Blockchain for Education

Project Overview

Zack Holmberg

Introduction

Introduction

How we got here

- Took a course on Distributed Computed, wrote a “Blockchain For Beginners” paper for my final project with a very basic proof-of-concept blockchain in python
- Prof thought it was cool
- Talked with him about how it’d be cool to have a blockchain assignment in that course
- **Problem**: Scope
    - A full-scale, Peer-to-Peer (P2P) blockchain is composed of many different parts, including hashing, distributed communication, concurrency, etc. Material not isolated to that course
- Prof suggested doing the Honours Project to develop something that could solve this problem

Introduction

How we got here

V1

- V1 Proposal ✔️
- V1 Requirement gathering with stakeholders ✔️
- V1 Architecture design 💩
- V1 Architecture redesign ✔️
- V1 Implementation ✔️

V2

- V2 Proposal ✔️
- V2 Design ✔️
- V1 Redesign and refactor ✔️
- V2 Implementation ✔️
- V2 Release 📍

Introduction

Blockchain Review - General

- **Block:** A data structure consisting of various properties such as an index, a piece of data, a timestamp, etc.
- **The Chain:** A glorified linked list consisted of Blocks
- **Peer:** A node on the Peer-to-Peer (P2P) network, also referred to as a “Miner”
- **Proof:** Some sort of data representing a Peer’s right to add a new Block to the Chain
- **Mining:** A Peer’s process of finding a new Proof, earning the Peer a reward
- **Consensus:** The process of Peers agreeing which Blocks should be appended to the chain and which copy of the chain should be shared among the nodes
- **Blockchain:** The technology as a whole

Introduction

Blockchain Review - Implementation Specific

- **Proof of Work:** Requires a Peer to complete a certain amount of work in order to find a Proof
- **Proof of Stake:** Selects validators in proportion to their quantity of holdings in the associated cryptocurrency.
- **Digital Signature:** A digital signature is a mathematical scheme for verifying the authenticity of digital messages or documents.
- **Longest Chain:** The longest copy of the Chain amongst Peers is used as the “Master” Chain

Introduction

Blockchain Review

- **Formally**:

  An append-only, immutable, distributed and digital ledger composed of blocks containing data that are linked together by a digital chain.

- **Self-plug:** Blockchain for Beginners

Problem

Problem

- A blockchain is traditionally a highly coupled piece of software.
- If you wanted students to write the hashing functionality within a blockchain, we’d have to provide them with the rest of the blockchain code, which could be overwhelming and distracting.
- Solution?

Modular Blockchain

Modular Blockchain

Idea

- A blockchain peer comprised of different components, where the methods of each are defined by an interface

- Peer doesn’t need to care about underlying implementation

- Plug-and-play

- Goal:

  To develop a custom blockchain that can be used as a pedagogical tool that will help teach future students about blockchain technology, as well as other various topics already in the curriculum, such as data structures and algorithms, distributed systems, networks, etc.

