nChain
Data without question

DR CRAIG WRIGHT – CHIEF SCIENTIST – NCHAIN GROUP

Bitcoin Satoshi's Vision and the Internet of Things: A P2P World Using IPv6

12.05.22

nChain

- Introduction – Bitcoin: A Peer-to-Peer System with IP-to-IP Connectivity
- Privacy and Accountability in a P2P Exchange
- IPv6 and the Direct Exchange of Digital Assets
- IP-to-IP Transactions Using IPv6
- IoT in a World of Bitcoin and IPv6
- Controlling the Performance of a Contract

INTRODUCTION

Bitcoin: A Peer-to-Peer System with IP-to-IP Connectivity

Introduction

Bitcoin: A Peer-to-Peer System with IP-to-IP Connectivity

1. ‘Peer-to-peer’ refers to the direct exchange between two users
2. The original construction of Bitcoin includes IP-to-IP connectivity
3. A transaction is conducted P2P and settled on the network
4. Identity can be determined without leaking private information to the public
5. The new model for commerce is one where records are immutable, cannot be lost, and allow parties to trade securely and privately

Privacy and Accountability in a P2P Exchange

Privacy and Accountability in a P2P Exchange

From the original Bitcoin website:

There are two ways to send money.

If the recipient is online, you can enter their IP address and it will connect, get a new public key and send the transaction with comments.

If the recipient is not online, it is possible to send to their Bitcoin address, which is a hash of their public key that they give you. They'll receive the transaction the next time they connect and get the block it's in.

This method has the disadvantage that no comment information is sent, and a bit of privacy may be lost if the address is used multiple times...

Bitcoin was constructed in a way that allowed people to make transactions whether they were online or not.

The primary methodology involved connecting directly to the individual you were paying and exchanging information with.

Privacy and Accountability in a P2P Exchange

P2P: The Way It Was Designed

In conducting a transaction, user Alice transmits information to Bob, whom she is paying.

Bob then settles the transaction on the network, by sending it to the nodes.

In the exchange process of **simplified payment verification (SPV)**, where invoices and purchase orders can be created and exchanged between individuals or companies, Bob and Alice exchange information that allows each party to determine the identity of the other.

In a peer-to-peer exchange where a buyer and seller directly exchange a Bitcoin transaction using the original IP-to-IP protocol, the transaction would be said to be instantaneous and, as such, would occur at the location where the recipient resides.

Privacy and Accountability in a P2P Exchange

Bitcoin Data Interchange & Accountability

When a user exchanges information with another user, such as through the original IP-to-IP protocol, they are simply **network clients**. They can take the form of an SPV client or a web client or many other forms, including one of a fat client that self-propagates.

The original client was set to use IP-to-IP connectivity, to allow peers to communicate directly. Communication between Alice and Bob is **private and direct**.

Tools mapping electronic data interchange (EDI) to a Bitcoin transaction would simply look like EDI tools of today.

Alice may record proof of connection with Bob, which can be signed, **verifying** that only Bob received the transaction. Bob can now **validate** the transaction received from Alice.

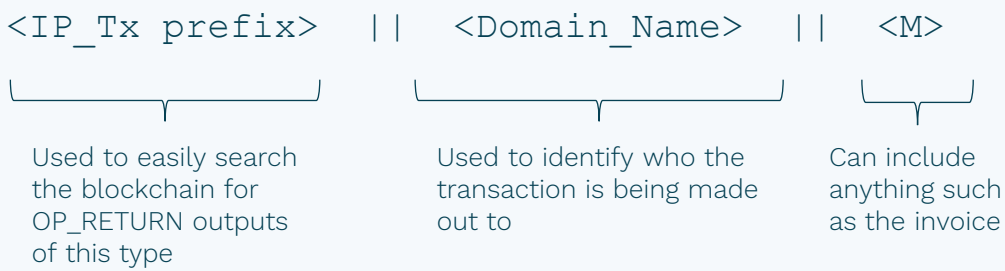
IP-to-IP Transactions Using IPv6

Non-Interactive Bitcoin IP Transaction Using IPv6 CGA(++)

The procedure here is a modification that takes advantage of the beneficial properties of IPv6 (and CGA).

To send a Bitcoin transaction to a recipient who isn't online (non-interactively), the sender can take advantage of the imbedded authentication, using a signature, in CGA++.

OP_RETURN Output Format



Example Non-Interactive Bitcoin IP Transaction Using IPv6 CGA(++)

$TxID$	
Inputs	**Outputs**
$< Sig\ P >\ <P>$	OP_RETURN <IP_Tx prefix> <IPv6_CGA(++)> <M>
	OP_DUP OP_HASH160 $<H(P_{IP})>$ OP_EQUAL OP_CHECKSIG

Domain Name Bitcoin Payments Using IPv6 CGA(++)

To send a payment to a domain name rather than an IPv6 CGA(++), the sender simply adds an extra step in the beginning of the procedure to resolve the IP address that maps to the domain name required.

