

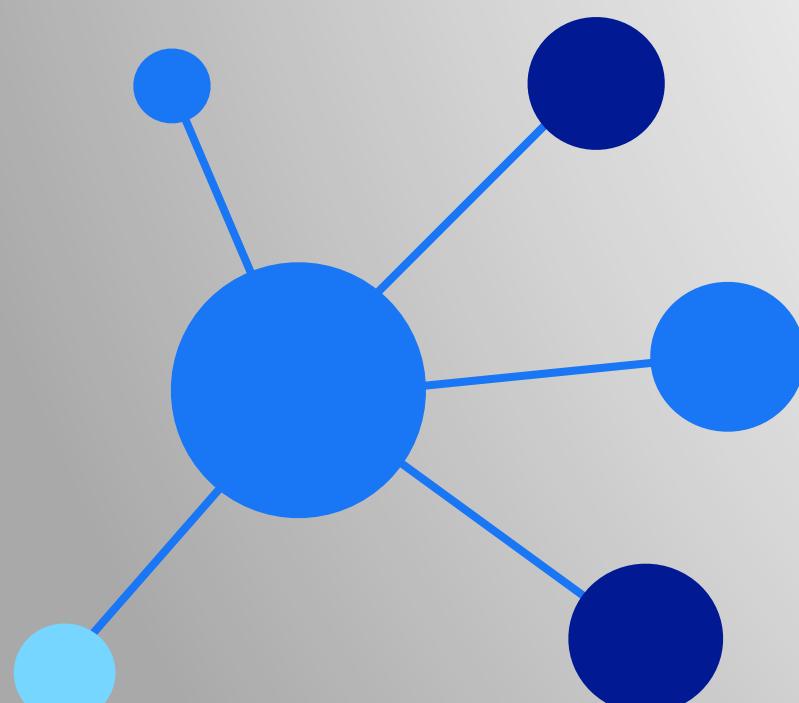
Alma Mater Studiorum - Università di Bologna  
Scuola di Scienze, Dipartimento di Informatica - Scienza e Ingegneria  
Corso di Laurea Magistrale in Informatica  
Curriculum in Informatica per il Management  
A.A. 2019/2020

---

# Holobook

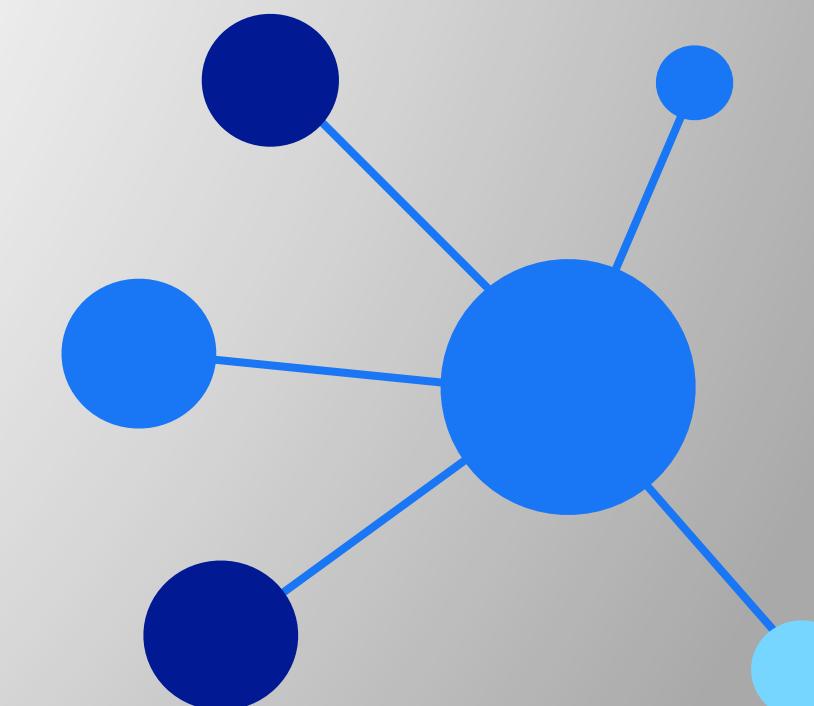
## Progetto di Sistemi Peer-to-Peer

---



Lorenzo Biagio Lanzarone

Cristian Romanello



# Social network tradizionali

- I social network che conosciamo e utilizziamo hanno impiegato pochi anni per **diffondersi** in modo capillare in tutto il mondo.
- Con la stessa rapidità si è assistito a **scandali** e **fenomeni controversi** legati a queste piattaforme: influenza degli algoritmi, filter bubble, manipolazione dell'opinione pubblica, fake news, disinformazione, censura, violazione della privacy e del copyright.
- Appaiono come un **ambiente pubblico** e neutro, senza esserlo realmente. Il flusso dei contenuti è determinato da algoritmi che influenzano l'esperienza e l'attività di ogni singolo utente.



# Social network tradizionali

- Un aspetto critico è l'**analisi dei dati**, dietro ai bottoni “mi piace” e “condividi” vi è un **sistema che elabora e registra le interazioni**, i gusti e le opinioni degli utenti, per poi vendere tali informazioni a società terze.
- Altri **svantaggi** dei social network tradizionali sono:
  - **Scalabilità** degli utenti
  - **Aumento dei costi** di gestione
  - **Fiducia** ridotta degli utenti



# Social network decentralizzati

- In controtendenza a questo paradigma, troviamo i **social network decentralizzati**.
- Non un **unico fornitore** ma un **insieme di pari** che condividono l'esecuzione del sistema.
- Implementazione di meccanismi per permettere agli utenti di **trovare i propri amici**.
- Da considerare inoltre la **disponibilità dei dati** e la **diffusione delle informazioni**.



