USPTO PROVISIONAL PATENT APPLICATION

COMPLETE USPTO PROVISIONAL PATENT FILING PACKAGE

UNITED STATES PATENT AND TRADEMARK OFFICE

System and Method for a Multi-Tenant Vertical Marketplace Operating System Featuring Temporal Incentives, Circulatory Multiplier Attribution, Cross-Business Coalition Loyalty Networks, Voice-Enabled AI Concierge, Hierarchical Sales Agent Networks, and Geospatial Velocity Fraud Detection

Filing Date: January 30, 2026

Application Number: _____

Applicant/Inventor: Thomas D. Bowling

Commercial Name(s): 1325.AI (dba Mansa Musa Marketplace)

Correspondence Address: 1000 E. 111th Street, Suite 1100, Chicago, Illinois 60628

Contact: 312.709.6006 | contact@1325.AI

PREPARED BY / ATTORNEY OF RECORD

Fraline J. Allgaier, Esq.

Allgaier Patent Solutions

405 N. Wabash #2912, Chicago, IL 60611

Phone: (847) 409-8670

www.allgaierpatentsolutions.com

TABLE OF CONTENTS

ABSTRACT

A comprehensive multi-tenant marketplace operating system designed to support minority-owned businesses through an integrated ecosystem of novel technical systems. The invention includes:1. Temporal Founding Member Status Assignment via database triggers with immutable lifetime benefits2. Economic Circulation Multiplier Attribution (CMAL) using culturally-derived 2.3x constant3. Cross-Business Coalition Loyalty Networks with tiered point multipliers (Bronze 1x, Silver 1.25x, Gold 1.5x, Platinum 2x)4. Geospatial Velocity-Based Fraud Detection using AI pattern analysis with impossible travel detection5. Intelligent B2B Matching with weighted multi-factor scoring (category 30pts, location 20pts, budget 15pts, rating 15pts)6. Voice-Enabled Concierge Services via WebSocket bridge to OpenAI Realtime API7. Hierarchical Sales Agent Commission Networks with team overrides and recruitment bonuses8. Gamification Layers including achievements, streaks, leaderboards, and group challenges9. QR-Code Transaction Processing with 7.5% integrated commission splitting via Stripe Connect10. AI-Powered Personalized Business Recommendations with Gemini 2.5 Flash integration11. Real-Time Voice AI Bridge Architecture with persona injection and VAD configuration12. AI Tool Registry for Voice Concierge with typed Zod validation13. Atomic Fraud Alert Batch Insertion using PostgreSQL JSONB array processing14. Economic Karma Scoring System for Cerebro-compatible data feeds15-20. Community Finance, Biometric Verification, QR Atomic Check-in, Impact Analytics, Closed-Loop Wallet, Velocity Analytics21-26. Partner Referral Attribution System with revenue sharing, Founding Partner tiers, embeddable widgets, analytics, vetting workflows, and multi-tier affiliate network27. Automated Partner Marketing Toolkit with dynamic attribution injection, ROI messaging, multi-format collateral generation, and content analytics

FIELD OF THE INVENTION

The present invention relates generally to electronic commerce platforms and more specifically to a comprehensive multi-tenant vertical marketplace operating system designed to support minority-owned businesses through an integrated ecosystem of:1. Temporal incentive structures with immutable founding member status2. Economic circulation tracking with culturally-derived multiplier constants3. Cross-business coalition loyalty programs with tiered point systems4. AI-powered business matching with multi-factor weighted scoring5. Voice-enabled concierge services with natural language processing6. Gamification systems with achievements, streaks, and group challenges7. Multi-tier sales agent networks with commission hierarchies8. Corporate sponsorship impact attribution with real-time metrics9. Geospatial velocity-based fraud detection using AI pattern analysis10. QR-code transaction processing with integrated commission splitting11. Real-time voice AI interface with WebSocket streaming12.

Community finance instruments (Susu circles)13. HBCU student verification and discount programs14.

Business import and lead generation systems

BACKGROUND OF THE INVENTION

Problem Statement

Traditional e-commerce and business directory platforms fail to address the unique economic challenges faced by minority-owned businesses, particularly Black-owned businesses in the United States. Research indicates that the "Black dollar" circulates within Black communities for only 6 hours compared to 20 days in other communities, representing a significant wealth disparity that perpetuates economic inequality.

Deficiencies in Existing Systems

1. Lack of Economic Circulation Awareness: No existing platform tracks, visualizes, or incentivizes economic circulation within specific communities to maximize wealth retention.
2. Fragmented Loyalty Programs: Current loyalty systems are siloed to individual businesses, failing to create network effects that benefit both consumers and business ecosystems.
3. Insufficient Fraud Detection: Existing QR-code based transaction systems lack sophisticated geospatial and temporal analysis to prevent abuse such as impossible travel scenarios or coordinated manipulation.
4. No Temporal Early-Adopter Incentives: Platforms fail to create immutable, time-based incentive structures that reward early participation with permanent benefits.
5. Weak B2B Connectivity: Minority-owned businesses lack tools to discover and transact with other minority-owned businesses for supply chain and service needs.
6. Limited Corporate Sponsorship Attribution: Corporate sponsors supporting minority communities cannot accurately measure the real economic impact of their investments.
7. No Voice-First Accessibility: Existing platforms lack natural language voice interfaces for users who prefer verbal interaction.
8. Absence of Commission-Based Growth Networks: No existing platform provides structured sales agent programs with hierarchical commission structures for platform growth.

SUMMARY OF THE INVENTION

System Architecture

Frontend Layer
Progressive Web Application (PWA) and native mobile capabilities

- React 18.3.1, TypeScript 5.x, and TailwindCSS 3.x
- Native mobile applications via Capacitor 7.4.3 for iOS and Android
- Real-time state management using TanStack Query (React Query) 5.56.2
- Framer Motion 12.10.0 for animated visualizations
- Offline-first architecture with service worker caching

Backend Layer

Serverless infrastructure with real-time capabilities

- Supabase/PostgreSQL 15+ database with Row-Level Security (RLS)
- Edge Functions (Deno 1.x runtime) for serverless business logic
- Real-time subscriptions via WebSocket connections
- Stripe Connect for payment processing with 7.5% platform commission
- Rate limiting at function level with in-memory stores

AI/ML Layer

Artificial intelligence and machine learning integration

- Lovable AI Gateway integration (Google Gemini 2.5 Flash, OpenAI GPT-4o)
- OpenAI Realtime API for voice interactions
- Custom prompt engineering for business-specific AI applications
- Rate-limited API consumption with fallback mechanisms
- Tool-based structured output extraction

FORMAL CLAIMS (14 Independent + 25+ Dependent)

CLAIM 1: TEMPORAL FOUNDING MEMBER STATUS SYSTEM

Independent Claim 1:

A computer-implemented method for automatically and irrevocably assigning founding member status to platform users, comprising:a) receiving a user registration request at a server, wherein said request includes an associated timestamp representing the moment of registration;b) executing a database trigger function that compares said registration timestamp against a predetermined temporal cutoff constant stored within said trigger function, wherein said cutoff constant is defined as a specific date and time value that cannot be modified at runtime;c) upon determining that the registration timestamp precedes the temporal cutoff constant, automatically setting a founding member boolean flag to TRUE within a user profile record;d) simultaneously storing a founding_member_since timestamp that preserves the exact registration moment with microsecond precision;e) enforcing immutability of said founding member status through database-level constraint triggers that prevent any subsequent modification, revocation, or deletion of the founding member flag through any application interface, administrative action, or direct database manipulation;f) persisting said founding member status across all account modifications including but not limited to password resets, email address changes, account merges, and platform version upgrades.

Dependent Claim 1.1: The method of Claim 1, wherein the temporal cutoff constant is encoded as a UTC timestamp in ISO 8601 format and stored as a CONSTANT declaration within the database trigger function, thereby preventing modification without database migration.

Dependent Claim 1.2: The method of Claim 1, further comprising a secondary constraint trigger that raises a database exception when any UPDATE statement attempts to change the is_founding_member field from TRUE to FALSE, with the exception message stating "Founding member status cannot be revoked."

Dependent Claim 1.3: The method of Claim 1, wherein the founding member status provides automatic enrollment in a lifetime benefits program comprising: Permanent 2x loyalty point multiplier on all transactions, Priority customer support queue placement, Waived premium subscription fees, Distinguished visual badge display across all platform interfaces, Reduced platform transaction fees.

Dependent Claim 1.4: The method of Claim 1, wherein the founding_member_since timestamp is used to establish tiered founding member classifications based on registration order, including "First 100," "First 1,000," and "First 10,000" designations, each with progressively differentiated benefit packages.

