CONTRACTNET

PROVIDES 3 REALLY SMART SOLUTIONS TO THE PROBLEMS AT THE CUTTING EDGE OF BLOCKCHAIN AND IOT

ContractNet combines a number of decentralized technologies into a unique solution for IoT

contractnet.com

CONTENTS

1/ INTRODUCING CONTRACTNET

ContractNet is a new public, permissionless blockchain which is purpose-built for the storage and sharing of IoT data streams. It provides entrepreneurs with a platform to host their businesses, using blockchain, smart contracts and IoT technology.

ContractNet's philosophy and reason for existence is to be the global exchange of IoT data, and the platform for some of the most exciting applications on the decentralized web.

It provides a solution to many of the current problems associated with combining blockchain, smart contracts and IoT technologies.

2/ DEFINING THE PROBLEMS

The purpose of IoT is the sharing of device data, and the analysis of this data for business purposes. While there are many people talking about the benefits of blockchain and smart contracts for IoT, very few real solutions exist.

Some of the *technical challenges* for IoT include:

1. Blockchain:
   - Scalability
   - Speed and latency
2. Smart contracts
   - Elimination of coding errors
   - Management of security risks
   - An oracle solution that allows for accessing and trusting external information
3. IoT
   - Storage of vast amounts of time--based data
   - Embedding of IEEE IoT standards into oracles
   - Global interfaces

Some of the *business challenges* include:

4. The lack of a platform specifically optimized for IoT and available across industries and for developers
5. Monetization options for all participants
6. Attracting participants – miners, developers, industry owners, investors – to the platform
7. Creating value for the underlying cryptocurrency

Perhaps an example will help to clarify the problems and at the same time provide a good use case:

At the Consensus 2017 Blockchain conference in New York, Toyota described a roadmap for developing a Blockchain for autonomous vehicles. It explained that to perfect this software, a trillion miles of data would be required for their machine learning algorithms. At that point, Toyota had accumulated several million miles. If it were possible to access similar data from competitors (eg BMW) and to share this data, the target could be reached. This could benefit the industry as a whole.

If we use this example, we can see that the *commercial problem* is in how to share data with competitors in a way that preserves ownership, ensures that the same network and technology standards are being applied, ensures integrity of the data, and provides a fair way to pay for the data if that is what the owner wanted.

Blockchain might solve the problems associated with ownership and integrity of data and credibility of vendors. Smart contracts can set up the conditions for sharing and payment. An internal currency can provide for the monetization of data. IoT devices can provide data streams to be shared. The IEEE (Institute of Electronics and Electrical Engineers) has defined a comprehensive list of standards that encompass the IoT.

However, the *technical problem* is in how to bring these together. In particular, millions of hours of IoT time-series data would have to be stored. The cost to do this on a blockchain is prohibitive – between 2,000 to 8,000 times more than storing it in regular cloud storage.

The *marketing problem* is in attracting IoT device owners, developers, miners and adopters to the platform.

