

# Ethereum

## *Common & Application Layers*

**Prof. James Won-Ki Hong**

**Distributed Processing & Network Management Lab.  
Dept. of Computer Science and Engineering  
POSTECH**

<http://dpnm.postech.ac.kr>  
[jwkhong@postech.ac.kr](mailto:jwkhong@postech.ac.kr)

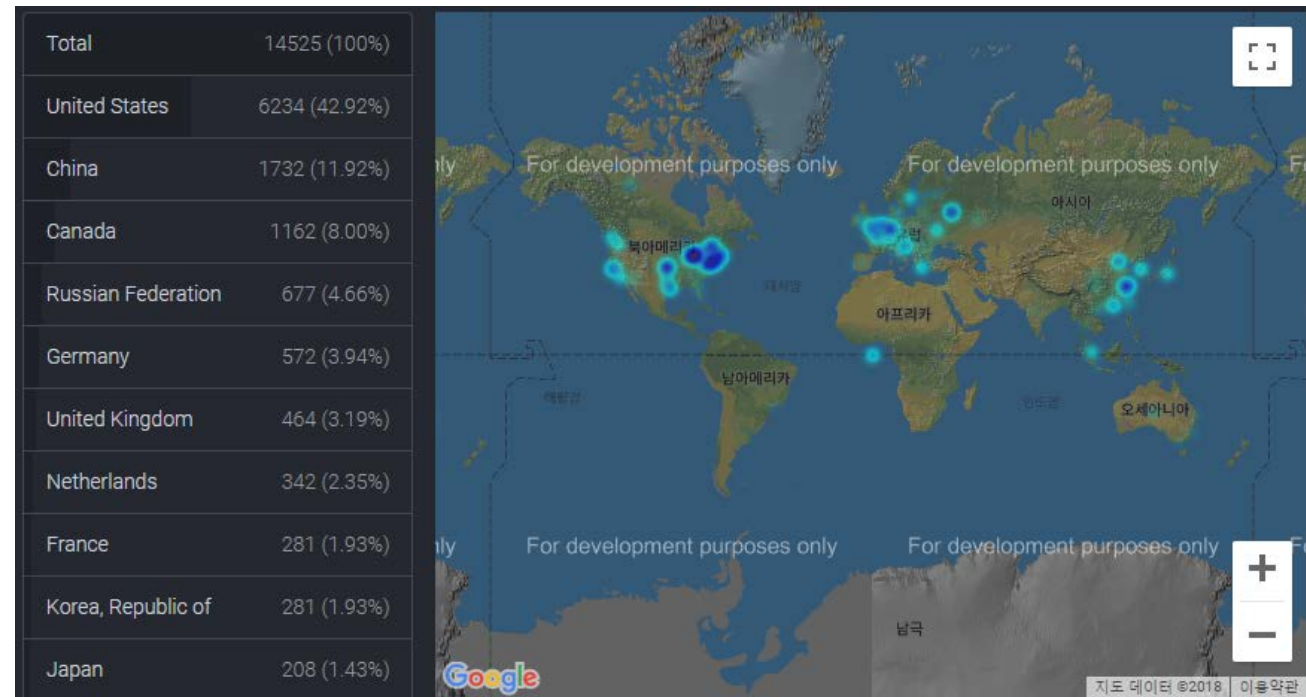
# Table of Contents

- Ethereum Overview
- Ethereum – Data Layer
- Ethereum – Consensus Layer
- Ethereum – Execution Layer
- **Ethereum – Common Layer**
- **Ethereum – Application Layer**

