Are Decentralised Applications the future of the Internet?

Andoni et al define a decentralised application (Dapp) as an application that can operate autonomously, through smart contracts, that run on a decentralised computing, blockchain or other distributed ledger systems (Andoni et al 2019). This technology has risen in popularity recently, particularly in the cryptocurrency industry. In this essay, I will discuss the technical advantages and disadvantages that Dapps have over their centralised counterparts in the areas of scalability, dependability, security, privacy and economic cost. I will also evaluate how society's desire for regulation and the permeance of modern laws throughout all aspects of our lives may pose the greatest challenge to the widespread adoption of Dapps.

Scalability.

Scalability is an area where currently Dapps are struggling to perform due to the stringent nature of their own features. Blockchain and other decentralised technology are by default immutable, irreversible and distributed. While this can make them a very attractive option it also comes at the cost of speed and excessive storage space (Repsys 2008). All of the extra data that must be preserved begins to complicate distribution across nodes and naturally put it at a scaling disadvantage to centralised alternatives. Furthermore, due to computing requirements being spread across nodes, network congestion can easily take place at a single slow node struggling to complete the computation that it was assigned. The most promising areas of improvement lie in off-chain scaling. This is the process of creating additional layers of transactions that can rely on the core blockchain by grouping transactions together to be processed at the same time (10Clouds 2022). However, compared to the scaling optimisations that centralised applications have made and the inherent weight of its features, it is unlikely that Dapps will outperform centralised apps in terms of scaling anytime soon.

Dependability.

Dependability has always been an area where Dapps are superior to centralised apps. In Dapps information and computational responsibility are distributed across the network. A single outage in the network is unlikely to have any noticeable effect, and due to the physical distance between nodes, it is unlikely that widespread outages will take place. In contrast, dependability is a very real concern for the designers of centralised apps. Information is stored in central locations and very often there are no alternative sources for data. This means that in the case of an outage the service is entirely unavailable. Furthermore, in large corporations, these central locations come in the form of large data centres. A single outage of one of these data centres will result in large sections of the service going down. Many decentralised apps can tout an almost 100% uptime of their service due to the resiliency offered by distribution. Conversely, the unavailability of very popular services such as Whatsapp and AWS is becoming a more common sight (Tech monitor 2022).

Security & Privacy

Security and Privacy is another domain in which Dapps enjoy an inherent advantage. Dapps make use of cryptographic techniques to hide the identity of users while maintaining the ability to verify a user's unique identity. A common way that this is achieved is through the use of public/private key pairs. For example, a user called Bob could encrypt a message using their private key and distribute the message across the network. Other users can decrypt that message using Bob's public key which is available to all users on a public ledger. This verifies that the message came from Bob while maintaining Bob's true identity as a secret. Furthermore, a response can be sent to Bob by encrypting a message with Bob's public key. Bob will be the only person able to read it as it can only be decrypted using Bob's private key. This method can be used for all kinds of interactions across the app where both identity verification and privacy need to be considered. This enables users to maintain full control over their data. The nature of Dapps combined with public/private keys ensures that no one can pose as the user or corrupt the network. Similarly, no compromising information can be found in the public ledger, as only hashed pointers are stored in it (Guy et al 2015).

This policy makes a very strong case for superiority over centralised apps, where user data is handed over to a middleman to store, protect and distribute. In recent years it has been found that companies like Meta and Google track, curate and sell this data to advertisers as a key source of revenue. Furthermore, it has been proven that the security methods of very large and established centralised apps are not infallible. In September 2019, the phone numbers and email addresses of up to 533m users appeared on an online hacking forum as a result of a data breach at Meta. The company has since been fined €265 Million by the Irish Data Protection Commission (BBC 2022). Needless to say, such a data leak would not have been possible in a decentralised system.