Modular Blockchain

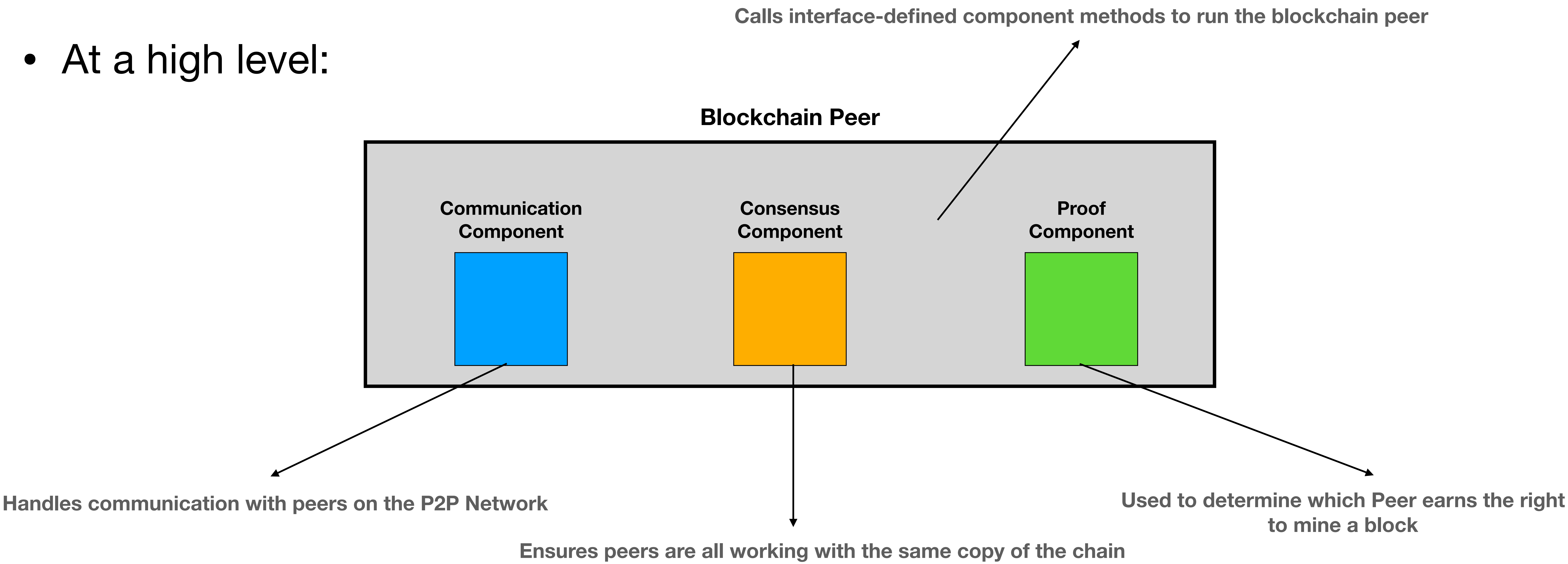
Idea - Applicable topics

- **Distributed Computing** - Implement communication between blockchain peers. Socket programming.
- **Computer Networks** - Peer-to-peer networking protocols and algorithms. Routing applications.
- **Data Structures and Algorithms** - Write the hashing functionality for the Blockchain's Proof of Work, implement the chain of blocks (basically a glorified linked list)
- Etc.

Modular Blockchain

V1 Design & Architecture

- At a high level:



Calls interface-defined component methods to run the blockchain peer

Handles communication with peers on the P2P Network

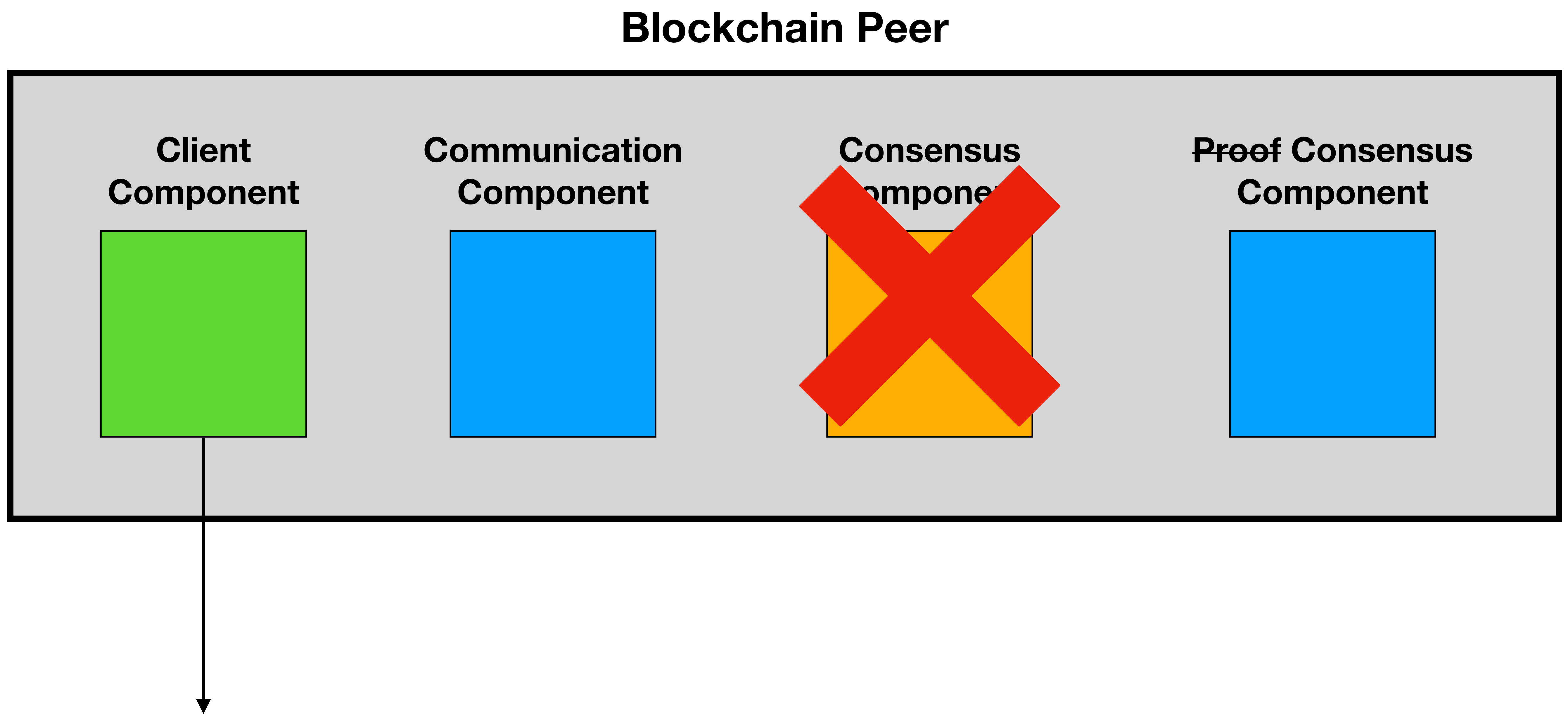
Ensures peers are all working with the same copy of the chain

