Ethereum Blockchain: Decentralized Applications (Dapps)

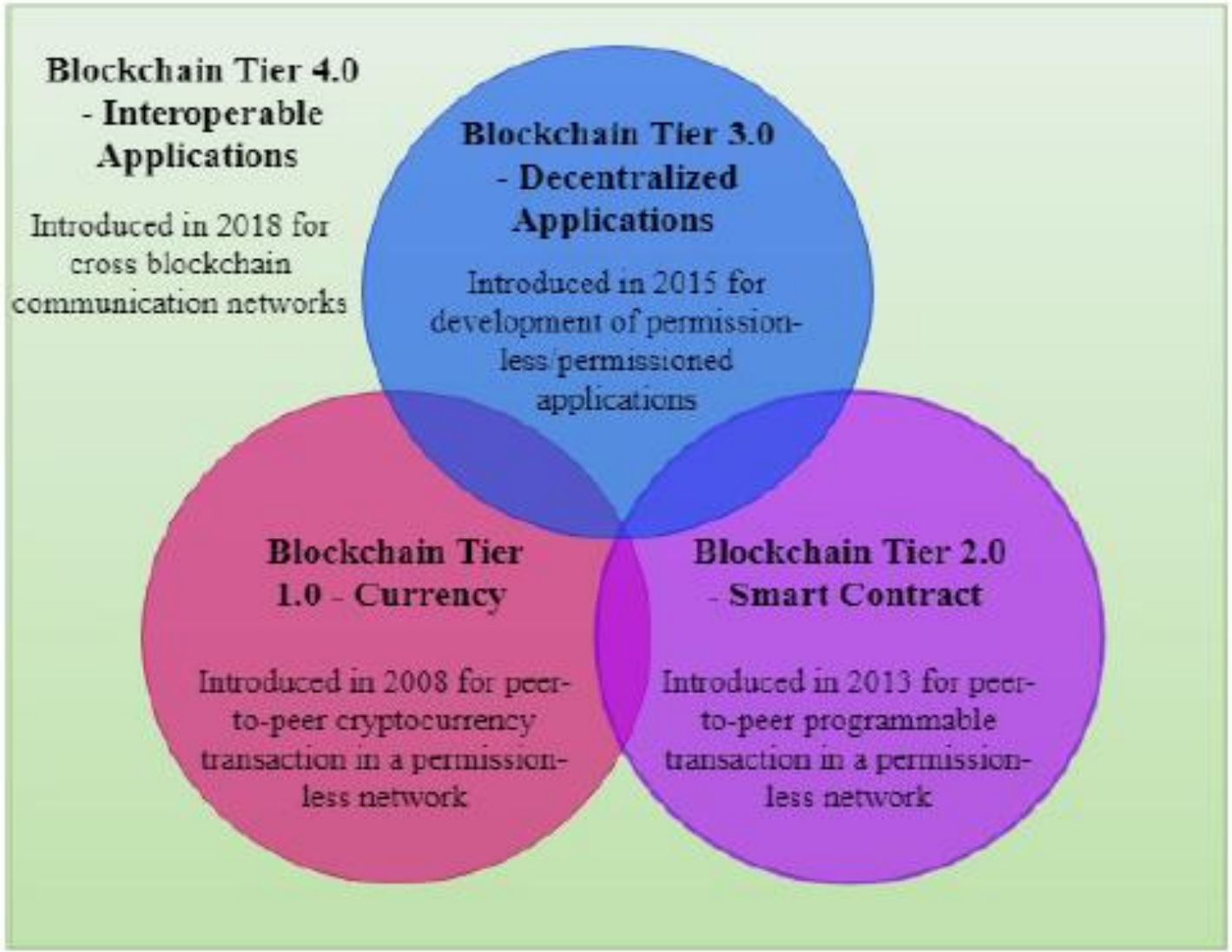


Figure 5. Blockchain Tiers.

[Source - Ismail, Leila, and Huned Materwala. "A review of blockchain architecture and consensus protocols: Use cases, challenges, and solutions." *Symmetry* 11.10 (2019): 1198]

Decentralized Applications (Dapp)

Provides blockchain features and services to the outside world for review, interactions and enjoyment.

