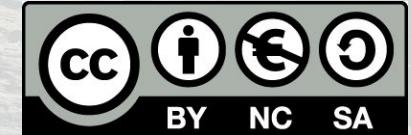


# Usi non finanziari delle principali blockchain pubbliche

Valerio Vaccaro

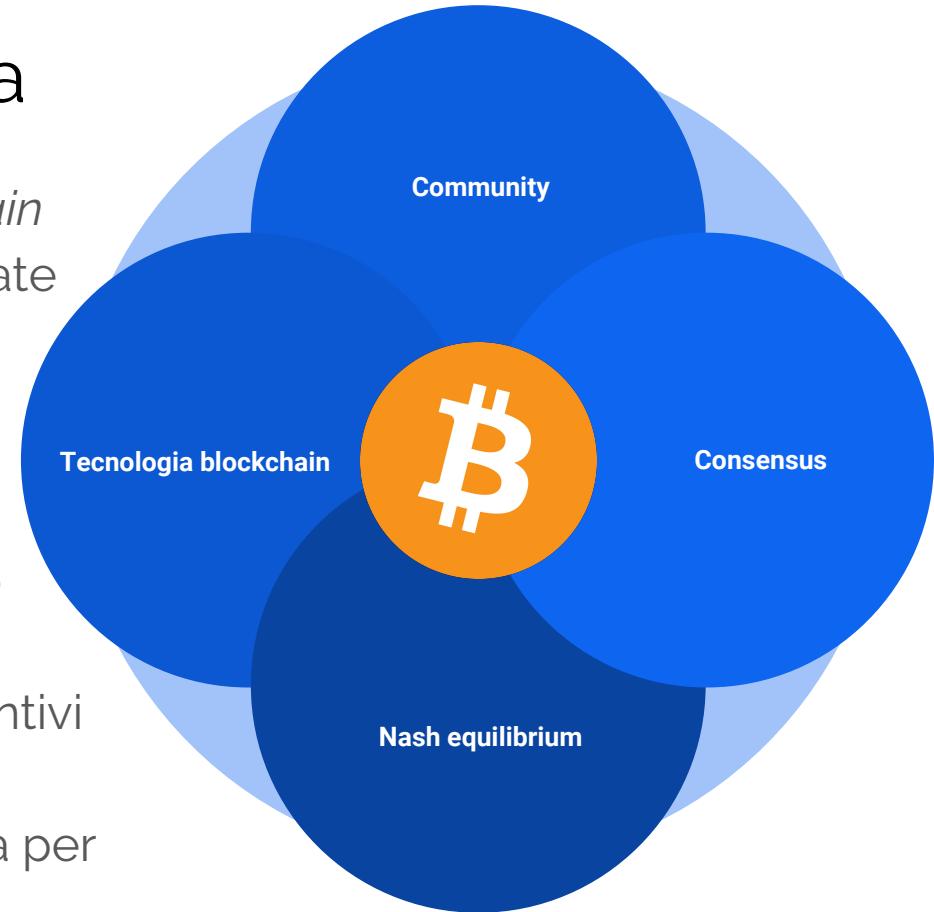
Blockchain for Business  
Università di Venezia "Ca' Foscari"  
20 febbraio 2019



# 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	
...	

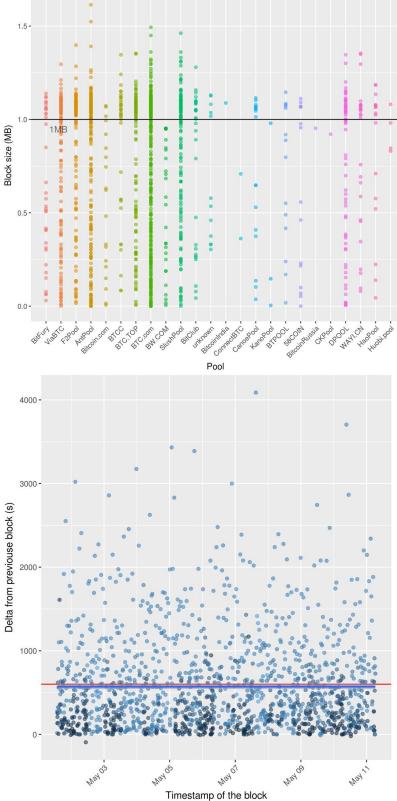
Blocco - contiene le transazioni da confermare

