WHITE PAPER

Blockchain
Transparent Supply Chain

Secure Your Business: End-to-End Supply Chain Traceability

Intel® Transparent Supply Chain, built on Ethereum blockchain, creates a step-by-step immutable transaction record.

Authors

Eduardo Cabre
Product Development Engineering Manager

Tom Dodson
Supply Chain Security Architect

Table of Contents

Introduction

The Intel® Transparent Supply Chain (Intel® TSC) enables platform- and component-level traceability for Intel® vPro™ systems. Intel TSC also meets the U.S. government's Defense Federal Acquisitions Regulations Supplement (DFARS) requirements providing traceability on the sourcing of electronic parts.

This objective of this DFARS rule is to avoid acquisition of counterfeit electronic parts by requiring U.S. government Department of Defense (DoD) contractors and subcontractors to buy electronic parts from trusted suppliers, in accordance with section 818(c)(3) of the National Defense Authorization Act (NDAA) for FY 2012.

Intel has been operating a centralized signing service (TSC SS) for the Transparent Supply Chain program since 2017. This service allows original design manufacturers (ODM) and original equipment manufacturers (OEM) participating in the program to submit platform-level information to be validated and signed. This service relies on a centralized Certificate Authority (CA), hosted by Intel in a secure facility. The CA is responsible for signing the data and providing the trust certification of this data to be consumed by platform owners.

The Intel Transparent Supply Chain blockchain proof of concept (POC) was developed as an alternative to the current centralized trust model of trusted suppliers and trusted manufacturers. This POC consisted of implementing a transactional model of the supply chain using blockchain transactions to record each step of the product's manufacturing and distribution process. Supply chain participants will have the ability to record platform-level information in the immutable blockchain with the use of a distributed application (DAPP) based on Ethereum smart contracts.

POC Goals and Objectives

The goal of this POC was to demonstrate the viability of implementing a blockchain-centric Transparent Supply Chain capability. This effort has the following objectives:

1. Understand the feasibility of developing a Transparent Supply Chain capability with available blockchain technology.
2. Develop better understanding of blockchain development tools and infrastructure.
3. Evaluate cost, performance, and security tradeoffs between public blockchain and private blockchain options.
4. Understand the overall hosting and infrastructure requirements for a public blockchain versus private blockchain solution.

5. Gather technical expertise developing blockchain-based distributed applications (DAPPs).
6. Solve anonymity and data confidentiality issues when designing a secure blockchain solution.
8. Design for scalability and maintainability.
9. Determine if the additional cost and complexity associated with the blockchain implementation is worth the benefits afforded by the technology.

Assumptions

In order to expedite the development of this POC, we made these assumptions:

- Ethereum was selected as the blockchain technology, given its maturity and large development tool availability.
- For the purposes of cost analysis, current market prices of gas and ether were used.
- No user management or authentication was implemented in the DAPP.
- A single web application was created to enable OEM, ODM, distributor, reseller, and platform owner use cases.
- A popular cloud storage solution was used to store platform data files. When productizing this solution, it may be advantageous to instead use distributed file storage mechanisms such as the InterPlanetary File System (IPFS).
- A small private Ethereum network was created to develop and deploy the POC.

Ecosystem Requirements

The infrastructure to support this solution included:

1. A network of dedicated servers running Ethereum nodes to support the blockchain application.
2. Each node hosted a Go Ethereum (Geth) node and executed the Geth node in mining mode.
3. Local Ethereum accounts were created on each node.
   a. In order for the account to be able to interface with the node, the account required ether to be granted directly.
   b. Ether was granted to each user account through a funding account created during genesis for that purpose.
4. The Ethereum blockchain genesis node was constructed and deployed using the Main node.
   a. The Main node was the server used by the application administrator to deploy and manage the application.
   b. The genesis block configuration is defined later in this document.
5. Ethereum smart contracts were deployed through the Main node.
6. User registration was performed by the application administrator through the Main node. For simplicity, the DAPP web application was also granted rights to dynamically register users in the blockchain. When productized, only the DAPP owner will have rights to register new users.

System Architecture

The TSC DAPP was built on top of the Ethereum blockchain. The application consisted of a series of Ethereum smart contracts, which provided the ability to store and retrieve records from the blockchain ledger. Unlike other blockchain technologies, Ethereum is based on smart contracts which are small applications running on the Ethereum Virtual Machine.

