

Usi non finanziari delle principali blockchain pubbliche

Valerio Vaccaro

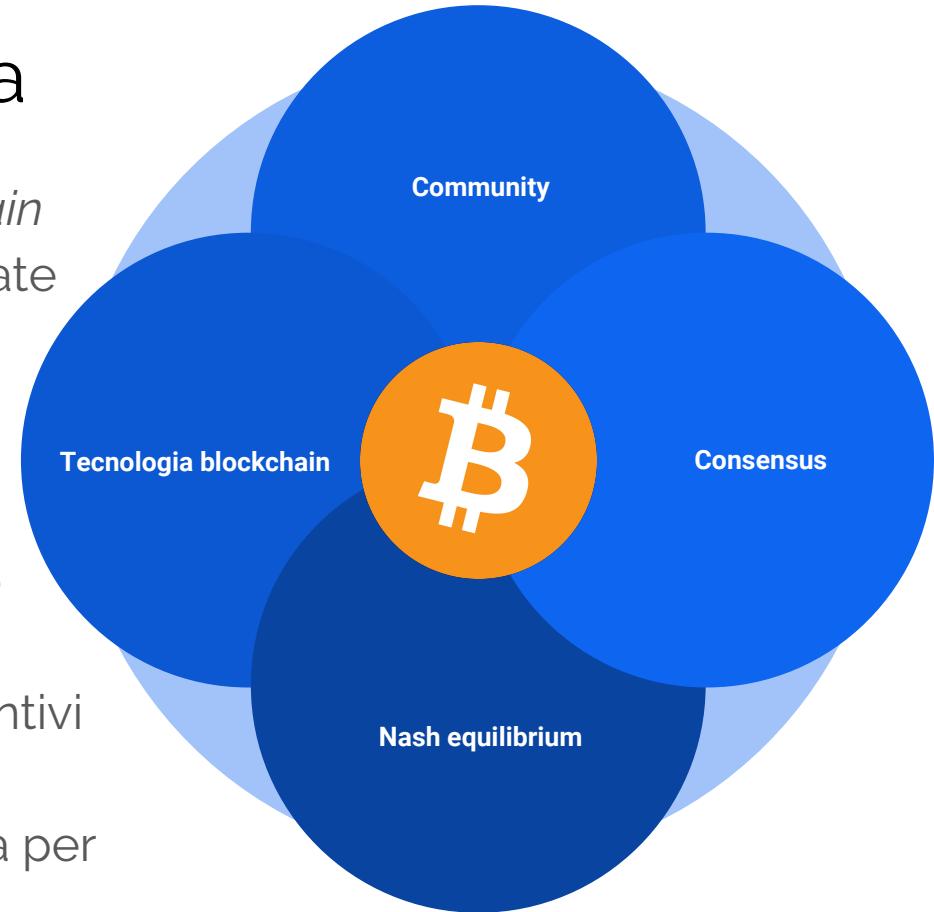
Blockchain for Business
Università di Venezia "Ca' Foscari"
20 febbraio 2019



Introduzione terminologica

Correntemente si usa il termine *blockchain* per identificare soluzioni complesse basate su:

- **community**: insieme di sviluppatori, utilizzatori, finanziatori, ...
- **consensus**: regole di funzionamento autosufficienti
- **nash equilibrium**: incentivi e disincentivi economici
- **tecnologia blockchain**: una struttura per salvare liste ordinate di eventi



AGENDA

COME È FATTA UNA BLOCKCHAIN?

USI FINANZIARI DI UNA BLOCKCHAIN

USI NON FINANZIARI DI UNA BLOCKCHAIN

PROOF OF PUBLICATION

TRUSTLESS TIMESTAMPING

SINGLE USE SEAL



Com'è fatta una blockchain?

Transazione - è il vettore per lo spostamento di valore su blockchain.