- 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

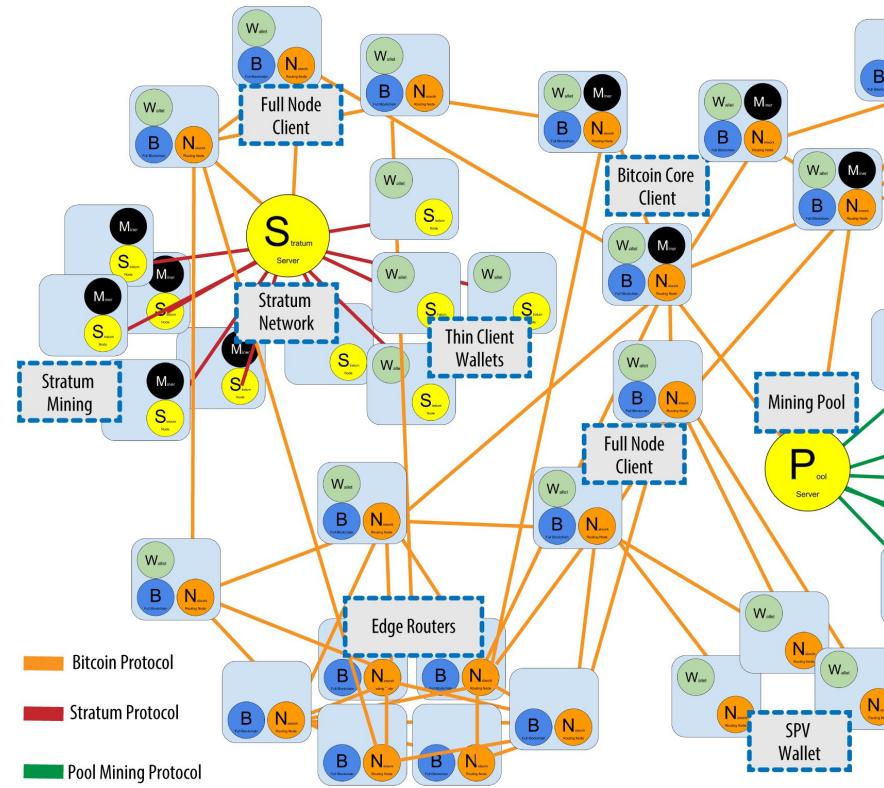
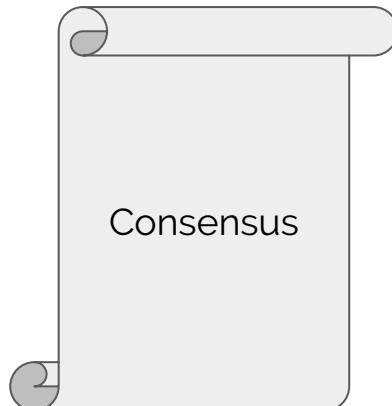
# Com'è fatta una blockchain?

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)

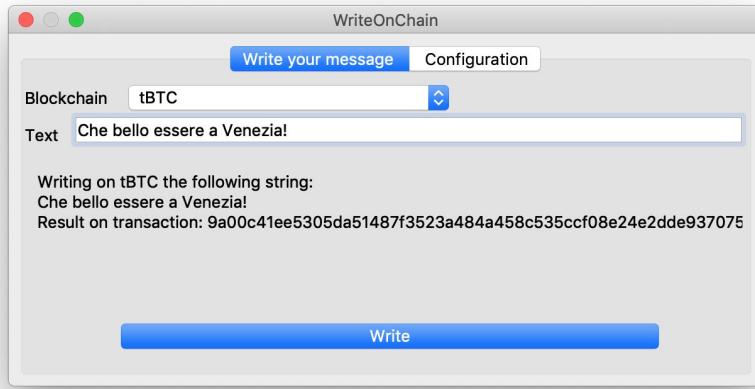
Uso di un comando specifico per la scrittura di un testo (nel caso di Bitcoin derivati OP\_RETURN).