The Ethereum blockchain contains both smart contract code and the smart contract state; this guarantees that both the application code and the application data are immutable. This property allows developers to create blockchain applications that are fully distributed and trusted, based on the security properties of the blockchain. Ethereum provides immutability of data (platform records are guaranteed not to change over time), and immutability of behavior (business rules do not change over time).

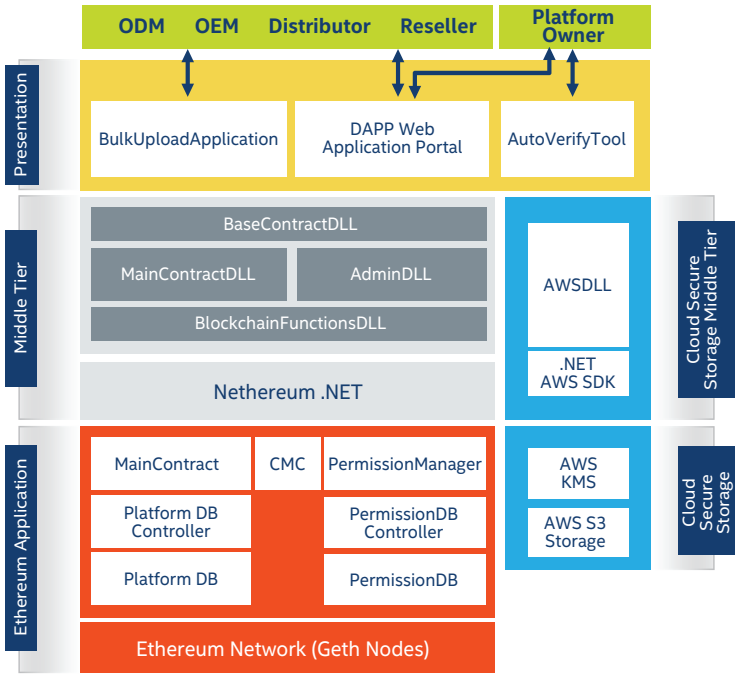

Figure 1. DAPP Architecture Tier Diagram

Figure 1 is an architectural representation of the TSC DAPP, which allows participants to create and track platform data and ownership through the supply chain. The TSC DAPP consists of these main components:

1. Platform data files: All platform configuration information is recorded in a set of XML-encoded files. These files contain platform information such as component lists, platform serial number, BIOS, and TPM PCRs.

a. As Built data file (ABD): An XML platform-specific file that contains a platform's As Built information. This file is generated by the ODM at time of manufacture.

b. Platform Certificate data file (PCD): An XML platform-specific file that contains all the parameters required to construct a platform certificate based on the Trusted Computing Group (TCG) specification.

c. Direct Platform data file (DPD): An XML platform-specific file that contains Trusted Platform Module (TPM) platform configuration register (PCR) values and SMBIOS configuration data. It is generated by the ODM at first boot.

d. Private data file (PD): A file containing secret platform data such as credentials and tokens.

e. Statement of Conformance (SOC): A document certifying the platform adherence to the Transparent Supply Chain compliance requirements.

2. The Ethereum network blockchain: Deployed as a private blockchain, where the application owner can control network connectivity to the nodes.

3. Ethereum account roles. Seven roles were defined to support the DAPP:

a. DAPP owner: This is the administration account. It is used to deploy the smart contracts, and has the rights to register contracts and assign security roles to users.

b. ODM: The original device manufacturer account which is granted the ODM security role. This account can register a new platform in the DAPP and upload platform-related files.

c. OEM: The original equipment manufacturer account which is granted the OEM security role. This account can download and upload platform files, and verify the platform.

d. Distributor: The platform distributor account which is granted the Distributor security role. This account can download platform files.

e. Reseller: The platform reseller account which is granted the Reseller security role. This account can download platform files, and upload DPD files only.

f. Platform owner: The party that owns the platform; it is granted rights to download platform files from the DAPP.

g. Funding account (Not in diagram): This is a virtual account which is initialized with a large amount of ether. This account is used to fund other accounts that are programmatically created by the DAPP.

4. Cloud secure storage: The cloud storage solution deployed to store the platform data files. It provides off-chain encrypted storage support.

5. The Ethereum application: The smart contracts that define the application logic and store the platform data. Contracts are written using the Solidity language.