Gives access to the blockchain for people and applications and systems, not necessarily known to each other to transact peer-to-peer.

It is an end-to-end application development process.

Essential Concepts

Blockchain Server

blockchain server
represents the infrastructure and the functionality the blockchain provides.

Dapp Architecture

Application Programming Interface (API)

Decentralized Applications (Dapp)

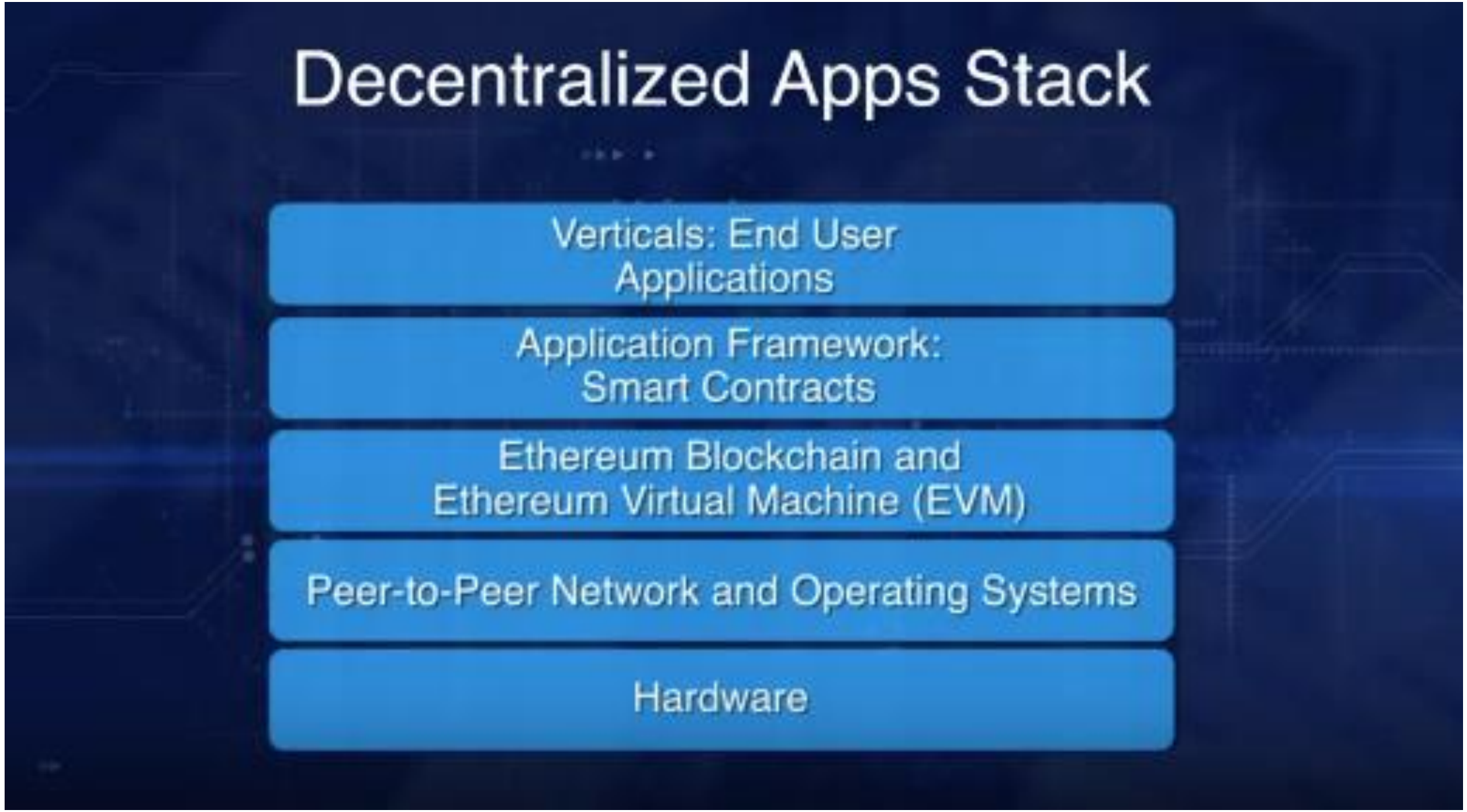
Depends on the functionality of a blockchain for its infrastructure and operations.