In the CGA parameters, there is an extFields parameter where an optional variable-length field (default length 0) can be added. If the receiver/server has a domain name, they can put that there, so that the client has added security guarantees that the DNS mapping is authentic.

Client issues DNS query to resolve the IPv6 (AAAA DNS record) address mapping to the required domain name.

Verify DNS record(s) – if DNSSEC is being used.

Check that the domain name in the DNS record matches the domain name in the extFields CGA parameter.

Procedure continues as with *Non-Interactive Bitcoin IP Transaction Using IPv6 CGA(++) or Interactive Bitcoin IP Transaction Using IPv6 CGA.*

IoT in a World of Bitcoin and IPv6

An IoT Communication Protocol on Bitcoin (BSV)

- Integration of payment and control into one network
- Using existing infrastructure to piggyback messages regarding device-state changes
- Faster user-device interaction

IoT systems generate large volumes of data and require systems with network scalability, strong cybersecurity, reliable connectivity, and minimal network latency.

P2P architectures offer an efficient solution, whereby neighbours interact directly with one another.

Blockchain technology is the foundation for secure P2P communication and the development of IoT systems.

An IoT Communication Protocol on Bitcoin (BSV)

- Integration of payment and control into one network
- Using existing infrastructure to piggyback messages regarding device-state changes
- Faster user-device interaction

Theme	IoT requirement	Blockchain solution
Scalability	Automation	Automated contracts
	Internet traffic	P2P transaction throughput
	Resource-constrained devices	Lightweight clients
Security/ Privacy	Cybersecurity in the cloud	Distributed data storage; Validation
	Access control	Key management; Derivative keys

Multilevel IOT (MIOT) Controller

With BSV's **high-capacity and low-fee microtransaction throughput** and scalable network infrastructure, devices can be connected reliably and on a global scale, communicating at minimal costs.

By combining a multilevel hierarchy and blockchain-based communication protocol, MIOT enables:

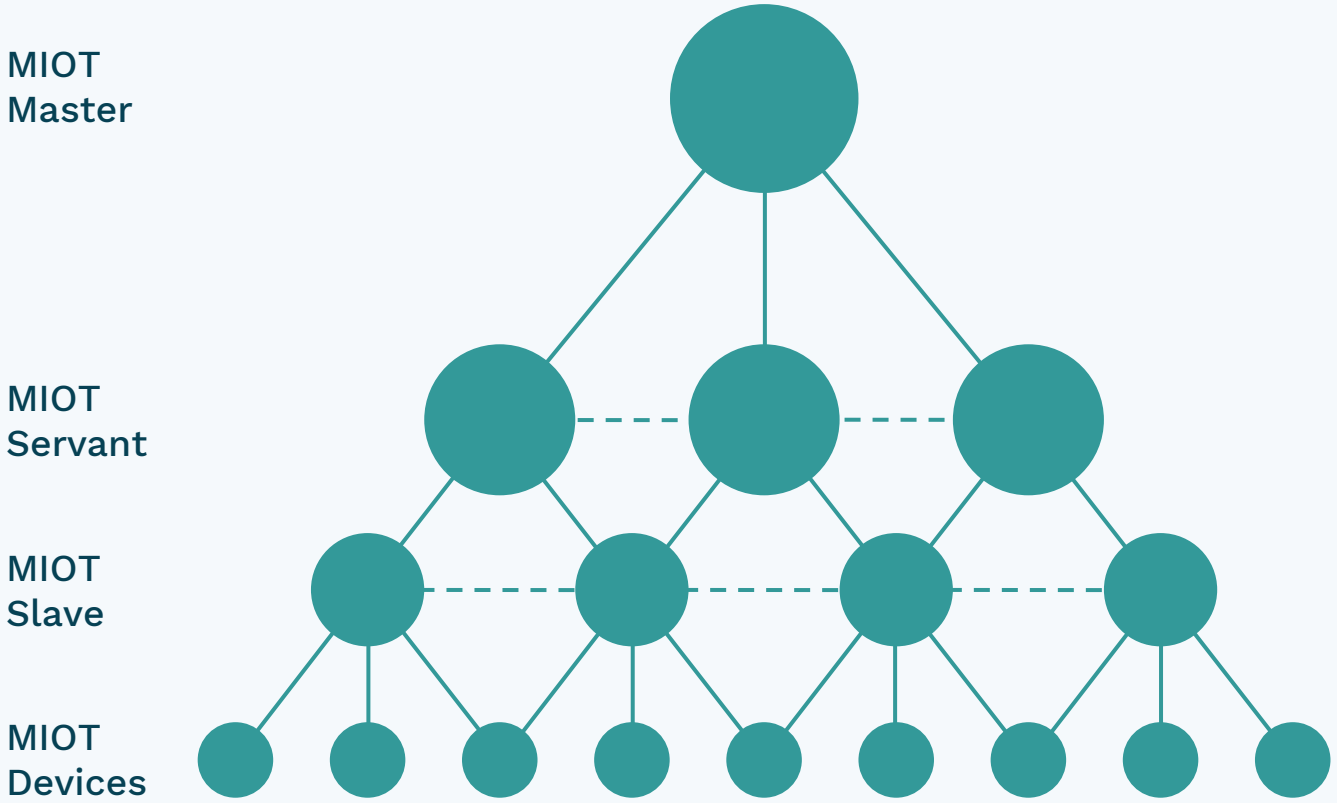
- large scale P2P communication using low-fee microtransactions;
- integration of digital asset transfer and control into one platform;
- independently verifiable identity and access management;
- low barriers to entry for IoT network devices;
- secure timestamped storage of IoT communication;
- IoT metadata accessible for auditing and performance monitoring; and
- privacy protection for user devices using end-to-end encryption with additional override, key-update, and privacy features.