6. Middle tier libraries: A set of libraries that provide connectivity between the presentation tier (web application, console applications, APIs, etc.) and the Ethereum application. Two versions of the library were developed, a .NET Core library and a .NET standard library.

7. Cloud secure storage middle tier: Libraries developed to ease interfacing with the cloud secure storage. They implement access to cloud storage, encryption, and decryption using a key management service (KMS). This tier also includes the .NET Storage SDK used to connect with cloud storage.

8. Presentation tier: This tier is comprised of the DAPP web application, the bulk upload application, the auto verify tool, and the administration console.

a. The web application allows users to upload platform-related information to the blockchain. A bulk upload application was created to perform bulk load of platform data.

b. The web application allows the platform owner to download the platform information once the platform is in their possession.

c. The TSC auto verify tool has been integrated to interface with the blockchain application to download and upload platform related data.

d. The administration console is used by the DAPP owner to deploy and register contracts and users in the Ethereum application.

System Flows

Creating a Platform

Platforms are created in the DAPP by the ODM upon completion of the manufacturing process. The ODM logs into the web application and submits the platform's unique ID and serial number to the blockchain, as depicted by the sequence diagram in Figure 2.

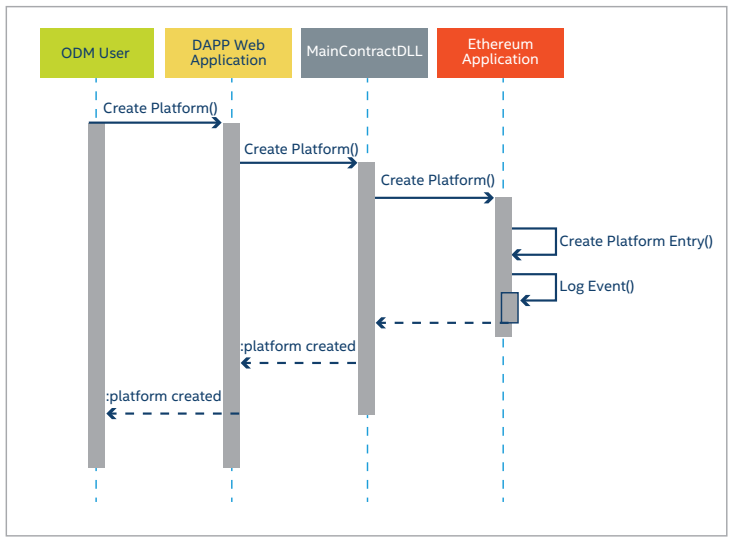

Figure 2. Platform Creation Sequence Diagram

The platform creation process is required before any platform-specific data can be added to the DAPP. This process creates the initial platform record on the blockchain and allocates the blockchain storage space required for future transactions. Each blockchain operation is logged in the blockchain for traceability. The log includes the initiating user account, the transaction type, and a timestamp. These logs are stored in the blockchain and can be used to verify the transactional history of a platform. They cannot be altered.

Uploading Platform Data Files

The ODM, OEM, reseller, and platform owners may upload platform data files through the web application. The web application calculates a hash of the contents of the file and replaces the file name with the hash. The original file name is added to the metadata payload, so the file can be renamed back. The file is then sent to the storage DLL for upload. The storage DLL library downloads the encryption key from KMS and encrypts the content of the file prior to uploading it. Once the file is successfully uploaded to the cloud, the web application submits the file hash to the main contract DLL to be registered in the blockchain. Upon successful completion of the hash registration, the process ends.

