Algorithms and Distributed Systems 2022/2023 (Lab02)

MIEI - Integrated Master in Computer Science and Informatics
MEI – Master in Computer Science and Informatics
Specialization block

João Leitão (jc.leitao@fct.unl.pt)
Alex Davidson (a.davidson@fct.unl.pt)

NOVA SCHOOL OF
SCIENCE & TECHNOLOGY

Introduction

- I’m Alex
- Research in cryptography, and privacy-preserving Internet protocols
- I previously worked in Industry as an academic researcher and cryptography engineer

Class structure:

- Project (Phase 1) Specification
- Explanation of different stages of development
- Some ideas to get you started

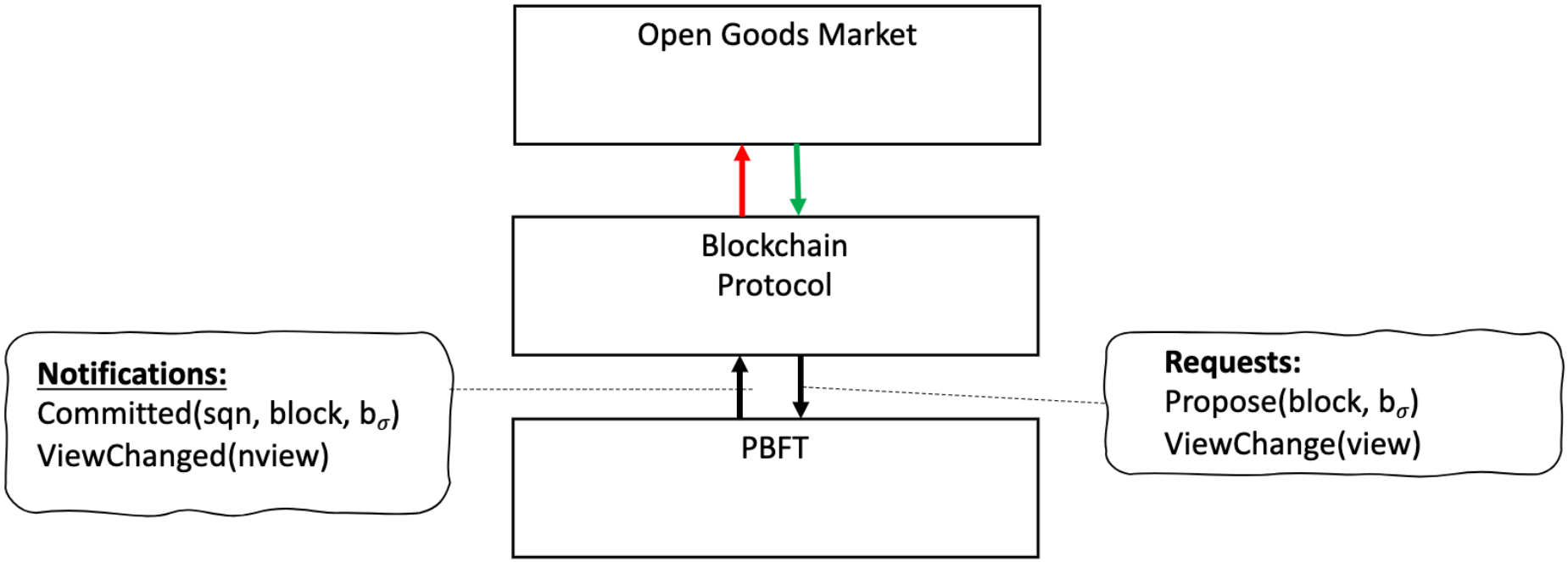
Project (Phase 1)

