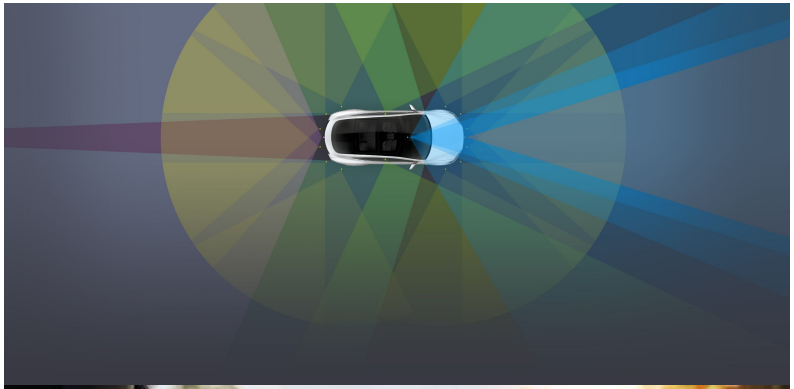
NFTGDP: Non-fungible token and Geo Data Processing

By: Zhaoming Yin
stplaydog@gmail.com

Zhaoming Yin had a joyful undergraduate study in Hunan University learning Software Engineering. After that he spent 3 years in Peking University under the supervision of Shiwen Yu for NLP research. Then he headed for Atlanta at GaTech and earned his PhD degree in the area of combinatorial research for bioinformatics and graph analysis advised by Prof David A. Bader. Zhaoming got his first job at Oracle in the Bay area working on Golden Gate, a database replication software. Two and half years later, he moved back to China and lives in Hangzhou, and worked in Alibaba Cloud on ODPS – a big data platform. After a short stint at TRIAS lab as Chief Algorithm Architect and Senior Software Expert in PayTM labs, he currently works for Google on Android Automotive Embedded systems.

Data Privacy is becoming more and more important these days

- **Euro GDPR**
  summary of GDPR fines
- **China DiDi Crackdown**
  didi crackdown.
- **US Privacy laws**
  privacy laws



Data privacy is not just politics

- ➔ **Should you driving data belong to the government?**
  NO!
- ➔ **Should you driving data belong to the Autonomous driving company?**
  NO!
- ➔ **Should your data belong to yourself?**
  YES!