Used to determine which Peer earns the right to mine a block

Modular Blockchain

V2 Design & Architecture

- At a high level:



Stores public and private keys. Implements methods that sign/verify transactions

Modular Blockchain

Implementation

- **Component implementations**
  - Communicator - Communication Component
  - ProofOfWork - Consensus Component
  - ProofOfStake - Consensus Component
  - Client - Client Component

Modular Blockchain

Design & Architecture

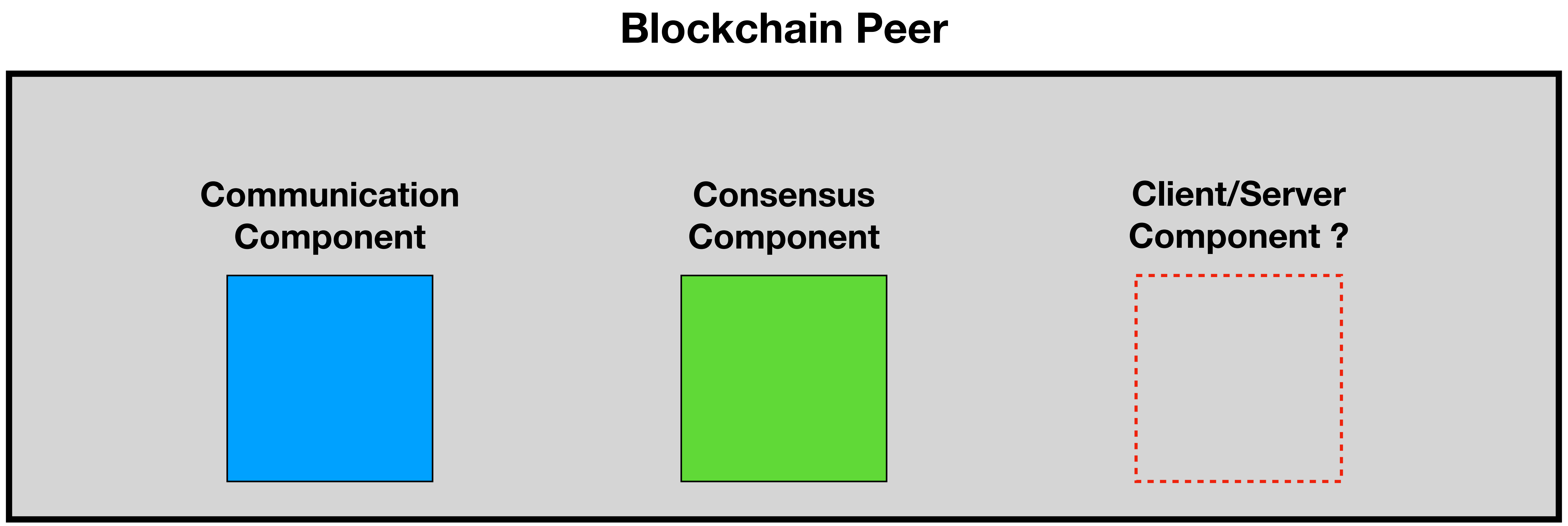
- At a lower level:

