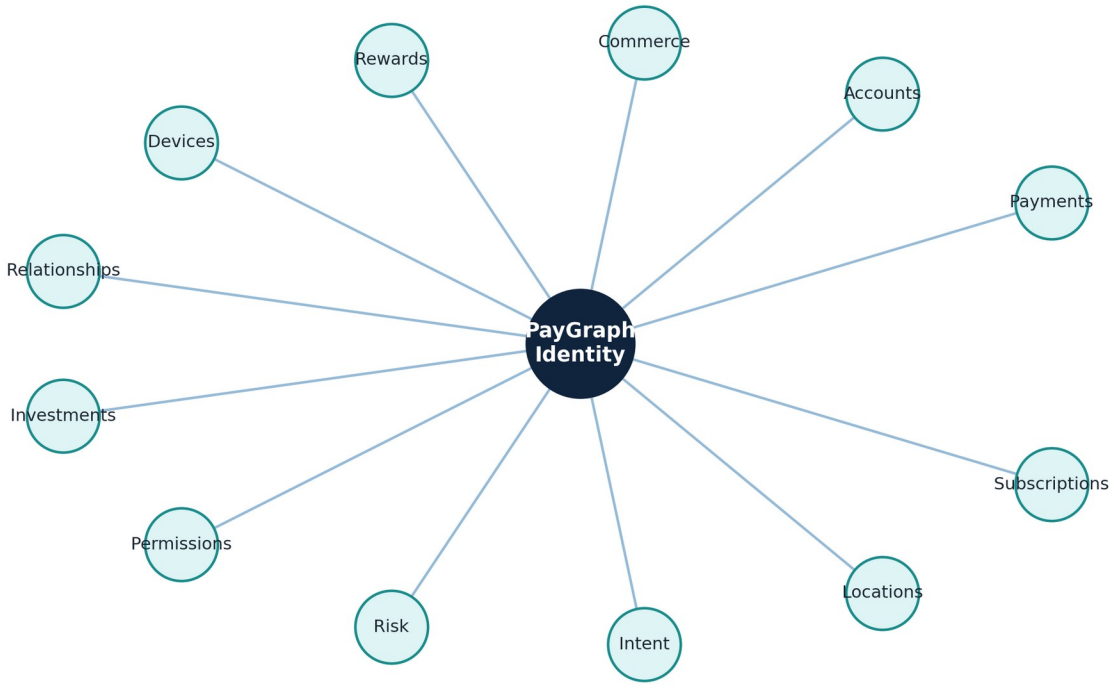
PAYGRAPH

The Financial Intelligence Layer

A strategic and technical foundation for identity-centric payments, contextual intelligence, secure personalization, and developer innovation.



A person-centered financial intelligence model connecting permissioned events, relationships, context, and decisions.

CONCEIVED BY DAVID STREBINGER • 2006

Published by mCards Inc. | Working Draft v0.1 | July 2026

Publication Note

This document is the first integrated working edition of the PayGraph flagship white paper. It establishes the narrative, vocabulary, architecture, governance principles, commercial model, and publication design system that will govern the broader PayGraph technical library.

> **PURPOSE OF THIS EDITION**
>
> To create a coherent, market-ready foundation that can be reviewed by mCards leadership, technical architects, privacy and security specialists, financial-institution partners, networks, prospective developers, and strategic investors before formal publication.

The document intentionally separates established platform capabilities, architectural requirements, proposed capabilities, and long-term ecosystem vision. Statements describing planned or future functionality should not be interpreted as representations that every capability is currently deployed in production.

The historical statement that David Strebinger conceived the PayGraph concept in 2006 is presented as an origin narrative supplied by the originator. Prior to public publication, supporting historical materials, dated notes, presentations, domains, filings, correspondence, or other archival records should be assembled where available.

PART I

The Thesis

Why the financial world needs an identity-centered intelligence layer.

Executive Summary

Financial information is organized around institutions, accounts, processors, programs, and applications. Consumers, however, experience money as one connected life. Their checking accounts, credit products, investments, loyalty balances, subscriptions, merchants, family obligations, employers, devices, and financial goals interact continuously—even though the systems that serve them rarely share a coherent understanding of those relationships.

The PayGraph is designed to close that gap. It is a permissioned financial intelligence layer that connects identities, events, relationships, context, policies, and derived insights into a continuously evolving graph. The graph does not replace banks, payment networks, ledgers, processors, or wallets. It sits above and between them, converting fragmented financial activity into secure, purpose-limited intelligence that can improve decisions and experiences.

At the center of the PayGraph is the individual—not the account. Every connected event can strengthen an understanding of what is normal, relevant, permitted, valuable, or suspicious for that identity. A transaction becomes more than an amount and merchant. It can be interpreted in relation to the user's usual behavior, funding preferences, household relationships, device trust, travel context, rewards opportunities, spending controls, and current intent.

This model creates opportunities across real-time authorization, fraud detection, security, personalization, rewards, marketing, sales, service, financial health, lending support, and AI-assisted commerce. It also creates material responsibilities. A commercially credible PayGraph must be designed around explicit permission, data minimization, tokenization, purpose limitation, revocation, policy enforcement, developer isolation, explainability, auditability, and restrictions on harmful or discriminatory uses.

mCards and Neobankify provide a practical path toward this model because they are designed to connect multiple funding sources, payment programs, controls, and experiences through a common orchestration layer. Combined with modern graph infrastructure, streaming event systems, privacy-enhancing technologies, and AI reasoning, these capabilities make the original

PayGraph vision technically and commercially achievable in a way that was not practical when the concept was first conceived in 2006.

> **THE CENTRAL PROPOSITION**
>
> The PayGraph converts permissioned financial activity into contextual intelligence—allowing software to understand not only what happened, but who it relates to, under what permissions, in what context, and what action is most appropriate next.

The PayGraph in One Page

Dimension	Definition
Core object	A living, permissioned graph centered on a verified identity.
Inputs	Financial events, account connections, commerce signals, relationships, devices, permissions, and contextual data.
Processing	Tokenization, normalization, identity resolution, graph enrichment, policy evaluation, scoring, inference, and audit.
Outputs	Decisions, confidence scores, recommendations, alerts, offers, explanations, and developer-accessible claims.
Primary value	More secure, relevant, efficient, and personalized financial experiences.
Trust model	Purpose-limited access, consumer controls, least privilege, zero trust, and separation of raw data from derived intelligence.
Commercial model	Embedded capability for mCards and Neobankify plus enterprise services and developer APIs.

Dimension	Definition
Long-term vision	A standardized financial intelligence layer that supports interoperable applications and agents.

1. The Origin of the PayGraph

In 2006, the emerging Social Graph demonstrated that relationships could become a foundational computing primitive. Rather than treating each profile, message, photo, or interaction as an isolated record, social platforms could organize information around people and the connections among them. David Strebinger recognized that financial services lacked an equivalent model.