Technical Implementation:

```
DECLARE v_cutoff_timestamp CONSTANT TIMESTAMP WITH TIME ZONE := '2026-09-01T23:59:59Z'; BEGIN IF
NEW.created_at < v_cutoff_timestamp THEN NEW.is_founding_member := true; NEW.founding_member_since :=
NEW.created_at; END IF; RETURN NEW; END;
```

CLAIM 2: ECONOMIC CIRCULATION MULTIPLIER ATTRIBUTION ENGINE (CMAL)

Independent Claim 2:

A computer-implemented system for calculating and attributing community economic impact through application of a culturally-derived circulation multiplier constant, comprising:a) a transaction monitoring component that captures all monetary transactions occurring within a multi-tenant marketplace platform, wherein each transaction includes at minimum: transaction amount, business identifier, customer identifier, and timestamp;b) a multiplier attribution engine that applies a circulation multiplier constant of 2.3 to each transaction amount, wherein said constant represents the empirically-derived number of times currency circulates within the target demographic community before exiting, as established by Federal Reserve Bank studies on minority business economics;c) a community reach calculator that estimates the number of individuals impacted by applying a reach multiplier constant of 10 to the count of unique customers, representing the extended network effect of each transaction;d) an aggregation service that computes cumulative metrics including: total businesses supported (count of unique business_ids), total transactions (count of all transaction records), raw transaction value (sum of transaction amounts), and total economic impact (raw transaction value multiplied by the 2.3 circulation constant);e) a sponsor attribution component that associates said metrics with corporate sponsor subscriptions, enabling real-time visualization of sponsor investment returns through dashboard interfaces;f) a persistent storage mechanism that stores daily snapshots of impact metrics with subscription_id and metric_date as a composite unique constraint, enabling time-series analysis of community investment performance.

Dependent Claim 2.1: The system of Claim 2, wherein the circulation multiplier constant of 2.3 is derived

from peer-reviewed economic research specific to Black American communities, including National

Bureau of Economic Research publications on minority business wealth retention patterns.

Dependent Claim 2.2: The system of Claim 2, further comprising an animated visualization component that displays the multiplication effect in real-time, showing the flow from initial purchase amount through community circulation to final economic impact, using motion graphics to illustrate the wealth multiplication process.

Dependent Claim 2.3: The system of Claim 2, wherein the economic impact calculation is performed by an edge function executing in a serverless environment, with the formula: economic_impact = SUM(transaction_amount) * CIRCULATION_MULTIPLIER, where CIRCULATION_MULTIPLIER equals 2.3.

Dependent Claim 2.4: The system of Claim 2, further comprising a sponsor dashboard interface that displays: Businesses supported count with comparison to previous period, Total economic impact with percentage growth, Community reach estimation with visualization, Return on Investment (ROI) calculation comparing sponsor contribution to generated economic activity.

CLAIM 3: CROSS-BUSINESS COALITION LOYALTY NETWORK

Independent Claim 3:

A computer-implemented cross-merchant loyalty point system enabling universal point earning and redemption across a coalition of participating businesses, comprising:a) a coalition points ledger maintaining a single unified point balance per customer, stored in a coalition_points table with customer_id as a unique constraint;b) a tiered membership system with four distinct tiers (Bronze, Silver, Gold, Platinum), each tier defined by lifetime point thresholds and associated with a point earning multiplier (1.0x, 1.25x, 1.5x, 2.0x respectively);c) an automatic tier promotion mechanism implemented as a database trigger that evaluates lifetime_earned points against tier thresholds after each point transaction and automatically updates the customer's tier designation;d) a point awarding function that calculates final points by multiplying base points by the customer's current tier multiplier, atomically updating both the current balance and lifetime earned total, and recording the transaction in an audit log;e) a cross-merchant redemption capability wherein points earned at any coalition member business may be redeemed for value at any other coalition member business, with the redemption transaction recording both the source and destination business identifiers;f) a coalition membership registry that tracks participating businesses with activation status, total points distributed, and total points redeemed, enabling network-wide analytics.

Dependent Claim 3.1: The system of Claim 3, wherein tier thresholds are defined as: Bronze: 0-999 lifetime points, Silver: 1,000-4,999 lifetime points, Gold: 5,000-14,999 lifetime points, Platinum: 15,000+ lifetime points.

Dependent Claim 3.2: The system of Claim 3, wherein the point awarding function is implemented as a PostgreSQL stored procedure with SECURITY DEFINER privileges, ensuring consistent application of tier multipliers regardless of calling context.

Dependent Claim 3.3: The system of Claim 3, further comprising a coalition transactions table that

records every point movement with transaction_type classification including: earn, redeem, transfer, bonus, referral, and expiry.

Dependent Claim 3.4: The system of Claim 3, wherein the coalition network creates a positive-sum economic model where: Customers are incentivized to patronize multiple coalition businesses to maximize point accumulation, Businesses benefit from increased traffic driven by cross-business point redemption, The platform benefits from increased transaction volume and user engagement.

CLAIM 4: GEOSPATIAL VELOCITY FRAUD DETECTION SYSTEM

Independent Claim 4:

An artificial intelligence-powered fraud detection system utilizing geospatial velocity analysis to identify impossible travel scenarios and abuse patterns, comprising:a) a data collection component that captures transaction and activity records including geographic coordinates, timestamps, and user identifiers from QR code scans, transactions, and platform activities;b) a velocity calculation engine that computes implied travel speed between sequential user actions using the formula: $V = D / delta\_t$, where V equals implied velocity in miles per hour, D equals Haversine distance between geographic coordinates, and delta_t equals the time delta between actions in hours;c) a configurable velocity threshold comparator, with a default threshold of 600 mph to account for commercial aviation, that flags any calculated velocity exceeding the threshold as a potential impossible travel scenario;d) an AI-mediated pattern analysis component utilizing a large language model (LLM) to analyze aggregated activity patterns and identify coordinated abuse, transaction anomalies, and suspicious account behavior;e) a structured alert generation mechanism using LLM function calling (tool use) to produce typed fraud alerts containing: alert_type, severity level (low/medium/high/critical), affected user_id, affected business_id, human-readable description, supporting evidence, and AI confidence score;f) an atomic batch insertion function that stores all generated alerts in a single database transaction using PostgreSQL jsonb_array_elements processing;g) a privacy-preserving data sanitization layer that removes or redacts personally identifiable information (IP addresses, email addresses, phone numbers, user agents) before transmitting data to external AI services.

Dependent Claim 4.1: The system of Claim 4, wherein the fraud alert types include: velocity_abuse (Action frequency exceeding human-possible rates), location_mismatch (Geographically impossible travel detected), qr_scan_abuse (Coordinated scanning patterns across accounts), transaction_anomaly (Unusual transaction amounts or frequencies), account_suspicious (New accounts with immediate high activity), review_manipulation (Burst patterns in reviews from related accounts).

Dependent Claim 4.2: The system of Claim 4, wherein detection of a velocity exceeding the threshold triggers an automatic escrow hold on associated points and pending payouts until manual review is completed.

Dependent Claim 4.3: The system of Claim 4, wherein the AI confidence score is a decimal value between 0.0 and 1.0, with alerts scoring above 0.85 automatically escalated to critical priority and alerts scoring below 0.5 requiring additional evidence before action.

Dependent Claim 4.4: The system of Claim 4, wherein the batch insertion function is implemented using PostgreSQL's jsonb_array_elements for atomic processing.

Dependent Claim 4.5: The system of Claim 4, wherein the atomic batch insertion (Claim 13) specifically prevents 'race conditions' wherein a fraudster attempts multiple simultaneous transactions before the first fraud alert is processed, by: (a) Using PostgreSQL SERIALIZABLE isolation level for fraud detection queries; (b) Implementing pessimistic locking on user accounts during velocity calculation via SELECT ... FOR UPDATE; (c) Queueing all transactions from the same user_id for sequential processing within a configurable time window (default: 500ms); (d) Maintaining a transaction_queue table with user_id, queued_at timestamp, and processing_status to ensure FIFO ordering of fraud checks.

CLAIM 5: AI-POWERED B2B MATCHING ENGINE

Independent Claim 5:

An intelligent business-to-business matching system connecting minority-owned businesses using multi-factor weighted scoring and AI-enhanced recommendations, comprising:a) a business needs registry where businesses post procurement requirements including category, budget range, urgency level, and preferred geographic area;b) a business capabilities registry where businesses advertise their products and services with category, pricing range, service area, lead time, and certifications;c) a multi-factor weighted scoring algorithm that evaluates each capability against each need using the following scoring weights: CATEGORY_MATCH: 30 points, SAME_CITY: 20 points, SAME_STATE: 10 points, SERVICE_AREA_OVERLAP: 15 points, BUDGET_COMPATIBILITY: 15 points, RATING_BONUS_MAX: 15 points, TIMELINE_MATCH: 10 points;d) a score normalization component that caps the maximum composite score at 100 points;e) a ranking and filtering service that sorts matches by score descending and returns the top 10 candidates;f) an AI enhancement layer that generates natural language recommendations for top matches by providing the need description, match details, and score reasons to a large language model;g) a connection tracking system that records initiated B2B relationships with status progression (pending, contacted, negotiating, contracted, completed, declined) and actual transaction values for network effect measurement.

Dependent Claim 5.1: The system of Claim 5, wherein the urgency to timeline matching uses the following day mappings: immediate: 3 days maximum, within_week: 7 days maximum, within_month: 30 days maximum, planning: 90 days maximum, flexible: 180 days maximum.