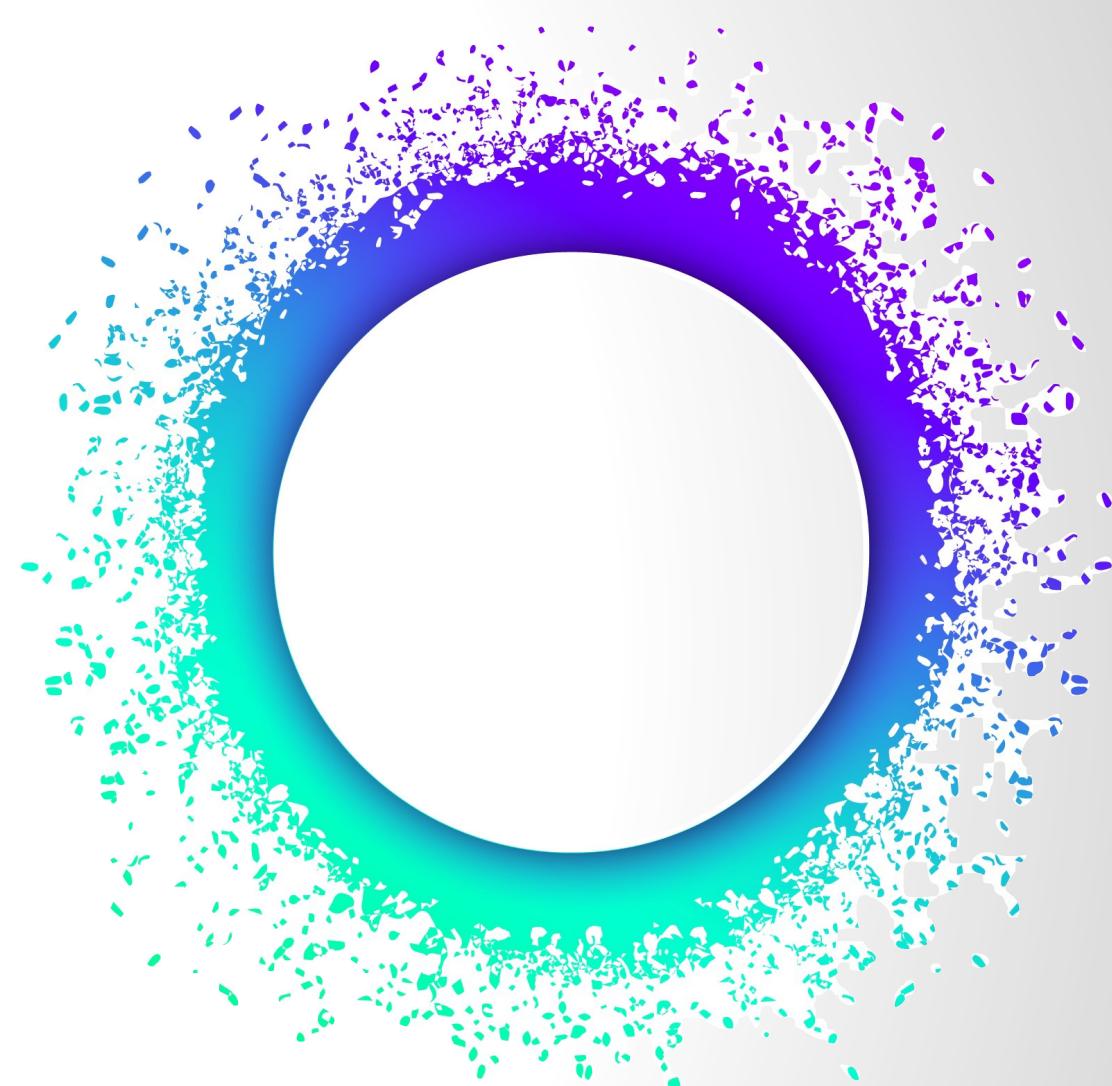
Nessuna proprietà  
centrale

Presenza di una  
criptovaluta nativa

I dati restano in mano  
agli utenti

# Cos'è Holochain?

- Un **framework peer-to-peer** per la condivisione, l'archiviazione e la convalida dei dati.
- Progetto **open source** scritto nei linguaggi Rust e WebAssembly.
- Creata da **Arthur Brock** e **Eric Harris-Brown**, è sviluppata da una community a partire da marzo 2017 (versione alpha).
- È disponibile da **novembre 2018**, mentre a settembre 2020 è stata annunciata la nuova versione chiamata **Holochain RSM**.



H O L O C H A I N

# Cos'è Holochain?



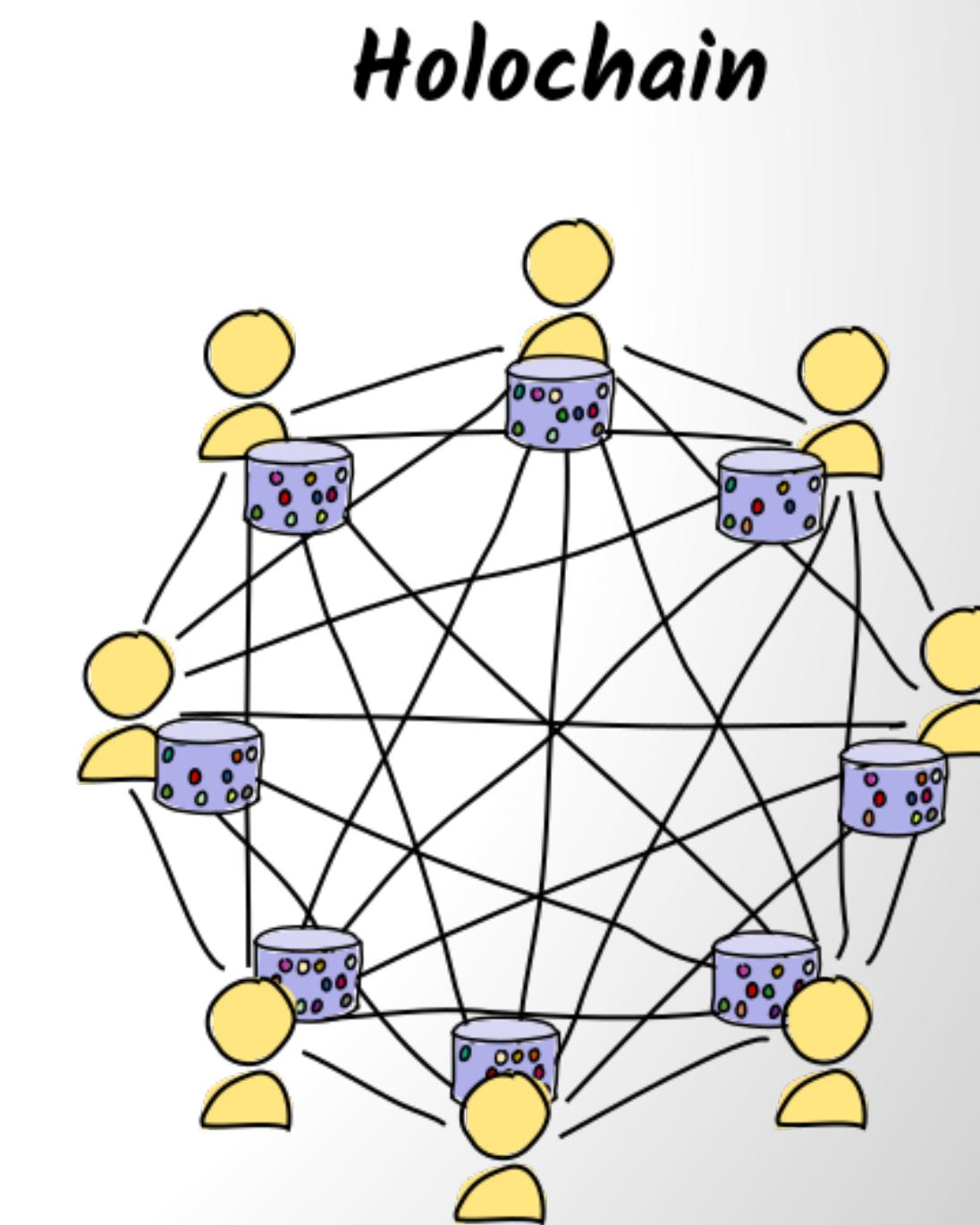
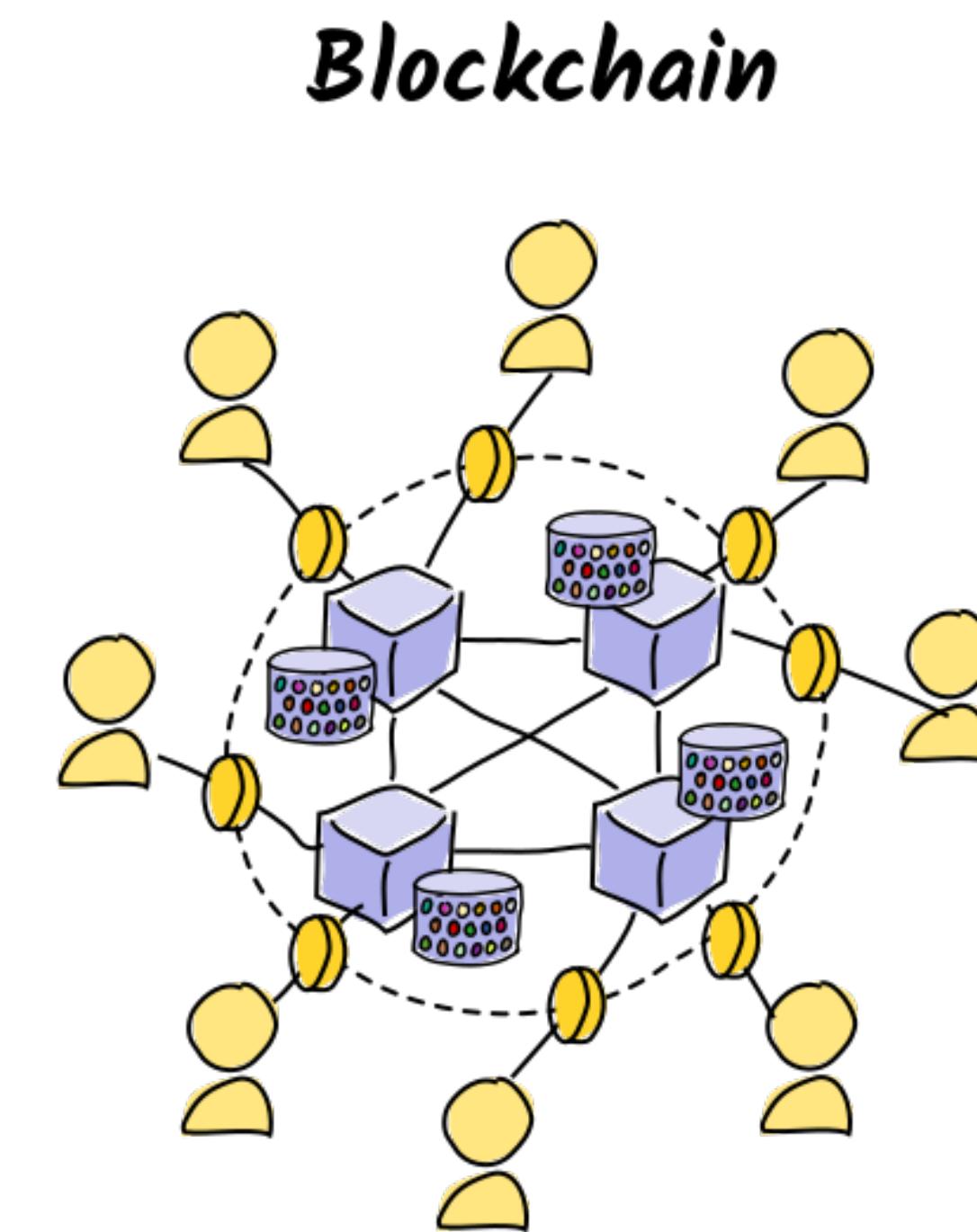
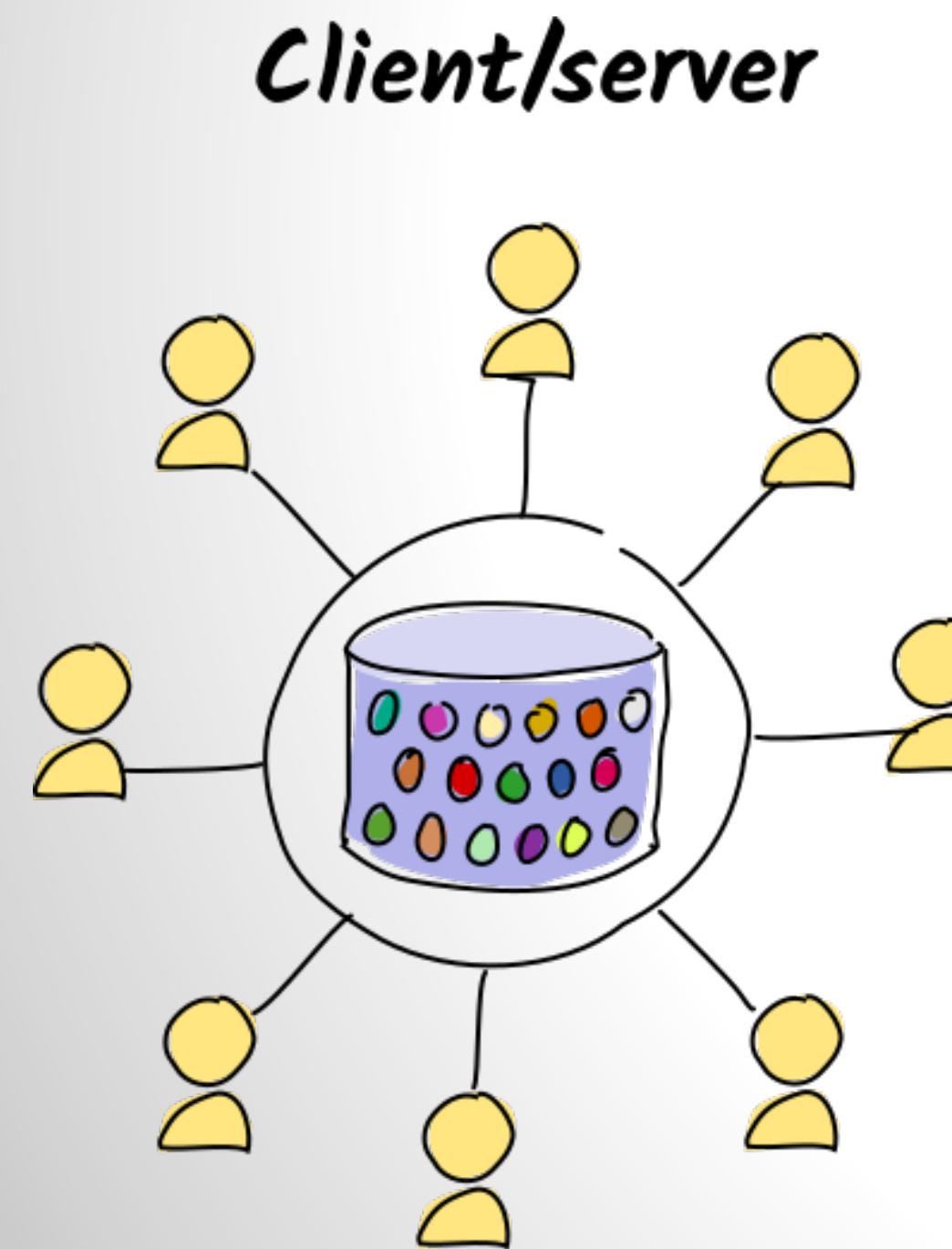
- Sistema post-blockchain
- Efficiente a livello energetico
- Piattaforma applicativa decentralizzata
- Sistemi di consenso agent-centric
- Ogni nodo ha il proprio ledger
- Indipendenza e/o interazione
- Permette la creazione di hApp
- Ogni utente controlla i propri dati
- I dati non sono venduti o esposti a terzi

