Blockchain for Education

Lunch ’n’ Learn

Zack Holmberg - December 22nd, 2020

Introduction

Introduction

How we got here

- Took a course about distributed computing, 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 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, possibly concurrency, etc. Material not isolated to that course
  - Revisit this problem later
- Prof suggested an honours project to develop something that could solve this problem

Introduction

How we got here

- Proposal ✓
- Requirement gathering with stakeholders ✓
- Architecture design 💩
- Architecture redesign ✓
- Implementation ✓
- Presentation 📍

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
- **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 with the rest of the blockchain code, which could be overwhelming and distracting.

- Solution?

Modular Blockchain

Modular Blockchain

Idea

- A blockchain peer is comprised of different components, where the methods of each are defined by an interface
- Doesn’t care about underlying implementation
- Plug-and-play

- Problem Statement:

  The goal of this project is 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. Could be TCP or UDP socket implementation
- **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

Idea - Applicable Assignments

- **COMP 3010 - Distributed Systems:** Blockchain exists, creating a peer to hold the blockchain. Could add consensus items, mining. Recovery. Implement an HTTP Server that handles clients sending new transactions.

- **COMP 4140 - Cryptography:** Applying hashing, Public/Private key implementations. Work with the blockchain, and design or implement a hash component to a running chain.

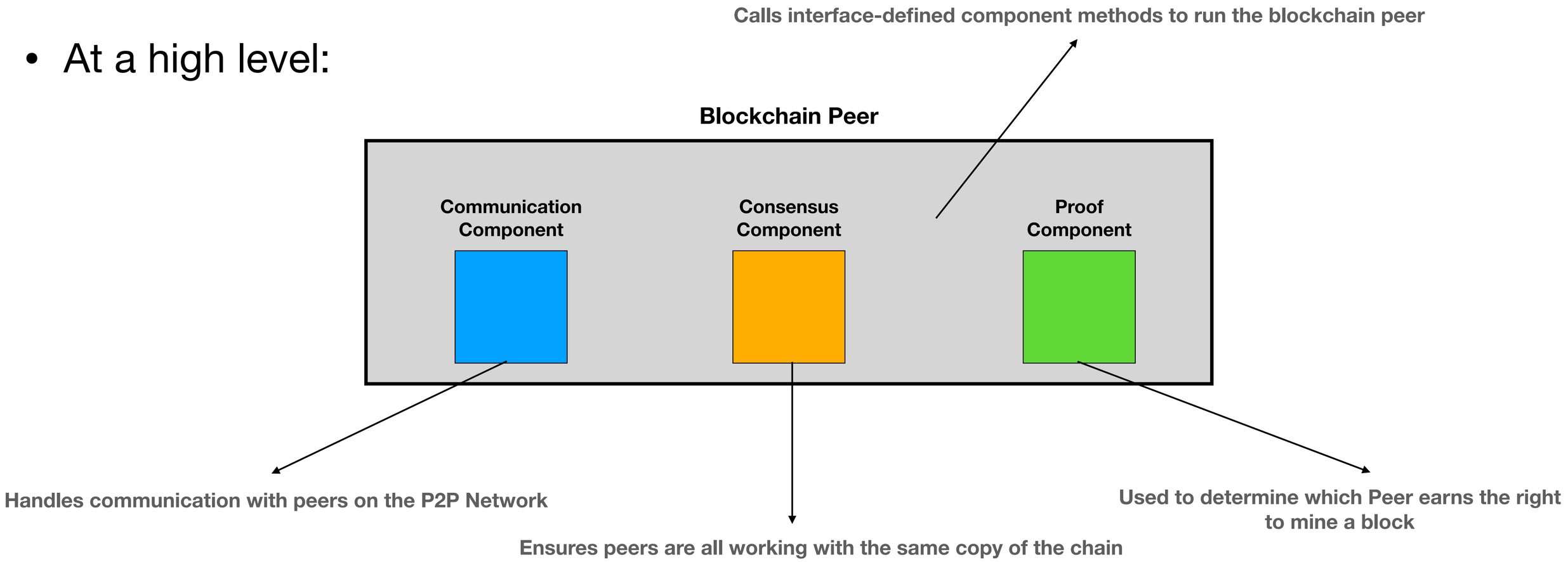
- **COMP 4580 - Computer Security:** Attack the blockchain, do a 51% attack. Explain consensus and peer-to-peer security. Make a compliant, but malicious peer.

- **COMP 2140 - Data Structures and Algorithms:** Use the concept of blockchain to help educate students on a variety of topics, including but not limited to linked lists, hashing, and blockchain technology itself.

Modular Blockchain

Design & Architecture

- At a high level:



Modular Blockchain

Implementation

- **Component implementations**
  - Communicator - Communication Component
  - LongestChain - Consensus Component
  - ProofOfWork - Proof Component

Modular Blockchain

Design & Architecture

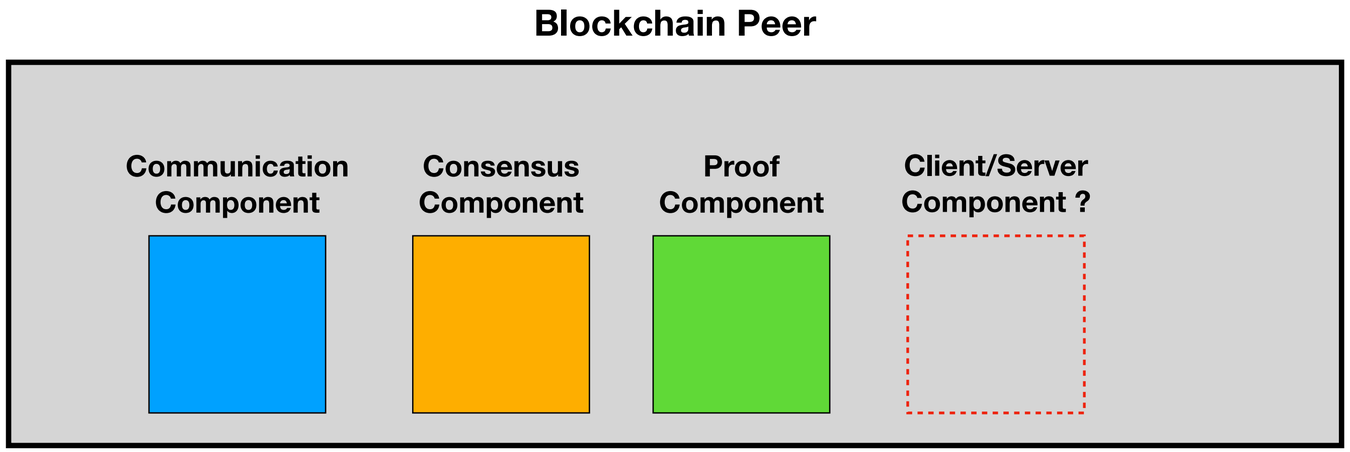
- At a lower level:

  lower level Architecture

Modular Blockchain

New transactions?

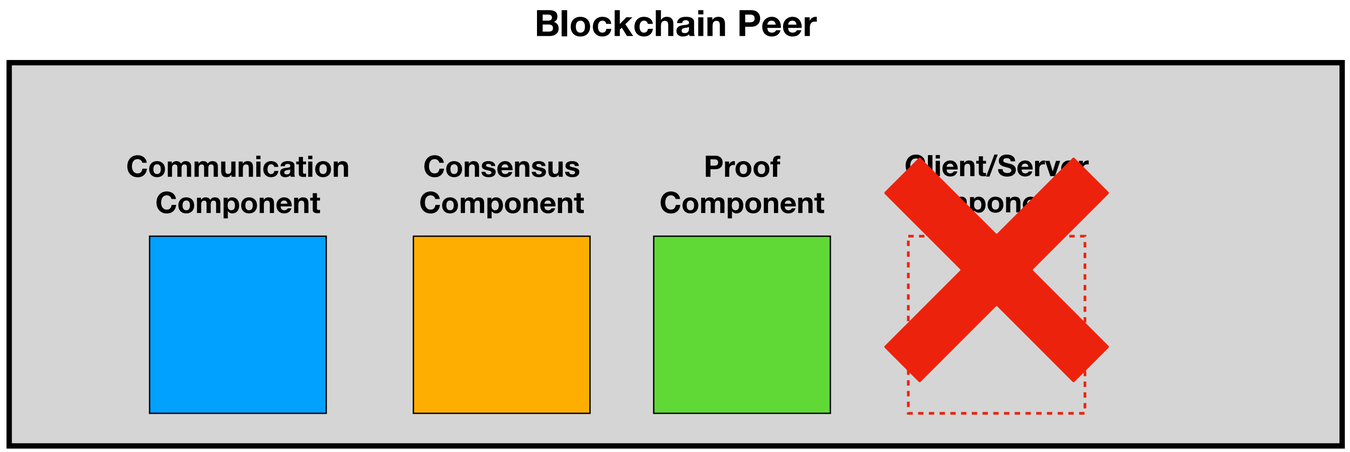
- How do we handle new transactions? Transaction Pool? HTTP Server?