3/THE CONTRACTNET SOLUTION

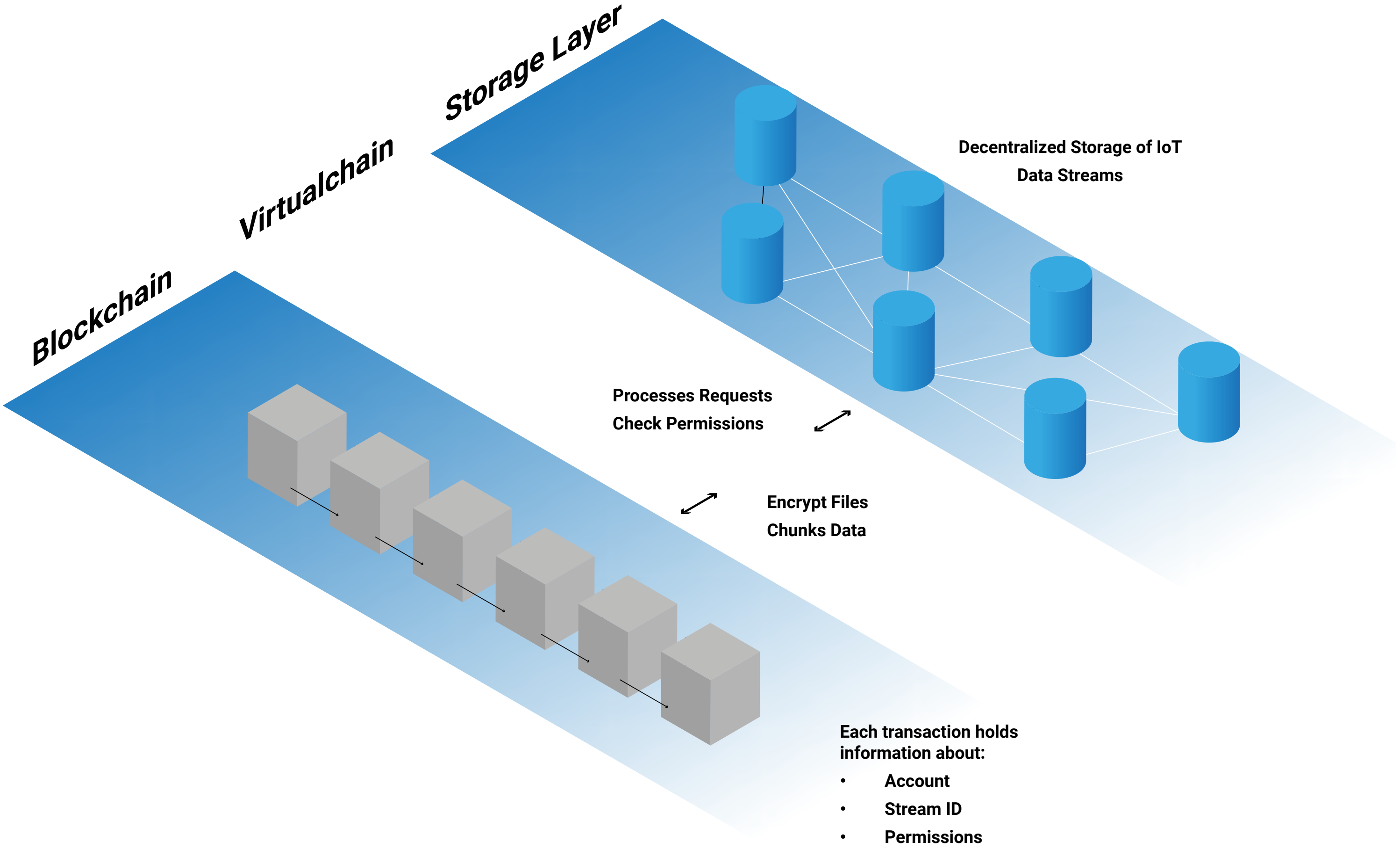
What ContractNet proposes, is to take a number of different decentralized technologies, and combine them into a hybrid solution that addresses all the problem areas.

SOLUTION #1: SEPARATE THE STORAGE FROM THE BLOCKCHAIN

The first solution is to *separate the storage from the blockchain*, and implement a "virtual-chain" as an intermediary between it and the storage layer.

This separation is shown in the following diagram, based on the work of Shafagh, Hithnawi, Burkhalter, Duquennoy in "*Towards Blockchain-based Auditable Storage and Sharing of IoT Data*" (https://arxiv.org/pdf/1705.08230.pdf):

Figure 1: A platform with 3 layers to solve the data storage problem



The platform will be broken up into 3 main parts:

BLOCKCHAIN

The blockchain is for transactions, execution of smart contracts and storage of access control data.

The IoT data in the system will be structured into streams. Each sensor will provide one stream of data. Each stream of data will be registered on the blockchain with its “stream identifier”. Each stream will have a clearly identified “owner”.

There is an account associated with each owner. This account has a password-protected private key. Each owner can register multiple streams to an account. The account defines ownership. The stream identifier defines the specific sensor.

Permission will be needed to share the data. If data is shared, this is a “transaction”. Every transaction on the platform will contain metadata describing both stream ownership and access control permissions. These access transactions will be written to the publicly auditable blockchain.