  lower level Architecture

Modular Blockchain

New transactions?

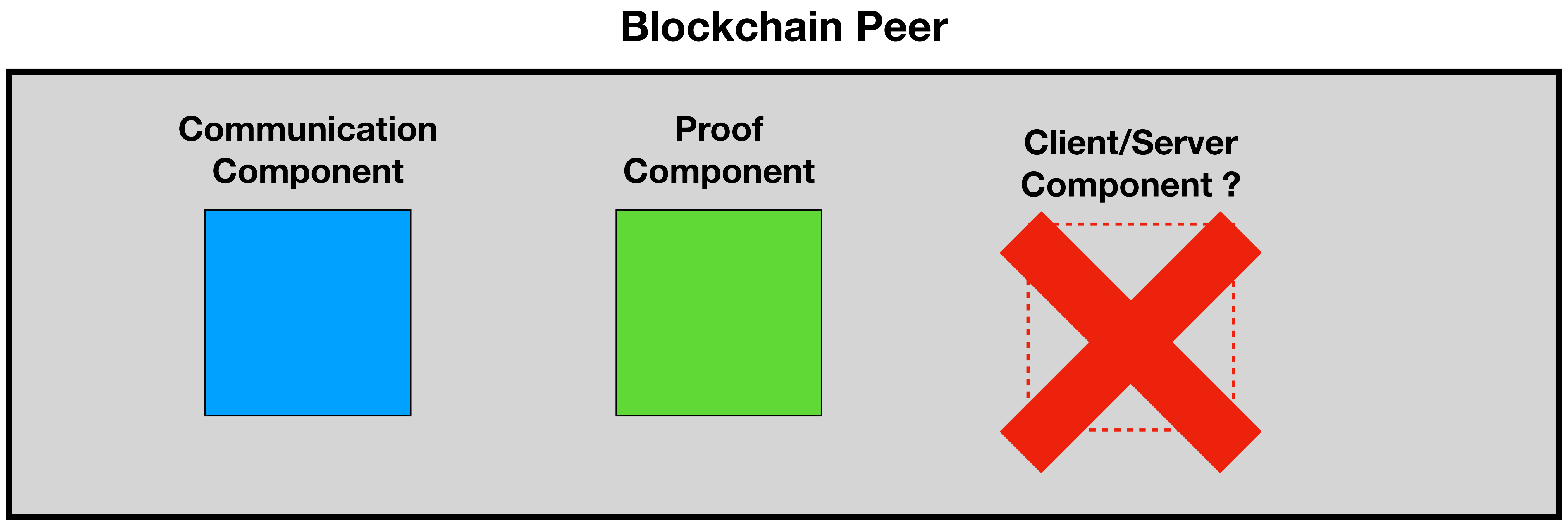
- How do we handle new transactions? Transaction Pool? HTTP Server?



Modular Blockchain

New transactions?

- Blockchain network is P2P - Don’t want to mix Client/Server architecture within the P2P network or else things are going to get very coupled, very quickly
- Complexity ⬆️
- Not ideal for students



