Tecnologie Cloud e Mobile

Lez. 14

BlockChain

Giuseppe Psaila

Università di Bergamo

giuseppe.psaila@unibg.it

Sistemi Centralizzati e Fiducia

- Prendiamo come esempio una supply chain
- Aziende diverse devono cooperare per tenere viva la catena di fornitura di
  - Materie prime
  - Semi-lavorati
  - Componenti
- Al fine di consentire la produzione del prodotto finale

Sistemi Centralizzati e Fiducia

- Soluzione centralizzata:
- Un’azienda, tipicamente quella che produce il prodotto finale (per esempio, un produttore di auto) mette a disposizione il suo sistema informativo per gestire l’intero flusso
- Le altre aziende devono «fidarsi» che il sistema centralizzato funzioni correttamente

Intermediario e Fiducia

- Altro contesto: Scambio di Denaro
- Le banche si scambiano denaro
- I clienti delle banche si appoggiano alle banche stesse come intermediari, per poter inviare/ricevere denaro
- Le regole nazionali proteggono il cliente, che altrimenti sarebbe soggetto a possibili frodi da parte della banca (commissioni esagerate)

Diritto di Proprietà e Fiducia

- Per vendere/acquistare un immobile non basta firmare un contratto tra le parti
- Serve un notaio, che svolge due attività fondamentali:
  - Mantiene un registro (registro notarile, in Inglese Ledger) dell'atto di compra/vendita
  - Verifica nei registri degli atti notai se chi vende ha effettivamente la proprietà del bene venduto

Il Ruolo degli Intermediari

- Gli intermediari hanno quindi un ruolo importante:
- Svolgono operazioni che il cliente non sarebbe in grado di svolgere
- Nelle situazioni più critiche, garantiscono la regolarità della procedura
- Ma se l’intermediario è d’accordo con una delle due parti? Per esempio, il notaio è un falso notaio?

Fiducia negli Intermediari

- Sorge, quindi, un problema di fiducia negli intermediari
- Ma gli intermediari sono proprio necessari?
- I loro sistemi informatici forniscono gli adeguati livelli di garanzia che i dati non verranno persi o modificati, per errore o in modo fraudolento?

Immutabilità e Condivisione

- La chiave per risolvere i problemi presentati prima è costituita da due concetti
- **Immutabilità**
- **Condivisione**

Immutabilità e Condivisione

- **Immutabilità**: la storia dell'intero processo che si vuole rendere trasparente deve essere archiviata in modo immutabile, cioè non può essere manipolata né volontariamente né accidentalmente
- Serve un Ledger, un registro dove si annotano tutti i passaggi in modo immutabile

Immutabilità e Condivisione

- **Condivisione**: il registro deve essere condiviso, cioè devono esistere molte copie continuamente allineate e gestite da sistemi in «competizione» tra loro
- L’autorizzazione a cambiare i dati, cioè a registrare una nuova versione di un certo processo (mantenendo la precedente) deve essere data da molti, non da pochi (consenso condiviso)

Le Monete Virtuali

- Una moneta virtuale (o valuta virtuale, Virtual Currency in Inglese) è una moneta che non esiste
- O meglio, non esiste fisicamente, non ci sono banconote o monete
- Esiste perché c’è un sistema informatico che la gestisce

Le Monete Virtuali

- Questo sistema informatico ha diversi compiti:
- Certificare il possesso di importi della valuta da parte degli utenti
- Gestire le transazioni, cioè lo scambio/trasferimento di importi da un utente ad un altro
- Evitare le frodi
- Evitare la perdita dei dati

Le Monete Virtuali

- Le monete virtuali sono comunque legate alle monete reali
- Si comprano monete virtuali pagandole con monete reali
- Quindi, se il sistema di gestione perde i dati di possesso
- Gli utenti perdono soldi veri!!!

Le Monete Virtuali

- Il sistema di gestione fa da intermediario
- Ma se il sistema di gestione è centralizzato, per quanti sforzi si facciano, è vulnerabile e può perdere in parte o totalmente i dati
- Per questa ragione, fino al 2009, le monete virtuali hanno avuto scarso successo: non c’era fiducia

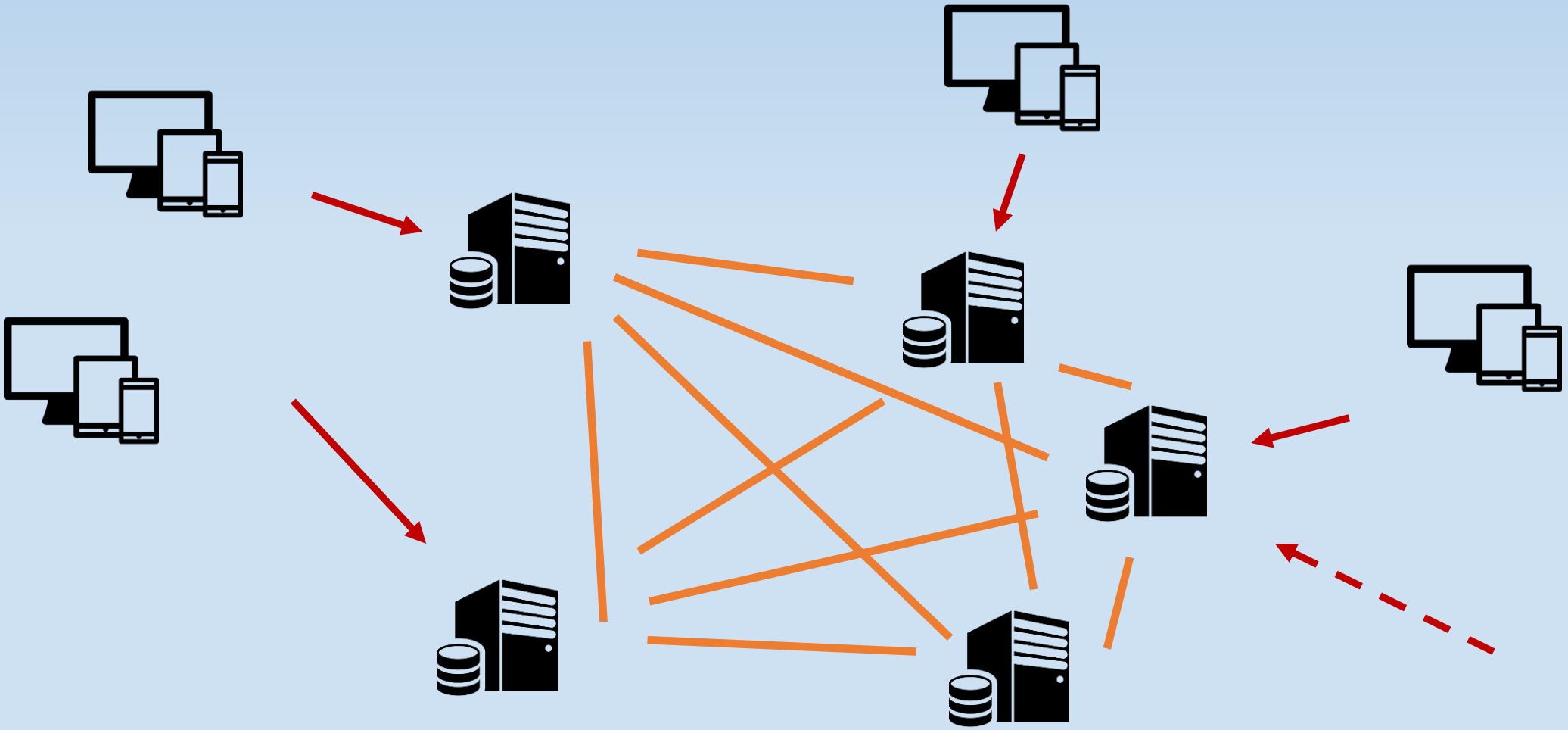
bitcoin

- Nel 2010 avviene un evento dirompente nel mondo delle monete virtuali:
- Nasce «bitcoin»
- Al momento, bitcoin è la moneta virtuale più nota e usata al mondo
- Capace di movimentare un’incredibile quantità di denaro
- Ma perché questo successo?

BlockChain

- Il successo di bitcoin risiede nella tecnologia su cui si basa la piattaforma Bitcoin (in maiuscolo)
- La piattaforma Bitcoin gestisce gli scambi di bitcoin (la moneta, in minuscolo) e certifica il loro possesso
- La piattaforma Bitcoin è distribuita ed è basata su una tecnologia inovativa (per il 2010) chiamata «BlockChain»

Peer-to-peer Network



Peer-to-peer Network

- Bitcoin è una «peer-to-peer network»
- Non esiste un master o controllore
- Tutti i peer o nodi partecipano al processo, dando il loro «consenso» all’operazione
- La distribuzione del consenso serve per non creare singole vulnerabilità

Catena di Blocchi

- «BlockChain» significa «catena di blocchi»
- Denota il modo in cui i dati vengono memorizzati, al fine di essere immutabili
- Un blocco contiene un gruppo di «transazioni» avvenute più o meno contemporaneamente, descrivendo i dati nello stato prodotto dall’esecuzione delle transazioni
- Ogni blocco punta al blocco precedente, come in una lista
- Quando si deve inserire un nuovo blocco, questo diventa la nuova testa della lista

Tutto Qui?

- Ovviamente no
- Non basta questo per garantire:
  - Immutabilità
  - Resistenza agli attacchi
  - Resistenza alle frodi (double spending, cioè la stessa unità di moneta viene spesa due volte dallo stesso utente)

Hash Code

- Per ottenere l’immutabilità, occorre usare il meccanismo degli Hash Code
- Un hash code è un codice che viene generato da una funzione matematica.
- Nelle BlockChain, deve identificare un blocco
- L’identificatore del blocco è generato a partire dal contenuto del blocco stesso, in base ad un meccanismo di cifratura

Hash Code e Catena

- Ogni blocco contiene al suo interno l’hash code del blocco precedente
- Quindi l’hash code del blocco dipende anche da quello
- **Risultato: un tentativo di alterare la catena, cambiando il contenuto del blocco o il riferimento al blocco precedente renderebbe non più valido l’hash code del blocco stesso**

Generazione dell'Hash Code

- La piattaforma impone che l'hash code debba soddisfare determinati requisiti
- Nello specifico, debba avere una lunghezza elevata e un certo numero di bit debba essere a zero
- La funzione di cifratura usata si basa su un numero casuale, detto «chiave di cifratura»
- Si apre una sfida: esiste una chiave di cifratura che permette di generare un hash code che rispetta i vincoli?

