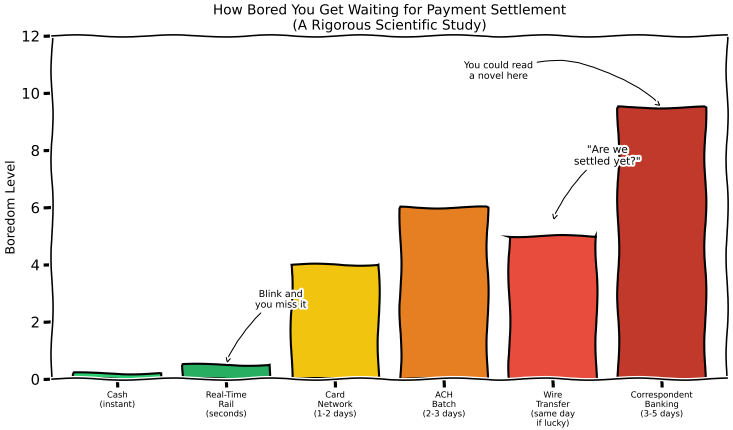
Lesson 1.3: Real-Time Payments and Cost Compression

Module 1: The Cost of Financial Intermediation

Digital Finance

The Long Wait for Settlement



Settlement speed varies dramatically across payment rails—from seconds to days.

Learning Objectives

By the end of this lesson, you will be able to:

1. Compare real-time payment architectures across different design patterns [Analyze]
2. Explain how ISO 20022 messaging standards enable interoperability [Understand]
3. Analyze how stablecoin rails offer an alternative to traditional correspondent banking [Analyze]
4. Evaluate the economics of Buy Now Pay Later (BNPL) for merchants and consumers [Evaluate]
5. Design a payment orchestration strategy that optimizes for cost, speed, and reliability [Create]

Bloom's taxonomy levels: Understand (2), Analyze (1,3), Evaluate (4), Create (5).

From Economics to Technology

In Lessons 1.1–1.2, we learned why costs exist.

Information asymmetry $\rightarrow$ intermediaries $\rightarrow$ fees

$\Downarrow$

Now we see how technology compresses them.

Real-time settlement, structured messaging, and programmable money can reduce layers, automate compliance, and shrink settlement windows from **days** to **seconds**.

Technology does not eliminate the need for trust—it changes who provides it and at what cost.

What is Real-Time Gross Settlement (RTGS)?

Definition:
Real-Time Gross Settlement (RTGS) is a funds-transfer system where each payment is settled *individually* and *immediately* on a gross basis, without netting or batching.

Key Properties:

- **Real-time:** Settlement occurs in seconds
- **Gross:** Each transaction settles individually (no netting)
- **Final:** Settlement is irrevocable once confirmed
- **Central bank money:** Settles on central bank ledger

Design Trade-offs:

- Requires participants to hold *liquidity buffers* at the central bank
- Higher per-transaction cost than batch systems
- Typically used for high-value / wholesale payments
- Examples: Fedwire (US), TARGET2 (EU), Clearing House Automated Payment System (CHAPS, UK)

Contrast with Net Settlement:

- Batch systems net obligations at end of day
- Lower liquidity need, but *delayed* finality
- Introduces counterparty risk during the day

RTGS eliminates intraday credit risk but demands higher liquidity—a classic cost-risk trade-off.

Instant Payment Schemes: Architecture Patterns

What are Instant Payment Schemes?
Retail-focused systems enabling 24/7/365 real-time credit transfers between bank accounts, typically within seconds.

Design Pattern A: Centralized Clearing House

- Single operator validates and routes messages
- Participants pre-fund a settlement account
- Deferred net settlement at intervals (e.g., every 15 min)
- Example pattern: SEPA Credit Transfer Instant (SCT Inst)

Design Pattern B: Distributed / Hub-and-Spoke

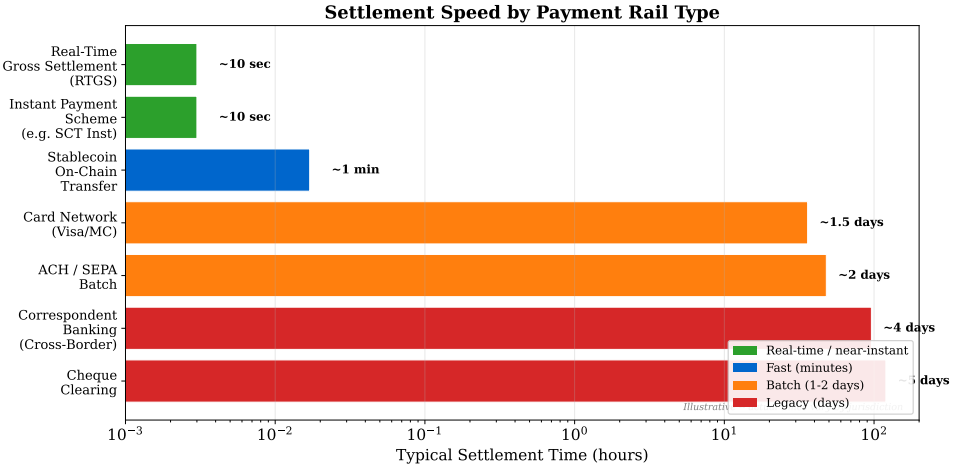
- Multiple interconnected clearing nodes
- Bilateral or multilateral settlement
- More resilient to single-point failure
- Example pattern: India's Unified Payments Interface (UPI design)

