

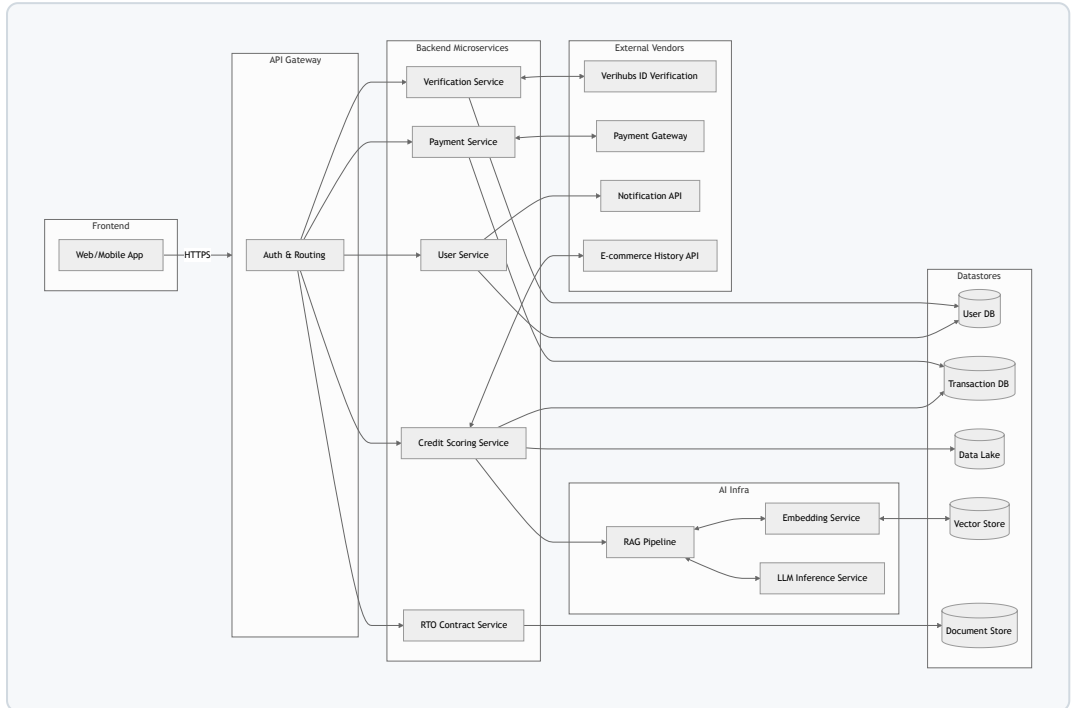
Indra Lesmana

+62 812 8766 2107 | indra6677@gmail.com

<https://indradevelop.github.io/tpm>

1. High-Level Architecture Design for Rent-to-Own (RTO) Platform

1.1. System Diagram Overview



1.2. API Specification: Credit Scoring System (RAG-Powered)

Base URL: <https://api.creditscorer.com/v1>

Endpoint: **POST** `/credit-scores`

Generates a credit score using RAG analysis of non-traditional data.

Request Headers:

- **Authorization:** Bearer <API_KEY> (Required)
- **Content-Type:** application/json
- **Accept:** application/json

