Using Blockchain Technology to Secure the Internet of Things

Presented by the Blockchain/ Distributed Ledger Working Group

cloud security alliance®

ABOUT CSA

The Cloud Security Alliance is a not-for-profit organization with a mission to promote the use of best practices for providing security assurance within Cloud Computing, and to provide education on the uses of Cloud Computing to help secure all other forms of computing. The Cloud Security Alliance is led by a broad coalition of industry practitioners, corporations, associations and other key stakeholders. For further information, visit us at www.cloudsecurityalliance.org and follow us on Twitter @cloudsa.

TABLE OF CONTENTS

ACKNOWLEDGMENTS

Initiative Lead:
Sabri Khemissa

Key Contributors:
Alex Brown
Giuliana Carullo
Elier Cruz
Kevin Fielder
Doug Gardner
Jas Khehra
Imre Kocsis
Paul Lanois
Ashish Mehta
Matt Murphy
Todd Nelson
Denis Nwanshi
Luc Poulin
Michael Roza
Brian Russell
Srinivas Tatipamula
Udo Gustavo von Blücher

CSA Staff:
Hillary Baron
Kendall Scoboria
John Yeoh

INTRODUCTION

In the last four years, technical experts, chief digital officers, marketing managers, journalists, bloggers and research institutions have discussed and promoted a new distributed model for secure transaction processing and storage using blockchain technology. IDC FutureScape predicted that by 2020, 20% of global trade finance will incorporate blockchain [1]. Coindesk reports that venture capitalists have invested over $1.8 billion in blockchain startups over the past few years [2]. Consortiums and alliances have sprouted up, such as the Enterprise Ethereum Alliance, which focus on identifying new uses cases for blockchain technology across sectors.

Blockchain, a public and distributed ledger of transactions grouped into blocks, promises to:

1. Increase speed, efficiency, and security of ownership transfer of digital assets
2. Eliminate need for central authorities to certify ownership and clear transactions
3. Reduce fraud and corruption by providing a transparent and publicly auditable ledger
4. Reduce administrative cost using agreements that can automatically activate, secure and certify trusted actions based on specific conditions ("smart contracts").

A key challenge associated with the adoption of blockchain is the need to identify relevant use cases that would benefit from the integration of blockchain technology. The Internet of Things (IoT) has long been associated with security weaknesses and challenges, and experts and organizations have begun exploring the use of blockchain to securing the IoT. Organizations like IOTA and the Trusted IoT Alliance have begun to focus on IoT security through the application of blockchain.

The IoT, in its own right, is transforming consumer behaviors and business processes. Distributed edge IoT devices collect and transmit data for processing. IoT systems rely upon this data to provide advanced services, automation features and tailored experiences to end users. IoT systems are dynamic and distributed. They include devices, mobile applications, gateways, cloud services, analytics and machine learning processes, networking infrastructure, web services, storage systems, fog layers and users. All of these systems write and read data that can be recorded as transactions on a ledger.

The Cloud Security Alliance IoT Working Group (IoT WG) has focused on documenting best practices for IoT security since 2014. Given the potential benefits associated with applying blockchain technology to the IoT security problem, the IoT WG has partnered with the CSA Blockchain/Distributed Ledger Technology Working Group to research and document some of the ways that blockchain can begin helping to secure IoT systems. As such, this paper addresses two technologies at different maturity levels:

- **Blockchain:** A technology enabler that has driven radical change and disruption throughout the digital economy through its support of rapidly evolving cryptocurrencies such as BitCoin, Ethereum, Litecoin and Dash. Blockchain's success as a foundation for cryptocurrencies has spawned new research within the industry aimed at securing systems and technologies using the distributed ledger technology. In 2017, many business initiatives focused on creating limited prototypes and proofs-of-concepts that serve mostly to master the intricacies of this complex technology.
- **The Internet of Things:** A fast-maturing set of technologies that support the transformation of business and mission processes. The IoT has reached varying levels of maturity across sectors such as consumer, transportation, energy, healthcare, manufacturing,retail and financial. The IoT is the inter-networking of physical devices such as connected vehicles , smart buildings, industrial control systems, drone and robotics systems and other items embedded with electronics, software, sensors, actuators and network connectivity that enable these objects to exchange data.

This paper describes a high-level overview of blockchain technology and outlines a set of architectural patterns that enable blockchain to be used as a technology to secure IoT capabilities. Specific use-case examples of blockchain for IoT security are also explored, although technical implementation of those use cases will vary across companies.

A blockchain service, or simply a "blockchain," is a transaction repository where transaction are grouped into blocks. "Every block contains a hash of the previous block. This has the effect of creating a chain of blocks from the genesis block to the current block" [3]. The contents of each block is digitally signed to ensure data integrity of recorded transactions.

A blockchain service includes three main components:

(1) A network of autonomous nodes

Independent nodes autonomously generate and register legitimate transactions into the distributed ledger. Neither a central authority nor a trusted third party is necessary to validate transactions. All nodes of the blockchain service (also called the blockchain platform) collaborate to maintain a consistency of the ledger.

Each node runs a programmed mechanism, called a consensus. The consensus is the process by which nodes agree on how to update the blockchain as a result of a set of transactions. Achieving consensus ensures the majority of nodes in the network have validated the same set of transactions.

The goal of distributed consensus is to keep the ledgers of a sufficient majority of the system peers correct and up to date (at a roughly granular time scale). Consensus mechanisms guard against malicious peers that can corrupt the integrity of the ledger by (a) retroactively modifying transactions; (b) performing semantically unpermitted transactions (e.g. "double-spending" and transferring not-owned assets in a cryptocurrency setting); or (c) blocking the acceptance and booking of correct transaction requests.

The consensus approach chosen during the development of the blockchain service guards against specific attacks. These attack mitigations are not purely technical in nature. With Bitcoin's "Proof of Work," for example, there is an economic disincentive to gaining control of 51 percent of the hashpower within the network. Gaining 51 percent of the mining hashrate would potentially allow an attacker to double-spend coins or alter a recent history of transactions. Also, gaining 51 percent of the hashrate and propagating malicious transactions would rapidly destroy confidence in the cryptocurrency and significantly decrease the value of the malicious party's stake. In addition, that malicious party could simply employ their hashpower toward the process of mining to generate gains for themselves.

In a permissioned (closed) system, economic disincentives may not be present. Permissioned systems also often employ reduced mining difficulty which allows for faster transactions within the network. These permissioned systems must be outfitted with traditional cybersecurity controls that include resilience safeguards, hardware-based wallets, access controls that restrict access to network miners, identity management and strong audit capabilities to enable potential regulatory involvement, litigation, and criminal investigation of misbehavior within the system.

