



# Sessional Assignment

## Software Design & Architecture

**Submitted By:** Muhammad Zain UI Abideen  
ID# 13740  
BSSE

**Submitted To:** Respected Ma'am Aasma Khan (Lecturer)

# Sessional Assignment

## Peer-to-Peer Architecture Style

**Definition:** Peer-to-peer architecture (P2P architecture) is a commonly used computer networking architecture in which each workstation, or node, has the same capabilities and responsibilities. It is often compared and contrasted to the classic client/server architecture, in which some computers are dedicated to serving others.

**Elements/components of peer to peer architecture:**

**Nodes (Peers):**

**Nodes** are key components for creating a decentralized network. A node is a client application installed on a user's computer. It performs processing and storage of transactions, execution and confirmation of the terms of the smart contracts, processes requests from third-party systems, and provides data upon requests.

Any network node has the same specified functionality and performs the functions described above by the means of the following five modules:

- CREDITS transmission/transfer protocol (CSP)
- The decision-making module
- Data Storage
- Smart contract processing module
- API

1. The general concept of **CREDITS** transmission/transfer protocol (CSP) is as follows: The TCP/IP network model is represented by the UDP (User Datagram Protocol) transmission protocol. Computer applications can send messages (in this case referred to as datagrams) using UDP to other hosts over an IP-network without obligatory preliminary messages for selection of a special data transmission protocol.

2. **Decision-making (consensus) module** conceptually, the description of the consensus algorithm for the CREDITS network is as follows:

Generation of the list of nodes participating in a particular round, according to DPoS technology

DPoS establishes a rating system of reputation. In other words, it creates a group of nodes with the right to validate transactions, generate a transaction pool, validate and send the pools to the nodes, for further chain recording.

Each new round is organized every 5 milliseconds ( $5 \times 10^{-3}$  seconds). Only one transaction pool is created within one round. The list of all used network nodes is formed. Each node is assigned a random number (rating). Based on these numbers, the list of nodes is sorted in a descending order.

3. **Data Storage** to store data this platform uses LevelDB, a high-performance, replicable NoSQL system with a key-value data storage format developed by Google. LevelDB Data Storage is written in C++ and connects to applications as a shared library, providing the possibility of storage of ordered data sets in which arbitrary binary keys are mapped to arbitrary binary values.  
Data storage algorithm. Data is stored in SS-tables (Sorted String Tables) as key-value pairs. The set of SS-tables forms a LSM-tree. LSM-trees (Log-structured merge-trees) have several levels of storage. The first (zero or MemTable) level is in RAM, and the rest is on a disk. Each level represents a tree or a list of trees, and each level has a size limit (usually 10x per level).
4. **Smart contract processing module** is a code written in Java.  
The smart contract processing module receives contracts from a desktop wallet and stores them in its data storage at a certain address, then they are executed by a contract constructor (named «initialization process»). After that, from a decision-making module we receive a signal for the execution of a certain part of the contract and values of the input data. In other words, a certain Java method in the class is executed and the global variables data are stored for each contract in the data storage. The new tokens creation function is also available. It interacts with the network via API providing a core.
5. **Node's API** is used to communicate with third-party services CREDITS platform uses the **Apache Thrift** technology. This is an interface description language used for determination and creation of services for different programming languages. It combines a program pipeline with a code generation engine to develop services, to some extent efficiently and easily working with such languages as C#, C++, Delphi, Java, Perl, PHP, Python, JavaScript, etc. In other words, Thrift is a binary communication protocol.  
The following methods are used to implement the connection of third-party services to the core of the platform.