MIOT Architecture

Localised control

The MIOT combines a command-control hierarchy with the use of a blockchain network infrastructure.

MIOT users create their own multilevel control hierarchy, which includes client-server and peer-to-peer relationships.

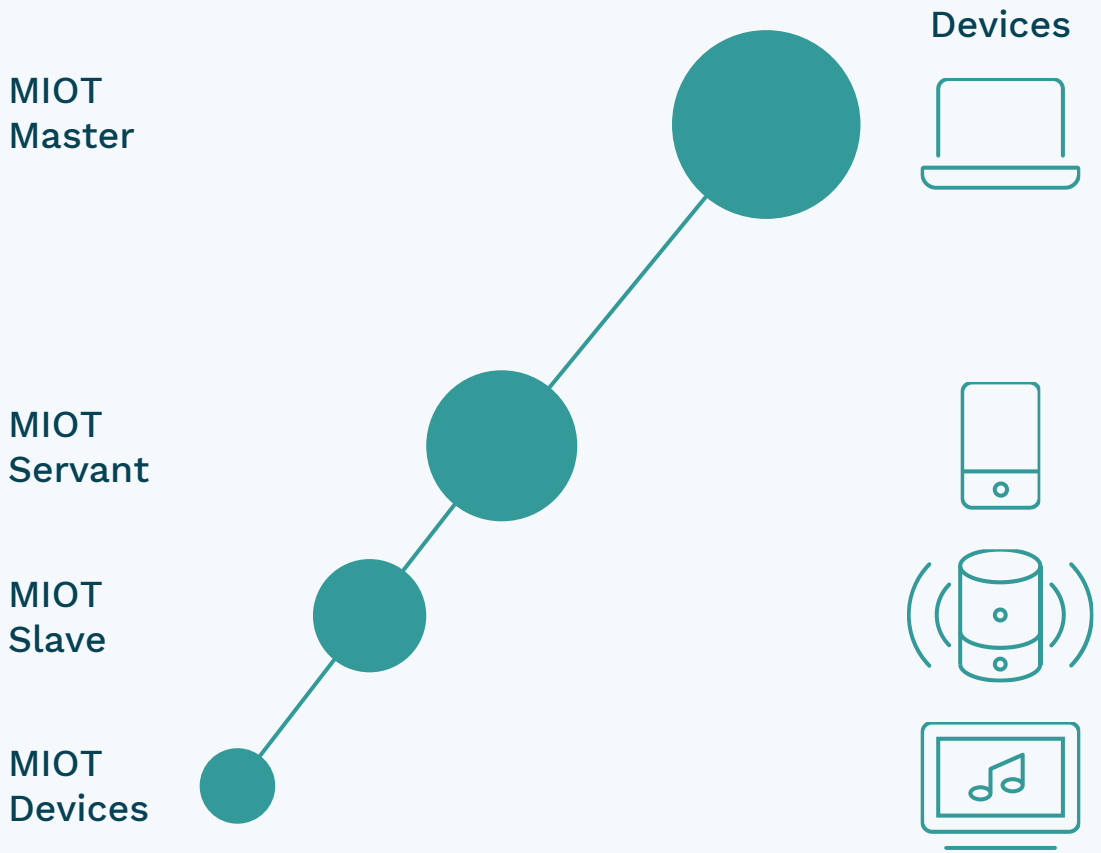


MIOT topology

MIOT Architecture

IoT classes

Devices on the MIOT network are classified according to computing resource and connectivity constraints.