Dependent Claim 5.2: The system of Claim 5, wherein the AI recommendation is generated using Google Gemini 2.5 Flash model with a prompt specifically instructing the model to act as a "B2B matchmaker helping Black-owned businesses connect."

Dependent Claim 5.3: The system of Claim 5, further comprising a B2B messaging system that enables direct communication between matched businesses while maintaining conversation history linked to the connection record.

CLAIM 6: REAL-TIME VOICE AI BRIDGE ARCHITECTURE

Independent Claim 6:

A WebSocket-based voice artificial intelligence interface enabling natural language interaction with a marketplace platform, comprising:a) an edge function that upgrades incoming HTTP requests to WebSocket

connections using Deno.upgradeWebSocket();b) a secondary WebSocket connection initiated from the edge function to an external real-time voice AI provider (OpenAI Realtime API), creating a bidirectional bridge between the client and AI service;c) a session configuration mechanism that injects platform-specific persona instructions, knowledge base, and behavioral guidelines into the AI session upon connection establishment;d) a bidirectional message relay that forwards all messages from the client WebSocket to the AI provider WebSocket and vice versa, maintaining real-time audio streaming capability;e) a voice-to-text and text-to-voice processing pipeline utilizing PCM16 audio format for both input and output streams;f) a server-side voice activity detection (VAD) configuration with tunable parameters including threshold (0.5), prefix padding (300ms), and silence duration (800ms) for turn detection;g) graceful connection lifecycle management that closes both WebSocket connections when either party disconnects.

Dependent Claim 6.1: The system of Claim 6, wherein the persona instructions define a character named "Kayla" with specific behavioral guidelines including: Use of natural contractions, Addition of filler words, Expression of genuine emotion, Response length limitation to 2-4 sentences, Platform-specific knowledge injection.

Dependent Claim 6.2: The system of Claim 6, wherein the voice output uses the "shimmer" voice preset with temperature setting of 0.8 for natural variation.

Dependent Claim 6.3: The system of Claim 6, further comprising a voice concierge tools function that provides structured tool implementations for: search_businesses, get_business_details, check_availability, get_recommendations, check_coalition_points, start_booking.

CLAIM 7: MULTI-TIER SALES AGENT COMMISSION NETWORK

Independent Claim 7:

A hierarchical sales agent network with tiered commission structures and multi-level incentive systems, comprising:a) a sales agent registry storing agent profiles with unique referral codes, commission rates, team override rates, tier designations, and recruitment relationships via a self-referencing recruited_by_agent_id foreign key;b) a five-tier progression system (Bronze, Silver, Gold, Platinum, Diamond) with tier advancement triggered by lifetime referral counts: Bronze: 0-24 referrals (10% commission), Silver: 25-99 referrals (12% commission), Gold: 100-199 referrals (13% commission), Platinum: 200-499 referrals (14% commission), Diamond: 500+ referrals (15% commission);c) a direct commission calculation engine that applies the agent's tier-based commission rate to subscription revenue from referred users;d) a team override commission system wherein a recruiter agent receives a percentage (default 2.5%) of commissions earned by agents they recruited, creating a two-tier income structure;e) a recruitment bonus mechanism that awards a fixed bonus (default $50) to the recruiter when a recruited agent achieves activation criteria;f) a commission status tracking system with states: pending, approved, processing, paid, and cancelled, with associated payment references and dates;g) a referral tracking table linking agents to referred users with conversion tracking, commission amounts, and tier designation (1 = direct, 2 = team bonus).

Dependent Claim 7.1: The system of Claim 7, wherein the team override commission is calculated as: override_amount = recruited_agent_commission * team_override_rate where default team_override_rate = 0.025 (2.5%).

Dependent Claim 7.2: The system of Claim 7, wherein tier progression is automatically calculated by a database trigger that fires on UPDATE of the lifetime_referrals column.

Dependent Claim 7.3: The system of Claim 7, further comprising an agent performance dashboard displaying: Total referrals with conversion rate, Current tier with progress to next tier, Pending, approved, and paid commission amounts, Team size and team-generated revenue, Leaderboard ranking among peers.

CLAIM 8: GAMIFICATION AND ACHIEVEMENT SYSTEM

Independent Claim 8:

A comprehensive gamification system incorporating achievements, streaks, leaderboards, and group challenges to incentivize platform engagement, comprising:a) an achievement definitions table storing unlock criteria as JSONB, allowing flexible condition specification for achievements across categories: shopping, social, loyalty, community, and special;b) a user achievements junction table recording achievement unlock timestamps with partial progress tracking for multi-step achievements;c) a streak tracking system monitoring continuous activity patterns with streak types including daily_visit, weekly_purchase, review_streak, and referral_streak, storing current_streak, longest_streak, and last_activity_date with streak_multiplier bonuses;d) a leaderboard system with period-based rankings (daily, weekly, monthly, all_time) and automatic rank calculation based on point accumulation within each period;e) a group challenges framework enabling community-wide goals with challenge types: spending, visits, referrals, reviews, and coalition_points, tracking target_value, current_value, participant_count, and collective reward distribution;f) a challenge participation table recording individual contributions toward group goals with contribution_value aggregation.

Dependent Claim 8.1: The system of Claim 8, wherein streak bonuses apply a multiplier to points earned, with the multiplier increasing based on streak length: 1-7 days: 1.0x, 8-14 days: 1.1x, 15-30 days: 1.25x, 31+ days: 1.5x.

Dependent Claim 8.2: The system of Claim 8, wherein secret achievements are hidden from the user interface until unlocked, creating surprise and delight moments that encourage platform exploration.

CLAIM 9: QR-CODE TRANSACTION PROCESSING ENGINE

Independent Claim 9:

A QR code-based transaction processing system with integrated commission splitting and loyalty point awarding, comprising:a) a QR code registry storing code configurations including type (loyalty, payment, checkin, promo, menu), discount parameters, base points value (default 25), points multiplier, and usage limits;b) a transaction processing edge function that validates incoming transaction requests using Zod schema validation for businessId (UUID), amount (positive, max 99,999.99), and optional fields;c) a business payment account lookup that retrieves the Stripe Connect account ID and verifies charges_enabled status before processing;d) a commission calculation component applying a platform commission rate (7.5%) to derive the application_fee_amount and business_amount: amountInCents = amount * 100, commission = amountInCents * 0.075, businessAmount = amountInCents - commission;e) a Stripe PaymentIntent creation with

application_fee_amount and transfer_data.destination for automatic split between platform and business;f) a QR scan recording component that logs the scan with qr_code_id, customer_id, business_id, timestamp, and geographic coordinates (GEOGRAPHY Point) for fraud detection;g) a loyalty points calculation awarding 10 points per dollar spent, recorded in the transaction metadata.

Dependent Claim 9.1: The system of Claim 9, wherein the geographic coordinates of QR scans are indexed for velocity fraud detection queries: CREATE INDEX idx_qr_scans_velocity ON qr_scans(customer_id, scan_date DESC);

Dependent Claim 9.2: The system of Claim 9, further comprising a commission recording function that atomically logs the platform commission for accounting and agent payout purposes.

CLAIM 10: AI-POWERED PERSONALIZED RECOMMENDATIONS

Independent Claim 10:

A machine learning-powered recommendation system generating personalized business suggestions, comprising:a) a user preferences table storing explicit preferences including preferred categories, locations, and notification settings;b) a business interactions tracker recording implicit signals including views, favorites, shares, reviews, and purchases with associated scores and timestamps;c) a candidate retrieval component that fetches verified businesses ordered by rating, filtering by active status;d) a context assembly mechanism that combines user preferences, recent interactions, and candidate businesses into a structured prompt for the recommendation model;e) an AI recommendation engine utilizing Google Gemini 2.5 Flash that analyzes the assembled context and generates exactly 5 recommendations with scores between 0.75 and 1.00;f) a recommendation persistence layer that stores generated recommendations with user_id, business_id, recommendation_score, recommendation_reason, and expiration timestamp (default 24 hours);g) a recommendation refresh trigger that regenerates recommendations when user interactions significantly change or upon expiration.

Dependent Claim 10.1: The system of Claim 10, wherein the AI model is instructed to consider: User's preferred categories and locations, Past interaction patterns, Business ratings and quality, Geographic proximity, Diversity of recommendations, Support for lesser-known but quality businesses alongside popular ones.

Dependent Claim 10.2: The system of Claim 10, wherein recommendation click-through is tracked with clicked boolean and clicked_at timestamp, enabling model feedback and recommendation quality measurement.

CLAIM 11: REAL-TIME VOICE AI BRIDGE ARCHITECTURE (Extended)

Independent Claim 11:

A WebSocket bridge architecture connecting client applications to external real-time AI voice services through an intermediary edge function, comprising:a) an HTTP-to-WebSocket upgrade handler in a serverless edge function that establishes a persistent connection with the client;b) a secondary WebSocket connection from the edge function to an external AI provider (OpenAI Realtime API) with authentication headers including API key

and beta feature flags;c) a session initialization sequence that waits for session.created event before transmitting session.update configuration with platform-specific parameters;d) persona injection through the session configuration containing: modalities specification (text, audio), detailed natural language instructions defining assistant personality, voice selection, audio format specification (pcm16), turn detection parameters, temperature setting for response variation;e) bidirectional message forwarding that relays all messages from client to AI and AI to client without modification, maintaining audio stream integrity;f) connection lifecycle management that closes both connections when either party disconnects, preventing orphaned connections.