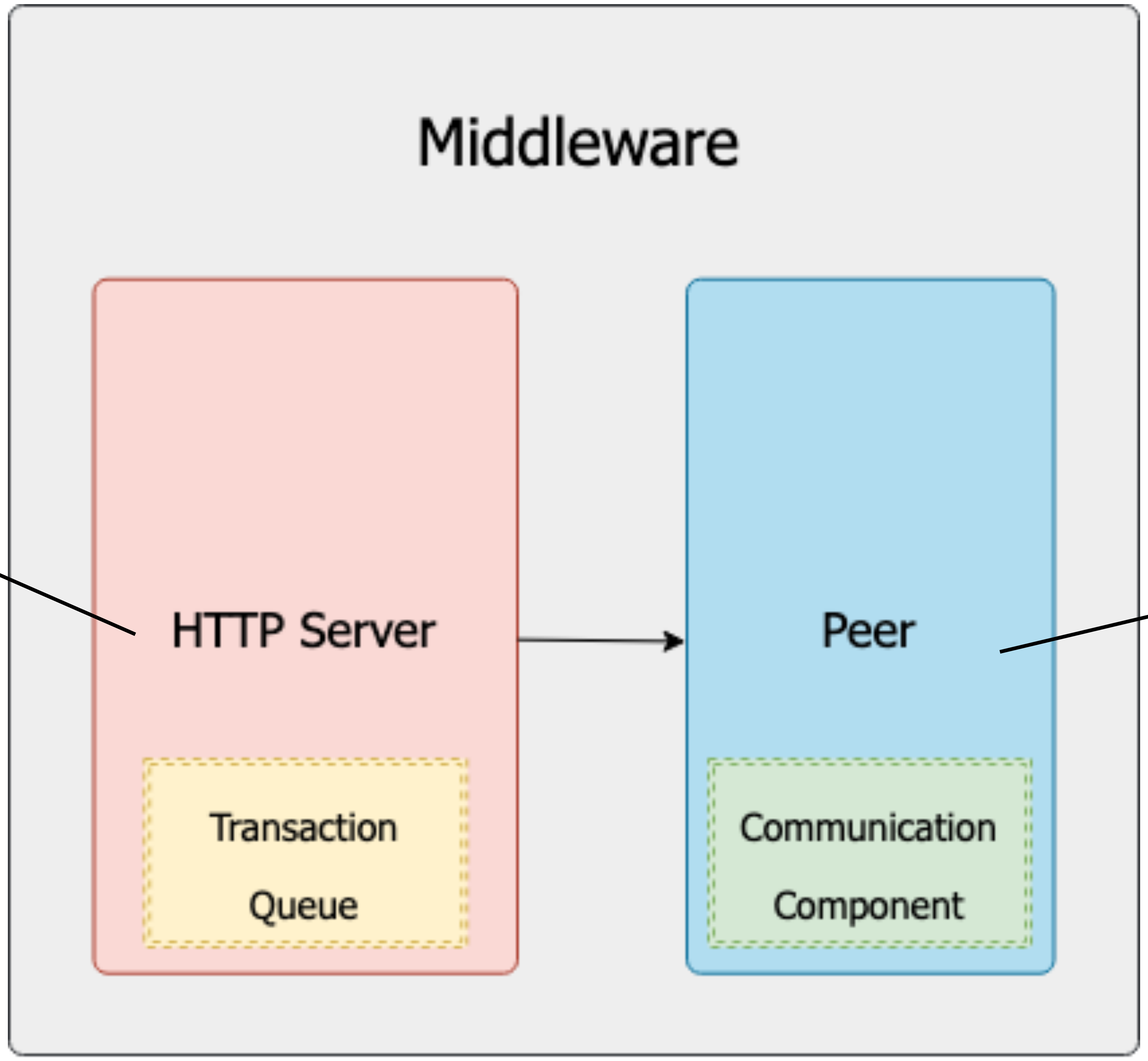
- Solution?