MIOT topology
with example devices

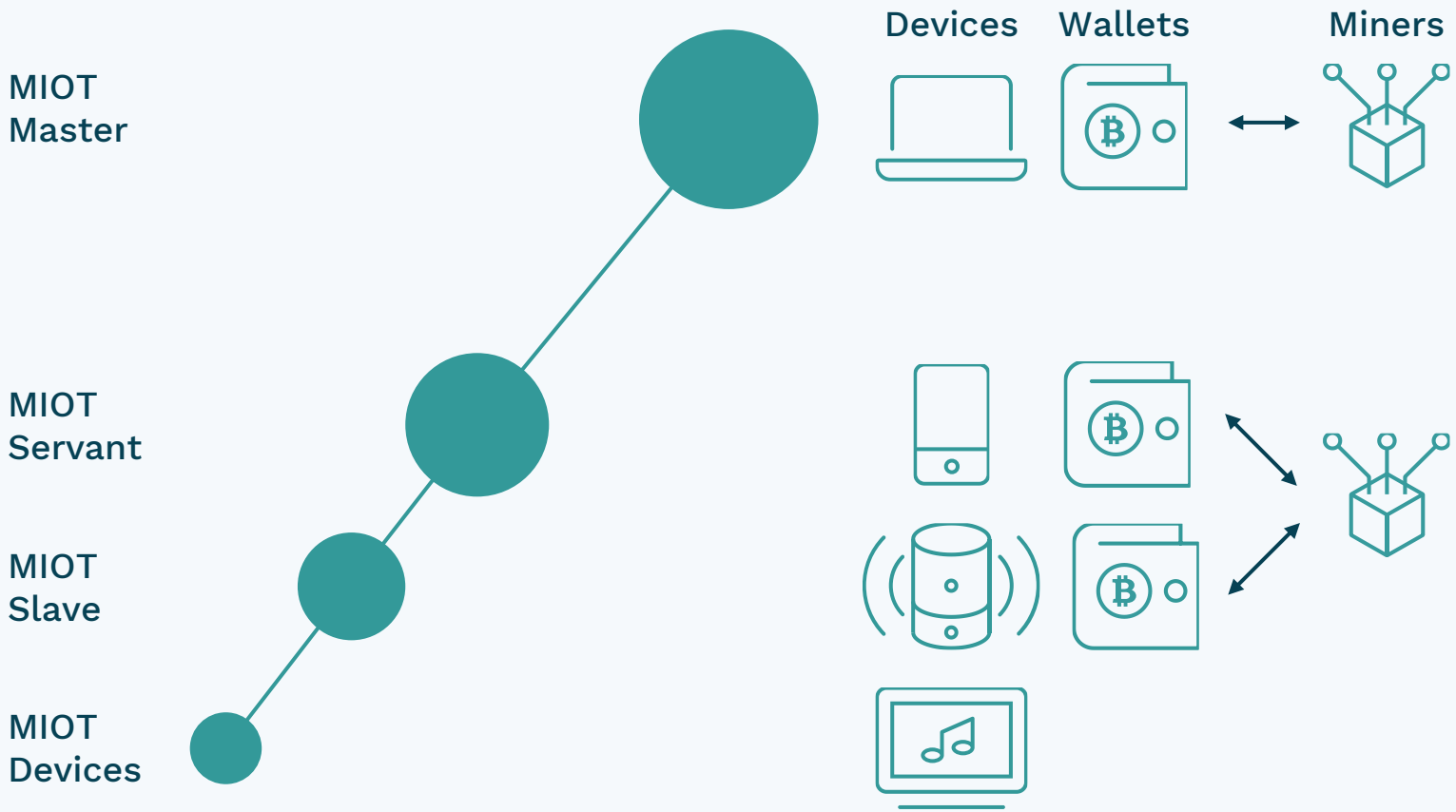
MIOT Architecture

MIOT devices

Devices on the MIOT network are classified according to computing resource and connectivity constraints.

MIOT wallet

- Secure messaging using Bitcoin transactions
- Including new peers with digital certificates
- Command acknowledgment using sighash
- Command approval using multisig transactions



MIOT topology
with example devices

Master, servant, and slave devices use the MIOT-configured Bitcoin wallet to communicate with each other and the Bitcoin network