Design Pattern C: Overlay on Existing Rails

- Built atop legacy batch infrastructure
- Adds a real-time messaging layer
- Settlement still occurs on underlying RTGS
- Lower deployment cost, but legacy constraints remain

Different countries choose different architectures based on existing infrastructure and policy goals.

Settlement Speed: A Quantitative Comparison



- **What you see:** Horizontal bar chart on log scale showing settlement time (hours) for 7 payment rails, color-coded from green (instant) to red (legacy)
- **Key pattern:** RTGS and instant schemes settle in ~10 seconds; card networks 1–2 days; correspondent banking 4–5 days; cheque clearing ~10,000

What is Request-to-Pay (RtP)?

Definition:
Request-to-Pay (RtP) is a messaging framework that allows a payee to send a structured payment request to a payer, who can then approve, reject, or negotiate the payment.

How It Works:

1. Payee sends RtP message (amount, reference, due date)
2. Payer receives notification (mobile app, banking portal)
3. Payer reviews and *authorizes* payment
4. Payment executes over instant payment rail

Why It Matters for Cost Compression:

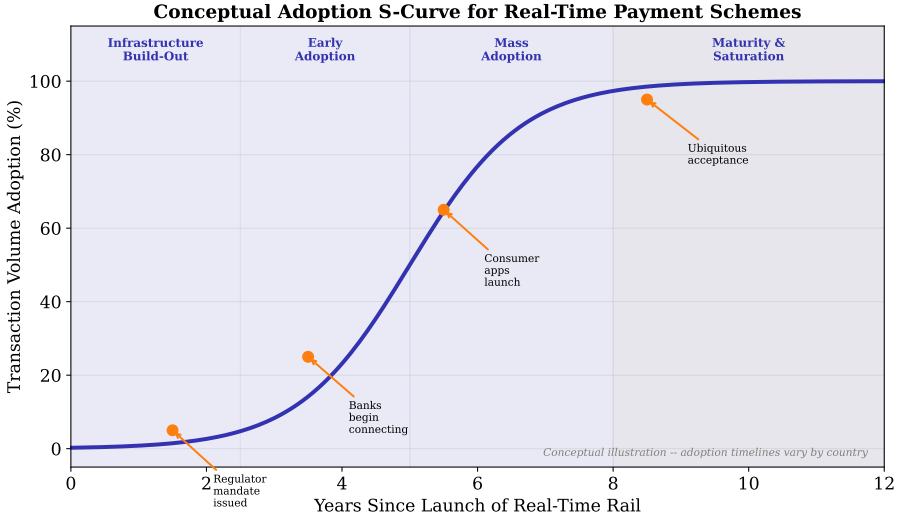
- Replaces direct debits (which have high dispute rates)
- Reduces failed-payment costs (payer confirms upfront)
- Enables flexible payment scheduling
- Rich data attached to each request (ISO 20022)

Use Cases:

- Bill presentment (utilities, subscriptions)
- E-commerce checkout (alternative to card payment)
- Business-to-business invoice settlement
- Government disbursements with confirmation

RtP shifts control to the payer while embedding rich data—reducing disputes and reconciliation costs.

Adoption Curves: How Real-Time Rails Scale



What is ISO 20022?

Definition:
ISO 20022 is a global standard for financial messaging that defines a common vocabulary and syntax (XML/JSON) for payment instructions, reporting, and securities settlement.

Key Features:

- **Structured data:** Named fields instead of free text
- **Rich remittance info:** Up to 140+ characters vs. 4 in legacy MT messages
- **Structured addresses:** Street, city, country as separate fields
- **End-to-end references:** Unique identifiers across the chain

Why It Enables Cost Compression:

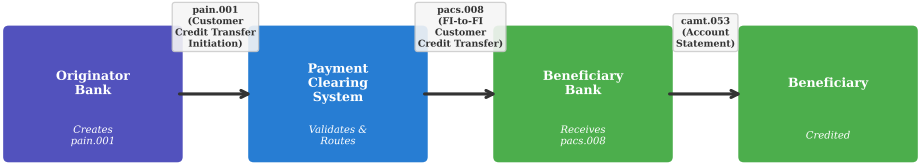
- **Straight-Through Processing (STP):** Machines can parse and route without human intervention
- **Fewer exceptions:** Structured data means fewer rejected or returned payments
- **Automated compliance:** Screening algorithms work on structured name/address fields
- **Interoperability:** Same message format across domestic and cross-border rails

Migration Timeline:

- SWIFT cross-border: migrating 2023–2025
- TARGET2 (EU): live since March 2023
- Fedwire (US): planned migration in progress

ISO 20022 is the "Rosetta Stone" of payments—enabling machines to process what humans used to reconcile manually.