Request Schema:

```
{
  "type": "object",
  "properties": {
    "ecommerce_transactions": {
      "type": "array",
      "items": {
        "type": "object",
        "properties": {
          "date": { "type": "string", "format": "date" },
          "amount": { "type": "number" },
          "merchant": { "type": "string" },
          "merchant_category": { "type": "string" },
          "status": { "type": "string", "enum": ["completed", "returned", "pending"] }
        },
        "required": ["date", "amount", "merchant"]
      },
    },
    "utility_payments": {
      "type": "array",
      "items": {
        "type": "object",
        "properties": {
          "utility_type": { "type": "string", "enum": ["electricity", "gas", "water"] },
          "payment_date": { "type": "string", "format": "date" },
          "amount_due": { "type": "number" },
          "amount_paid": { "type": "number" }
        },
        "required": ["utility_type", "payment_date", "amount_paid"]
      },
    },
    "rental_payments": {
      "type": "array",
      "items": {
        "type": "object",
        "properties": {
          "date": { "type": "string", "format": "date" },
          "amount": { "type": "number" },
          "landlord": { "type": "string" }
        },
        "required": ["date", "amount"]
      },
    },
    "bank_account": {
      "type": "object",
      "properties": {
        "average_balance_6mo": { "type": "number" },
        "overdraft_count_1yr": { "type": "integer" },
        "monthly_transaction_volume": { "type": "number" }
      },
    },
    "bnpl_history": {
      "type": "array",
      "items": {
        "type": "object",
        "properties": {
          "provider": { "type": "string" },
          "total_credit": { "type": "number" },
          "outstanding_balance": { "type": "number" },
          "missed_payments": { "type": "integer" }
        },
      },
    },
    "public_records": {
      "type": "object",
      "properties": {
        "declared_income": { "type": "number" },
        "tax_filings_last_3yr": { "type": "integer" },
        "income_sources": { "type": "array", "items": { "type": "string" } }
      },
    },
    "telco_data": {
      "type": "object",
      "properties": {
        "monthly_bill_amount": { "type": "number" },
        "late_payments_1yr": { "type": "integer" }
      },
    },
  },
  "minProperties": 1 // At least one data category must be provided
}
```

Response Schema:

```
{
  "type": "object",
  "properties": {
    "request_id": { "type": "string" },
    "credit_score": { "type": "integer", "minimum": 300, "maximum": 850 },
    "score_breakdown": {
      "type": "object",
      "properties": {
        "payment_consistency": { "type": "number" },
        "debt_utilization": { "type": "number" },
        "income_stability": { "type": "number" }
      }
    },
    "reasons": {
      "type": "object",
      "properties": {
        "positive": { "type": "array", "items": { "type": "string" } },
        "negative": { "type": "array", "items": { "type": "string" } }
      }
    },
    "rag_context": { "type": "string" }, // RAG-generated explanation
    "timestamp": { "type": "string", "format": "date-time" }
  },
  "required": ["credit_score", "rag_context"]
}
```

Example Request:

```
{
  "ecommerce_transactions": [
    { "date": "2023-10-05", "amount": 150.0, "merchant": "Shopify", "status": "completed" }
  ],
  "telco_data": {
    "monthly_bill_amount": 45.0,
    "late_payments_1yr": 0
  }
}
```

Example Response:

```
{
  "request_id": "req_abc123",
  "credit_score": 720,
  "score_breakdown": {
    "payment_consistency": 0.85,
    "debt_utilization": 0.45
  },
  "reasons": {
    "positive": ["Consistent e-commerce spending", "No late telco payments"],
    "negative": ["Limited rental payment history"]
  },
  "rag_context": "The score reflects strong payment consistency in e-commerce and telecom. However, limited rental history reduces confidence in housing expense management.",
  "timestamp": "2023-10-10T14:30:00Z"
}
```

Error Handling:

HTTP Code	Error Type	Response Body Example
400	Invalid Data	{ "error": "invalid_data", "message": "Missing required field: amount", "details": {...} }
401	Unauthorized	{ "error": "unauthorized", "message": "Invalid API key" }
404	Not Found	{ "error": "not_found", "message": "Endpoint /v1/credit-scores does not exist" }
429	Rate Limit Exceeded	{ "error": "rate_limit", "message": "Exceeded 10 requests per minute" }
500	Server Error	{ "error": "server_error", "message": "Internal processing failure" }

Notes:

- 1. **Authentication:** Uses API keys with `Bearer` token authentication.
- 2. **RAG Context:** Explains scoring rationale in natural language, generated dynamically based on input data patterns.

1.3. Comparison of Payment Gateway Options: Midtrans vs. Xendit

Here is a comparison between two popular payment gateway providers in Indonesia: **Midtrans** and **Xendit**, based on cost, performance, and regulatory compliance.

💰 Transaction Fees

Payment Methods	Midtrans	Xendit
Bank Transfer	IDR 4,000 per transaction	IDR 4,000 (Aggregator), IDR 2,000 + bank fee (Switcher)
Credit Card	2.9% + IDR 2,000 per transaction	2.9% + Rp2,000 (Aggregator); 2.9% + bank fees (Switcher)
E-Wallet	GoPay: 2%, QRIS: 0.7%, ShopeePay: 2%, DANA: 1.5%	DANA: 1.5%, OVO: 1.5%–3.18%, ShopeePay: 2%–4%, LinkAja: 1.5%–3.15%
PayLater	Akulaku: 1.7%, Kredivo: 2%	Akulaku: 1.7%, Kredivo: 2.3%, Atome: 5%
Payout/Disbursement	Rp5,000 to bank account, Rp1,000 to GoPay	Rp5,000 to bank account

Note: All fees are subject to 11% VAT.