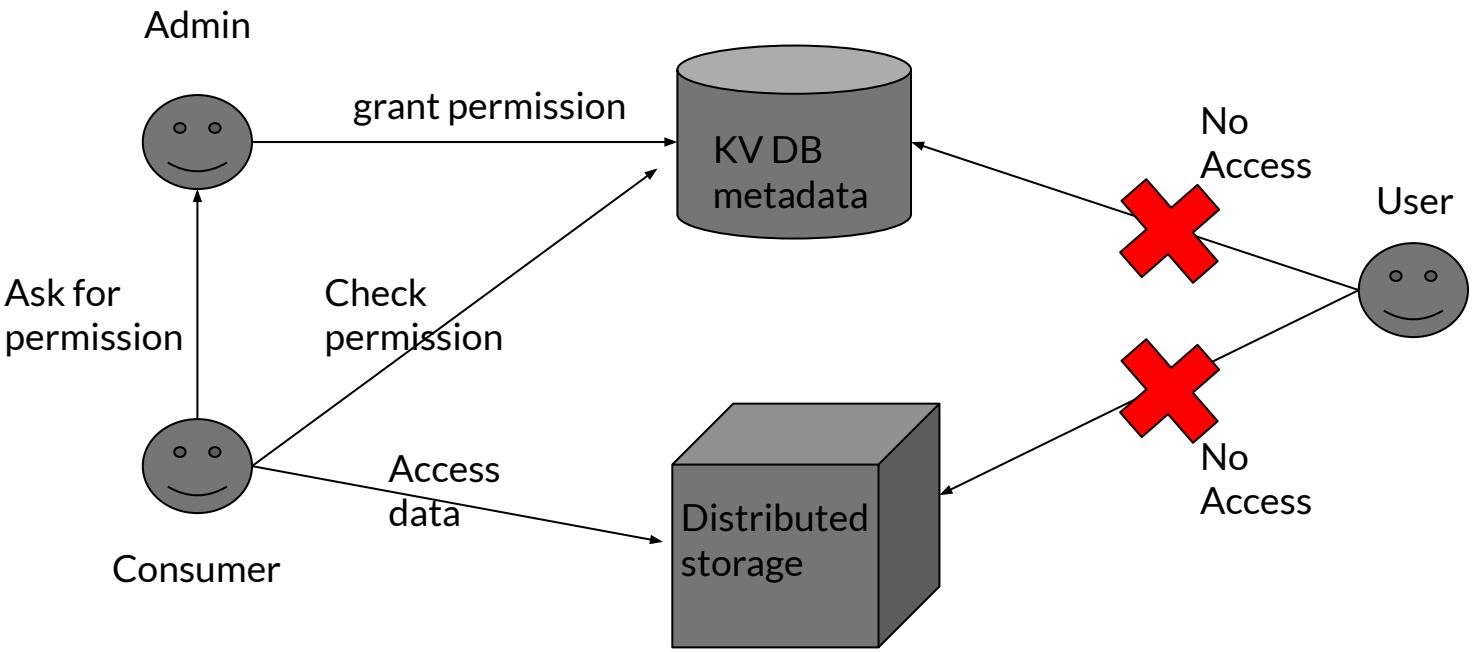
Limitations of centralized tech stack

- Tech companies promise they will treat your data nicely with their privacy 'technologies', but it turns out these technologies are more about self regulation, but lack of real crypto technology.

- Ask yourself: do you control any private keys on your data from: Apple, Amazon, Facebook etc?

- If no, then you don't have control over your data

How data privacy is enforced in centralized systems



Solution

Buyer

Install transferred model, and drive with seller's experience.

Data company

Buy data
Train Model

Autonomous Driving company

Seller

List driving data and ask for a price

Solution

Insurance company

Data company

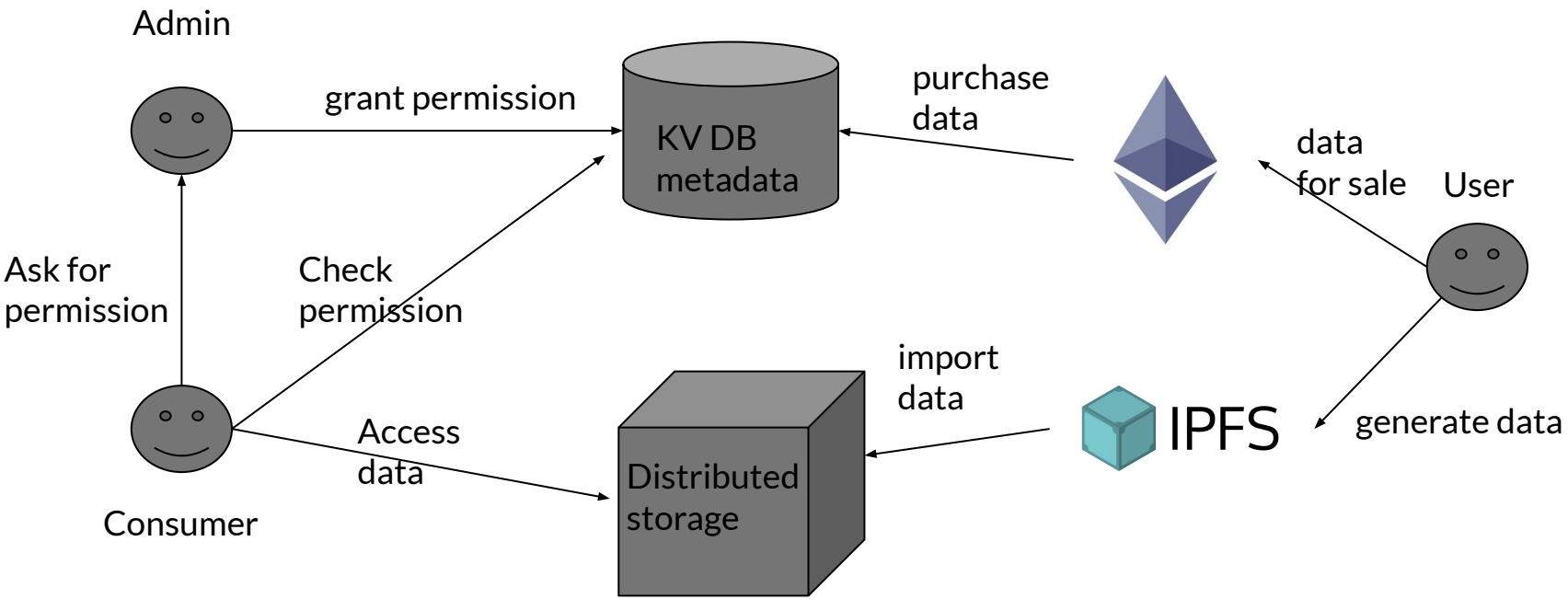
Seller

List driving data and ask for a price

Charge insurance based on driver's record data.