Economic cost

The greatest economic advantage that Dapps provide is the removal of a middleman for transactions in financial systems. In centralised systems, the transfer of assets and funds comes with a transaction fee to be paid to the network that facilitated the trade. Trusted companies such as MasterCard charge a transaction fee of 1.29% - 3.29% (Mastercard 2020). Dapps circumvent this through the use of smart contracts, which facilitate safe and secure transactions without the need for a middleman. They are simply a program stored on the blockchain, attached to a user's account. Typically, they are used to automate the execution of a transaction so that all parties can be certain of the outcome (IBM 2022). Smart contracts in reputable systems are subject to standards such as ERC-20 which define how functions in smart contracts are to operate. Furthermore, a very small transaction cost is tied to invoking these functions which exist to deter bad actors. This is called gas and is most notably used in the Ethereum system. In conclusion, assuming all correct implementations, Dapps can provide a cheaper alternative to transactional systems by eliminating middlemen without compromising security. Fig 1. Found in the appendix, gives an example of a smart contract written in Solidity that could handle a safe remote Purchase.

Regulation

In the above sections I have illustrated that Dapps have several clear and significant technical and societal advantages over their centralised counterparts. Despite this, a desire for regulation may prevent Dapps from being the key technology behind any mainstream applications. Dapps are notoriously difficult to regulate and control by design. There is no middleman to monitor and control the interaction between users nor does the system have any notion of the rules that govern the outside world. Furthermore, the incredible resiliency of Dapps makes it difficult for outside organisations to shut them down. It is possible to disrupt them by targeting key anchor points but the underlying network will persist. It is difficult to find and prosecute lawbreakers due to the anonymity provided by the system and it is difficult to accrue evidence due to messages being securely encrypted. Overall these features mean that regulators will have a hard time enforcing their rules. This challenge is further illustrated by technologies such as cryptocurrency being key enablers of illegal activity in recent years (Europol 2021).

Despite these difficulties, we have seen many calls for regulation over cryptocurrency and other blockchain technology. Most recently the collapse of the cryptocurrency exchange FTX prompted a fresh set of calls for regulation from both European and American bodies. Last week the European Central Bank president Christine Lagarde said that regulation and supervision of crypto was an absolute necessity, while the United States House Financial Services Committee Chair Maxine Waters announced that lawmakers will explore the collapse of FTX in a Dec. 13 inquiry, with the goal of assessing what kind of regulation would have been most effective in averting that disaster (Coin Telegraph 2022). While regulation may be effective at streamlining the actions of crypto exchanges, which really act as middlemen between users and the blockchain itself, it is unlikely that effective regulation can be imposed on Dapps themselves. Any attempt to do so would undermine the key features that identify them and render them far closer to the kind of apps that they are designed to be an alternative to. I believe that these recent calls for regulation, are regulatory bodies trying to control the uncontrollable and in reality, modern societies simply don’t have the stomach for decentralisation.

Conclusion

In Conclusion, Dapps provide key improvements over centralised apps in the areas of dependability, economic cost, and security and privacy. They are currently inferior in terms of scalability but research and development in off-chain scaling may reduce this technical gap in the future. Despite, these advantages I think that the difficulty to regulate and control Dapps by nature will prevent them from being widely adopted into widespread technology. For this reason, Dapps will have a place among some internet communities that can appreciate their benefits, but they are not the future of the internet.

Bibliography

Andoni, M., Robu, V., Flynn, D., Abram, S., Geach, D., Jenkins, D., McCallum, P. and Peacock, A. (2019). Blockchain technology in the energy sector: A systematic review of challenges and opportunities. *Renewable and Sustainable Energy Reviews*, [online] 100(1), pp.143–174. doi:10.1016/j.rser.2018.10.014.

Repsys, V. (2019). *How to build a decentralized application that scales? Try less blockchain.* [online] HackerNoon.com. Available at: https://medium.com/hackernoon/how-to-build-a-decentralized-application-that-scales-try-less-blockchain-b3e61b1d7bd6 [Accessed 4 Dec. 2022].

10Clouds. (n.d.). *DApps Can Be Fast If You Know How to Use Scaling*. [online] Available at: https://10clouds.com/blog/defi/the-speed-of-your-dapp-use-dapp-scaling/ [Accessed 4 Dec. 2022].