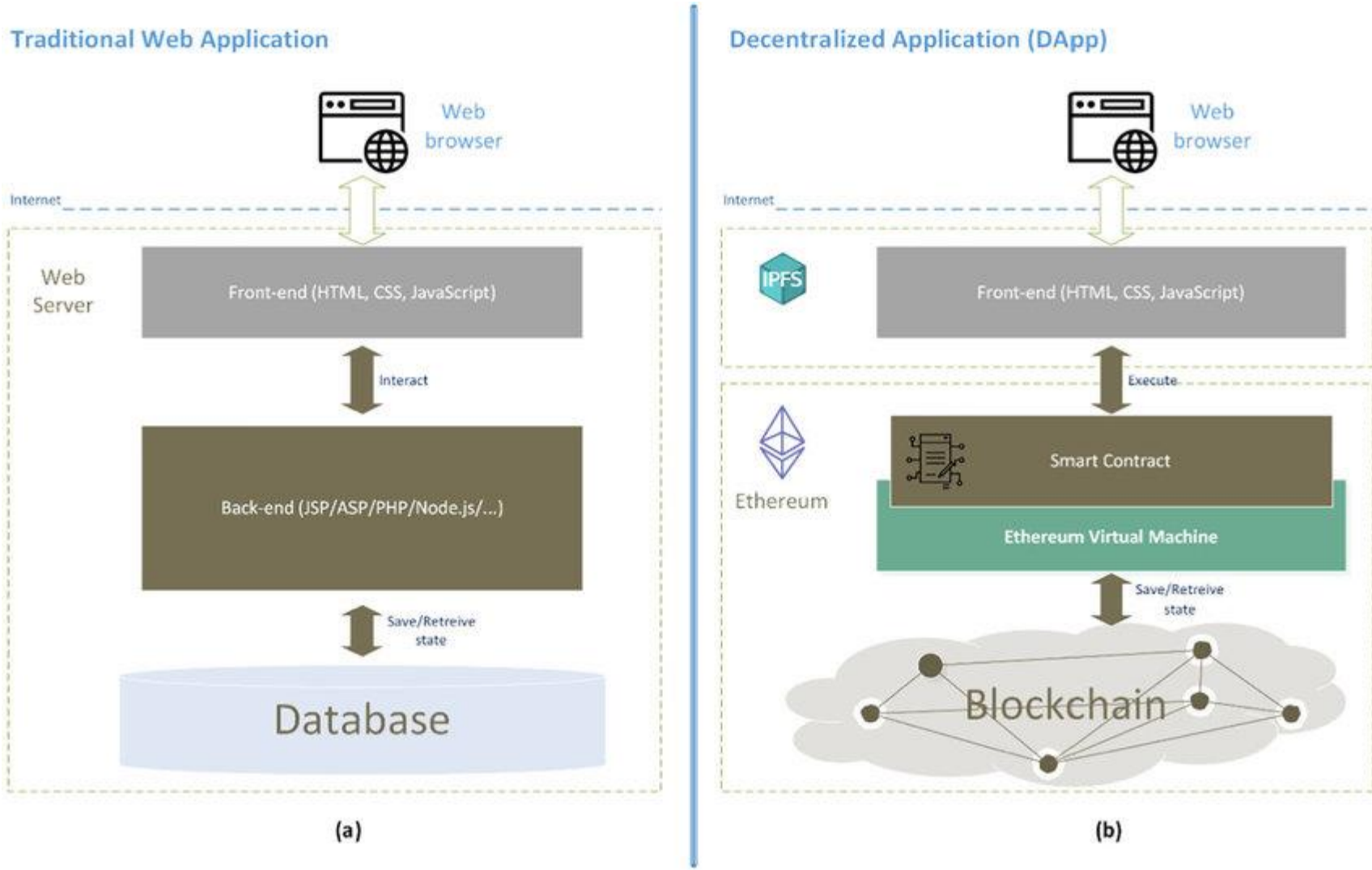
Its simplest form has a client interface as a front-end and a back-end that includes the blockchain and smart contracts.