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!

The screenshot shows a blockchain transaction details page. The transaction ID is 9a00c41ee5305da51487f3523a484a458c535ccf08e24e2dde937075fe10b0a6. The transaction is a simple P2WPKH (OP\_RETURN) output with 0 tBTC sent to the script `OP_RETURN`. The `OP_RETURN DATA` field contains the message "Che bello essere a Venezia!". The transaction also includes a witness script and previous output details.

OUTPUT	OP_RETURN
#0 <code>e07b5e83eda089c10f197715021c8a3d64b502365</code> 0.01000742 tBTC <code>a1cdcb042444fa0c619f22a:2</code>	0 tBTC
SCRIPTSIG (ASM)	<code>OP_PUSHBYTES_22_0014b9311baf9e7dd3b00691549</code> <code>c5793eb88ccccaa302</code>
SCRIPTSIG (HEX)	160014b9311baf9e7dd3b00691549 9c5793eb88ccccaa302
WITNESS	304402206e46d2f459b7dc45a08b7 a68d67037b282196008eb37dc136b 4c305714583b8d022005dd51bf168 2ad42b863cb75d1fb5cfd7dbb8753 bbff4f5e4fecaaed3a39f3bf701_03 5a88c776da2750a6573daad1e5e41 4f57c57bc77f6f9fd930b73d73e31 fdbfd9
NSEQUENCE	0xffffffff
PREVIOUS OUTPUT SCRIPT	<code>OP_HASH160 OP_PUSHBYTES_20_16</code> <code>7ce97hb646fb956d080488c57a0a7</code>
SCRIPTPUBKEY (ASM)	<code>OP_HASH160 OP_PUSHBYTES_20_06</code> <code>1dc1608ca26c28f0634989dbe3c06</code> <code>bc7bb10c9 OP_EQUAL</code>
SCRIPTPUBKEY (HEX)	a914061dc1608ca26c28f0634989d be3c6bcb7bb10c9