Modular Blockchain

New transactions?

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



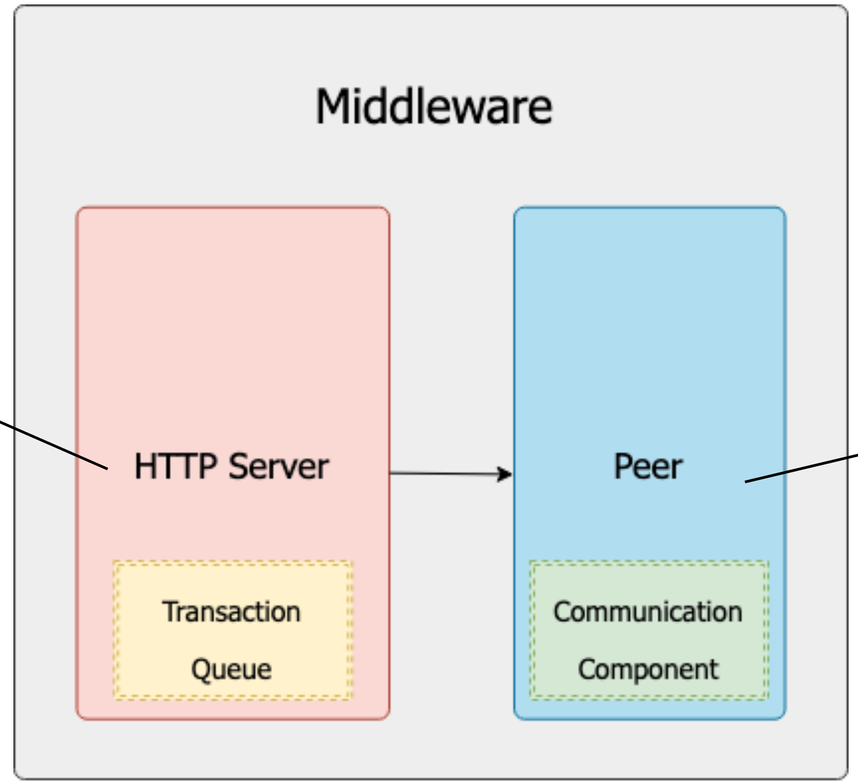
- Solution?

Middleware

Middleware

Design & Architecture

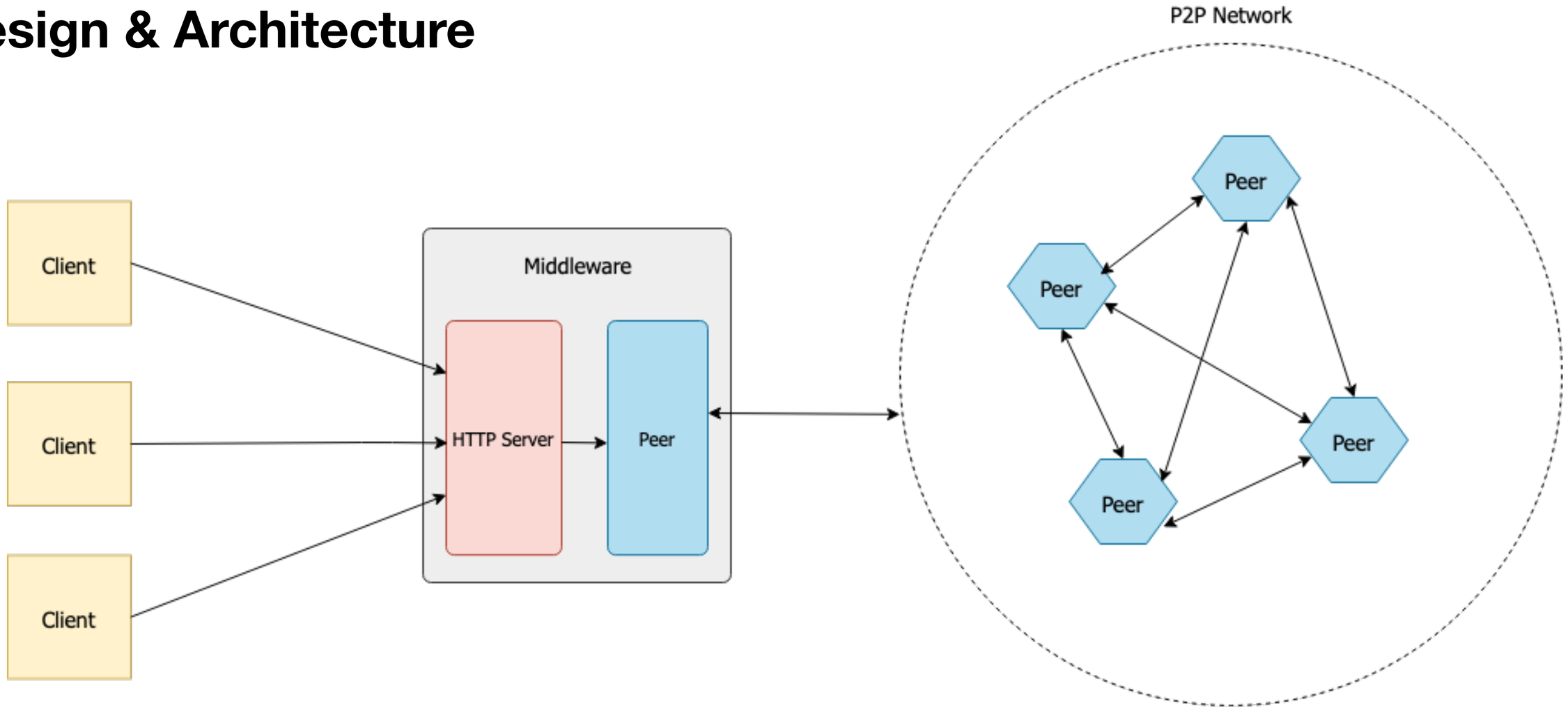
Receives HTTP POST requests from clients containing data representing a new transaction to be mined.
Converts POST data to a Transaction and adds it to a Transaction pool



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

+ Distributed system is consistent

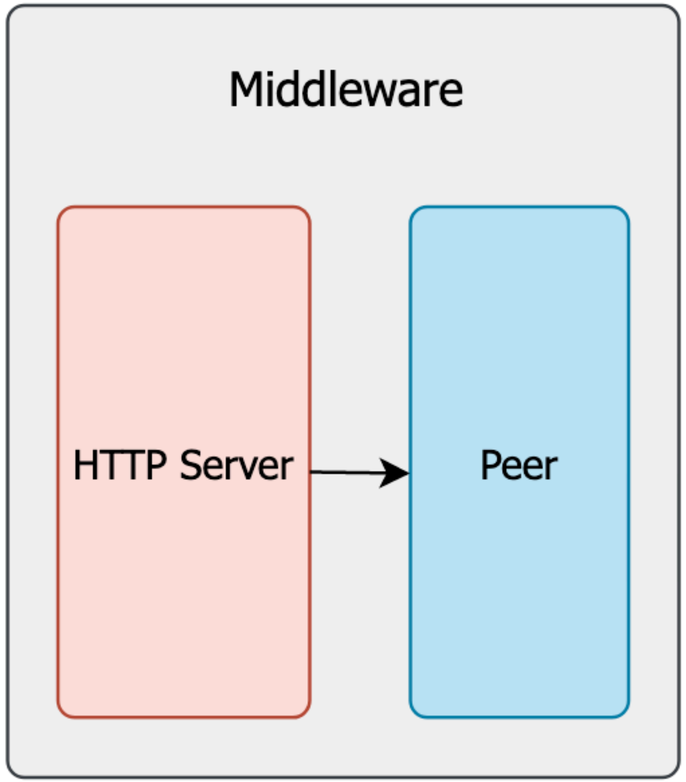
− Middleware is a single point of failure

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

How does it work?