Generazione dell'Hash Code

- La «sfida» viene chiamata «mining», cioè scoprire la chiave di cifratura»
- La sfida è «molto» impegnativa
- Per un solo computer, potrebbero essere necessari anni per trovare la chiave di cifratura
- Ma la risposta deve arrivare in pochi minuti
- La probabilità che ciò avvenga su una sola macchina è estremamente bassa

Mining Distribuito

- Se esistono moltissimi «miners», cioè molti computer che portano avanti la sfida, generando in modo casuale le chiavi di cifratura, allora la probabilità di risolvere la sfida in tempi brevi aumenta notevolmente
- Appena un miner trova la chiave, comunica a tutti i nodi l’hash code e la chiave di cifratura
- Gli altri nodi verificano che sia corretta

Mining Distribuito

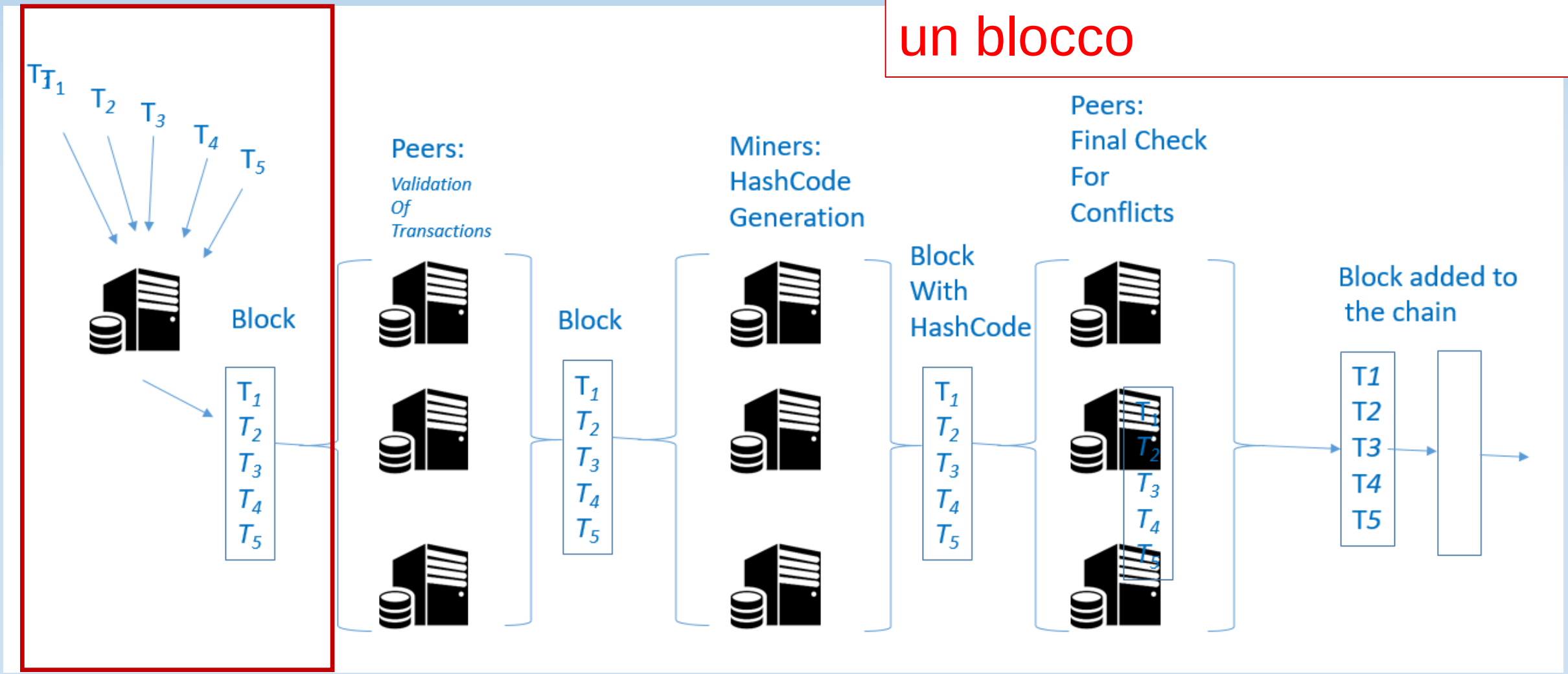
- Così facendo, il consenso ad aggiungere il nuovo blocco alla catena è distribuito
- **Risultato: anche ammesso che un nodo della rete sia in grado di generare un blocco con transazioni fraudolente, non avrebbe mai la capacità di generare un hash code che rispetta i vincoli in tempi ragionevoli**
- La piattaforma rende più o meno stringenti i vincoli, in base ai miners disponibili

Proof of Work

- Il meccanismo che abbiamo appena visto si chiama «Proof of Work»
- Ma in realtà, manca ancora un pezzo: come si evita il double spending?
- È insito nel meccanismo stesso, che adesso vediamo in dettaglio in tutte le sue fasi

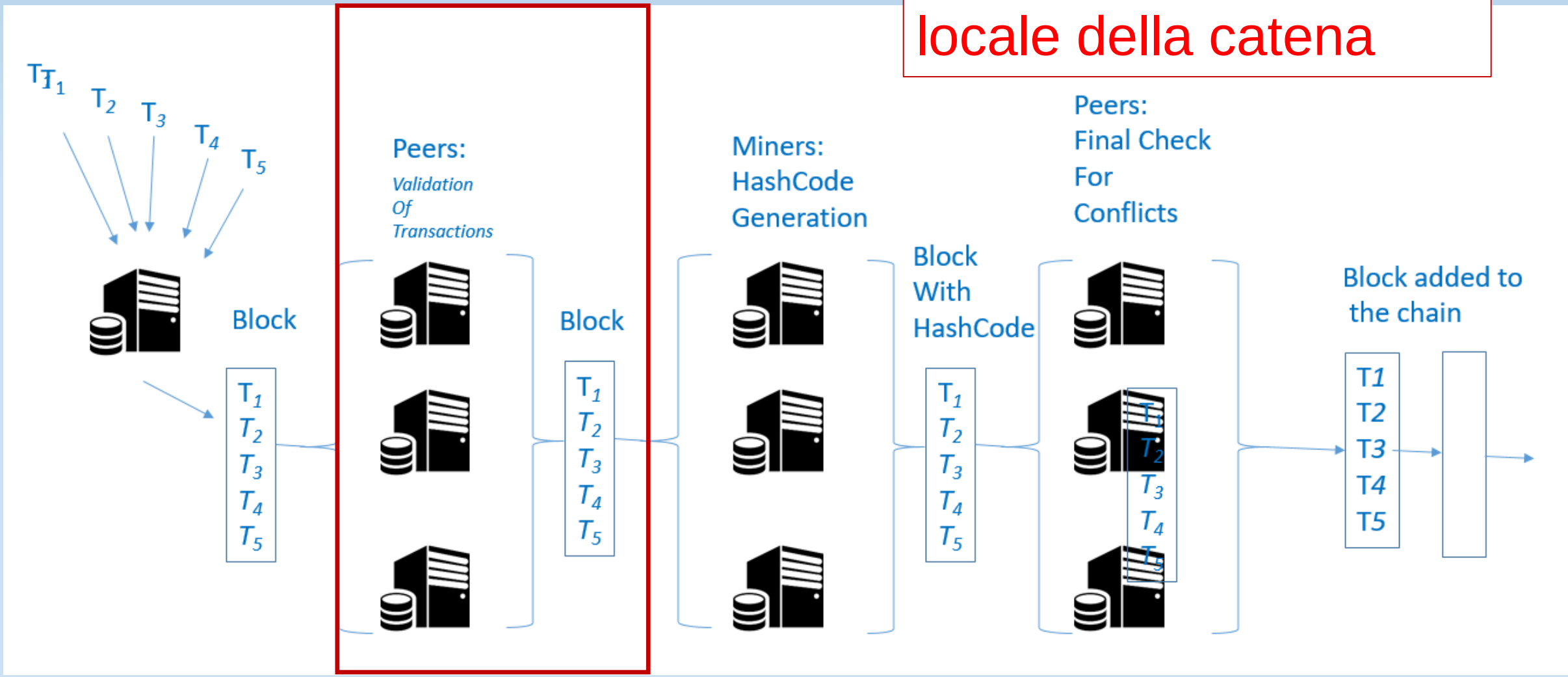
Step 1: Gathering

Le transazioni ricevute in un intervallo di tempo molto limitato sono raccolte in un blocco



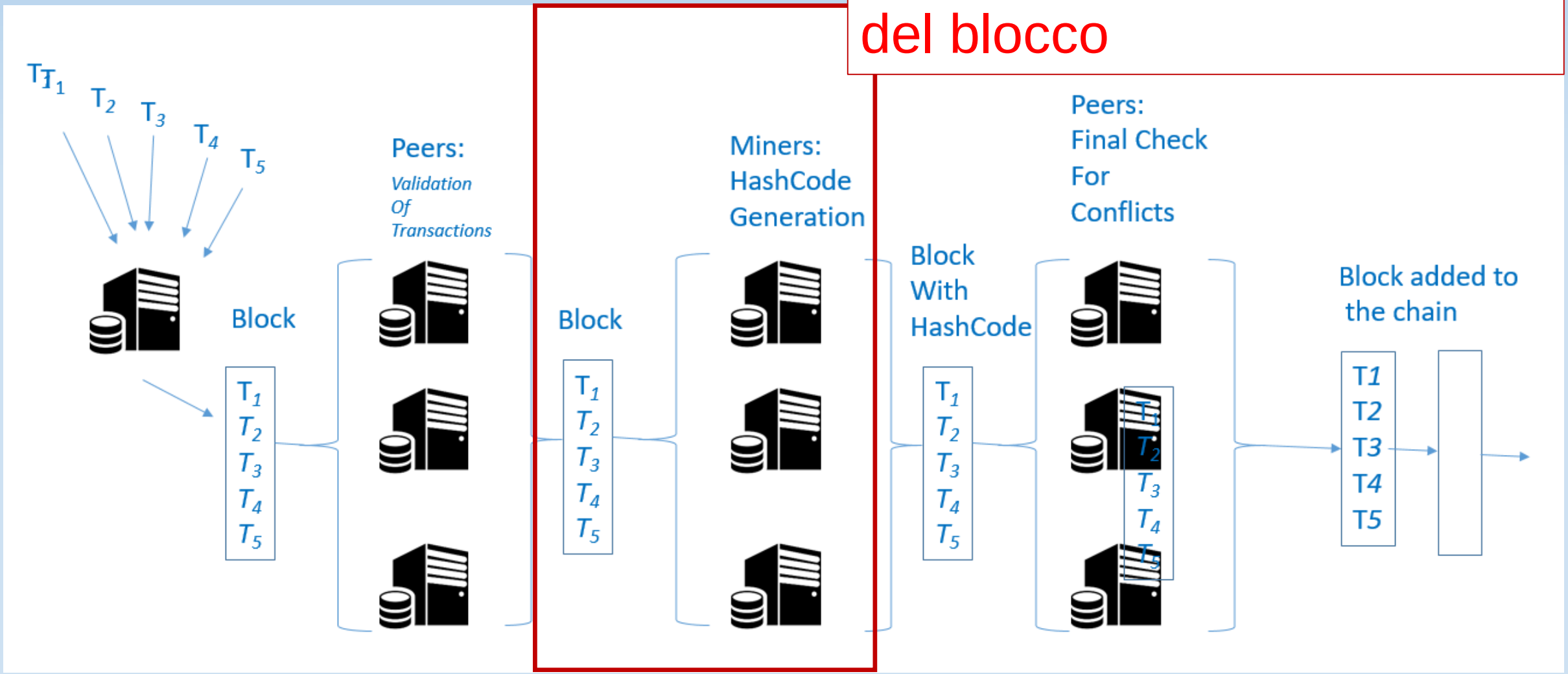
Step 2: Validation

Tutti i peer validano le transazioni nel blocco, leggendo la loro copia locale della catena