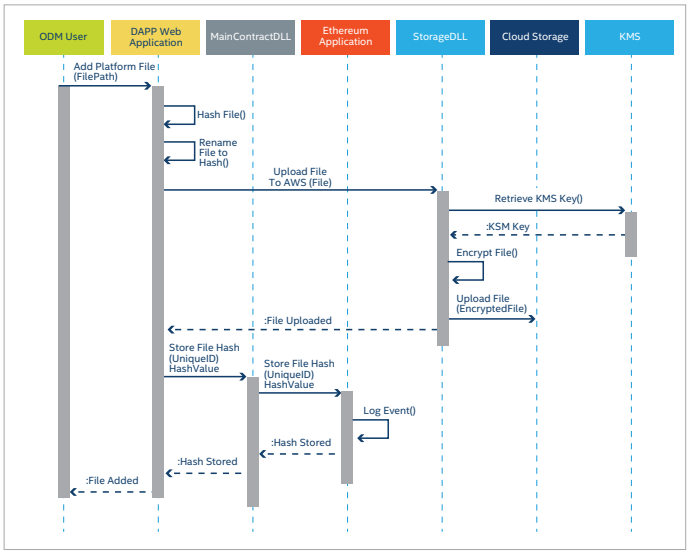

Figure 3. Platform File Upload Sequence Diagram

Download Platform Data Files

The ODM, OEM, distributor, reseller, or platform owner downloads the platform data files from the DAPP through the web application by providing the platform unique ID information: platform OEM, model, and serial number.

Figure 4 depicts a detailed sequence diagram. The web application retrieves the file hash pointer from the blockchain through the main contract DLL. Subsequently, the web application calls the storage DLL to retrieve the file from cloud storage by providing the hash pointer. Once retrieved, the file is decrypted with the KMS key and renamed back to its original name. The contents of the file are hashed and compared to the hash pointer. If the hashes match, the file is sent back to the web application and made available to the user. A failure to match the file hash with the hash pointer indicates the file integrity has been compromised. At that point, the file is considered invalid and an error message is displayed to the user.

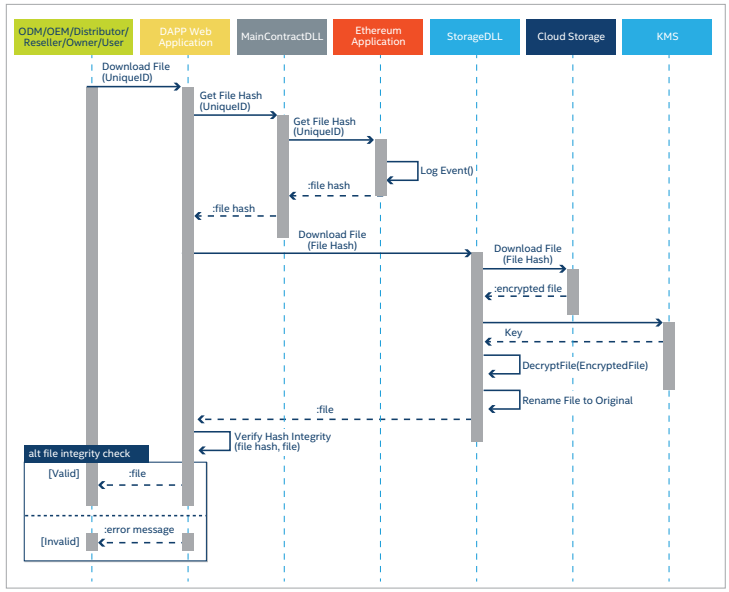

Figure 4. Platform File Upload Sequence Diagram

Transfer Platform Ownership

Figure 5 illustrates a platform owner transferring ownership of a platform to another Ethereum account by submitting an ownership transfer request through the DAPP web application. The request is forwarded to the main contract DLL and Ethereum application containing the unique ID of the platform to transfer, the Ethereum account of the requesting party (RequestAcct), and the Ethereum address of the new platform owner (NewOwnerAcct). The Ethereum application confirms that the RequestAcct is the current owner of the platform; if so, it changes the platform ownership record to the NewOwnerAcct.