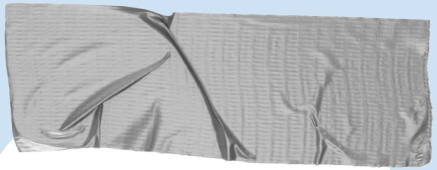
Pay as you drive!!

How data privacy is enforced in decentralized systems



OEMs

Platform

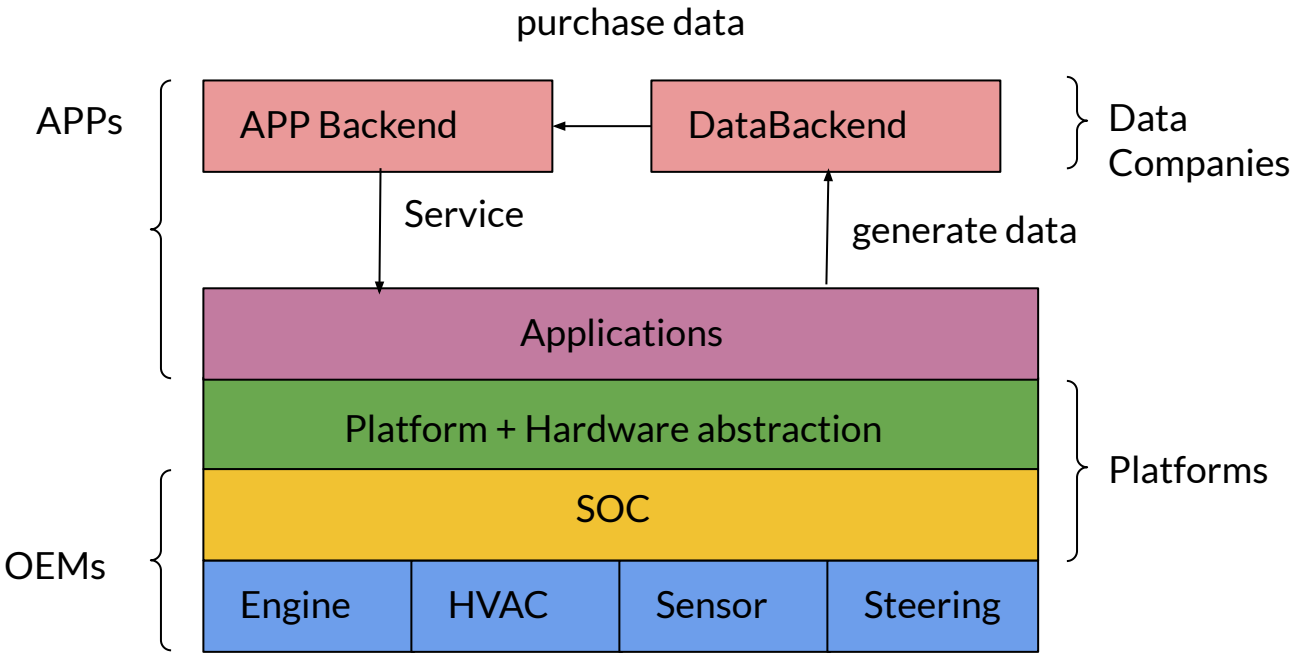


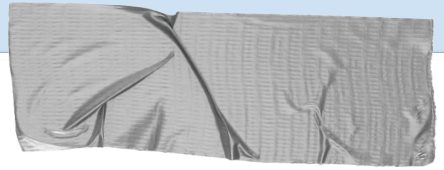
How?