Three predominant mechanisms provide consensus in a blockchain:

- ***Byzantine Fault Tolerance (BFT) Algorithms*** are designed to avoid attacks and software errors that cause faulty nodes to exhibit arbitrary behavior (Byzantine faults). BFT [4] provides consensus despite participation of maliciously misbehaving (Byzantine) nodes. A drawback of this approach, however, is the scalability limit in terms of number of nodes that form the blockchain network [5]. Alternative approaches to BFT have been proposed, including Practical Byzantine Fault Tolerance (PBFT) [5]. Examples of blockchain implementations currently exploiting PBFT are Linux Foundation Hyperledger fabric (0.6) and Ripple.
- ***Proof-of-Work (POW)***, used by Bitcoin and Ethereum, is the widely known mechanism for establishing consensus. In

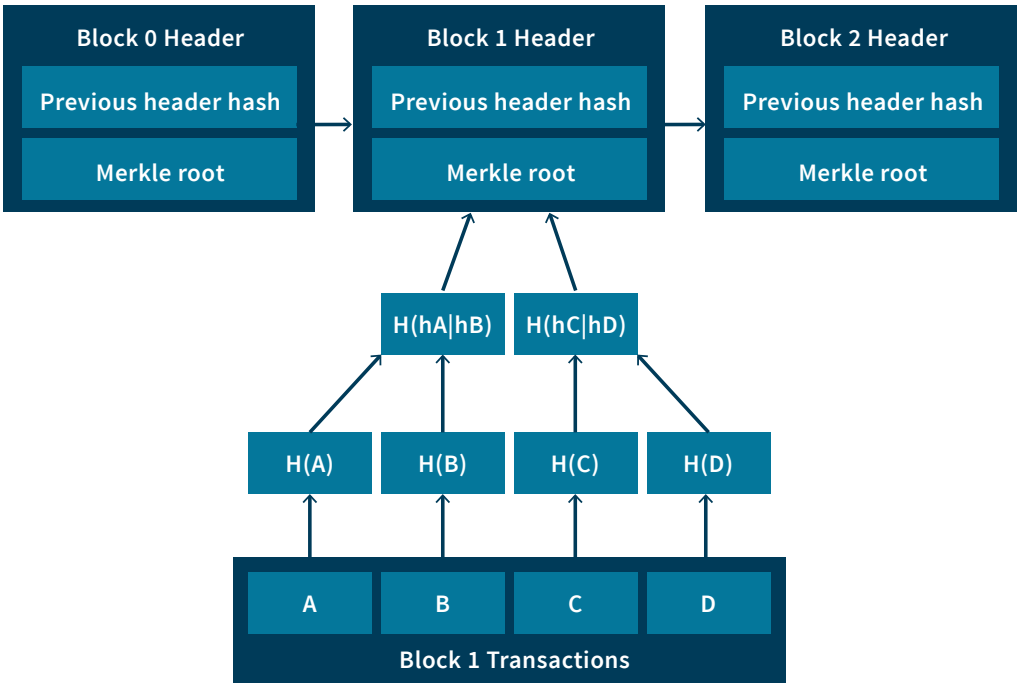
POW, a single node can provide its conclusions to others nodes, which can be in turn validated by the other nodes in the network. A node submitting a generated block, in order to have the consensus reached, must also provide proof of the work it performed, which is a computationally difficult task (a "cryptographically hard puzzle" based on hash functions). POW provides great network stability [6] . However, POW is particularly costly because of the computational resources expended. "Miners" are incentivized to participate to earn a cryptomonetary reward, which is granted in return for a successful block generation.

- ***Proof-of-Stake (POS)*** is similar to POW: nodes are rewarded upon generating a block . However, only a few nodes can participate in this phase [7]. Indeed, the next generator node is picked up deterministically based on the accumulated wealth (i.e., the "Stake"). The mining process for a blockchain based on POS is usually referred to as "forgery" or "minting." The technology that launched PoS was PeerCoin.

(2) A ledger of transactions

A database is composed of blocks (thus "blockchain"). Each block contains a list of valid transactions, a timestamp, and information linking the current block to a previous block. The chaining through the link of each block to the preceding block creates the ledger.

The heart of the ledger is the cryptographic hash, a mathematical algorithm that maps data with variable size to a fixed size string . All transactions -- A, B, C, D - are hashed - H(A), H(B), H(C), H(D) -- then aggregated into successive hashes -- H(hA|hB), H(hC|hD) - to constitute a Merkle tree. The top hash, or the Merkle tree root, is integrated into the block header.



Merkle tree connecting block transactions to block header Merkle root

(3) A distributed database

A ledger is built over time as new transactions are added, and it is available and replicated across nodes in the system (thus "distributed ledger"). Every node on the network has its own copy of the database and can access the history of any

transaction.

The blockchain size of a particular cryptocurrency will drive requirements for storage capacity within IoT and other devices that host the ledger. The table below provides the blockchain size for popular cryptocurrencies as of August 14, 2017 [30].

Cryptocurrency	Blockchain Size (snapshot 8/14/17)
Bitcoin	151.74 GB
Ethereum	98.94 GB
Ethereum Classic	20.12 GB
Litecoin	8.62 GB
Dash	3.69 GB

TRANSACTION PROPAGATION AND BLOCKCHAIN CONSTRUCTION

Below is a generic processing flow for blockchain transactions. When a transaction is committed into a node, a blockchain service typically runs as follows [6]:

1. New transactions are broadcast to all nodes.
2. Each node collects new transactions into a block.
3. Each node works on a consensus algorithm for its block (generally, this task is expensive in terms of node processing and electrical consumption).
4. When a node completes consensus algorithm processing, it broadcasts the block and the processing results to all nodes, then receives compensation for this work. (In the case of Bitcoin, the compensation is a transaction fee processed by and received by the Bitcoin miner.)
5. Nodes accept the block only if all transactions in it are valid.
6. Nodes express their acceptance of the block by working on creating the next block in the chain, using the hash of the accepted block as the previous hash.

This technology is not new: it involves digital signatures, cryptographic hashing algorithms, peer-to-peer connectivity, distributed databases, and more. Blockchain technology, which effectively combines these discrete technologies, is necessary in the current landscape because increased computing capabilities and Internet speed enable distributed computing.

SMART CONTRACTS

Smart contracts are self-executing code residing on a ledger. Using smart contracts, two parties conduct a transaction. For example, one party can render a service while the other party provides payment for that service. Smart contracts enforce the rules of the transaction and can also enforce penalties associated with non-compliance.

In the context of the IoT, devices can be preconfigured to interact with smart contracts based on the contract addresses on the blockchain. These devices can then enter into transactions between each other. The smart contract monitors the flow of the transaction and validates that rules have been followed prior to releasing funds or allowing an action.

Implementers of IoT systems that make use of smart contracts must consider potential misuse cases and install rules within the contracts themselves. For example, a smart contract developer might enforce escrow requirements that hold funds until verification of the smart contract terms of completion. Other security considerations when working with smart contracts include the need to avoid race conditions whereby the contract may be executed again prior to the first contract transaction being completed, validating that the sender and receiver of the contract are not using the same address and ensuring that only authorized devices use the smart contracts. You can learn more about smart contract security at https://consensys.github.io/smart-contract-best-practices/.

OFF-CHAIN STORAGE SOLUTIONS

Solution developers charged with implementing blockchain technology should realize that there are no confidentiality protections associated with using public blockchain networks. Even private/permissioned networks lack sufficient confidentiality tools to enable storing sensitive data "on the chain." Instead, many organizations will need to stand up "off-chain" storage solutions that can be used to store data products while the blockchain records the hashes of those products as transactions. These off-chain storage solutions should be encrypted per any regulatory requirements or industry-best practices.