Payments for each transaction are via the underlying currency, the CNET.

STORAGE LAYER

The storage layer is for storage of data streams. Storing of data directly on the blockchain is too public, and too expensive.

ContractNet proposes to optimise IPFS (Inter Planetary File System) technology for this storage. The data stream will be encrypted (and periodically re-encrypted) using a key generated from the virtualchain, and will be stored using the same stream identifier committed to the blockchain.

Compared with storing data on the blockchain, this will, at present time, work out to be 2,000-8,000 times cheaper and could eventually be cheaper than cloud storage.

In addition, just like the “gas” that is used to pay miners for computational power, miners can be incentivized to provide storage to the network.

The benefits of using such a storage solution are:

- De-duplication
- Self-Distribution
- Peer-to-Peer Transfer
- Archiving
- Directory Browsing

VIRTUALCHAIN

The virtualchain is the logic layer that sits between the blockchain and the storage layer.

As in Ethereum, there are two types of accounts on the blockchain:

- » *Contract* accounts (smart contract owned account)
- » *Regular* accounts (normal user accounts)

Requests for the sharing of data are initiated by a transaction. For example, a Toyota smart contract requests the data from the BMW sensor on the headlight of their Car #10. This is accomplished by issuing a transaction from the Toyota contract account, to the BMW user account (by wallet address), with a file stream identifier to specify the stream required. This transaction is picked up by the virtualchain, which will query the blockchain history to confirm that the Toyota smart contract public key has permission to do so. If the necessary permission exists, the virtualchain will fetch the file from the storage layer, decrypt it and stream it to the requester.

The reason the streams are encrypted is to prevent a malicious storage node from handing out data without permission. The key management for this encryption will be implemented using low-cost key renewal with regression.

The virtualchain performs an important function in managing access. It will allow, for example for owners to revoke access to certain data, or for access to be granted for limited time ranges. The key management approach will be to frequently update keys and re-encrypt streams.

We are providing the platform that will allow developers to define sharing permissions on a blockchain, and smart contracts can be used to define the terms of how these streams are shared. Each smart contract will have a contract account which can transact with an owner account and can query the streams within that account.

Problems solved by

Solution #1

1. Smart contracts
   - Management of security risks
2. IoT
   - Storage of vast amounts of time-based data
   - Global interfaces

SOLUTION #2: ADDRESS PROBLEMS INHERENT TO BLOCKCHAIN AND SMART CONTRACTS

ContractNet proposes to use a variety of new technologies to address the *underlying challenges in blockchain itself, and in smart contracts*,

Speed and latency

The ContractNet blockchain is a fork from Ubiq and Ethereum. In its initial formulation, the ContractNet blockchain targeted a *block-time* of about 90 seconds. This optimized it for lower memory and low latency IoT devices, and provided the capacity to process 6,000 transactions per block (3 times as much as Bitcoin). The optimal block time is still being considered in light of better solutions for IoT as outlined above.

Multi-pool mining

The network will use the *Flux Difficulty Algorithm* to prevent centralized attacks (such as 51% attacks) during the initial stages of mining. This is done by dynamically adjusting the difficulty when there are rapid changes in hashrate. At the same time, the Flux algorithm also ensures more consistent block times during volatile/variable hash rates.

Scalability

By moving the storage of IoT data streams off-chain into a decentralized platform, the scalability of the Blockchain is preserved. ContractNet's approach to implement a virtualchain as an intermediary between the blockchain and the data plane creates a massive cost and efficiency saving when compared to solutions which try to store data within the blockchain.

Impact of errors in coding

Errors in smart contracts are impossible to roll back, so coding errors must be eliminated. ContractNet proposes to do this by implementing FSolidM, which addresses common vulnerabilities in the Solidity language and the greater EVM ecosystem. This solution designs contracts as Finite State Machines (FSM) and implements safe coding patterns in the form of plugins. It aims to eliminate errors in contracts from the outset rather than trying to fix them afterwards.

Security risks of smart contracts

Because a smart contract requires processing and exists over time (as opposed to single action transactions on a blockchain), it is vulnerable to attacks and to confidentiality and integrity issues. Use of the *FSolidM* technology will to a large extent remove these problems.