Apps

Data Companies

System Architecture (Automotive)





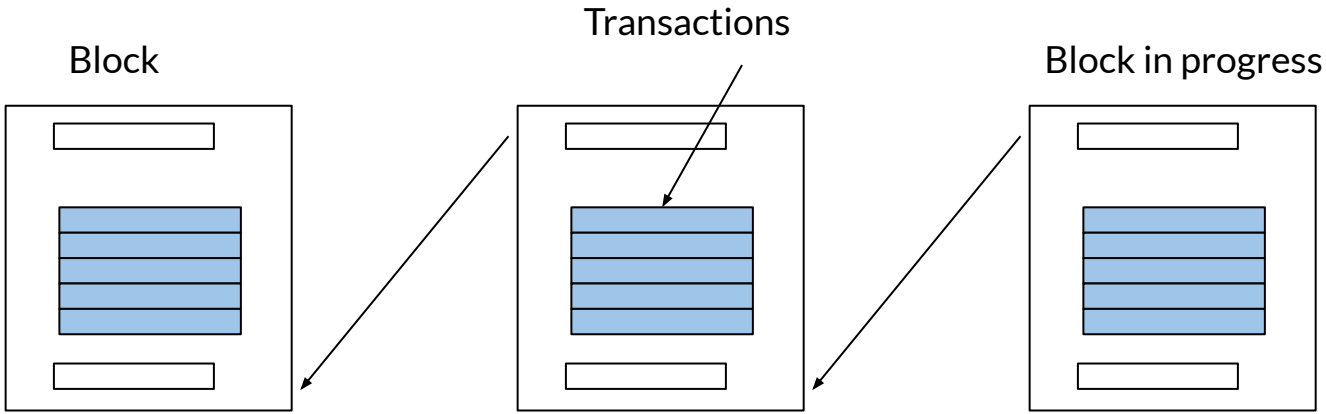
Data companies are just Smart contracts trading NFTs!

– Block-chain

Fully Distributed Database like BTC

Properties:

- Highly Secure
- Transparent
- Immutable



Smart contract

- Executable code
- Turing Complete (this is what smart contract different from BTC)
- A contract can hold fund and make the decision based on the contract logic (Yes, it's like a company!)

Smart contract

- Every node contains a virtual machine (similar to Java)
  - Called the Ethereum Virtual Machine (EVM)
  - **Compiles** code from high-level language to bytecode
  - Executes smart contract code and broadcasts state
- ***Every full-node on the blockchain processes every transaction and stores the entire state, so no centralized party decide the code execution result.***

Fungible token (ERC20)

- ERC20 is a token standard for create ethereum token contracts
  - ERC: Ethereum request for comments
- Any token contract adhere to specification follows ERC20, and most of ICOs are done on ERC20 (UNI, LINK, SUSHI, SHIB etc...)

Non-Fungible token (ERC721)

- Non-Fungible Tokens are not interchangeable and not divisible. These tokens are unique and cannot be replaced by another item.
- The ERC-721 Standard is the backbone for the creation and trading of non-fungible assets.
- Michael took Uber from Brooklyn to Long Island on 07/16/2021 at 7:00AM, this event will only happen once in the history, it's **Non-Fungible**

—

ERC721
ERC20

How?

IPFS is a public decentralized data storage system


```
encrypted encrypt(public_key,
                  trace_data,
                  camera_data,
                  sensor_data,
                  operation_data,
                  gps_data,
                  .....) {.....}

ipfs_address send_to_ipfs(encrypted)
```

Ethereum

➔ User encrypt data and

➔ put it into ipfs

ERC721
ERC20

How?

erc721.mint(ipfs_address,
owner_address,
) {.....}

Ethereum

➔ user create an NFT and put the decryption key in the NFT and ask for a price.

ERC721
ERC20

How?

➔ Data consumer buys the NFT and get the decryption key

Ethereum

```
erc20.transferfrom(buyer_address,
                   seller_address,
                   price
                  ) {.....}
```

```
erc721.transferfrom(seller_address,
                    buyer_address,
                    NFT_id
                   ) {.....}
```

ERC721
ERC20

How?

➔ Data consumer use the decryption key to decrypt data and use it for model training

Ethereum

ipfs_address erc721.tokenURI(NFT_id) {.....}

Challenges

- ## complexity to implement

  The complexity to implement is low, because ethereum is in charge of every complexity

- ## potential failures

  Every movement is irreversible, for example, if there is data leaks, no one can change back