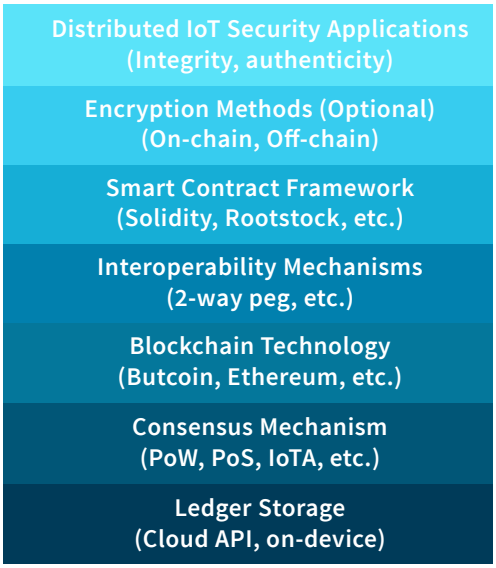
DEPLOYMENT OPTIONS

Blockchain can be deployed within three areas [11]:

- **Unpermissioned area (public):** Every node can read and send transactions and can also participate in the consensus process. The POW consensus algorithm is best suited to unpermissioned areas.
- **Consortium area (e.g. partially permissioned):** Defined nodes can participate in the consensus process. Reading and sending transactions may be public or restricted. The BFT consensus algorithm is best applied in consortium deployments [13], such as the Hyperledger - an open source effort governed by the Linux Foundation.
- **Permissioned area (private):** Trusted organizations can write transactions into the blockchain, and the consensus mechanism is irrelevant. This kind of deployment works best for regulated industries or between organizations that belong to the same legal entity [11]. Vertical initiatives, such as the one proposed by the project R3 [19] and Chain Core [20] for financial institutions, will likely be permissioned (private) blockchains governed by a central authority.

Bitcoin [12] and Ethereum [14] are unpermissioned blockchain implementations that have gained popularity for their support of distributed applications (DAP). Ethereum includes the Solidity programming language that can be used to easily construct smart contracts that enable autonomous peer-to-peer transactions between IoT devices. Solutions, such as BTC Relay [15], provide capabilities for settling a smart contract in Ethereum with payment flowing through Bitcoin.

A blockchain implementation is a framework for building services, such as cryptocurrency, distributed application and smart contracts, on the blockchain network. The framework describes the theoretical concepts to be used and how they can be combined (e.g. consensus mechanism, digital signature, cryptographic methods and communication properties). It is up to the implementer to specify and detail technical components that implement the framework. The figure to the right provides a view into the technology components to consider when designing a blockchain-based IoT security solution.



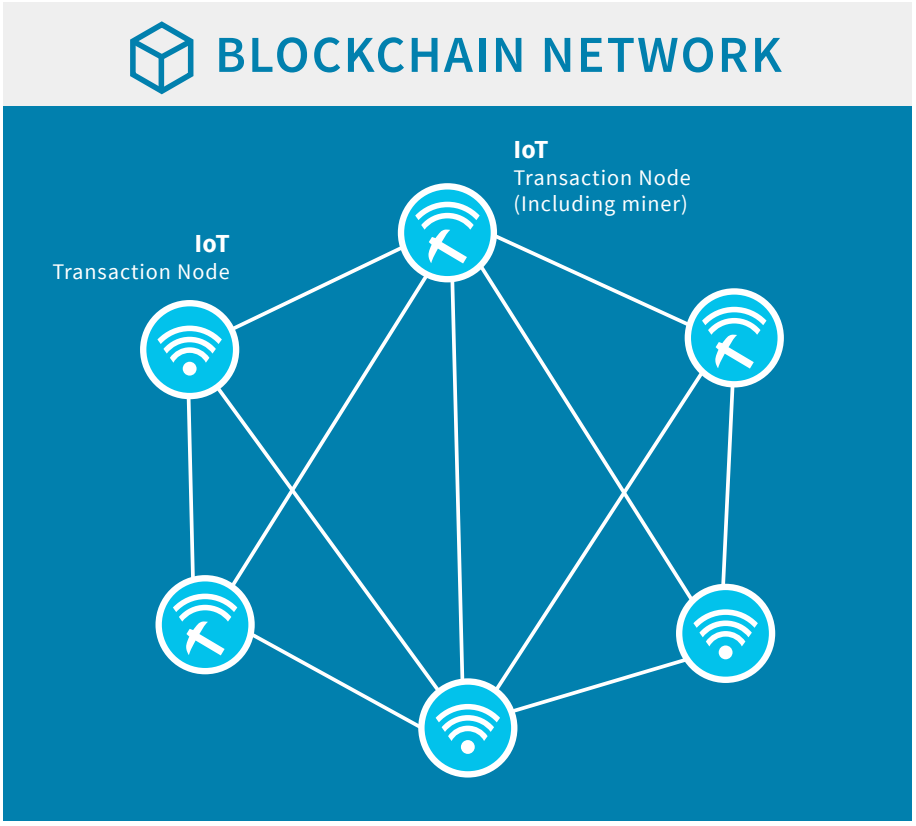
Multiple blockchain implementations are possible, and each proposes different

usages and services. A “trustless exchange” between these networks will be possible using relays, named 2-way peg [9], that allow access to another blockchain’s functionalities. Rootstock [10] is an open-source smart contract platform with a 2-way peg to Bitcoin.

Adaptation of blockchain technology to the IoT requires the examination of an IoT architectural pattern based on the blockchain service. The defined pattern should include three components, as outlined in this section:

THE COMMUNICATION MODEL

The communication model describes the installation of blockchain software directly on IoT nodes and/or in the cloud with Application Programming Interfaces (APIs) to the IoT nodes. The figure below shows a common and accepted model combining blockchain technology and IoT when the IoT edge devices have robust capabilities that make them capable of hosting the transaction node software, storing the ledger and maintaining communications across the network of nodes.



Each IoT node acts as a blockchain transaction node

IoT Transaction Nodes

In the previous figure, each IoT device hosts the ledger and is capable of participating in blockchain transactions, including mining. Each device is provisioned with a private key or includes functionality to internally self-generate a private key to participate in network transactions. This end-state model provides three fundamental capabilities that can be enabled with a blockchain service:

- A network of autonomous IoT devices, including autonomous coordination (e.g. consensus and peer-to-peer messaging)
- A ledger of transactions where any IoT device can create a transaction running cryptographic features
- A distributed database where any IoT device has an up-to-date version of the ledger

Hardware limitations make adopting this model difficult for IoT at the present time. Challenges include:

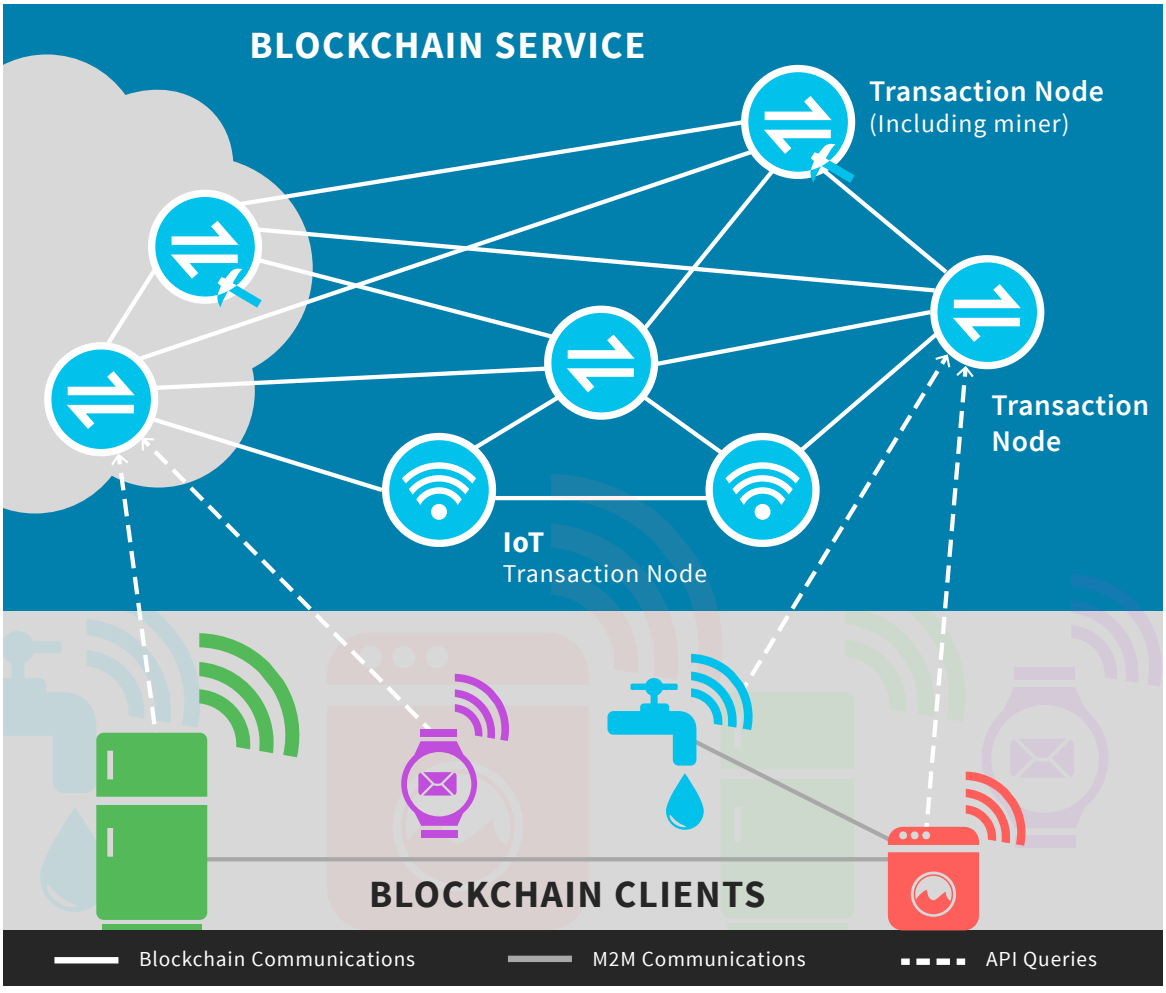
1. **Low processing:** Computations in a blockchain service require high CPU, memory and power capabilities. The major potential hardware hogs in a blockchain platform are mining for POW; smart contract execution; and cryptographic primitive execution.
2. **Small storage:** The volume of transactions added to the ledger grows and becomes cumbersome to maintain even with small transaction data.
3. **Limited connectivity:** An IoT device may make use of low bandwidth Internet or radio access, which can introduce performance issues during download and synchronizing with the ledger.

Some companies, such as IOTA [8], propose new approaches for "blockchaining" tiny sensors, including the following solutions:

- Simplify the mining process to decrease hardware requirements
- Implementing micro-transactions relevant for IoT interactions
- Maintain lightweight ledgers

Cloud-Enabled IoT Blockchain Network

In a cloud-enabled blockchain network, transaction and mining nodes are located both in the cloud and on-premise. Depending on the implementation, the nodes may be enterprise servers; enterprise/personal computers or smart devices (e.g. phones or tablets); cloud-based virtual machines and IoT devices with sufficient hardware resources (CPU, RAM, storage, etc.).



IoT devices with limited hardware resources act as blockchain clients. They do not store the distributed ledger. These clients interact with upstream cloud-based blockchain transaction nodes through APIs. APIs will likely be either HTTP REST or JSON RPC.

IoT devices collect data relayed to transaction nodes for processing by the blockchain service or participate in smart contract transactions by pointing to the blockchain nodes running in the cloud. In this context, IoT devices are still provisioned with private keys to sign their data. The signed data are then sent upstream to the transaction nodes for processing. A separate agreement of trust between the IoT device and the transaction node must be in place in order to securely send data. For example, a one-to-one relationship may use whitelisting and two-way authentication between two devices (an IoT device and a transaction node). Hardware security should also be employed to securely store private signing keys.

For a permissioned area (private blockchain service), access to mining nodes may be restricted to authorized operators.

In a consortium area (partially permissioned blockchain service) or a permissioned area (private blockchain service), members can decide to implement this architecture pattern for improving security or for regulation compliance purposes.

Bitcoin implementation proposes this kind of feature using "thin clients," also named Simplified Payment Verification (SPV) [21], which do not store a complete copy of every block of the ledger. These "thin clients" communicate with a node using Bitcoin Client API (BCCAPI) [22].

Messages can be exchanged between multiple IoT devices. These messages contain data that is integrated into the transactions relayed by IoT devices participating in the exchanges to the transaction nodes. Communications protocols and message formats between IoT devices are outside the scope of the blockchain implementation: these communications refers to machine-to-machine communications [23], such as Message Queue Telemetry Transport (MQTT).