- N input (necessariamente transazioni non spese precedenti)
- M output
- Script di autorizzazione e firma (programmabilità)
- Crittografia

version	01 00 00 00
input count	01
input	previous output hash (reversed) 48 4d 40 d4 5b 9e a0 d6 52 fc a8 25 8a b7 ca a4 25 41 eb 52 97 58 57 f9 6f b5 0c d7 32 c8 b4 81
	previous output index 00 00 00 00
	script length
	scriptSig script containing signature
	sequence ff ff ff ff
output count	01
output	value 62 64 01 00 00 00 00 00
	script length
	scriptPubKey script containing destination address
block lock time	00 00 00 00

Com'è fatta una blockchain?

version	02000000
previous block hash (reversed)	17975b97c18ed1f7e255adf297599b55 330edab87803c817010000000000000000
Merkle root (reversed)	8a97295a2747b4f1a0b3948df3990344 c0e19fa6b2b92b3a19c8e6badc141787
timestamp	358b0553
bits	535f0119
nonce	48750833
transaction count	63
coinbase transaction	
transaction	
...	

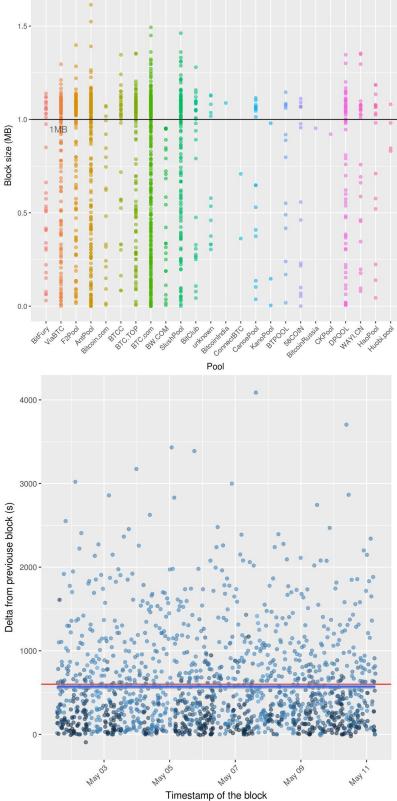
Blocco - contiene le transazioni da confermare

- Può essere scritto sulla blockchain da chi risolve un certo problema matematico computazionalmente rilevante (PoW)
- Contiene un premio per chi trova la soluzione
- La difficoltà è adattata in funzione delle condizioni della rete

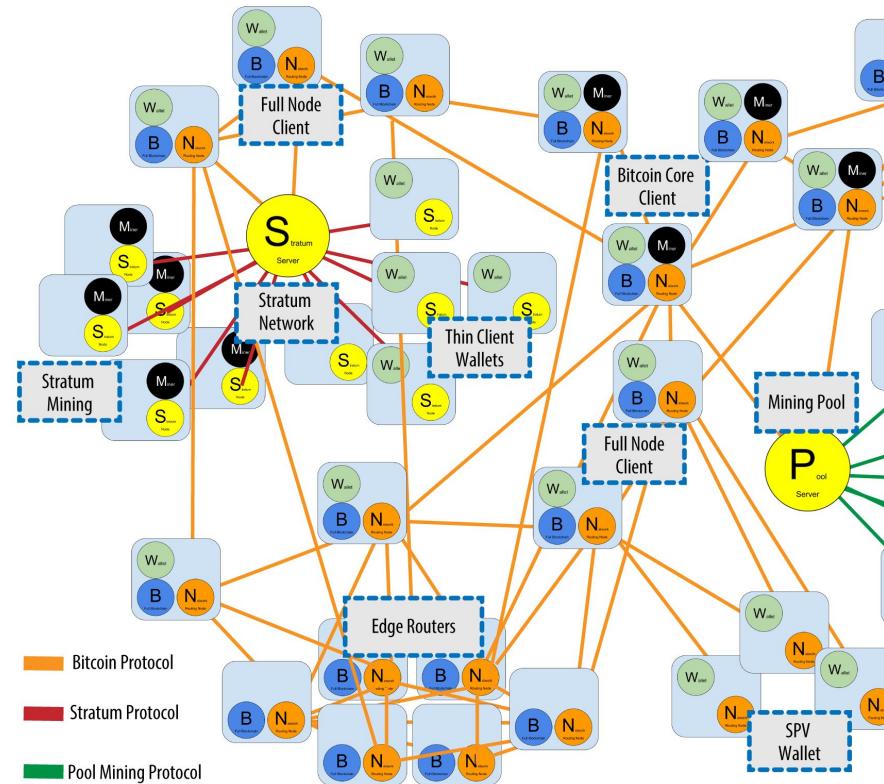
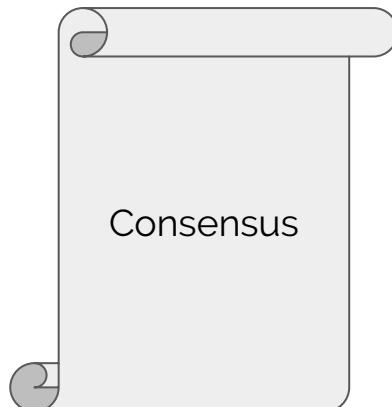
Com'è fatta una blockchain?