Glover, C. (2022). *WhatsApp outage leads analysts to ask: Should we rely globally on a centralised application?* [online] Tech Monitor. Available at: https://techmonitor.ai/applications/whatsapp-decentralised-application [Accessed 4 Dec. 2022].

Zyskind, G. and Nathan, O., 2015, May. Decentralizing privacy: Using blockchain to protect personal data. In *2015 IEEE Security and Privacy Workshops* (pp. 180-184). IEEE.

Facebook: Meta fined €265m by Irish Data Protection Commission. (2022). *BBC News*. [online] 28 Nov. Available at: https://www.bbc.com/news/world-europe-63786893 [Accessed 4 Dec. 2022].

Visa USA Interchange Reimbursement Fees Visa Supplemental Requirements Visa Public. (2020). [online] Available at: https://usa.visa.com/dam/VCOM/download/merchants/visa-usa-interchange-reimbursement-fees.pdf.

www.ibm.com. (n.d.). *What are smart contracts on blockchain? | IBM*. [online] Available at: https://www.ibm.com/topics/smart-contracts#:~:text=Smart%20contracts%20are%20simply%20programs.

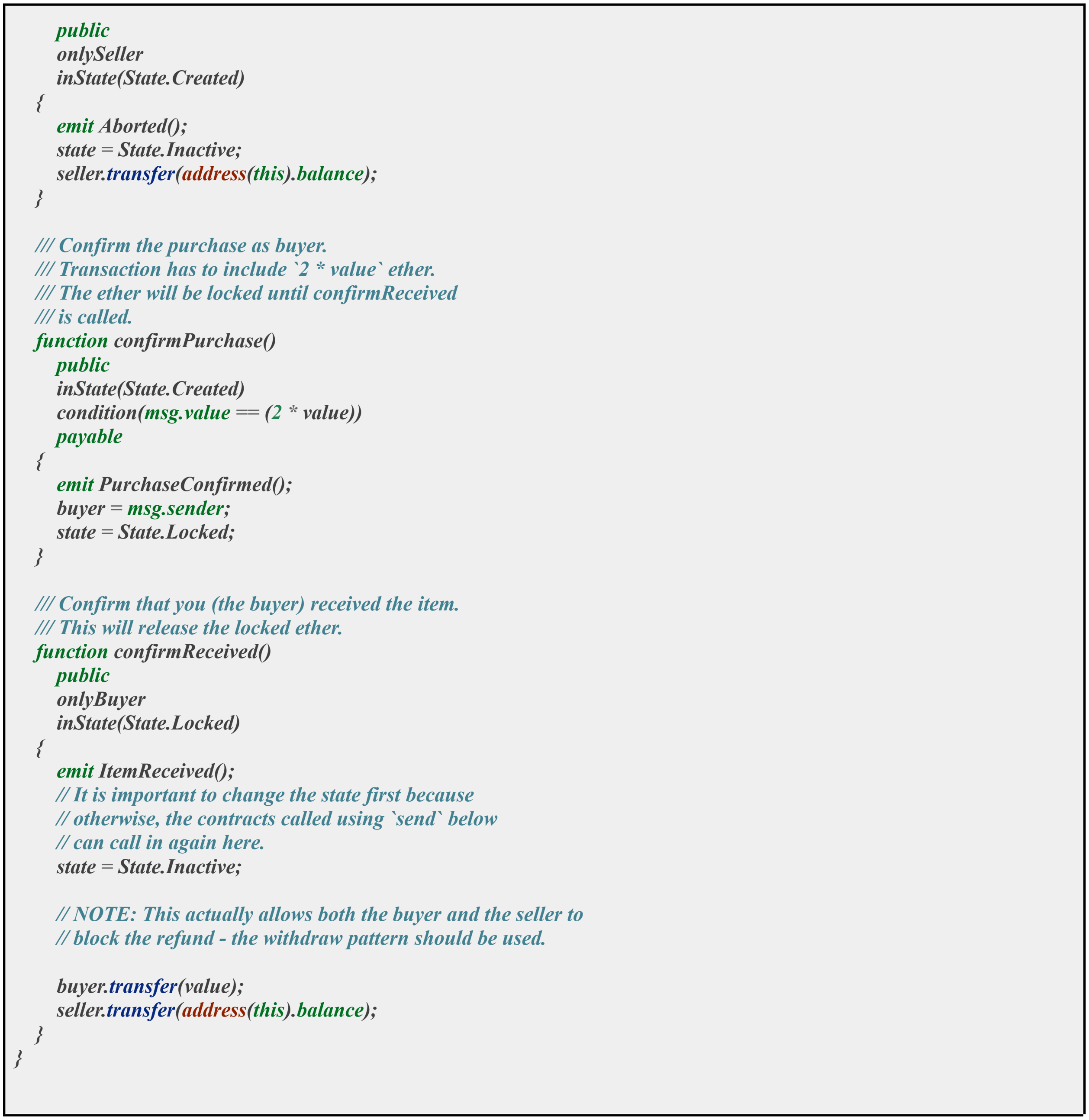
docs.soliditylang.org. (n.d.). *Solidity by Example — Solidity 0.5.3 documentation*. [online] Available at: https://docs.soliditylang.org/en/v0.5.3/solidity-by-example.html#blind-auction [Accessed 4 Dec. 2022].