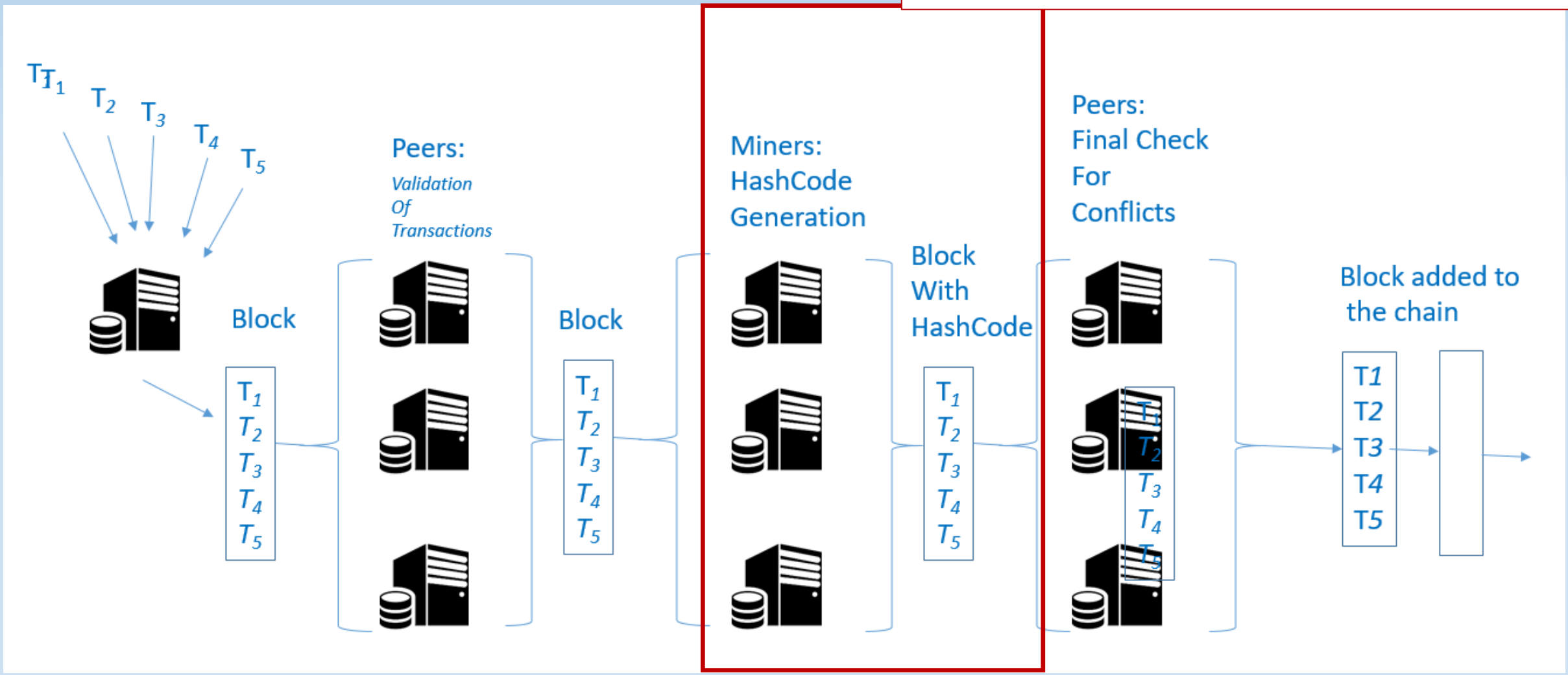
Step 3: Hash-Code Generation

Si deve generare un Hash Code univoco, come identificatore univoco del blocco



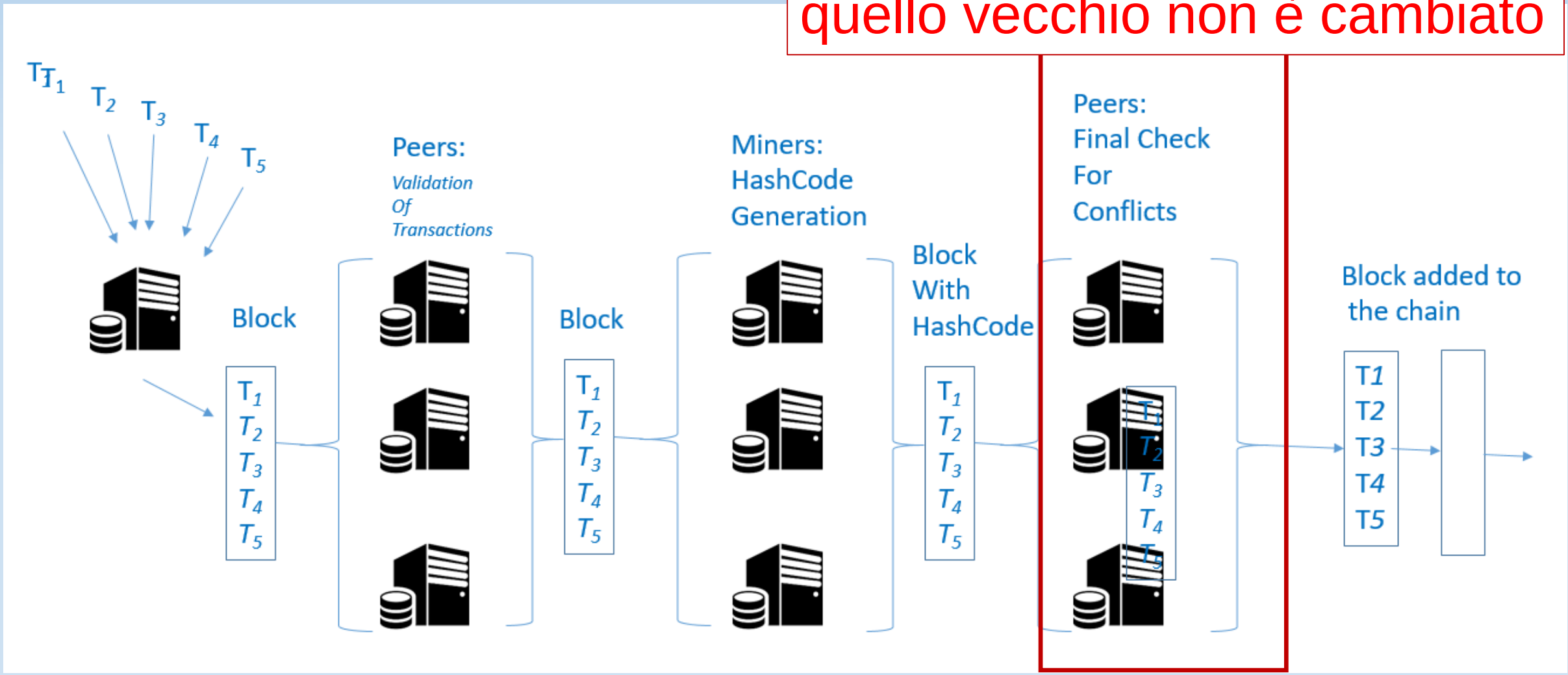
Step 3b: Hash-Code Trovato

Un miner trova l'hash code e lo manda ai peer per validarlo



Step 4: Checking for Conflicts

I peer validano l'hash cpde e rendono il blocco il nuovo primo blocco, se quello vecchio non è cambiato



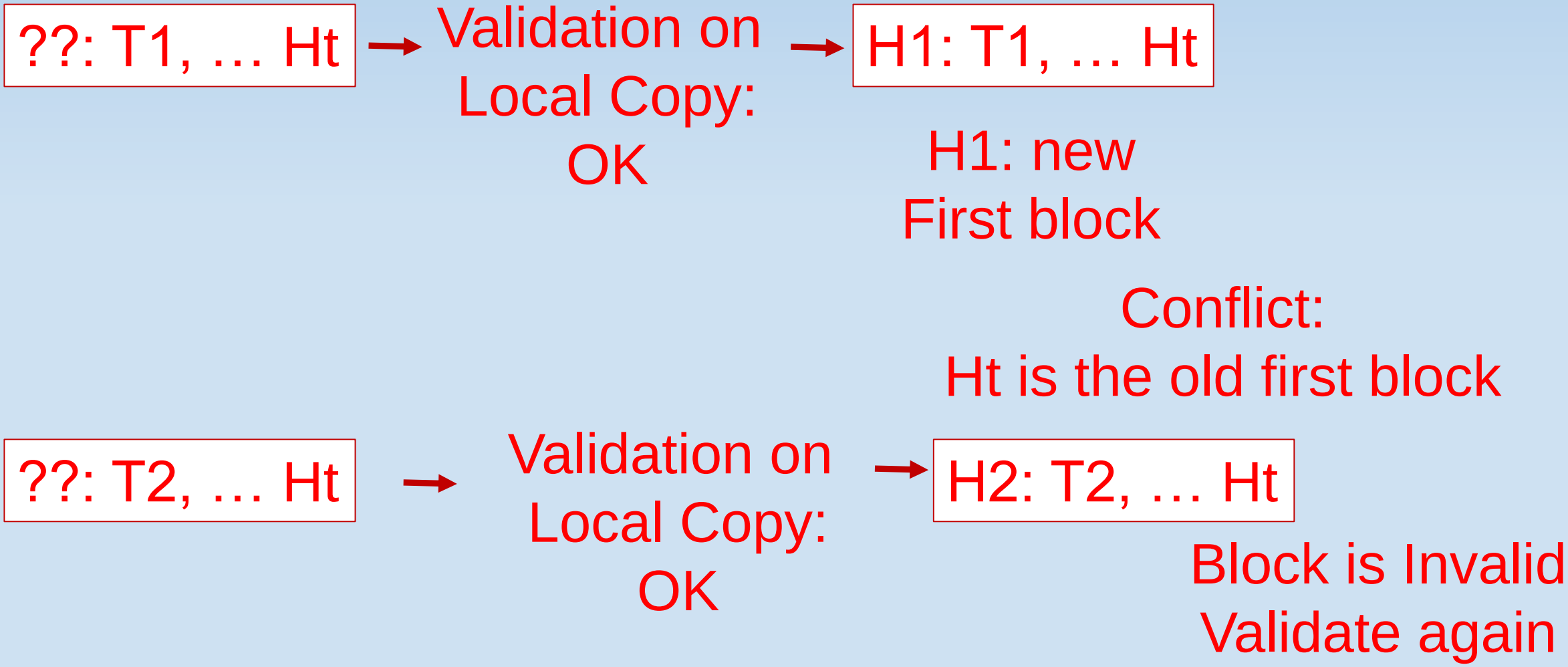
Il Double Spending?