Dependent Claim 11.1: The system of Claim 11, wherein the edge function is deployed to a globally distributed edge network, minimizing latency for voice interactions regardless of user geographic location.

Dependent Claim 11.2: The system of Claim 11, wherein the voice AI session includes input_audio_transcription configuration enabling Whisper-based speech-to-text for logging and analysis purposes.

CLAIM 12: AI TOOL REGISTRY FOR VOICE CONCIERGE

Independent Claim 12:

A structured tool registry enabling voice AI assistants to execute platform operations through validated tool invocations, comprising:a) a Zod schema definition specifying valid tool names as an enumerated set: search_businesses, get_business_details, check_availability, get_recommendations, check_coalition_points, start_booking;b) argument validation using z.record(z.unknown()) with runtime type checking for tool-specific parameters;c) a switch-based tool router that executes the appropriate database query or service call based on the validated tool_name;d) natural language response formatting that converts database results into human-readable messages suitable for voice synthesis;e) rate limiting protection with in-memory stores tracking request counts per IP address with configurable thresholds (default: 20 requests per 60 seconds);f) user context injection allowing tools to access user_id when available for personalized responses (e.g., coalition points balance).

Dependent Claim 12.1: The system of Claim 12, wherein the search_businesses tool supports optional filters: category: ilike pattern matching, city: ilike pattern matching, min_rating: greater-than-or-equal comparison, query: OR-combined ilike search across business_name, description, and category.

Dependent Claim 12.2: The system of Claim 12, wherein tool responses include both structured data and a pre-formatted message field for immediate voice synthesis.

CLAIM 13: ATOMIC FRAUD ALERT BATCH INSERTION

Independent Claim 13:

A PostgreSQL stored function for atomically inserting multiple AI-generated fraud alerts from JSONB input, comprising:a) a function signature accepting a JSONB array of alert objects;b) iteration over array elements using jsonb_array_elements();c) field extraction using the ->> operator for text fields and -> operator for nested JSONB objects;d) NULL handling using NULLIF() for optional fields that may be empty strings;e) type casting using ::UUID and ::DECIMAL for strongly-typed database columns;f) atomic insertion ensuring all alerts

succeed or all fail together, preventing partial state;g) return value of inserted row count using GET

DIAGNOSTICS for confirmation.

Dependent Claim 13.1: The function of Claim 13, wherein the implementation uses SECURITY DEFINER privileges with SET search_path = public for secure execution context.

CLAIM 14: ECONOMIC KARMA SCORING SYSTEM

Independent Claim 14:

A gamified contribution scoring system that tracks and rewards user participation in community economic circulation, comprising:a) an economic_karma score stored per user, representing their cumulative contribution to community wealth velocity;b) karma earning events triggered by: Transactions at platform businesses (karma proportional to transaction amount), Business referrals that result in registration, User referrals that result in activation, Review contributions, Coalition point circulation (earning and redeeming across different businesses), Participation in group challenges;c) karma multiplier effects where users with higher karma scores receive: Increased visibility in recommendation algorithms, Priority in customer support queues, Enhanced matching scores in B2B connections, Bonus points on transactions;d) karma decay mechanism that gradually reduces inactive karma over time, incentivizing continued platform engagement;e) karma leaderboards displaying top contributors within geographic regions and platform-wide;f) karma milestone achievements that unlock special badges and platform privileges at defined thresholds.

Dependent Claim 14.1: The system of Claim 14, wherein karma earning is calculated as: transaction_karma = transaction_amount * circulation_multiplier * 0.1, referral_karma = base_referral_karma * (1 + referred_user_tier_bonus), review_karma = review_length_factor * helpfulness_score.

Dependent Claim 14.2: The system of Claim 14, wherein karma decay applies a 5% monthly reduction to karma scores for users with no platform activity in the preceding 30 days, implemented via a scheduled PostgreSQL cron job executing: UPDATE profiles SET economic_karma = economic_karma * 0.95 WHERE last_activity_at < NOW() - INTERVAL '30 days'.

Dependent Claim 14.3: The system of Claim 14, wherein karma is designed as a "Cerebro-compatible feed" that can be exported as a data stream for integration with external investment analytics platforms, providing signal on community economic velocity patterns.

CLAIM 15: COMMUNITY FINANCE DIGITAL ESCROW AND DISTRIBUTION SYSTEM (SUSU)

Independent Claim 15:

A computer-implemented rotating savings and credit association (ROSCA) system for community-based collective savings, comprising:a) a circle creation mechanism with configurable parameters including: contribution_amount (decimal), cycle_length (weekly/biweekly/monthly), max_participants (integer with default 12), payout_order_type (sequential/lottery/auction), and minimum_credit_score threshold;b) a membership

registry implemented as a susu_memberships table tracking: participant position_in_cycle, scheduled_payout_date, contribution_status (pending/paid/late/defaulted), join_date, and UNIQUE constraint on (circle_id, user_id) preventing duplicate enrollment;c) a digital escrow holding mechanism using a dedicated escrow_balance column that aggregates contributions until payout conditions are met, with all balance modifications performed through atomic database transactions;d) an automatic distribution engine implemented as a PostgreSQL trigger that releases pooled funds to the designated recipient when: all contributions for the current cycle are received, the scheduled payout date is reached, and the recipient has no active defaults;e) a contribution enforcement system with: grace_period_days (default 3), late_fee_percentage (default 5%), automatic membership suspension after consecutive_defaults_allowed (default 2), and reputation impact on economic_karma score;f) an audit trail recording all contribution and payout events in a susu_transactions table with: transaction_type (contribution/payout/late_fee/refund), amount, participant_id, cycle_number, and timestamp for regulatory compliance;g) integration with Stripe Connect for automatic ACH contribution collection on scheduled dates with retry logic for failed payments.

Dependent Claim 15.1: The system of Claim 15, wherein susu_status progression follows the state machine: forming active paused completed dissolved, with state transitions validated by database constraints.

Dependent Claim 15.2: The system of Claim 15, wherein the payout calculation formula is: payout_amount = contribution_amount * (participant_count - 1) - platform_fee, where platform_fee = total_contributions * SUSU_PLATFORM_FEE_RATE (default 1.5%).

Dependent Claim 15.3: The system of Claim 15, wherein lottery-based payout ordering uses a cryptographically secure random selection (crypto.getRandomValues()) excluding members who have already received payouts in the current cycle.

Dependent Claim 15.4: The system of Claim 15, wherein auction-based payout ordering allows members to bid a discount percentage on their payout, with the highest bidder receiving priority and the discount distributed proportionally to other members as a bonus.

CLAIM 16: BIOMETRIC-SECURED GEOSPATIAL TRANSACTION VERIFICATION

Independent Claim 16:

A multi-factor transaction verification system combining geospatial fraud detection with biometric authentication for enhanced security, comprising:a) integration with native device biometric APIs through Capacitor 7.x plugins (@capacitor-community/biometric-auth) supporting FaceID on iOS, TouchID on iOS/MacOS, and fingerprint sensors on Android devices;b) a verification workflow requiring biometric confirmation for transactions exceeding a configurable threshold amount (default BIOMETRIC_THRESHOLD = $100), implemented as a pre-transaction middleware check;c) combination of biometric verification timestamp with geospatial coordinates to create a multi-factor fraud detection signal, stored as: biometric_verified_at TIMESTAMP, biometric_location GEOGRAPHY(Point), biometric_device_id TEXT;d) fallback authentication mechanisms including PIN entry and password verification when biometric hardware is unavailable or fails, with fallback events logged for security audit;e) biometric verification audit logging in a biometric_verifications table with columns: user_id, verification_type (faceid/touchid/fingerprint/pin/password), device_id, success BOOLEAN,

failure_reason TEXT, coordinates GEOGRAPHY(Point), and created_at timestamp;f) a velocity correlation component that compares biometric verification location with transaction location, flagging discrepancies exceeding 100 meters as potential device compromise or relay attacks.

Dependent Claim 16.1: The system of Claim 16, wherein biometric verification strengthens the geospatial velocity fraud detection (Claim 4) by confirming physical presence of the authenticated user at the transaction location, reducing false positives from shared devices or family account usage.

Dependent Claim 16.2: The system of Claim 16, wherein repeated biometric failures (default: 5 consecutive failures) trigger automatic account lock with notification to registered email and phone number.

Dependent Claim 16.3: The system of Claim 16, wherein biometric templates are never stored on platform servers; verification occurs entirely on-device using the Secure Enclave (iOS) or Trusted Execution Environment (Android), with only success/failure status transmitted to the server.

