Decentralized Applications (Dapps) ?

- A decentralized application (dapp) is an application built on a decentralized network that combines a smart contract and a frontend user interface.
- A "smart contract" is simply a program that runs on the Ethereum blockchain. It's a collection of code (its functions) and data (its state) that resides at a specific address on the Ethereum blockchain.

What 'Dapps' are so special?

- No Owners :
  - Once deployed to Ethereum, dapp code can’t be taken down. And anyone can use the dapp’s features. Even if the team behind the dapp disbanded you could still use it. Once on Ethereum, it stays there.
- Decentralized – No one controlling power
- Open Source – Code is available to everyone
- Incentive – Crypto tokens are used to reward network users/miners etc.
- Protocol / Algorithm – To generate tokens that reward network participants
- Blockchain – Data stored cryptographically in a blockchain

How does Dapps work ?

- Dapps have their backend code (smart contracts) running on a decentralized network and not a centralized server. They use the Ethereum blockchain for data storage and smart contracts for their app logic.
- A smart contract is like a set of rules that live on-chain for all to see and run exactly according to those rules. Imagine a vending machine: if you supply it with enough funds and the right selection, you'll get the item you want. And like vending machines, smart contracts can hold funds much like your Ethereum account. This allows code to mediate agreements and transactions.
- Once dapps are deployed on the Ethereum network you can't change them. Dapps can be decentralized because they are controlled by the logic written into the contract, not an individual or a company.

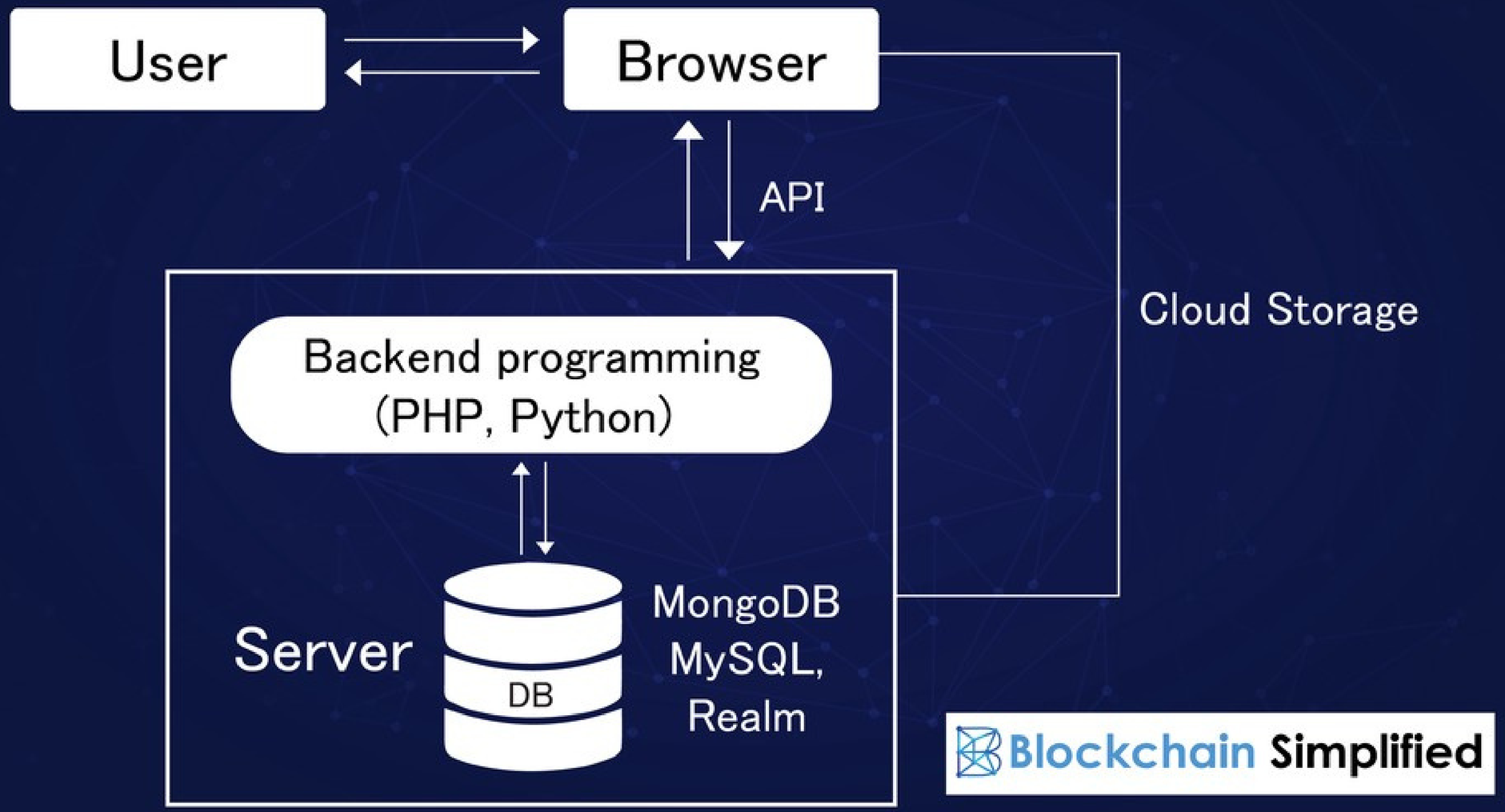
Benifits of Dapp Development

- Zero downtime
  - once the smart contract at the core of an app is deployed and on the blockchain, the network as a whole will always be able to serve clients looking to interact with the contract. Malicious actors therefore cannot launch denial-of-service attacks targeted towards individual dapps.
- Privacy
  - you don’t need to provide real-world identity to deploy or interact with a dapp.
- Resistance to censorship
  - no single entity on the network can block users from submitting transactions, deploying dapps, or reading data from the blockchain.