Distribution of blocks per pool - 10 days analyzed.

2015-05-11 - vaccaro.vaccaro.it



Mining e rete



Usi finanziari

- Strumenti di pagamento
- Store of value
- Denaro programmabile
- Creazione e movimentazione di asset (Token, coloured coins, ...)
- Base per 2nd layer e sidechain
- Simulazione e realizzazione di sistemi complessi di pagamento (atomic swap, ...)



Usi non finanziari

Effettuazione di una transazione che muove informazioni oltre che valore*:

- Proof of Publication
- Trustless Timestamping
- Single use seal



* spesso non è desiderato muovere informazioni importanti e valore in una medesima transazione, occorre comunque pagare le fee di transazione.

Proof of Publication

Scrivo un testo direttamente sulla blockchain (in chiaro o meno)

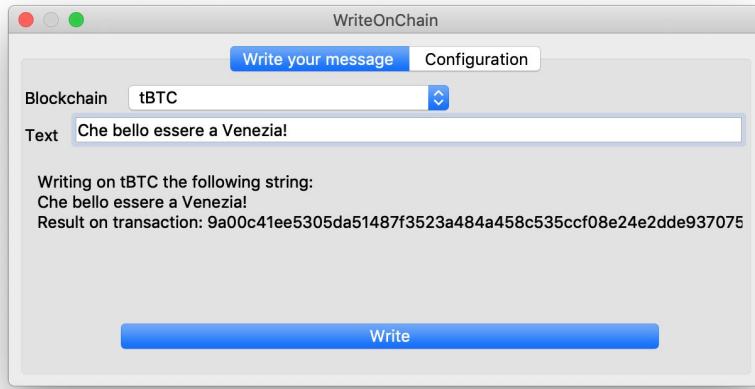
Uso di un comando specifico per la scrittura di un testo (nel caso di Bitcoin derivati OP_RETURN).

Massimo 80 caratteri per transazione (nel caso di Bitcoin)

Posso provare:

- proprietà del testo (la transazione è firmata con la mia chiave privata),
- data di pubblicazione,
- unicità,
- sequenzialità.

Proof of publication



<https://gitlab.com/valerio-vaccaro/writeonchain>

Non scala!

9a00c41ee5305da51487f3523a484a458c535ccf08e24e2dde937075fe10b0a6

DETTAGLI —

#0	e07b5e83eda089c10f197715021c8a3d64b502365	0.01000742 tBTC
a1cddb042444fa0c619f22a:2		