CLAIM 17: QR CODE ATOMIC CHECK-IN AND LOYALTY ACCRUAL SYSTEM

Independent Claim 17:

A computer-implemented atomic QR code scanning system that validates, records, and rewards customer check-ins in a single indivisible transaction, comprising:a) a QR code generation component that creates unique, cryptographically-signed QR codes for each business, encoding: business_id (UUID), code_type (loyalty/discount/payment), points_value, scan_limit (optional), expiration_date (optional), and HMAC signature for tamper detection;b) a real-time scanning interface using html5-qrcode library with camera permission handling, video stream management, and scan result parsing with format validation;c) an atomic scan processing pipeline implemented as a PostgreSQL transaction with SERIALIZABLE isolation level, executing in sequence: QR code validation active status check scan limit enforcement user authentication verification scan record insertion loyalty points accrual scan count increment;d) a scan limit enforcement mechanism that atomically checks current_scan_count against max_scan_limit before processing, preventing race conditions where multiple simultaneous scans could exceed limits using SELECT ... FOR UPDATE row locking;e) a loyalty points integration that automatically calculates and credits points based on: base_points_value * tier_multiplier (from Claim 3) * founding_member_bonus (from Claim 1), all within the same database transaction;f) a scan history registry storing: scan_id, qr_code_id, user_id, business_id, scanned_at timestamp, points_awarded, device_fingerprint, and geolocation coordinates for fraud correlation;g) a business analytics feed providing real-time scan metrics including: total scans, unique customers, points distributed, peak scanning hours, and geographic heatmaps of customer activity.

Dependent Claim 17.1: The system of Claim 17, wherein the atomic transaction rollback occurs if any step fails, ensuring no partial state where points are awarded without scan record or vice versa.

Dependent Claim 17.2: The system of Claim 17, wherein duplicate scan prevention uses a composite unique constraint on (qr_code_id, user_id, DATE(scanned_at)) preventing multiple scans from the same user on the same day for daily-limited codes.

Dependent Claim 17.3: The system of Claim 17, wherein the geolocation captured during scan is

correlated with the business location, flagging scans occurring more than 500 meters from the registered business address as potential fraud indicators feeding into Claim 4.

Dependent Claim 17.4: The system of Claim 17, wherein scan processing integrates with biometric verification (Claim 16) for high-value reward codes exceeding BIOMETRIC_THRESHOLD, requiring FaceID/TouchID confirmation before points accrual.

Technical Implementation:

```
-- Atomic scan processing with race condition prevention BEGIN; SET TRANSACTION ISOLATION LEVEL
SERIALIZABLE; -- Lock the QR code row to prevent concurrent modifications SELECT * FROM qr_codes
WHERE id = $1 FOR UPDATE; -- Validate and check limits atomically IF current_scan_count >=
max_scan_limit THEN RAISE EXCEPTION 'Scan limit exceeded'; END IF; -- Insert scan record INSERT INTO
qr_scans (qr_code_id, user_id, business_id, scanned_at, points_awarded) VALUES ($1, $2, $3, NOW(),
calculated_points); -- Update scan count UPDATE qr_codes SET current_scan_count = current_scan_count
+ 1 WHERE id = $1; -- Award loyalty points UPDATE coalition_points SET current_balance =
current_balance + calculated_points WHERE customer_id = $2; COMMIT;
```

CLAIM 18: COMMUNITY IMPACT ANALYTICS ENGINE

Independent Claim 18:

A comprehensive business analytics system that calculates, visualizes, and reports community economic impact metrics for platform-participating businesses, comprising:a) a transaction aggregation service that continuously collects and indexes all monetary transactions, QR scans, loyalty redemptions, and B2B interactions associated with each business, stored in an optimized analytics_events table with composite indexes on (business_id, event_type, created_at);b) a community impact calculator applying the CIRCULATION_MULTIPLIER constant (2.3x from Claim 2) to compute: raw_revenue (sum of transactions), community_impact (raw_revenue * 2.3), estimated_jobs_supported (community_impact / average_local_wage), tax_revenue_generated (community_impact * local_tax_rate);c) a customer retention analytics module tracking: unique_customers_count, repeat_customer_percentage, average_visit_frequency, customer_lifetime_value (CLV), and churn_rate calculated using cohort analysis over 30/60/90 day windows;d) a comparative "What If" analysis engine that models business performance scenarios including: revenue projection without platform participation, customer acquisition cost comparison vs. traditional advertising, and loyalty program ROI calculation;e) a coalition network contribution tracker showing: cross-referral revenue (customers acquired from other coalition businesses), points distributed to customers, points redeemed from other businesses, and net coalition balance;f) a geographic impact heatmap using Mapbox GL integration visualizing customer density, transaction concentration, and economic circulation patterns across zip codes and neighborhoods;g) an exportable report generator producing PDF and CSV formats for: tax documentation, investor presentations, grant applications, and SBA loan support materials.

Dependent Claim 18.1: The system of Claim 18, wherein the analytics dashboard displays real-time metrics updated via Supabase real-time subscriptions, with automatic refresh on new transaction events.

Dependent Claim 18.2: The system of Claim 18, wherein the "What If" analysis uses the formula: missed_community_impact = (current_transactions * CIRCULATION_MULTIPLIER) - current_transactions, showing businesses the community wealth they are enabling by participating.

Dependent Claim 18.3: The system of Claim 18, wherein QR code analytics (Claim 17) feed directly into the impact calculations, with scan-to-transaction conversion rates tracked as a key performance indicator.

Dependent Claim 18.4: The system of Claim 18, wherein sponsor attribution (Claim 2) links business performance metrics to corporate sponsor dashboards, enabling sponsors to track the specific businesses their investments support.

Dependent Claim 18.5: The system of Claim 18, wherein historical trend analysis uses time-series aggregation to show month-over-month and year-over-year growth in: revenue, customer count, coalition points, and community impact score.

CLAIM 19: CLOSED-LOOP PLATFORM WALLET ECOSYSTEM

Independent Claim 19:

A computer-implemented closed-loop digital wallet system that maximizes economic circulation within a minority business ecosystem, comprising:a) a platform wallet component maintaining a persistent digital balance for each user, stored in a wallet_balance field within the user's profile record, wherein said balance is denominated in the platform's base currency;b) an automatic payout crediting mechanism that, upon successful completion of a Susu rotational savings round (as described in Claim 15), automatically credits the payout amount (minus platform fees) directly to the recipient member's platform wallet balance rather than disbursing to external accounts;c) a wallet transaction ledger that records all wallet activity including: credits from Susu payouts, credits from refunds, debits from business purchases, debits from cash-out requests, with each transaction containing: transaction_id, user_id, amount, transaction_type, source_type, source_id, balance_after, and timestamp;d) a restricted redemption system wherein platform wallet balances are redeemable exclusively at verified Black-owned businesses registered on the platform, thereby ensuring funds circulate within the target economic ecosystem before potential exit;e) an atomic payment processing function that, when a user elects to pay with wallet balance at a participating business: validates sufficient balance, debits the wallet atomically using database transactions with SERIALIZABLE isolation, records the business transaction, and credits the business owner's receivables;f) an optional cash-out request system wherein users may request withdrawal of wallet funds to external bank accounts, subject to: minimum withdrawal thresholds, platform fee deduction, administrative review and approval, and processing delays that encourage in-ecosystem spending.

Dependent Claim 19.1: The system of Claim 19, wherein the platform wallet integration with Susu payouts creates a 'soft lock' that keeps community savings circulating within the Black business ecosystem, with analytics tracking the average number of circulation cycles before cash-out.

Dependent Claim 19.2: The system of Claim 19, wherein wallet payments at businesses award coalition loyalty points (as described in Claim 3) at the same rate as card payments, incentivizing wallet usage.

Dependent Claim 19.3: The system of Claim 19, wherein the cash-out request system applies a graduated fee structure: 2% for withdrawals under $500, 1.5% for $500-$2000, and 1% for amounts exceeding $2000, incentivizing larger balance accumulation and in-ecosystem spending.

Dependent Claim 19.4: The system of Claim 19, further comprising a wallet-to-wallet transfer capability allowing users to send funds to other platform users, facilitating peer-to-peer transactions within the

ecosystem.

Technical Implementation:

```
-- Wallet payment atomic transaction BEGIN; -- Validate and debit wallet UPDATE profiles SET
wallet_balance = wallet_balance - $amount WHERE id = $user_id AND wallet_balance >= $amount; --
Record wallet transaction INSERT INTO wallet_transactions (user_id, amount, transaction_type,
source_type, source_id, balance_after) VALUES ($user_id, -$amount, 'debit', 'business_purchase',
$business_id, (SELECT wallet_balance FROM profiles WHERE id = $user_id)); -- Create business
transaction record INSERT INTO transactions (business_id, customer_id, amount, payment_method) VALUES
($business_id, $user_id, $amount, 'wallet'); COMMIT;
```

CLAIM 20: ECONOMIC CIRCULATION VELOCITY ANALYTICS

Independent Claim 20:

A computer-implemented analytics system for measuring and optimizing the velocity of economic circulation within a closed-loop minority business ecosystem, comprising:a) a circulation event tracker that monitors all fund movements within the platform wallet ecosystem (Claim 19), categorizing each movement as: ecosystem_entry (Susu payout, external deposit), internal_circulation (business purchase, peer transfer), or ecosystem_exit (cash-out withdrawal);b) a velocity calculation engine that computes circulation velocity metrics including: average_cycles_before_exit (mean number of business transactions before cash-out), average_dwell_time (mean duration funds remain in ecosystem), circulation_multiplier_actual (empirically measured multiplier based on real transaction patterns);c) a comparative analytics module that compares the platform's measured circulation velocity against the theoretical maximum (funds never exit) and industry benchmarks (traditional 6-hour circulation for Black dollar vs. 20-day for majority communities);d) a business impact attribution system that tracks which businesses receive the most circulating funds, identifying high-velocity nodes that maximize economic retention within the ecosystem;e) a predictive modeling component using historical patterns to forecast: expected ecosystem retention rates, optimal incentive structures for increasing circulation velocity, and projected community wealth accumulation over time horizons of 1, 5, and 10 years;f) a real-time dashboard displaying circulation health metrics including: current_velocity_score (0-100), funds_in_circulation, projected_annual_impact, and comparison to ecosystem without platform intervention.