Middleware

Middleware

Design & Architecture

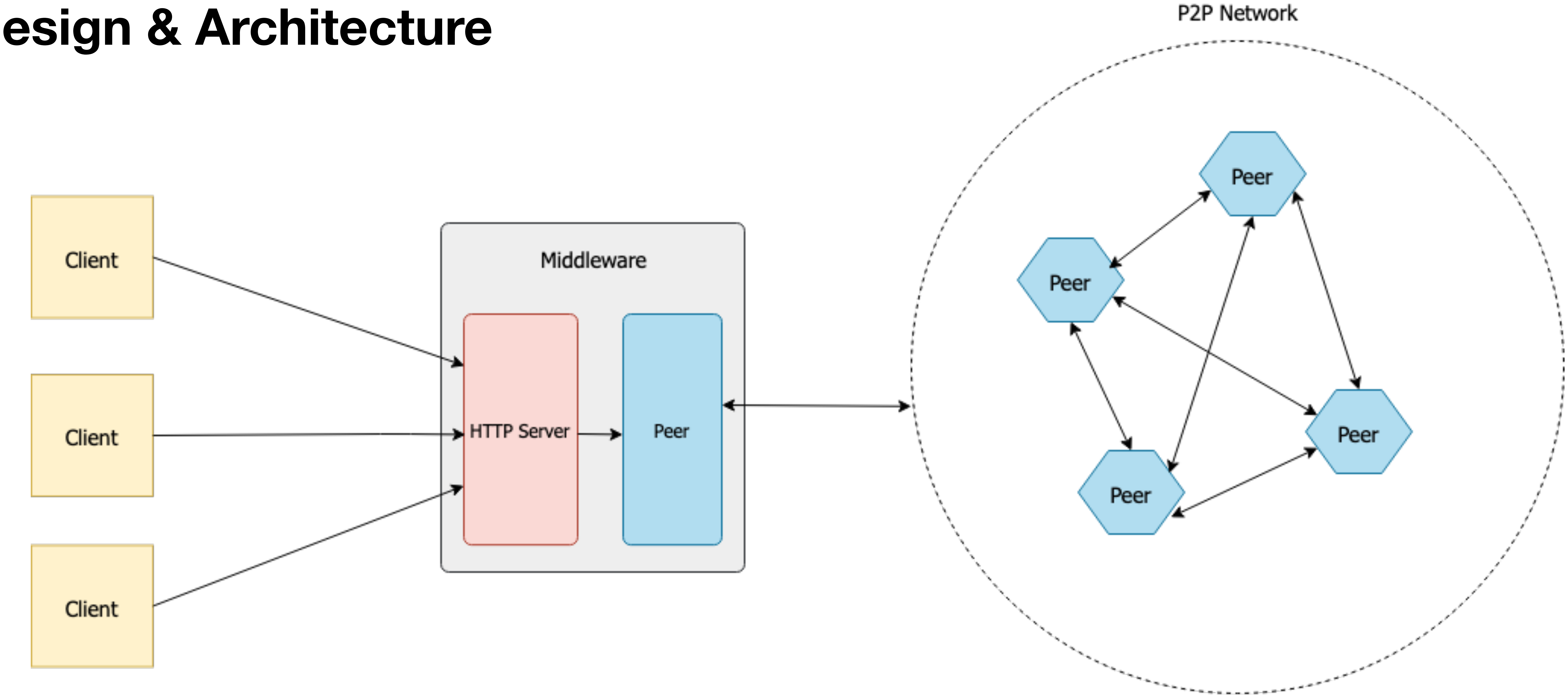
Receives HTTP POST requests from clients containing data representing a new transaction to be mined. Converts POST data to a Transaction and adds it to a Transaction pool



Sends/receives messages from blockchain peers, a part of the P2P network

Middleware

Design & Architecture



Modular Blockchain

Pros and Cons

+ Conceptually simpler

+ Blockchain peers are more cohesive and less coupled

+Modular design

− Middleware is a single point of failure

− The network is not partition tolerant, any peer not in the middleware’s partition will fail

V2

V2

New Work

- Component redesign and refactor
- Plug-n-play Proof of Stake
- Block validation
- Digital signing (new component)

Applicability

What’s the point?

Error Handling

- Writing functions that catch errors and pass them up
- Help students learn how to decide when to:
  - Swallow an error
  - Handle an error but continue running
  - Fail completely

What’s the point?

Debugging

- Understanding state and flow of execution
- Determining why communication between peers is broken
- Debugging different components and moving parts that make up a distributed system

What’s the point?

Networking/Distributed Systems

- Socket programming
- Sending/Receiving/Handling HTTP Requests
- Cookies
- Working with Multicast DNS/services
- Designing client/server/P2P distributed networks

What’s the point?

Software Development Principles

- Abstraction
- Modularity
- Testing
- Architecture Design

Future Work

Future Work

What’s next?

- Testing suite
- Smart contracts (a new component!)
- Run-time blockchain peer assembly
- Experimentation of this new pedagogical material in a real classroom setting

Thank you!

Repo Link