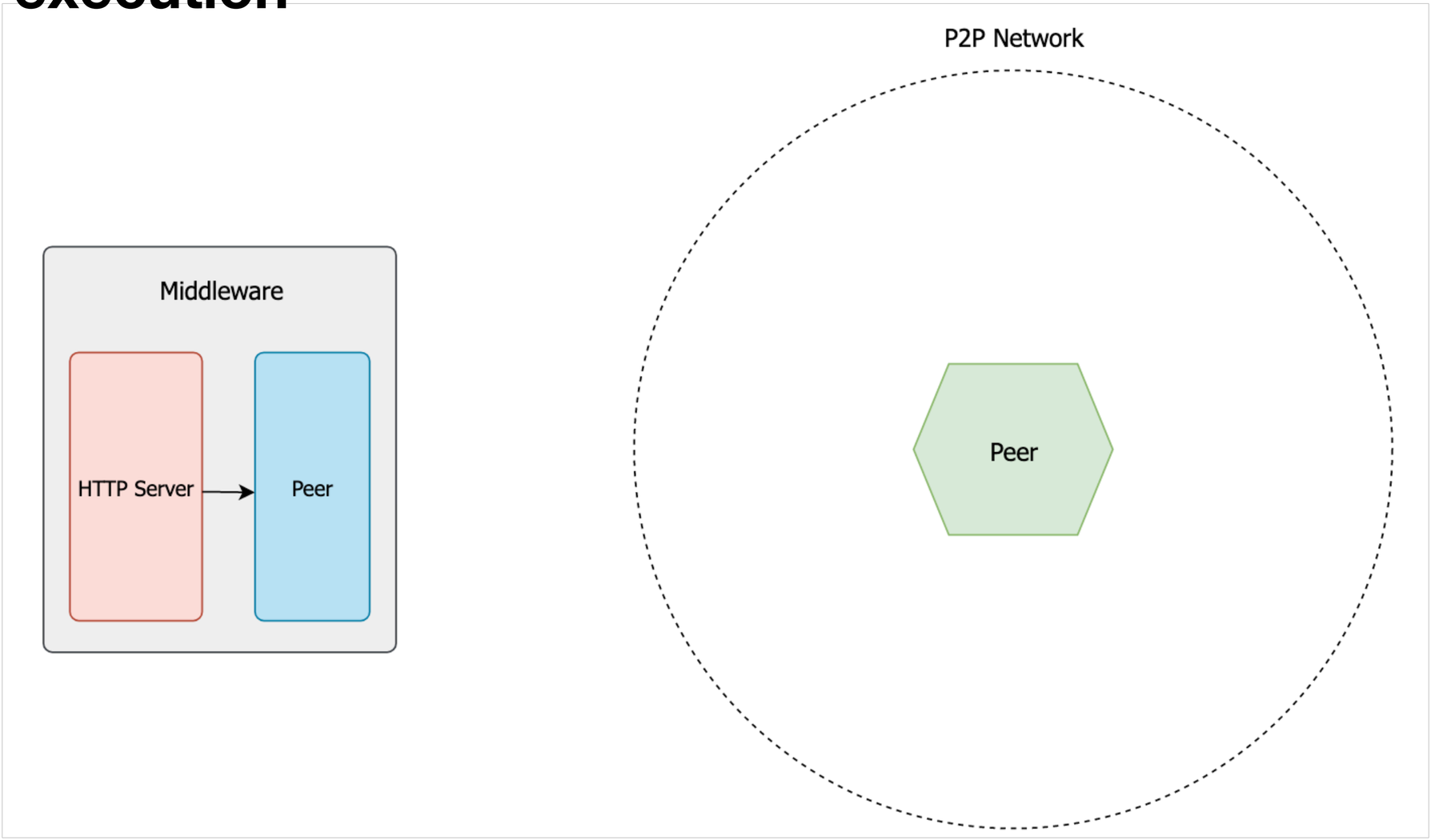
How it works

Flow of execution



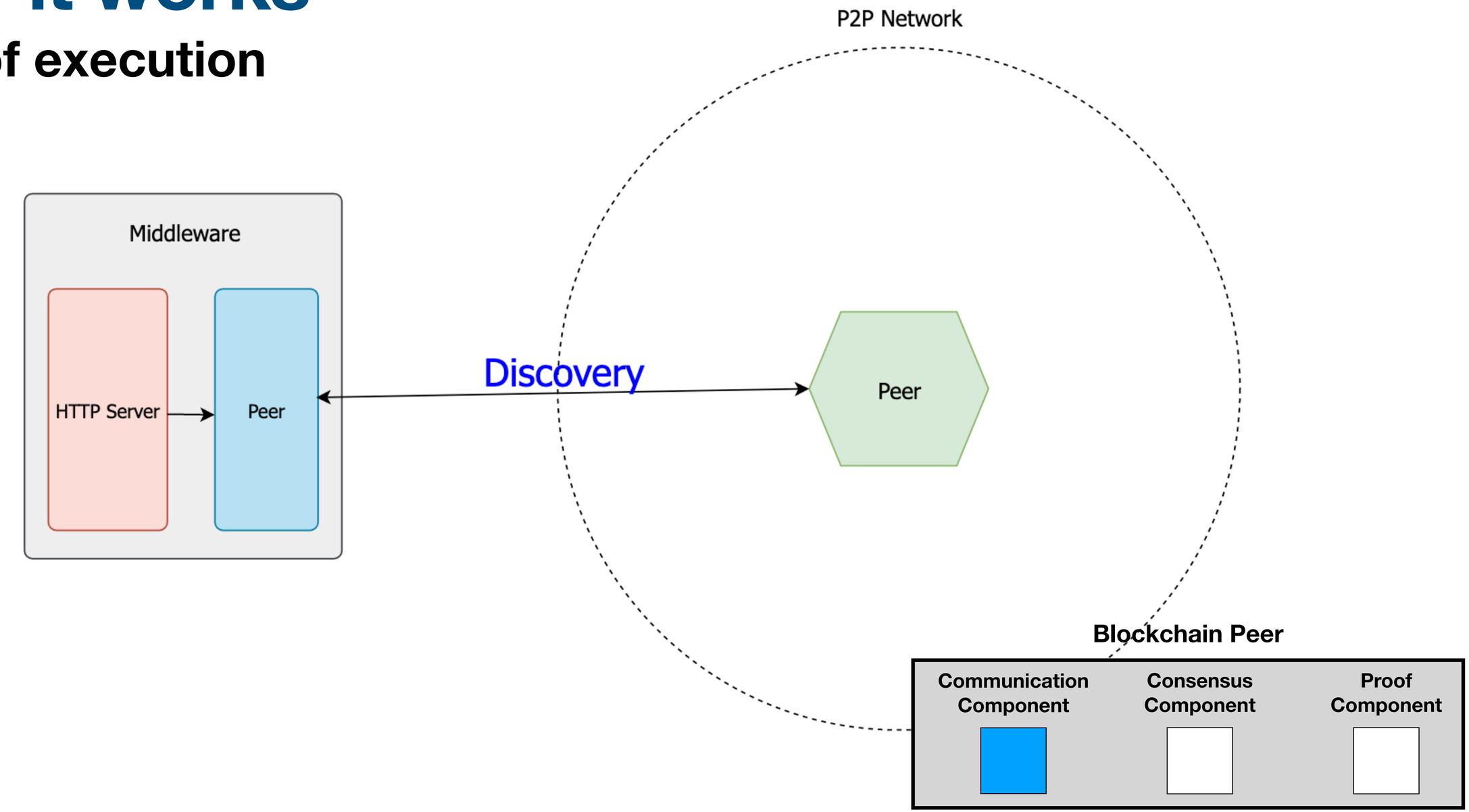
How it works

Flow of execution



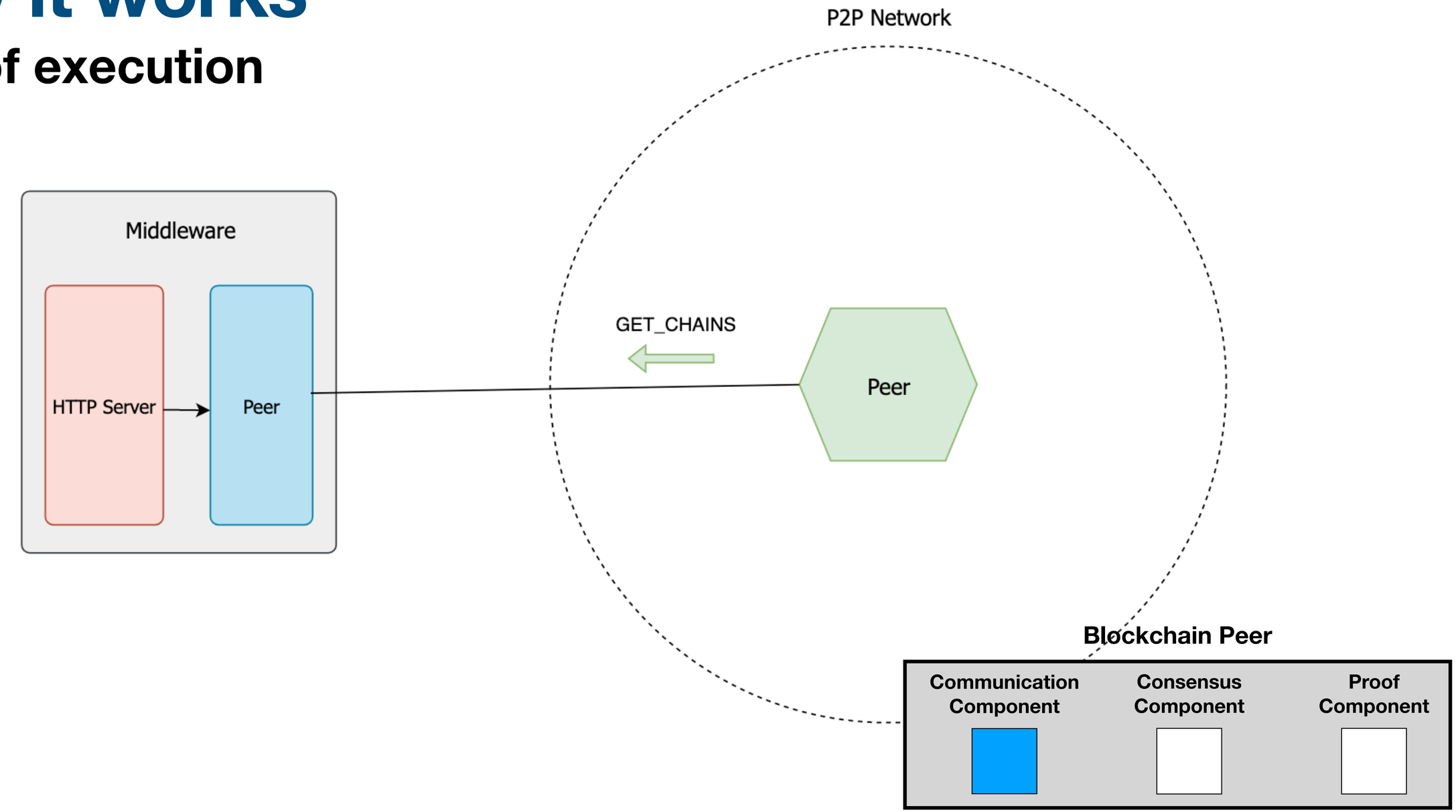
How it works

Flow of execution



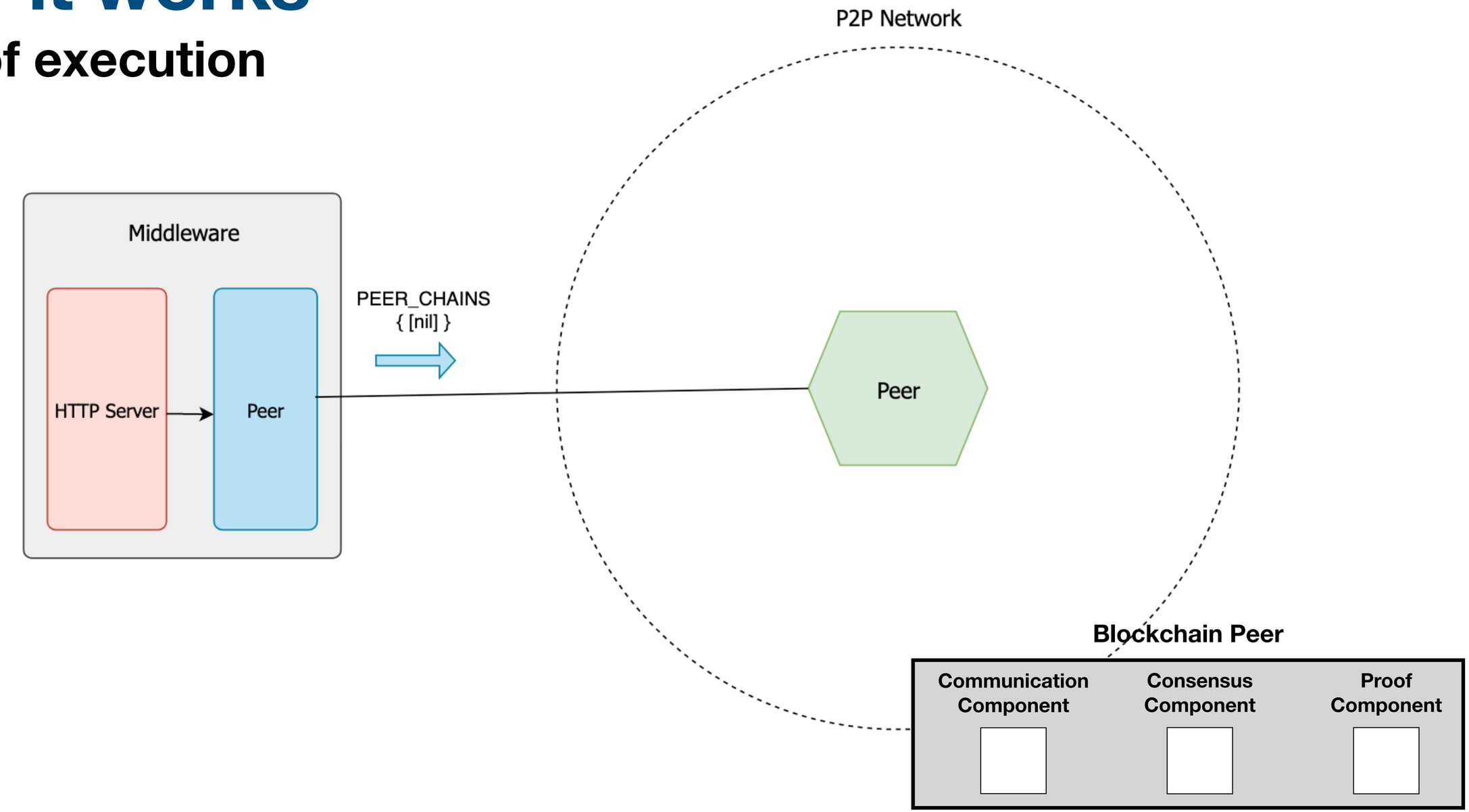
