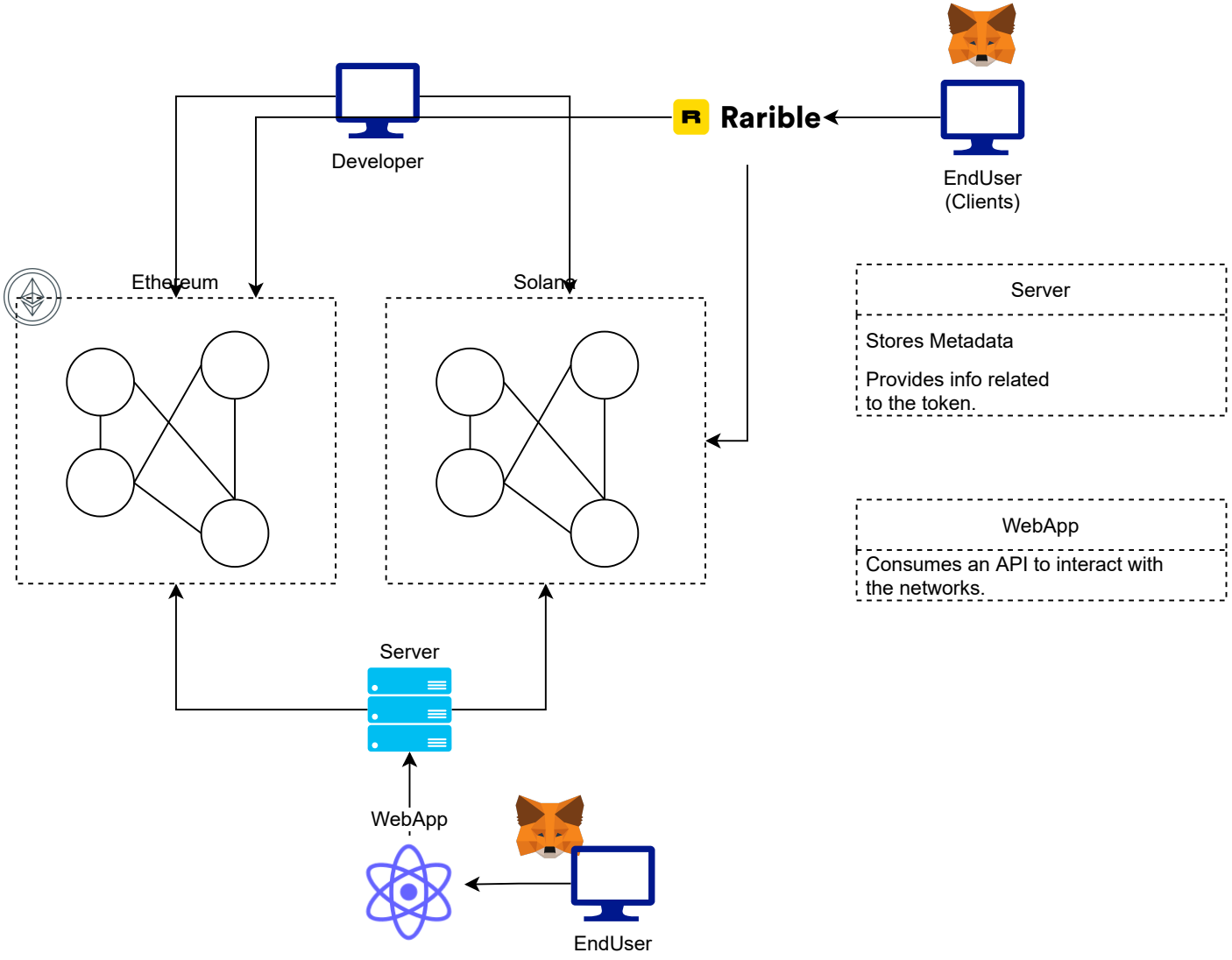