OUTPOINT	e07b5e83eda089c10f197715021c8a3d64b502365 19f22a:2
SCRIPTSIG (ASM)	OP_PUSHBYTES_22_0014b9311baf9e7dd3b006915499c5793eb88ccca302
SCRIPTSIG (HEX)	160014b9311baf9e7dd3b006915499c5793eb88ccca302
WITNESS	304402206e46d2f459b7dc45a08b7a68d67037b282196008eb37dc136b4c305714583b8d022005dd51bf1682ad42b863cb75d1fb5c5d7dbb8753bbff4f5e4fecada3a39f3bf701_035a88c776da2750a6573daad1e5e414f57c57bc77f6f9fd930b73d73e31fdbfd9
NSEQUENCE	0xffffffff
PREVIOUS OUTPUT SCRIPT	OP_HASH160 OP_PUSHBYTES_20 16 7ce97b46f6bd56d080488c57a0a7
SCRIPTPUBKEY (ASM)	OP_HASH160 OP_PUSHBYTES_20 06 1dc1608ca26c28f0634989dbe3c06bc7bb10c9_OP_EQUAL
SCRIPTPUBKEY (HEX)	a914061dc1608ca26c28f0634989dbe3c6bcb7bb10c9

OP_RETURN DATA: Che bello essere a Venezia!

OP_RETURN: 0 tBTC

TIPOLOGIA: OP_RETURN

SCRIPTPUBKEY (ASM): OP_RETURN OP_PUSHBYTES_27 436 8652062656c6c6f20657373657265 206120565656e657a696121

SCRIPTPUBKEY (HEX): 6a1b4368652062656c6c6f20657373657265 206120565656e657a696121

OP_RETURN DATA: Che bello essere a Venezia!

TIPOLOGIA: P2SH

SCRIPTPUBKEY (ASM): OP_HASH160 OP_PUSHBYTES_20 06 1dc1608ca26c28f0634989dbe3c06bc7bb10c9_OP_EQUAL

SCRIPTPUBKEY (HEX): a914061dc1608ca26c28f0634989dbe3c6bcb7bb10c9

Timestamping



Il timestamping è l'atto di dare una data ad un documento, il primo e più comune esempio è il **timbro postale**

- Alice scrive a Bob, la data della comunicazione è apposta da terzi, né Alice né Bob possono contraffarla facilmente
- Attenzione: deve essere apposto sul documento e non sulla busta!

Per importanti documenti deve essere posta da pubblico ufficiale, il notaio

- Ad esempio per il rogito della casa

E per i documenti digitali?