MIOT Security

Encryption of the MIOT messages

Key masking

Identity and access management

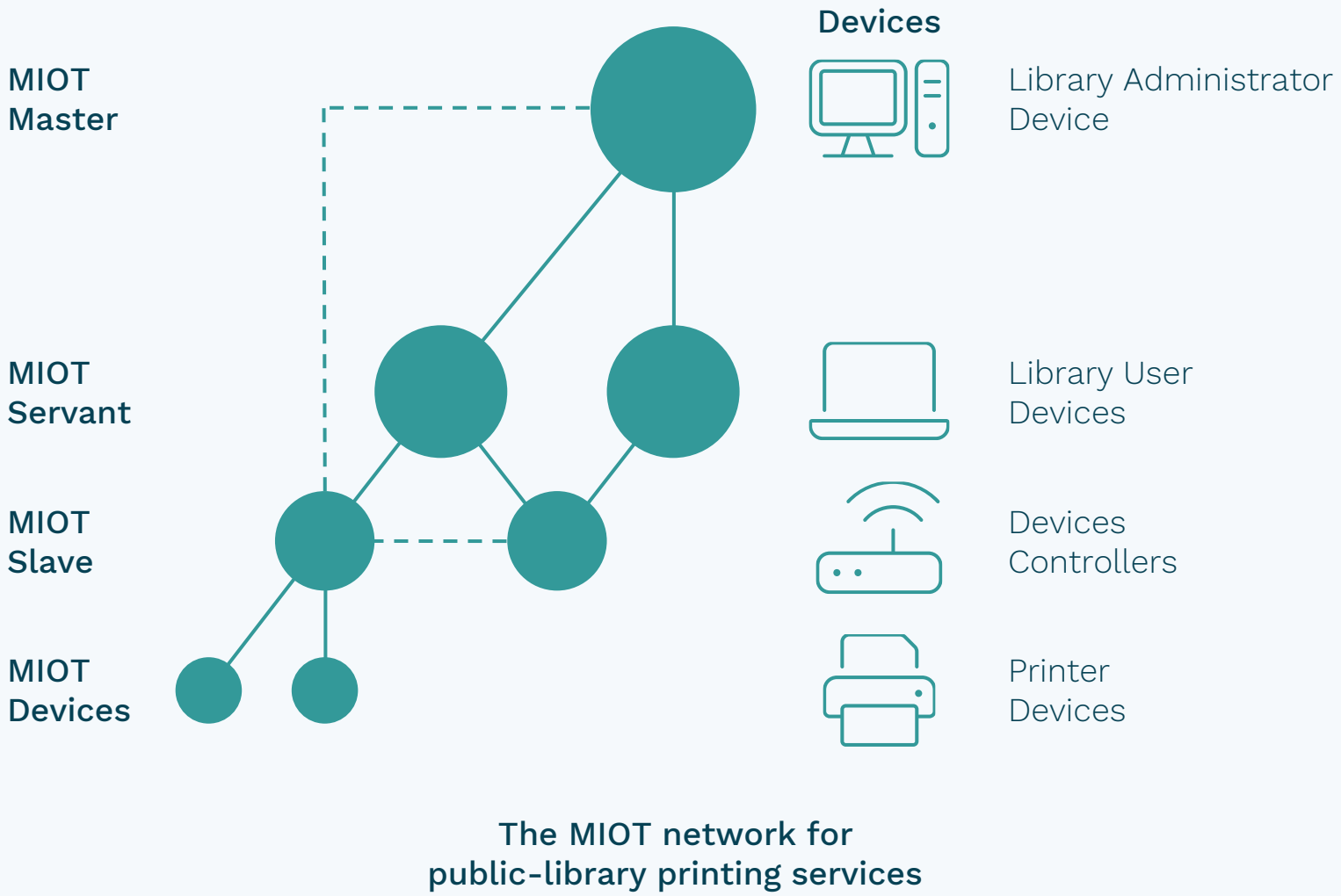
- Updating certificates
- Revoking certificates