How it works

Flow of execution



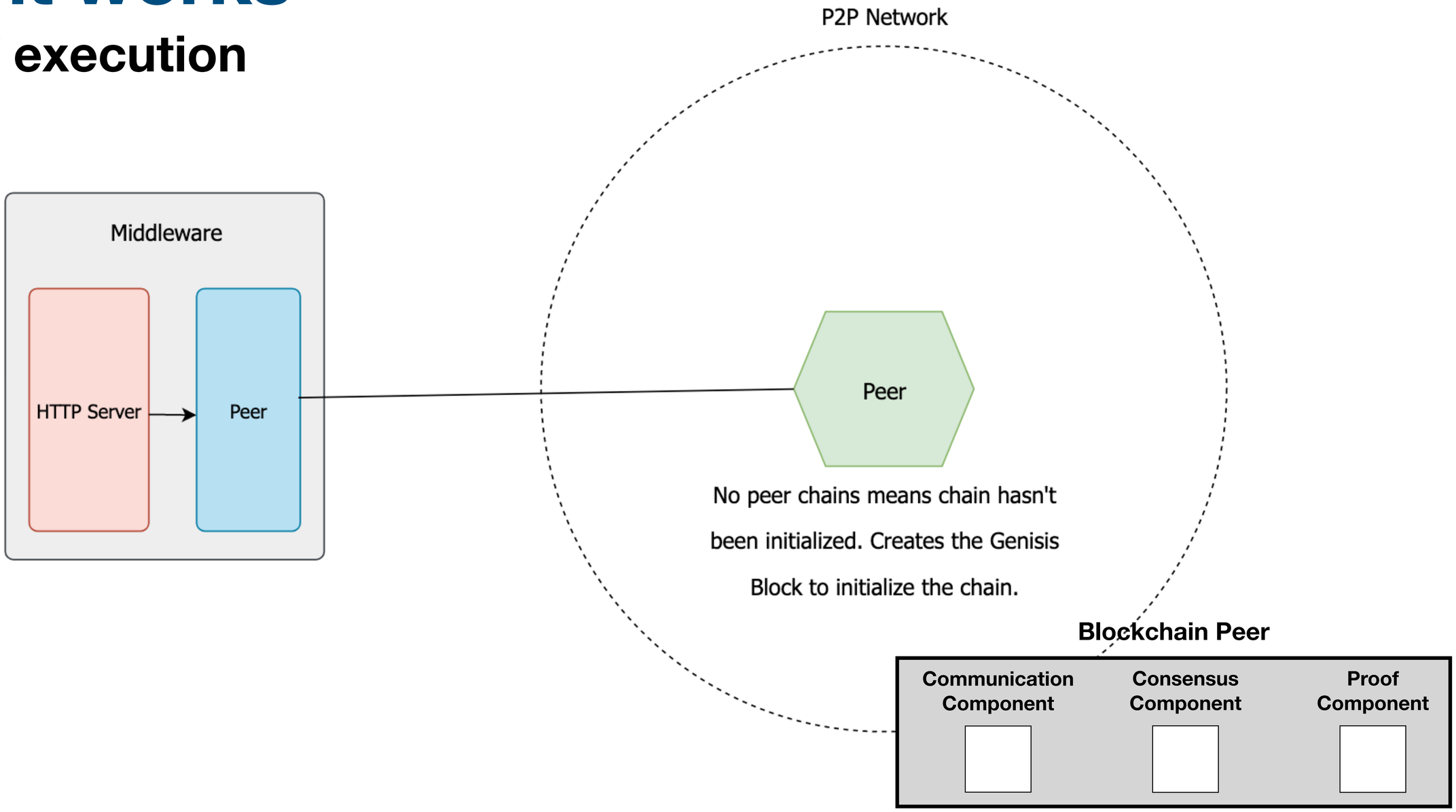
How it works

Flow of execution



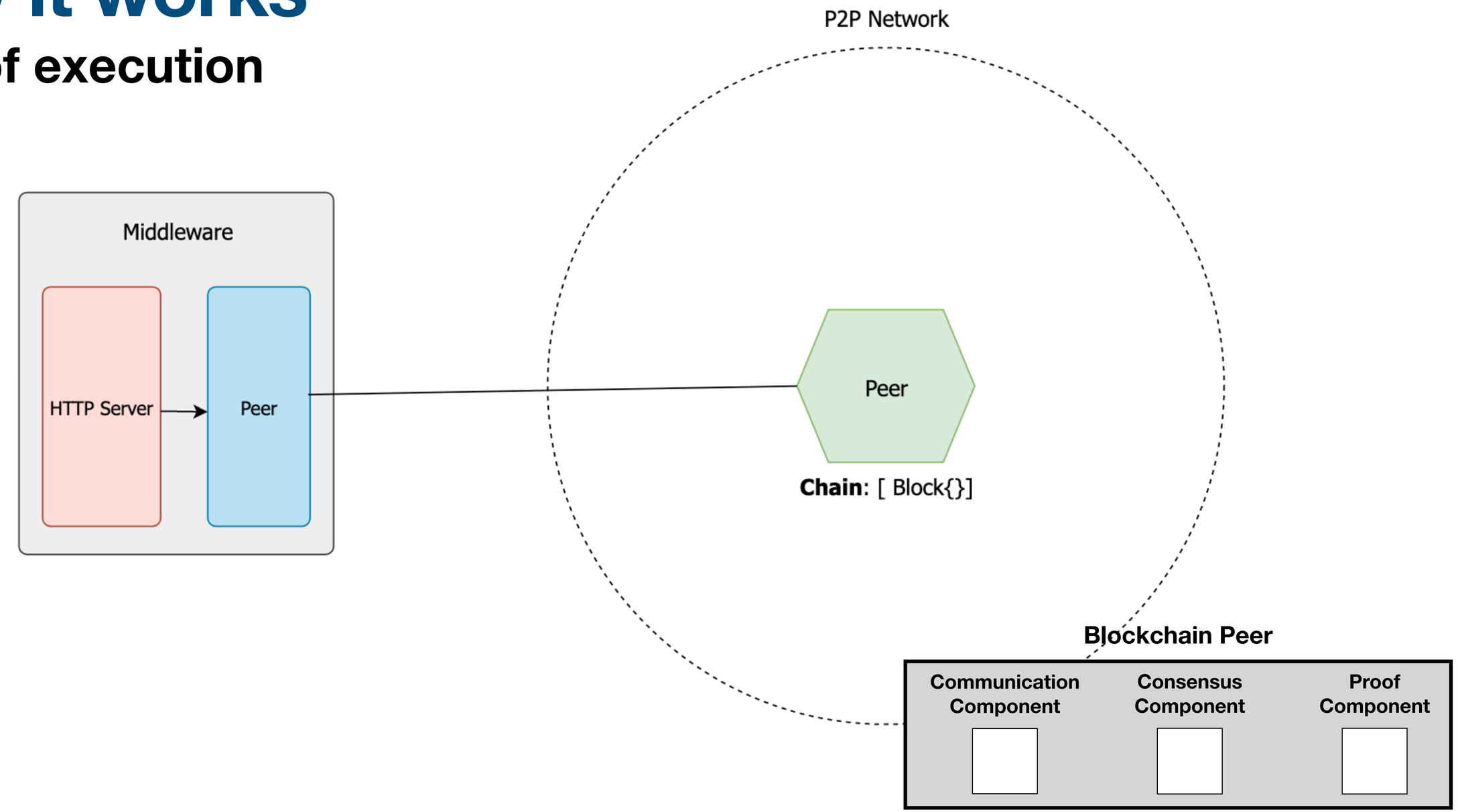
How it works

Flow of execution



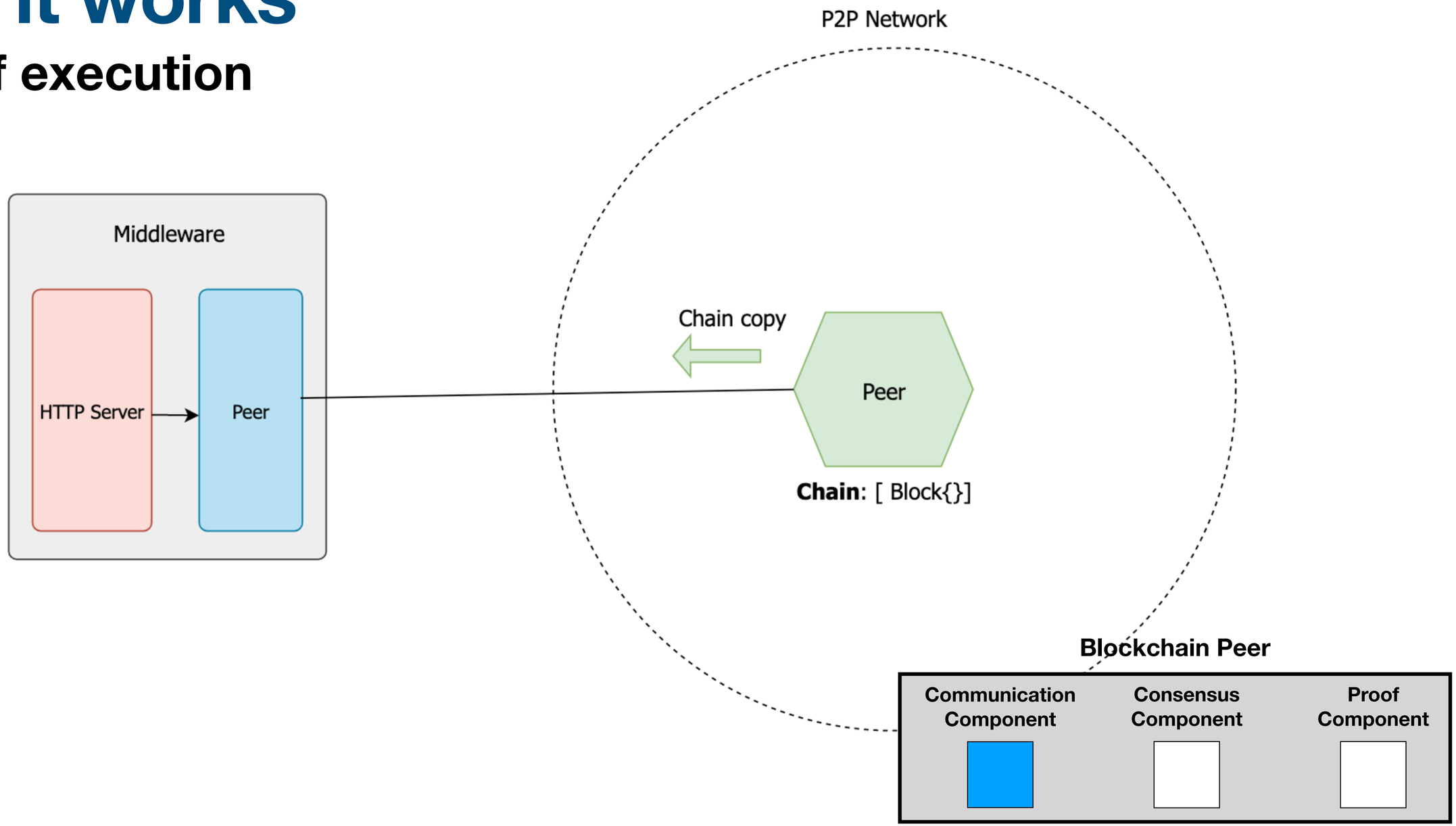
How it works

Flow of execution