The financial world contained equally powerful relationships: people to accounts, people to merchants, people to employers, people to family members, people to investments, merchants to categories, purchases to life events, and payments to shared social or economic contexts. Yet these relationships were trapped inside separate systems. A bank could see one account. A network could see one stream of card activity. A merchant could see its own customers. A loyalty program could see points. None could understand the consumer's broader financial identity with appropriate permission and context.

The PayGraph concept emerged from the idea that an individual's financial life could be mapped as a growing network of identities, activities, relationships, and meaning. Over time, the graph could learn the difference between a person and their peers, between ordinary and anomalous behavior, between useful relevance and unwanted intrusion, and between a generic financial interaction and the next best action for a specific individual.

The original barriers were practical. Consumers lacked a simple way to connect their entire wallet. Financial APIs were limited. Cloud infrastructure was immature. Graph databases were specialized and expensive. Real-time event processing was difficult. Tokenization, consent management, identity resolution, and AI reasoning were not available as composable building blocks. The vision required an ecosystem that did not yet exist.

That ecosystem now exists. Open-finance connectivity, modern issuing and processing, cloud-scale graph stores, streaming infrastructure, confidential computing, privacy engineering, machine learning, large language models, and agentic software create the components needed to build the PayGraph responsibly. mCards adds another critical element: a consumer-facing and transaction-facing mechanism through which multiple payment and funding modalities can be connected and orchestrated.

2. The Fragmented Financial Identity

Most financial systems still treat the account as the primary unit of understanding. This design is natural for ledgers and regulatory reporting, but inadequate for intelligence. A person may maintain several bank accounts, credit cards, installment products, digital wallets, loyalty programs, investment accounts, employer benefits, subscriptions, and family obligations. Each system sees only the slice it administers.

Fragmentation creates predictable failures. Fraud systems generate false positives because they cannot distinguish unusual behavior from a legitimate life event. Marketing systems promote products the consumer already owns or does not need. Rewards remain stranded because programs cannot optimize across the wallet. Customer service teams respond after problems occur instead of anticipating them. Developers rebuild narrow profiles inside every application. Consumers repeatedly supply the same information without gaining a more coherent experience.

The problem is not simply that data is unavailable. The deeper problem is that data lacks a shared identity model, relationship model, consent model, and contextual interpretation layer. Combining records in a warehouse does not automatically create a graph. The PayGraph therefore requires more than aggregation. It requires semantics: a precise understanding of what each object represents, how it relates to other objects, who is allowed to use it, for what purpose, and with what level of confidence.

> **ACCOUNT-CENTRIC VS. IDENTITY-CENTRIC**
>
> Account-centric systems ask: “What happened in this account?” Identity-centric systems ask: “What does this event mean for this person, given the connected relationships, permissions, context, and history they have chosen to make available?”

3. Why Now

The PayGraph becomes commercially plausible because several technology curves have converged. No single innovation is sufficient. Together, they create a new architectural possibility.

Enabler	Contribution to PayGraph
Open finance and connectivity	Permissioned access to accounts, balances, transactions, identity attributes, and financial relationships.
Modern issuing and orchestration	A programmable point of interaction across cards, wallets, funding sources, controls, and transaction decisions.
Streaming event infrastructure	Low-latency capture, normalization, and enrichment of high-volume financial events.
Graph databases	Native representation of identities, relationships, paths, communities, and evolving context.
Tokenization and privacy engineering	Separation of raw identifiers from graph-level references and developer-accessible claims.
AI and LLMs	Natural-language interfaces, semantic retrieval, agent orchestration, explanation, and synthesis across structured graph context.
Cloud security and observability	Policy enforcement, encrypted processing, event lineage, monitoring, and elastic scale.
Developer platforms	Standard APIs, SDKs, sandboxes, and marketplaces that allow new applications to be built on common infrastructure.

PART II

The Model

How identities, events, relationships, context, and permissions become a living financial knowledge base.

4. What a PayGraph Is

A PayGraph is a dynamic, identity-centered knowledge graph that represents permissioned relationships across payments, accounts, commerce, money movement, investments, rewards, devices, organizations, policies, and contextual signals. It combines a factual graph—what is known or observed—with a derived intelligence layer—what may be inferred, scored, recommended, or predicted.

The PayGraph is not a replacement for a transaction ledger. Ledgers establish authoritative balances and postings. The PayGraph references and interprets events around those systems. It is also not a universal repository in which every raw financial record must be copied. A mature design may use federated access, secure enclaves, tokenized references, feature stores, and time-bound retrieval to minimize centralization.

Each PayGraph belongs conceptually to an identity or authorized entity. The graph can include direct relationships, such as a consumer owning an account, and indirect relationships, such as a consumer transacting at a merchant associated with a location, category, offer, and social or household context. Relationships are time-aware, permission-aware, and confidence-scored. They can expire, be revoked, or change meaning.

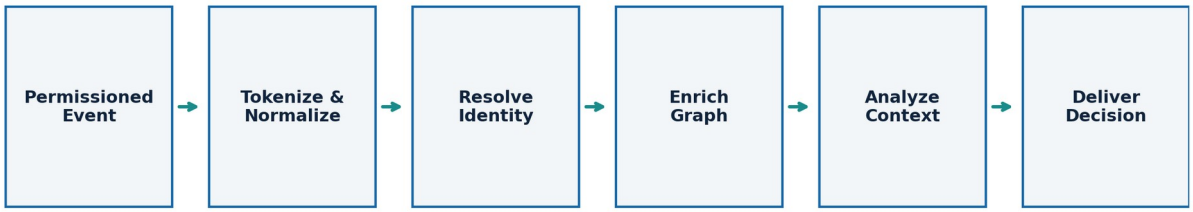


Figure 2. A permissioned event moves through tokenization, identity resolution, graph enrichment, contextual analysis, and a governed decision.

5. The Fundamental Objects

Object	Purpose	Examples
Identity	Represents a verified person or authorized organization.	Consumer, child, employee, merchant, employer, institution.
Node	Represents an entity or concept in the graph.	Account, card, merchant, device, reward, location, goal.
Edge	Represents a typed relationship between nodes.	Owns, funds, trusts, works for, purchased from, authorized by.
Event	Represents something that occurred at a point in time.	Authorization, purchase, transfer, login, consent grant, reward redemption.
Signal	Represents observed or supplied context.	Device reputation, velocity, location, balance range, merchant category.
Claim	Represents a governed statement usable by another service.	Device trusted, user traveling, category preference, account verified.
Policy	Represents permissions and restrictions.	Purpose, retention, age, geography, spending rule, developer scope.
Score	Represents a bounded assessment with lineage.	Transaction confidence, anomaly, identity confidence, offer relevance.
Insight	Represents a higher-order interpretation.	Likely subscription, duplicate expense, emerging travel intent.
Action	Represents an approved output.	Approve, step up, recommend, suppress, notify, route, explain.