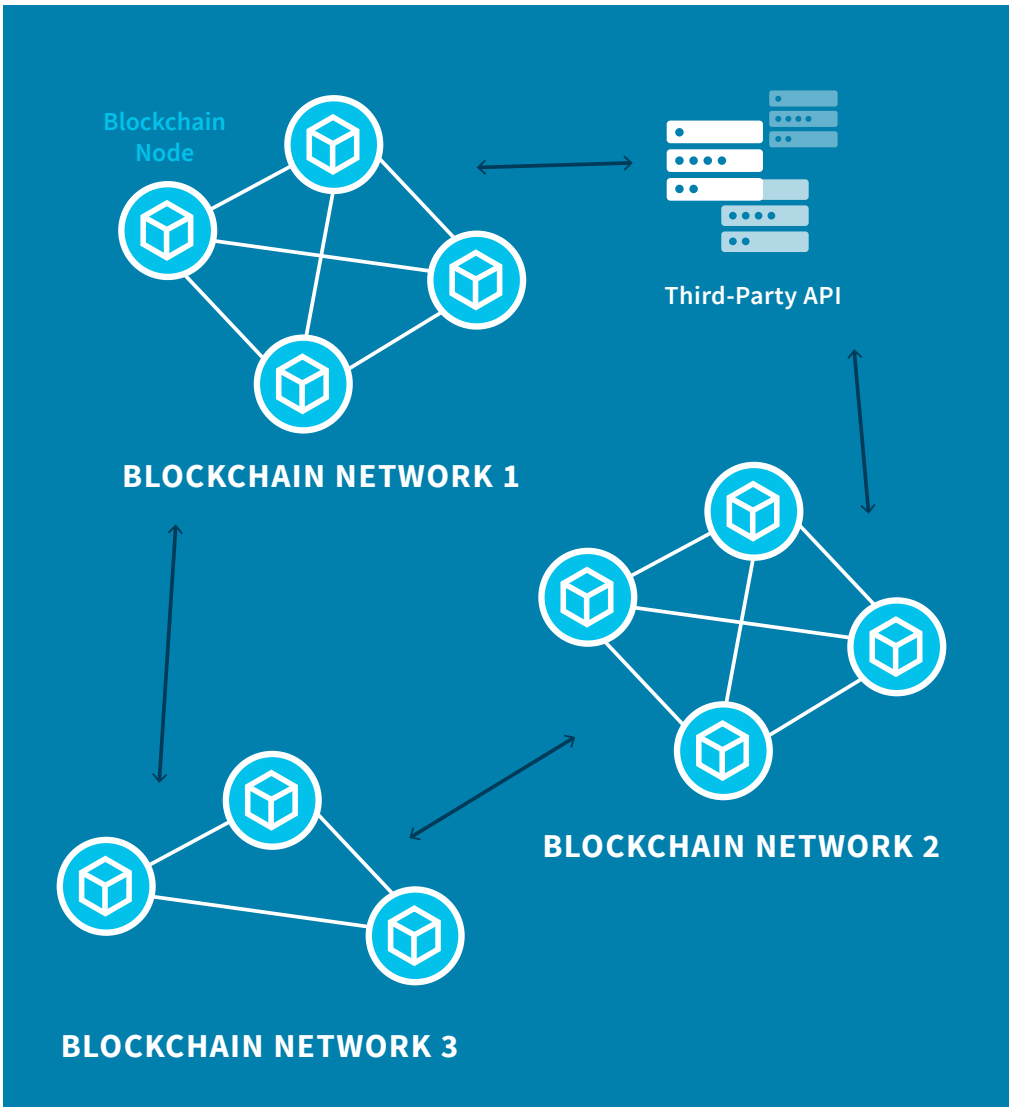
A RICH ECOSYSTEM FOR LEVERAGING INTEROPERABILITY CAPABILITIES

Developing an ecosystem around blockchain technology would be an opportunity to accelerate its adoption. This potential ecosystem would provide capabilities for simplifying the integration of IoT into a blockchain service.

- **Service providers**, such as Blockcypher [24], provide API capabilities that simplify interactions between IoT and blockchain service clients and services. API intermediation allows the development of features on IoT that communicate with different blockchain services by focusing on the value of the service instead of the technical implementation of the blockchain technology.
- **Solution providers**, such as Credits [25], provide frameworks for quickly building a private blockchain service. These frameworks run on transaction nodes. Each node is accessed by clients through APIs. These frameworks also provide capabilities to interact with other blockchain services.

COHABITATION BETWEEN MULTIPLE BLOCKCHAIN SERVICES

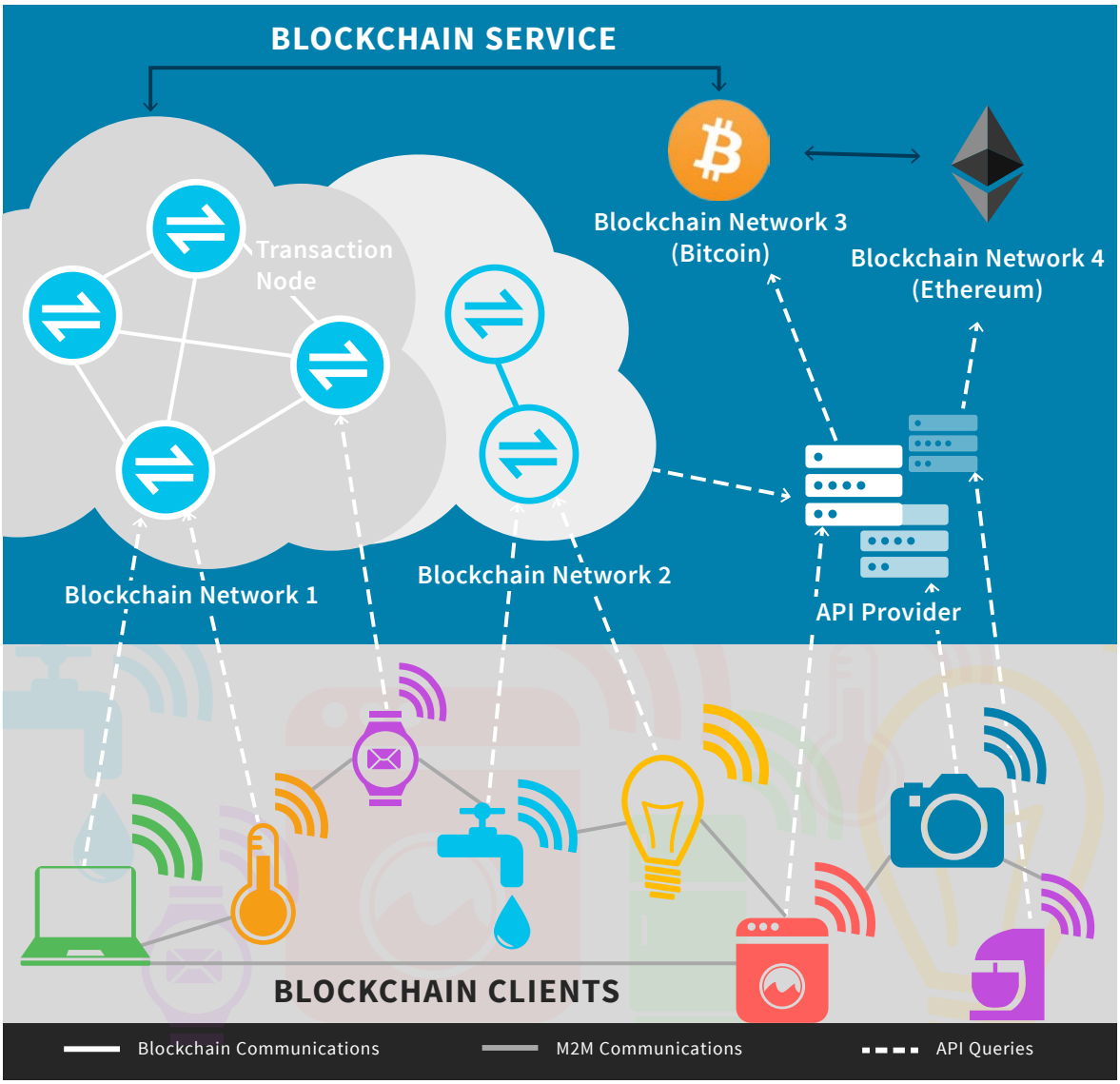
As shown in the diagram on the following page, another concept focuses on multiple blockchain services, each offering different features and currencies, and their number will grow. These blockchain services will propose complementary features. Each blockchain service can be pegged to others natively or using APIs provided by third parties [26].



Each blockchain service can run in different contexts, such as personal home network, enterprise and the Internet.

THE IOT ARCHITECTURE PATTERN BASED ON A BLOCKCHAIN TECHNOLOGY

The CSA IoT and Blockchain/Distributed Ledger Technology Working Groups propose the following system by which IoT clients within a multi-blockchain service can collaborate.



Blockchain service 1 is a dedicated enterprise implementation:

- Transaction nodes are corporate computers or servers hosted in the cloud
- IoT blockchain clients are sensors and smart devices deployed within the enterprise area

Blockchain service 2 is a consumer smart home:

- Transaction nodes are personal computers and other devices or cloud subscriptions
- IoT blockchain clients are smart devices, such as refrigerators, temperature sensors and security cameras.

The architecture in which IoT devices are clients of a blockchain service is primarily adopted by current industry efforts for implementing blockchain technology.

SELECTION OF BLOCKCHAIN TECHNOLOGY FOR IOT SECURITY

Blockchain technology can help secure IoT devices. IoT devices can be configured either to make use of public blockchain services or to communicate with private blockchain nodes in the cloud over a secure API. Incorporating blockchain technology into the security framework of an IoT system allows IoT devices to securely discover each other, encrypt machine-to-machine transactions using distributed key management techniques, and validate the integrity and authenticity of software image updates, as well as policy updates.

Based on the potential architectural patterns detailed in this report, an IoT device will communicate with a blockchain transaction node via an API, allowing even constrained devices to participate in the blockchain service.