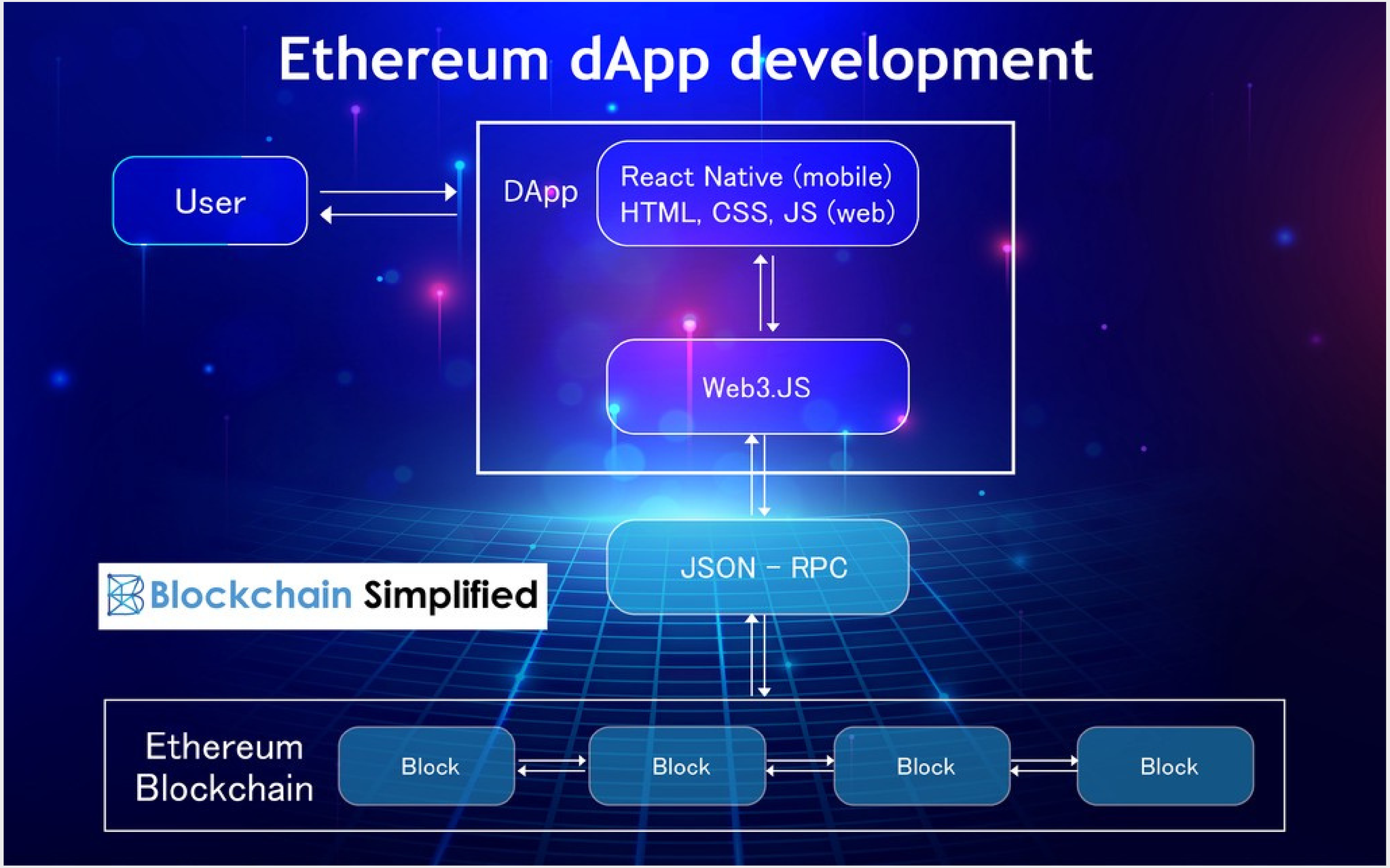
CONT...

- Complete data integrity :
  - data stored on the blockchain is immutable and indisputable, thanks to cryptographic primitives. Malicious actors cannot forge transactions or other data that has already been made public.
- Trustless computation/verifiable behavior :
  - you don’t need to provide real-world identity to deploy or interact with a dapp.
- Resistance to censorship :
  - smart contracts can be analyzed and are guaranteed to execute in predictable ways, without the need to trust a central authority.
  - This is not true in traditional models; for example, when we use online banking systems, we have to trust that financial institutions will not misuse our financial data, tamper with records, or get hacked.

Traditional App development



Blockchain Simplified


Ethereum dApp development
User
DApp
React Native (mobile)
HTML, CSS, JS (web)
Web3.JS
JSON – RPC
Blockchain Simplified
Ethereum
Blockchain
Block
Block
Block
Block

Steps to build Dapp:

- Step-1 :
  - Deciding on what technology you will use - Defining your environment.
  - You will need to determine exactly what technology you are going to use for these key areas of development:
    - Database
    - Frameworks
    - Hosting
    - Frontend
    - Frontend contracts
    - Contract‘s programing language
    - API‘s
    - Development server.

- Step-2 :
  - Setup the project.
  - This step will involve firstly installing the development framework.
- Step-3:
  - Code the application.
- Step-4:
  - Deploy and test your app.
- Step-5:
  - Launch your Dapp.