# Common Layer

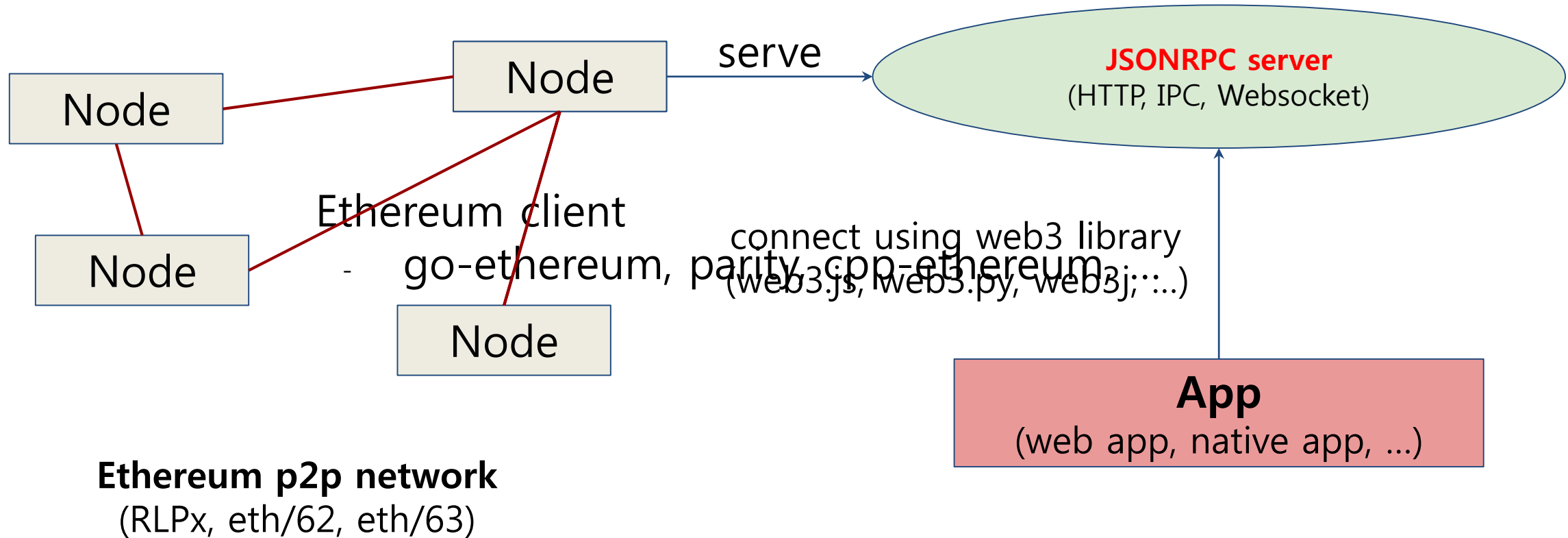
## ■ Ethereum Network, Node

- Consist of a **fully distributed Peer-to-Peer (P2P) topology**
- Classification of nodes
  - Role of node:
    - **Miner** node
    - **General (user)** node
  - Synchronization method:
    - **Full** node
    - **Light** node



source: <https://www.ethernodes.org/network/1>

## ■ Ethereum P2P Network Overview



source: Geth Network Layer & Core (by 4000D)  
: <https://www.youtube.com/watch?v=O0ahKd8Ldxw&feature=youtu.be>

## ■ Ethereum P2P Protocol: RLPx/devP2P

- Ethereum uses an encrypted network protocol called **RLPx** for normal transmission and communication between applications over a P2P network
- RLPx includes P2P network capabilities that are used throughout Ethereum
  - including the ability to **discover peer-to-peer nodes**, **UDP protocol signed with ECDSA**, and **encrypted TCP protocol**
- RLPx is commonly used in high-level application protocols such as **eth**, **shh**, **bzz** protocol
  - After Ethereum finds the node through Node Discovery Protocol, then decide which application protocol to use, such as eth, shh, bzz

# Common Layer

## ■ Node Discovery Protocol (1)

- RPC Protocol based on UDP to discover RLPx nodes connected to the network
- Ethereum constructs a P2P overlay network and performs Node Discovery through RLPx protocol
- Bootstrap node
  - Used to find peer nodes on the network
  - Does not store blockchain information
  - Maintains a list of connected nodes for a period of time

## ■ Node Discovery Protocol (2)

- Process of node discovery
  1. Normal nodes receive a list of nodes that are initially connect through the bootstrap node
  2. Start to connect with nodes, using Node Discovery Protocol
  3. After step 2, stop connection to the bootstrap node
- 4 packet types
  - Ping, pong, findnode, neighbors