⚡ Performance & Features

Aspects	Midtrans	Xendit
Settlement	T+1 to T+3 business days, depending on payment method	Instant settlement for most methods, except BCA (T+1)
Refund	Not explicitly stated	Supports full and partial refunds directly from the dashboard
Integrations	Limited to platforms like Shopify and WooCommerce	Supports multiple platforms with features like recurring payments and e-wallet tokenization
Customer Support	There have been complaints about slow responses and fund holds	Received positive reviews regarding the speed and quality of customer service

✅ Compliance & Security

Aspects	Midtrans	Xendit
Certification	PCI DSS Level 1, ISO 27001, registered with Bank Indonesia and Kominfo	PCI DSS Level 1, ISO 27001, registered with Bank Indonesia and Kominfo
Data Encryption	Using AES-256 encryption	Using AES-256 encryption

📄 Summary

Criteria	Midtrans	Xendit
Pros	<ul style="list-style-type: none">- Fixed fee for bank transfers- Integrated with GoTo ecosystem- Supports 25+ payment methods	<ul style="list-style-type: none">- Instant settlement- Direct refund from dashboard- Extensive integration with various platforms- Responsive customer service
Cons	<ul style="list-style-type: none">- Slower settlement- Limited integration- Complaints related to customer service	<ul style="list-style-type: none">- E-wallet fees vary depending on merchant type- Some advanced features may require additional settings

🧠 Recommendation

- **Choose Midtrans** if you are looking for a solution with a fixed fee for bank transfers and is already integrated with the GoTo ecosystem.
- **Choose Xendit** if you need instant settlement, flexible refund features, and extensive integration with various platforms.

2. Rent-to-Own Product MVP Analysis

Here's a concise summary of the recommended approach, followed by detailed sections for each deliverable. We lean on industry best practices in MVP feature selection, fintech onboarding, AI-driven credit underwriting (via RAG + LLM), payments, and fraud controls to structure a six-month rollout that balances speed, risk, and business impact.

2.1. Prioritized Roadmap

Here's how I'd prioritize the features for your Rent-to-Own MVP, focusing on delivering core functionality and a smooth user experience first:

Rank	Feature	Justification
1	User onboarding with digital KYC	Seamless KYC is the gateway for customer acquisition in fintech – poor onboarding drops activation by up to 40%. Digital KYC reduces manual errors and regulatory friction.
2	Integration with e-commerce partners	Transaction history is critical for credit assessment and engagement. Aggregating e-commerce data boosts underwriting accuracy by 20–30%. It also increases user “stickiness” through contextual offers.
3	AI-driven credit limit calculator (RAG + LLM)	RAG architectures let LLMs leverage proprietary transaction data for explainable credit decisions. This accelerates personalization and risk management versus static scoring models.
4	Payment scheduling & reminders	Recurring payments drive revenue predictability; automated reminders cut late payments by 30–50% and improve cash-flow for both user and platform.
5	Fraud detection system	Essential for risk mitigation, but a full ML-powered fraud engine can follow after basic product-market fit. Initial rule-based checks suffice in MVP, with advanced ML fraud as Phase 2.

2.2. Release Plan

Timeline & Dependencies

Month	Features	Dependencies	Success Metrics
1–2	Digital KYC	Vendor selection (e.g. Verihubs), UX flows	80% onboarding completion rate; <5% manual review escalation
2–3	E-commerce integration	KYC live, API partnerships (Shopee, Tiktok, etc.)	90% successful data fetch; 95% transaction coverage

Month	Features	Dependencies	Success Metrics
3-4	AI credit limit calculator (AI RAG/LLM)	Transaction data pipeline, LLM & vector store setup	85% accuracy in credit limit recommendations; <10% user disputes.
4-5	Payment scheduling & reminders	Credit decision live, calendar/notification service	90% user satisfaction with payment flow; <2% failed transactions.
5-6	Fraud detection (AI RAG/LLM)	Transaction & payment modules live	<1% fraud loss rate; false positive rate <5%