- I blocchi sono processati in parallelo, ma richiede tempo
- Quindi, il tentativo di double spending potrebbe essere fatto mandando due transazioni ravvicinate, prima che la validazione sia avvenuta
- Sperando che la rete non si accorga

Conflitto

- Il conflitto si verifica perché uno dei due blocchi processati in parallelo arriva tardi
- Entrambi puntano allo stesso vecchio primo blocco della catena
- Il primo che arriva, viene inserito
- Il secondo, quando arriva, punta ad un blocco che non è più il primo della catena

Double-Spending Transactions



Si Blocca il Double Spending

- Le transazioni nel blocco risultato non valido vengono validate di nuovo
- E così si scopre che l’importo è già stato speso
- La transazione fraudolenta viene rifiutata

Problema

- L’approccio funziona, ma
- Richiede un incredibile sforzo computazionale
- È abbastanza lento (circa 10 minuti per transazione)
- Si consuma una quantità abnorme di energia elettrica

Smart Contract

- Che coa è uno Smart Contract?
- Un contratto tra due o più parti
- Che non richiede l’intervento umano per essere portato avanti
- Si tratta di un programma che contiene le regole di correttezza e le regole di trasformazione dei dati associati al contratto
- La BlockChain garantisce l’integrità temporale di tutti gli stati del contratto

Smart Contract

- Ma se le operazioni sul contratto non vengono validate da un essere umano
- Il contratto può anche perdere la sua validità legale
- Quindi, ci sono smart contract che non hanno bisogno di validità legale e altri che ne hano bisogno

Come Caratterizzare le Piattaforme di BlockChain

Accesso alla Piattaforma

- **Permissionless BlockChain**
  Chiunque può entrare nella rete, basta installare il software necessario e rispettare le regole di comportamento
- **Permissioned BlockChain**
  Solo i peer autorizzati possono entrare
  C’è un amministratore della rete che concede le autorizzazioni

Supporto agli Smart Contract

- **Internal Code**
  Il codice è memorizzato ed eseguito nella piattaforma
  - **Contract-Specific Code**
    il codice non viene condiviso
  - **Contract-Family Code**
    una famiglia di contratti condivide il codice (template)
  - **Global Code**
    il codice è globale, può operare su tutti i dati

Supporto agli Smart Contract

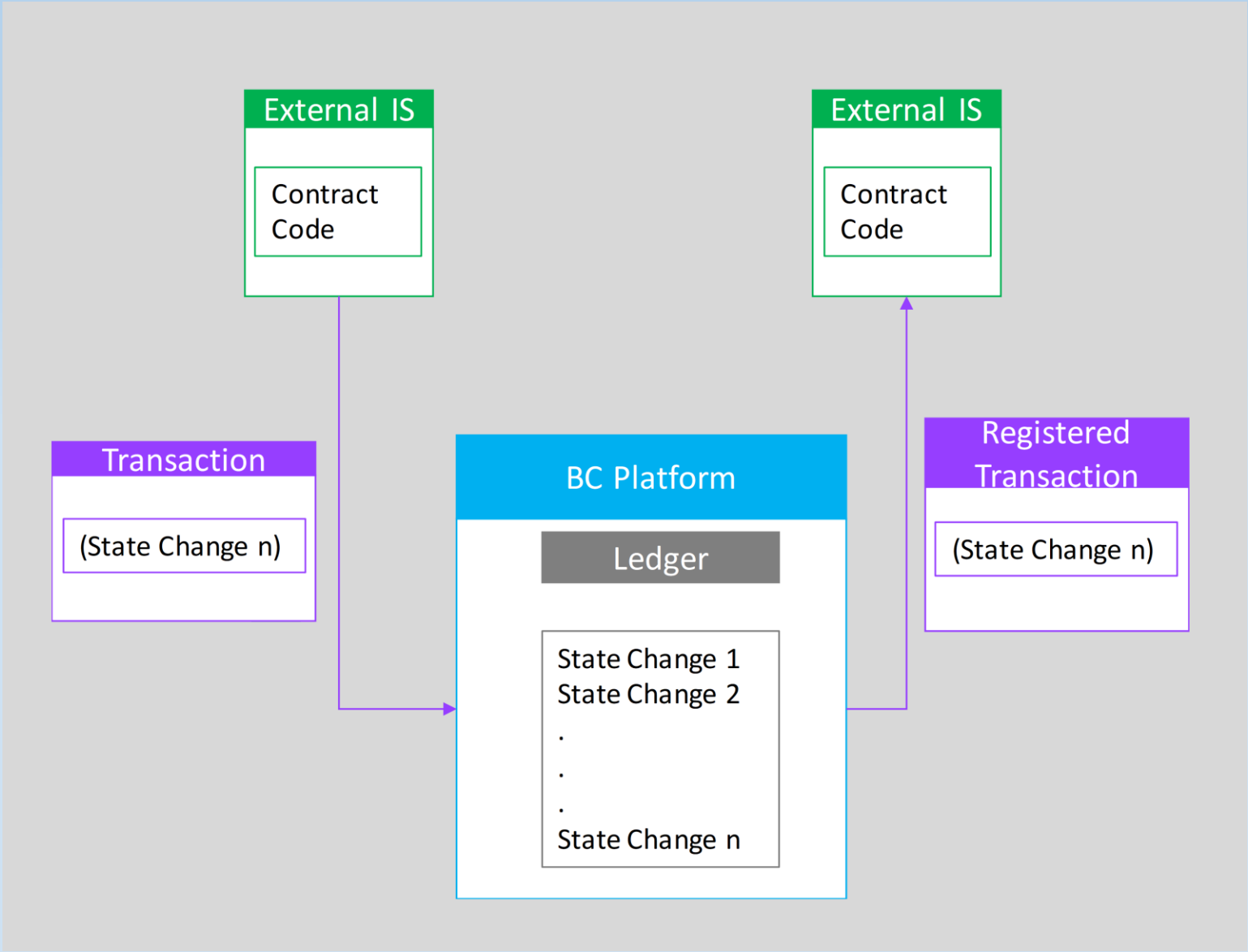
- **External Code**
  Il codice è memorizzato ed eseguito al di fuori della piattaforma
- Perché la piattaforma non supporta l’esecuzione di codice al suo interno

Caratterizzazione delle Piattaforme di BlockChain più famose

Bitcoin

- **Permissionless platform**
- Moneta virtuale: **bitcoin**
- Nessun supporto per *Smart Contracts* (eseguiti al di fuori della piattaforma)
- Meccanismo di consenso: P**roof of Work**

Bitcoin


External IS
Contract Code
External IS
Contract Code
Transaction
(State Change n)
BC Platform
Ledger
State Change 1
State Change 2
.
.
.
State Change n
Registered Transaction
(State Change n)

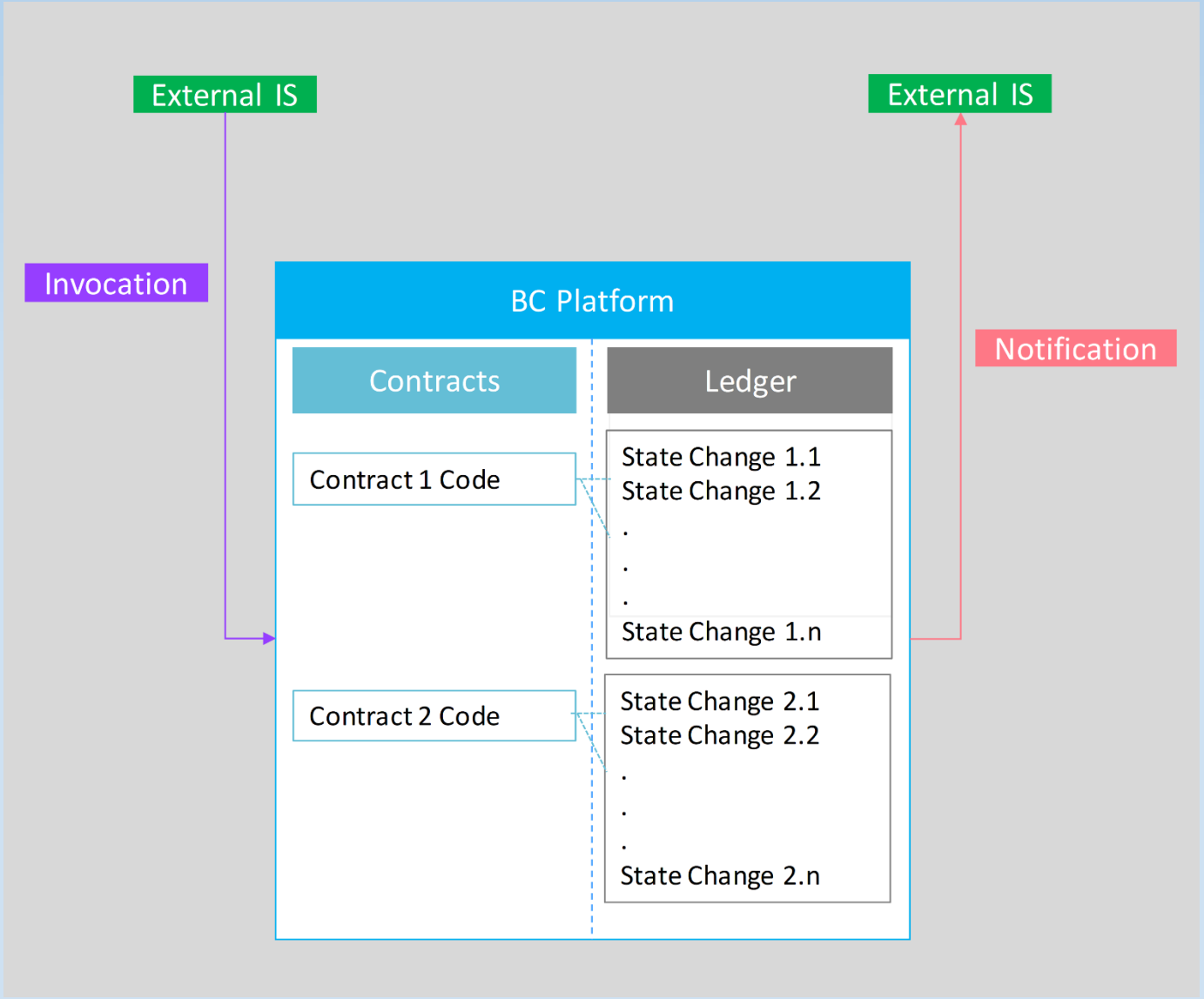
Bitcoin

- I sistemi informativi esterni che gestiscono lo smart contract eseguono il codice al loro interno
- I peer coinvolti si scambiano messaggi crittografati che solo loro possono leggere
- La piattaforma non garantisce alcun controllo sulla corretta esecuzione del contratto