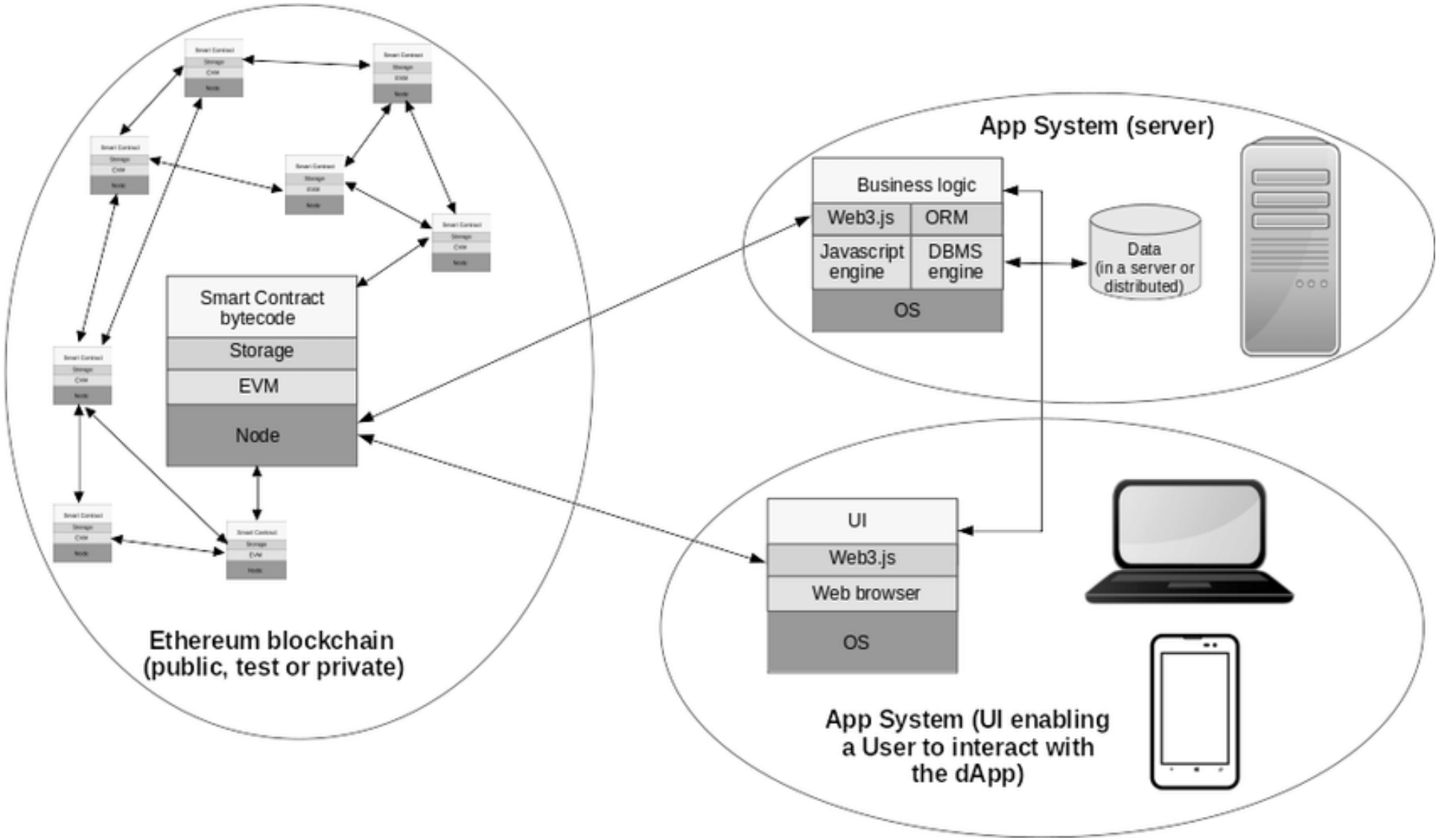
The client or the front-end can be

- A web app, HTML and Javascript framework.
- A command line interface, CLI, a desktop application,
- A mobile application, or
- An IoT, Internet of Things.