6. Data Domains

The PayGraph should use a modular domain model. Not every implementation or user needs every domain. Each domain can be connected independently, governed by its own purpose, retention, and access policy.

Domain	Representative information
Identity	Verified names, aliases, identity tokens, credentials, age bands, verification methods, authorized roles.
Financial accounts	Account tokens, institution relationships, account type, ownership, status, balance ranges, available credit where authorized.
Payments	Authorizations, clearing events, amounts, currencies, merchants, channels, funding sources, outcomes, disputes.
Commerce	Products, receipts, merchants, categories, subscriptions, returns, warranties, invoices, purchase intent.
Rewards	Programs, balances, earn rules, redemption options, partner relationships, optimization opportunities.
Investments and assets	Account relationships, asset classes, contributions, goals, risk bands, liquidity needs where authorized.
Relationships	Family, household, employer, team, school, organization, trusted payee, authorized delegate.
Devices and channels	Device tokens, app instances, authentication methods, browser sessions, wallet credentials, vehicle or wearable endpoints.
Location and context	Merchant location, coarse user context, travel state, time, channel, environmental or event context where relevant and permitted.

Domain	Representative information
Permissions and consent	Data source, purpose, recipient, duration, scope, revocation state, legal basis, policy version.
Behavior and preference	Frequency, sequence, seasonality, payment preference, merchant affinity, budget patterns, chosen communication preferences.
Risk and trust	Identity confidence, device trust, anomaly signals, fraud outcomes, chargebacks, velocity, policy exceptions.
Communication	Offers shown, messages delivered, response, suppression, preferred channel, service interactions.
Goals and intent	Savings goals, upcoming travel, home purchase, education, financial health objectives, declared preferences.

7. The Personalized Financial Knowledge Base

The PayGraph becomes more useful as it develops a longitudinal understanding of an identity. This does not mean accumulating every possible record indefinitely. It means maintaining the minimum durable knowledge required to support approved services while preserving lineage and user control.

The knowledge base can distinguish facts from observations and inferences. A connected account relationship is a fact supplied by an authorized source. A purchase is an observed event. A preference for a merchant category is a derived pattern. A prediction that the user may be planning travel is an inference. These classes should never be collapsed into one undifferentiated profile because their accuracy, sensitivity, explainability, and permitted uses differ.

Knowledge class	Example	Required treatment
Declared	The user states a goal to save for a home.	Retain source and user-edit capability.

Knowledge class	Example	Required treatment
Verified	An institution verifies account ownership.	Preserve verifier, time, and assurance level.
Observed	A payment authorization occurs.	Record event lineage and applicable retention.
Derived	Dining spend is higher on business-travel weeks.	Record transformation, model, and confidence.
Inferred	The user may be planning an international trip.	Label as inference; restrict sensitive use; allow correction.
Aggregated	Anonymous cohort shows a seasonal pattern.	Apply de-identification and minimum cohort thresholds.

> **A CRITICAL DESIGN RULE**
>
> The PayGraph must never present an inference as a fact. Every derived attribute should retain provenance, confidence, freshness, policy, and a path to correction or deletion where applicable.

PART III

The Intelligence

From isolated records to contextual decisions, recommendations, and trusted automation.

8. Transaction Confidence

Transaction Confidence is one of the first high-value intelligence services the PayGraph can support. Payment authorization often occurs under strict latency limits. The decision engine must determine whether a transaction should be approved, routed, challenged, deferred for additional information, or declined. Traditional rules evaluate a narrow set of attributes. The PayGraph can add identity-level context without requiring every upstream system to query every source in real time.

A Transaction Confidence score can combine precomputed graph features with real-time event signals. Examples include device trust, merchant familiarity, transaction velocity, funding-source health, travel context, behavioral consistency, relationship permissions, prior disputes, balance confidence, and the strength of identity resolution. The score should not be a black-box verdict. It should produce reason codes, contribution ranges, uncertainty, freshness, and policy constraints.

Layer	Examples	Latency approach
Precomputed identity features	Usual merchants, device set, travel patterns, recurring obligations, funding preferences.	Continuously updated; retrieved from low-latency feature store.
Real-time event features	Amount, merchant, channel, location, timestamp, device, velocity.	Calculated during authorization.
Graph relationship features	Known merchant, trusted beneficiary, family permission, employer allowance.	Cached graph claims or bounded-depth traversal.

Layer	Examples	Latency approach
External risk signals	Network intelligence, sanctions screening, device reputation, institution response.	Parallel calls with strict timeouts and fallbacks.
Policy evaluation	Program rules, spending controls, geography, age, purpose.	Deterministic rules before or alongside model scoring.
Decision output	Approve, route, step-up, hold, decline, explain.	Completed within the applicable authorization budget.

The graph also supports simulation and back-testing. Historical events can be replayed against model and policy versions to estimate approval rates, false-positive reductions, fraud capture, customer impact, funding-route success, latency, and operational review volume. This capability is essential for safely tuning the system before production rollout.

9. Fraud Detection and Security

Fraud rarely appears as one obviously invalid transaction. It emerges as a pattern across identities, devices, merchants, accounts, beneficiaries, locations, and time. Graph analysis is particularly well suited to detecting connected behavior: shared devices across unrelated identities, rapid movement through newly linked accounts, circular transfers, coordinated merchant abuse, synthetic identity clusters, or a sudden break from an established behavioral path.

The PayGraph can improve fraud detection in three ways. First, it adds personal context, allowing the system to distinguish legitimate exceptions from suspicious anomalies. Second, it exposes network relationships that are invisible to account-level rules. Third, it enables adaptive security, where the system selects the least disruptive control appropriate to the risk—silent approval, passive monitoring, step-up authentication, funding-source confirmation, or manual review.

Security services should be architecturally separated from commercial personalization. A consumer may grant broad permission for fraud prevention while declining marketing. The PayGraph consent model must support these distinctions. Security teams should receive only the information necessary to protect the transaction, while marketing systems should never inherit privileged fraud or identity data merely because it exists elsewhere in the graph.

10. Personalization, Recommendations, and Financial Health

Personalization becomes materially more useful when it accounts for the consumer's entire connected financial context. A reward recommendation can consider eligible programs, merchant acceptance, current balances, expiration dates, funding costs, consumer goals, and program rules. A financial-health alert can identify duplicate subscriptions, unusually high fees, missed optimization opportunities, or a recurring cash-flow mismatch. A service agent can explain why a payment failed and propose the next best funding route.