Type of packet	Value of type	Description
<b>ping</b>	1	Verify that the node is online. Send a ping to the first node among the connected peer nodes, and the node that receives a ping responds by sending a pong.
<b>pong</b>	2	A response packet to ping.
<b>findnode</b>	3	It is delivered to peer nodes in the vicinity of the target node.
<b>neighbors</b>	4	A response packet to findnode, including adjacent nodes of the requested target node. If the receiver knows the nodes located near the target node, it returns a list of the nodes included in the neighbors packet.

## ■ enode: node's address

- Every node in Ethereum is represented by a URL called **enode**

e.g.,

enode://9157807b41da7be331120e8bd94afabae22d99b8c312c80ed1223fde71cbe33a304e8b0e3a9ed8f0a3551e4ec38ad6225ab5ecb7393e4e6765a53bb75de3ce9e@10.5.57.7:30303?discport=30301

- Elements

- enode URL → 512bit public key signed by private key, using ECDSA
- @ → delimiter
- 10.5.57.7 → IP address of the host
- 30303 → TCP port number
- discport=30301 → UDP port number



## ■ Several methods to connect to the bootstrap node

1. Try to connect by referring to the hard-coded bootstrap node list in the program
2. Use the `--bootnodes` option to specify the bootstrap node directly
  - e.g., `geth --bootnodes "enode URL"`
3. Use the `admin.addPeer()` in geth console to directly specify the node you want to connect to
  - e.g., `admin.addPeer("enode://9157807...ce9e@10.5.57.7:30303")`
4. Use the static node function to connect to a specific node
  - Create `static-nodes.json`
    - ➔ `[ "enode://publicKey@IPaddress:UDPport" ]`

## ■ Ethereum Database

- Ethereum uses **LevelDB** as a basic storage through **ethdb** package
  - LevelDB wrapped in a package called ethdb
  - LevelDB: key / value storage
- LevelDB stores state, transaction, receipt information of merkle tree and non-volatile storage of EVM
- Ethereum tries to change LevelDB to MongoDB
  - Etheruem/ecp (Ethereum chain parser) project

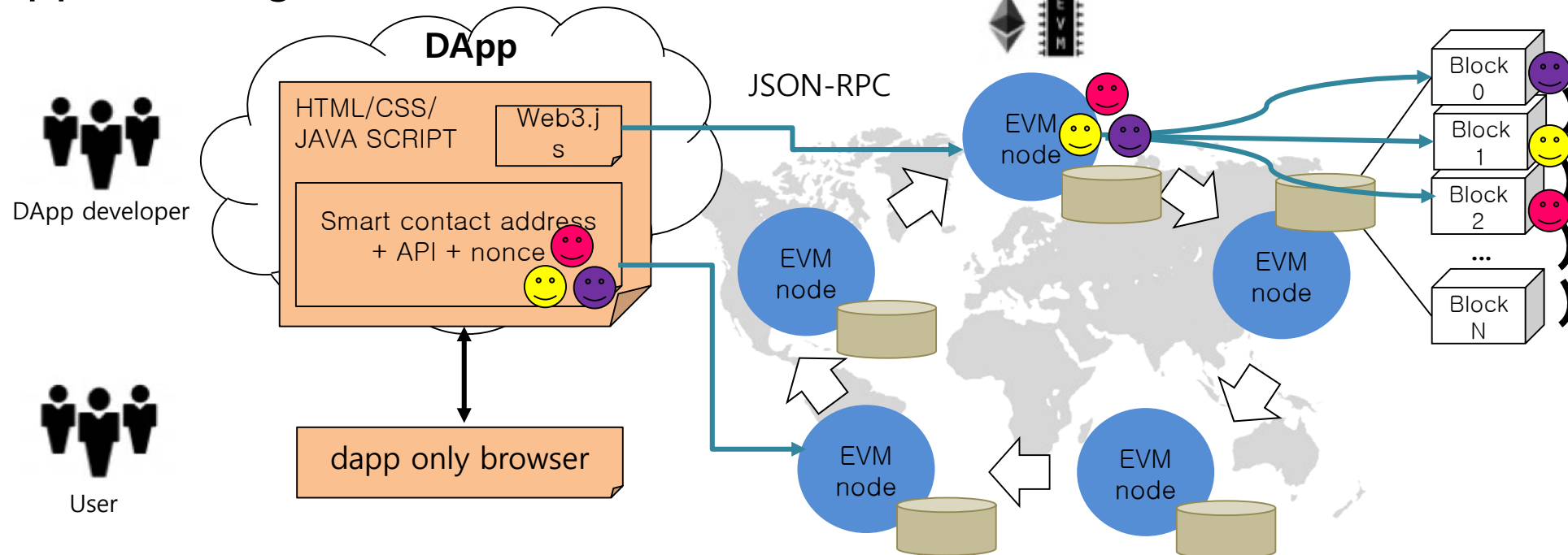
## ■ RLP (Recursive Length Prefix) Encoding