ISO 20022: Structured Message Flow



ISO 20022 Data Richness (vs. Legacy MT/SWIFT Messages)

MT103: 4 remittance chars	-->	**pain/pacs: 140+ remittance chars**
Unstructured address fields	-->	**Structured address (street, city, country)**
Limited purpose codes	-->	**Rich purpose codes + categories**
No structured references	-->	**Structured end-to-end references**

Legacy MT Messages: The Problem

Legacy MT Messages (SWIFT FIN):

- Fixed-length fields, positional syntax
- Free-text remittance (Field 70: 4 $\times$ 35 chars)
- Unstructured addresses
- Limited character set (ASCII subset)
- No end-to-end unique identifier

Result:

- High exception rate ($\sim$5–10% of cross-border payments)
- Manual repair of garbled addresses
- Compliance screening on unstructured text = false positives

Legacy MT messages force humans to interpret free-text fields—creating costly exceptions and compliance false positives.

ISO 20022 Messages: The Solution

ISO 20022 Messages:

- XML/JSON with named elements
- Structured remittance (140+ chars, typed fields)
- Structured addresses (street, building, city, country code)
- Unicode support
- Mandatory end-to-end transaction ID

Result:

- Near-zero exception rate for compliant messages
- Automated reconciliation (STP rates $>99\%$)
- Compliance screening on structured fields = fewer false positives

The shift from MT to ISO 20022 is not cosmetic—it fundamentally changes what machines can do with payment data.

What are Stablecoin Payment Corridors?

Definition:
A stablecoin payment corridor uses blockchain-based tokens pegged to a fiat currency (e.g., USD, EUR) to transfer value across borders, bypassing traditional correspondent banking chains.

How It Works:

1. Sender deposits fiat with an on-ramp provider
2. Provider mints or transfers stablecoins on-chain
3. Stablecoins are sent to recipient's wallet (seconds)
4. Recipient uses an off-ramp to convert back to local fiat

Cost Compression Mechanisms:

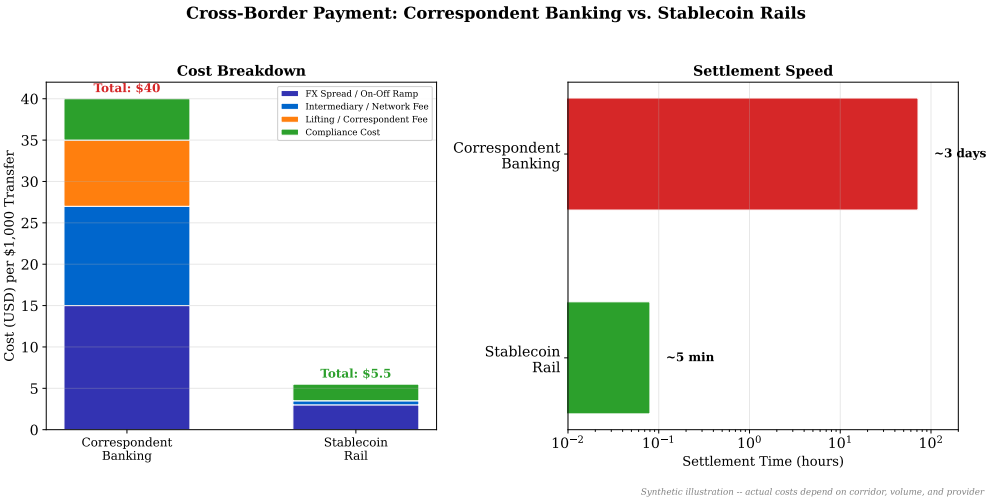
- **Disintermediation:** Removes 2–4 correspondent banks
- **24/7 availability:** No batch windows or banking hours
- **Transparent fees:** On-chain gas fees are visible
- **Programmable:** Smart contracts can enforce compliance

Challenges:

- On/off-ramp friction and regulatory licensing
- FX conversion still needed at endpoints
- Reserve transparency and audit requirements
- Regulatory uncertainty in many jurisdictions

Stablecoins compress the correspondent banking chain but introduce new risks at the on/off-ramp layer.