KEY FINDINGS 2 EUROPOL SPOTLIGHT -CRYPTOCURRENCIES: TRACING THE EVOLUTION OF CRIMINAL FINANCES. (n.d.). [online] Available at: https://www.europol.europa.eu/cms/sites/default/files/documents/Europol%20Spotlight%20-%20Cryptocurrencies%20-%20Tracing%20the%20evolution%20of%20criminal%20finances.pdf

Cointelegraph. (2022). Calls for regulation get louder as FTX contagion continues to spread. [online] Available at: https://cointelegraph.com/news/calls-for-regulation-get-louder-as-ftx-contagion-continues-to-spread [Accessed 4 Dec. 2022].

Appendix

```
pragma solidity >=0.4.22 <0.6.0;

contract Purchase {
    uint public value;
    address payable public seller;
    address payable public buyer;
    enum State { Created, Locked, Inactive }
    State public state;

    // Ensure that `msg.value` is an even number.
    // Division will truncate if it is an odd number.
    // Check via multiplication that it wasn't an odd number.
    constructor() public payable {
        seller = msg.sender;
        value = msg.value / 2;
        require((2 * value) == msg.value, "Value has to be even.");
    }

    modifier condition(bool _condition) {
        require(_condition);
        _;
    }

    modifier onlyBuyer() {
        require(
            msg.sender == buyer,
            "Only buyer can call this."
        );
        _;
    }

    modifier onlySeller() {
        require(
            msg.sender == seller,
            "Only seller can call this."
        );
        _;
    }

    modifier inState(State _state) {
        require(
            state == _state,
            "Invalid state."
        );
        _;
    }

    event Aborted();
    event PurchaseConfirmed();
    event ItemReceived();

    /// Abort the purchase and reclaim the ether.
    /// Can only be called by the seller before
    /// the contract is locked.
    function abort()
```



```
        public
        onlySeller
        inState(State.Created)
    {
        emit Aborted();
        state = State.Inactive;
        seller.transfer(address(this).balance);
    }

    /// Confirm the purchase as buyer.
    /// Transaction has to include `2 * value` ether.
    /// The ether will be locked until confirmReceived
    /// is called.
    function confirmPurchase()
        public
        inState(State.Created)
        condition(msg.value == (2 * value))
        payable
    {
        emit PurchaseConfirmed();
        buyer = msg.sender;
        state = State.Locked;
    }

    /// Confirm that you (the buyer) received the item.
    /// This will release the locked ether.
    function confirmReceived()
        public
        onlyBuyer
        inState(State.Locked)
    {
        emit ItemReceived();
        // It is important to change the state first because
        // otherwise, the contracts called using `send` below
        // can call in again here.
        state = State.Inactive;

        // NOTE: This actually allows both the buyer and the seller to
        // block the refund - the withdraw pattern should be used.

        buyer.transfer(value);
        seller.transfer(address(this).balance);
    }
}
```

Fig 1. Smart Contract (docs.soliditylang.org 2022)