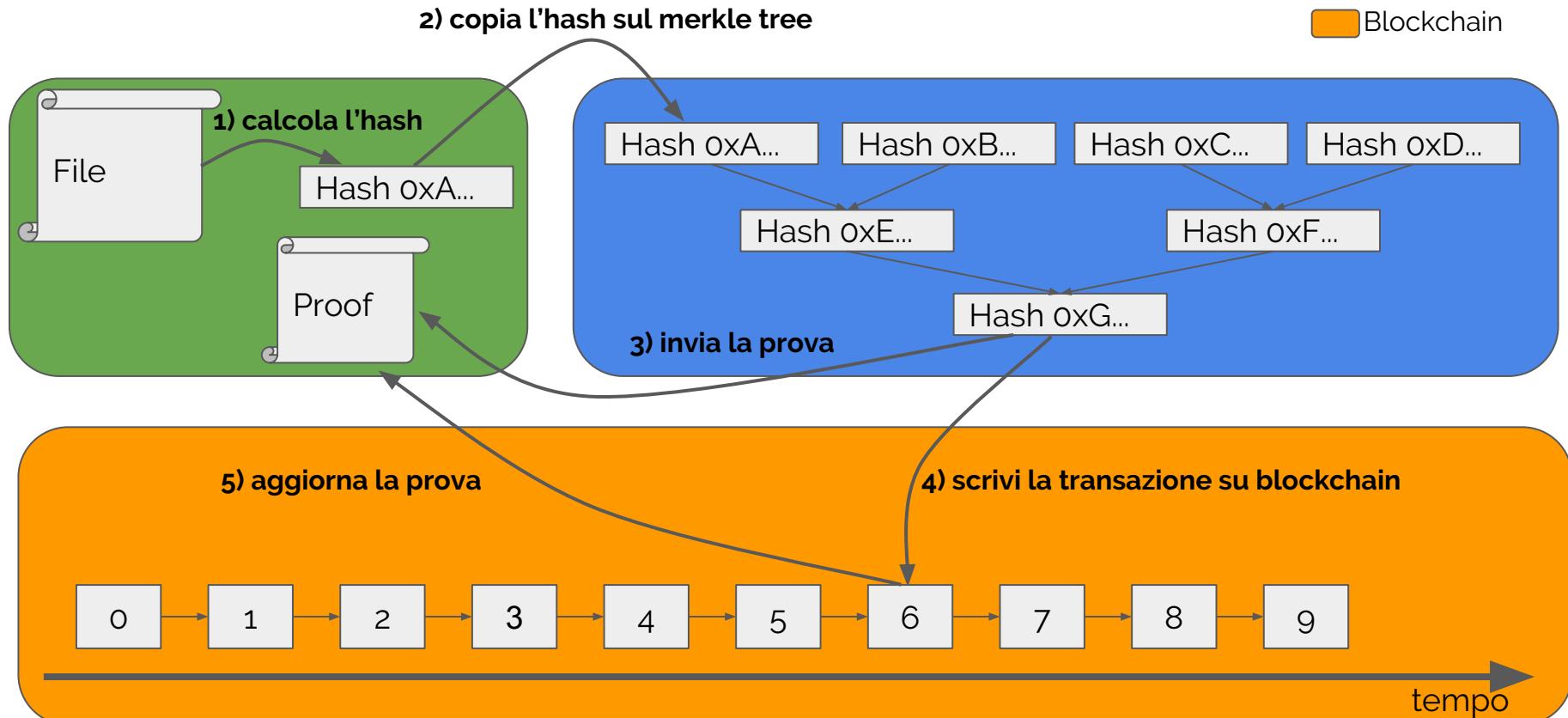
Timestamping: OpenTimestamps

OpenTimestamps è un protocollo standard per la notarizzazione di qualsiasi informazione digitale con le seguenti caratteristiche:

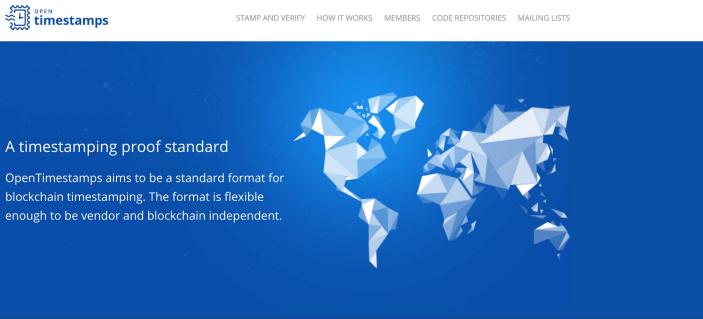
- **Trust** - OTS usa la blockchain bitcoin risultando decentralizzato, pubblicamente verificabile e rimuovendo la necessità di una certification authority,
- **Cost** - OTS può condensare un numero illimitato di timestamps in una singola transazione,
- **Convenience** - OTS genera un timestamp verificabile direttamente da una terza parte in un solo secondo.

Timestamping: OpenTimestamps

Client
Calendar server
Blockchain

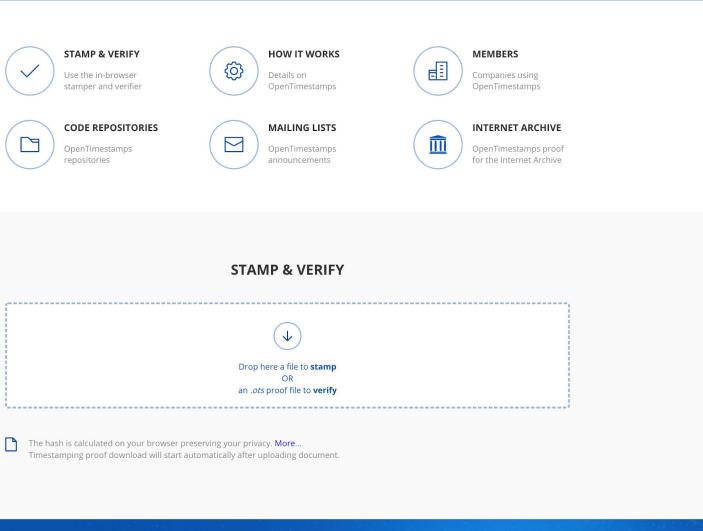


Timestamping: OpenTimestamps



E' possibile timestampare un documento direttamente dal sito
<https://opentimestamps.org/>

