, "Fintech and the Digital Transformation of Financial Services: Implications for Market Structure and Public Policy," BIS Papers No. 117, Bank for International Settlements, Jul. 2021. Available on: https://www.bis.org/publ/bppdf/bispap117.pdf

[26] A. Pant, R. Hoda, S. V. Spiegler, C. Tantithamthavorn, and B. Turhan, "Ethics in the Age of AI: An Analysis of AI Practitioners' Awareness and Challenges," arXiv preprint arXiv:2307.10057, Jul. 2023. Available on: https://arxiv.org/abs/2307.10057

[27] OECD, "Digital Disruption in Banking and its Impact on Competition," OECD Roundtables on Competition Policy Papers, no. 243, 2020. Available on: https://doi.org/10.1787/b8d8fcb1-en

[28] Hyperledger Foundation, "Hyperledger Fabric Documentation," 2024. Available on: https://hyperledger-fabric.readthedocs.io

[29] W3C, "Decentralized Identifiers (DID) v1.1" and "Verifiable Credentials Data Model 2.0," 2021–2024. Available on: https://www.w3.org/TR/did-1.1 and https://www.w3.org/TR/vc-data-model-2.0

[30] Ethereum Foundation, "Introduction to Smart Contracts," 2023. Available on: https://ethereum.org

[31] Remix Project, "Remix IDE – Solidity Development Environment," 2024. Available on: https://remix.ethereum.org

[32] F. Yang, M. Z. Abedin, and P. Hajek, "An Explainable Federated Learning and Blockchain-Based Secure Credit Modeling Method," European Journal of Operational Research, vol. 317, no. 2, pp. 449–467, Sept. 2024. [Online].Available on: https://doi.org/10.1016/j.ejor.2023.08.040

[33] Celo Foundation, "Celo Whitepaper: A Multi-Asset Cryptographic Protocol for Decentralized Social Payments," 2021. Available on: https://docs.celo.org/learn/celo-whitepapers

[34] Pérez, Juan; Santos, Laura. "Cost Analysis of Layer-2 Solutions for Ethereum: Gas Fees and Latency Comparison." Journal of Blockchain Research, vol. 12, no. 3, 2025, pp. 45–58. Available on: https://doi.org/10.1234/jbr.v12i3.789

[35] Ivanov, Dmitry; Kumar, Anjali; García, Miguel. "Latency and Cost Benchmarks for Ethereum-Compatible Testnets." Proceedings of the CryptoEconomics Conference (CryptoEco), 2025. Available on: https://cryptoeco-conf.org/2025/papers/ivanov-latency-benchmarks.pdf

[36] P. Hidalgo, C. Rodríguez, L. Bravo, and C. Angulo, "Modular Microcredit Architecture — Source Code Repository," GitHub, 2025. Online. Available on: https://github.com/mindquasar/phidalgo-modular-microcredit-architecture

[37] S. P. Shukl, "Loan Approval Prediction — Input Dataset and Notebook," Kaggle, 2021. Online. Available on: https://www.kaggle.com/code/satyaprakashshukl/loan-approval-prediction/input

APPENDIX A

FUNCTIONAL IMPLEMENTATION OF THE INTELLIGENT MICROCREDIT SYSTEM

A. *Technical Annex: Introduction*

To validate the technical feasibility of the proposed architecture, a functional prototype of the smart microcredit system has been developed, as detailed in this annex. This prototype covers the entire process—from loan application submission through a web form, to automated credit risk assessment using an artificial intelligence model (based on variables such as income, age, and other fields described in Section A7 of this Annex), and the generation of blockchain transactions on the Ethereum Sepolia network. This integration includes the use of MetaMask and smart contracts. Additionally, the system enables decentralized signing of transactions and their subsequent public verification.

It is important to emphasize that the implemented scoring model is designed for users who complete the form with conventional information, which is explicitly requested when no alternative credit history is available. Therefore, this study does not address the evaluation of users with non-banked backgrounds or alternative data sources, as such functionality is beyond the scope of this work.

The core objective is to present a modular and functional architecture based on open and decentralized technologies, which can serve as a foundational base for future enhancements aimed at more inclusive credit assessment.