# 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 nè 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?

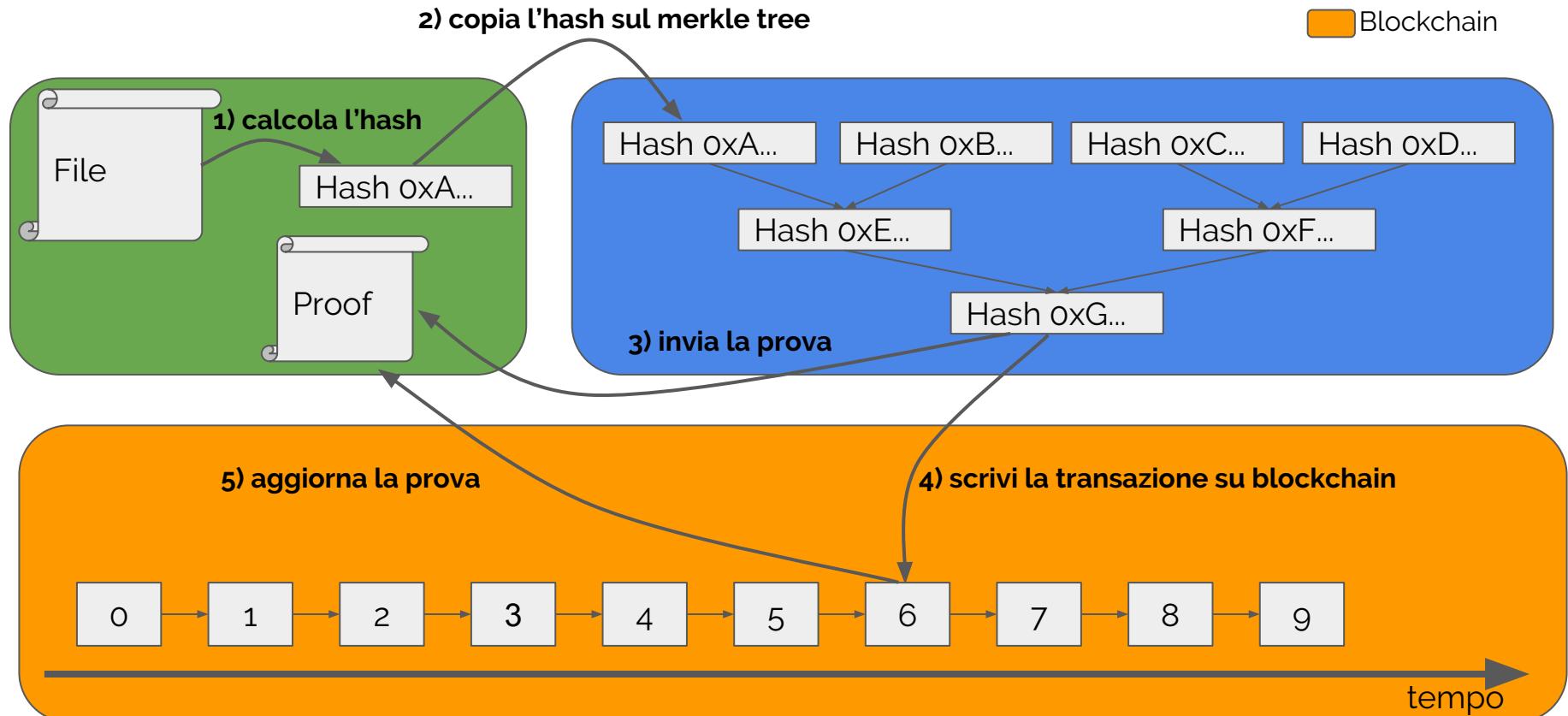
# Timestamping: OpenTimestamps

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

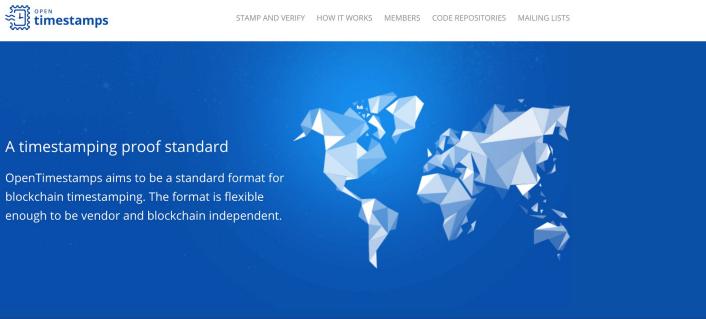
- **Trust** - OTS usa la blockchain bitcoin risultando decentralizzato, pubblicamente verificabile e rimuovendo la necessità di una certification authority,
- **Cost** - OTS può condensare un numero illimitato di timestamps in una singola transazione,
- **Convenience** - OTS genera un timestamp verificabile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


timestamps
  
**Timestamp of** 58061bedef2a
  
 Document digest sha256:  
58061bedef2a64c72cdff7f4cfa0903d27c077fe013e48838ce79c768df
  
[DOWNLOAD THIS TIMESTAMP](#)