The objective is not to maximize engagement at any cost. The objective is to improve relevance and measurable consumer value. Recommendations should be ranked by expected benefit, confidence, timeliness, user preference, and potential harm. Systems should be able to suppress an offer when the graph indicates that it would be inappropriate, duplicative, financially harmful, or inconsistent with the consumer's declared goals.

> **PERSONALIZATION STANDARD**
>
> A PayGraph recommendation should be useful because it understands context—not invasive because it exposes context. The consumer should be able to understand the benefit, the data category used, and how to turn the capability off.

11. Marketing, Sales, and Advertising

The PayGraph can shift commercial engagement from demographic targeting toward permissioned, event-aware relevance. Traditional systems infer broad segments from limited behavior. A graph can instead evaluate whether an offer is appropriate for the identity, whether the user already owns an equivalent product, whether the timing aligns with a declared goal or observed need, and whether the expected benefit exceeds a defined threshold.

For marketers, this enables fewer but more useful communications. For sales teams, it supports contextual lead prioritization, next-best conversation, life-event servicing, and intelligent handoffs. For merchants, it supports closed-loop measurement without requiring unrestricted access to the consumer's broader financial history. For consumers, it can reduce irrelevant messages and increase the value of rewards, incentives, and services.

Commercial use must remain purpose-limited. A developer should not receive a raw transaction history merely to determine whether an offer is relevant. Instead, the PayGraph can return a

scoped claim such as “eligible for offer,” “high relevance,” or “existing product relationship detected,” accompanied by policy and expiry. This claims-based model preserves utility while reducing disclosure.

Old model	PayGraph-enabled model
Large audience, low relevance	Smaller eligible audience, higher demonstrated value.
Static segments	Dynamic, permissioned context and event triggers.
Repeated product promotion	Suppression when the consumer already owns or rejects the product.
Raw-data exports	Scoped eligibility and relevance claims.
Channel-first campaigns	Consumer-preference and timing-aware orchestration.
Attribution by last click	Graph-based path and relationship measurement with privacy controls.

12. AI Agents and Graph-Based Reasoning

Large language models are powerful interfaces and reasoning tools, but they require reliable context. A generic model does not know which financial records belong to the user, which information is current, what the user has authorized, or which actions are permitted. The PayGraph can provide a governed context layer for financial agents.

An agent should not receive an unrestricted graph dump. It should request a task-specific context package assembled by policy. For example, an agent helping the user optimize a purchase may receive eligible payment instruments, reward opportunities, constraints, and explanations—but not unrelated medical, employment, or family data. The context package can include citations to graph nodes and events, confidence, freshness, and an action budget.

GraphRAG can combine structured traversal with semantic retrieval. Structured graph queries identify relevant entities, relationships, and policies. Semantic retrieval brings in documents, product rules, receipts, statements, and support knowledge. The language model synthesizes the result into an explanation or plan. Deterministic services retain control over money movement, authorization, eligibility, and policy enforcement.

CONTROL PRINCIPLE

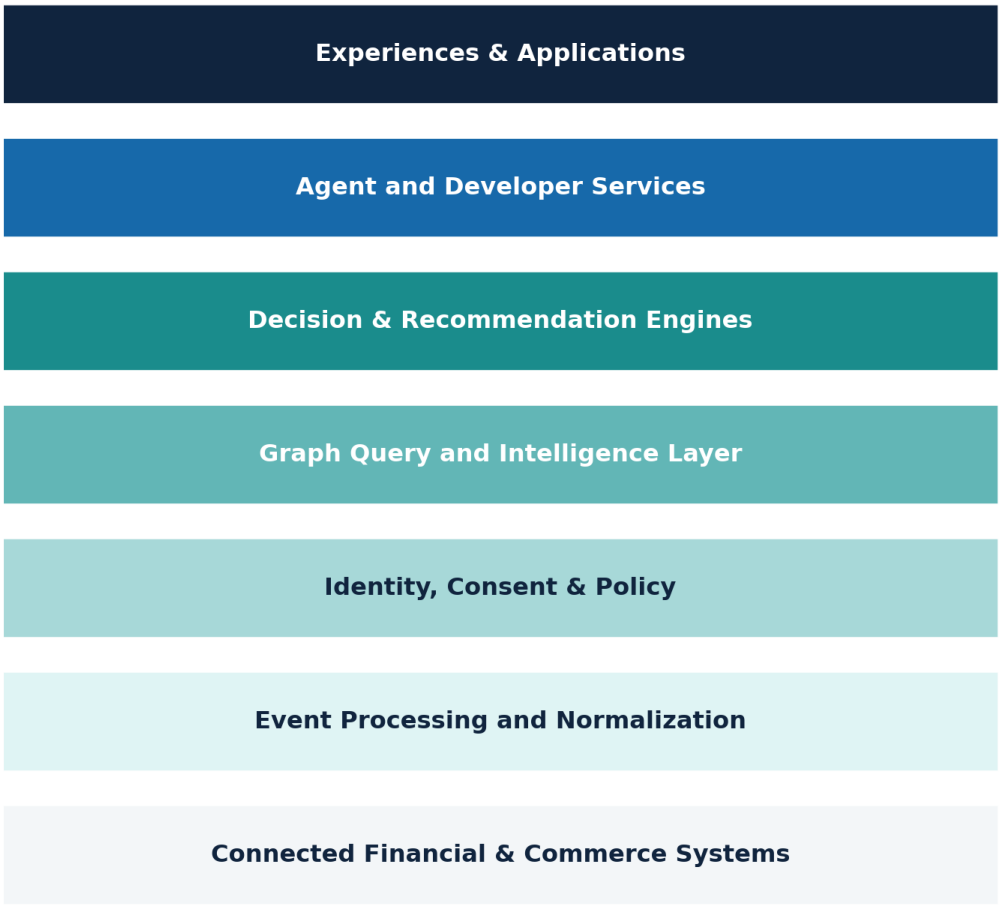
The LLM may interpret, summarize, and propose. Deterministic systems and authorized users approve consequential financial actions.

PART IV

The Architecture

A composable stack for low-latency decisions, deep analysis, developer services, and secure AI context.

13. The PayGraph Stack



The PayGraph Stack

Figure 3. The PayGraph Stack separates source connectivity, event processing, identity and consent, graph intelligence, decision services, and user experiences.

The architecture should separate systems of record from systems of intelligence. Banks, processors, networks, ledgers, wallets, and commerce platforms remain authoritative for the records they administer. The PayGraph consumes approved events and references, resolves them to governed identities, enriches the graph, and exposes intelligence services through carefully bounded interfaces.