- Trascina il file sullo spazio **STAMP&VERIFY**
- Il sito calcolerà automaticamente l'hash e manderà una richiesta ai calendar server
- Il risposta verrà fornito un file .ots contenete la prova (parziale) del timestamping



Timestamping: OpenTimestamps

Tutto il software di OpenTimestamps è open source e già disponibile su [github](#).

<https://github.com/opentimestamps>

Client e Server sono disponibili per installazioni personali e sono già sviluppate le librerie per:

- Python
 - Javascript
 - Java
 - Rust
 - Node-RED

Esistono dei **server pubblici** e **gratuiti** che consentono di essere già operativi! Da ora!

365 timestamps

Timestamp of **58061bedef2a**

Document digest (sha256):

58061bedef2a2d84c72cd0f774fc4a0903d27c077e011eaa8838e7c9c786f6

DOWNLOAD THIS TIMESTAMP

Block Address	Pending	Unknown Address	Comments	Fork
aspend7af066...)		58061bedef2a2d84c72cd0f774fc4a0903d27c077e011eaa8838e7c9c786f6 c917e0b0094513...		
sha21461		ad8f37321ef3d27817817312e15a6e56a71930951aae9113f1e5d77d91		
	Fork	Fork in 2 paths		
append979a3...)		ad448f57219a2478f12537a132e15a6e56a71930951aae9113f1e5d7722987 c917e0b0094513...		
sha2150		197fc5a220a17c0320a1695a13e0f92917f0c9917244231a69101c5a0		
append979d1...)		19a19c9c2a21917c02a1695a13e0f92917f0c9917244231a69101c5a0 c917e0b0094513...		
sha2160		1a921273217e162d61446e25920d7573a2910e2a1507a014548ed		
append9408d...)		3a62f273217e162d61446e25920d7573a2910e2a1507a014548ed c917e0b0094513...		
sha2150		634461598360e7341997a0a3521071a087a21738305ea8a986		
prepend97071d...)		13371e1e34a6c0983e0873464937e0cd5207172aM50a74037305eccc9f66		
append9817a...)		13371e1e34a6c0983e0873464937e0cd5207172aM50a74037305eccc9f66 c917e0b0094513...		
	Pending Attestation	Verifying attestation: server https://alice.tbc.calendar.openTimestamp.org		
sha2150		1a1755c14812192682a16949051604951892a75737d19c5f5d12ba		
append9407d...)		1a1755c14812192682a16949051604951892a75737d19c5f5d12ba c917e0b0094513...		
sha2150		1a45f5c4542c05e057a673b89a927b7844677989938179a87e77125		
prepend97044c...)		1a45f5c4542c05e057a673b89a927b7844677989938179a87e77125 c917e0b0094513...		
sha2150		1a4494eef098819616a0f5e2250d8467635036d573a4666991 c917e0b0094513...		
append9344c...)		1a4494eef098819616a0f5e2250d8467635036d573a4666991 c917e0b0094513...		
sha2150		1b3030451446179786559916e16d1a232909696451671e192504070		
append97e28...)		2a2130451446179786559916e16d1a232909696451671e192504070 c917e0b0094513...		
sha2150		1b3a79d3f0325fb5b2164d47e17e22448919f93a4412e2a20a2e3		
append97983...)		1b3a79d3f0325fb5b2164d47e17e22448919f93a4412e2a20a2e3 c917e0b0094513...		
sha2150		1c2025a0e147491a5280e13c51e94d16617950a9939b134451025		
prepend97057...)		1c2025a0e147491a5280e13c51e94d16617950a9939b134451025 c917e0b0094513...		
sha2150		1c2025a0e147491a5280e13c51e94d16617950a9939b134451025		
prepend94589...)		1c44856846e0ef2999771a45334a615c1bd12a410712b6ff0404e3f7209220000 c917e0b0094513...		
sha2150		1c44856846e0ef2999771a45334a615c1bd12a410712b6ff0404e3f7209220000 c917e0b0094513...		