- The purpose of RLP is to encode arbitrarily nested arrays of binary data
- RLP is the main encoding method used to serialize objects in Ethereum
- The encoding process is very simple
  - Reduce the encoding size
  - Ensure consistency of byte units
- Definition
  - The RLP encoding function takes in an item
  - An item is defined as follows:
    - A string (byte array)
    - A list of items

# Application Layer

## ■ DApp (Decentralized App)

- Provides services based on smart contracts
  - Without a centralized server
- Features
  - No trust to a specific entity, Transparency, Improved security, ...
- DApp Running Environment



## ■ **Swarm: P2P File System**

- An incentive-based P2P file system and a content delivery channel such as Content Delivery Network (CDN)
- Store contents such as DApp's code and data in a distributed way and deliver the contents
- Related P2P File System: Torrent
  - It does not have the incentive to operate continuously and are not stable to operate
  - It can not guarantee performance and quality
- SWAP (Swarm Accounting Protocol) Incentive System
  - When the content is downloaded, the user pays the peer nodes that provided the contents
  - Providing popular content is paid for by peers using the content

## ■ **Whisper**: P2P Messaging System

- Communication protocol for DApps to communicate with each other
  - Designed for **easy** and **efficient** broadcasting
- Untraceable P2P messaging protocol
  - Support various node-to-node messages such as Multi-casting, broadcasting and M-to-M
- Use Encryption Algorithm
  - Asymmetric: SECP-256k1 public-key cryptography
  - Symmetric: AES-GCM
- How to run Whisper
  - >> `geth --shh`

## ■ Common Layer

- Network
- Database
- RLP encoding

## ■ Application Layer

- DApp
- Swarm
- Whisper

- <https://github.com/ethereum/wiki/wiki/%C3%90%CE%9Evp2p-Wire-Protocol>
- <https://github.com/ethereum/devp2p/blob/master/rpx.md>
- <https://github.com/ethereum/wiki/wiki/Ethereum-Wire-Protocol>
- <https://medium.com/onther-tech/geth-network-layer-e12e65760829>
- <https://github.com/ethereum/wiki/wiki/Whisper>
- <https://github.com/ethereum/wiki/wiki/RLP>
- Jaehyun Park, **core ethereum programming**, Jpub, 2018
- <https://www.youtube.com/watch?v=TDGq4aeevgY>
- <https://www.youtube.com/watch?v=WSN5BaCzsbo%26%2365533%3B%26%2365533%3B>
- [https://www.youtube.com/watch?v=-\\_Qs0XdPpw8%26%2365533%3B%26%2365533%3B](https://www.youtube.com/watch?v=-_Qs0XdPpw8%26%2365533%3B%26%2365533%3B)
- [https://www.youtube.com/watch?v=M3EFi\\_POhps](https://www.youtube.com/watch?v=M3EFi_POhps)
- <https://www.youtube.com/watch?v=ZE2HxTmxfrl>