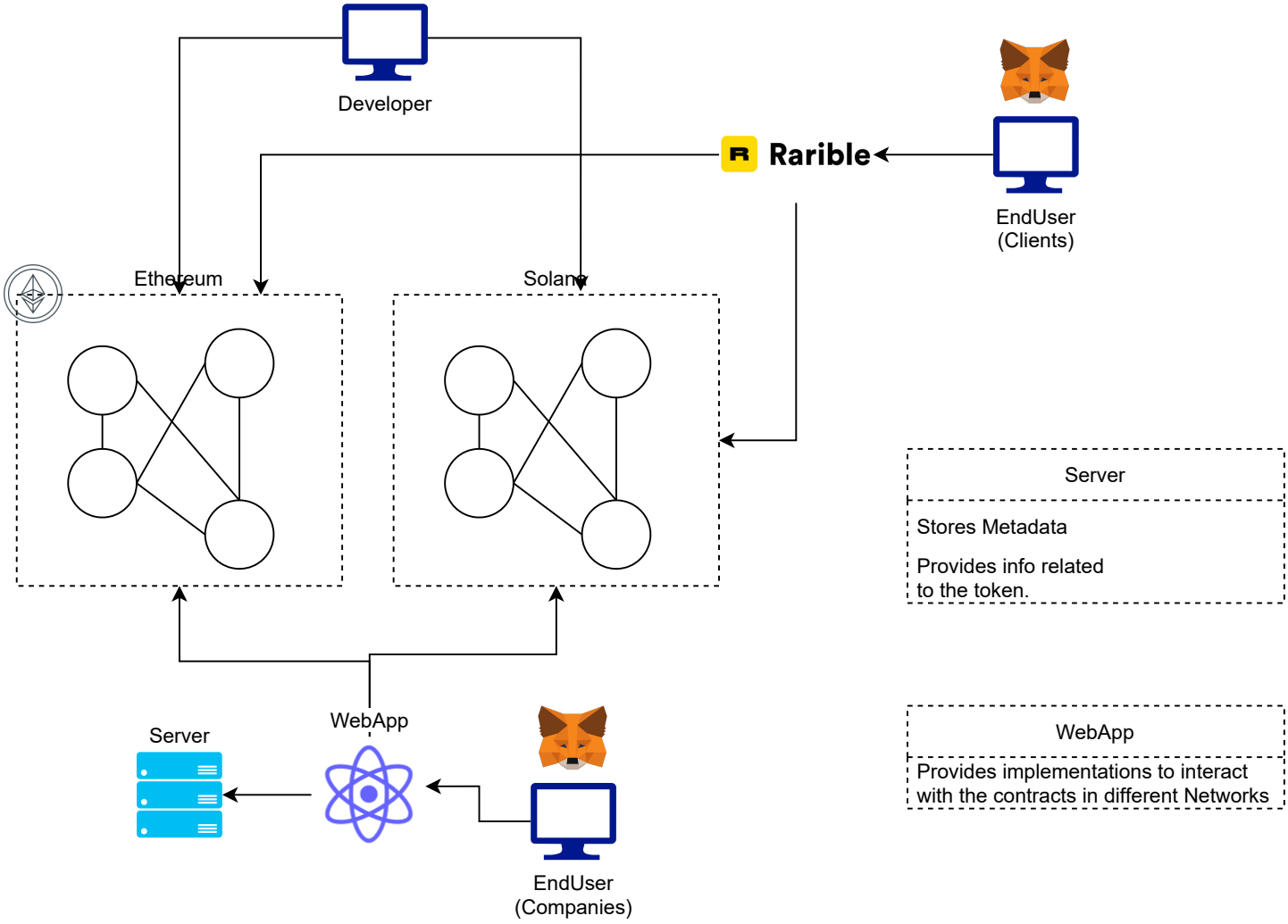