Key Milestones

- **End M2:** 1,000 onboarded users, transaction data flowing
- **End M4:** 80% credit decisions automated via RAG+LLM
- **End M6:** MVP processes 10,000 monthly transactions with <1% fraud losses

2.3. Stakeholder Trade-off: Negotiating with the CEO

1. **Understand the CEO's Concerns:** Ask the CEO to elaborate on the reasons for prioritizing the fraud detection system last. Understanding the rationale can help in addressing specific concerns.
2. **Present Data-Driven Insights:** Use industry benchmarks and case studies to demonstrate the criticality of foundational features like user onboarding and credit limit calculation for the MVP's success.
3. **Highlight Interdependencies:** Explain how the early implementation of other features directly impacts the effectiveness of the fraud detection system. For instance, accurate transaction history and credit limits are essential for a robust fraud detection model.
4. **Propose a Phased Approach:** Suggest a minimal viable fraud detection mechanism that can be implemented earlier, with a plan to enhance it in subsequent phases. This shows commitment to security without delaying critical features.
5. **Risk Mitigation Plan:** Present a temporary risk mitigation strategy to address fraud concerns during the MVP phase. This could include manual reviews, enhanced monitoring, or third-party fraud detection services.
6. **Align with Business Goals:** Emphasize how the proposed prioritization aligns with the company's broader business goals, such as user acquisition, revenue generation, and market validation.

3. AI Credit System Case Study

3.1. Data Strategy: Alternative Data Sources

- **Digital Wallet Transactions** (e.g., GoPay, OVO, Dana):
 - **Why?** Captures broader financial behavior (e.g., peer-to-peer transfers, offline merchant payments, bill splits).
 - **Use Case:** Users with limited e-commerce activity may still demonstrate financial responsibility through consistent digital wallet usage.

- **Gig Economy Platforms** (e.g., Gojek, Grab, freelance marketplaces):
 - **Why?** Validates income stability for freelancers via delivery/ride-hailing earnings or project-based income.
 - **Use Case:** Substitute for traditional payroll records, especially for unbanked gig workers.
- **Mobile App Usage Patterns** (with user consent):
 - **Why?** Analyzes financial app engagement (e.g., budgeting tools, investment apps) to infer financial literacy.
 - **Use Case:** Users without e-commerce history might use financial apps regularly, signaling creditworthiness.

3.2. Integration Plan: Data Pipeline for RAG LLM

Ensure seamless data flow with this pipeline:

1. API Integration & Authentication:

- Partner with third-party vendors (Shopee, Gojek, Telco providers) to access APIs.
- Use **OAuth 2.0** for secure authorization and **tokenization** to protect sensitive data (e.g., bank account numbers).

2. Data Ingestion:

- Deploy **real-time streams** (Apache Kafka) for dynamic data (e-commerce transactions) and **batch processing** (Airflow) for static data (utility bills).

3. Preprocessing & Enrichment:

- Clean data (handle missing values, normalize formats) and enrich with derived features:
 - **Payment Consistency Score:** % of on-time payments across utilities, rentals, and BNPL.
 - **Income Stability Index:** Variance in gig economy earnings over 6 months.

4. Vectorization for RAG LLM:

- Convert structured data (e.g., transaction amounts) and unstructured data (e.g., public records) into embeddings using **BERT-based models**.
- Store in a **vector database** (e.g., FAISS) for retrieval during credit scoring.

5. Model Integration:

- The RAG LLM retrieves context from the vector DB (e.g., similar user profiles, payment trends) and combines it with real-time data to generate a score.

6. Monitoring & Feedback Loop:

- Track pipeline health (Latency, API errors) and model performance (AUC-ROC, bias metrics).
- Retrain models quarterly using updated data.

3.3. Risk Mitigation for Model Bias

To reduce bias against users with no e-commerce history:

- **Dynamic Feature Weighting:**
 - Use **ensemble models** where non-e-commerce features (utility payments, telco data) receive higher weights for users lacking e-commerce activity.