To ensure security, care should be taken during the bootstrapping of an IoT device onto a particular blockchain service. Below is a use case for IoT discovery that supports the enrollment of an IoT device into a transaction node. The IoT device must first be provisioned with credentials that can be used to prove authorization in order to be added to a transaction node. This credential provisioning must be done in a secured environment that safeguards against threats of a particular IoT device ecosystem.

Our review of blockchain technology and the market initiatives available to develop it highlights five features to consider when securing the IoT using blockchain technology:

1. Scalable IoT discovery
2. Trusted communication
3. Message authentication/signing (Chain of Things [27])
4. IoT configuration and updates
5. Secure firmware image distribution and update

1. Scalable IoT discovery

Smart cities and large enterprise IoT deployments will result in the activation of potentially thousands or tens of thousands of IoT devices [28] that must work together. Often, these devices will coordinate with each other in autonomous machine-to-machine transactions. The devices must also be able to discover legitimate peers and services with which to interact. IoT systems can take advantage of scalable IoT discovery using both public and private blockchain implementations.

Within Bitcoin for example, a set of hard-coded Named DNS seeds provides bootstrap services for new users and devices. These DNS Seeds can be preconfigured within an IoT device. IoT devices query these addresses and are provided with the IP address of a full node. The IoT device then registers itself into a node and requests a list of other IoT devices on the network. When provisioned, the IoT device can begin peer-to-peer communications while promulgating peer discovery information to neighbors across the network.

The preconfiguration (hard-coding) of the Named DNS Seed addresses reduces the ability to perform a man-in-the-middle (MITM) attack. IoT devices receive information from multiple DNS Seeds before choosing a node to enroll within. DNS Sec must be used to secure the name resolution of root servers and mitigate DNS spoofing attacks.

Named DNS Seed addresses should be hard-coded into the firmware; Case 5 of this paper provides a way to secure firmware image distribution and update.

A private blockchain service can also support the bootstrap and enrollment of IoT devices onto a network. Transaction nodes will authenticate the IoT devices prior to providing a trusted node list. IoT devices are provided with enrollment

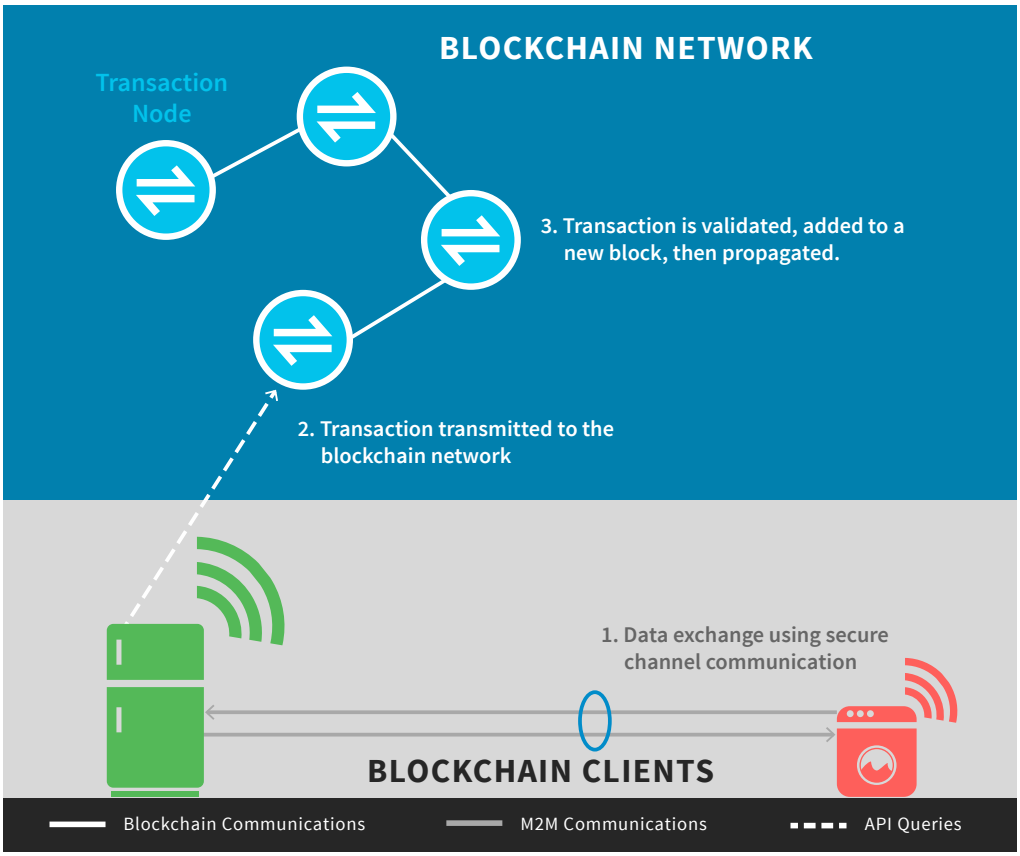
credentials that can include the following:

1. Security credentials installed on or internally self-generated in the IoT device during setup must be generated and provided using a safe process that could be part of blockchain implementation.
2. Credentials provided by the owner or installation technician of the IoT device would initialize the device enrollment into a security server to get specific credentials for the IoT.

In either case, the enrollment process must be enforced to ensure only legitimate IoT devices can be added to the blockchain service. All communication described must be authenticated and encrypted to ensure confidentiality and integrity.

For further information on the ability to register device identities onto the blockchain, visit the Trusted IoT Alliance to review the blockchain APIs developed for registering thing identities on the blockchain.

2. Trusted Communication

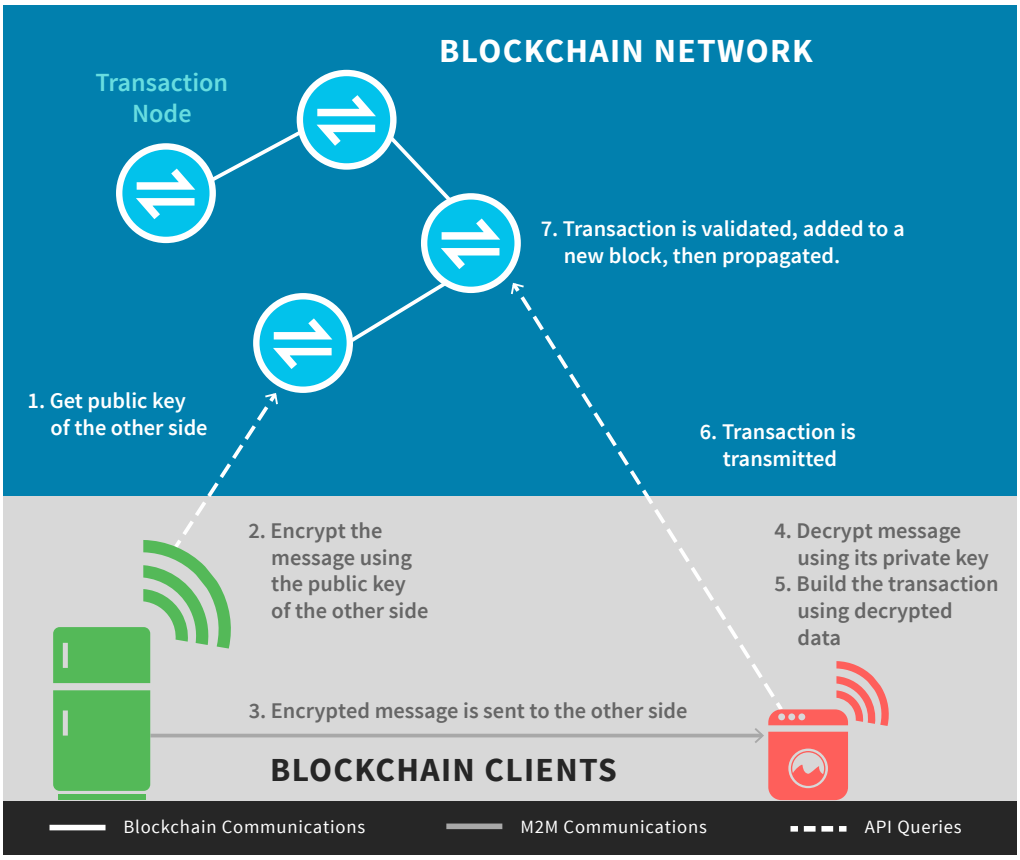


In some contexts, e.g. public deployment, IoT devices require the use of a secure communication channel for exchanging the data needed to build the transaction to be stored into the ledger. This ledger can also be used to store public encryption keys.