Accessing data outside of the blockchain

Smart Contracts inherently cannot access data outside their network. They rely on “oracles” to provide the link between the smart contract and the IoT device. An oracle could be software (eg a website or desktop app) or hardware (eg a smart sensor). The purpose of the oracle is to create a transaction, that is verified by all the nodes, and pushes data into the blockchain.

The problem associated with this is that current oracle solutions rely on data from centralized services, and so remove the smart contract’s ability to be trustless and tamperproof.

ContractNet solves these challenges in the following ways:

Rating & Review

ContractNet will rigorously validate the any new Oracles, to ensure that they pass a stringent set of standards, including code review and a thorough KYC (Know Your Customer). In addition, other developers who use these Oracle plugins in their smart contracts will be encouraged to rate and review their experiences, which will either bolster confidence or raise a flag for the greater community. Should an Oracle developer not be able to maintain at least a 4 out of 5 star rating, that plugin will be pulled from the hub.

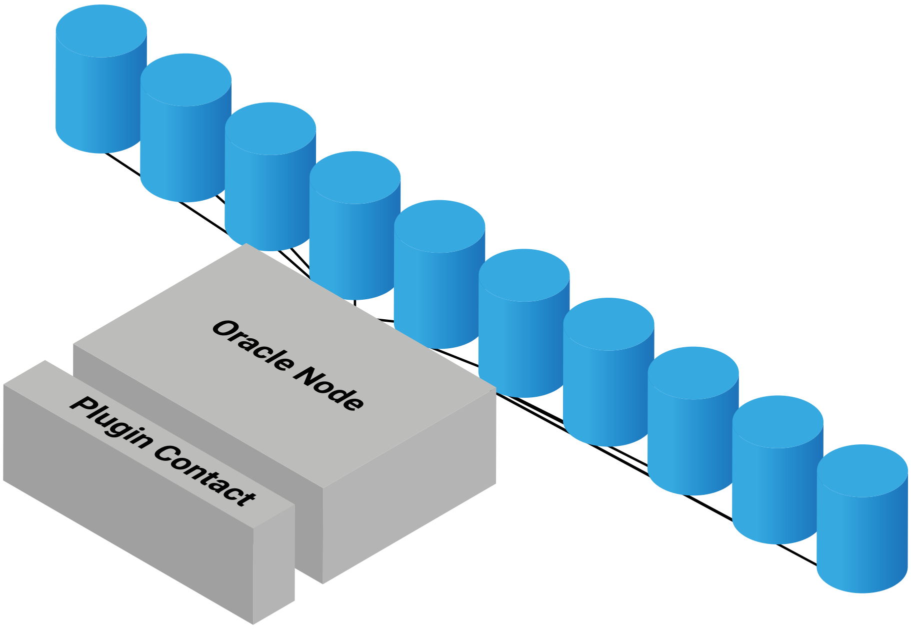
By Design

As a result of ContractNet’s technical approach to storing IoT data streams, and the nature of Blockchain, data stream writers will be hesitant to write anything but the “truth” as this data will be tamperproof and unalterable. This will serve as a record of fraud without recourse to “cover it up”.