Layer	Responsibilities
Connected systems	Banks, payment networks, processors, wallets, merchants, loyalty programs, investment platforms, identity providers.
Event ingestion	Authentication, schema validation, source attribution, deduplication, timestamp normalization, replay protection.
Tokenization and normalization	Replace sensitive identifiers, standardize merchants and accounts, classify events, preserve lineage.
Identity and consent	Identity resolution, roles, permissions, legal basis, purpose, revocation, policy versioning.
Graph and feature stores	Time-aware nodes and edges, graph projections, feature computation, embeddings, historical snapshots.
Intelligence services	Scoring, anomaly detection, recommendation, optimization, eligibility, explanation, simulation.
Developer and agent services	Claims APIs, GraphQL or domain APIs, SDKs, sandbox, webhooks, agent context, metering.
Experiences	mCards, Neobankify, partner apps, portals, merchant journeys, support tools, administrative consoles.

14. Identity Resolution

Identity resolution is the process of determining when records, credentials, devices, and financial relationships refer to the same person or authorized entity. It is foundational and high risk. Incorrect merges can expose data or distort decisions. Missed matches reduce intelligence. The PayGraph should therefore use layered assurance rather than one permanent universal identifier.

A robust model can combine verified identity tokens, institution assertions, device-bound credentials, account ownership checks, mobile and email verification, program enrollment, and user-confirmed relationships. Each linkage carries an assurance level, source, timestamp, and

expiry. High-risk actions require stronger evidence than low-risk personalization. Conflicting identity evidence should trigger quarantine or review rather than automatic merge.

Assurance tier	Illustrative basis	Permitted use
Observed association	Same session or device, unverified.	Low-risk analytics; never expose across trust boundaries.
User asserted	User connects or declares a relationship.	Personal organization and low-risk preferences.
Source verified	Institution or identity provider confirms relationship.	Financial servicing and controlled data access.
Strongly verified	Multi-factor, account ownership, government or bank-grade verification.	High-risk authorization and regulated actions.
Delegated authority	Verified person grants scoped authority to another.	Parent-child, employer-employee, fiduciary, support access.

15. Event Model and Temporal Graph

Financial meaning changes over time. A device that was trusted last month may be compromised today. A merchant relationship may be seasonal. An employee allowance may expire at the end of a trip. The PayGraph must therefore be temporal: nodes, relationships, permissions, scores, and claims should have effective dates, observation times, expiry, and version history.

Every event should include a globally unique event identifier, source, source event identifier, event time, ingestion time, identity references, object references, policy context, data classification, lineage, and integrity controls. Corrections should create new versions or compensating events rather than silently rewriting history.

The system should support both streaming and batch workloads. Streaming services update low-latency features and respond to time-sensitive events. Analytical workloads perform deeper graph traversal, cohort analysis, model training, simulation, and reporting. A common event contract ensures that the same underlying meaning is preserved across both paths.

16. Graph, Feature, and Knowledge Stores

No single database is likely to satisfy every PayGraph workload. The architecture should be polyglot by design. A graph database can manage relationship traversal. A transactional store can manage policies and consent. A feature store can serve model inputs at low latency. An object store can preserve immutable events and documents. A vector index can support semantic retrieval. A warehouse or lakehouse can support governed analytics and simulation.

The important requirement is not technological uniformity but semantic consistency. Each representation should map to a shared ontology, identifiers, lineage, and policy. Data copied into derived stores must retain deletion and revocation pathways. Sensitive raw data should not be replicated into systems that do not require it.

17. Developer and Enterprise Interfaces

The PayGraph should become commercially valuable through services rather than unrestricted database access. Developers can request a defined decision, score, claim, or context object. Enterprise partners can contribute events and receive governed intelligence in return. The interface should make the secure path the easiest path.

Service family	Illustrative operations
Identity	Create identity token, verify relationship, resolve identity, list assurance, revoke link.
Consent and policy	Grant purpose, inspect scope, renew, revoke, evaluate policy, retrieve audit receipt.
Events	Submit event, validate schema, retrieve status, replay to sandbox, subscribe to webhook.
Transaction confidence	Score authorization, retrieve reason codes, request step-up, simulate policy.
Fraud and trust	Evaluate anomaly, device trust, beneficiary risk, relationship risk, network pattern.
Recommendations	Request eligible offers, reward optimization, next best action, suppression reason.

Service family	Illustrative operations
Claims	Request scoped attribute, verify issuer, inspect expiry, validate signature.
Agent context	Create task context, retrieve citations, propose action, request user approval.
Simulation	Replay cohort, compare model versions, calculate impact, export evaluation report.

Every API response should include a request identifier, policy decision, data classification, confidence where applicable, freshness, expiry, explanation or reason codes, and audit references. High-risk services should support signed responses and non-repudiation controls.

18. Simulation, Testing, and Model Operations

A PayGraph decision service must be measurable before it is trusted. The platform should provide a simulation environment capable of replaying historical transactions and graph states against candidate policies, models, feature versions, and thresholds. Teams can compare outcomes without affecting production users.

Evaluation dimension	Examples
Decision quality	Approval rate, false positives, false negatives, fraud capture, routing success.
Consumer impact	Step-up frequency, unnecessary declines, time to resolution, benefit delivered.
Operational impact	Manual review volume, alert quality, support contacts, exception rates.
Model quality	Calibration, drift, segment performance, feature contribution, uncertainty.
Fairness and safety	Outcome differences, proxy risk, prohibited-feature leakage, harm scenarios.

Evaluation dimension	Examples
System performance	P50/P95/P99 latency, availability, timeout rate, cache effectiveness.
Economic impact	Fraud loss, interchange preservation, rewards value, conversion, cost to serve.

Production model operations should include versioning, approval workflows, shadow mode, canary releases, automated rollback, drift monitoring, incident response, and complete reproducibility. A score without a model version and feature snapshot is not auditable.

PART V

Trust by Design

How PayGraph can deliver intelligence without creating uncontrolled exposure of financial lives.