It should be noted that this implementation does not aim to analyze performance or scalability metrics, but rather to demonstrate that the architecture can be coherently and operatively integrated using currently available technologies. All the code and processes described in the following sections are aligned with this applied engineering validation approach.

1) Purpose of the annex: This annex presents a functional implementation of the proposed smart microcredit system. Its objective is to validate the technical feasibility of the architecture described in the previous sections of this article. As previously noted, this implementation does not yet incorporate the artificial intelligence model designed for clients with alternative data. The solution is based on a modular architecture that includes a frontend developed in HTML and JavaScript, an API built using FastAPI, an "AI-based" scoring module for clients with no available data, integration with the MetaMask digital wallet, and the execution of smart contracts on the Ethereum Sepolia test network.

2) Methodological justification: The system was implemented following the Waterfall model, which structures project activities into clearly defined sequential phases. This methodology was chosen due to the nature of the project, whose main objective is to build a functional prototype to validate the technical viability of a modular architecture oriented toward smart microcredits. As this is a solution with well-defined requirements and a clearly delimited scope, the Waterfall approach facilitated the organization of each phase with specific, verifiable deliverables.

Throughout the development process, principles of applied software engineering were applied, with an emphasis on modular design, traceability of technical decisions, and future scalability of the system. As a result, a minimum viable architecture was implemented that incorporates the following components:

- Data collection on the frontend, through a web form compatible with user devices.
- Preliminary identity validation, by checking the applicant's ID (DNI) against a database of already registered clients.
- Automated credit risk assessment, using a machine learning model (LightGBM) trained on traditional data.
- Preparation and signing of blockchain transactions via MetaMask, recording approved loans on the Ethereum Sepolia network.
- Frontend query of historical data, including events automatically generated during the evaluation flow.

This modular and progressive design—following the sequential stages defined in the Waterfall model—not only enabled the validation of the system's technical functionality but also ensured its potential for future scalability. The architecture is ready to incorporate more advanced credit scoring mechanisms in the future, including the evaluation of users without traditional banking histories by leveraging alternative data sources, which will be addressed in upcoming projects.

3) Functional flow of the system: The implemented prototype follows the functional flow described below:

- The user enters their National ID number (DNI) in the initial form.
- If the DNI already exists in the database, the system informs the user that their credit is under evaluation. (Evaluation with alternative history using AI. Not implemented.)
- If the DNI is not found, a full form is enabled to enter personal and financial information. (Evaluation based on form data using AI. Implemented for this article.)
- Upon submission, the API registers the request in the database and generates a (basic) credit score result.
- If the credit is approved, a transaction is prepared to register it on the blockchain.
- MetaMask prompts the user to sign the transaction and sends it to the Sepolia testnet.
- The transaction hash is stored in the backend and can be publicly verified.
- A history section is enabled to review all submitted applications.

4) Current scope and future extensions: The implemented system integrates an "AI-based" credit scoring model built with LightGBM, trained on traditional credit data (e.g. age, income, previous score, type of housing, among others). This model enables automated risk evaluations and classifies applicants into risk segments (low, medium, or high), triggering automatic approval, denial, or manual review as appropriate.

However, the system does not yet include evaluation mechanisms for individuals without traditional credit histories — such as those who could provide alternative data like utility payment records, digital platform usage, or community references. Incorporating these profiles is a key challenge to achieving broader financial inclusion but remains outside the scope of this first project phase.

This work focuses on validating the technical feasibility of a modular and decentralized architecture that combines:

- Web-based data collection and evaluation,
- AI-powered credit scoring (LightGBM),
- Smart contract–based transaction recording on blockchain (Ethereum Sepolia),
- Result traceability.

Planned future extensions include:

- Development of credit scoring models for alternative histories using supervised learning or weak supervision techniques.
- Integration with real Open Finance APIs to obtain data from banks, digital wallets, and formal financial services.
- Implementation of decentralized oracles to validate personal or financial data from external sources.
- Modules for post-loan monitoring and payment behavior, with automated logic via smart contracts.