# Holochain vs Blockchain

	Blockchain	Holochain
Approccio	Data-centric, unico set di dati globale condiviso su tutti i nodi.	Agent-centric, consente ai nodi di agire indipendentemente o in coordinazione.
Uso energetico	Bitcoin consuma più dello 0,1% dell'elettricità mondiale per alimentare meno dello 0,0001% del denaro mondiale.	Non c'è mining, non sono necessari processori ma bastano computer o telefoni a basso consumo.
Volume delle transazioni	Bitcoin ed Ethereum elaborano una manciata di transazioni al secondo.	Si prevede che supererà la rete Visa che ha un massimo di 56.000 transazioni al secondo.
Scalabilità	Seri limiti di scalabilità per la sincronizzazione di un libro mastro globale attraverso molti nodi.	Con un DHT distribuito il carico di transazione per nodo diventa più leggero al crescere della rete.
Piattaforma	Funziona efficacemente solo con piattaforme di mining speciali e algoritmi dispendiosi.	Può essere eseguito anche su un Raspberry Pi o uno smartphone.
Efficienza computazionale	$O(n*m)$ per convalidare le transazioni.	$O(n/m*\log m)$ per convalidare le transazioni.
Effetti del consenso	Si centralizza il potere, i ricchi diventano più ricchi. Il proof-of-work porta a un sovraccarico computazionale in continua crescita.	Nessun mining e consenso. Non vulnerabile agli attacchi di maggioranza. Fidarsi del codice sul proprio nodo e validazione della entry.