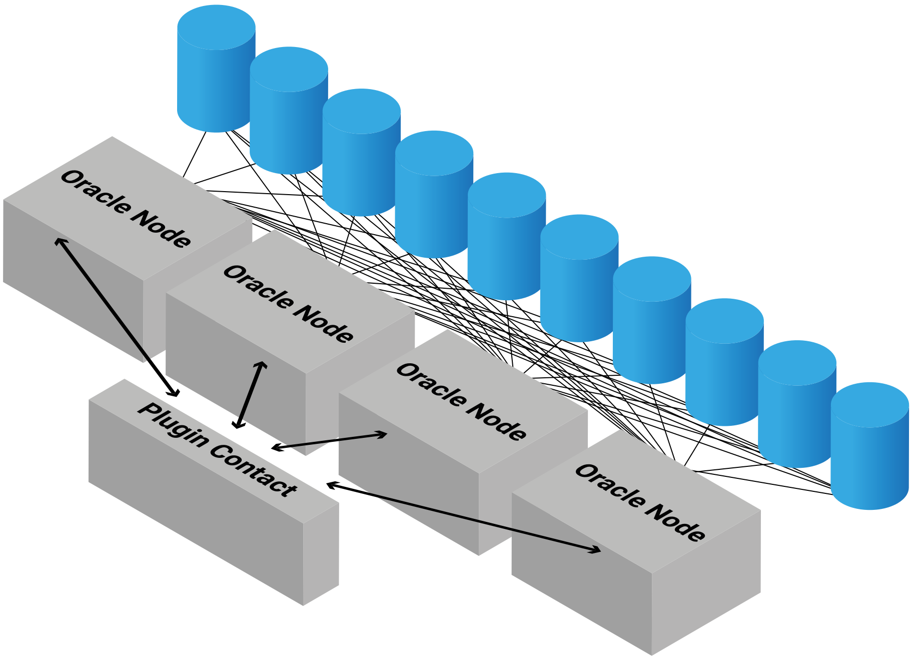
Architecture

In addition to rating and review, to further dilute the overall influence of a rogue node, the following strategies and patterns will be recommended to Oracle developers to further bolster their rating and the overall security of their offering. In essence, we are introducing decentralization to the Oracle layer:

» **Multi-Sensor Approach** - In order to ensure higher redundancy in mission critical implementations, multiple identical sensors can be deployed and the node can aggregate these values together by either delivering an average value, or by removing statistical outliers that could be supplying incorrect readings.