Stablecoin vs. Correspondent Banking: Cost and Speed



- **What you see:** Two panels — left shows stacked cost breakdown (FX, intermediary, lifting, compliance) per $1,000 transfer; right

Stablecoin Corridor Design — Chain and Reserve Model

Chain Selection:

- Layer-1 (e.g., Ethereum): Higher security, higher gas cost
- Layer-2 (e.g., rollups): Lower cost, faster finality
- Purpose-built chains: Optimized for payments
- Trade-off: decentralization vs. throughput vs. cost

Reserve Model:

- Full fiat reserve (1:1 backing)
- Over-collateralized crypto reserve
- Algorithmic (no reserve—higher risk)
- Attestation vs. full audit transparency

Chain selection is the first design decision—it determines throughput, cost, and settlement finality for the entire corridor.

Stablecoin Corridor Design — Regulation and Economics

Regulatory Requirements:

- Know Your Customer (KYC) at on/off-ramps
- Anti-Money Laundering (AML) transaction monitoring
- Travel Rule compliance (originator/beneficiary data)
- Licensing: e-money, payment institution, or banking

Corridor Economics:

- On-ramp fee: typically 0.1–1.0% of value
- Network/gas fee: $0.01–$5 depending on chain
- Off-ramp fee: typically 0.1–1.5% of value
- Total: often $<1\%$ vs. 3–7% for legacy corridors

The "last mile" of stablecoin corridors—fiat on/off-ramps—is where most cost and friction remain.

What is Buy Now Pay Later (BNPL)?

Definition:
Buy Now Pay Later (BNPL) is a short-term consumer financing product that splits a purchase into installments (typically 3–4 payments), often with zero interest to the consumer. The merchant pays a Merchant Discount Rate (MDR) to the BNPL provider.

How It Works:

1. Consumer selects BNPL at checkout
2. BNPL provider runs a soft credit check (seconds)
3. Provider pays merchant immediately (minus MDR)
4. Consumer repays in installments over 4–6 weeks

Key Parties and Incentives:

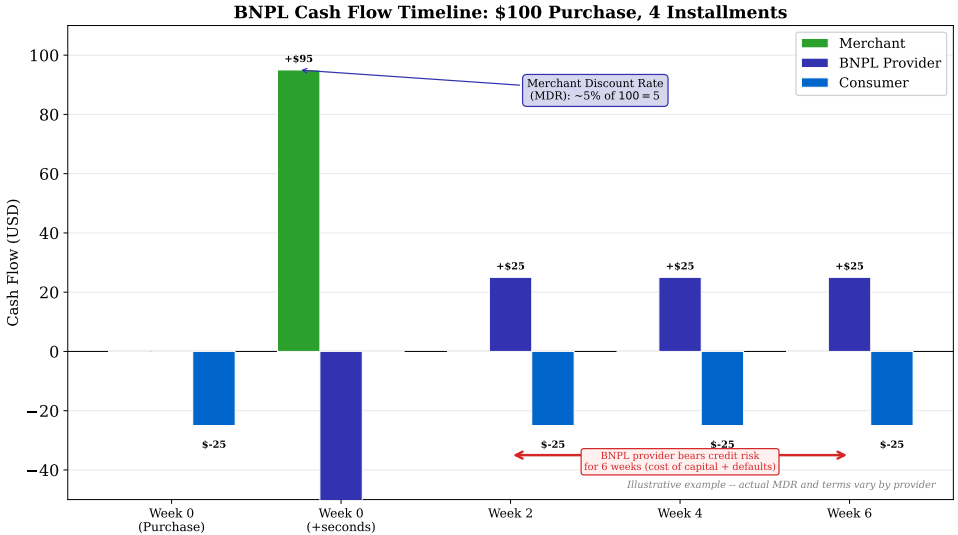
- **Consumer:** Interest-free installments $\rightarrow$ higher willingness to buy
- **Merchant:** Higher conversion rate, larger basket size $\rightarrow$ worth the MDR
- **BNPL Provider:** Earns MDR ($\sim$3–8% of transaction value)

Revenue Model for BNPL Provider:

- Primary: Merchant Discount Rate (MDR)
- Secondary: Late fees from consumers
- Tertiary: Interest on longer-term plans
- Data monetization (consumer spending patterns)

BNPL shifts credit cost from the consumer to the merchant—the merchant pays a higher MDR in exchange for increased sales.

BNPL Cash Flow Mechanics



BNPL Economics — Merchant and Provider Perspectives

Merchant's Calculation:

- MDR for BNPL: typically 3–8% (vs. 1.5–3% for cards)
- Justified if conversion rate uplift $\times$ basket size increase $>$ incremental MDR cost
- Example: If MDR increases from 2% to 5%, merchant needs a $>3\%$ revenue uplift to break even

BNPL Provider's Unit Economics:

- Revenue per transaction: MDR (e.g., 5% of \$100 = \$5)
- Cost of capital: funding the advance ($\sim$0.5–1%)
- Credit losses: consumer defaults ($\sim$1–4%)
- Operations: underwriting, servicing ($\sim$0.5–1%)
- Contribution margin: thin (often $<2\%$)

BNPL provider margins are razor-thin: MDR revenue must cover cost of capital, credit losses, and operations—leaving less than 2% contribution margin.