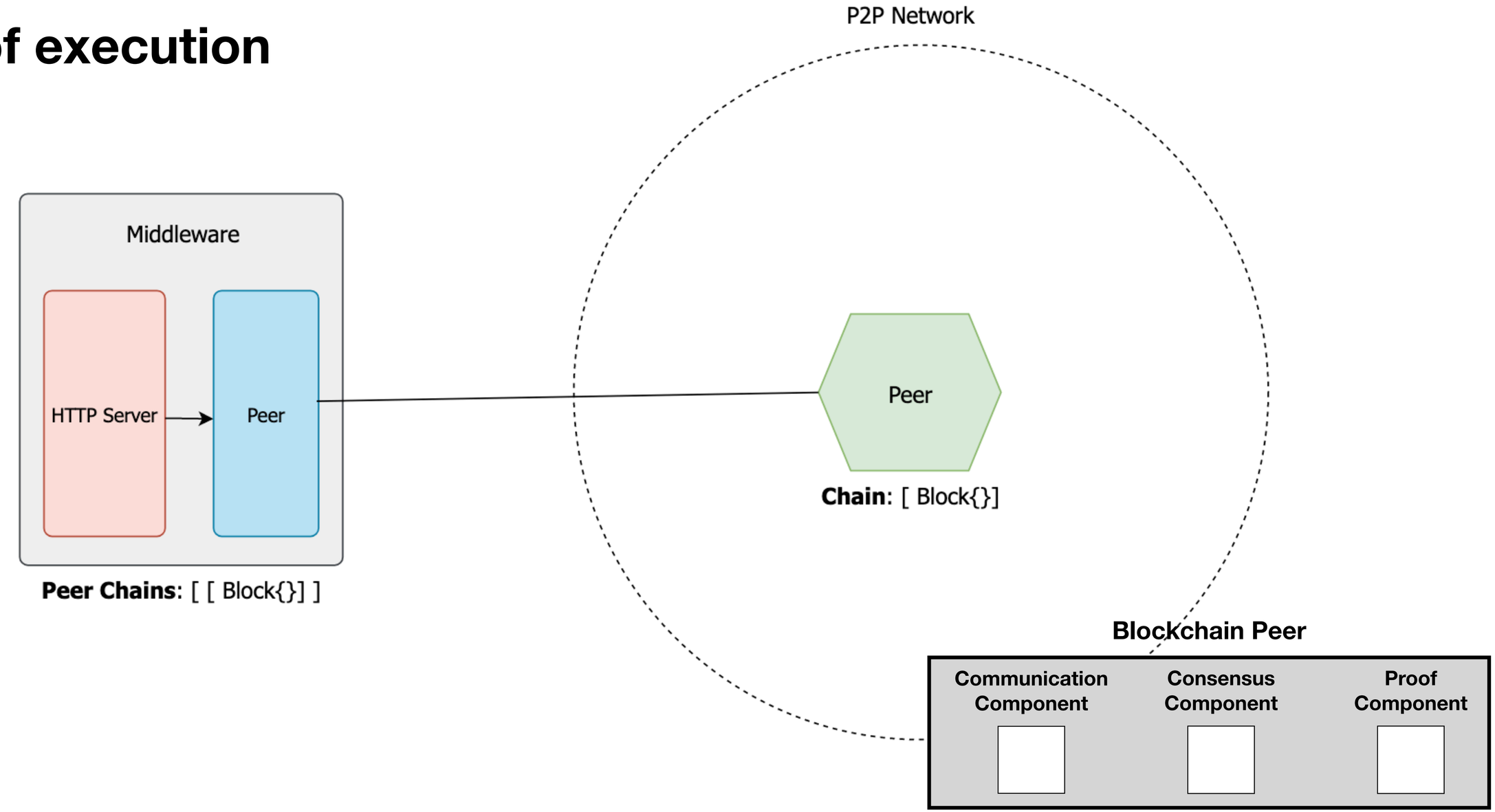
How it works

Flow of execution



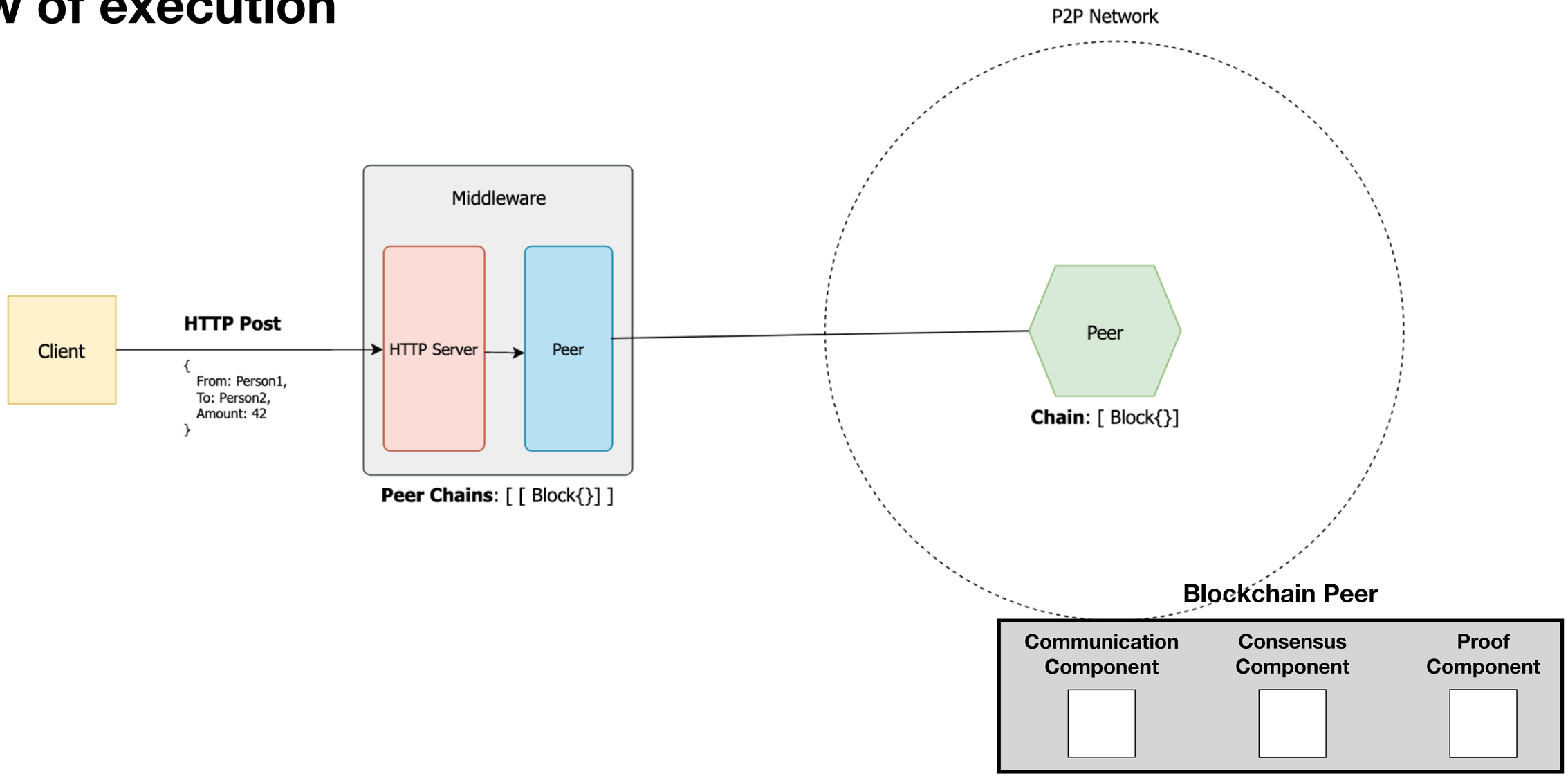
How it works

Flow of execution



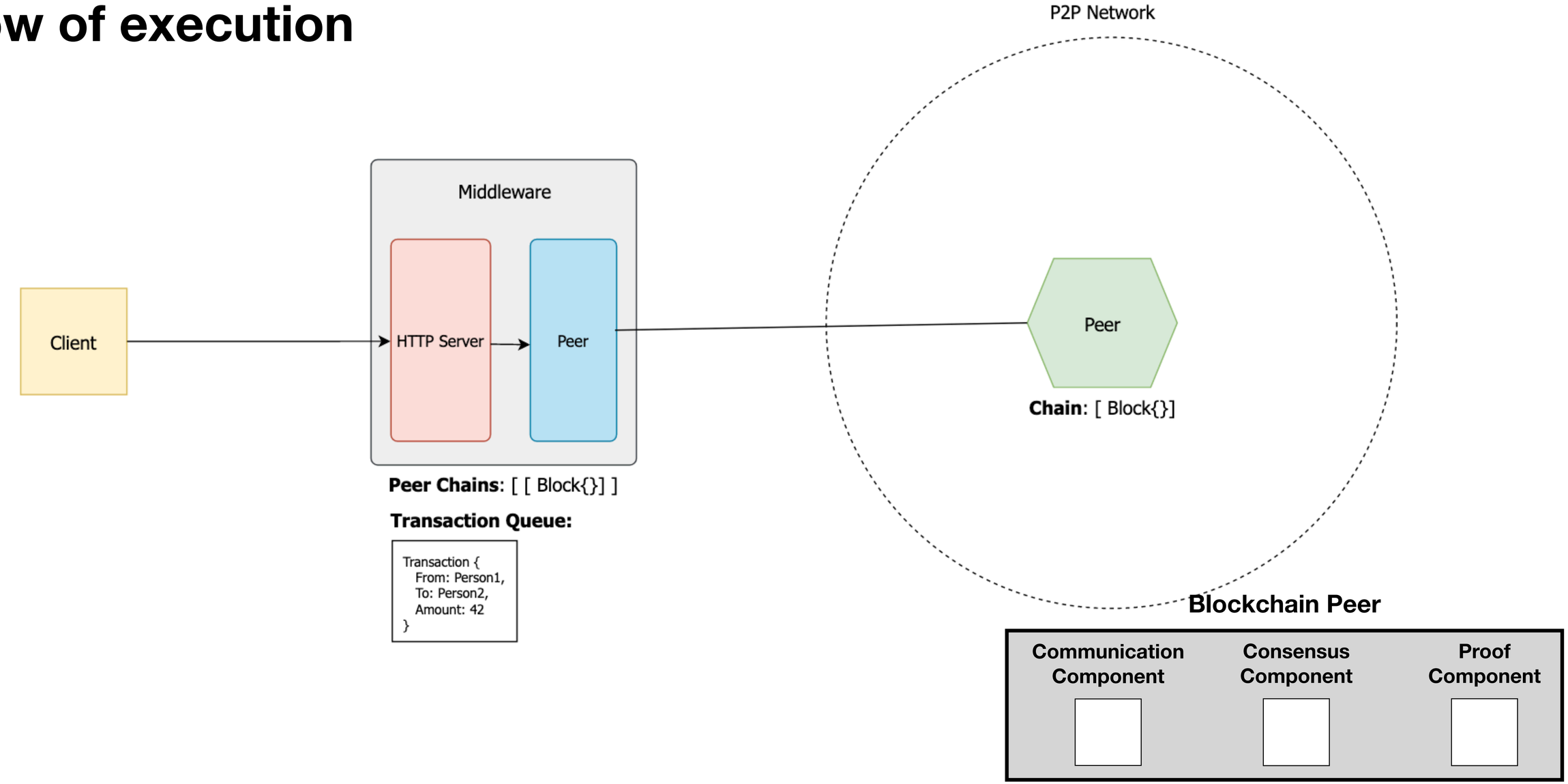
How it works

Flow of execution



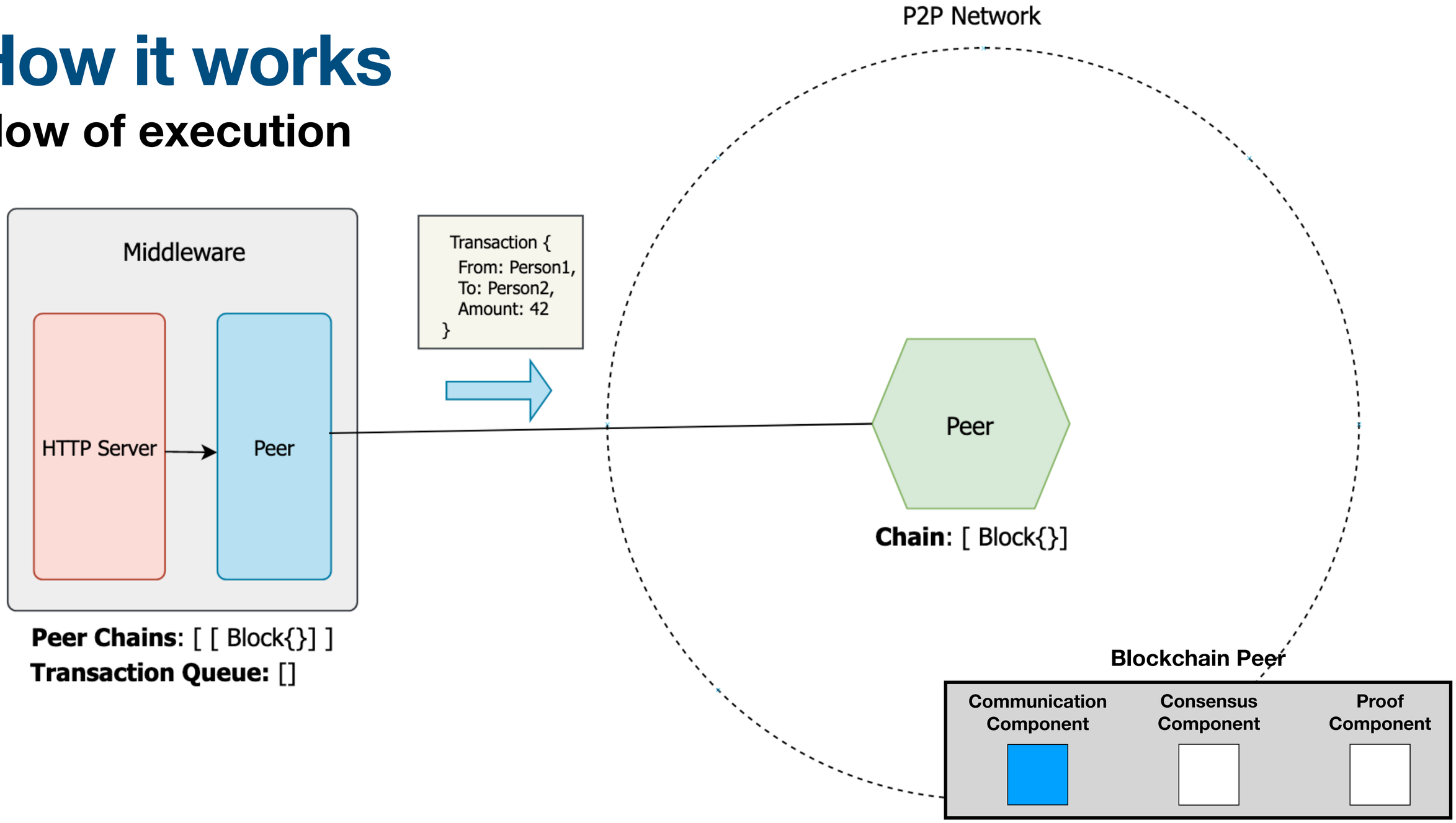
How it works

