

Streamlet Project – Phase 1 Report

Group 6

Members: Enzo Chatalov 54414, Yuliya Sasinskaya 66416, Matouš Macák 66722

Objective

Implement the basic (failure-free) version of the **Streamlet consensus algorithm**, where nodes propose, vote, and finalize blocks in epochs.

Language and Tools

- **Language:** Python 3
- **IDE:** Visual Studio Code
- **Libraries:** `socket`, `threading`, `pickle`, `hashlib`, `time`, `collections`

Python was chosen for its simplicity and strong support for network programming.

System Behavior

- Each node runs independently, maintaining its local blockchain and sets of votes, notarized, and finalized blocks.
- Epochs advance every $2 * \Delta$ seconds.
- The leader of each epoch proposes a new block referencing the latest notarized parent.
- Other nodes vote for the proposal if it extends their longest notarized chain.
- When a node observes a majority of votes ($> n/2$) for a block, it notarizes it.
- Seeing three consecutive notarized blocks triggers finalization of the middle block.

Project Summary

The system is divided into three main components:

- **Common:** defines transactions, blocks, and messages.
- **Network:** manages server communication and multicast broadcasting.
- **Protocol:** implements the Streamlet logic for proposing, voting, and finalizing blocks.

Result

All members contributed collaboratively to the design, coding, and testing of the system, dividing responsibilities across networking, communication, and consensus logic. The Phase 1 implementation successfully demonstrates multi-node communication, block proposal, voting, and finalization under normal network conditions. This version serves as the foundation for Phase 2, which will introduce crash recovery and fault tolerance.