BNPL Economics — Consumer Risk and Regulatory Response

Consumer's Perspective:

- "Free" if paid on time—but is it?
- Late fees can exceed credit card interest
- Encourages spending beyond means
- No credit-card consumer protections (chargebacks)
- Multiple BNPL obligations may not appear on credit reports

Regulatory Response:

- EU Consumer Credit Directive: BNPL classified as credit
- UK FCA: bringing BNPL under regulation
- Affordability checks being mandated
- Disclosure requirements increasing

BNPL appears "free" to consumers, but merchants pay 2–3$\times$ card MDR—and default risk ultimately falls on the provider.

What is Payment Orchestration?

Definition:
Payment orchestration is a technology layer that sits between a merchant and multiple payment processors / acquirers, intelligently routing each transaction to optimize for cost, speed, and authorization success rate.

Core Functions:

- **Smart routing:** Select the lowest-cost or highest-success processor for each transaction
- **Failover:** If processor A declines, automatically retry with processor B
- **Multi-acquirer:** Connect to multiple acquirers in parallel
- **Payment method fan-out:** Cards, bank transfers, wallets, BNPL

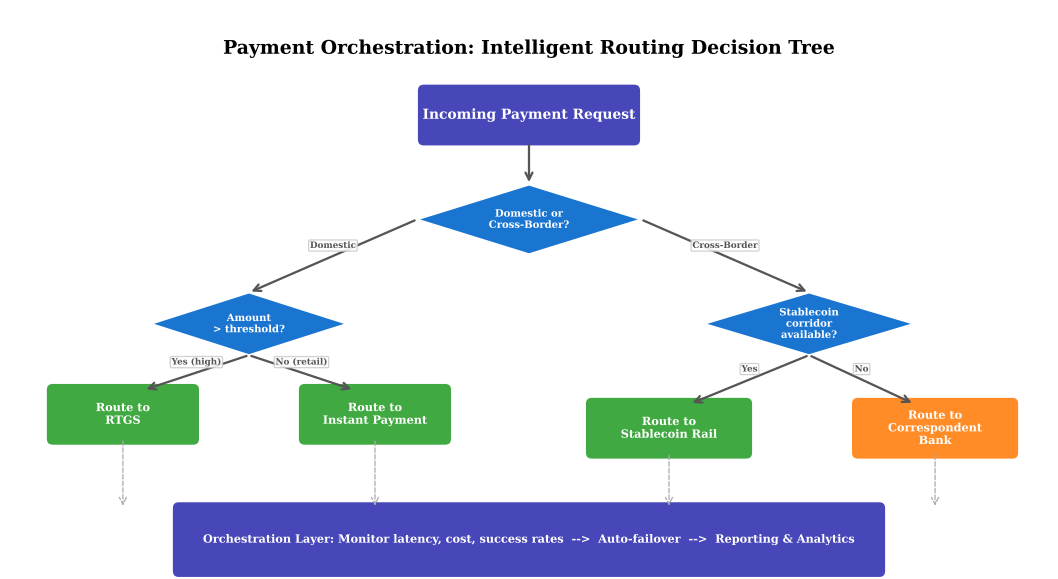
Why It Matters:

- A 1% improvement in authorization rate can yield millions in recovered revenue for large merchants
- Cross-border routing: local acquiring reduces FX markup
- Compliance: route to processors with correct licenses per jurisdiction

Orchestration Strategies:

- Cost-optimized: minimize processing fees
- Success-optimized: maximize authorization rate
- Speed-optimized: minimize settlement latency
- Balanced: weighted multi-objective routing

Payment orchestration turns payment processing from a single pipe into an intelligent, multi-path network.


Payment Orchestration: Intelligent Routing Decision Tree
Incoming Payment Request
Domestic or Cross-Border?
Domestic
Cross-Border
Amount > threshold?
Stablecoin corridor available?
Yes (high)
No (retail)
Yes
No
Route to RTGS
Route to Instant Payment
Route to Stablecoin Rail
Route to Correspondent Bank
Orchestration Layer: Monitor latency, cost, success rates --> Auto-failover --> Reporting & Analytics

Orchestration Metrics — Authorization and Cost

Authorization Rate:

- Percentage of attempted payments that are approved
- Industry benchmark: 85–95% depending on sector
- Each 1% improvement $\approx$ significant revenue recovery
- Affected by: Bank Identification Number (BIN) routing, 3-D Secure (3DS) version, retry logic

Cost Per Transaction:

- Interchange + scheme fee + acquirer markup
- Varies by card type, region, and MCC (Merchant Category Code)
- Orchestrator can route to minimize this
- Savings: 10–30 basis points per transaction

Authorization rate is the single most impactful metric—a 1% lift can recover more revenue than a fee reduction.