### **Constraints that define how elements can be integrated to form the system of peer to peer Architecture:**

The final app microarchitecture is peer-to-peer architecture. The same device acts as a client and as a server in this arrangement, with significant elements of each of the four functions of the app present on it. Because each device serves simultaneously as a client and a server, the consolidated device is often referred to as a *servlet*. In its pure form, there is no separate server or centralized point of control. This means that every client also simultaneously acts as a server. Therefore all devices connected to peer-to-peer architecture can simultaneously initiate requests and fulfill requests from each other. The key advantage of this approach is immense scalability: The addition of every new client simultaneously adds server capacity to the network. Scaling the capacity of any other architecture usually requires additional capacity on the server side, the need for which is eliminated by the use of peer-to-peer microarchitectures. Skype is an example of such architecture; it allows tens of millions of users to simultaneously use the service and can readily and automatically scale to meet rising demand. The incremental cost of adding another use is therefore pennies, and adding more users improves app performance unlike all other app microarchitectures where adding more users degrades performance. However, this microarchitecture has two caveats. First, there is little or no control that the

app developer has over the users of such apps. This limits the utility of this arrangement to only a few types of applications where central coordination and control are not needed and need for scalability is extremely high. Second, this architecture rarely exists in its purely decentralized form. Some centralized control is often needed and accomplished by retaining some of the server-side functionality on a separate server. This hybrid form of peer-to-peer architecture infuses some elements of the other microarchitectures (e.g., adding a central server) into the completely decentralized pure-form peer-to-peer arrangement.

### **Connectors of Peer to Peer Architecture:**

Protocols,

- Remote procedure calls (RPC)

**Protocols:** A communications protocol that provides a structure for requests between client and server in a network. For example, the Web browser in the user's computer (the client) employs the HTTP protocol to request information from a website on a server

**Remote procedure calls (RPC):** A remote procedure call is an interprocess communication technique that is used for client-server based applications. It is also known as a subroutine call or a function call. A client has a request message that the RPC translates and sends to the server. This request may be a procedure or a function call to a remote server. When the server receives the request, it sends the required response back to the client. The client is blocked while the server is processing the call and only resumed execution after the server is finished.

### **Advantages of Peer to Peer Architecture:**

1. Each computer in the peer to peer network manages itself. So, the network is quite easy to set up and maintain.
2. In the client server network, the server handles all the requests of the clients. This provision is not required in peer to peer computing and the cost of the server is saved.
3. It is easy to scale the peer to peer network and add more nodes. This only increases the data sharing capacity of the system.
4. None of the nodes in the peer to peer network are dependent on the others for their functioning.

### **Disadvantages of Peer to Peer Architecture:**

1. It is difficult to back up the data as it is stored in different computer systems and there is no central server.
2. It is difficult to provide overall security in the peer to peer network as each system is independent and contains its own data.
3. In a peer-to-peer network, the absence of centralized server make it difficult to backup data as data is located on different workstations.
4. Security is weak as each system manages itself only.
5. There is no central point of data storage for file archiving.

### **Characteristics of Peer to Peer Architecture:**

1. Peer to peer networks are usually formed by groups of a dozen or less computers. These computers all store their data using individual security but also share data with all the other nodes.
2. The nodes in peer to peer networks both use resources and provide resources. So, if the nodes increase, then the resource sharing capacity of the peer to peer network increases. This is different than client server networks where the server gets overwhelmed if the nodes increase.
3. Since nodes in peer to peer networks act as both clients and servers, it is difficult to provide adequate security for the nodes. This can lead to denial of service attacks.
4. Most modern operating systems such as Windows and Mac OS contain software to implement peer to peer networks.
5. Peer-to-peer networking is common on small local area networks (LANs), particularly home networks. Both wired and wireless home networks can be configured as peer-to-peer environments.
6. Computers in a peer-to-peer network run the same networking protocols and software. Peer network devices are often situated physically near one another, typically in homes, small businesses, and schools. Some peer networks, however, use the internet and are geographically dispersed worldwide.
7. Home networks that use broadband routers are hybrid peer-to-peer and client-server environments. The router provides centralized internet connection sharing, but files, printer, and other resource sharing are managed directly between the local computers involved.