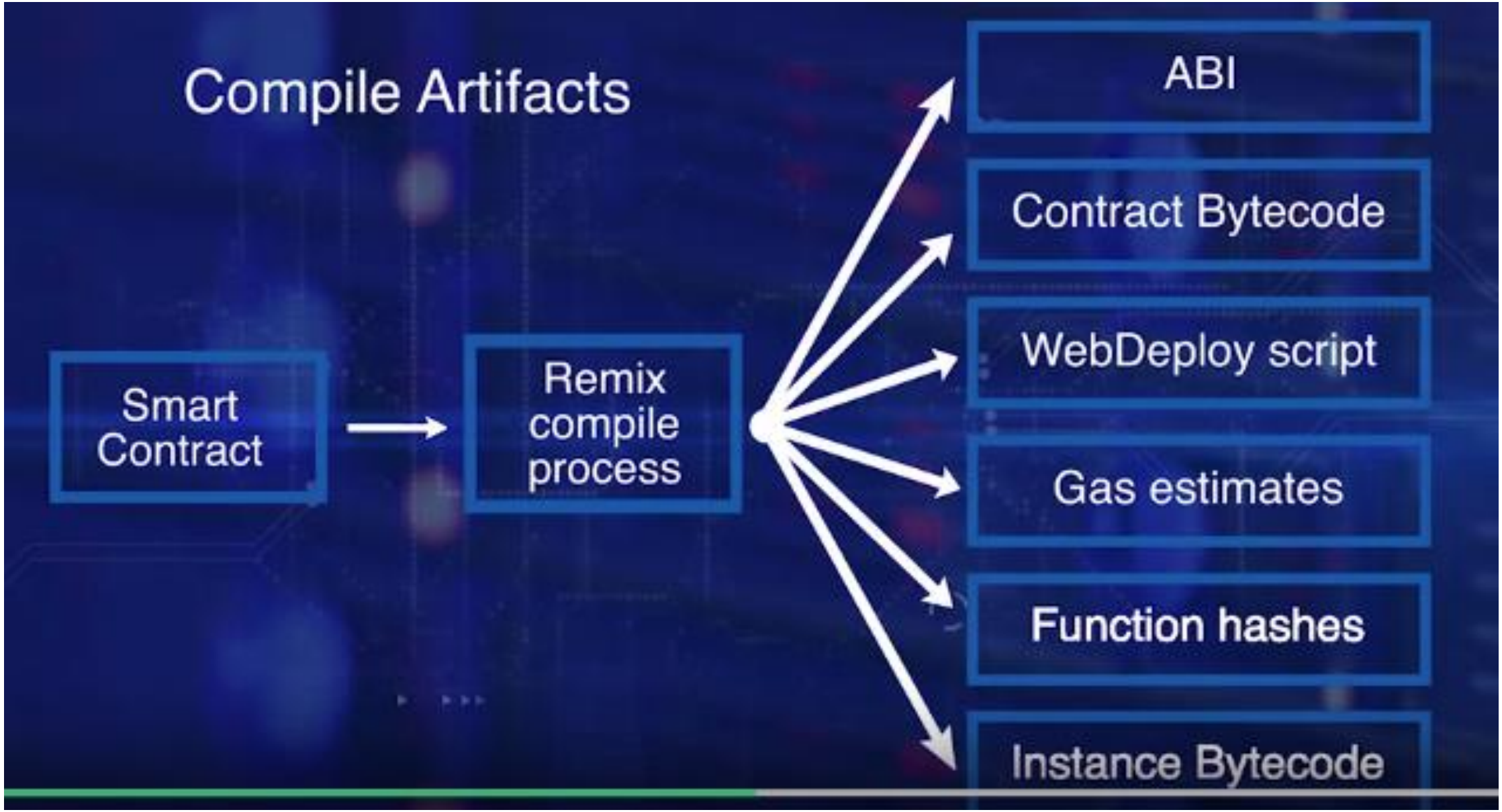
Decentralized Apps Stack



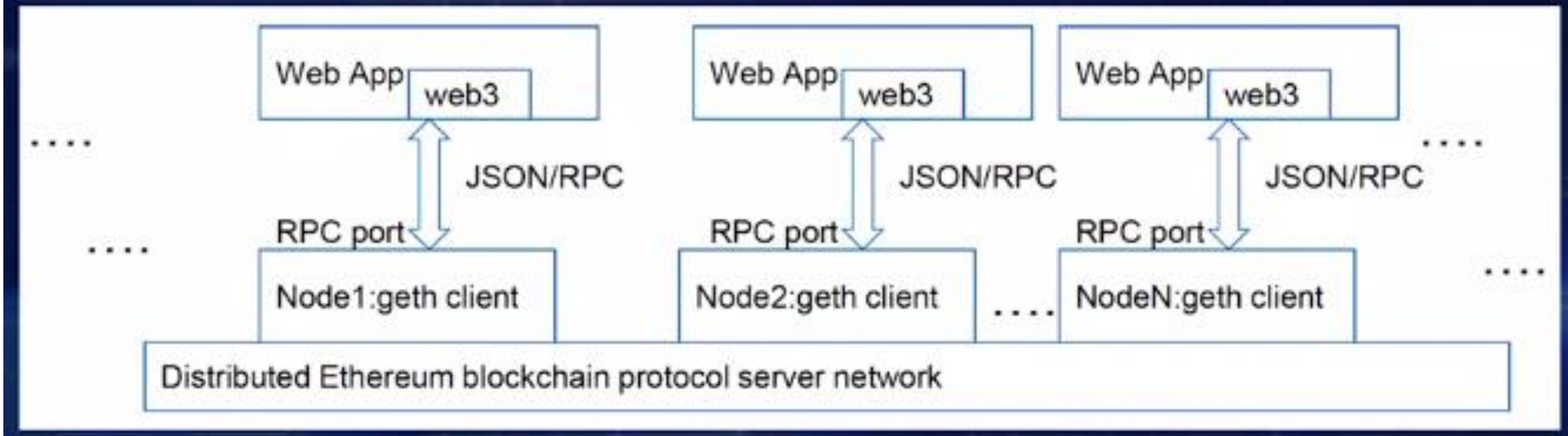

Traditional Web Application
Web browser
Internet
Web Server
Front-end (HTML, CSS, JavaScript)
Interact
Back-end (JSP/ASP/PHP/Node.js/...)
Save/Retreive state
Database
(a)
Decentralized Application (DApp)
Web browser
Internet
IPFS
Front-end (HTML, CSS, JavaScript)
Execute
Ethereum
Smart Contract
Ethereum Virtual Machine
Save/Retreive state
Blockchain
(b)


Smart Contract
bytecode
Storage
EVM
Node
Ethereum blockchain
(public, test or private)
App System (server)
Business logic
Web3.js
ORM
Javascript
engine
DBMS
engine
OS
Data
(in a server or
distributed)
UI
Web3.js
Web browser
OS
App System (UI enabling
a User to interact with
the dApp)

A Dapp, or decentralized application, solves a problem that requires blockchain services and blockchain infrastructure for realizing its purpose.


Compile Artifacts
Smart Contract
Remix compile process
ABI
Contract Bytecode
WebDeploy script
Gas estimates
Function hashes
Instance Bytecode

Architecture of a Dapp



Ethereum APIs

API = Application Programming Interface

Set of functions to invoke operations, access data, and store data

Ethereum APIs:

Management APIs - e.g., admin, debug, miner, personal, txpool

Web3 APIs- e.g., web3, eth, net


Admin API
Example: admin.addPeer()
admin.nodeInfo()

Debug API

Example: debug.dumpBlock(16)

Miner API

Example: miner.start()

miner.stop()

miner.start(6)

Personal API

Example: personal.newAccount()