Dependent Claim 20.1: The system of Claim 20, wherein the circulation velocity metrics feed into the Community Impact Analytics Engine (Claim 18) to provide enhanced accuracy in economic impact calculations.

Dependent Claim 20.2: The system of Claim 20, wherein the velocity analytics identify 'leakage points' where funds exit the ecosystem, enabling targeted interventions such as special promotions at underutilized businesses.

Dependent Claim 20.3: The system of Claim 20, wherein the predictive modeling component uses machine learning trained on historical transaction data to optimize the timing and structure of Susu payout schedules for maximum ecosystem retention.

Dependent Claim 20.4: The system of Claim 20, further comprising a gamification layer that rewards users for maintaining high personal circulation velocity scores, with badges and bonus points for spending

wallet funds at multiple different businesses before cashing out.

Technical Implementation:

```
-- Calculate circulation velocity for a user SELECT user_id, COUNT(CASE WHEN transaction_type =
'debit' AND source_type = 'business_purchase' THEN 1 END) as business_transactions, COUNT(CASE WHEN
transaction_type = 'debit' AND source_type = 'cash_out' THEN 1 END) as cash_outs, CASE WHEN
COUNT(CASE WHEN transaction_type = 'debit' AND source_type = 'cash_out' THEN 1 END) > 0 THEN
COUNT(CASE WHEN transaction_type = 'debit' AND source_type = 'business_purchase' THEN 1 END)::float /
COUNT(CASE WHEN transaction_type = 'debit' AND source_type = 'cash_out' THEN 1 END)::float ELSE NULL
END as avg_cycles_before_exit FROM wallet_transactions WHERE user_id = $1 GROUP BY user_id;
```

CLAIM 21: PARTNER REFERRAL ATTRIBUTION AND REVENUE SHARING SYSTEM

Independent Claim 21:

A computer-implemented partner referral system that tracks attribution, calculates revenue shares, and manages multi-tier partner relationships, comprising:a) a partner registration mechanism that creates unique partner profiles with: partner_id (UUID), referral_code (unique alphanumeric string), referral_link (platform URL with embedded code), revenue_share_percentage (configurable per partner tier), and directory_name (human-readable slug);b) an attribution tracking component that captures referral source from: URL query parameters (?ref=CODE), cookie persistence (30-day attribution window), and manual code entry during registration, storing the referring_partner_id on the new user or business profile;c) a revenue share calculation engine that, upon each subscription payment from a referred user, computes: partner_commission = subscription_amount * partner_revenue_share_percentage, applying tier-based rates (Standard: 10%, Premium: 15%, Elite: 20%);d) a commission ledger maintaining all earned commissions with: commission_id, partner_id, referred_entity_id, referred_entity_type (user/business), subscription_id, commission_amount, commission_status (pending/approved/paid), and created_at timestamp;e) a payout request and processing system enabling partners to request withdrawal of accumulated commissions, with minimum payout thresholds, payment method selection (ACH, PayPal, Stripe), and administrative approval workflow;f) a partner analytics dashboard displaying: total referrals (users and businesses), active subscriptions from referrals, lifetime earnings, pending commissions, and conversion funnel metrics (clicks signups paid conversions).

Dependent Claim 21.1: The system of Claim 21, wherein the referral code generation uses a combination of partner initials and cryptographically random suffix, creating memorable yet unique codes (e.g., 'TB-7X9K').

Dependent Claim 21.2: The system of Claim 21, wherein the attribution window of 30 days persists across sessions using browser localStorage and first-party cookies, with the original referrer maintained even if the user encounters multiple referral links.

Dependent Claim 21.3: The system of Claim 21, wherein recurring subscription payments continue to generate partner commissions for the lifetime of the subscription, creating passive income streams for successful partners.

Dependent Claim 21.4: The system of Claim 21, further comprising a partner notification system that sends real-time alerts via email and in-app notifications when: new referral signs up, referral converts to

paid, commission is approved, and payout is processed.

Technical Implementation:

```
-- Partner commission calculation trigger CREATE OR REPLACE FUNCTION calculate_partner_commission()
RETURNS TRIGGER AS $$ DECLARE v_partner_id UUID; v_revenue_share DECIMAL; v_commission DECIMAL; BEGIN
-- Get referring partner from the subscriber's profile SELECT referring_partner_id INTO v_partner_id
FROM profiles WHERE id = NEW.user_id; IF v_partner_id IS NOT NULL THEN -- Get partner's revenue share
percentage SELECT revenue_share_percentage INTO v_revenue_share FROM partners WHERE id =
v_partner_id; -- Calculate commission v_commission := NEW.amount * (v_revenue_share / 100); -- Insert
commission record INSERT INTO partner_commissions (partner_id, subscription_id, amount, status)
VALUES (v_partner_id, NEW.id, v_commission, 'pending'); END IF; RETURN NEW; END; $$ LANGUAGE plpgsql;
```

CLAIM 22: FOUNDING PARTNER TIER SYSTEM WITH IMMUTABLE BENEFITS

Independent Claim 22:

A temporal partner incentive system that permanently designates early-enrolled partners as "Founding Partners" with enhanced lifetime benefits, comprising:a) a Founding Partner eligibility determination mechanism implemented as a database trigger that compares partner enrollment timestamp against a predetermined cutoff constant (September 1, 2026, 23:59:59 UTC);b) automatic assignment of is_founding_partner boolean flag to TRUE for partners enrolling before the cutoff, with founding_partner_since timestamp preserving the exact enrollment moment;c) enhanced revenue share rates for Founding Partners: 15% base rate (vs. 10% standard), 20% Premium tier (vs. 15%), 25% Elite tier (vs. 20%), representing a permanent 5% uplift across all tiers;d) immutability enforcement through database-level constraint triggers that prevent any modification, revocation, or downgrade of Founding Partner status through application interfaces, administrative actions, or direct database manipulation;e) exclusive Founding Partner benefits including: priority support queue, early access to new features, invitation to annual partner summit, distinguished badge display on partner profiles and directories;f) legacy protection ensuring Founding Partner status persists across: account migrations, platform version upgrades, tier changes, and any future partner program restructuring.

Dependent Claim 22.1: The system of Claim 22, wherein the Founding Partner designation integrates with the Temporal Founding Member Status System (Claim 1), providing both partner-level and user-level founding benefits for individuals who qualify for both.

Dependent Claim 22.2: The system of Claim 22, wherein Founding Partner status is prominently displayed in the public partner directory, providing social proof and prestige that incentivizes early enrollment.

Dependent Claim 22.3: The system of Claim 22, further comprising a Founding Partner certificate generation system that produces downloadable PDF certificates commemorating founding status for marketing and promotional use.

Technical Implementation:

```
-- Founding Partner status trigger CREATE OR REPLACE FUNCTION set_founding_partner_status() RETURNS
TRIGGER AS $$ DECLARE v_founding_cutoff CONSTANT TIMESTAMP WITH TIME ZONE := '2026-09-01T23:59:59Z';
BEGIN IF NEW.created_at < v_founding_cutoff THEN NEW.is_founding_partner := true;
NEW.founding_partner_since := NEW.created_at; NEW.revenue_share_percentage :=
NEW.revenue_share_percentage + 5; -- 5% founding bonus END IF; RETURN NEW; END; $$ LANGUAGE plpgsql;
```

CLAIM 23: EMBEDDABLE PARTNER WIDGET AND REFERRAL TRACKING SYSTEM

Independent Claim 23:

A system for generating and tracking embeddable referral widgets that partners can deploy on external websites, comprising:a) a widget generation component that produces embeddable HTML/JavaScript code snippets customized with: partner referral code, partner directory name, configurable widget styles (colors, sizes, layouts), and tracking parameters;b) multiple widget format options including: banner advertisements (various IAB standard sizes), call-to-action buttons, floating badges, inline text links, and QR codes that encode the partner's referral URL;c) a widget rendering engine that fetches real-time platform data including: current promotional offers, partner-specific messaging, and dynamic call-to-action text based on visitor context;d) cross-domain tracking integration using postMessage API and first-party proxy endpoints to maintain referral attribution when visitors navigate from partner websites to the platform;e) widget analytics tracking: impressions (widget views), clicks (interaction count), click-through rate, conversions (signups from widget), and attribution accuracy metrics;f) a widget management dashboard where partners can: generate new widgets, preview widget appearance, copy embed codes, view widget-specific analytics, and A/B test different widget configurations.