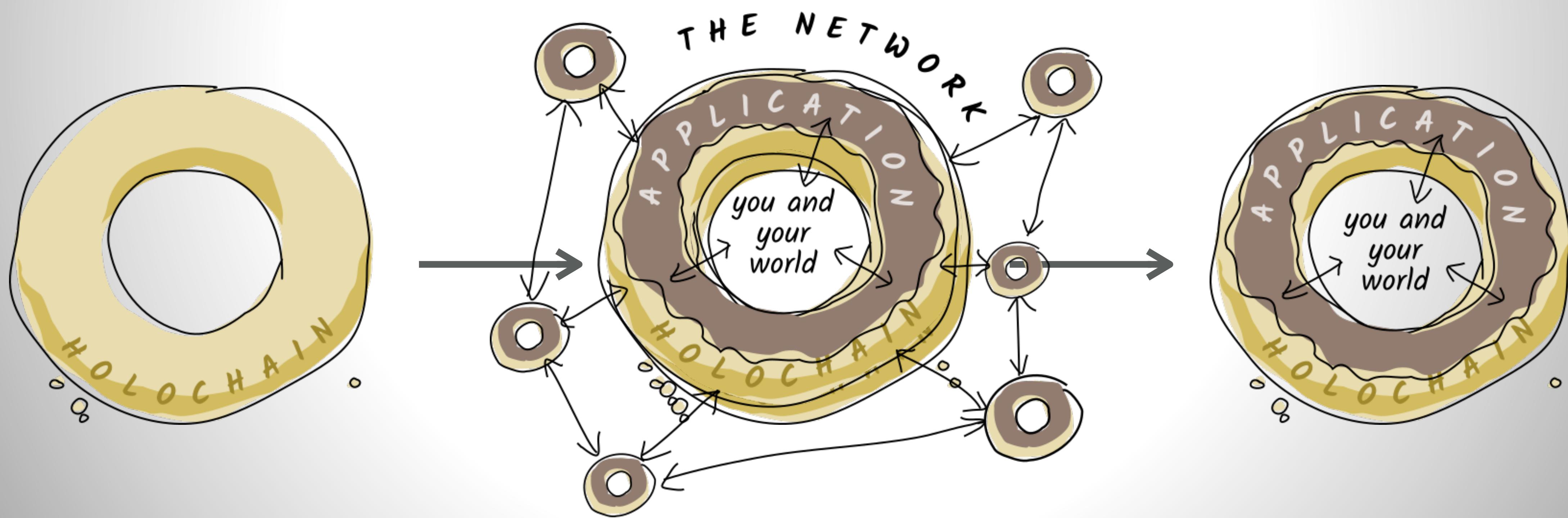
# Le basi di Holochain

- È possibile creare e distribuire in modo sicuro applicazioni che **non utilizzano un server centrale**. Ogni utente esegue l'applicazione sul proprio **dispositivo**, crea e archivia i propri dati e comunica direttamente con altri utenti **senza supervisione centrale o consenso globale**.



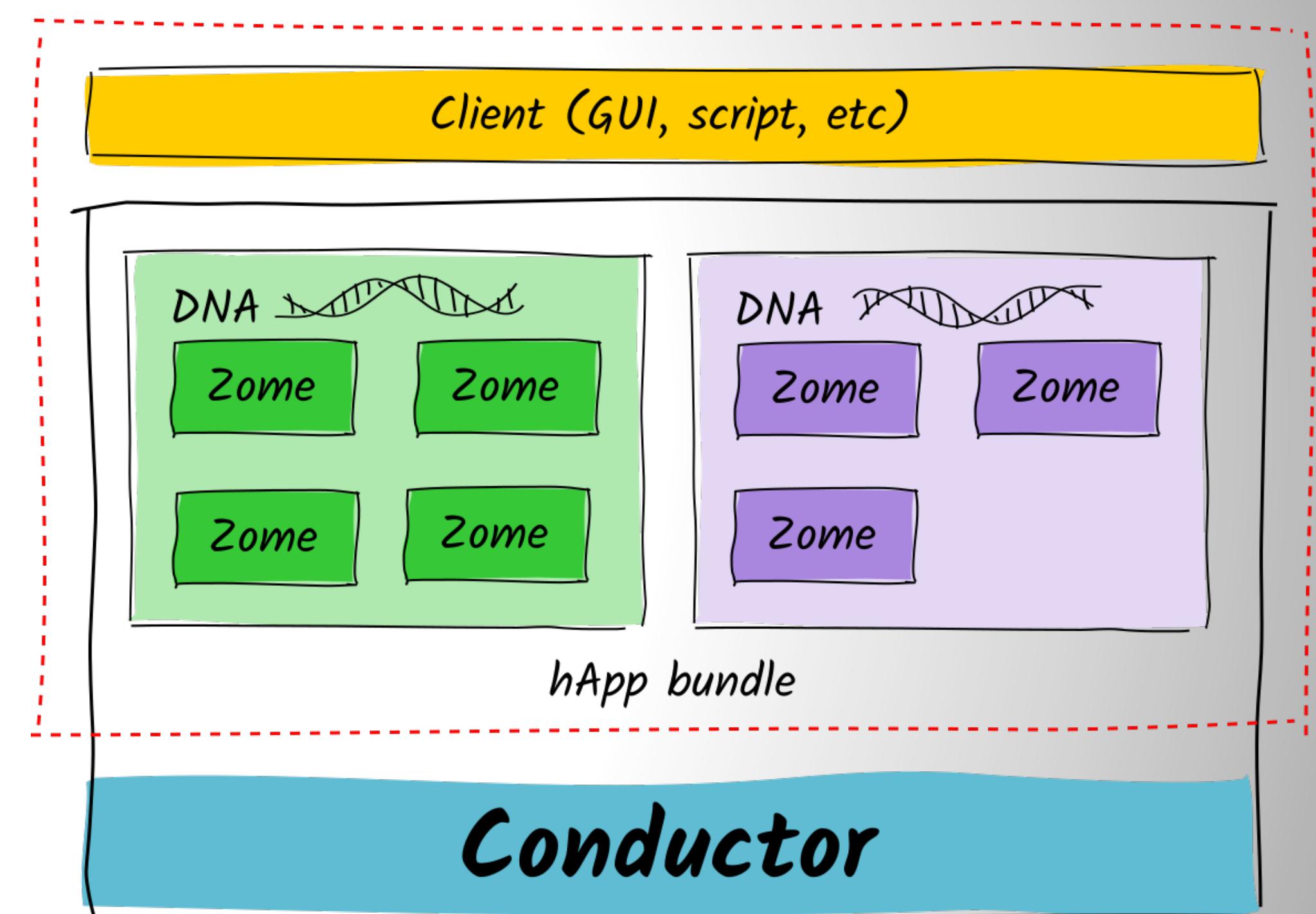
# Architettura di Holochain

- Le applicazioni realizzate con Holochain sono altamente **modulari** in termini di funzionalità e architettura. Ciò semplifica la condivisione del codice e la composizione di pezzi più piccoli in pezzi grandi. Ogni applicazione Holochain (chiamata hApp) ha il proprio **set di regole**, rete privata e database distribuito.