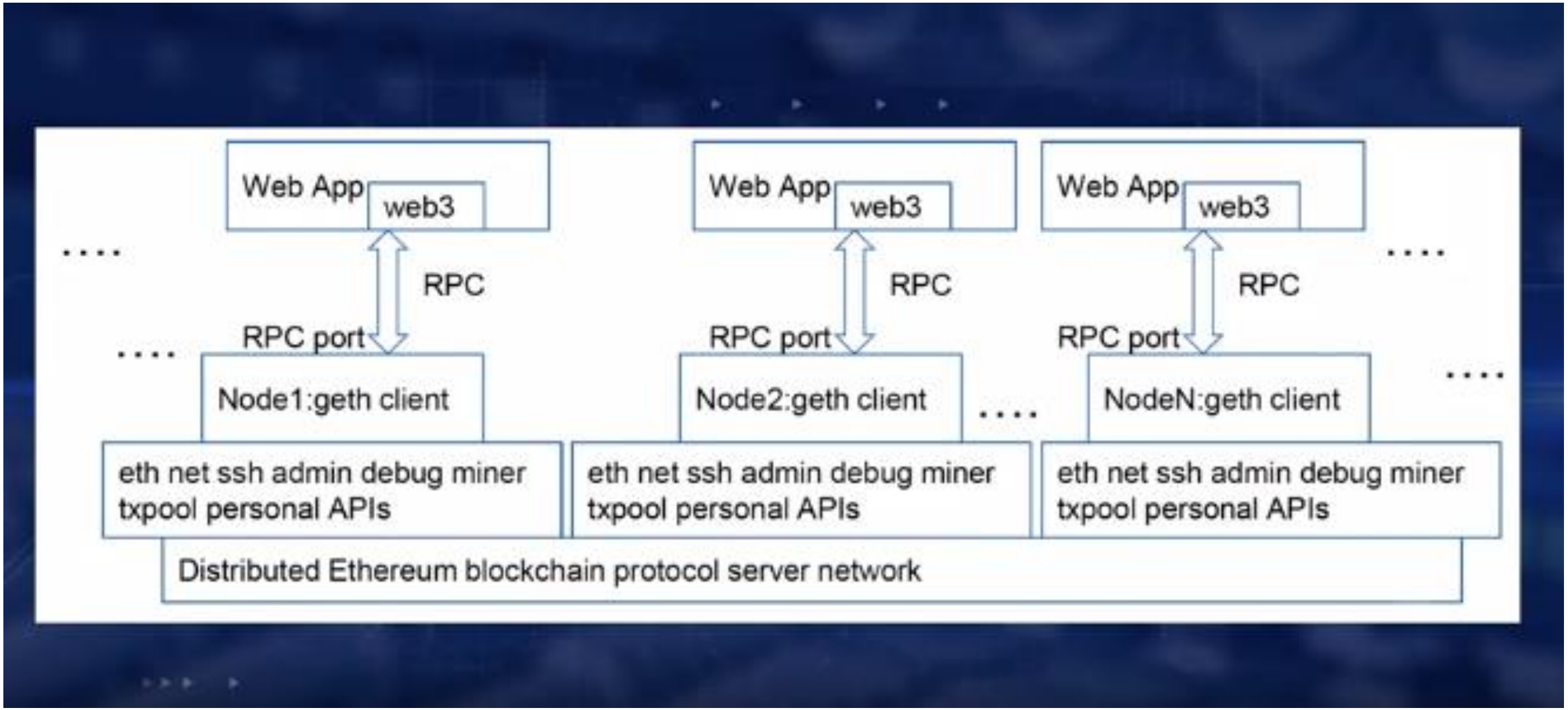
Txpool API

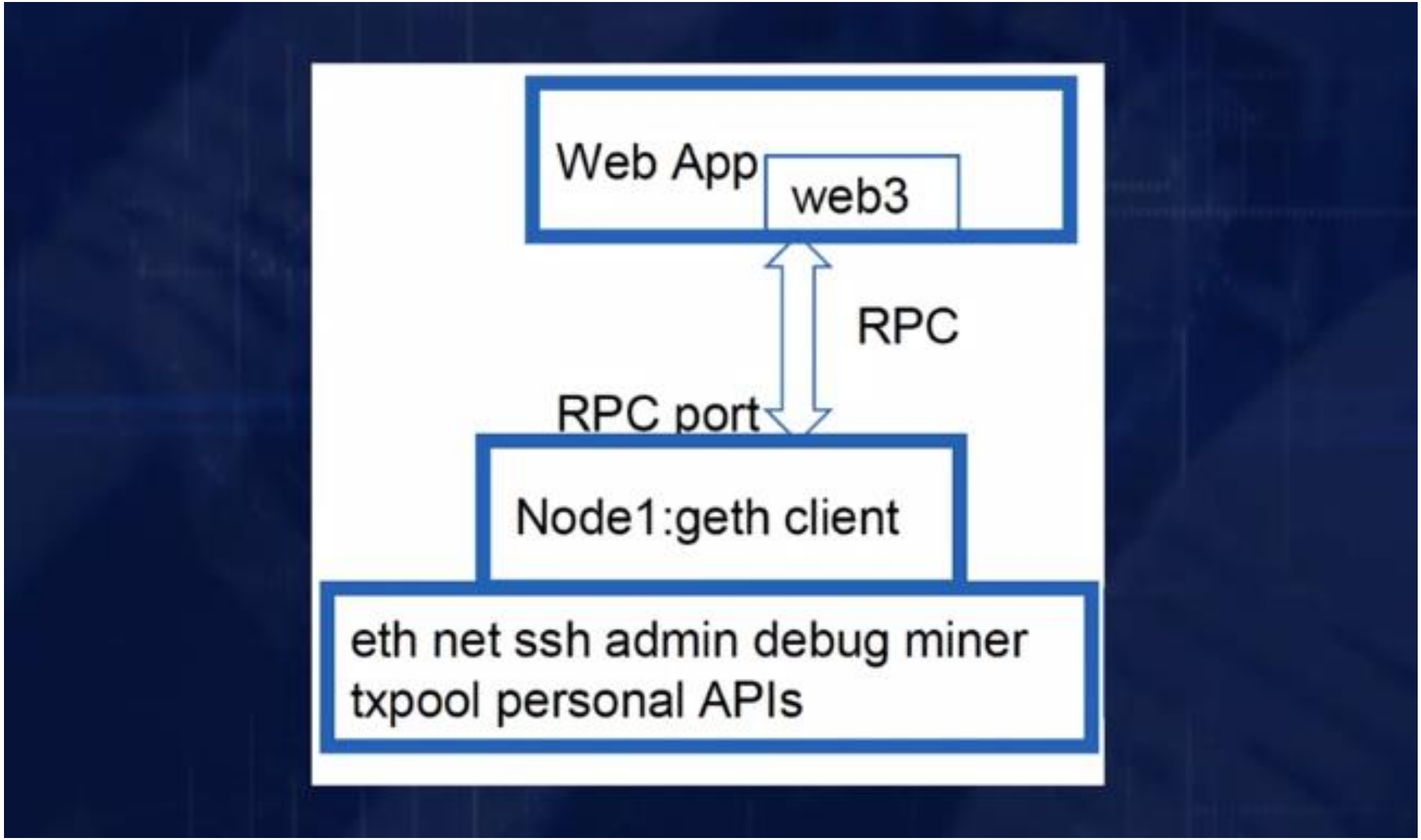
Example: txpool.inspect()

web3 API

```
<script type="text/javascript" src="../dist/web3.js"></script>
```


Whisper API
Example: web3.ssh


Web App
web3
RPC
RPC port
Node1:geth client
Web App
web3
RPC
RPC port
Node2:geth client
Web App
web3
RPC
RPC port
NodeN:geth client
eth net ssh admin debug miner
txpool personal APIs
eth net ssh admin debug miner
txpool personal APIs
eth net ssh admin debug miner
txpool personal APIs
Distributed Ethereum blockchain protocol server network


Web App
web3
RPC
RPC port
Node1:geth client
eth net ssh admin debug miner
txpool personal APIs