Ethereum

- **Permissionless platform**
- Moneta virtuale: **Ether**
- Supporta gli *Smart Contracts*
- Linguaggio di programmazione: **Solidity**
- **In-platform code, Contract-specific code**
- Meccanismo di consenso: P**roof of Work**

Ethereum


External IS
External IS
Invocation
BC Platform
Notification
Contracts
Ledger
Contract 1 Code
State Change 1.1
State Change 1.2
.
.
.
State Change 1.n
Contract 2 Code
State Change 2.1
State Change 2.2
.
.
.
State Change 2.n

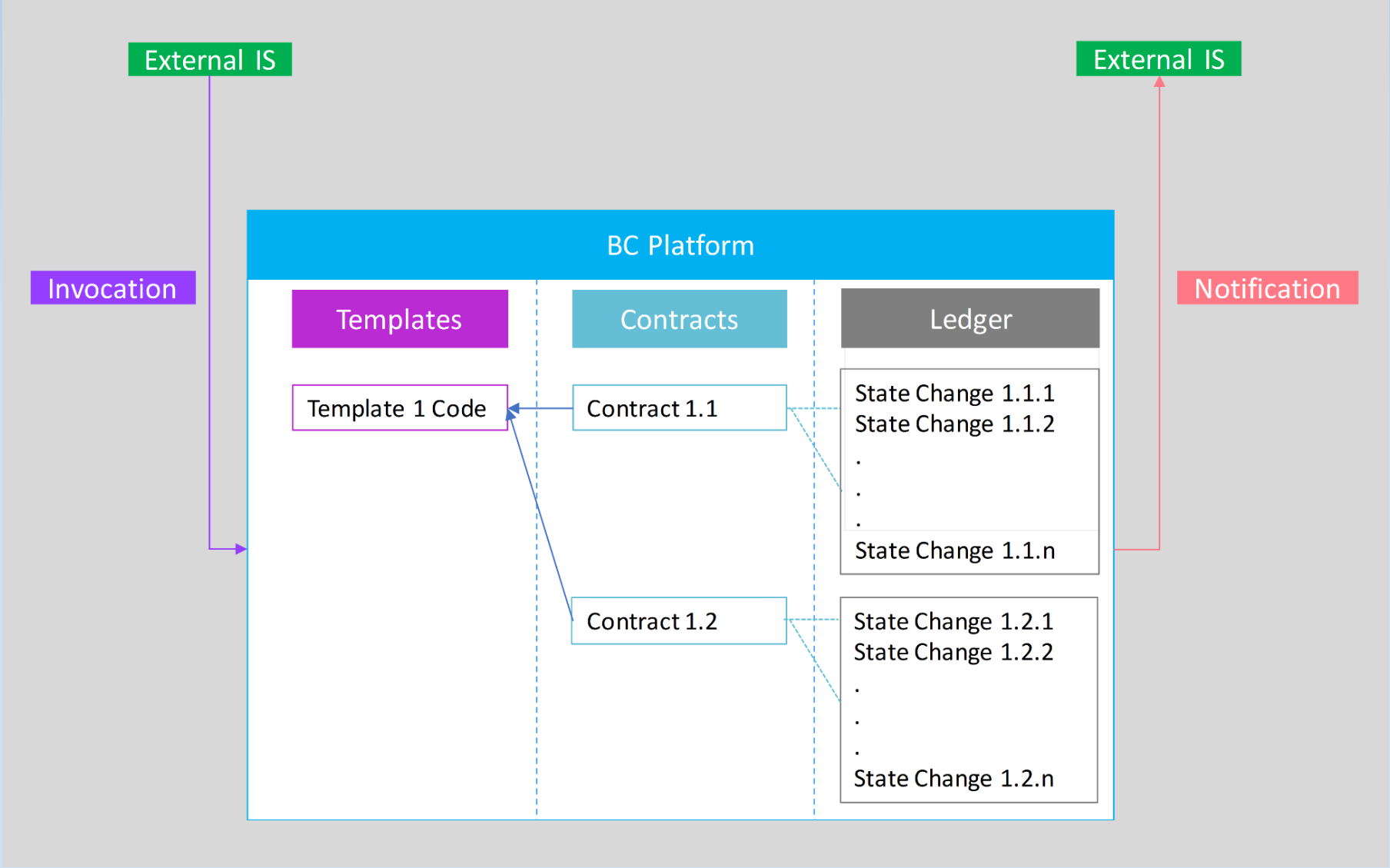
Ethereum

- Gli smart contract possono lavorare anche senza la moneta virtuale
- Il codice viene eseguito da tutti i peer, durante il processo di validazione
- Ulteriore appesantimento del carico computazionale dovuto al Proof of Work

Corda

- **Permissioned platform**
- **No virtual currency**
- Supporta gli *Smart Contracts*
  Legal-prose version of smart contracts
- **In-platform Code, Contract-family code (contract template)**
- Linguaggi di programmazione: **Java, Kotlin**
- Meccanismo di consenso: **Proof of Knowledge**

Corda


External IS
External IS
Invocation
Notification
BC Platform
Templates
Contracts
Ledger
Template 1 Code
Contract 1.1
State Change 1.1.1
State Change 1.1.2
.
.
.
State Change 1.1.n
Contract 1.2
State Change 1.2.1
State Change 1.2.2
.
.
.
State Change 1.2.n

Corda

- Gli utenti possono creare famiglie di contratti (contract template) che si comportano nello stesso modo, ma cambiano per i dettagli del contratto
- Esempio: mutui
  Importo finanziato
  Tasso di interesse
  Numero di rate

Corda

- La caratteristica principale di Corda si chiama Legal-Prose version
- Cioè uno smart contract DEVE avere una versione testuale
  - dei termini del contratto
  - delle operazioni svolte sul contratto
- In questo modo, il contratto mantiene validità legale a tutti gli effetti

Proof of Knowledge

- A differenza del Proof of Work
- L’approccio Proof of Knowedge basa il consenso sulla conoscenza di ciò che avviene
- Non tutti i peer vengono coinvolti, ma solo alcuni
- Questi confermano che sono a conoscenza che le operazioni si possono svolgere
- **Risultato: pochi peer coinvolti equivale ad avere bisogno di poche risorse e quindi si ottiene maggiore velcoità**

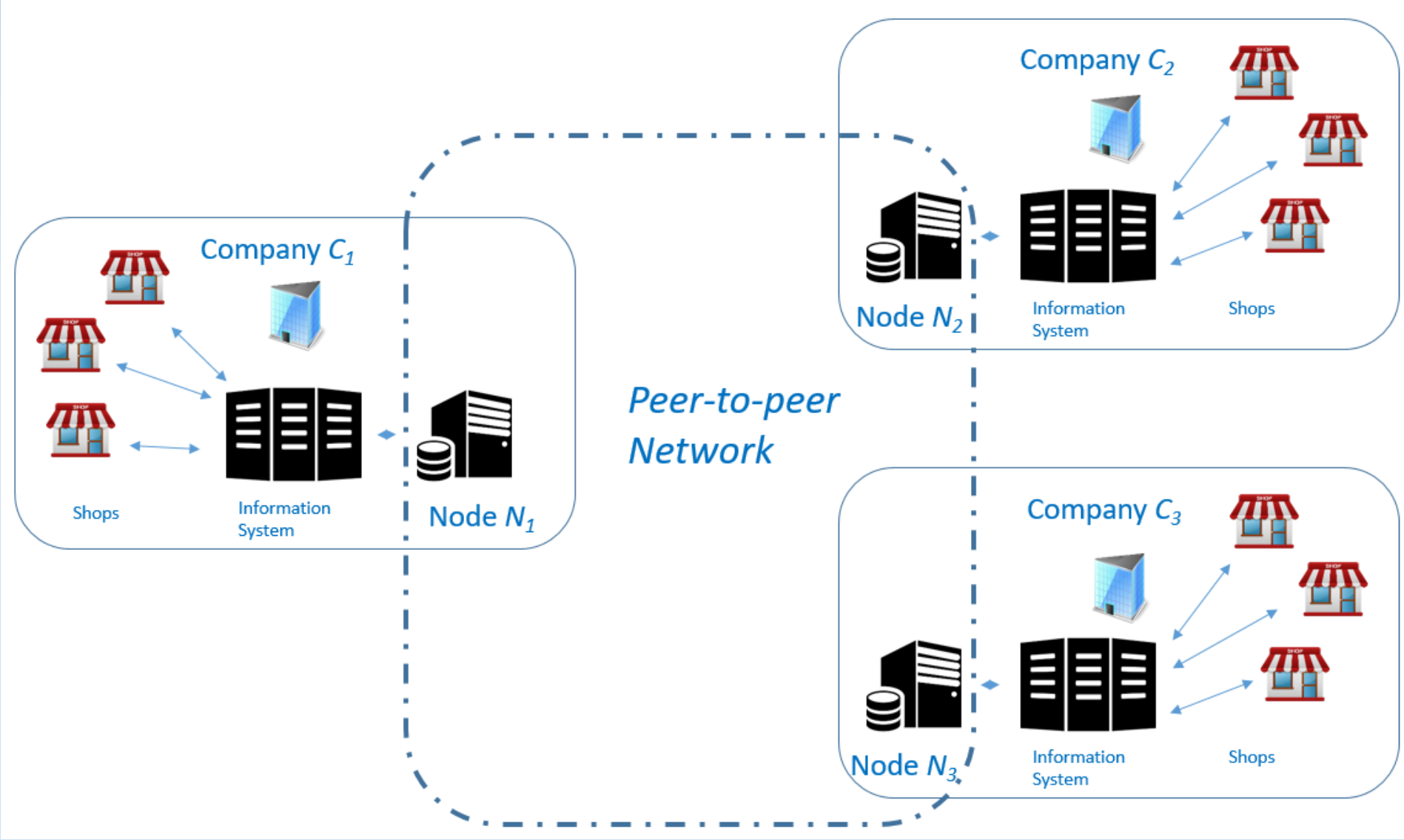
HyperLedger Fabric

- Linux Foundation ha avviato il progetto HyperLedger
- Obiettivo: sviluppare piattaforme di tipo permissioned
- Di supporto ai sistemi informativi
- Per creare dei database distribuiti e condivisi da sistemi informativi che devono cooperare
- Il prodotto più famoso: HyperLedger Fabric

Esempio Applicativo

- Alcune aziende della grande distribuzione decidono di gestire in comune un’unica tessera fedeltà
- Con questa tessera, i clienti possono accumulare punti facendo acquisti presso uno qualsiasi dei negozi delle aziende consorziate
- E possono ottenere premi e sconti presso una qualsiasi delle aziende consorziate
- Chi gestisce il «wallet dei punti» di ciascun cliente?