# Struttura delle hApp di Holochain

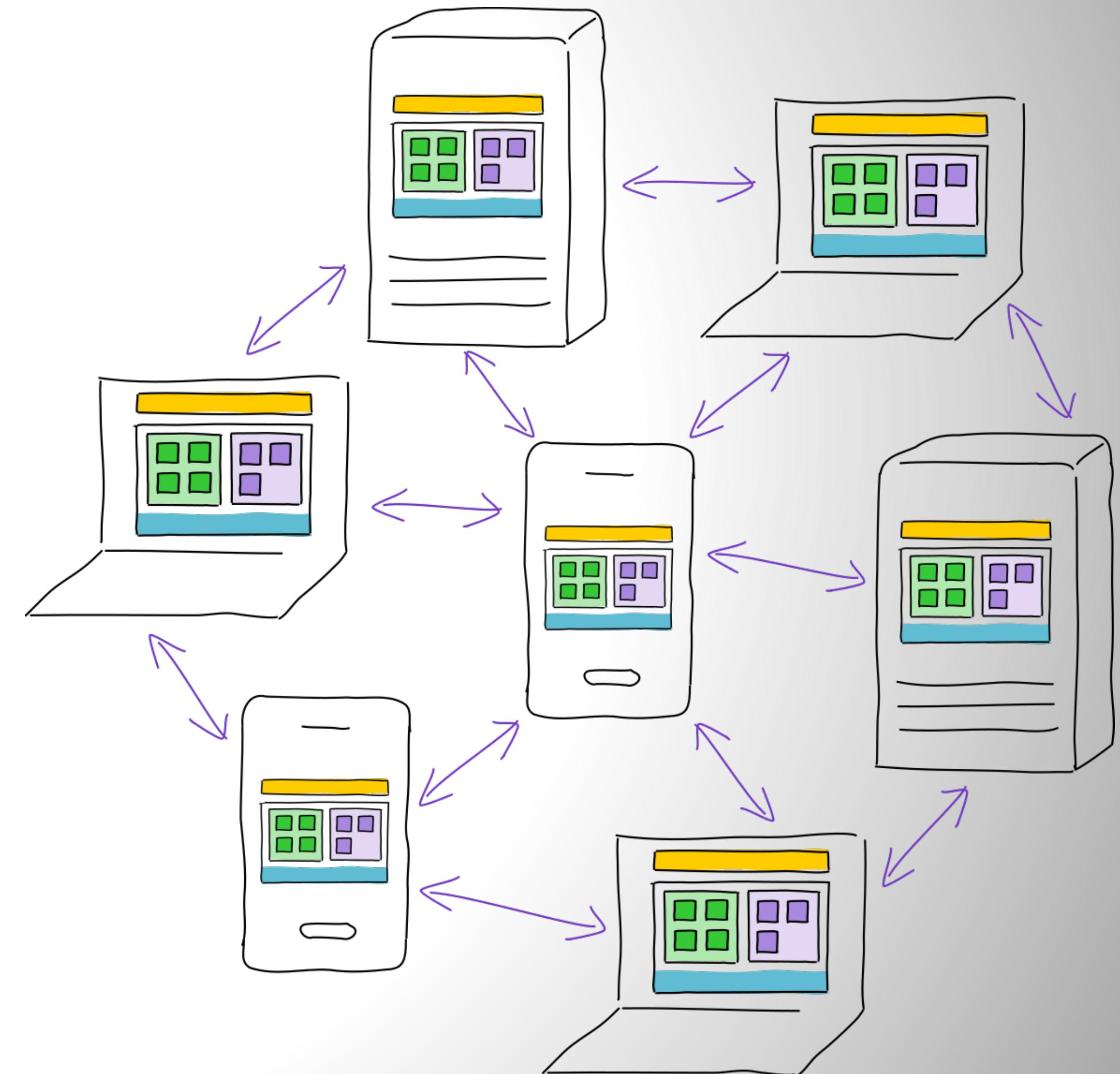
- I moduli di codice chiamati **zome** definiscono la logica della hApp e le comunicazioni tra gli utenti.
- Più zomes sono uniti in un **DNA** che definisce le funzionalità di base per un'applicazione. Gli zomes possono parlarsi attraverso le loro API pubbliche.
- Più DNA sono combinati in un **bundle** di DNA che specifica tutte le funzionalità necessarie.
- Un **client** sul dispositivo dell'utente, con il suo **front-end** (che può essere scritto in qualsiasi linguaggio), comunica con le API pubbliche degli zomes tramite un'interfaccia **Remote Procedure Call**.
- Tutti i DNA sono ospitati nel **conductor** dell'utente, una sorta di **server** che esegue il codice, oltre a gestire dati e connessioni.



# La rete Holochain

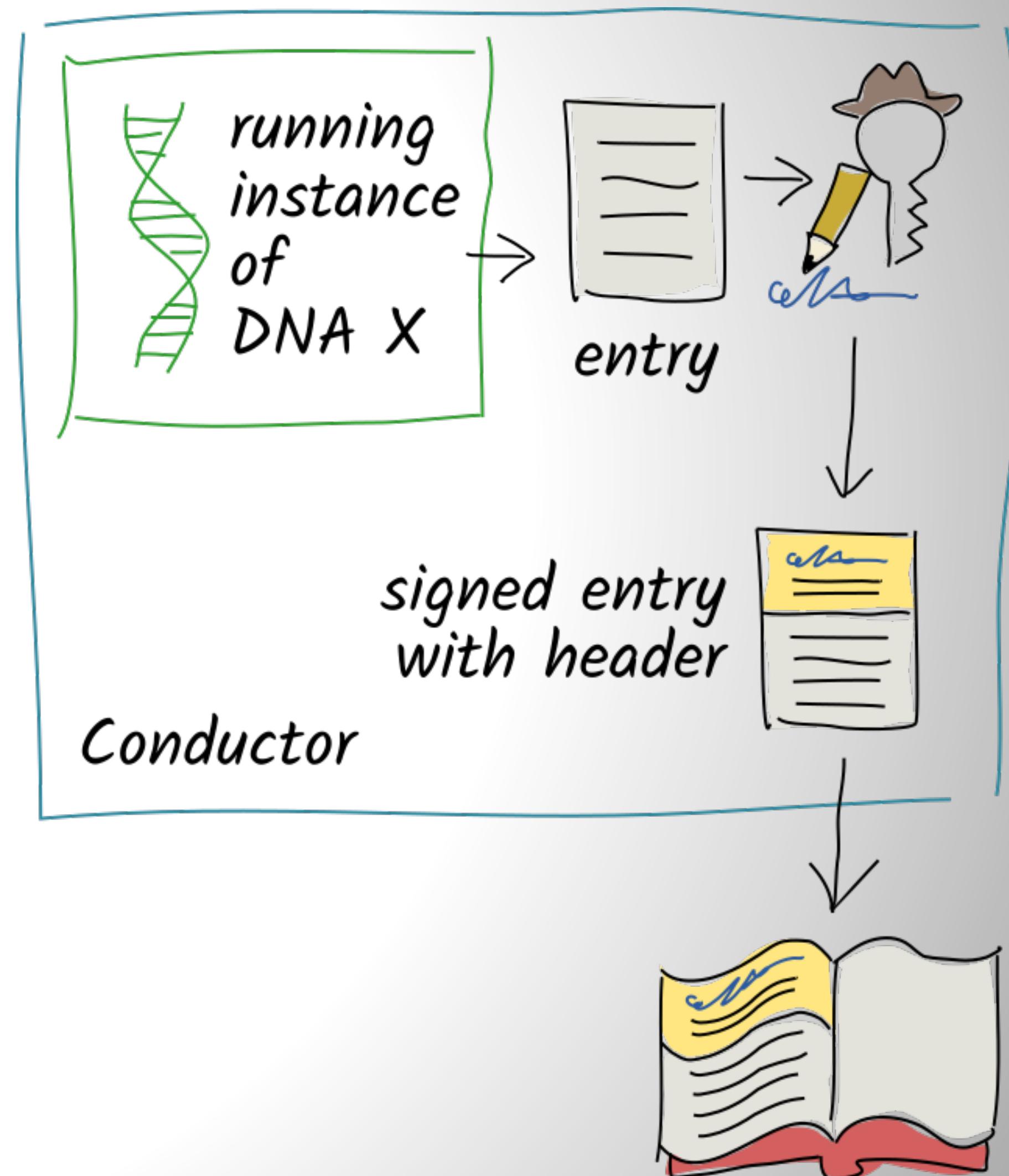
- Ogni utente è un nodo in una rete di agenti peer-to-peer che usano la stessa hApp.
- Oltre ad essere **responsabile** della propria computazione e archiviazione, ogni utente possiede la propria copia di:
  - **Front-end (client)**
  - **Back-end (DNA)**
  - **Server (conductor)**