Single use seal*

Sfrutta le proprietà di singola spesa delle transazioni delle blockchain per “bloccare” l’aggiornamento di un contenuto*.

Il contenuto risulta valido ed aggiornato solo se esiste una “storia” plausibile dello stato del contenuto e se l’ultimo step contiene una transazione non ancora spesa.

La modifica dello stato può essere effettuata solo da chi possiede le chiavi private capaci di spendere l’ultima transazione.

All’atto dell’aggiornamento del contenuto occorre esplicitare chi potrà effettuare la modifica successiva (inserendo i riferimenti di una transazione non spesa).

* E’ una fortissima semplificazione di un concetto molto più ampio e complesso.
<https://petertodd.org/2016/commitments-and-single-use-seals>

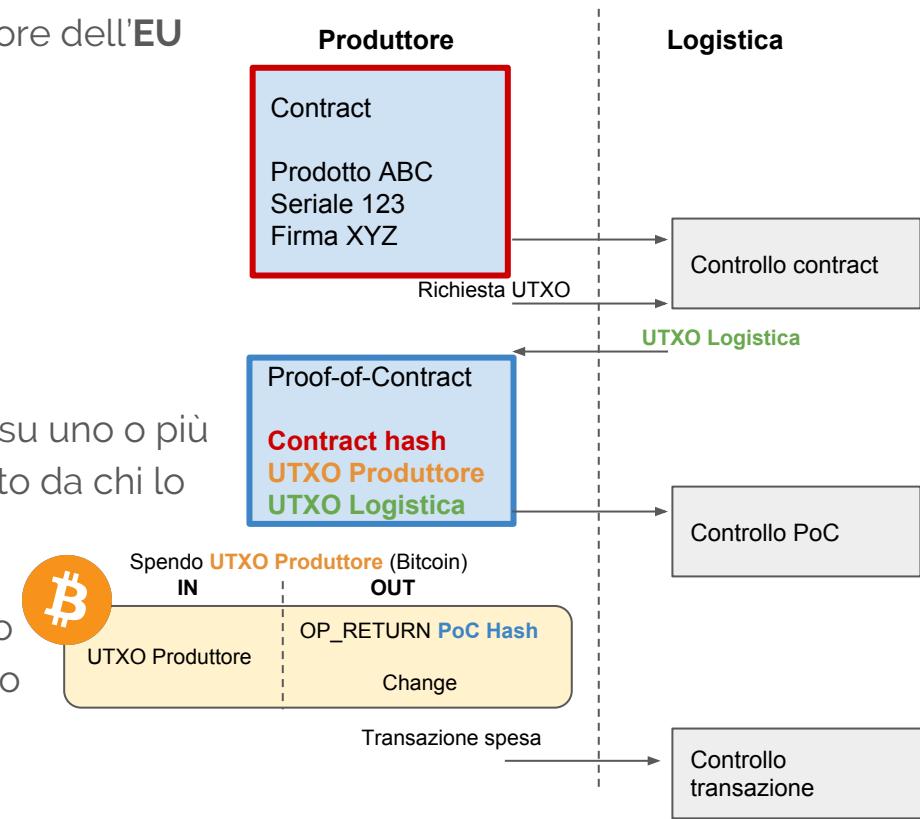
Single use seal: Eonpass

Protocollo per l'anticontraffazione di prodotto vincitore dell'**EU Blockathon 2018**