Developer
Rarible
EndUser
(Clients)
Ethereum
Solana
Server
Stores Metadata
Provides info related
to the token.
WebApp
Consumes an API to interact with
the networks.
Server
WebApp
EndUser


Developer
Rarible
EndUser
(Clients)
Ethereum
Solana
Server
Stores Metadata
Provides info related
to the token.
WebApp
Server
WebApp
Provides implementations to interact
with the contracts in different Networks
EndUser
(Companies)

MetaData Model	
NFTId	Id of the NFT created
ItemName	Name of the item
Descritption	Description of the Item
Payload	Extra data especified by the vendor. In Base64

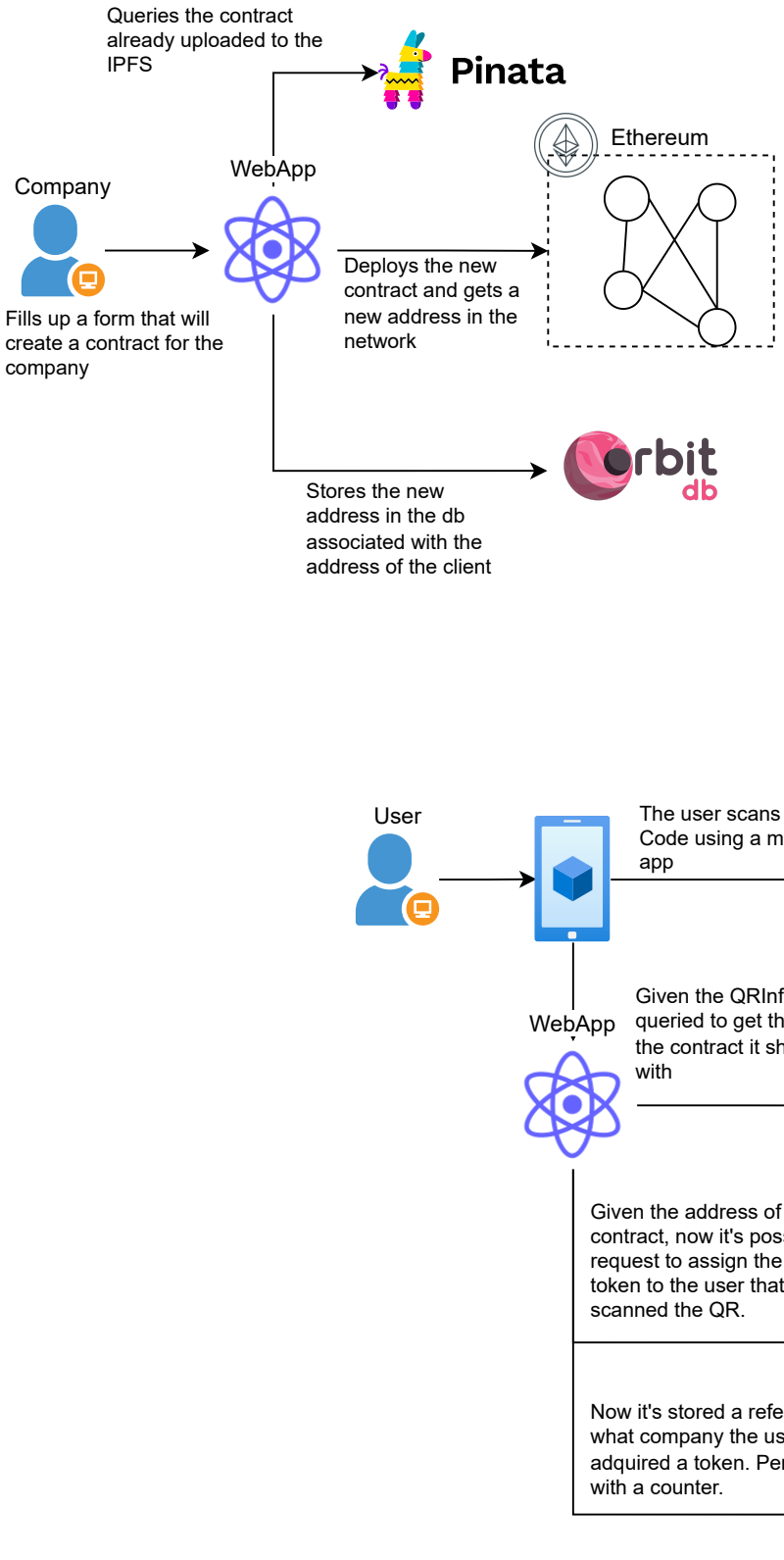
ContractStorage
exit83Address: address
clientAddress: address
mintedNFT: mapping(address => MetaDataModel)
mintableNFT: mapping(id => MetaDataModel)

Payload

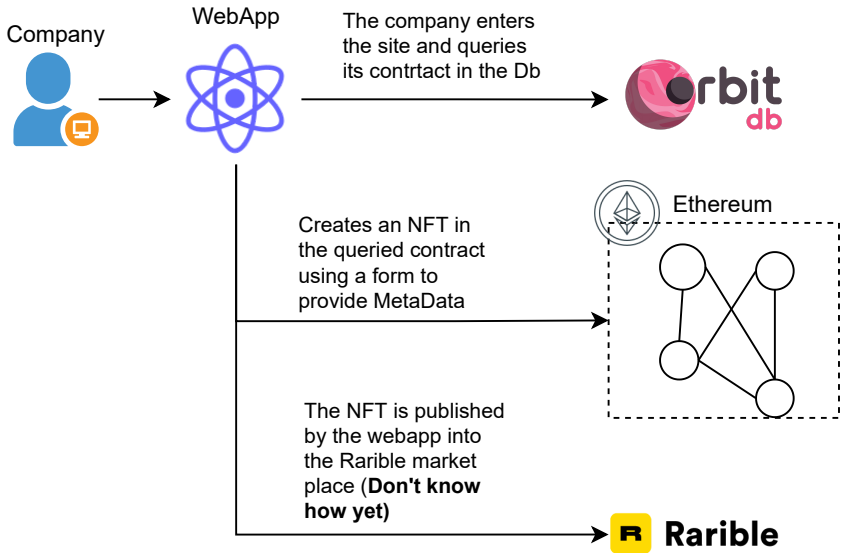
The payload defined by the vendor can be assigned to any content that can be valuable for them. Either a JSON file encoded in base 64 or even an image.

To consider: This can be standarized and make it more user friendly, by creating a key value form in the web app and then map it to a JSON file and encode it to a base64 format.

First use of the App



Creation of an NFT



NFT obtained when scanned

QRInfo	
CompanyName	Company Name
DateGenerated	Date when the QR code was generated. Used for validation.
ExpirationTime	Time expected to live the QR. This will reduce the lifetime and the chance to spread it among other users.
CompanyAddress	Address in the blockchain network

Reference to companies which the user has NFTs adquired.

This is done to reduce the load given to the contract. So next time the User gets into the platform will be displayed the list of companies rather than the NFTs. Once the user clicks on a company a request to the contract associated to the company will be queried to determine what NFTs the user has in that company.