- PDF on CLIP and here: https://github.com/UNL-MEI-CSD/lab-materials

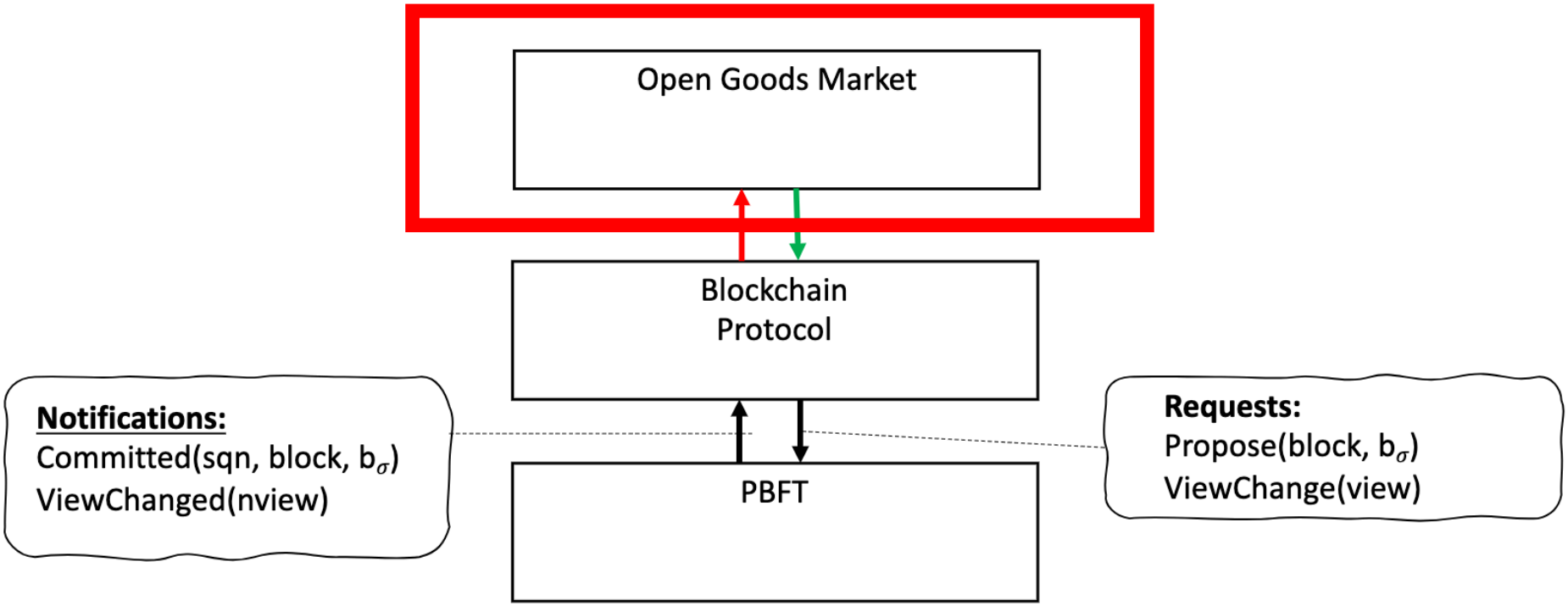
- Overall goal: Build an open marketplace application that allows users to advertise selling/buying products at minimum/maximum prices.
  - The application should automatically match compatible buy/sell offers

- Stages:
  - 1) Implement a PBFT protocol for ensuring fault-tolerance
  - 2) Implement a blockchain that manages a distributed ledger for storing different requests
  - 3) Implement an application-layer interface for interacting with the marketplace