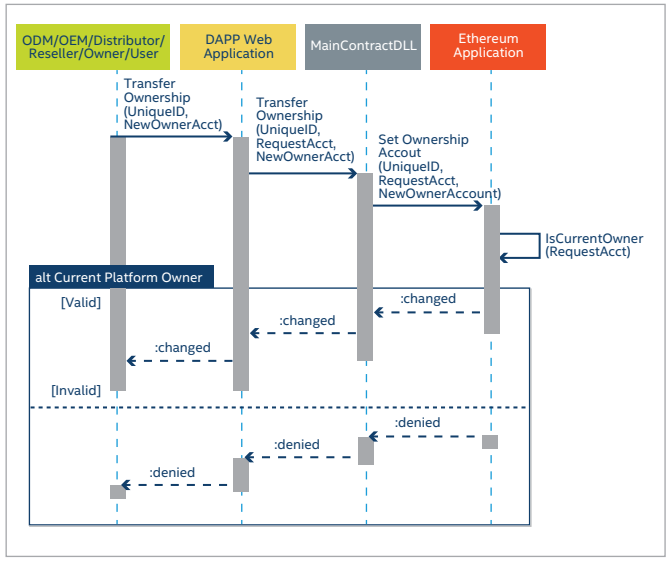

Figure 5. Transfer Platform Ownership

Cost Analysis

The cost of implementing the TSC blockchain capability varies significantly depending on the network used. In the public network, costs will be driven by transaction volume. Transaction prices are determined by the market price of Ethereum and gas price. Gas is the unit of processing; each transaction requires a certain amount of gas to execute. Every gas unit has a price in ether that is driven by market demand. And as expected, ether has a corresponding market-driven price in dollars.

Transaction -> gas units (deterministic) -> gas price in ether (variable) -> ether price in USD (variable)

There will also be costs associated with infrastructure to host the DAPP web application and middle tier libraries, and the cost of cloud storage. Using the public network would result in large infrastructure savings and increased security, since thousands of nodes are already available to process the transactions. Nonetheless, these savings will be offset by transaction costs. On the other hand, deploying a private Ethereum network will require large infrastructure investment in the form of Ethereum nodes.

Performance Analysis

Public Ethereum Network

Blockchain performance depends on a number of factors. Some are external (not controlled by the user), and others are internal (controlled by the user).

1. External factors
    a. Network demand: This is reflected in the number of transactions waiting to be mined
    b. Gas price: What others are willing to pay for gas
2. Internal factors
    a. Gas price: The amount of gas a user is willing to pay for a transaction

Block Mining Time

The Ethereum network is configured to maintain a block mining time of between 10 to 19 seconds. This is achieved through adjusting the difficulty target to stay within the desired block mining range. This results in an average block mining time of 14.5 seconds per block. A block mining time of 14.5 seconds results in a capacity of 2,174,896 blocks per year.

Note that the block mining times are not guaranteed. Changes to the algorithm can significantly affect the overall mining time and therefore the performance of the DAPP. In mid-2017, the Ethereum foundation introduced a change in the mining difficulty algorithm in preparation to the migration from the Proof of Work (PoW) consensus algorithm to the Proof of Stake (PoS) consensus algorithm. This change doubled the mining time in late 2017. This algorithm was reverted once the foundation decided to postpone the PoS transition.

Confirmation Time

Once a block has been mined, it will take a certain amount of time for the network to confirm that the block is valid. This is the amount of time that it takes for the fork that contains the block to grow to a certain length. The DAPP provides a mechanism for the user to determine whether its block has been confirmed. The DAPP web application displays the depth of the mined block with respect to the leaf block (last block in the chain), and it displays the number of confirmation blocks achieved.

Confirmation blocks are critical to the security of the application. Mathematical models estimate that for 17-second blockchain mining (slightly longer than Ethereum today), it requires about 10 confirmation blocks to equate to the 6 confirmation blocks typical of bitcoin. This is based on a Byzantine fault-tolerant model which assumes a certain percentage of the miners are attackers [1]. For this application, we believe that a 10-block confirmation wait is acceptable; this corresponds to 145 seconds (14.5 seconds / block x 10 blocks).

Block Gas Limit

On average, the Ethereum nodes are configured to an 8,000,000 gas limit. On average, Ethereum transactions cost 76,364 gas. For TSC, transaction gas costs are higher on average, and are dependent on the type of transaction being executed.

Theoretical Operating Capacity

In order to understand the overall transaction process time of the DAPP, we need to determine the number of transactions per day that the public network can process. The reader must be mindful that this estimate assumes that the network usage for all non-TSC transactions is static. Based on our estimates, the overall processing time for our application is as follows:

1. Block per year capacity = 2,174,896 blocks / year
2. Block gas limit = 8,000,000 gas
3. Total gas available
    a. 2,174,896 blocks / year x 8,000,000 gas / block = 17,399,168,000,000 gas / year
4. Gas consumed by other Ethereum transactions
    a. 76,364 avg. gas / transaction [3] x 203,599,190 transactions / year = 15,547,648,545,160 gas / year
5. Total gas available to Transparent Supply Chain
    a. 17,399,168,000,000 gas / year - 15,547,648,545,160 gas / year = 1,851,519,454,840 gas / year
6. Total gas cost per platform
    a. The most common transactions performed on a per-platform level