**Agent-centric computing**



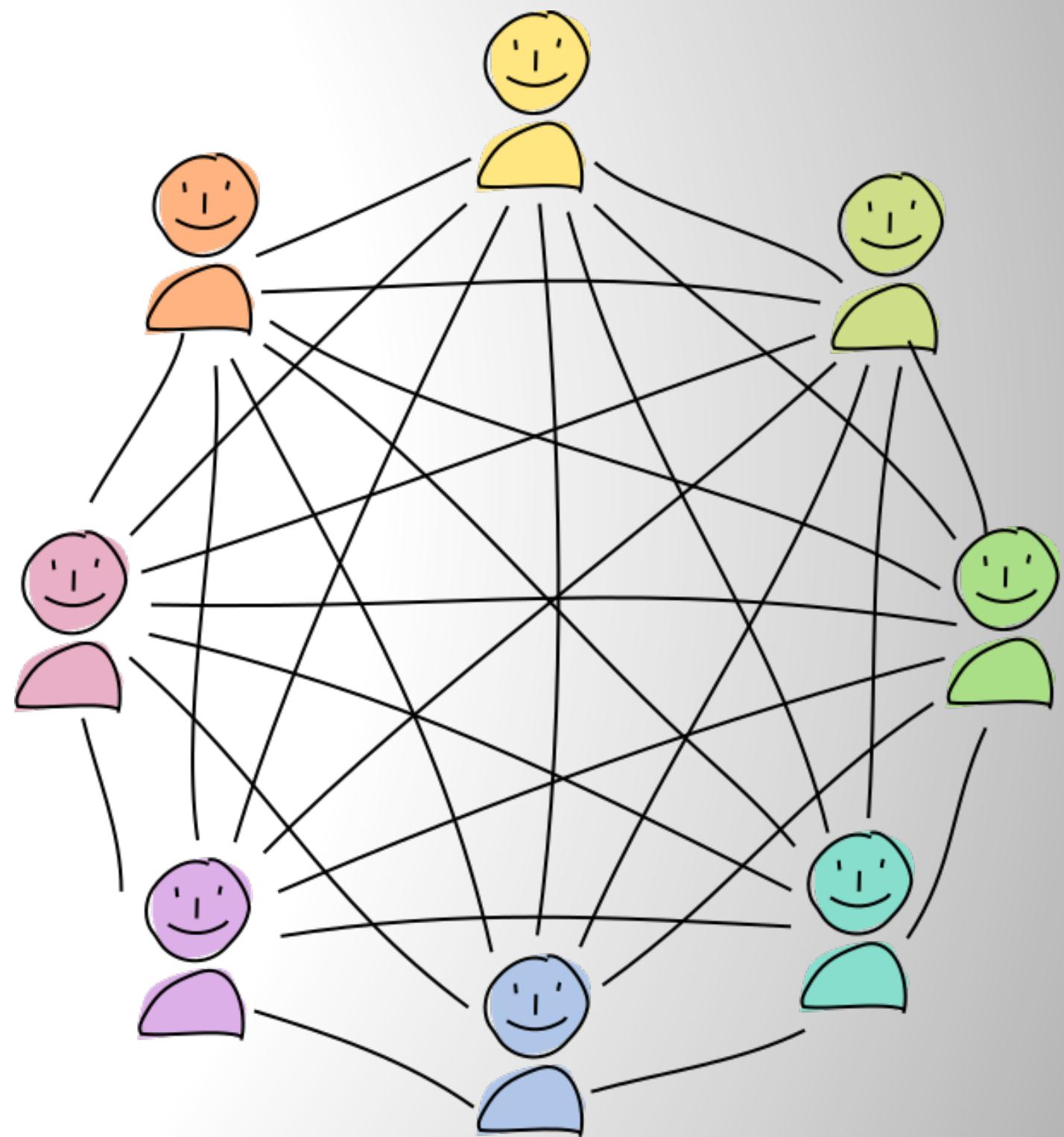
# Dati privati sulla local source chain

- Unendosi ad una rete si crea la propria identità generando una coppia di chiavi pubblica e privata.
- Ogni utente **crea e archivia i propri dati** in un diario chiamato **source chain**, memorizzato nel proprio dispositivo.
- Ogni **entry** è firmata crittograficamente dal suo autore, è **immutabile** una volta scritta e l'header contiene l'hash dell'header precedente.
- La source chain inizia con le **genesis' entries**:
  - Hash del DNA: l'utente accetta le "regole del gioco".
  - Agent ID: contiene la chiave pubblica dell'utente.



# Dati pubblici sulla DHT

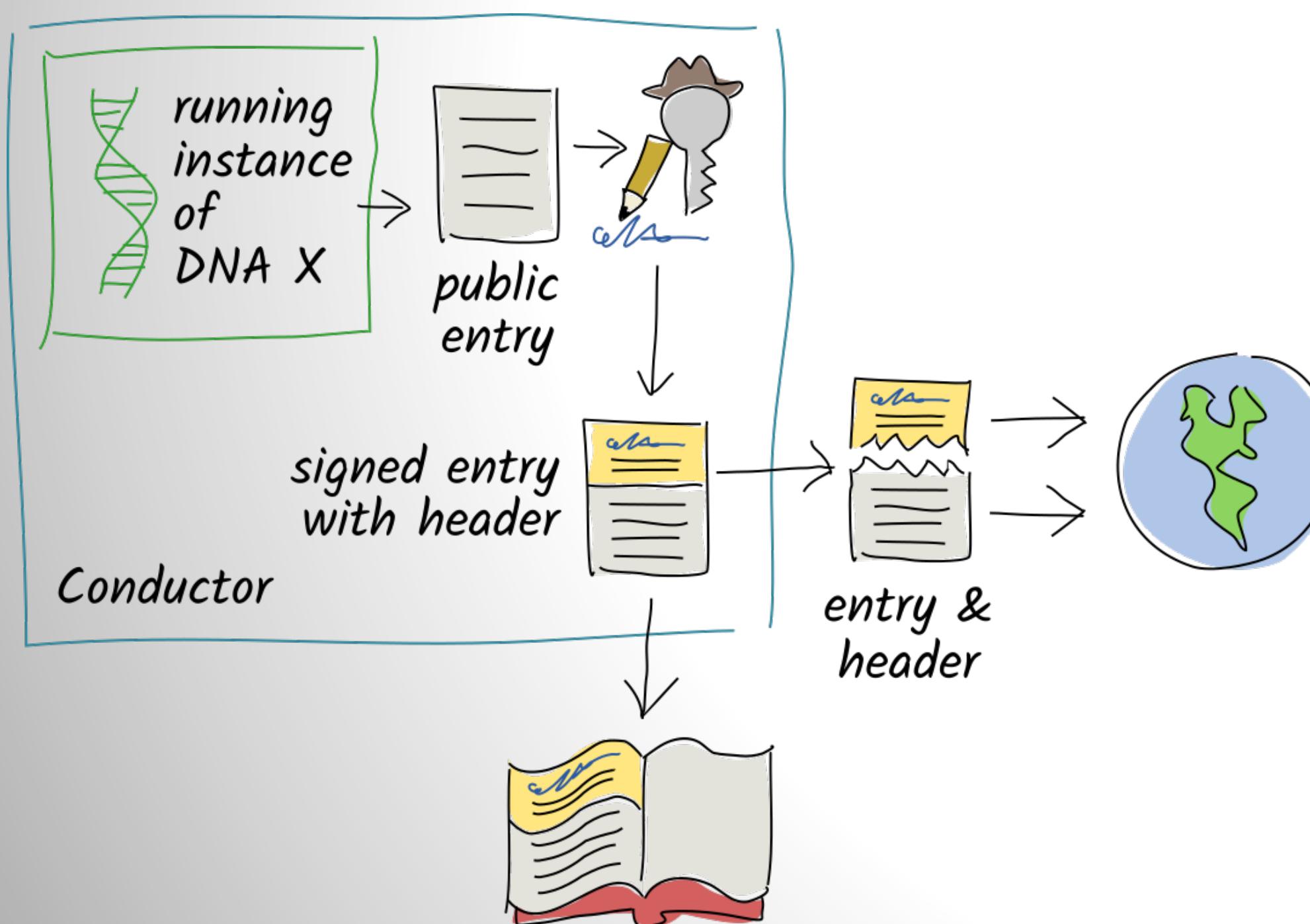
- Gli agenti condividono le loro **chiavi pubbliche**, gli **header** della source chain e le **entry pubbliche** con gli altri nodi in una **tabella hash distribuita chiave/valore (DHT)**.
- Ciò fornisce **ridondanza** e disponibilità per i dati anche quando il proprietario è offline.
- Inoltre offre alla rete il potere di **rilevare la corruzione**, poiché gli **headers** della source chain sono **pubblici**.
- I nodi nella Holochain fanno **gossip** tra loro su nuovi peer e nuove entry di dati.
- Nodi e dati vivono entrambi nello stesso spazio degli **indirizzi**.