Esempio Applicativo



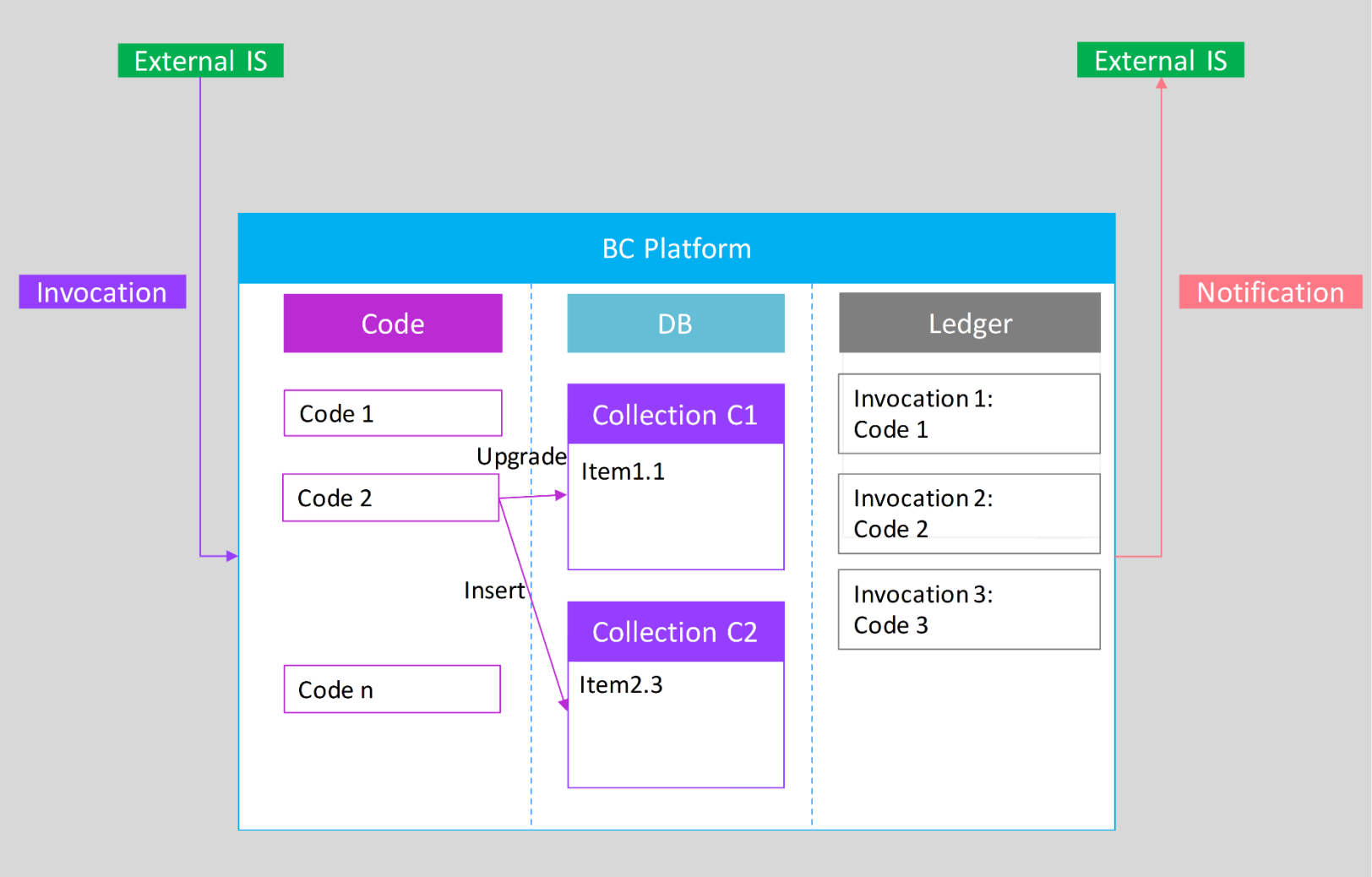
Esempio Applicativo

- Nessuno in particolare
- Con HyperLedger Fabric si crea un database condiviso
- I sistemi informativi di ciascuna azienda forniscono il loro nodo e interagiscono con questo
- La business logic della gestione dei punti è gestita dalla BlockChain (global code)

HyperLedger Fabric

- **Permissioned platform**
- No Moneta Virtuale
  Fornisce il concetto di database distribuuito
  No smart contracts, ma **ChainCode**
- **In-platform Code, Global Code**
- Linguaggi di programmazione: java, JavaScript, Go
- Meccanismo di consenso: di tipo Proof of Knowledge Byzantine Fault Tolerant (BFT) consensus mechanism

HyperLedger Fabric



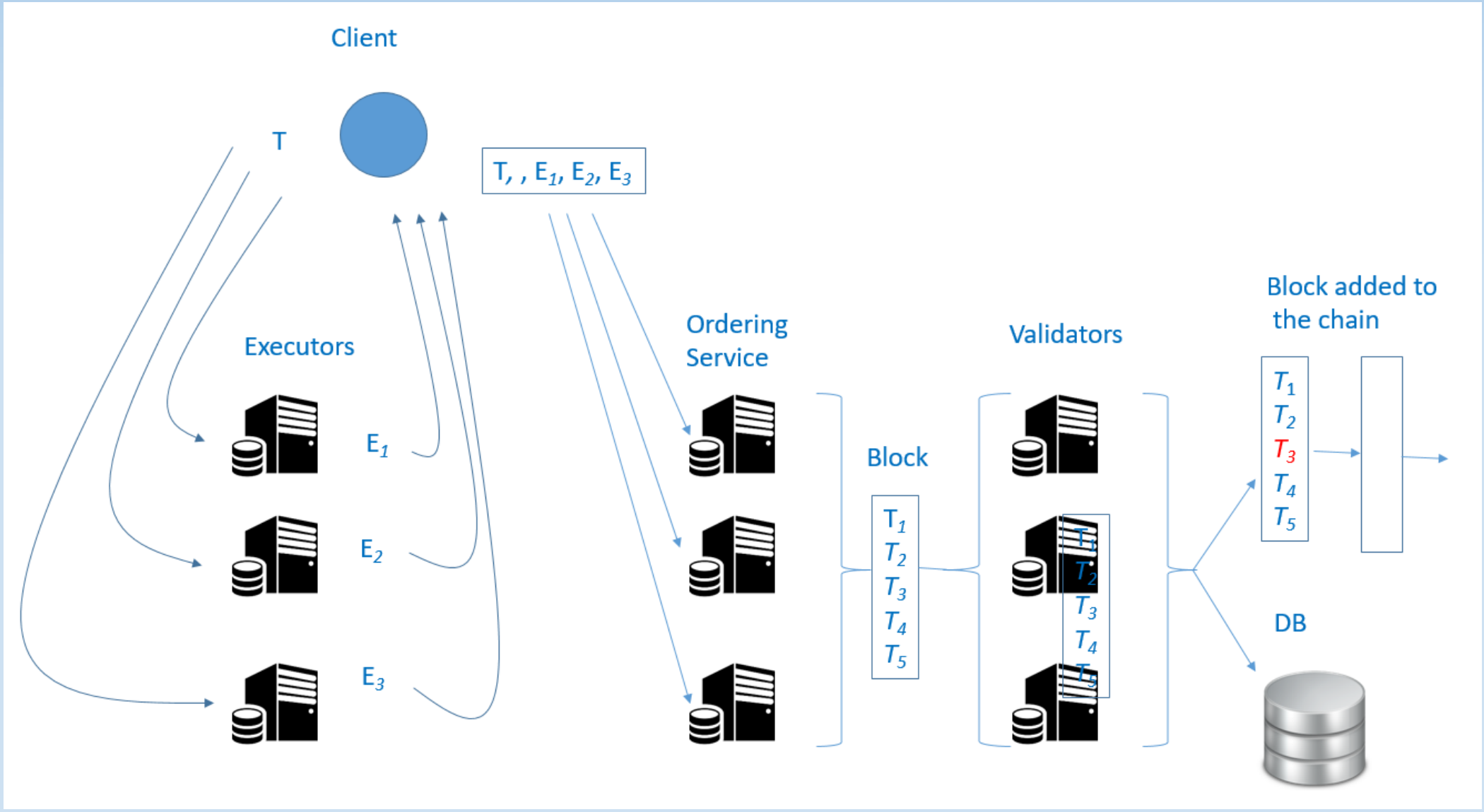
Quale Database?

- Il database è un NoSQL database
- Per l’esattezza è un JSON Document Store
- Viene usato CouchDB
- Scelto perchè fornisce il support transazionale
- Le query possono essere fatte direttamente sul db
- Il Ledger fa da log del DB: registra tutte le operazioni che hanno portato allo stato corrente del DB