Dependent Claim 23.1: The system of Claim 23, wherein widgets are served from a CDN-cached edge function ensuring minimal load time impact on partner websites.

Dependent Claim 23.2: The system of Claim 23, wherein widget styles automatically adapt to light/dark mode based on the host page's color scheme using CSS media queries.

Dependent Claim 23.3: The system of Claim 23, wherein QR code widgets integrate with the QR Code Atomic Check-in System (Claim 17) to enable physical-world referral tracking at partner business locations.

CLAIM 24: PARTNER PERFORMANCE ANALYTICS AND LEADERBOARD SYSTEM

Independent Claim 24:

A comprehensive analytics and gamification system for tracking, comparing, and incentivizing partner performance, comprising:a) a real-time analytics engine that calculates partner performance metrics including: total referrals (lifetime and periodic), conversion rate (signups to paid), average revenue per referral, total earnings, and growth trajectory;b) a partner leaderboard system that ranks partners by configurable metrics (total referrals, conversion rate, earnings) with: global rankings, regional rankings, tier-specific rankings, and time-period rankings (weekly, monthly, all-time);c) a gamification layer that awards achievement badges for partner milestones: First Referral, 10 Referrals, 100 Referrals, $1,000 Earned, Top 10 Monthly, Conversion Champion (highest conversion rate), and Streak Master (consecutive months with referrals);d) a performance trend analysis component showing: month-over-month growth, seasonal patterns, funnel stage performance, and

comparison to cohort averages;e) an earnings projection calculator that estimates future earnings based on: current referral rate, historical conversion rates, average subscription value, and partner tier benefits;f) an export and reporting system generating downloadable reports (PDF, CSV, Excel) for: tax documentation, business planning, and performance reviews.

Dependent Claim 24.1: The system of Claim 24, wherein leaderboard rankings are updated in real-time using Supabase real-time subscriptions, enabling partners to see their position change immediately upon new referral activity.

Dependent Claim 24.2: The system of Claim 24, wherein achievement badges integrate with the Agent Badges System (Claim 7) for partners who are also sales agents, creating unified gamification across partner roles.

Dependent Claim 24.3: The system of Claim 24, further comprising a partner comparison tool allowing partners to anonymously benchmark their performance against aggregate statistics of similar partners (same tier, same tenure, same region).

CLAIM 25: PARTNER VETTING AND QUALITY ASSURANCE WORKFLOW

Independent Claim 25:

A systematic partner application and vetting process ensuring partner quality and platform alignment, comprising:a) a partner application form collecting: business/individual information, website/social media presence, audience demographics, marketing approach description, referral volume estimates, and acknowledgment of partner terms;b) an automated pre-screening component that validates: email domain authenticity, website existence and content quality, social media account verification, and absence from platform blacklists;c) an administrative review queue presenting pending applications with: applicant summary, automated screening results, risk indicators, and approve/reject/request-more-info action buttons;d) a conditional approval system supporting: immediate approval (meets all criteria), probationary approval (limited referral volume until proven), and tiered activation (unlocking features as trust is established);e) an ongoing quality monitoring system that tracks: referral quality (conversion rates, churn rates of referred users), compliance with partner guidelines, customer complaint correlation, and fraud indicators;f) an escalation and remediation workflow that: flags underperforming or problematic partners, initiates warning notifications, implements temporary restrictions, and processes termination for policy violations.

Dependent Claim 25.1: The system of Claim 25, wherein the automated pre-screening integrates with external verification services to validate business registration status and tax identification numbers.

Dependent Claim 25.2: The system of Claim 25, wherein the quality monitoring system correlates partner referral patterns with the Geospatial Velocity Fraud Detection System (Claim 4) to identify partners potentially involved in fraudulent referral schemes.

Dependent Claim 25.3: The system of Claim 25, further comprising a partner feedback loop where referred users can rate their referral experience, providing quality signals for partner evaluation.

CLAIM 26: MULTI-TIER PARTNER AFFILIATE NETWORK

Independent Claim 26:

A hierarchical partner network enabling multi-level referral relationships and override commissions, comprising:a) a partner recruitment tracking system that records when one partner refers another partner to join the program, establishing a recruited_by_partner_id relationship;b) a recruitment bonus mechanism that awards a one-time bonus to recruiting partners when their recruited partners achieve activation milestones (e.g., first paid referral);c) an override commission structure wherein recruiting partners receive a percentage of their recruited partners' earnings: 5% of direct recruit earnings, 2% of second-level recruit earnings, capped at two levels to prevent pyramid dynamics;d) a downline visualization component displaying: recruited partner tree, downline performance metrics, override earnings by level, and inactive recruit indicators;e) anti-pyramid safeguards including: requirement that override earnings never exceed direct referral earnings, caps on total override percentage, and prohibition on purchase requirements for partner enrollment;f) a compliance monitoring system that ensures the partner network adheres to FTC guidelines for multi-level marketing programs, with automatic flagging of structures that approach regulatory boundaries.

Dependent Claim 26.1: The system of Claim 26, wherein the override commission calculation occurs automatically upon each commission payment to recruited partners, crediting recruiting partners in the same payment batch.

Dependent Claim 26.2: The system of Claim 26, wherein the partner tree visualization integrates with Partner Performance Analytics (Claim 24) to show aggregate downline performance metrics alongside individual partner data.

Dependent Claim 26.3: The system of Claim 26, further comprising a 'promote to partner' feature allowing existing users or sales agents to upgrade to partner status while maintaining their referral relationships from the Agent Network (Claim 7).

Technical Implementation:

```
-- Calculate override commissions for partner hierarchy CREATE OR REPLACE FUNCTION
calculate_override_commissions() RETURNS TRIGGER AS $$ DECLARE v_level1_partner_id UUID;
v_level2_partner_id UUID; BEGIN -- Get direct recruiter (level 1) SELECT recruited_by_partner_id INTO
v_level1_partner_id FROM partners WHERE id = NEW.partner_id; IF v_level1_partner_id IS NOT NULL THEN
-- Award 5% override to level 1 INSERT INTO partner_overrides (recruiter_id, earner_id,
commission_id, override_rate, override_amount) VALUES (v_level1_partner_id, NEW.partner_id, NEW.id,
0.05, NEW.amount * 0.05); -- Get level 2 recruiter SELECT recruited_by_partner_id INTO
v_level2_partner_id FROM partners WHERE id = v_level1_partner_id; IF v_level2_partner_id IS NOT NULL
THEN -- Award 2% override to level 2 INSERT INTO partner_overrides (recruiter_id, earner_id,
commission_id, override_rate, override_amount) VALUES (v_level2_partner_id, NEW.partner_id, NEW.id,
0.02, NEW.amount * 0.02); END IF; END IF; RETURN NEW; END; $$ LANGUAGE plpgsql;
```

CLAIM 27: AUTOMATED PARTNER MARKETING TOOLKIT WITH DYNAMIC ATTRIBUTION INJECTION

Independent Claim 27:

A computer-implemented system for generating, personalizing, and distributing marketing materials that automatically embed partner attribution data across multiple channels and formats, comprising:a) a marketing content generation engine that produces pre-formatted, brand-compliant promotional materials including: email templates with subject lines and body copy, social media posts optimized for each platform (LinkedIn, Twitter/X, Facebook, Instagram), and website embed code snippets;b) a dynamic attribution injection mechanism that automatically personalizes all generated content with partner-specific data including: referral code, referral URL, partner directory name, custom partner messaging, and partner contact information, using template variable substitution;c) an ROI messaging calculator that computes and inserts personalized value propositions based on platform economics: monthly cost savings ($700 value $100 cost = $600 savings), annual savings projection, and comparison to traditional alternatives;d) a multi-format collateral generator producing: printable PDF flyers with embedded QR codes linking to partner referral URLs, professional one-pagers for in-person distribution, and digital banner advertisements in standard IAB sizes;e) a talking points and script generator that creates: phone/conversation talking points with objection handling, video script templates for partner-created promotional content, and success story templates with fill-in-the-blank formatting for testimonial collection;f) a digital welcome kit bundler that packages: partner onboarding guide, platform overview materials, target audience profiles, and promotional best practices into a downloadable resource package;g) a content analytics layer tracking: material downloads, share events, click-through rates from generated materials, and conversion attribution to specific content pieces.

Dependent Claim 27.1: The system of Claim 27, wherein the email template generator produces platform-specific formatted content for major email service providers (Gmail, Outlook, Apple Mail) with responsive design for mobile viewing.

Dependent Claim 27.2: The system of Claim 27, wherein social media posts include platform-optimized hashtags, character counts within platform limits, and image size recommendations for maximum engagement.

Dependent Claim 27.3: The system of Claim 27, wherein printable flyers integrate with the QR Code System (Claim 17) to enable scan-based attribution tracking from physical marketing materials.

Dependent Claim 27.4: The system of Claim 27, wherein the talking points generator uses AI (similar to Claim 5) to customize messaging based on target industry, audience demographics, and partner's previous successful conversion patterns.