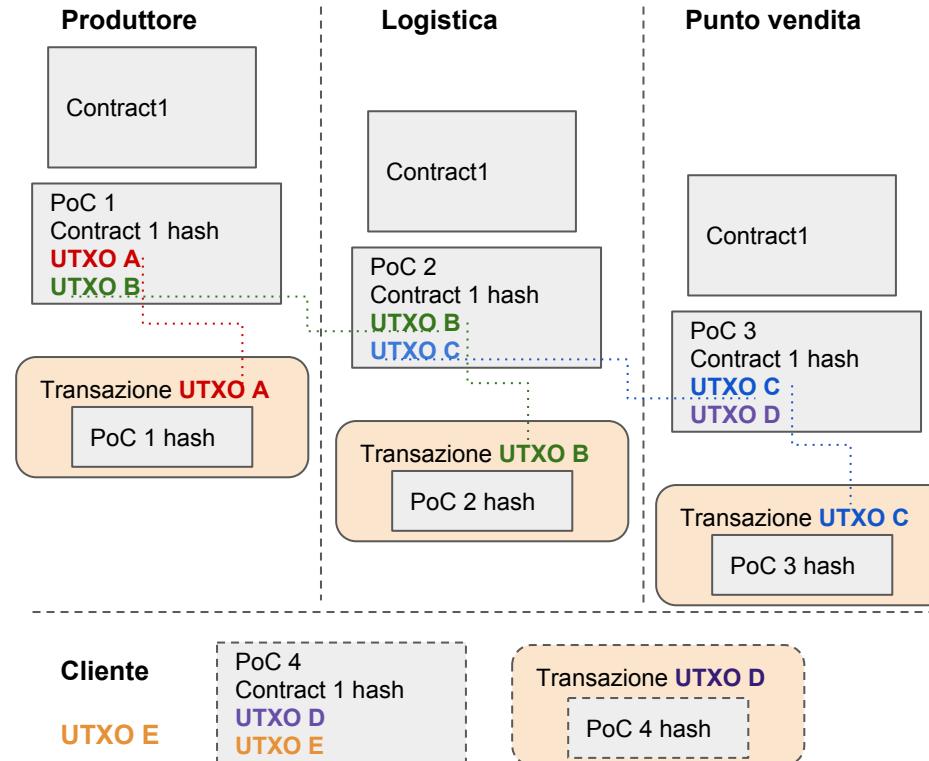
- ID agnostic
- Implementazioni open-source
- Multi-blockchain (anche contemporanee)
- GDPR friendly

Una contratto è un file che contiene le informazioni su uno o più prodotti non necessariamente omogenei ed è firmato da chi lo redige.

Un Proof-of-contract è l'atto con cui viene dichiarato l'aggiornamento di un contratto e l'eventuale cambio di permessi.



Single use seal: Eonpass



Il controllo può essere fatto navigando tutta la storia, il PoC 3 è valido se:

- esiste una storia valida da PoC 1 a PoC 3
- UTXO A, UTXO B e UTXO C risultano spese con un commitment a PoC 1, PoC 2 e PoC 3 rispettivamente
- UTXO D è non spesa

L'acquisto da un Cliente è confermato dalla creazione di un PoC 4 con la transazione non spesa del cliente (UTXO E) e dalla spesa di UTXO D con un commitment al PoC 4 creato.

Sono previsti meccanismi di split e merge dei contratti.

Fonti

- **Mastering Bitcoin** - <https://github.com/bitcoinbook/bitcoinbook>
- **Opentimestamps** -
<https://petertodd.org/2016/opentimestamps-announcement>
- **Single use seal** -
<https://petertodd.org/2016/commitments-and-single-use-seals>
- **Blockchain and (I)IoT** -
<https://medium.com/@valerio.vaccaro/blockchain-and-i-iot-agec599e0df1>
- **Eonpass** - <https://gitlab.com/Eonpass/specs/>

The end

Valerio Vaccaro

email: valerio.vaccaro@gmail.com

twitter: [@Tulipan81](https://twitter.com/Tulipan81)

linkedin: <https://www.linkedin.com/in/valeriovaccaro/>

github: <https://github.com/valerio-vaccaro>

gitlab: <https://gitlab.com/valerio-vaccaro>