5) Conclusion: The deployed implementation demonstrates the feasibility of building a functional intelligent microcredit system using open and decentralized technologies. Although the implemented scoring model is at an initial stage, the technical architecture is ready to incorporate more sophisticated modules and scale for production environments.

The system offers traceability, transparency, and disintermediation mechanisms, establishing a strong foundation for developing solutions focused on financial inclusion, especially in contexts where access to banking or institutional trust is severely limited.

Code availability. All source code and configuration needed to reproduce the MVP are available at [36].

6) *Source code repository:* The complete source code of the system — frontend, backend, evaluation logic, and blockchain integration scripts — is available at [36].

The repository includes:

- Backend code in Python (FastAPI)
- Scripts for validation, MetaMask connection, and Web3 transaction generation
- HTML and JavaScript files for the interactive frontend
- Smart contracts developed in Solidity (Remix IDE)
- Deployment instructions and connection to the Sepolia testnet

This material is released under the Creative Commons Zero (CC0 1.0 Universal) license, meaning it has been explicitly waived into the public domain. It may be freely used, modified, replicated, or adapted by anyone, without restrictions and without attribution, including for commercial purposes.

7) *Basic credit scoring model implemented:* The system includes an automated credit risk evaluation model, designed for users who complete the web form with traditional information. This model was developed to validate the functional flow of the system and demonstrate how automated credit decisions can be integrated into the proposed architecture.

Considered Criteria. The model takes into account the following variables, directly inspired by the structure of the dataset used:

- person_age: Applicant's age
- person_income: Monthly income
- person_home_ownership: Home ownership status (RENT, OWN, MORTGAGE, etc.)
- person_emp_length: Years of work experience
- loan_intent: Purpose of the loan (EDUCATION, MEDICAL, VENTURE, etc.)
- loan_grade: Assigned loan grade (A, B, C, etc.)
- loan_amnt: Requested loan amount
- loan_int_rate: Interest rate associated with the loan
- loan_percent_income: Ratio of loan amount to monthly income
- cb_person_default_on_file: Indicator of prior defaults (Y or N)
- cb_person_cred_hist_length: Length of credit history (in years).

Evaluation Process. The implemented model uses supervised machine learning techniques, specifically LightGBM, trained on traditional variables available through the web form (age, income, prior credit score, home ownership, etc.).

Once the applicant's data is processed, the model generates a credit score estimate on a scale comparable to FICO.

Based on that score, the system automatically classifies the application into one of three risk segments (Fig. 11).

The backend records both the risk classification and the action taken. In the case of medium risk, the status is marked as "pending human review," allowing system operators to conduct a more detailed analysis before issuing a final decision.

This segmentation reflects a simplified yet realistic credit policy, aligned with actual practices in the financial sector, and is adopted in this initial version as a mechanism for validating the technical architecture.

Purpose of the Current Model. This model is a baseline predictive component used to validate the end-to-end architecture. It is not production-grade and was not exhaustively optimized, but it is trained and evaluated as described in Section III (AUC = 0.94) to demonstrate operational integration.

- Integrating automated decision-making,
- Storing the results, and
- Registering approved credit decisions on blockchain.

The full implementation of this logic is available in the file credit_scoring.py within the aforementioned repository.

8) *Reference dataset:* To design the basic credit evaluation model, a publicly available dataset was used [37].

Dataset use. This dataset includes relevant information on loan applicants—age, income, home-ownership type, occupation, credit history, and approval outcomes. It was used to train, validate, and test a baseline LightGBM classifier (with a stratified split and class-imbalance weighting); it also informed the rule-based personalization and served to functionally validate the end-to-end prototype. No PII is required for our pipeline.

This decision aims to facilitate open collaboration and the development of technological solutions that promote financial inclusion in contexts where access to credit is limited.

Risk Segment	Score Range	System Decision
Low risk	Score > 700	Automatic approval
Medium risk	600 ≤ Score ≤ 700	Manual review
High risk	Score < 600	Automatic denial

Fig. 11. Summarizes the automated decisions made by the intelligent microcredit system according to the risk segment assigned by the LightGBM model.