Research seminar: Decentralised Message Protocol Built on Top of Blockchain

Mentor: Aleksandar Tošić

Student: Ekaterina Bochvaroska 89252018

Faculty of Mathematics, Natural Sciences and Information Technologies, 2026

Topic and Research Context

Messaging systems are critical digital infrastructure

Used for:

- social communication
- economic coordination
- political interaction

Most systems today rely on centralized architectures End-to-end encryption protects message content, but it does not by itself remove the architectural dependence on central servers. Delivery, availability, and the communication flow passes through operators

Motivation

- Centralized messaging systems create several risks:
  - **Single points of failure**: service disruption or infrastructure failure can affect the whole platform.
  - **Surveillance and censorship exposure**: communication passes through operator-controlled servers.
  - **Metadata leakage**: even when message content is encrypted, providers often still observe communication patterns, timing, and participants.
  - **Limited fault tolerance**: failures cannot be dynamically rerouted across independent peers.

Why decentralization helps?

Decentralization helps bydistributingresponsibilityacross many independent participants rather than one service operator.

Potential benefits:

- no single infrastructure & reduced dependence on one provider
- greater fault tolerance
- stronger resistance to censorship
- possibility of user or community-operated infrastructure

Existing systems

Current systems show three main architectural strategies:

1. Centralized encrypted messaging
2. Decentralized or metadata-resistant messaging
3. Federated social communication

These systems prove that parts of the problem can be solved, but not all of it at once.

Signal

- strong end-to-end encryption
- privacy-focused design
- still relies on server infrastructure for service operation

Existing systems

Session

- decentralized private messenger
- onion routing to hide IP/network metadata
- messages routed through Session nodes
- temporary encrypted storage on the network

Mastodon

- decentralized
- federation through ActivityPub
- independent servers interoperate with no single global server

None of them combine decentralization, scalability, efficient retrieval, and verifiability in a unified way.

Blockchain theory

A blockchain is a distributed network that maintains an ordered sequence of blocks linked by hashes.

Each block typically contains:

- a reference to the previous block
- a batch of transactions
- cryptographic commitments to the block contents

The key idea is that blockchains already solve one part of the systems problem: they coordinate many distributed nodes and propagate structured data.

Blockchain theory

For blockchain transactions, immutability is essential because transactions represent state changes, transfers, or contract execution.

For messages, the requirement is different:

- the message content itself does not always need permanent on-chain storage
- what matters is a verifiable record that the message existed and was linked to the network history

We only want messages to inherit enough blockchain structure to become verifiable.

Main idea

Using existing blockchain networks for:

- propagation support,
- cryptographic anchoring,
- and historical ordering,

while storing encrypted message payloads alongside the chain rather than directly inside it.

Research Objective

Existing decentralized or blockchain-based messaging approaches attempt to reduce dependence on centralized servers, but many of them introduce new limitations:

- storing complete message data directly on-chain,
- requiring full replication across all nodes,
- lacking efficient retrieval mechanisms,
- or providing only limited experimental validation.

Retention rates

A decentralized messaging system cannot assume that all nodes will store all messages forever. Why?

- storage is costly
- bandwidth is costly
- long-term free retention is unrealistic
- heterogeneous nodes have different capacities

Therefore nodes should advertise retention rates or retention windows:

- some keep only recent encrypted message blocks
- archive nodes may keep all history

Research Questions

1.How can message storage be separated from blockchain verification without losing integrity?

2.How can queries be **executed efficiently** in a decentralized network with **heterogeneous** storage capacities?

3.Can a retrieval protocol based on query decomposition and parallel dispatch maintain stable latency under concurrent workloads?

Protocol idea

Proposed protocol:

- messages are encrypted client-side
- encrypted messages are stored temporarily in extension blocks
- the blockchain stores a cryptographic pointer from each block to its extension block
- the chain provides ordering and anchoring
- the extension layer provides scalable message storage
- nodes retain extension blocks according to retention policy
- older extension blocks may be pruned, while the chain record remains

Blockchain-agnostic design

The protocol is designed to be blockchain-agnostic.

All it needs is:

- a hash-linked chain
- an authenticated reference from the block to an extension block

This means it can be adapted to:

- native block formats,
- or smart-contract platforms through a simple contract that stores the pointer or commitment.

Thank You!

Questions