Byzantine Fault Tolerant Mechanism

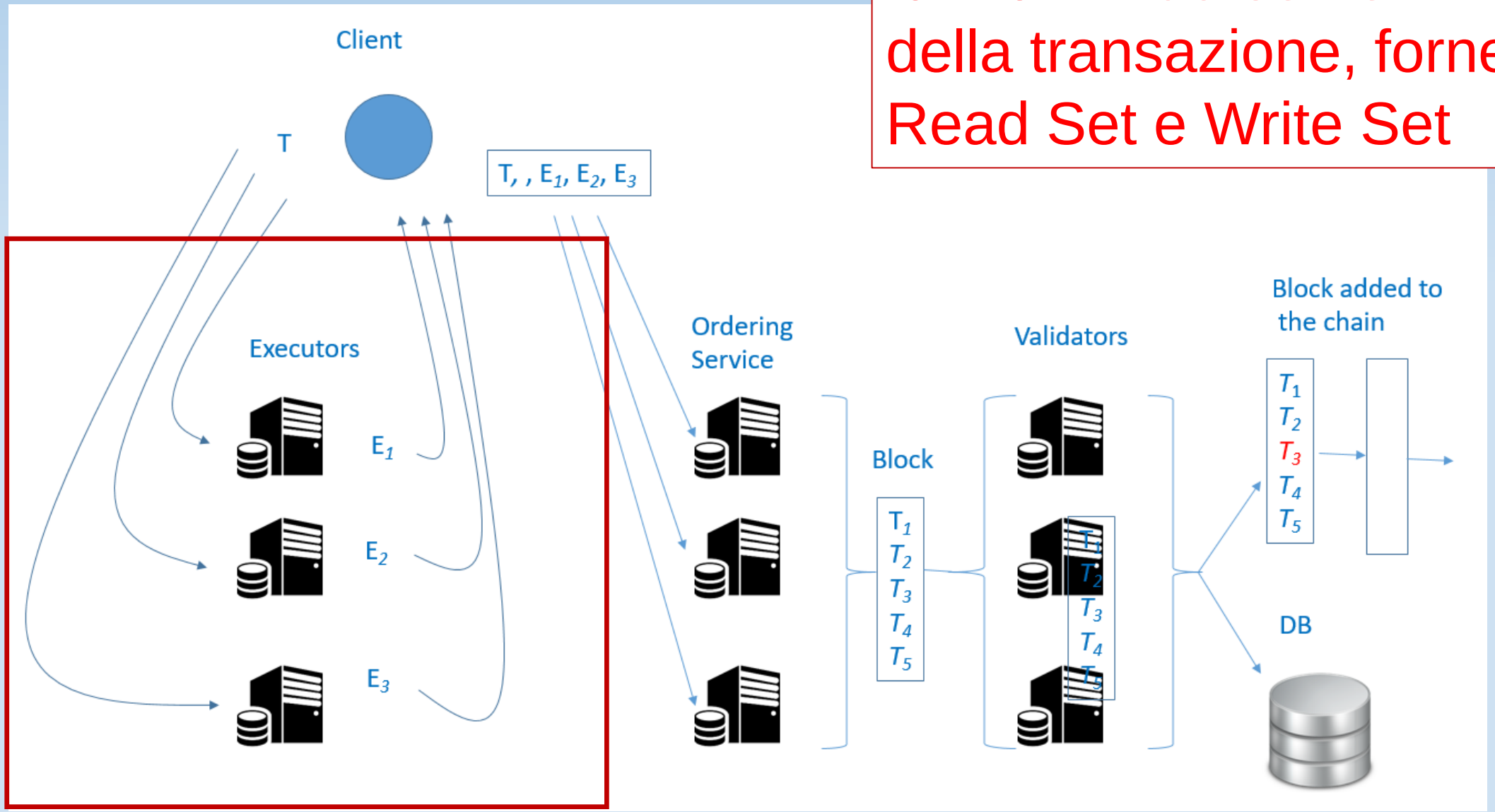
- Perché Byzantine?
- Perché si gestisce lo stesso la transazione, verificando i conflitti a posteriori, lavorando sulle dipendenze tra gli aggiornamenti
- **Read Set**: insieme degli oggetti modificati dalla transazione
- **Write Set**: insieme dei nuovi oggetti
- Vediamo nel dettaglio

Byzantine Fault Tolerant Mechanism



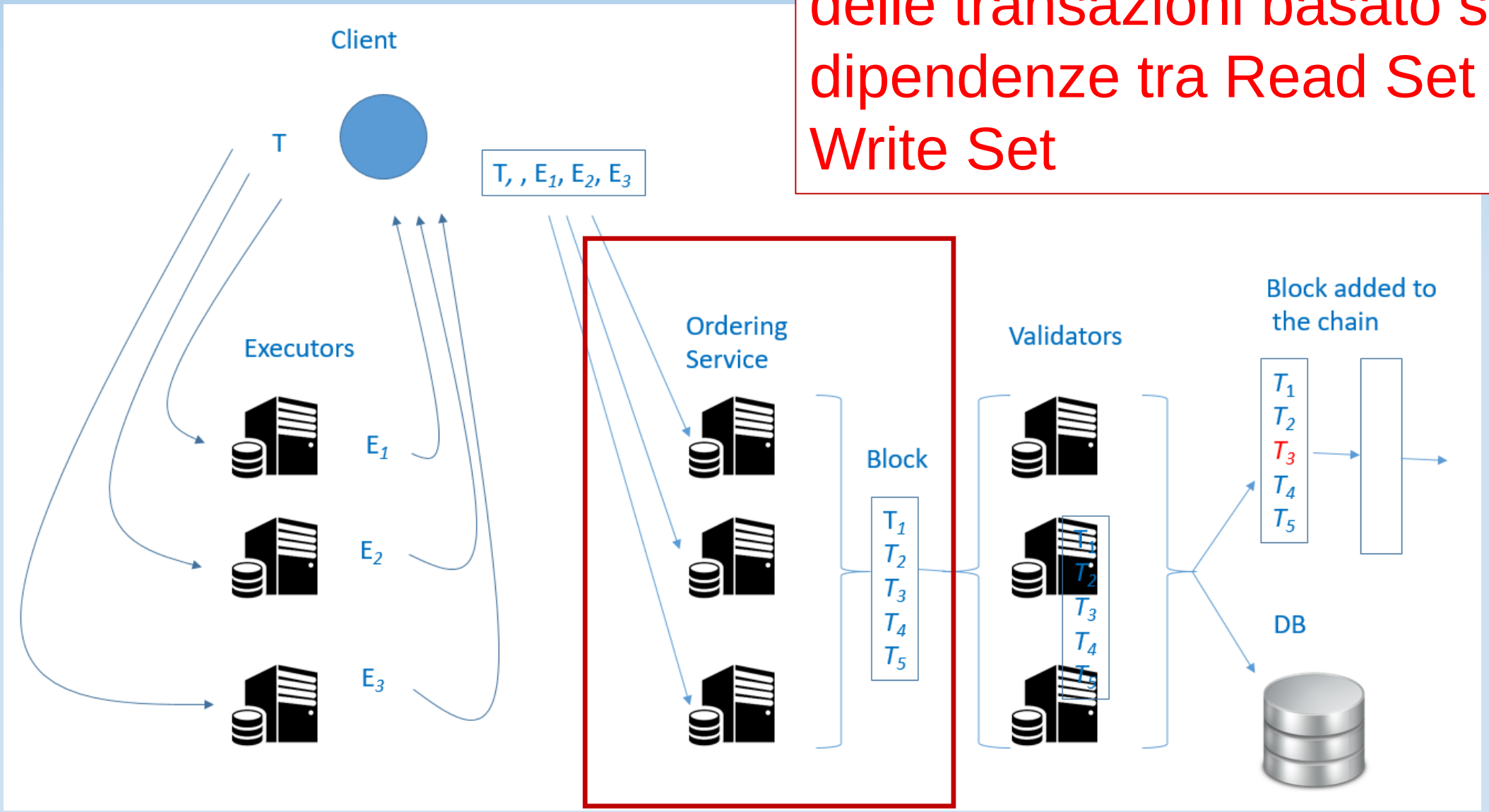
Step 1: Execution

Gli Executor eseguono e fanno l'Endorsement della transazione, fornendo Read Set e Write Set



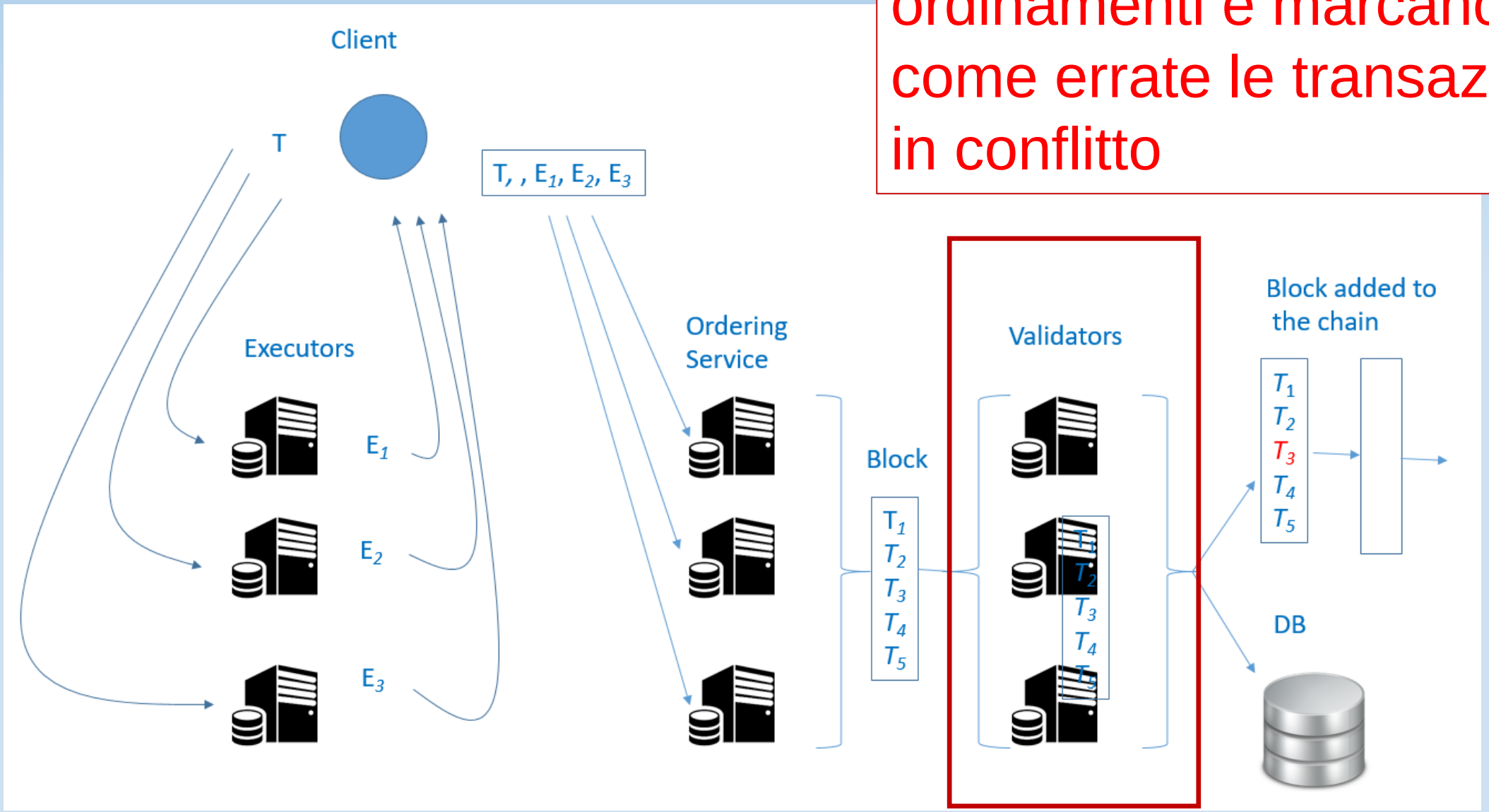
Step 2: Ordering

Gli Ordering Services costruiscono un ordinamento delle transazioni basato sulle dipendenze tra Read Set e Write Set