Orchestration Metrics — Settlement, Declines, and Chargebacks

Settlement Latency:

- Time from authorization to funds received
- Ranges from T+0 (instant) to T+3 (batch)
- Cash-flow-sensitive merchants optimize for speed

Decline Analysis:

- Hard declines: insufficient funds, stolen card
- Soft declines: temporary holds, velocity limits
- Orchestrator retries soft declines on alternate rail
- Reduces false declines (legitimate transactions blocked)

Chargeback Rate:

- Disputed transactions as % of volume
- Threshold: typically $<1\%$ to avoid scheme penalties
- Routing can avoid high-chargeback corridors

Decline analysis reveals hidden revenue leaks. Many "declined" transactions are legitimate customers blocked by overly aggressive fraud rules.

What is Open Banking?

Definition:
Open banking is a regulatory and technological framework that requires banks to share customer account data and payment initiation capabilities with authorized third-party providers (TPPs) via standardized APIs, with the customer's explicit consent.

Regulatory Framework Types:

- **Second Payment Services Directive (PSD2, EU, 2018):** Mandates API access; defines AISP and PISP roles
- **Third Payment Services Directive (PSD3, EU, proposed):** Strengthens API performance standards, adds fraud-sharing
- **UK Open Banking:** CMA-mandated; standardized API specs
- **Market-led:** US, Australia—industry standards without mandate

Key Roles (PSD2/PSD3 Terminology):

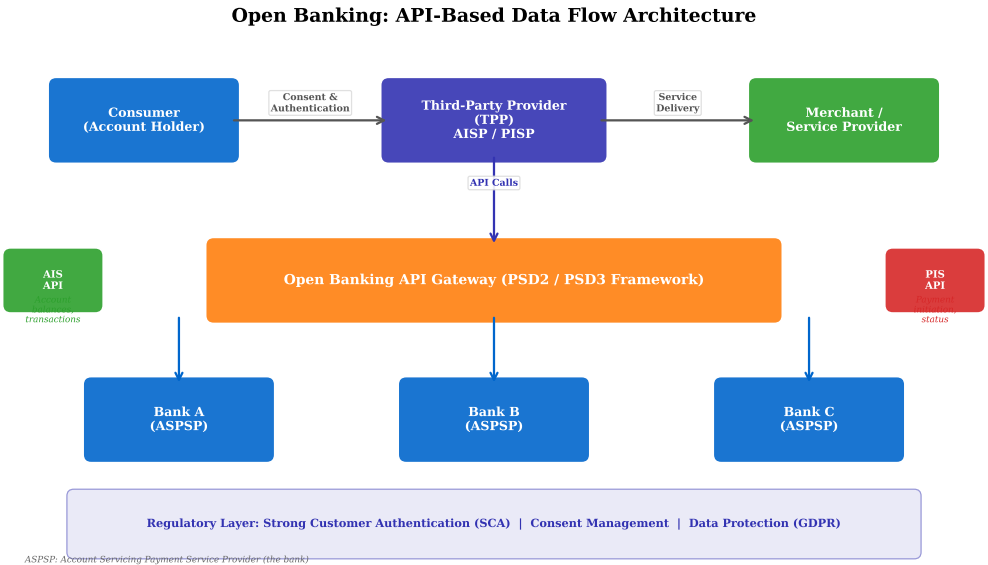
- **ASPSP:** Account Servicing Payment Service Provider—the bank
- **AISP:** Account Information Service Provider—reads balances/transactions (read-only)
- **PISP:** Payment Initiation Service Provider—initiates payments on behalf of the customer
- **TPP:** Third-Party Provider—umbrella term for AISP/PISP

Cost Compression Mechanism:

- Account-to-account (A2A) payments bypass card networks
- No interchange fee, no scheme fee
- Direct bank transfer at near-zero marginal cost

Open banking mandates API access to bank accounts—enabling account-to-account payments that bypass card networks entirely.

Open Banking Data Flow Architecture



PSD2: Achievements and Limitations

PSD2 (Second Payment Services Directive, 2018):

- Mandated banks to provide APIs to TPPs
- Introduced Strong Customer Authentication (SCA)
- Defined AISP and PISP licenses

Limitations encountered:

- API quality varied widely across banks
- Screen-scraping fallback created security tension
- No penalties for poor API performance
- Fraud liability unclear for open-banking payments

PSD2 opened the door to open banking but left implementation quality to the banks—resulting in wildly uneven API experiences.