Project

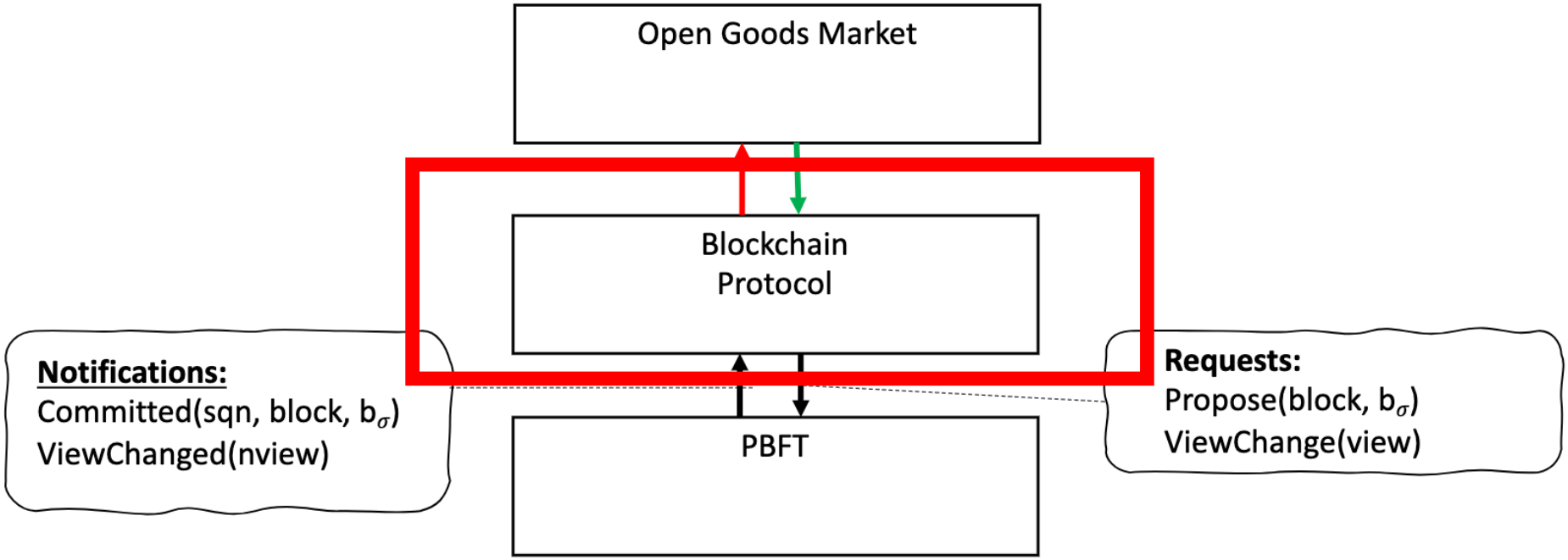

Open Goods Market
Blockchain Protocol
PBFT
Notifications:
Committed(sqn, block, b_σ)
ViewChanged(nview)
Requests:
Propose(block, b_σ)
ViewChange(view)

Project



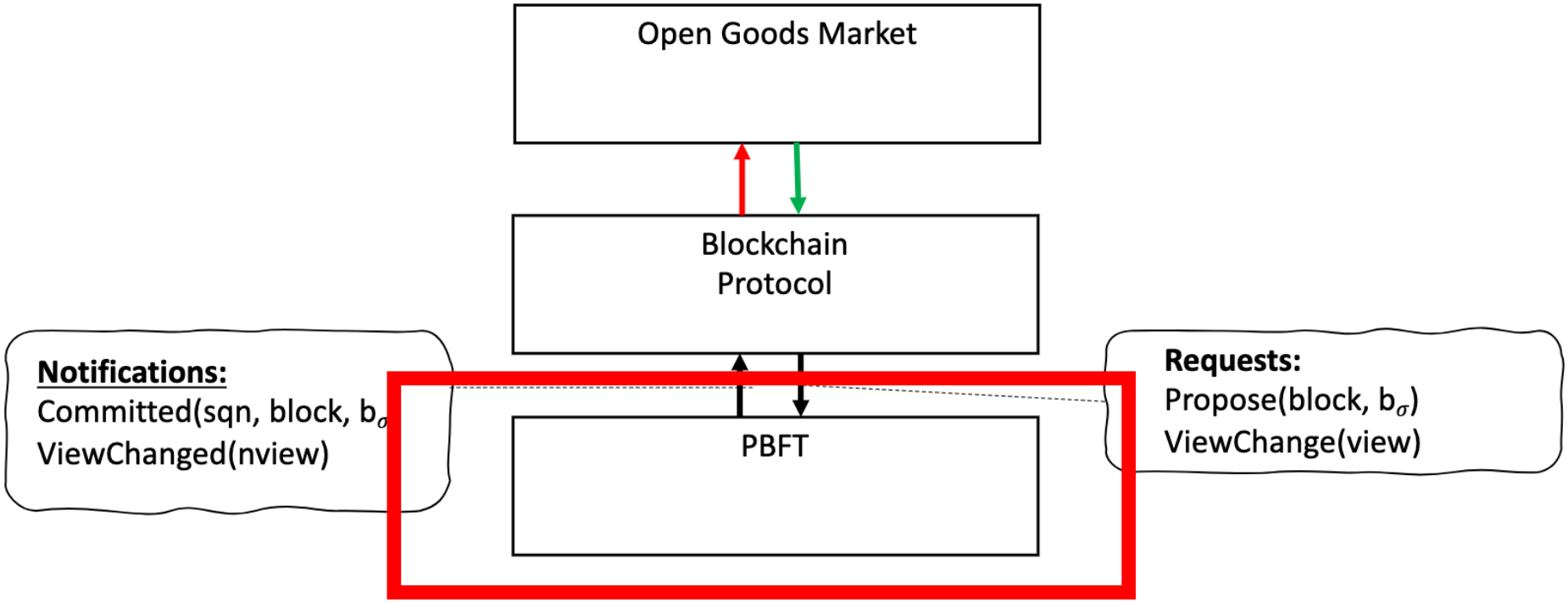
- Application logic:
  - Interactions with clients
  - Maintaining state about offers/requests/matches