# 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>

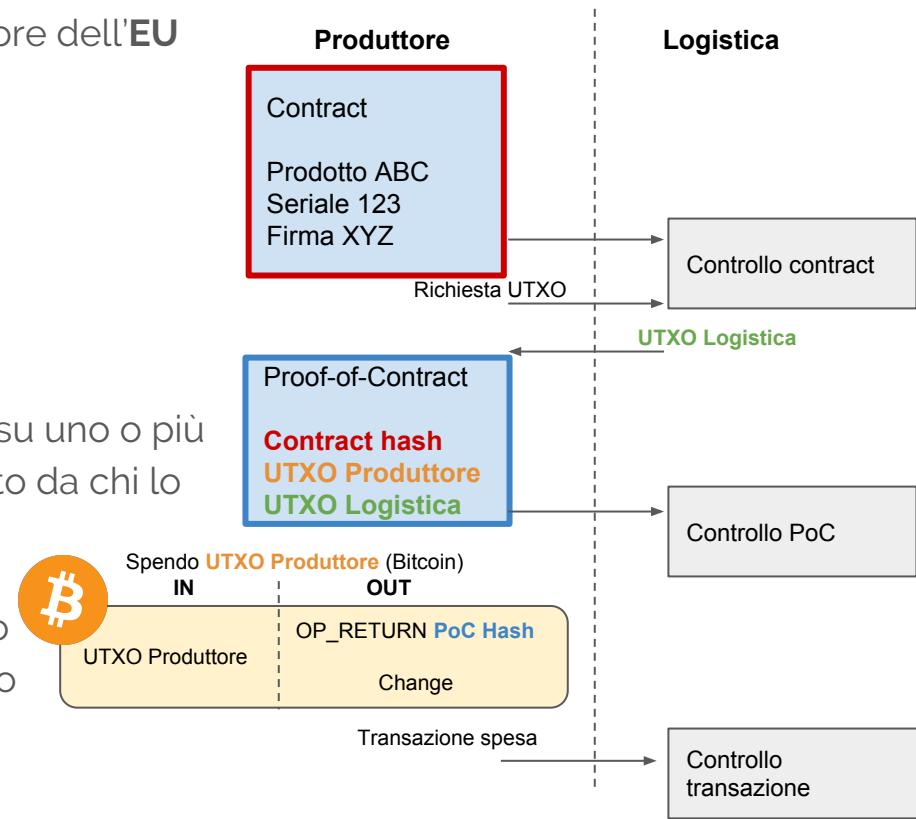
# Single use seal: Eonpass

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

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

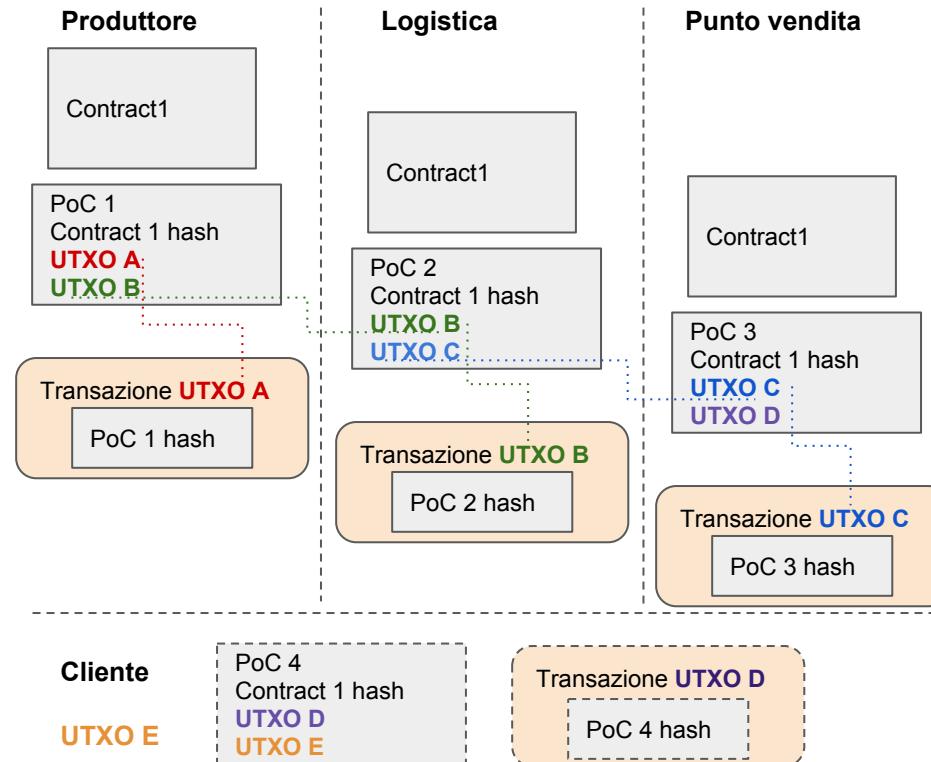
Una contratto è un file che contiene le informazioni su uno o più prodotti non necessariamente omogenei ed è firmato da chi lo redige.

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



\* <https://gitlab.com/Eonpass/specs>

# 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

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

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

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