

Module 6 Summary: The Infrastructure Problem

Prof. Dr. Joerg Osterrieder

Digital Finance — BSc Course

L1: Payment Rails

- ACH/SEPA: batch/DNS, hours–days, payroll and bill pay
- RTGS (Fedwire/TARGET2): real-time, large-value interbank
- SWIFT: messaging layer for cross-border instructions
- Gross (RTGS) vs. net (DNS) settlement trade-offs

L2: Core Banking Systems

- General ledger at the foundation; product engines above
- Legacy COBOL systems vs. cloud-native challengers
- Cloud migration trade-offs: latency, regulation, cost
- Global market: \$18B (2024), growing 10%+ annually

L3: API Economy

- REST, GraphQL, webhooks: when to use each
- API gateway as control plane of modern banking
- Banking-as-a-Service (BaaS) and embedded finance
- Security layers: OAuth 2.0, mTLS, rate limiting

L4: Next-Gen Infrastructure

- CBDCs: retail vs. wholesale, account vs. token-based
- RWA tokenization: bonds, equities, real estate on-chain
- Design choices: centralized ledger vs. DLT
- Programmable money: expiry dates, spending restrictions

Module 6 answers: What pipes carry money, how old are they, and what will replace them?

Settlement Types

RTGS (Real-Time Gross Settlement): Each transaction settled individually in real time. Zero counterparty risk, high liquidity cost.

DNS (Deferred Net Settlement): Transactions netted and settled in batches. Lower liquidity need, but settlement risk during the netting window.

Netting Efficiency

$$\text{Net Position}_i = \sum_j \text{Payments Received}_{j \rightarrow i} - \sum_j \text{Payments Sent}_{i \rightarrow j}$$

Netting reduces total liquidity required. Example: if A owes B €100 and B owes A €80, only €20 settles.

Core Banking Architecture (5-Layer Model)

Channels (top) → API Gateway → Business Logic → Product Engines → General Ledger (bottom).

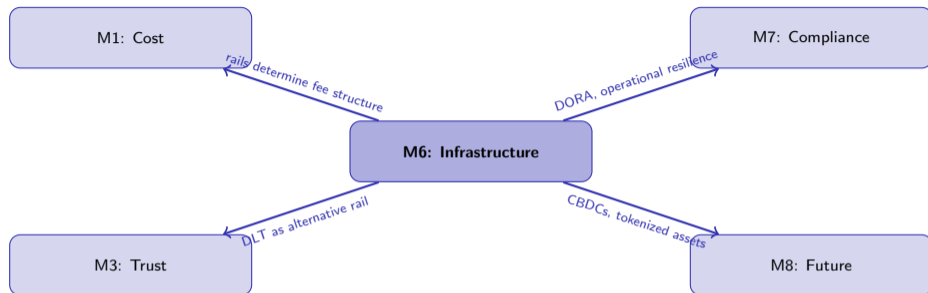
Every payment, balance, and interest calculation flows through this stack.

CBDC Design Taxonomy

Four design dimensions: (1) Retail vs. Wholesale, (2) Account-based vs. Token-based, (3) Centralized vs. Distributed ledger, (4) Intermediated vs. Direct access.

Infrastructure is invisible to customers but determines the speed, cost, and resilience of every financial transaction.

Connections to Other Modules



- **Infrastructure** → **Cost (M1)**: Payment rail choice (RTGS vs. DNS vs. card network) determines the fee structure merchants and consumers face
- **Infrastructure** → **Trust (M3)**: Distributed ledger technology is both a competitor to and a potential replacement for centralized payment rails
- **Infrastructure** → **Compliance (M7)**: DORA mandates operational resilience for financial infrastructure; cloud outsourcing requires regulatory approval
- **Infrastructure** → **Future (M8)**: CBDCs could replace the retail payment stack; RWA tokenization could replace settlement infrastructure

Today's infrastructure was built for yesterday's finance. The question is whether to renovate or rebuild.