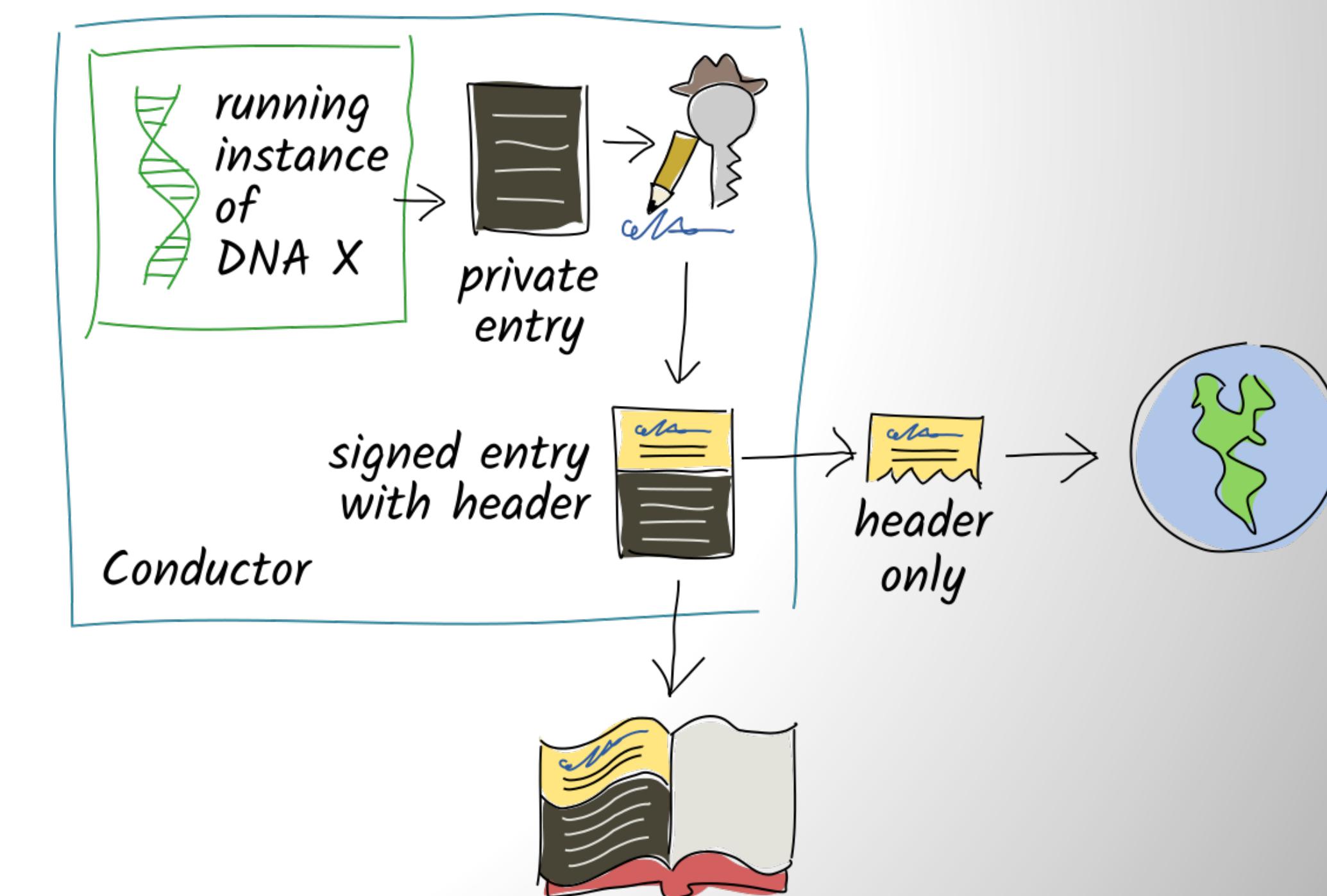
# Dati pubblici sulla DHT

- Ogni nodo conosce i suoi **vicini** e ha alcune **conoscenze lontane**.
- Con il commit di una **entry privata**, essa rimane nella source chain ma il suo **header** è condiviso.

Entry pubblica:

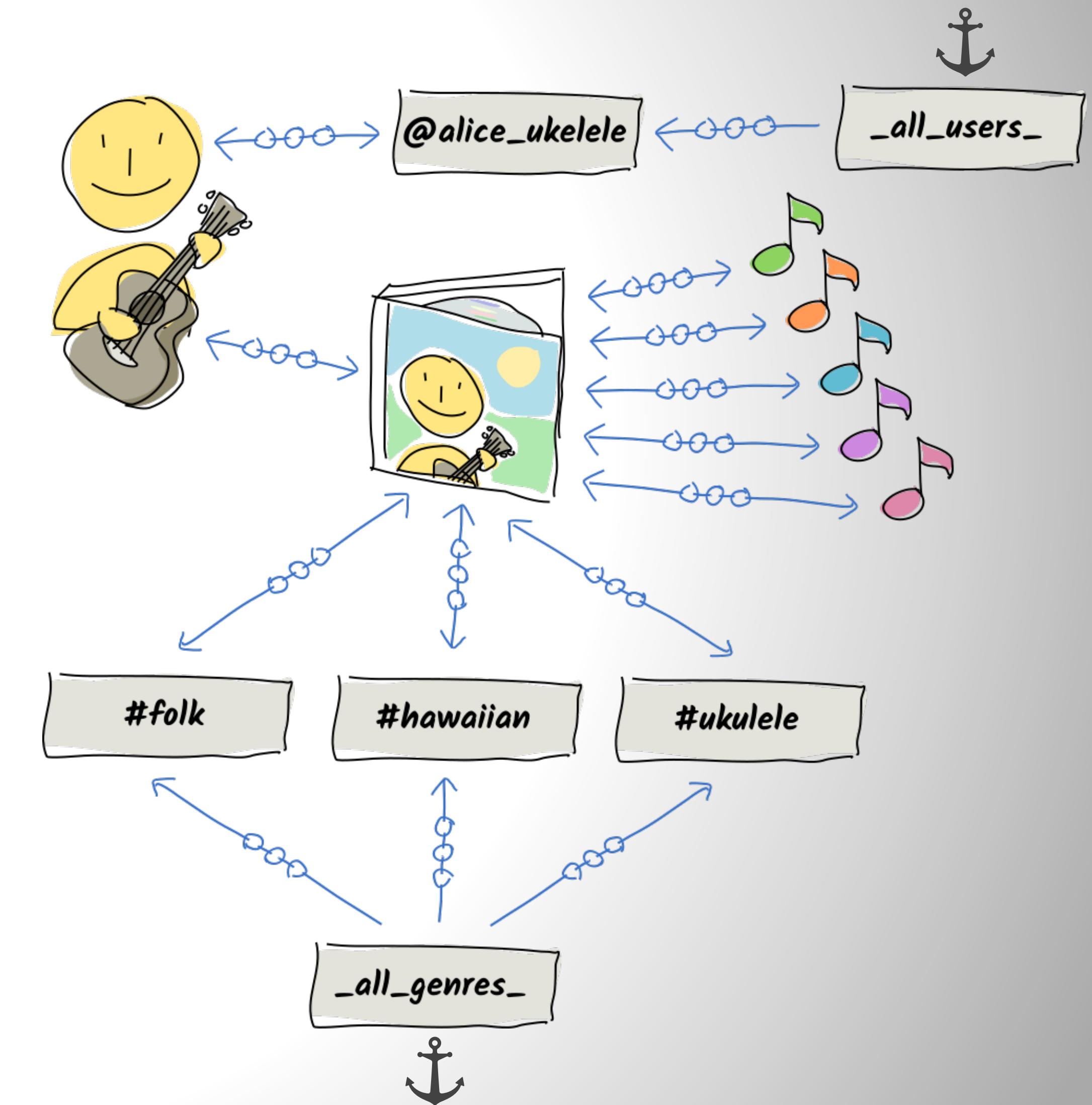


Entry privata:



# Collegare i dati insieme

- Per trovare un dato bisogna conoscerne l'hash. Ma per generare un hash è necessario il dato stesso.
- È possibile **collegare due entry qualsiasi insieme**. Ciò consente di collegare cose note (agent ID e ancore) a cose sconosciute, che poi diventano cose conosciute e così via.
- Il link è memorizzato nel DHT come **metadato della entry che collega**.