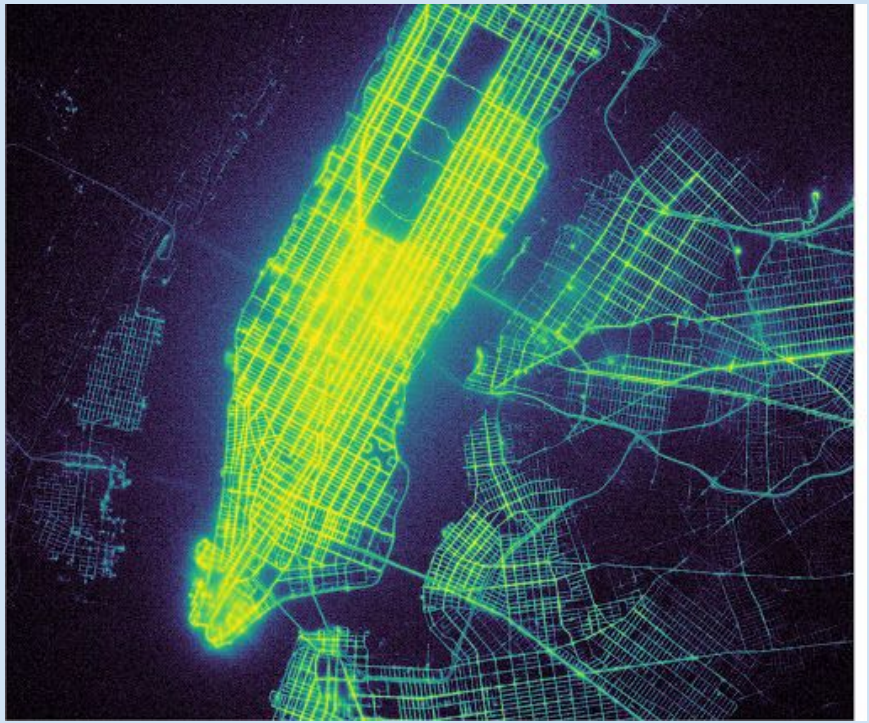
- ## critical aspects

The data flow speed is the key, currently the public block-chain has the high latency and low bandwidth issues.

- ## NFTs management

No human beings will be involved in the NFT management, everything will be controlled by smart contract.

Geo data representation in the public blockchain



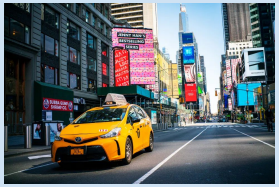
Example: NYC taxi data, This data is about the taxi trip information in New York City. Now there are 4 different type of operators:

Green, Yellow, FHV trip, high volume FHV (uber, lyft)

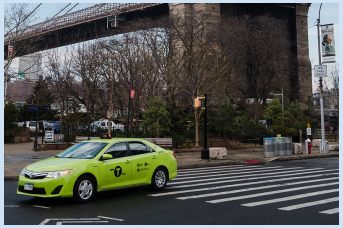
Geo data representation in the public blockchain

Example: NYC taxi data format

A	B	C	D	E	F	G	H	I	J	K	L	M	N	O	P	Q	R
VendorID	tpep_pickup_dat	tpep_dropoff_da	passenger_coun	trip_distance	RatecodeID	store_and_fwd_f	PULocationID	DOLocationID	payment_type	fare_amount	extra	mta_tax	tip_amount	tolls_amount	improvement_su	total_amount	congestion_surcharge
1	2020-01-01 0:28	2020-01-01 0:33	1	1.2	1	N	238	239	1	6	3	0.5	1.47	0	0.3	11.27	2.5
1	2020-01-01 0:35	2020-01-01 0:43	1	1.2	1	N	239	238	1	7	3	0.5	1.5	0	0.3	12.3	2.5
1	2020-01-01 0:47	2020-01-01 0:53	1	0.6	1	N	238	238	1	6	3	0.5	1	0	0.3	10.8	2.5
1	2020-01-01 0:55	2020-01-01 1:00	1	0.8	1	N	238	151	1	5.5	0.5	0.5	1.36	0	0.3	8.16	0
2	2020-01-01 0:01	2020-01-01 0:04	1	0	1	N	193	193	2	3.5	0.5	0.5	0	0	0.3	4.8	0
2	2020-01-01 0:09	2020-01-01 0:10	1	0.03	1	N	7	193	2	2.5	0.5	0.5	0	0	0.3	3.8	0
2	2020-01-01 0:39	2020-01-01 0:39	1	0	1	N	193	193	1	2.5	0.5	0.5	0.01	0	0.3	3.81	0
2	2019-12-18 15:2	2019-12-18 15:2	1	0	5	N	193	193	1	0.01	0	0	0	0	0.3	2.81	2.5
2	2019-12-18 15:3	2019-12-18 15:3	4	0	1	N	193	193	1	2.5	0.5	0.5	0	0	0.3	6.3	2.5
1	2020-01-01 0:29	2020-01-01 0:40	2	0.7	1	N	246	48	1	8	3	0.5	2.35	0	0.3	14.15	2.5
1	2020-01-01 0:55	2020-01-01 1:12	2	2.4	1	N	246	79	1	12	3	0.5	1.75	0	0.3	17.55	2.5
1	2020-01-01 0:37	2020-01-01 0:51	1	0.8	1	N	163	161	2	9.5	3	0.5	0	0	0.3	13.3	2.5
1	2020-01-01 0:56	2020-01-01 1:21	1	3.3	1	N	161	144	1	17	3	0.5	4.15	0	0.3	24.95	2.5
2	2020-01-01 0:21	2020-01-01 0:27	1	1.07	1	N	43	239	1	6	0.5	0.5	1.96	0	0.3	11.76	2.5
2	2020-01-01 0:38	2020-01-01 1:15	1	7.76	1	N	143	25	1	28.5	0.5	0.5	4.84	0	0.3	37.14	2.5
1	2020-01-01 0:15	2020-01-01 0:27	3	1.6	1	N	211	234	2	9	3	0.5	0	0	0.3	12.8	2.5



