Automating Trust: A Technical Framework for Murabaha using Blockchain and Cloud Integration

Hamza Bensliman
Department of Economics and Management
Sidi Mohamed Ben Abdellah University
Fez, Morocco
email: [hamza.bensliman@usmba.ac.ma]

***Abstract*—This paper presents a technical prototype for digitizing Islamic Murabaha contracts using a hybrid approach that combines permissioned ledger concepts with cloud-based centralization. We address the challenge of "trust" in banking transactions by utilizing an immutable audit trail for the three stages of Murabaha: Wa'd (Promise), Qabd (Possession), and Sale. The proposed system integrates a Python-based frontend (Streamlit) with Google Cloud Platform (GCP) acting as a core banking simulation, ensuring real-time data synchronization and transparency. This framework aims to reduce administrative latency and enhance Sharia-compliance auditability.**

***Index Terms*—Islamic Finance, Blockchain, Smart Contracts, Murabaha, Cloud Integration, Streamlit.**

I. Introduction

Digital transformation in Islamic Finance faces unique challenges, primarily the need to ensure Sharia compliance in automated transactions [3]. Traditional Murabaha contracts rely heavily on manual verification to ensure the sequence of "Constructive Possession" (Qabd), strictly following standards set by AAOIFI [1]. This paper explores how modern web technologies can automate this process.

II. Problem Statement

Current banking systems lack transparency in the sequence of Murabaha transactions. There is a risk of selling the asset before the bank actually owns it, which violates Sharia principles regarding risk-sharing and ownership [4]. Furthermore, manual auditing is time-consuming and prone to human error.

III. Proposed Solution

We propose a hybrid system named "ShariaChain" (or Automating Trust) that utilizes blockchain principles to create an immutable ledger [2]. The system utilizes:

- **Streamlit Interface:** For officer interactions.
- **Google Cloud Platform (GCP):** As a secure, centralized database representing the Core Banking System.
- **Hashing Algorithms:** To create a unique digital fingerprint for each transaction stage (Wa'd, Qabd, Sale).

IV. Technical Implementation

The prototype was built using Python. The connection to the database was established using Google Service Accounts and OAuth2 authentication.

A. System Architecture

The system connects a frontend application to a Google Sheet database via an API key, simulating a permissioned ledger where only authorized officers can record transactions.

B. Cloud Integration

We encountered challenges in integrating local Python scripts with cloud-based storage. These were resolved by implementing secure JSON key authentication and utilizing the `gspread` library for real-time synchronization.

V. Conclusion

This project demonstrates that it is possible to build a low-cost, secure, and transparent Murabaha auditing system using open-source tools. Future work will focus on migrating the database to a decentralized Ethereum testnet to fully align with blockchain economy principles [2].

References

[1] Accounting and Auditing Organization for Islamic Financial Institutions (AAOIFI), "Shari'ah Standard No. 8: Murabaha," Manama, Bahrain, 2017.
[2] M. Swan, *Blockchain: Blueprint for a New Economy*. O'Reilly Media, Inc., 2015.
[3] M. H. Kamali, "FinTech and the Future of Islamic Finance," in *Islamic FinTech: Insights and Solutions*, Springer, 2021, pp. 15–35.
[4] Z. Iqbal and A. Mirakhor, *An Introduction to Islamic Finance: Theory and Practice*, 2nd ed. John Wiley & Sons, 2011.