PLATFORM	GAS UNITS	OPERATION COUNT	TOTAL GAS
Register platform	853,750	1	853,750
Add ABD hash pointer	160,000	1	160,000
Add DPD hash pointer	158,085	5	790,425
Add PCD hash pointer	157,713	1	157,713
Add PDF hash pointer	160,000	1	160,000
Add SOC hash pointer	160,000	1	160,000
Verify a platform	160,000	1	160,000
Transfer ownership	251,192	5	1,255,960
		Total	**3,697,848**

   b. Average gas / platform = 3,697,848 gas

7. Transaction / year capacity

   a. Available gas capacity = 1,851,519,454,840 gas / year

   b. Available platform capacity = 1,851,519,454,840 (gas / year) / 3,697,848 (gas / platform) = 500,701 platforms / year

Based on our assumptions, the public blockchain can support 500,701 platforms per year. This is the theoretical maximum capacity of the system, and does not take into consideration demand variations over time.

Private Ethereum Network

A key factor in determining the private blockchain mining time is the overall latency of the system. In this section, we use theoretical and empirical data to estimate the private blockchain performance tradeoffs.

Block Mining Performance

In this analysis we will assume that the block mining time performance will be similar to that of the public network. Assume a 14.5 second per block mining performance, although faster block mining times may be available due to reduced network latency.

Block Gas Limit

Since all the nodes in the private Ethereum network are under our control, we can arbitrarily increase the gas limit configuration on all the nodes. By increasing the gas limit, we increase the gas capacity of the entire network, resulting in significant gains in capacity. Increasing the gas limit increases the number of transactions included in a block, therefore improving the transactional performance. Through experimentation, we chose a gas limit of 320,000,000 gas, which resulted in 423 transactions per block.

Theoretical Operating Capacity

Utilizing the same approach as in the public network scenario, these are the operating capacity numbers for the private network.

1. Block per year capacity = 2,174,896 blocks / year
2. Block gas limit = 320,000,000 gas
3. Total gas available to Transparent Supply Chain
   a. 2,174,896 blocks / year x 320,000,000 gas / block = 695,966,720,000,000 gas / year
4. Total gas cost per platform
   a. As in the public Ethereum option, the average gas / platform = 3,697,848 gas / platform
5. Transaction / year capacity
   a. Available gas capacity = 695,966,720,000,000 gas / year
   b. Available platform capacity = 695,966,720,000,000 (gas / year) / 3,697,848 (gas / platform) = **188,208,579 platforms / year**

Based on our experiments, we estimate the private blockchain can support **188,208,579 platforms** per year, a significant improvement over the public Ethereum option.

Security Considerations of the Ethereum Blockchain

In order to understand the security properties of the Ethereum-based DAPP, we need to understand the security properties of the underlying blockchain. The security of the DAPP application is based on four main factors:

1. Security of the smart contracts: Most security vulnerabilities found in smart contract-based blockchain applications have to do with vulnerabilities introduced into the smart contract code by the developers. These risks can be mitigated by utilizing well-understood patterns, code templates, perform code reviews, penetration testing, and third-party audits.
2. Security of the blockchain: Blockchain security is obtained through the block confirmation process. This process requires participants to confirm the transaction by waiting for a certain number of blocks to be mined. This process reduces the risk of the successful double-spend or Finney attack [2]. The larger the number of confirmations, the lower the probability of a double-spending attack. The potential success of this attack increases with the amount of hashing power the attacker obtains. Author Meni Rosenfeld, in Figure 4 of his paper titled “Analysis of hashrate-based double-spending” [5], charts the probability of successful double-spend attacks as a function of the attacker’s hashrate, for different numbers of confirmations. This figure shows that for a low confirmation of only two blocks (n = 2), even with a low hashrate of 10% (1 compromised server in our private network), an attacker

can achieve a double-spend attack with around 8% probability. For a much larger confirmation (n = 10), the probability of success would be near 0. The attacker would need to obtain about 30% of the hashrate power to obtain a similar success rate. At a hashrate of 50%, the blockchain is considered compromised regardless of the number of confirmations.

3. Size of the network: One advantage of using the public Ethereum network is the large availability of nodes. As of this writing, there are 8,897 active Ethereum mining nodes in the world [4]. This provides a security advantage over the private network, which consists of a handful of nodes. This risk is reduced by controlling which entities have access to the private network. Nonetheless, an authorized participant may still successful attack the private network by deploying a node large enough to overtake the hashrate of the overall system. It is necessary to monitor network traffic to detect such an attack.