Purchase data



Ethereum

user create an NFT and put the decryption key in the NFT and ask for a price.

User encrypt taxi request

put it into ipfs

IPFS

UBER

lyft

Provide service

IPFS data Sotrage

Add files to IPFS

```
ipfs add taxi_request -r
added QmdNMBVqQJ9baQDAJMUwci8LzVK2a52cTLEHoinaKFEdFY
taxi_request
```

Data Privacy and data Sharing

Data privacy and data sharing in smart contracts are all decentralized, they provide protection of different granularity

- only owner of contract can access
-

```
abstract contract Ownable is Context {
  address private _owner;
  event OwnershipTransferred(address indexed previousOwner, address indexed newOwner);
  function owner() public view returns (address);
  modifier onlyOwner();
  function renounceOwnership() public onlyOwner;
  function transferOwnership(address newOwner) public onlyOwner;
  function _transferOwnership(address newOwner) internal;
}
```

Data Privacy and data Sharing

Data privacy and data sharing in smart contracts are all decentralized, they provide protection of different granularity

- only owner of contract can access
- role based access control
-

```
abstract contract AccessControl is Context {
    function hasRole(bytes32 role, address account) public view returns (bool) {
    function getRoleMemberCount(bytes32 role) public view returns (uint256) {
    function getRoleMember(bytes32 role, uint256 index) public view returns (address)
    function getRoleAdmin(bytes32 role) public view returns (bytes32) {
    function grantRole(bytes32 role, address account) public virtual {
    function revokeRole(bytes32 role, address account) public virtual {
    function renounceRole(bytes32 role, address account) public virtual {
    function _setupRole(bytes32 role, address account) internal virtual {
    function _setRoleAdmin(bytes32 role, bytes32 adminRole) internal virtual {
    function _grantRole(bytes32 role, address account) private {
    function _revokeRole(bytes32 role, address account) private {
}
```

Data Privacy and data Sharing

Data privacy and data sharing in smart contracts are all decentralized, they provide protection of different granularity

- only owner of contract can access
- role based access control
- discretionary access control (access control list)

```
function tokenURI(uint256 tokenId) external view
returns (string memory) {
        require(
            _msgSender() == ownerOf(_tokenId),
            "Only NFT owner can update"
        );
        // return token URI address in IPFS
}
```

Data ownership transfer protocol

```
contract IGeoDataMarket  {

   // meta data contains the IPFS info
    function create_token_NFT (
            address currency_token,
            string calldata meta,
            uint256 price,
            uint256 unit_size,
            uint256 num_units
    ) external ;

    function buy_token_NFT (
            uint nft_id,
            uint token_id
    ) external ;

    function sell_token_NFT (
            uint nft_id,
            uint token_id,
            uint256 price
    ) external override;

    function redeem_token_NFT (
            uint nft_id,
            uint token_id
    ) external override ;
}
```

Data scam detection and prevention

We can use Oracle to judge the correctness of data, such as Augur

Data auction theory and implementation

Bounce protocol can be used for data auction
crypto kittens arbitratge