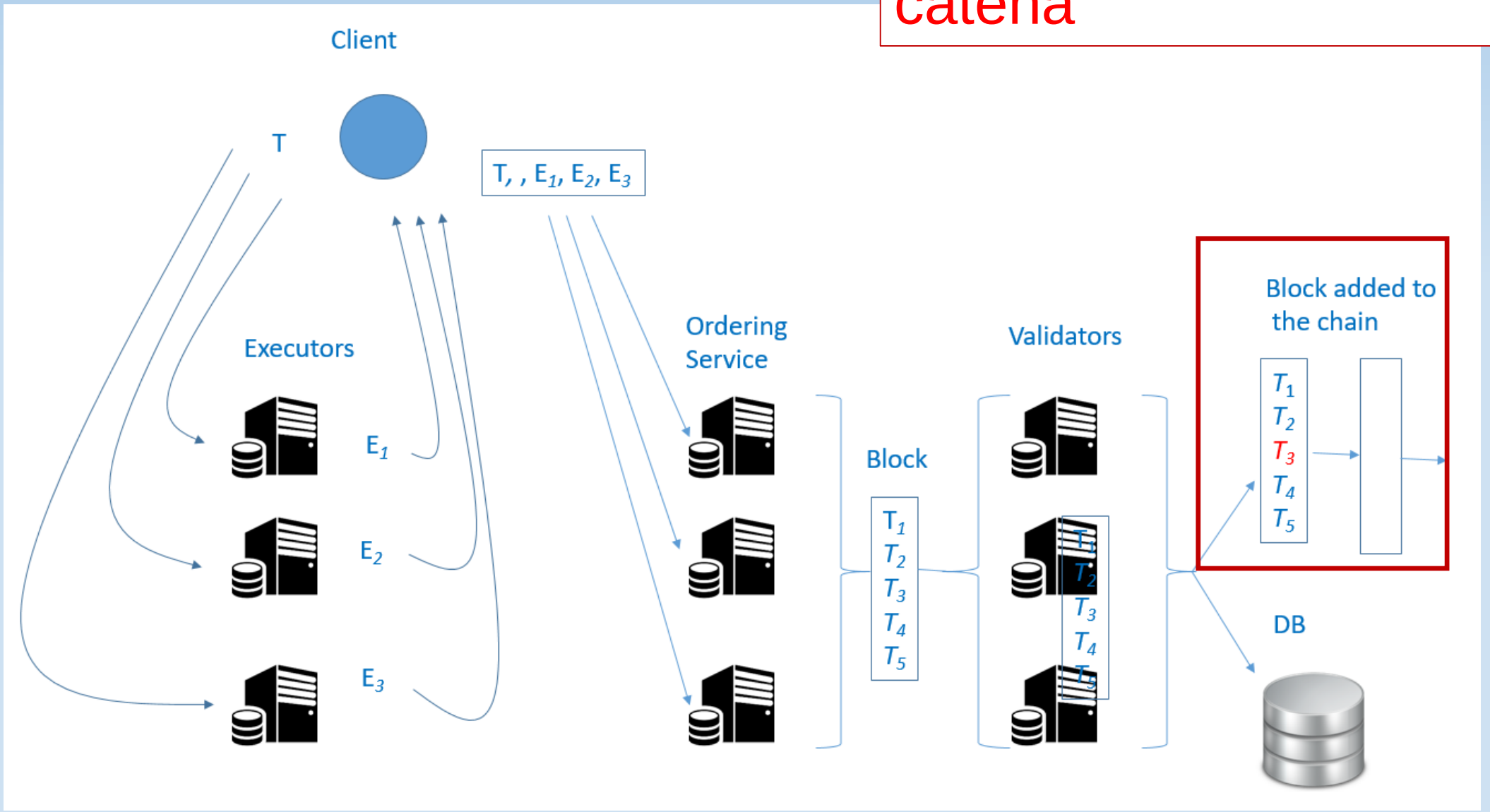
Step 3: Ordering

I Validators verificano la presenza di conflitti negli ordinamenti e marcano come errate le transazioni in conflitto



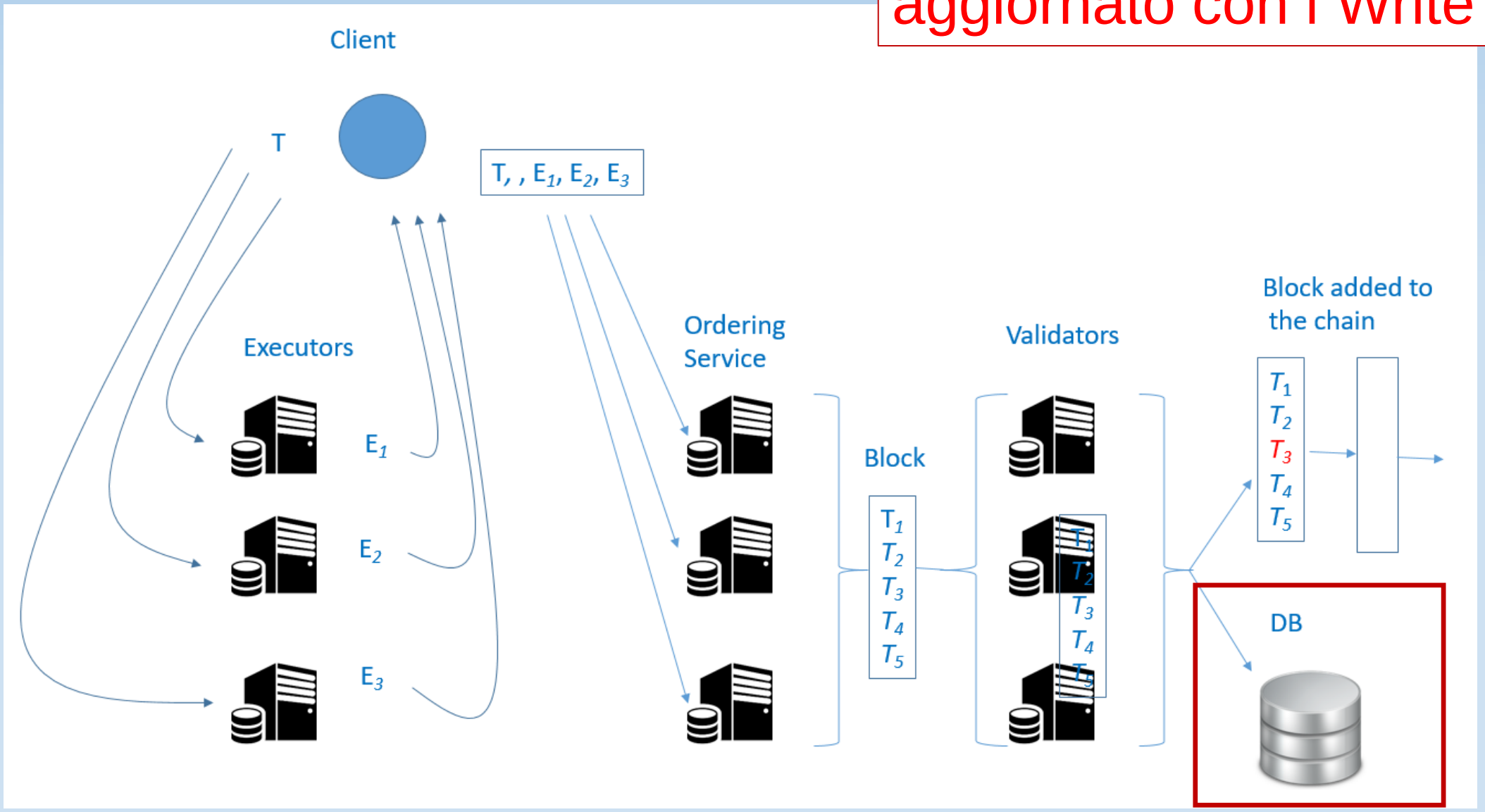
Step 4: Ordering

Il blocco è aggiunto alla catena



Step 4: Ordering

Il database è aggiornato con i Write Set



BFT: Vantaggi

- Questo meccanismo di consenso richiede limitata capacicà computazionale
- Quindi ottiene velocità molto alte, rispetto al Proof of Work
- Quanto veloce? Da alcuni stress test in letteratura, si ottiene un tempo massimo di circa 30 secondi per transazione, contro alcune decine di minuti
- Ma per processare transazioni di carte di credito, è ancora troppo alto

Algorand

- **Permissionless platform**
- Monete virtuali:
  - ALGO: quella nativa di Algorand
  - Tether
  - USD Coin (USDC)
  - Sovereign, moneta virtuale delle Isole Mashall
- Supporta gli *Smart Contracts*
- Linguaggi di progr.: Go, Java, JavaScript e Python
- **In-platform code, Contract-specific code**
- Meccanismo di consenso: P**roof of Stake**

Proof of Stake

- Un terzo meccanismo di consenso si sta affermando
- Si chiama «Proof of Stake»
- Non ho trovato molti dettagli, ma i principi di funzionamento sì
- Obiettivo: ridurre il «dispendio di energia»

Proof of Stake

- La piattaforma suddivide il tempo in «**time slots**», cioè in periodi di tempo tutti della stessa durata e rigorosamente consecutivi
- Per ogni time slot, al più un blocco può essere aggiunto al ledger:
  contiene tutte le transazioni effettuate durante il time slot.

Proof of Stake

- Chi valida il blocco?
  Uno Stakeholder a caso
- Chi è un **Stakeholder**? Un utente che possiede coin e capacità computazionali (un nodo)
- Lo stakeholder estratto a sorte valida le transazioni nel blocco, usando la sua copia del ledger e, se sono tutte valide, autorizza l’aggiunta del blocco al ledger

Proof of Stake

- Su quale base viene estratto lo Stakeholder?
  In base al numero di coin posseduti, infatti viene estratto il coin, non lo Stakeholder
- Che vantaggio ha uno Stakeholder a mettere a disposizione la propria potenza di calcolo?
  Riceve un coin come ricompensa per il lavoro svolto

Proof of Stake: Vantaggi

- Un solo peer è coinvolto nella validazione
- Elevata velocità di validazione (quasi istantanea)
- Ridottissimo consumo di energia

- Ma è robusto?
  Sembra di sì, infatti viene adottato da molte piattaforme, tra cui Algorand
- Sembra che Ethereu, stia passando al Proof of Stake

NFT

- NFT: Non-Fungible Token
- Un gettone/coin/token che rappresenta qualche cosa di unico, non un valore in denaro.
- Perché «non-fungible»?
- Fungible significa «sostituibile»
  I token che rappresentano denaro possono essere sostituiti (perché vengono spesi parzialmente)

NFT

- Un NFT rappresenta un oggetto (reale o virtuale)
- Non può essere speso o sostituito da altri oggetti
- Può essere solo Ceduto
- Quindi, un NFT è un «Certificato di Proprietà» di un oggetto (anche virtuale)

Algorand e NFT

- Algorand supporta gli NFT
- La SIAE (società Italiana Autori ed Editori) nel Marzo 2021 ha caricato più di 4 Milioni di NFT per certificare i diritti di autore degli autori italiani

Ethereum e NFT

- L’altra piattaforma che supporta gli NFT è Ethereum (penso da più tempo di Algorand)

- Il 6 Aprile 2021 Morgan (il cantante) lancia un’asta per la canzone inedita «*Premessa della premessa*»
  la vende a 10 Ether (21000 Euro)

Ethereum e NFT

- L'aggiudicatario dell'Opera, è cos’ diventato l'unico proprietario e titolare esclusivo
- Ha poi ricevuto da Morgan anche le stampe uniche e originali autografate con i testi del brano