- **Proxy Indicators:**
 - Infer reliability from correlated behaviors (e.g., consistent mobile bill payments = high responsibility).
- **Unsupervised Clustering:**
 - Group users without e-commerce data into clusters (e.g., "high-risk freelancers" vs. "stable gig workers") using telco, rental, and gig platform data.
- **Fairness Audits:**
 - Regularly test for disparate impact (e.g., ensure approval rates for freelancers vs. salaried users differ by <20%).
- **Fallback Mechanisms:**
 - Deploy **rule-based checks** (e.g., manual review if the user has strong rental history but no e-commerce data).
- **RAG LLM Augmentation:**
 - Retrieve anonymized profiles of similar users (e.g., "freelancers with 12+ months of on-time utility payments") to contextualize scoring.

4. Vendor Conflict Resolution

4.1. Action Plan to Resolve the Conflict

- **Step 1: Review Contract and Documentation**
 - Assess the original contract to verify if the disputed change requests were documented and align with terms. Check clauses related to payment conditions, deadlines, and change management.
 - Document all undocumented change requests and cross-reference them with the vendor's claims about unclear requirements
- **Step 2: Mediate Between Stakeholders**
 - Facilitate a meeting with the vendor, legal/finance team, and business team to clarify expectations and resolve disputes. Use mediation to address the unpaid invoice and missed deadline
 - Negotiate a partial payment for completed work (if justified) to unblock progress, while tying the remaining payment to revised deliverables
- **Step 3: Adjust Project Plan for MVP Launch**
 - Prioritize core features required for the MVP launch, leveraging iterative development to meet the deadline Reduce scope creep by freezing non-critical changes.
 - Set incremental milestones with strict deadlines and daily progress checks to ensure on-time delivery
- **Step 4: Formalize Resolution and Communication**
 - Draft a written agreement outlining revised deadlines, payment terms, and approved change requests. Ensure all parties sign to avoid future disputes
 - Notify stakeholders (e.g., launch event organizers) of the adjusted timeline and mitigation steps

4.2. Preventive Measures for Future

A. Strengthen Requirement Clarity

- Define UI/UX, tech stack (e.g., frameworks, languages), and acceptance criteria upfront during contract finalization. Use prototypes or wireframes to align expectations.

B. Formalize Change Request Process

- Include a clause in contracts mandating written approval for changes affecting scope, cost, or timelines. Only allow changes that align with core features

C. Adopt Flexible Contracts for Startups

- Use time & material contracts for MVP development to accommodate agility, then shift to retainer-based agreements post-launch for ongoing support. Avoid fixed-price contracts for early-stage projects with evolving requirements.

D. Proactive Monitoring and Feedback

- Implement weekly progress reviews and real-time collaboration tools (e.g., Jira, Slack) to flag delays early.
- Include penalty clauses for missed deadlines and reward timely delivery to incentivize vendors.

E. Centralize Documentation

- Maintain a single source of truth (e.g., shared portal) for all requirements, change requests, and approvals to prevent "he-said-she-said" disputes.

5. User Growth Strategy

5.1. Hypothesis: Why Are Users Dropping Off?

• Hypothesis 1: Complexity and Length of the KYC Process

Users may find the KYC process too complicated or time-consuming, leading to frustration and drop-off.

- **A/B Test Proposal:** Simplified KYC process with fewer steps and clearer instructions.

• Hypothesis 2: Lack of Trust and Privacy Concerns

Users may be hesitant to provide sensitive personal information due to concerns about data security and privacy.

- **A/B Test Proposal:** KYC process with enhanced security assurances, such as trust badges, privacy policy links, and a brief explanation of how data is protected.

5.2 Metrics: Defining KPIs to Measure Success

1. Analytic Logging

Identify specific points **User Retention Rate, Onboarding Completion Rate, Drop-off Rate, Time to Complete** and **User Feedback Score** in the KYC process where users are dropping off.

2. Analyze Logs Using Metrics

Quantify the drop-off rate and identify patterns or common issues. Use the defined KPIs to analyze the logged data, focusing on the drop-off rate, completion rate, and time to complete KYC.

3. Advanced Analysis Using LLM

Gain deeper insights into user behavior and potential reasons for drop-off. Utilize a Large Language Model (LLM) to analyze user feedback, identify sentiment trends, and suggest improvements based on natural language processing.