4. Security of the account private keys: Ethereum uses the ECC P-256 signing key to sign each of the transactions. This signature is used to verify the authenticity of the sending account. If an account's private key is compromised, an attacker may obtain access to a user's ether and may impersonate user transactions. Most applications rely on software security mechanisms such as Ethereum Wallets to protect the private key, but hardware security modules can be used to better secure the key in sensitive applications.

With respect to privacy and anonymity of transactions, the private Ethereum network provides additional protections. Only TSC participants are granted access to the blockchain, which reduces the number of entities with access to the transactional data. Nonetheless, participants may monitor transactions in the blockchain and be able to determine the identity of a user and associated user accounts. These attacks are much more prevalent in the public blockchain.

Conclusions

1. We have demonstrated that the current Transparent Supply Chain capability can be successfully implemented into a blockchain solution utilizing Ethereum blockchain technology. We have also demonstrated that by increasing the network's gas limit, we can significantly increase the capacity and performance of the network (transactions / minute). These options were considered in order to improve performance:

   a. Transaction batching: Implementing transaction batching by forcing the network nodes to process as many transactions as possible in the same block. As a result, the block mining time is spread over more transactions.

   b. Programmatically reduce the mining difficulty: The mining difficulty algorithm can be modified to reduce block mining times. Block mining times are driven by the ability of the network to propagate new blocks. A private network on the other hand, could be built to allow for fast blockchain synchronization, allowing the reduction of the mining difficulty.

   c. Alternative blockchain technologies: As the application user base grows, if Ethereum does not meet the capacity requirements, alternative blockchain technologies such as Hyperledger Sawtooth and Hyperledger Fabric will be considered.

2. The costs to operate the public network are very difficult to predict given the large variations on market price of ether and gas. Given that the public network costs are primarily driven by transactions, the costs of running this system will significantly increase as we scale the TSC capability (more platforms, more users, etc.). Using a private network allows us to control the generation of ether, eliminating real transaction cost.

3. Privacy in the blockchain is always a concern. Transactions on the blockchain are pseudonymous. Determining the identity of a TSC participant in the public Ethereum blockchain is fairly easy to achieve by doing an inference attack. The private blockchain reduces the overall risk by limiting blockchain data access to TSC participants only. An attacker may infer a participant's identity and associate that identity to an Ethereum account. Nonetheless, these attacks would reveal only account ownership, so that transaction creation can be tracked to a given participant. Data confidentiality is protected by cloud storage access controls and encryption.

4. With respect to designing for maintainability, there are design patterns available that allow for easy maintainability and upgrading the Ethereum application. We implemented the 5-type-model design pattern, which allows us to deploy contract changes without impacting the rest of the contracts. This pattern seems to be flexible enough to support feature growth.

5. With respect to scalability, we demonstrated that transaction batching is achievable in the private network case, which can significantly increase network capacity.
6. It is evident that a private Ethereum-based solution is much more appropriate than the public network, due to a number of important factors:
   a. Unpredictability of public network costs.
   b. Large marginal cost of the public network. As the volume of transactions increases, cost of the system increases linearly.
   c. Larger theoretical capacity of the private network.
   d. More predictable performance.
   e. Smaller attack surface, resulting in lower exposure to attacks.

Learn More

Intel® Transparent Supply Chains
https://www.intel.com/content/www/us/en/servers/transparent-supply-chain.html

References

[1]. Vitalik Buterin, "On Slow and Fast Block Times," Ethereum Blog, 2015 https://blog.ethereum.org/2015/09/14/on-slow-and-fast-block-times/

[2]. Finney attack discussion, https://bitcointalk.org/index.php?topic=3441.msg48384#msg48384

[3]. Eric Conner, "Ethereum network throughput under Shasper," Medium, 2018 https://medium.com/@eric.conner/_ethereum-network-throughput-under-shasper-390e219ec2b5

[4]. The Ethereum Nodes Explorer (Ethernodes). https://www.ethernodes.org/network/1

[5]. Meni Rosenfeld, "Analysis of hashrate-based double-spending," 2012 https://www.bitcoil.co.il/Doublespend.pdf