19. Privacy Architecture

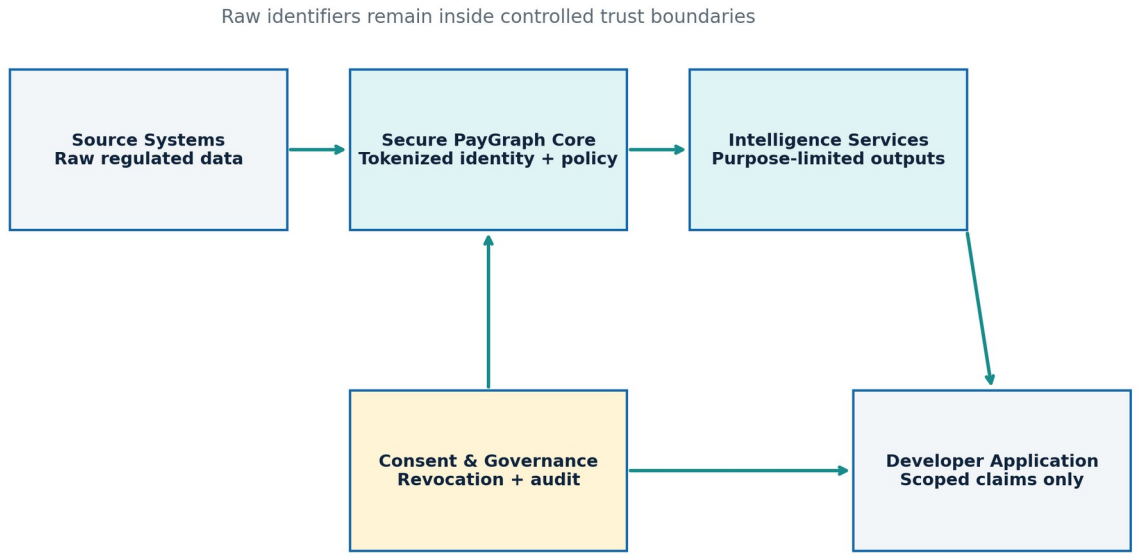


Figure 4. Raw identifiers remain within controlled trust boundaries; developers receive only purpose-limited, expiring intelligence outputs.

The strongest version of the PayGraph is not the one that centralizes the most data. It is the one that produces the most useful intelligence with the least exposure. Privacy architecture therefore begins with separation: raw regulated data, tokenized references, derived features, inference, and developer-accessible claims should occupy distinct trust zones.

A tokenization service can replace account numbers, card numbers, identity attributes, device identifiers, and partner-specific references with domain-limited tokens. The same underlying identifier may produce different tokens for different tenants or purposes, reducing the risk of

unauthorized correlation. Detokenization should be limited to explicitly authorized services operating inside controlled environments.

Where possible, the PayGraph can compute intelligence near the source or inside a secure processing boundary and return only the result. For example, a partner may ask whether an account is verified and eligible without receiving the account number or full history. This approach reduces breach impact and simplifies developer responsibilities.

20. Tokenization and Cryptographic Controls

Control	Design objective
Domain-scoped tokens	Prevent the same token from being reused to correlate identities across unrelated applications.
Format-preserving or surrogate tokens	Support integration without exposing original regulated identifiers.
Key separation	Use distinct keys and access policies by environment, tenant, data class, and function.
Hardware-backed protection	Protect key operations using managed HSM or equivalent controlled services.
Signed claims	Allow recipients to validate issuer, scope, integrity, and expiry without accessing source data.
Envelope encryption	Encrypt objects with data keys protected by centrally governed master keys.
Rotation and revocation	Support key rotation, token invalidation, credential revocation, and incident containment.
Confidential processing	Evaluate selected sensitive workloads within hardened or attested environments where appropriate.

Tokenization does not make data anonymous. Tokens remain personal or regulated data when they can be linked back to an individual or used to make decisions about them. The security model

must therefore combine tokenization with access control, purpose limitation, minimization, logging, retention, and governance.

21. Consent, Permission, and Consumer Control

Consent must be represented as a first-class graph object, not buried in a terms-of-service record. A consent object should identify the granting party, receiving party, data categories, purpose, permitted actions, duration, jurisdiction, policy version, delegation, revocation method, and evidence of the user interaction.

The system should distinguish consent from other lawful or contractual bases that may apply to security, fraud prevention, transaction execution, or regulatory obligations. The user experience should make those distinctions clear. Consumers should be able to see connected sources, active purposes, applications receiving claims, recent access, and the effects of revocation.

Consumer control	Expected behavior
Connect	Authorize a source or relationship with clear scope and purpose.
Inspect	View what is connected, what was derived, and which applications used it.
Correct	Update declared data and challenge material inferences or identity errors.
Pause	Temporarily disable a service or category without closing the entire identity.
Revoke	Terminate future access and invalidate dependent claims where appropriate.
Delete	Request deletion subject to legal, security, and contractual retention obligations.
Export	Receive a usable record of relevant data, permissions, and access history.
Explain	Understand material decisions and the main factors that affected them.

22. Zero Trust and Platform Security

Every request to the PayGraph should be authenticated, authorized, evaluated against policy, and logged. Network location alone should confer no trust. Services should use short-lived credentials, workload identity, mutual authentication, least privilege, segmented environments, and continuous posture evaluation.

The platform should maintain separate control planes and data planes; isolate tenants; protect secrets and keys; scan dependencies; validate event schemas; rate-limit abusive clients; detect anomalous access; and preserve tamper-evident audit records. Production access to sensitive data should be exceptional, time-bound, approved, monitored, and attributable to an individual or workload identity.

Security architecture must include resilience. PayGraph services used in transaction authorization need graceful degradation when dependencies fail. The system should define which claims can be served from cache, which decisions require fresh data, which timeouts trigger fallback, and which conditions require fail-safe decline or alternative routing.

23. Responsible Intelligence and Governance

Financial intelligence can create value and harm. Governance must therefore define not only how models are built, but which uses are acceptable. The PayGraph should maintain an approved-use catalog, prohibited-use catalog, risk classification, model inventory, decision ownership, escalation process, and independent review for sensitive applications.

Governance principle	Operational requirement
Purpose limitation	Every query and model maps to an approved business purpose.
Data minimization	Only necessary categories and precision are used.
Human accountability	Named owners remain accountable for automated outcomes.
Explainability	Material decisions expose understandable reasons and uncertainty.

Governance principle	Operational requirement
Fairness review	Models are evaluated for disparate outcomes and proxy variables.
User agency	Consumers can inspect, correct, revoke, and appeal where appropriate.
Auditability	Data, features, policies, models, and outputs retain lineage.
Safety by default	High-impact actions require stronger assurance and control.
Prohibited exploitation	No design that intentionally targets vulnerability, distress, addiction, or protected traits.
Separation of duties	Security, commercial, development, and governance privileges remain distinct.

Before external publication, legal and compliance counsel should map the design to applicable privacy, banking, payments, consumer-protection, credit, advertising, AI, and sector-specific obligations in each target jurisdiction. The white paper should describe design principles, not claim blanket compliance merely because a control appears in the architecture.

PART VI

Commercialization

How PayGraph can become an mCards capability, an enterprise service, and a developer ecosystem.

24. PayGraph Within mCards and Neobankify

mCards can act as both a source of permissioned financial events and an execution surface for PayGraph intelligence. Because an mCard can connect multiple funding relationships, spending rules, rewards, and partner services, it provides a practical mechanism for translating graph intelligence into real-time action. Neobankify can expose the same capabilities to brands, associations, enterprises, and institutions launching payment-led financial experiences.

The initial commercial value is likely to come from embedded services: transaction confidence, funding optimization, fraud and anomaly detection, rewards optimization, consent management, contextual servicing, and administrative intelligence. These services improve existing programs before the broader developer ecosystem is introduced.

