Implementation Models for Banks in the Context of the Digital Euro

Zohaib Shaikh

December 7, 2025

Abstract

This thesis investigates technical implementation models for bank back-ends integrating with the Eurosystem's Digital Euro Service Platform (DESP). It analyses API options (REST/gRPC), data models (including pseudonymisation and alias management), Dedicated Cash Account (DCA) handling, and advanced features such as conditional payments and offline synchronization. Architectural patterns are evaluated across High-, Mid-, and Low-tier banks and for In-house, Vendor, and Hybrid implementation models. The result is a set of technical blueprints and best practices tuned for scalability, security, and maintainability.

Keywords: Digital Euro; DESP; DCA; CBDC; API integration; bank architecture

Contents

Chapter 1

Introduction

1.1 Background & Motivation

Provide context on the digital euro as a retail CBDC, the Eurosystem preparation phase (2023–2025), and why banks must plan for DESP integration.

1.2 Research Problem and Objectives

Define the integration problems (interoperability, privacy, advanced features) and state the thesis objectives.

1.3 Research Questions

List the research questions (data mapping, API trade-offs, conditional/offline impacts, cost-effectiveness by tier/model).

1.4 Scope and Limitations

Define the technical scope (bank back-ends for euro-area banks) and limitations (reliance on public materials, evolving DESP specs).

Chapter 2

Background on the Digital Euro

2.1 Conceptual Framework

Definitions: DESP, PSPs, DEANs, DCA, conditional payments.

2.2 Eurosystem's Digital Euro Project Evolution

Timeline and milestones: investigation phase (2021–2023), preparation (2023–2025), rulebook drafts, platform provider selection.

2.3 Key Components of the Digital Euro Infrastructure

Detailed descriptions of DESP, DCAs, privacy features, delegation to providers.

2.4 Role of Banks in the Ecosystem

Banks as access managers, alias providers, and integrators with existing payment rails.

Chapter 3

Literature Review

3.1 Overview of CBDC Implementation Globally

Compare approaches such as e-CNY and Sand Dollar and lessons learned.

3.2 Technical Standards and Guidelines

Summarise ECB rulebook guidance, interoperability recommendations, and standard reuse.

3.3 Cost and Feasibility Studies

Discuss PwC Digital Euro Cost Study (2025) and national IT surveys (e.g., Italian survey).

3.4 Existing Research on Bank Integration

Review EBA guidance, IMK Study, and DESP experimentation outputs.

3.5 Gaps in Literature

Identify where the thesis contributes (API blueprints, tier-specific models, offline handling).

Chapter 4

Methodology

4.1 Research Design

Qualitative and comparative design: document analysis, modeling, case studies for bank tiers.

4.2 Data Sources

Primary: ECB reports and innovation platform outputs. Secondary: cost reports and academic studies.

4.3 Analytical Framework

Criteria: performance (latency), security (privacy techniques), maintainability (scalability). Modeling + scenario analysis.

4.4 Tools and Techniques

UML/sequence diagrams, cost-benefit analysis, simple simulations; mention software (e.g., PlantUML, JMeter) if used.

4.5 Ethical Considerations

Data privacy in examples, informed assumptions, citation of sources.

Chapter 5

Technical Architecture of the DESP and Bank Back-Ends

5.1 DESP Overview

Core platform functions: settlement, data segregation, support for offline/conditional features.

5.2 Bank Back-End Systems

Mapping typical core banking, ledger systems, payment gateways, target interfaces.

5.3 Data Models and Mapping

Example mappings: internal customer IDs -> pseudonymised DEANs, alias tables, example ER diagrams (add figures).

5.4 Liquidity Management via DCAs

DCA lifecycle, waterfall configurations, funding automation and reconciliation.

Chapter 6

Integration Pathways and APIs

6.1 API Selection and Protocols

Compare REST and gRPC; include example request/response skeletons (JSON and Protobuf) in the appendix.

6.2 Integration Challenges

Authentication/authorization approaches, idempotency, retries, schema evolution.

6.3 Advanced Functionality Processing

6.3.1 Conditional Payments

Reservation, hold, settlement flows and edge cases.

6.3.2 Offline Transaction Synchronization

Design patterns for local secure stores, replay protection, reconciliation and conflict resolution.

6.4 Security and Compliance

Privacy-preserving practices, encryption-at-rest/in-transit, monitoring and incident response considerations.

Chapter 7

Evaluation of Architectural Patterns

7.1 Microservices vs. Monolithic Integration

Discuss trade-offs: deployment cadence, scaling, observability, latency.

7.2 Impact Assessment

Proposed KPIs and testing approaches: latency percentiles, throughput, mean time to recovery, maintenance cost estimates.

7.3 Hybrid Approaches

Staged migrations, vendor-managed components with in-house orchestration.

Chapter 8

Implementation Models by Bank Tier

8.1 High-Tier Banks (Large, Advanced IT)

Blueprint: in-house microservices, high-performance connectors, advanced DCA automation.

8.2 Mid-Tier Banks (Regional, Moderate Maturity)

Blueprint: hybrid model leveraging vendor gateways plus in-house DCA logic.

8.3 Low-Tier Banks (Small, Basic Infrastructure)

Blueprint: vendor-managed services, simplified APIs, focus on compliance and offline support.

8.4 Comparative Analysis

Insert a comparative table (cost, timeline, technical risk) and narrative analysis.

Chapter 9

Technical Blueprints and Best Practices

9.1 Blueprints Development

Instructions for diagrams: component diagrams, sequence flows, data mapping charts.

9.2 Best Practices

Scalability: cloud-hybrid; Security: strong key management, pseudonymisation; Maintainability: API versioning and clear contracts.

9.3 Sensitivity Analyses

Suggested scenarios: peak transaction volumes, provider outages, rulebook changes.

9.4 Policy Implications

Practical suggestions for ECB standardisation and testing frameworks to ease bank integrations.

Chapter 10

Conclusion

10.1 Summary of Findings

Concise summary of the main technical conclusions.

10.2 Contributions

Highlight blueprints, tiered recommendations and gaps filled in literature.

10.3 Recommendations

Practical guidance for banks: adopt hybrid models for cost-effectiveness, invest in modular APIs.

10.4 Future Research

Directions: blockchain alternatives, cross-border CBDC interoperability, post-2025 pilot evaluations.

Appendix A

Appendices

A.1 Sample API Schemas

A.2 Simulation Parameters and Results

A.3 Extended Tables and Figures