Dependent Claim 27.5: The system of Claim 27, wherein the success story template integrates with the Community Impact Analytics Engine (Claim 18) to auto-populate quantified impact metrics for referred businesses.

Dependent Claim 27.6: The system of Claim 27, further comprising a content versioning system that tracks material updates, notifies partners of new content availability, and maintains archive of previously generated materials for compliance purposes.

Technical Implementation:

```
// Partner Marketing Content Generation interface PartnerMarketingContent { emailTemplates:
EmailTemplate[]; socialPosts: SocialPost[]; printableMaterials: PrintableMaterial[]; talkingPoints:
TalkingPoint[]; videoScripts: VideoScript[]; } function generatePartnerContent(partner: Partner):
PartnerMarketingContent { const attribution = { referralCode: partner.referral_code, referralUrl:
`https://platform.com/?ref=${partner.referral_code}`, directoryName: partner.directory_name,
partnerName: partner.display_name }; const roiCalculation = { monthlyValue: 700, monthlyCost: 100,
monthlySavings: 600, annualSavings: 7200, roiMultiplier: 7 // $700/$100 = 7x value }; return {
emailTemplates: generateEmailTemplates(attribution, roiCalculation), socialPosts:
generateSocialPosts(attribution, roiCalculation), printableMaterials: generatePrintables(attribution,
roiCalculation), talkingPoints: generateTalkingPoints(roiCalculation), videoScripts:
generateVideoScripts(attribution, roiCalculation) }; }
```

KEY PROTECTED CONSTANTS

Constant	Value	Location	Claim
CIRCULATION_MULTIPLIER	2.3	calculate-sponsor-impact	Claim 2
REACH_MULTIPLIER	10	calculate-sponsor-impact	Claim 2
COMMISSION_RATE	7.5%	process-qr-transaction	Claim 9
POINTS_PER_DOLLAR	10	process-qr-transaction	Claim 9
FOUNDING_CUTOFF	2026-09-01T23:59:59Z	Database migration	Claim 1, 22
FRAUD_VELOCITY_THRESHOLD	600 mph	detect-fraud	Claim 4
RACE_CONDITION_WINDOW	500ms	detect-fraud transaction queue	Claim 4.5
KARMA_DECAY_RATE	5% monthly	economic-karma cron job	Claim 14
SUSU_PLATFORM_FEE	1.5%	susu-payout-calculation	Claim 15
BIOMETRIC_THRESHOLD	$100	transaction verification middleware	Claim 16
QR_SCAN_DAILY_LIMIT	1 per user per code	qr-scan-validation	Claim 17
SCAN_PROXIMITY_THRESHOLD	500 meters	qr-fraud-detection	Claim 17.3
AVERAGE_LOCAL_WAGE	$45,000/year	jobs-impact-calculation	Claim 18
WALLET_MIN_CASHOUT	$10	withdrawal-requests	Claim 19
WALLET_CASHOUT_FEE	2%	withdrawal-requests	Claim 19.3

Constant	Value	Location	Claim
PARTNER_ATTRIBUTION_WINDOW	30 days	partner-referral-tracking	Claim 21
PARTNER_STANDARD_RATE	10%	partner-commission-calculation	Claim 21
PARTNER_FOUNDING_BONUS	5%	founding-partner-trigger	Claim 22
PARTNER_OVERRIDE_L1	5%	partner-override-calculation	Claim 26
PARTNER_OVERRIDE_L2	2%	partner-override-calculation	Claim 26
PARTNER_ROI_VALUE	$700/mo	marketing-toolkit-generator	Claim 27
PARTNER_ROI_COST	$100/mo	marketing-toolkit-generator	Claim 27

TECHNOLOGY EQUIVALENTS MATRIX

Technology	Covered Equivalents
PostgreSQL database triggers	Any relational database trigger system, MongoDB change streams, Firestore triggers, DynamoDB streams, smart contracts, event-driven serverless functions
2.3x circulation multiplier	Any multiplier constant > 1.0 applied to represent community economic circulation, whether 2.3, 2.0, 2.5, or any empirically-derived value
Supabase Edge Functions	AWS Lambda, Google Cloud Functions, Azure Functions, Cloudflare Workers, Vercel Edge Functions, Deno Deploy, Netlify Functions, any FaaS platform
OpenAI Realtime API	Any real-time voice AI provider including ElevenLabs, Deepgram, AssemblyAI, Amazon Transcribe/Polly, Google Cloud Speech-to-Text, Azure Speech Services
Stripe Connect	Any multi-party payment splitting system including PayPal Commerce, Adyen, Square, Plaid, any payment processor supporting application fees and destination charges
Google Gemini 2.5 Flash	Any large language model including OpenAI GPT-4/GPT-5, Anthropic Claude, Meta Llama, Mistral, Cohere, any transformer-based AI model
WebSocket connections	Any persistent bidirectional communication protocol including Server-Sent Events, WebRTC data channels, gRPC streaming, MQTT

Technology	Covered Equivalents
Zod schema validation	Any runtime type validation library including Yup, Joi, io-ts, Valibot, AJV, any JSON Schema validator
Coalition Loyalty Network (Claim 3)	Blockchain/Distributed Ledgers (Ethereum, Solana, Polygon, etc.), Smart Contracts, Decentralized Autonomous Organizations (DAOs), decentralized identity systems, any token-based loyalty system whether centralized or decentralized, NFT-based membership tokens
Susu/ROSCA System (Claim 15)	Any rotating savings mechanism including blockchain-based savings circles, smart contract escrow systems, DeFi lending pools with rotating distribution, mobile money chit funds, any digital implementation of traditional community savings
Biometric Authentication (Claim 16)	Any biometric verification method including FaceID, TouchID, fingerprint sensors, iris scanning, voice recognition, behavioral biometrics, any hardware security module (HSM) based authentication
Closed-Loop Wallet (Claim 19)	Any digital wallet system with restricted redemption including prepaid cards, store credit systems, loyalty currency wallets, stablecoin ecosystems, any closed-loop payment network
Circulation Velocity Analytics (Claim 20)	Any economic flow tracking system including blockchain transaction analysis, money velocity metrics, working capital cycle analysis, any system measuring fund circulation frequency within a defined ecosystem
Partner Referral Attribution (Claims 21-26)	Any affiliate tracking system including cookie-based attribution, UTM parameter tracking, referral code systems, influencer marketing platforms, affiliate networks, multi-level marketing software
Partner Marketing Toolkit (Claim 27)	Any marketing automation system including HubSpot, Marketo, Mailchimp, brand asset management platforms, content generation engines, AI-powered marketing copy generators

PCT PRESERVATION LANGUAGE

This provisional application expressly preserves priority rights for international filing under the Patent Cooperation Treaty (PCT) within 12 months of the filing date. The inventor reserves the right to file corresponding applications in all PCT member states.

Target Countries for PCT Filing

European Patent Office (EPO), Japan Patent Office (JPO), Korean Intellectual Property Office (KIPO), China National Intellectual Property Administration (CNIPA), India Patent Office, Brazil National

Institute of Industrial Property (INPI), Canadian Intellectual Property Office (CIPO), Australian IP Office, United Kingdom Intellectual Property Office (UKIPO), African Regional Intellectual Property Organization (ARIPO), Nigeria (particularly relevant for diaspora connection), Ghana (growing tech hub), Kenya (East African tech leader), South Africa (continental economic leader)

LEGAL SAFEGUARD CLAUSES

Broad Interpretation

- Cloud-based servers (AWS, Google Cloud, Azure, or any infrastructure provider)
- Distributed ledgers (blockchain, DAG-based systems, or other decentralized technologies)
- Localized edge computing or hybrid cloud-edge architectures
- Any combination of relational databases, NoSQL databases, or graph databases
- Any programming languages, frameworks, or runtime environments
- Native mobile applications, progressive web applications, or hybrid solutions
- Alternative AI/ML providers or self-hosted machine learning models

Doctrine of Equivalents

Pursuant to the doctrine of equivalents established in Graver Tank & Mfg. Co. v. Linde Air Products Co., 339 U.S. 605 (1950), and Warner-Jenkinson Co. v. Hilton Davis Chemical Co., 520 U.S. 17 (1997), the claims extend to any system, method, or apparatus that performs substantially the same function in substantially the same way to achieve substantially the same result as the inventions described herein.

INVENTOR DECLARATION

I, Thomas D. Bowling, hereby declare that:

1. I am the original inventor of the subject matter claimed in this provisional patent application.
2. I have reviewed and understand the contents of this application.
3. I acknowledge my duty to disclose to the USPTO all information known to me to be material to patentability.
4. I understand that willful false statements may jeopardize the validity of the application or any patent issued thereon.

Signature: _____ Date: _____ Name: Thomas D. Bowling

FILING CHECKLIST

Document	Status
Specification (this document)	READY
Formal Claims (27 independent + 56 dependent)	READY
System Diagrams	READY
Abstract	READY
Inventor Declaration	PENDING SIGNATURE
Application Data Sheet (USPTO Form PTO/AIA/14)	PENDING
Filing Fee Payment	PENDING

Filing Fees (Provisional Application)

Entity Type	Fee
Micro Entity (recommended)	$80
Small Entity	$160
Large Entity	$320