Flow of execution



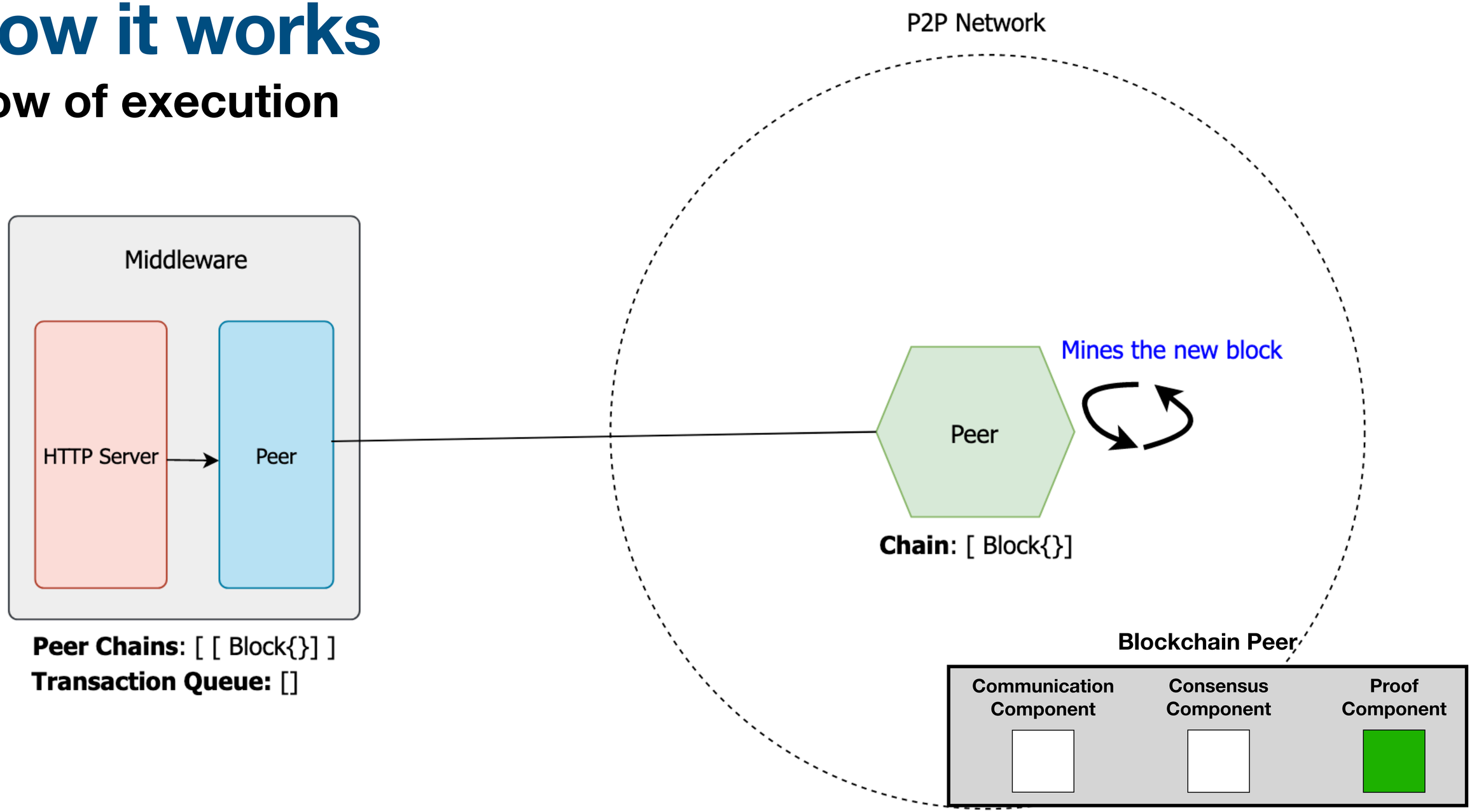
How it works

Flow of execution



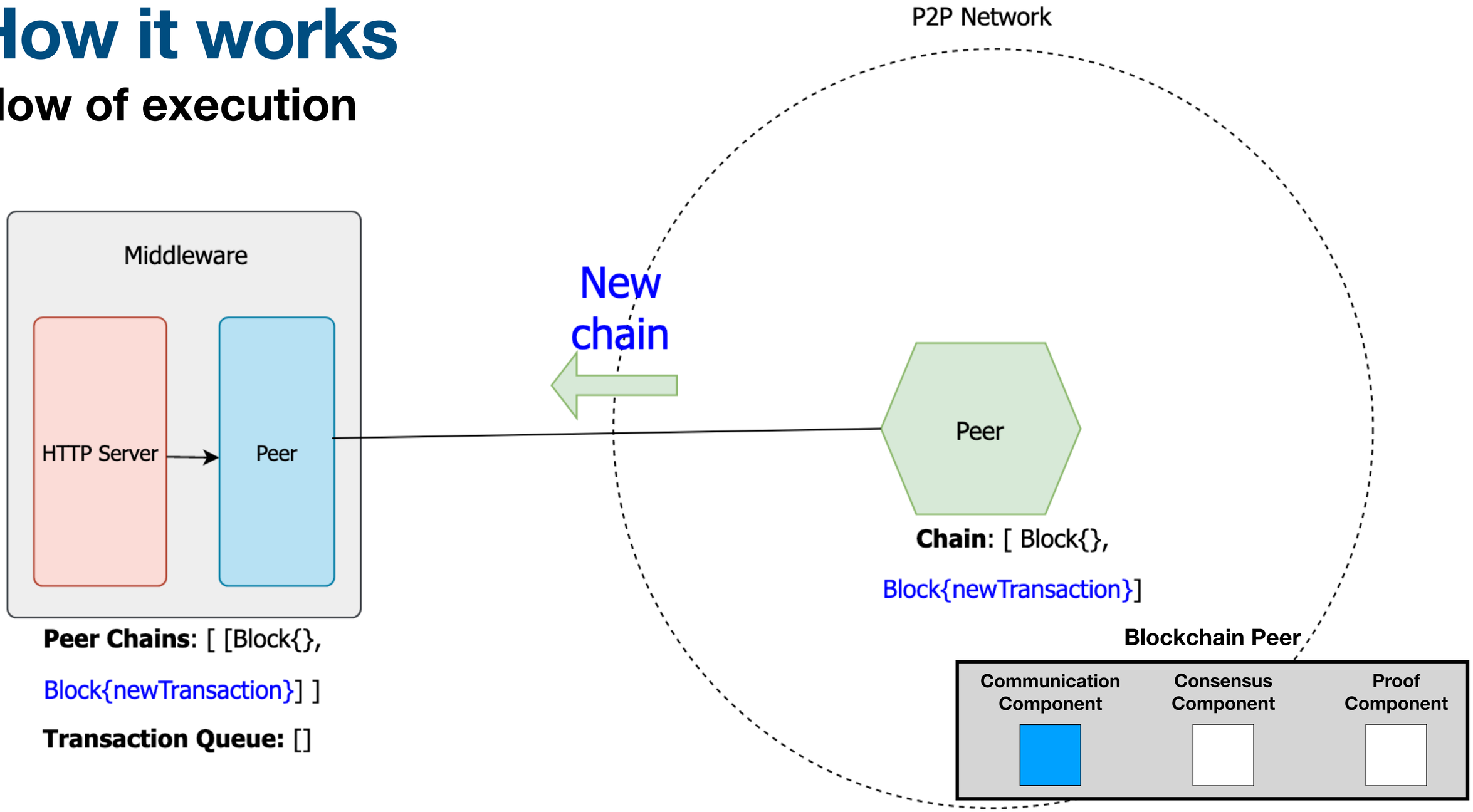
How it works

Flow of execution



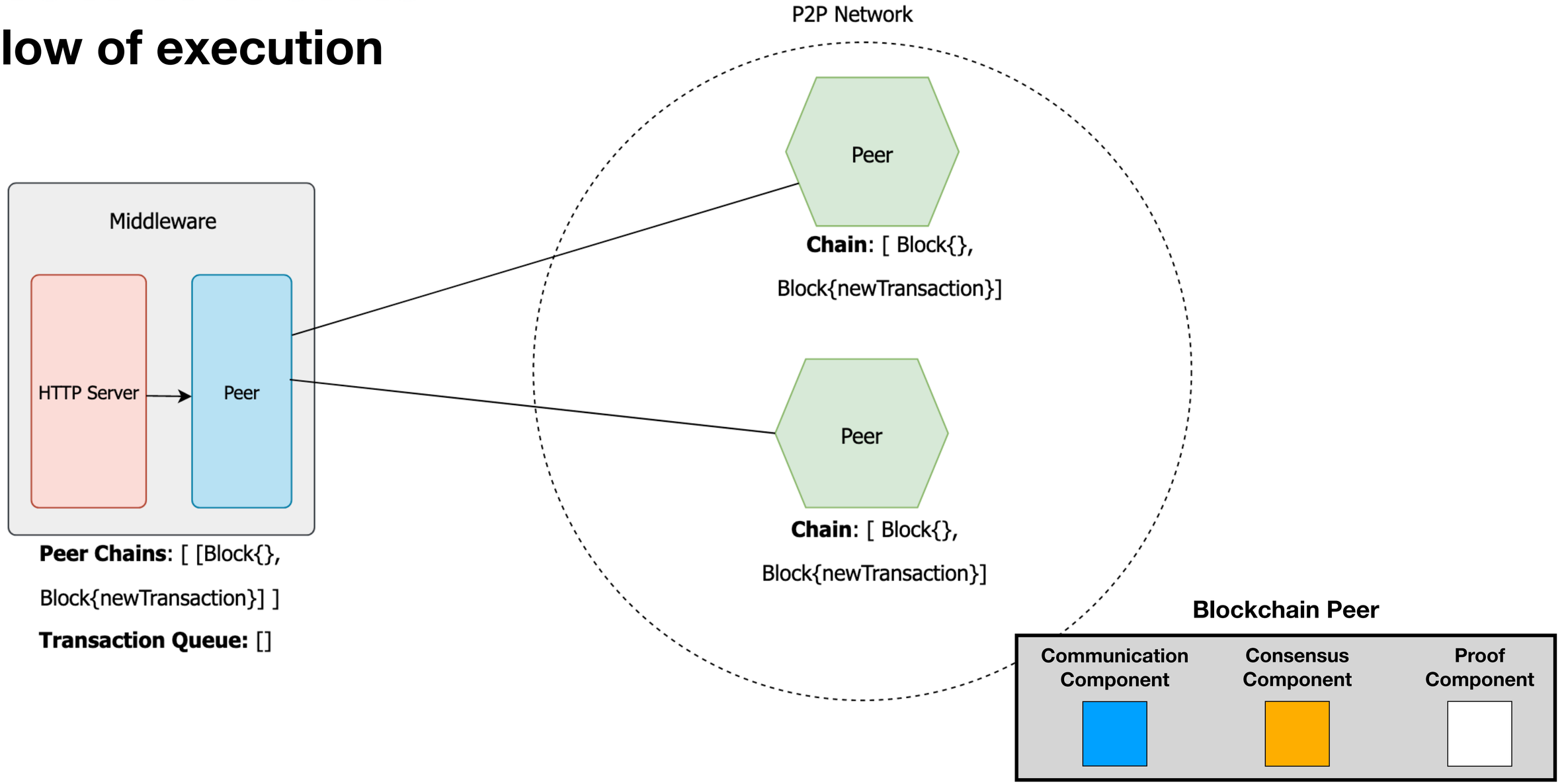
How it works

Flow of execution



How it works

Flow of execution



What’s the point?

What's the point?

Error Handling

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

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?

Web Development/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

Future Work

Future Work

What's next?

- Testing Suite, composed of Unit and Integration tests for the current implementation
- Course-specific material
- Formal pedagogical paper
- Use in an actual course assignment

Questions?

Thank you!

Repo: https://github.com/ZackHolmberg/Blockchain-Honours-Project