A second stage can expose enterprise APIs and configurable intelligence modules. Partners can select only the services and data domains relevant to their program. A university may use family permissions and rewards. An employer may use allowance policies and expense context. A mortgage platform may use homeowner goals and commerce relationships. A merchant may use offer eligibility without receiving raw financial history.

25. Priority Use Cases

Use case	PayGraph contribution	Primary beneficiary
Real-time authorization	Contextual confidence, policy evaluation, route selection, step-up.	Consumer, issuer, processor.

Use case	PayGraph contribution	Primary beneficiary
Funding optimization	Select eligible source based on cost, rewards, balance, rules, and preference.	Consumer, program, partner.
Parent-child controls	Model delegated authority, budgets, merchants, time, geography, goals.	Families and youth programs.
Employer-funded spend	Connect employee, trip, allowance, purpose, receipt, and policy.	Employers and employees.
Rewards intelligence	Unify earn, burn, expiry, merchant offers, and payment choice.	Consumers, brands, loyalty programs.
Fraud networks	Detect shared devices, synthetic clusters, suspicious paths, coordinated abuse.	Issuers, networks, merchants.
Financial health	Find recurring waste, cash-flow gaps, fees, optimization, and goal progress.	Consumers and advisors.
Service personalization	Provide context-aware explanations and next best resolution.	Consumers and support teams.
Offer eligibility	Return purpose-limited relevance and eligibility claims.	Consumers, merchants, distributors.
Agentic finance	Provide governed context and action boundaries to AI assistants.	Consumers, developers, institutions.

26. Developer Platform and Marketplace

The long-term opportunity is to allow thousands of developers to build on the PayGraph without becoming custodians of unrestricted financial data. A developer registers an application, declares its purposes, selects approved service scopes, completes risk review, and receives sandbox

credentials. The application uses signed claims and task-specific APIs rather than downloading the consumer's graph.

The platform can offer composable products: identity confidence, transaction confidence, merchant understanding, rewards optimization, household permissions, financial-health insights, agent context, and simulation. Developers can combine these products into specialized experiences while the PayGraph enforces consistent consent, policy, security, and audit.

A marketplace can include certification, quality ratings, security posture, pricing, usage metering, consumer permissions, revenue sharing, and revocation. The ecosystem should reward applications that produce demonstrable consumer value and penalize unnecessary data access, opaque decisioning, or manipulative engagement.

27. Commercial Models

Model	Description
Embedded platform capability	Included within mCards or Neobankify program tiers.
Usage-based intelligence API	Per decision, event, claim, graph query, or agent context package.
Enterprise subscription	Committed capacity, administrative tools, governance, reporting, and support.
Outcome-aligned pricing	Share of measurable fraud reduction, rewards value, conversion lift, or operational savings where appropriate.
Developer marketplace	Revenue share on approved third-party applications and services.
Data collaboration network	Privacy-preserving intelligence exchange among qualified institutions and partners.
Simulation and analytics	Model evaluation, historical replay, benchmarking, and strategy services.
Standards and certification	Conformance testing, secure implementation certification, and partner accreditation.

Commercialization should not depend on selling identifiable consumer data. The preferred model monetizes services, decisions, infrastructure, software, and measurable value. This positioning aligns economic incentives with privacy engineering and increases trust with consumers and institutional partners.

28. Product and Ecosystem Roadmap

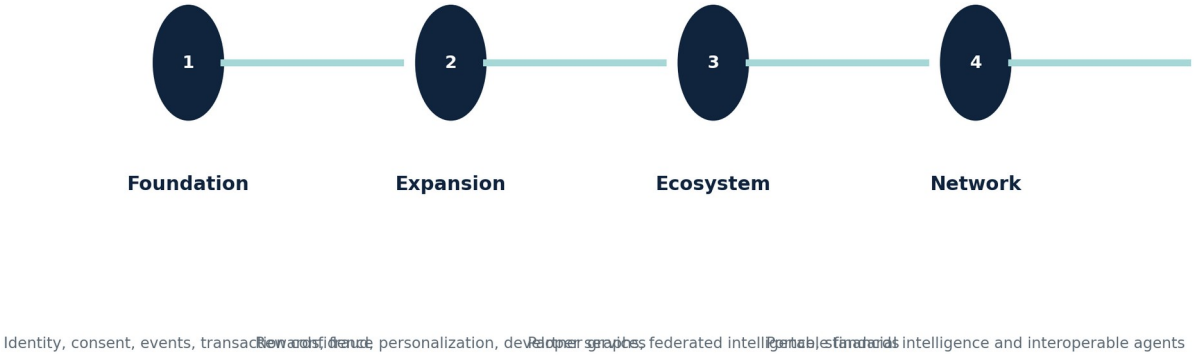


Figure 5. A staged roadmap moves from core identity and transaction services toward an interoperable financial intelligence network.

Phase	Primary deliverables	Exit criteria
1 — Foundation	Ontology, identity tokens, consent, event bus, graph store, transaction confidence, simulation.	Reliable replay, audited policy decisions, production latency targets, initial partner use.
2 — Expansion	Fraud graph, rewards, recommendations, admin tools, claims APIs, agent context.	Measured value across several use cases; developer sandbox operational.
3 — Ecosystem	Partner contribution model, marketplace, cross-program intelligence, certification.	Multiple external applications; repeatable governance and commercial model.
4 — Network	Portable intelligence, federated graph collaboration, broader standards and interoperability.	Independent implementations or partners conform to shared specification.

29. How the Market Should Understand PayGraph

The PayGraph should be introduced through a disciplined hierarchy. At the highest level, it is the financial intelligence layer. For banks and networks, it improves trust, authorization, and customer understanding. For enterprises and brands, it makes payment-led experiences more relevant and secure. For developers, it provides governed financial context without requiring them to rebuild connectivity, identity, consent, and intelligence infrastructure. For consumers, it creates a more coherent financial experience under their control.

The market narrative should avoid presenting PayGraph as surveillance, a universal consumer score, or a centralized repository of every financial fact. It should also avoid suggesting that AI replaces regulated decision-makers or deterministic payment systems. The strongest positioning is practical: PayGraph helps approved software make better decisions from connected, permissioned context while revealing less raw information.

> **POSITIONING LINE**
>
> PayGraph is the permissioned intelligence layer that helps financial software understand the relationships between identity, money, commerce, trust, and intent.

PART VII

The Standard

The principles that define a credible PayGraph implementation and the path toward a broader specification.