If the exchanges must be kept confidential, an IoT device (sender) sends encrypted message to a peer IoT device (receiver) using the public key of the receiver IoT device, which is stored in the blockchain service.

In order to facilitate this secured transaction, the IoT sender requests its transaction node to get from the blockchain ledger the public key of the IoT receiver. The IoT sender encrypts the message using the public key of the IoT receiver.

Only the receiver can decrypt the message with their private key. Key agreement algorithms such as Elliptic Curve Diffie Hellman (ECDH) should be used to create keys to protect transactions such as Content Encryption Keys (CEK) and /or Traffic Encryption Keys (TEKs).



In this use case, the blockchain service acts as a distributed public key infrastructure [29]. Public keys are stored within transactions: when a new IoT device is enrolled into a private or a public blockchain service (see previous section), a new transaction is created. This transaction is composed of IoT properties including its public key. If an IoT device must renew its certificate, re-enrollment occurs. Revoked certificates can also be added to the blockchain service as transactions. The secured transactions history recorded in the ledger provides the consistency of IoT device keys across their lifetime.

There are multiple types of cryptographic keys used within a blockchain implementation. Keys used to secure blockchain transactions are often known as wallet keys. The keys discussed in this use case represent Identity Keys and can be used to generate Traffic Encryption Keys (TEKs) or Content Encryption Keys (CEKs) to safeguard communications between IoT device peers.

- **IoT Identity Keys:** Asymmetric key pair used to generate key material for encryption of message content and traffic flows between IoT devices
- **Wallet keys:** Used to secure transaction stored in the ledger; may include IoT Identity keys

3. Semi-Autonomous Machine-to-Machine Operations

A critical enabler of IoT technology is the ability for machines to work together in a semi-autonomous fashion towards achievement of a specific goal. Blockchain can act as a security-enabler of these autonomous transactions using smart contract functionality.

Smart contracts can be written to include the rules, penalties and conditions of the contract. Edge IoT devices can then be configured with an API to interact with the smart contract to enter into agreements with peer devices and/or services. Each transaction must meet the conditions of the contract prior to execution and all transactions are written to the blockchain.

Smart contracts can enforce access restrictions as to who (which IoT devices) can enter into transactions. Each transaction is signed with the wallet key of the IoT node and wallets should be stored in hardware security containers. Transaction recording on the blockchain ensures that the transactions cannot later be repudiated (for example when a service providers' IoT device enters into a transaction with a consumer's IoT device).

4. IoT Configuration and Update Controls

Blockchain technology is promising in the area of trusted, secure configurations as more IoT devices are natively connected to cloud services. Below are three security approaches:

1. The ledger can host IoT properties, such as the last version of validated firmware and configuration details. During bootstrap, the IoT device asks the transaction node to get its configuration from the ledger. The configuration should be encrypted in the ledger to avoid discovery of IoT network topology by analyzing the content of the public ledger.
2. The ledger can host the hash value of the latest configuration file for each IoT device. The IoT device downloads the latest trusted configuration file each night (or set period of time) from a cloud service and then utilizes the transaction node API to retrieve and match the hash value stored on the blockchain. This allows administrators to flush bad configurations on a regular basis and reboot the IoT devices in their network with fresh configurations.
3. The same process discussed in #2 above can be applied to firmware images of IoT devices, although this may require additional bandwidth capacities at the point of IoT device.

5. Secure Firmware Image Distribution and Update

Similar to supporting the download of known-trusted configurations from a cloud service provider, blockchain technology can also support the trusted imaging process for IoT devices. An IoT device developer that also authors IoT device firmware can implement their own blockchain or use a public blockchain. The developer can take hashes of the latest known trusted images for its device families and load those hashes into the blockchain. This method supports enhanced IoT device security in three ways:

1. IoT devices can be configured via API to download new firmware images on a recurring basis. Because most IoT devices do not need to keep-state or store data in memory, they can be overwritten as needed. Setting up a daily or weekly image update process, for example, could be enabled by validating the image hash against the vendor's blockchain.
2. IoT devices can use a blockchain-based image update process to validate all updates provided by the vendor.
3. IoT devices can use either method 1 or 2 above to validate all updates, and in addition require the device owner to approve the firmware update (using a secure method).

IoT manufacturers should enhance current standard software signature approaches by storing the digital signature of the firmware in the ledger, instead of publishing it in its website Before applying the update, the IoT devices get the digital signature of the new firmware from the ledger, then verify it using a maintenance public key. This maintenance public key could be fused at fabric/hardware level (no change/update capabilities).

Warning: The maintenance private key of a manufacturer must be secured to avoid compromising all firmware. An attacker that obtains the private key could make available malicious firmware with a seemingly "valid" digital signature.