PSD3 / PSR: Closing the Gaps

PSD3 / PSR (proposed):

- Mandatory API performance standards with Service Level Agreements (SLAs)
- Dashboard for consumer consent management
- Fraud-data-sharing framework between banks and TPPs
- IBAN/name verification to prevent misdirected payments
- Direct access to payment systems for non-bank PSPs
- Removal of screen-scraping fallback

Net Effect on Costs:

- Higher-quality APIs $\rightarrow$ higher STP rates
- Fraud sharing $\rightarrow$ lower fraud losses
- Direct system access $\rightarrow$ lower intermediation cost

---

PSD3 addresses PSD2's implementation gaps—mandating API quality, fraud sharing, and direct system access for non-banks.

Payment Rails: Feature Comparison Table (NEW)

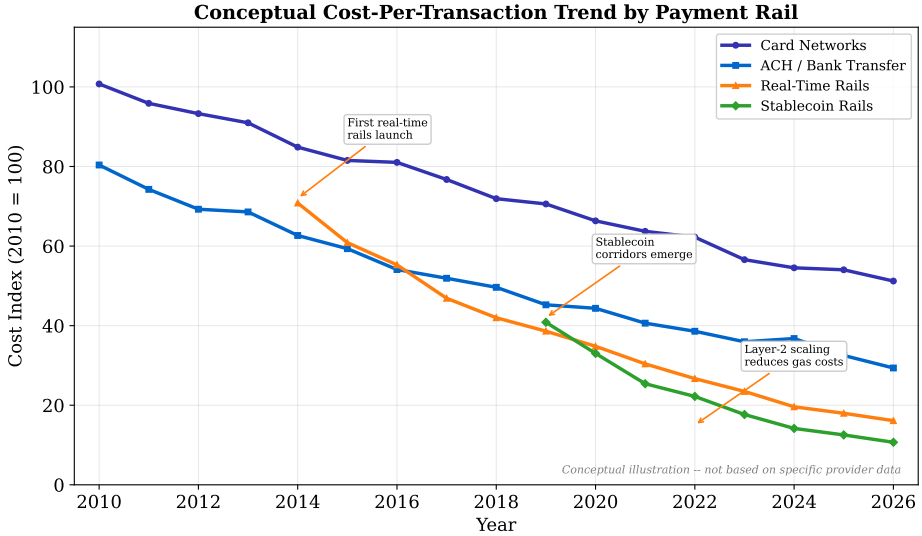
Rail	Settlement	Availability	Cost (%)	Data Richness	Finality	Reversibility
Card Network	T+1 to T+2	24/7	1.5–3.0	Low	Yes (T+2)	Chargebacks
ACH/SEPA	T+1 to T+2	Business hours	<0.5	Medium	Yes (T+1)	Limited
RTGS	Seconds	24/5 (varies)	<0.1	Medium	Immediate	No
Instant Payment	Seconds	24/7/365	<0.2	High (ISO 20022)	Immediate	Conditional
Open Banking A2A	Seconds	24/7/365	<0.1	High	Immediate	Limited
Stablecoin	Seconds	24/7/365	0.3–1.5	Variable	On-chain	No
BNPL	T+0 (merchant)	24/7	3.0–8.0	Medium	Merchant paid	Consumer defaults

Key observations:

- RTGS and instant rails combine low cost, immediate settlement, and 24/7 availability
- Card networks remain dominant due to ubiquity and consumer protections (chargebacks)
- BNPL has the highest merchant cost but drives conversion uplift

Tabular comparison reveals tradeoffs: instant rails win on cost and speed, but cards win on acceptance and consumer protection.

Cost-Per-Transaction Trends



The Coordination Problem — Network Effects and Resistance

Network Effects in Payments:

- A payment rail is only useful if both sender and receiver can use it
- Value scales with n^2 (Metcalfe's Law)—but so does the coordination cost
- Early adopters bear cost, late adopters get the benefit
- Result: chicken-and-egg problem

Incumbent Resistance:

- Banks earn interchange revenue from card networks
- Real-time A2A payments bypass this revenue stream
- Rational for incumbents to delay or under-invest in APIs
- Regulatory mandates (PSD2, UPI) overcome this

Payment rails face a classic chicken-and-egg problem: senders will not join until receivers are on board, and vice versa.

The Coordination Problem — Strategies and Adoption Patterns

Strategies to Overcome Coordination Failure:

- **Regulatory mandate:** Government requires participation (EU: SCT Inst, India: UPI)
- **Public infrastructure:** Central bank builds and operates the rail
- **Killer use case:** One application drives mass adoption (e.g., QR-code merchant payments)
- **Subsidy:** Temporarily zero-fee to build volume, then monetize

Lessons from Adoption Patterns:

- Mandates accelerate adoption by 3–5 years
- Mobile-first design drives retail uptake
- B2B adoption lags retail by 2–3 years
- Interoperability across schemes remains unsolved

Payment adoption is a coordination game—technology alone is insufficient without regulatory push or a killer use case.