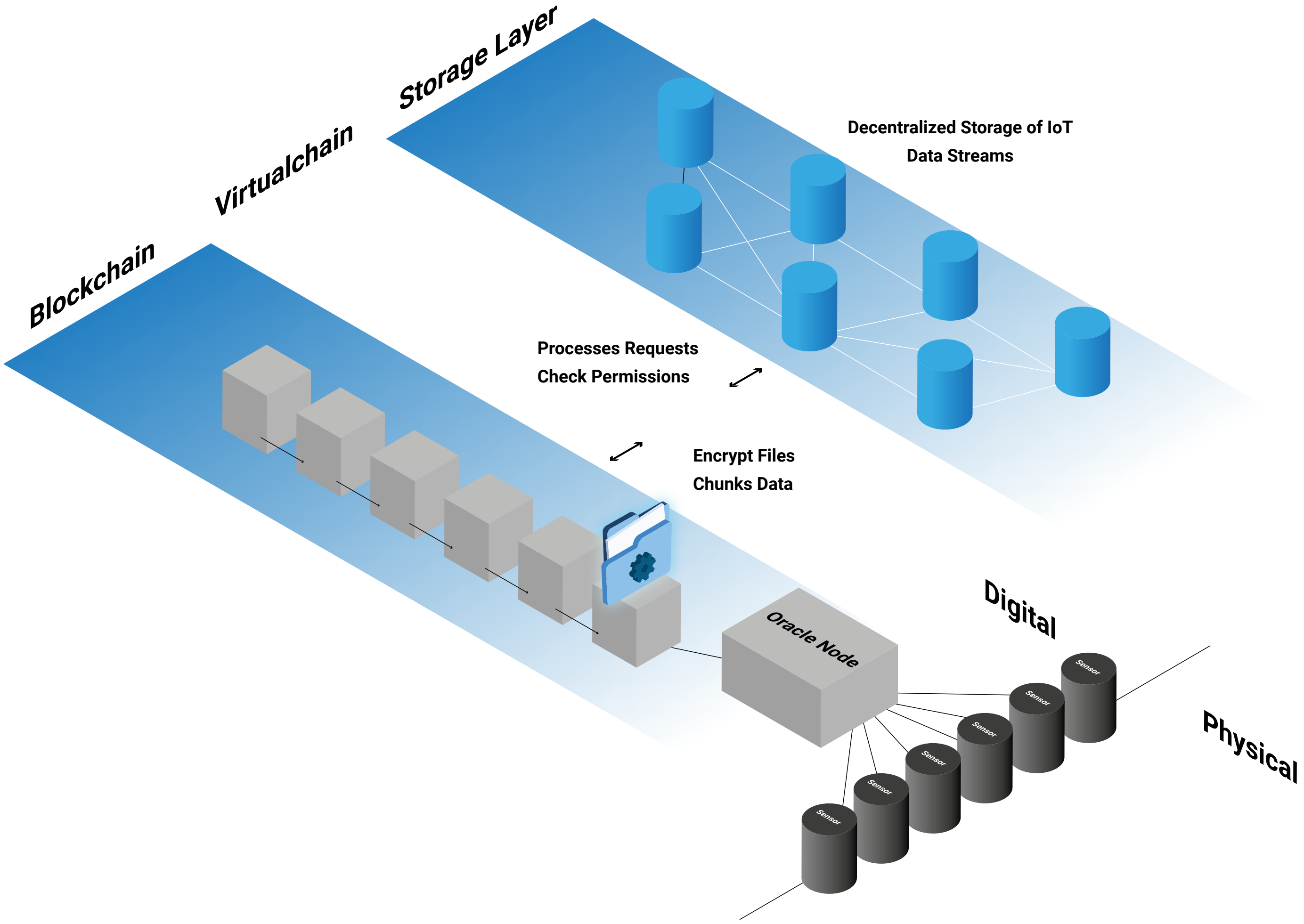


» **Multi-Node Approach** - Should a single node be compromised, a multi-node, multi-sensor approach could be adopted for further redundancy and higher availability. With this approach, in addition to aggregating sensor data, node data can be queried and compared, either by aggregation of values, or by removing statistical outliers. A comparably lower cost consensus algorithm such as BFT (Byzantine Fault Tolerance) can be implemented to further bolster the trust in this layer.



The full solution to the integration of blockchain, smart contracts, IoT and oracles is shown in the following diagram:

Figure 2: The ContractNet solution - at the interface of blockchain, smart contracts, IoT and oracles



Problems solved by

Solution #2

1. Blockchain
   - Scalability
   - Speed and latency
2. Smart contracts
   - Elimination of coding errors
   - Management of security risks
   - An oracle solution that allows for accessing and trusting external information
3. IoT
   - Embedding of IEEE IoT standards into oracles
4. Business
   - Availability of a platform specifically optimized for IoT

SOLUTION #3: ATTRACT MINERS, IOT DEVICE OWNERS, ADOPTERS AND DEVELOPERS

The platform uses a base currency (CNET) to act as a store of value, and also as the unit of exchange for payment, computation and storage.

Use of the CNET as the medium of exchange on the platform also allows for monetization options for participants:

» Miners will earn CNET in return for their computational power in the Proof-of-Work consensus mechanism underpinning the blockchain

» In addition, miners can earn additional CNET for providing storage capacity on their computers

» IoT device owners can sell their streams of data to developers or other users

» Developers have access to an open source, fully optimized platform on which to develop and monetize their own Dapps. These Dapps provide physical input into the Blockchain

» Developers can create new Oracles and sell these on the Oracle Hub.

» ContractNet will earn an income from every transaction on the blockchain

» Early adopters can invest in the development of this exciting network through our token sale and own a stake in the asset

Problems solved by

Solution #3

- Monetization options and earning potential for all participants will drive usage of the platform, the value of the CNET coin and profitability of the ContractNet business itself.

IN SUMMARY, CONTRACTNET – THE GLOBAL EXCHANGE FOR IOT DATA

1. There is a long list of solutions provided by the ContractNet platform:
2. Decentralized, resilient and auditable access control management of data streams
3. Stream ownership and cryptographically secure sharing
4. Secure data storage
5. IoT compatibility
6. Blockchain: Scalability; speed and latency
7. Smart contracts
    a. Elimination of coding errors
    b. Management of security risks
    c. An oracle solution that allows for accessing and trusting external information
4. Availability of a platform specifically optimized for IoT, providing a secure platform for Dapps
5. Business success
    a. Monetization options for all participants
    b. Earning options for ContractNet itself
    c. Value driver for CNET coin

ContractNet is at the cutting edge of a technology wave that promises to transform the way many industries and businesses function.

contractnet.com