Project



- Blockchain Protocol
  - Replicated ledger for storing all client operations
  - Each block is then proposed to the PBFT layer to establish ordering

Project



- Practical Byzantine Fault-Tolerant network
  - Facilitates agreement on which blocks to add to ledger
  - Requires implementing “Propose”, “ViewChange” messages

Evaluation criteria

- Based on multiple factors:
  - Written report
  - Correctness, performance, and quality of code
- See PDF for more details

Lab sessions

- Each lab session will aim to tackle a small part of this overall system
  - You will be given a task to complete each week
  - Over the weeks, each individual task will combine

- In these first few weeks, we will be focusing on building the PBFT (stage one)

Resources

- Javadoc: https://asc.di.fct.unl.pt/~jleitao/babel/
- Base code: https://classroom.github.com/a/-w4hjyDu
- Materials for labs: https://github.com/UNL-MEI-CSD/lab-materials (will also upload to CLIP)
- Instructions for running code in README
  - You will need to install maven

Today's Assignment

- Picking up the code that was offered, you will have a simple structure for the Project with lots of incomplete code.
  - The application is just for the purposes of example and testing.
  - PBFTProtocol is a highly incomplete class for materializing PBFT
  - You already have incomplete messages for the main messages (SignedProtoMessages) of PBFT (they need state and serializers)

Today's Assignment

- The current code is just an **example**, but each node can generate random blocks (that are signed) and pushed to PBFT using a method (this interface will not be employed in the final Project).

- That interface generates a ProposeRequest which is delivered to the local PBFT protocol.

- The PBFT protocol already has a sequence number (seqn) set initially to zero and a view but no view number (you have to add that).

Today's Assignment

- The idea is that you use this base code to implement PBFT, by implementing behaviours of the protocol, namely:
  - Sending and receiving messages of the protocol (you can consider the ProposeRequest as the entry point, at which point PrePrepareMessages can be generated -- after checking the signature of the block)
  - When receiving messages remember to verify if signatures are correct.

Today's Assignment

- Suggestion:
  - Start by simply creating and validating the messages employed by PBFT to achieve agreement (without considering the existence of a leader)
  - Evolve the protocol state to materialize the right behaviours of the protocol for handling each message.

- For the rest of the week: finish the implementation of PBFT, including all behaviours of the protocol for all messages.

Remember

- Keep referring back to the lecture notes and other materials for understanding the PBFT construction
- I am here to help
- It’s okay if things are not clear, try to get a plan of the PBFT construction before you implement it

Last things

- My office hours are 14:00 - 16:30, P2;17
- My email: a.davidson@fct.unl.pt
- Any questions: please ask!