Per-Transaction Cost Comparison Across Rails

Payment Rail	Merchant Cost	Settlement	Availability	Data Richness
Card Network	1.5–3.5%	T+1 to T+3	24/7	Low (4-party model)
ACH / SEPA Batch	$<0.5\%$	T+1 to T+2	Business days	Medium
Instant Payment Rail	$<0.2\%$	Seconds	24/7/365	High (ISO 20022)
Open Banking (A2A)	$<0.1\%$	Seconds–minutes	24/7/365	High
Stablecoin Corridor	0.3–1.5%	Seconds–minutes	24/7/365	Variable
BNPL	3–8% MDR	T+0 (merchant)	24/7	Medium

Key Observations:

- Instant rails and A2A are cheapest per transaction
- Card networks remain dominant due to consumer protection and ubiquity
- BNPL is the most expensive for merchants but increases conversion

Per-transaction cost varies by an order of magnitude across rails—but this is only part of the total cost picture.

Total Cost of Ownership Across Payment Rails

Beyond Per-Transaction Fees:

- Per-transaction fee is only one component
- Integration cost, fraud liability, chargeback handling
- Reconciliation effort (high for legacy, low for ISO 20022)
- Cash-flow impact of settlement timing

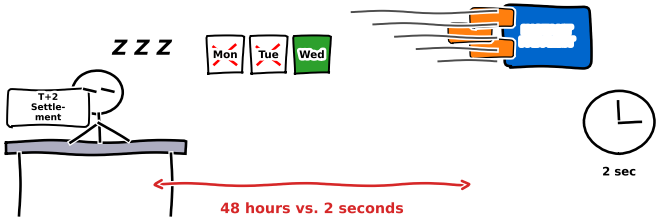
Hidden Cost Drivers:

- API integration and maintenance overhead
- Regulatory compliance per rail
- Failed payment rates and retry costs
- Dispute resolution and customer support

Bottom line: The cheapest rail per transaction is not always the cheapest in total cost of ownership. Merchants must evaluate integration, fraud, reconciliation, and cash-flow impact holistically.

Total cost of ownership includes integration, fraud, reconciliation, and cash-flow impact—not just the per-transaction fee.

One Last Thought...



Real-time settlement isn't the future—it's already here. Legacy systems just haven't woken up yet.

Sometimes the best way to remember a concept is to laugh about it.

Key Takeaways

1. **RTGS and instant payment schemes** compress settlement from days to seconds, but require liquidity buffers and coordinated adoption
2. **ISO 20022** is the messaging standard that enables straight-through processing, reducing manual reconciliation and exception handling
3. **Stablecoin corridors** offer an alternative to correspondent banking, compressing cost and time—but on/off-ramp friction and regulation remain barriers
4. **BNPL shifts cost** from consumer to merchant via higher MDR; the provider bears credit risk and operates on thin margins
5. **Payment orchestration** intelligently routes transactions across multiple rails to optimize cost, speed, and success rate
6. **Open banking (PSD2/PSD3)** mandates API access, enabling account-to-account payments that bypass card-network fees entirely
7. **Adoption is a coordination problem:** technology alone is insufficient—regulatory mandates, public infrastructure, or killer use cases are needed to reach critical mass

Cost compression is technically possible today—the binding constraints are coordination, regulation, and incumbent incentives.

Real-Time Payments and Cost Compression

Core insight:	Technology *can* compress settlement from days to seconds
But:	Adoption faces coordination problems and incumbent resistance
RTGS / Instant Rails:	Real-time, gross, final settlement in central bank money
ISO 20022:	Structured messaging → straight-through processing
Stablecoins:	Bypass correspondent chains; on/off-ramp is the bottleneck
BNPL:	"Free" for consumers; merchants pay 3–8% MDR
Orchestration:	Intelligent routing across rails for cost/speed/reliability
Open Banking:	A2A payments bypass card networks at near-zero cost

Next: Lesson 1.4 — We examine how these building blocks combine in specific market segments and business models.

The gap between what is technically possible and what is widely adopted defines the opportunity space in digital payments.

Self-Check #3: Real-time Payments

Attempt these before turning the page.

1. [Understand] What does ISO 20022 add that ISO 8583 and legacy SWIFT MT messages cannot carry?
2. [Apply] A retailer accepts Klarna BNPL with a 5% MDR vs card at 2.2%. Klarna claims 30% higher AOV (average order value) and 20% lower cart abandonment. At $100 baseline AOV with 40% margin, compute which is more profitable.
3. [Analyze] Why would a bank oppose adopting FedNow/SEPA Instant even though customers demand it? Give two self-interest reasons.

Solutions hidden unless `\solutionstrue` is set before compiling.