30. PayGraph Design Principles

1. Identity is the center of understanding; accounts remain authoritative systems of record.
2. Permission and policy travel with data, features, claims, and decisions.
3. Raw financial data is not the product; governed intelligence services are the product.
4. Facts, observations, derived attributes, and inferences remain visibly distinct.
5. The minimum necessary data and precision are used for every purpose.
6. Consumers can inspect, correct, pause, revoke, and understand material uses.
7. Deterministic controls govern consequential financial actions.
8. Models expose confidence, reason codes, freshness, and lineage.
9. Security and fraud permissions do not automatically authorize commercial use.
10. Developers receive scoped claims and task context rather than unrestricted graph access.
11. Every high-impact capability is testable through simulation, replay, and audit.
12. Commercial success is aligned with measurable value and trust, not data extraction.

31. Toward the PayGraph Specification

The flagship white paper defines the thesis and reference architecture. A future PayGraph Specification can define interoperable implementation details: canonical node and edge types, event schemas, identity assurance, consent receipts, policy expressions, token formats, signed claims, score envelopes, model documentation, audit events, agent context packages, API conventions, versioning, and conformance tests.

The specification should be modular. Institutions may implement selected domains while retaining interoperability. A payment processor might implement transaction events and confidence claims. A rewards provider might implement program, balance, and redemption objects. A developer platform might implement consent, claims, and agent context. Conformance profiles can define the minimum requirements for each role.

Open standards should be considered carefully. Some aspects may become public specifications to accelerate adoption. Other components—models, orchestration, specialized graph features, and implementation methods—may remain proprietary mCards intellectual property. The publication and IP strategy should be established before detailed schemas are released externally.

Conclusion

The financial world has accumulated extraordinary amounts of data without giving individuals or software a coherent understanding of how their financial lives fit together. The result is fragmented experiences, avoidable fraud, irrelevant marketing, stranded rewards, repeated verification, and applications that know too little or expose too much.

The PayGraph offers a different model. It organizes permissioned financial intelligence around identity, relationships, events, context, and trust. It provides a foundation for better transaction decisions, stronger security, meaningful personalization, responsible marketing, optimized rewards, and AI agents that can reason from verified context without receiving unrestricted access to raw financial records.

The concept originated in 2006, before the necessary infrastructure was commercially accessible. Today, mCards, Neobankify, modern payment orchestration, graph systems, tokenization, cloud security, and AI create a practical path forward. The opportunity is not merely to build another fintech feature. It is to define a new layer of financial computing—and to do so with privacy, consumer agency, and institutional trust built into the architecture from the beginning.

PART A

Appendices

Controlled vocabulary, maturity labels, validation requirements, and publication roadmap.

Appendix A — Controlled Vocabulary

Term	Canonical definition
PayGraph	The complete permissioned financial intelligence graph and service layer associated with an identity or authorized entity.
Financial identity	The set of verified relationships, roles, credentials, and financial contexts connected to an identity.
Graph event	A time-stamped occurrence that may create, update, or contextualize nodes, edges, features, policies, or scores.
Signal	A supplied or observed input used for policy, scoring, or interpretation.
Claim	A purpose-limited, signed or verifiable statement returned to an authorized service.
Derived attribute	A value computed from one or more facts or events.
Inference	A probabilistic interpretation that is not directly observed or verified.
Transaction Confidence	A contextual assessment supporting authorization, routing, step-up, or review.

Term	Canonical definition
Identity Confidence	An assessment of the strength and consistency of identity resolution.
GraphRAG	A retrieval architecture combining graph traversal, semantic retrieval, and language-model synthesis.
Purpose limitation	Restriction that information may be used only for declared and approved purposes.
Consent receipt	A durable record of what was authorized, by whom, for whom, for what purpose, and for how long.
Task context package	A minimized set of graph claims, evidence, and policies assembled for a specific AI or application task.
PayGraph Specification	A proposed set of interoperable schemas, protocols, controls, and conformance requirements.

Appendix B — Capability Maturity Labels

Label	Meaning	Publication treatment
Current	Demonstrable or deployed capability supported by evidence.	May use present tense with precise scope.
In development	Actively designed or being built with an approved plan.	Use forward-looking language and indicate status.
Reference architecture	Technically defined target design not necessarily implemented.	Describe as architecture or requirement.
Research direction	Requires experimentation, validation, or partner collaboration.	Present as hypothesis or research agenda.

Label	Meaning	Publication treatment
Long-term vision	Aspirational ecosystem or standards outcome.	Clearly label as future vision.

Appendix C — Publication Validation Checklist

- Assemble documentary support for the 2006 origin narrative and establish approved wording.
- Validate all descriptions of current mCards, Neobankify, DPE, issuing, connectivity, and processing capabilities.
- Confirm which transaction-confidence components are deployed, prototyped, specified, or conceptual.
- Complete architecture review with engineering, data, security, and payment-processing leaders.
- Complete privacy and regulatory review for target jurisdictions and intended data uses.
- Define prohibited-use policy, sensitive-inference policy, fairness framework, and consumer appeal process.
- Identify external technical standards and sources that should be cited in the published edition.
- Commission threat modeling, data-flow mapping, and API abuse-case review.
- Determine IP, trademark, patent, publication, and open-specification strategy.
- Create evidence-backed case studies, measured outcomes, and partner-approved examples.
- Perform editorial, legal, accessibility, and design QA before public release.

Appendix D — PayGraph Publication Family

Publication	Audience	Purpose
Flagship White Paper	Executives, banks, networks, investors, policy leaders, technical buyers.	Define the thesis, model, architecture, trust framework, and commercial opportunity.
Executive Brief	Senior decision-makers and prospective partners.	Provide a 10–15 minute visual explanation and partnership pathway.

Publication	Audience	Purpose
Technical Architecture Paper	Architects, engineering, data, security, diligence teams.	Detail components, data flows, latency, graph models, operations, and integration.
Privacy, Security & Governance Paper	Banks, regulators, compliance, privacy, procurement.	Explain controls, responsibilities, risk boundaries, and responsible intelligence.
Developer Specification	Developers, partners, platform engineers.	Define objects, events, claims, APIs, SDKs, sandbox, webhooks, and conformance.
Industry Solution Briefs	Vertical partners and sales teams.	Translate PayGraph services into concrete use cases and outcomes.
Research Notes	AI, graph, privacy, fraud, and economic researchers.	Publish experiments, benchmarks, methods, and open questions.

Appendix E — Proposed Publication Credits

Concept Originator: David Strebinger

Publisher: mCards Inc.

Recommended review credits for the public edition: Executive Sponsor; Product and Platform Architecture; Payments Engineering; Data and AI; Privacy and Security; Legal and Regulatory; Editorial Director; Information Design; Independent Technical Review.

A formal acknowledgments section should identify contributors accurately and distinguish authorship, review, design, and advisory roles.

PAYGRAPH

Identity. Money. Commerce. Trust. Intelligence.

Working Draft v0.1 — Strategic and Technical Foundation

mCards Inc. • July 2026