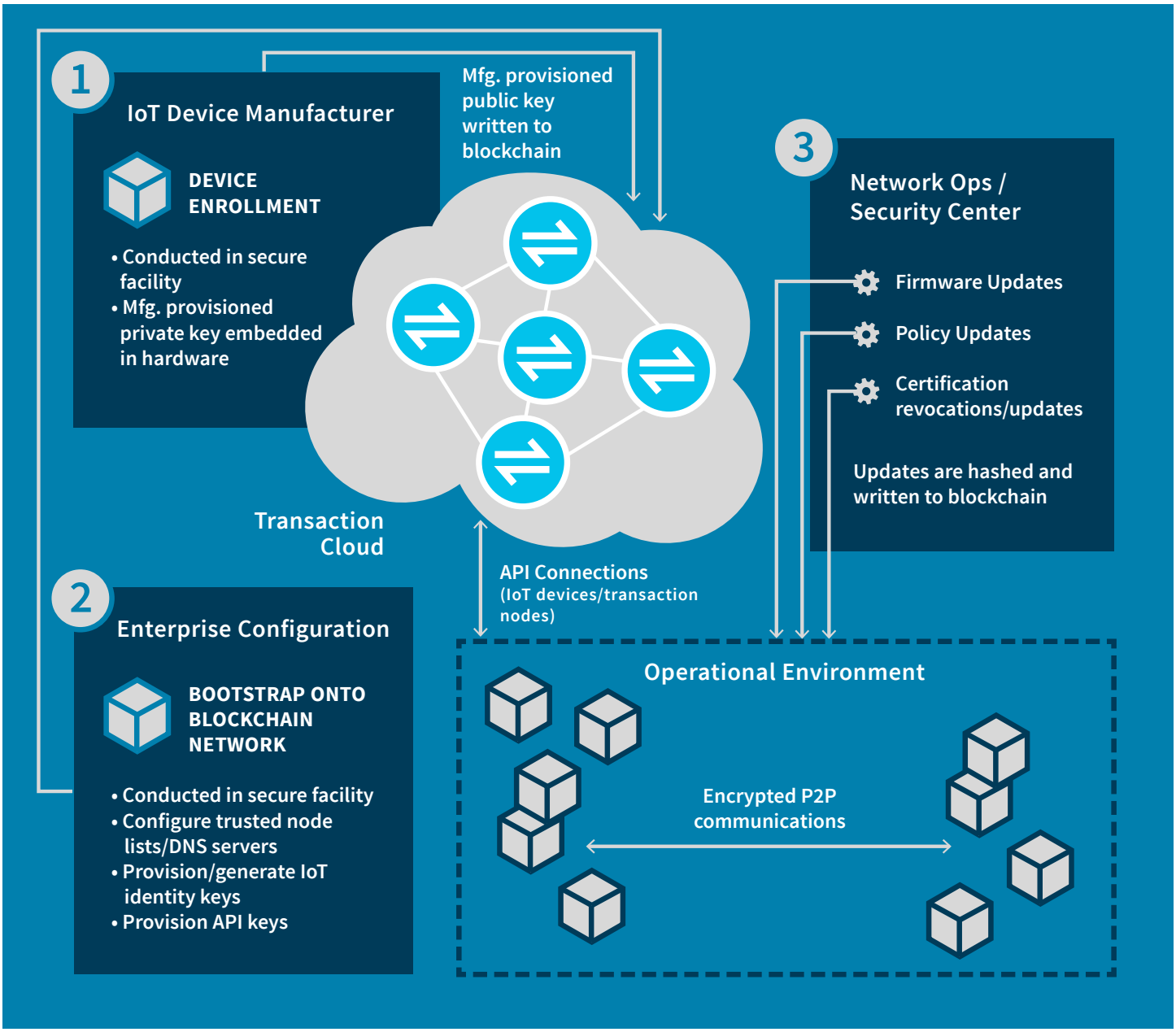
The process of changing the public key of the manufacturer on all devices requires significant effort.

Firmware reputation-based update (Chain of Things [27]):

The transaction history capabilities of the ledger can aggregate notation of new firmware via a community of experts in order to improve trust of the firmware by avoiding installing firmware infected by malware and backdoored firmware.

The owner/administrator of the IoT devices should also configure an automatic update of the IoT when the firmware reputation reaches a specific level of positive votes in the ledger. This "acceptance" of the IoT device in the blockchain service could be based on the device's firmware reputation in the ledger, which would yield the following benefits:

- Avoid having vulnerable devices connected to a blockchain service
- Enforce security update processes on IoT devices
- Define minimum required security requirements on blockchain service



CONCLUSION

Organizations implementing IoT solutions continue to experience challenges identifying security technologies and approaches sufficient to mitigate unique threats to IoTs. Blockchain technology promises to play a major role in addressing these challenges. Niche security vendors will begin to offer these services, but it is possible to take advantage immediately of the integrity and authenticity services provided by blockchain implementations.

Throughout this paper, we have highlighted features to consider when attempting to secure connected devices using blockchain technology. Yet, due to hardware limitations of IoT, we conclude that in a context of several hundred thousand or more IoT devices [28], many of these devices could not serve as transaction nodes (generating transactions, providing consensus, etc.), and thus would fall outside the secure blockchain. Many devices will benefit from the security and other features offered by blockchain services through APIs from upstream transaction nodes of networks or by specialized intermediaries. Those upstream capabilities can be used to secure IoT devices (configuration and update control, secure firmware update) and communications (IoT discovery, trusted communication, message authentication/signing).

We hope this document inspires business leaders and developers embracing the blockchain opportunity to extend the capabilities of this technology to secure the Internet of Things.

REFERENCES

[1] IDC FutureScape https://www.idc.com/url.do?url=/getfile.dyn?containerId=US42259417&attachmentId=47254824&elementId=54425583&term=&position=1&page=1&perPage=50&id=b28d2b1c-ddd5-4e60-a2c2-de3a4f7ee253

[2] Bitcoin Venture Capital https://www.coindesk.com/bitcoin-venture-capital/

[3] Blockchain https://en.bitcoin.it/wiki/Block_chain

[4] Leslie Lamport, Robert Shostak, and Marshall Pease. The Byzantine generals problem. ACM Trans. Program. Lang. Syst., 4:382–401, July 1982. http://citeseerx.ist.psu.edu/viewdoc/download?doi=10.1.1.126.9525&rep=rep1&type=pdf

[5] Miguel Castro and Barbara Liskov. Practical Byzantine fault tolerance and proactive recovery. ACM Trans. Comput. Syst., 20(4):398–461, November 2002. http://citeseerx.ist.psu.edu/viewdoc/download?doi=10.1.1.84.6725&rep=rep1&type=pdf

[6] Nakamoto, Satoshi. "Bitcoin: A peer-to-peer electronic cash system." (2008): 28. https://bitcoin.org/bitcoin.pdf

[7] Vasin, Pavel. "Blackcoin's proof-of-stake protocol v2." (2014) https://blackcoin.co/blackcoin-pos-protocol-v2-whitepaper.pdf

[8] What is IOTA? https://iota.readme.io/v1.1.0/docs

[9] Trustless exchange and pegging of BTC in Ethereum https://medium.com/@ConsenSys/taking-stock-bitcoin-and-ethereum-4382f0a2f17#.h6nhib6ql

[10] Rootstock http://www.rsk.co/

[11] On Public and Private Blockchains https://blog.ethereum.org/2015/08/07/on-public-and-private-blockchains/

[12] Bitcoin https://bitcoin.org/

[13] The Quest for Scalable Blockchain Fabric: Proof-of-Work vs. BFT Replication http://vukolic.com/iNetSec_2015.pdf

[14] Ethereum https://www.ethereum.org/

[15] BTC Relay http://btcrelay.org/

[16] Ethereum Blockchain as a Service now on Azure https://azure.microsoft.com/fr-fr/blog/ethereum-blockchain-as-a-service-now-on-azure/

[17] Hyperledger https://www.hyperledger.org/

[18] IBM Blockchain on Bluemix https://www.ibm.com/blockchain/offerings.html

[19] Project R3 https://r3cev.com/

[20] Chain Core https://chain.com/technology/

[21] Thin Client Security https://en.bitcoin.it/wiki/Thin_Client_Security

[22] BCCAPI (Bitcoin Client API) https://en.bitcoin.it/wiki/BCCAPI

[23] Machine-to-Machine https://en.wikipedia.org/wiki/Machine_to_machine

[24] Blockcypher https://www.blockcypher.com/

[25] Credits http://credits.vision/

[26] Drivechains sidechains and hybrid 2-way peg designs http://www.the-blockchain.com/docs/Drivechains%20sidechains%20and%20hybrid%202-way%20peg%20designs%20-%20Sergio%20Lerner%20-%202016.pdf

[27] Chain of Things http://www.chainofthings.com/

[28] Gartner Says 8.4 Billion Connected "Things" Will Be in Use in 2017, Up 31 Percent From 2016 http://www.gartner.com/newsroom/id/3598917

[29] Decentralized Public Key Infrastructure http://www.weboftrust.info/downloads/dpki.pdf

[30] Cryptocurrency Statistics https://bitinfocharts.com/