Optimal MIOT network architecture

- Multifactor authentication and HPCC

Payment for Printing

Example use case



The MIOT network for public-library printing services

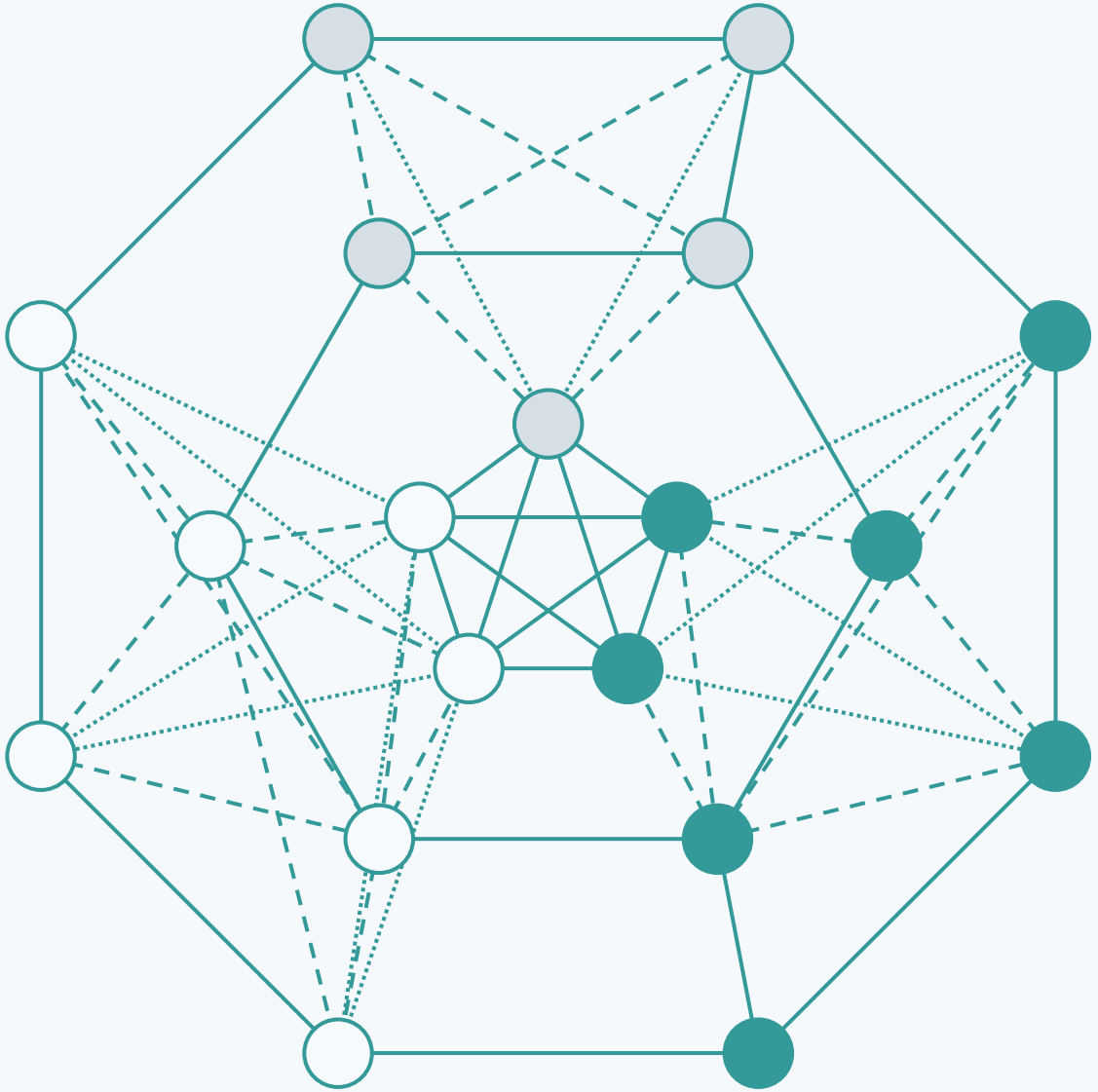
Bitcoin as a Base Layer

Bitcoin as a Base Layer

Example of a layered network

Key:

- Node
- Node
- Node
- Intra-layer
- Ancestor
- Core Ancestor

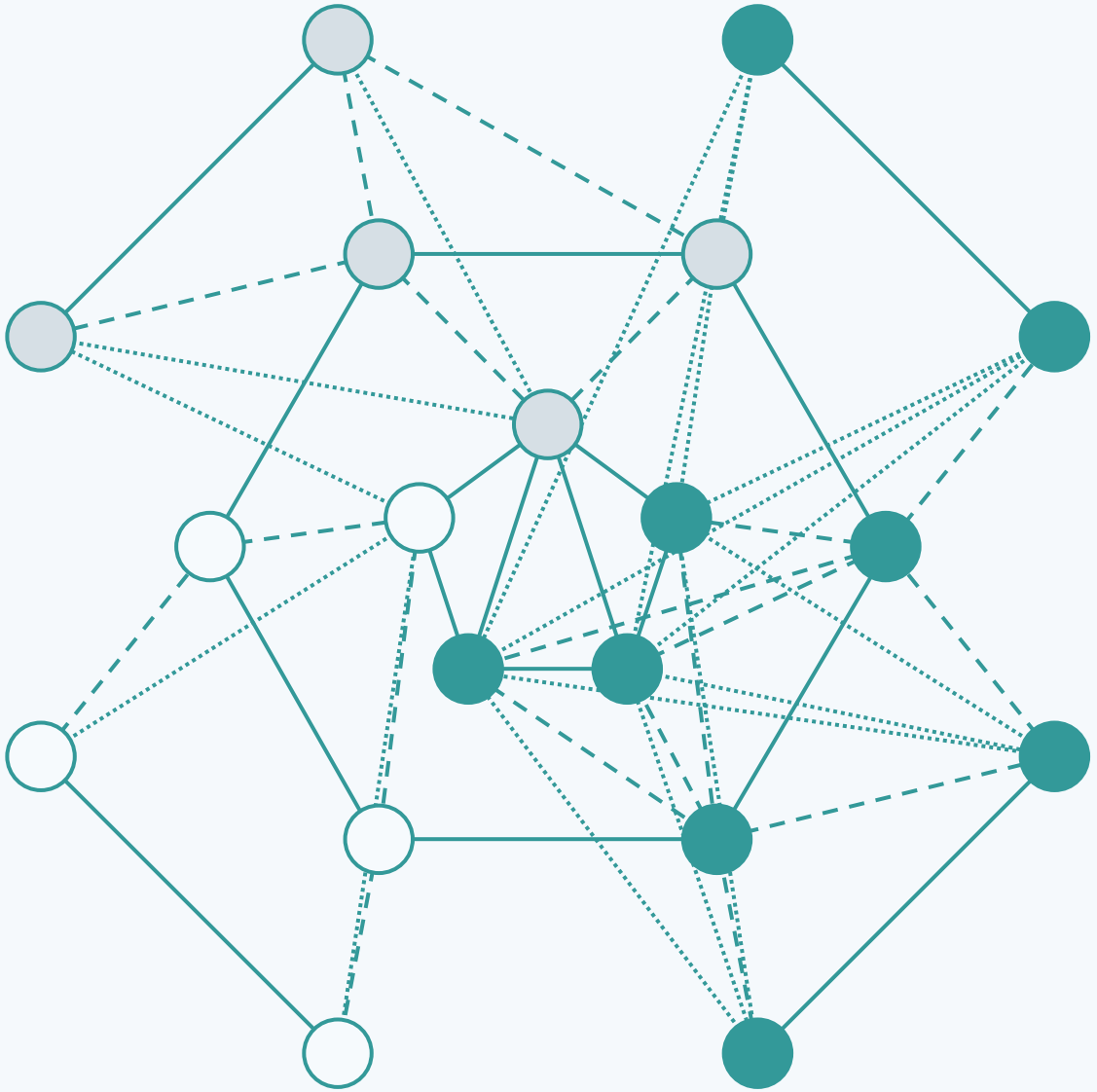


Bitcoin as a Base Layer

Example of a layered network

Key:

- Node
- Node
- Node
- Intra-layer
- Ancestor
- Core Ancestor

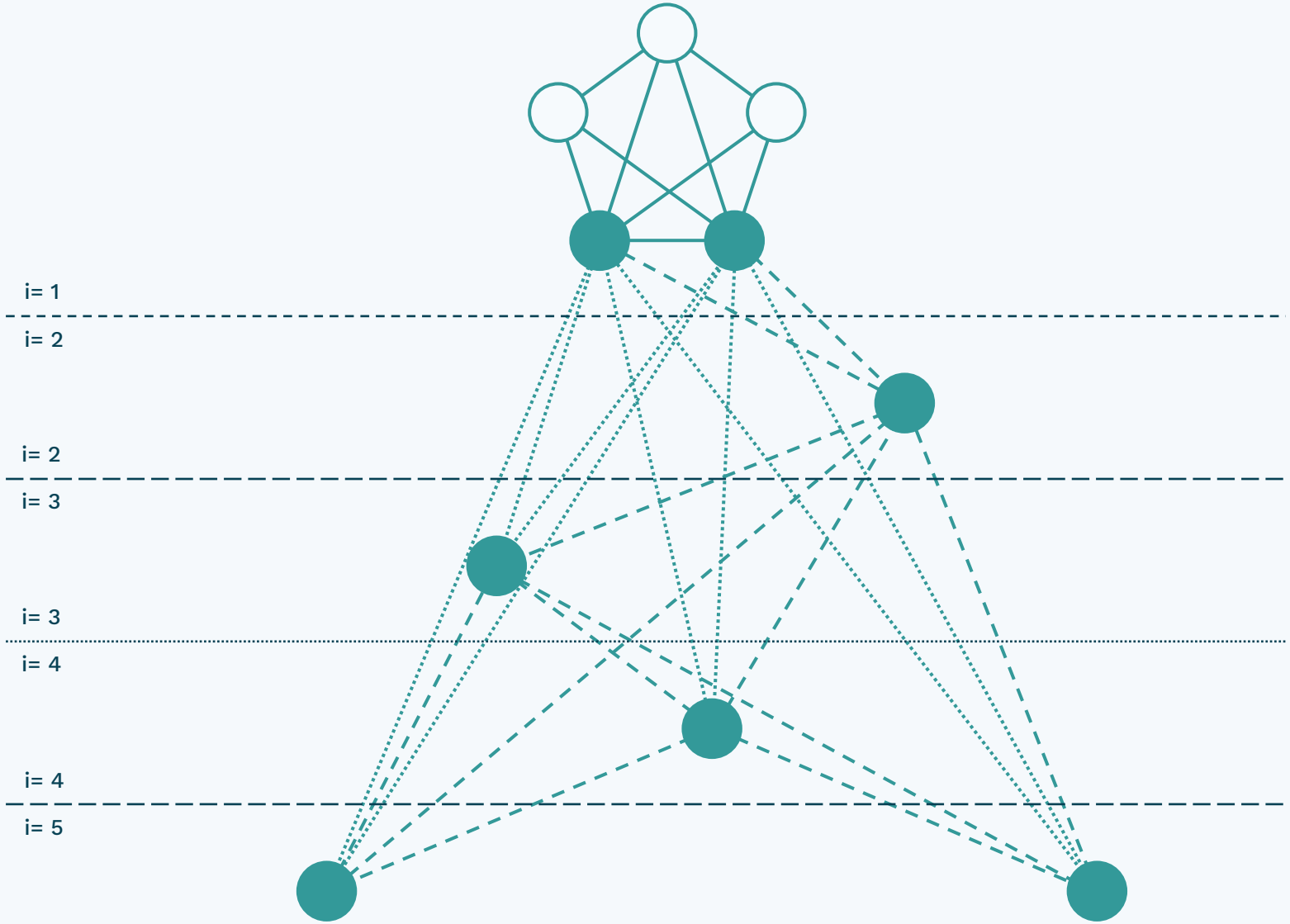


Bitcoin as a Base Layer

Example of a layered network

Key:

- Node
- Node
- Node
- Intra-layer
- Ancestor
- Core Ancestor



Controlling the Performance of a Contract

Licensing music and computer software

Blockchain technology can be used to facilitate an **inexpensive, secure, and transparent** procedure for the **purchase, integrity, and licensing of proprietary software**.

Consumers benefit from continuous **integrity** and **transparency**.

Software vendors can ensure that the **conditions** associated with their **license** are adhered to on a **continuous** and **cost-effective** basis.

The software vendors could enhance their **reputation** and increase their **trustworthiness** by being seen to be **transparent**.

The immutable record of transactions between the software vendors and consumers could be valuable for both **audits** and cases of **dispute resolution**.

Software Signing

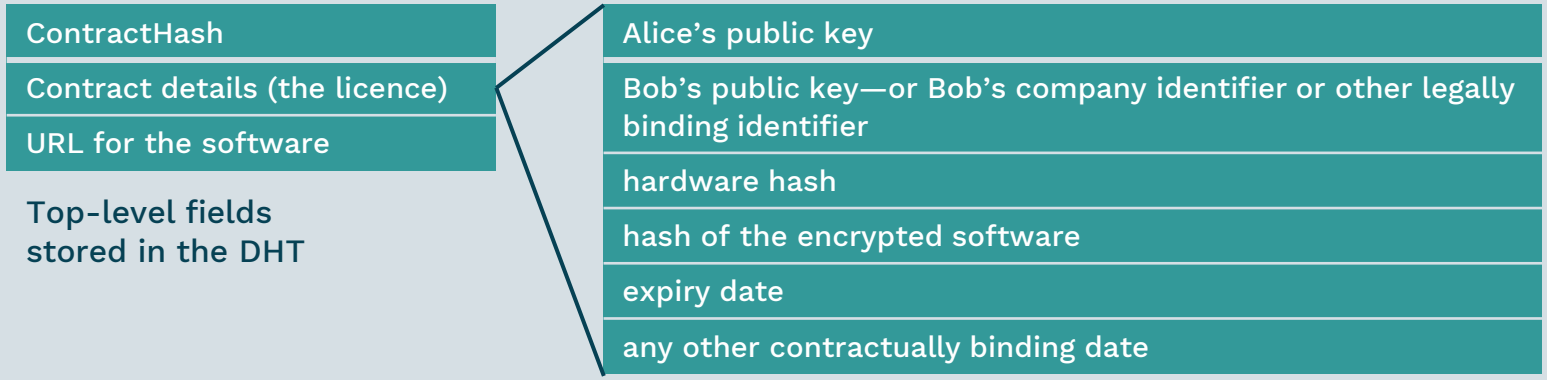
Example use case

Distributed hash table (DHT)
Stores Alice's public key and a hash of Alice's hardware string

Bob (software vendor)
Populates the DHT with software licence, a hash of the encrypted executable and the URL for the software download

Alice (end user)
Can sign transaction on the blockchain
Can query whether the licence is valid
Can revoke licence

Oracle (trusted third party)
Signs transactions when a user-provided expression evaluates to true
Employs a DHT
Can revoke licence



Top-level fields stored in the DHT

Contract details (the licence)

Field	Sub-field	Comments
Metadata1	ContractType	Coded value indicates type of contract.
	ContractPointer	IPv6 address identifying the DHT.
	ContractTypeData1	Format depends on value of ContractType. Padded with zeros.
Metadata2	ContractHash	RIPEMD-160(SHA256(actual contract file addressed by ContractPointer))
	ContractTypeData2	Format depends on value of ContractType. Padded with zeros.

Generic contract metadata format

Digital Artwork

Example use case

Benefits

- Security: Access control
- Monetisation: Automated payments

Token-protected objects can be secured using a third-party access token.

For instance, if Charlie creates digital artwork, he can issue an **access token** that decrypts it.

Charlie sells the access token to Alice.

Alice can send the access token to Bob.

As Alice no longer has the token, the system is configured to only allow authorised access, in accordance with the latest updates.

Both **access controls and payments services** can be directly integrated using IPv6 and Bitcoin.

THANK YOU

nChain

Data without question

Contact us

Switzerland
Grafenauweg 6,
6300 Zug,
Switzerland

United Kingdom
30 Market Place,
London W1W 8AP
United Kingdom

contact@nchain.com