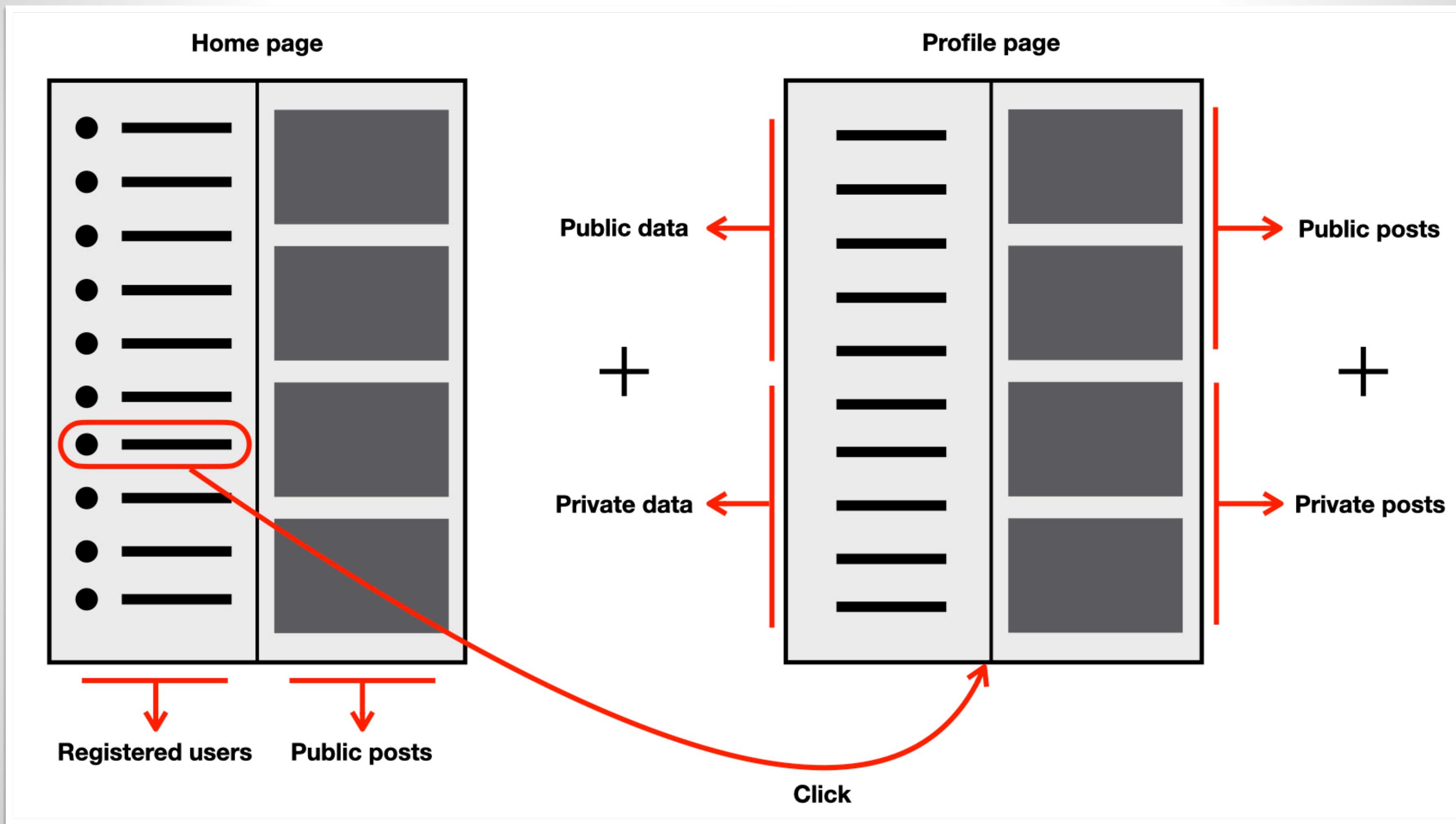


# Il nostro progetto: Holobook

- Abbiamo realizzato un'hApp utilizzando Holochain. Si tratta di un social network decentralizzato che abbiamo chiamato **Holobook**.
- L'utente si **registra**, può pubblicare **post** pubblici o privati, inserire **informazioni** pubbliche o private e aggiungere altri utenti alla **lista degli amici più stretti**. Tali utenti sono in grado di vedere i dati privati e il loro funzionamento è molto simile alla lista degli amici più stretti nelle Instagram Stories.



# Holobook: schema dell'interfaccia grafica



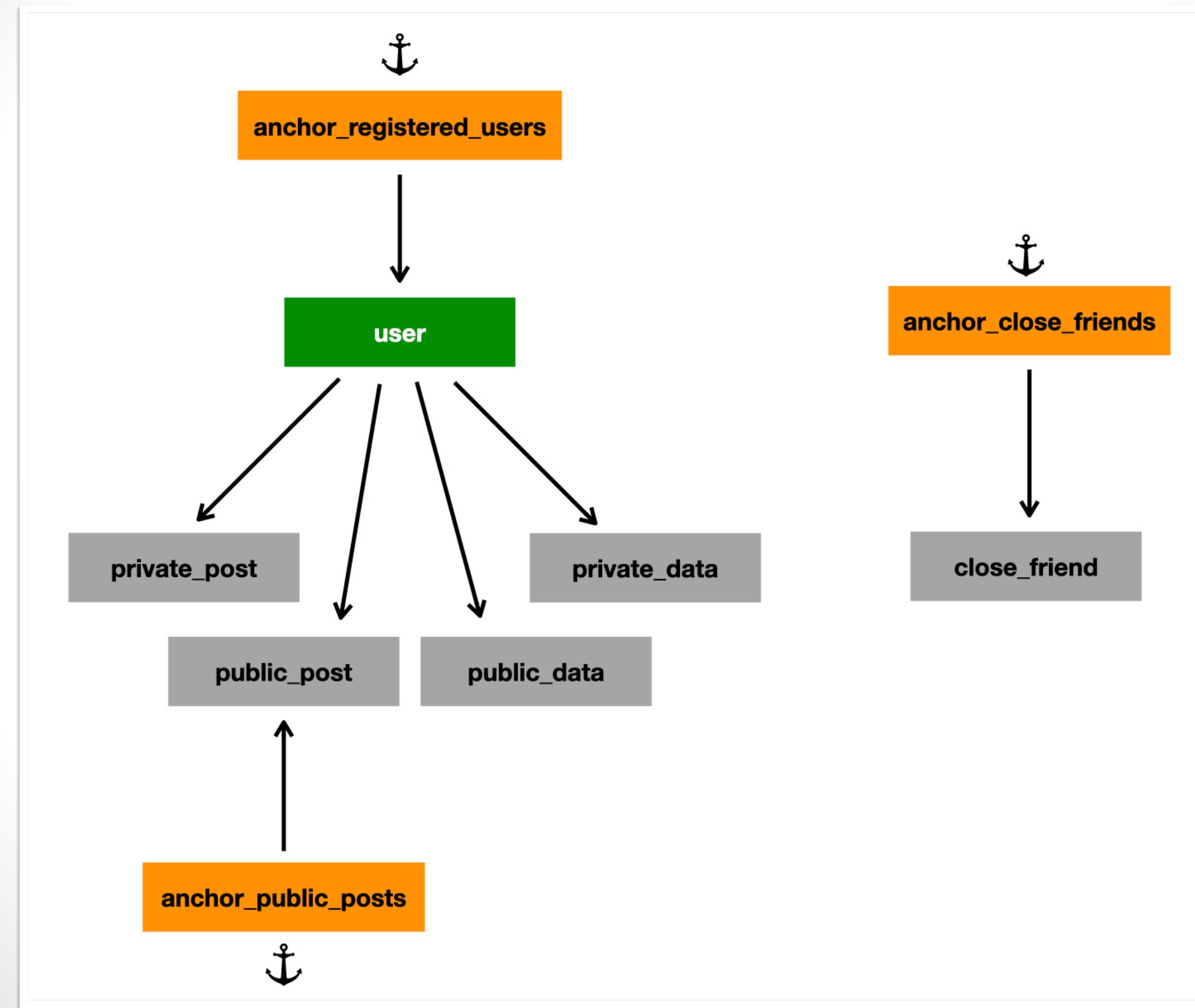
# Demo di Holobook

- Screenshot: schermata di login, schermata home e schermata del profilo utente:

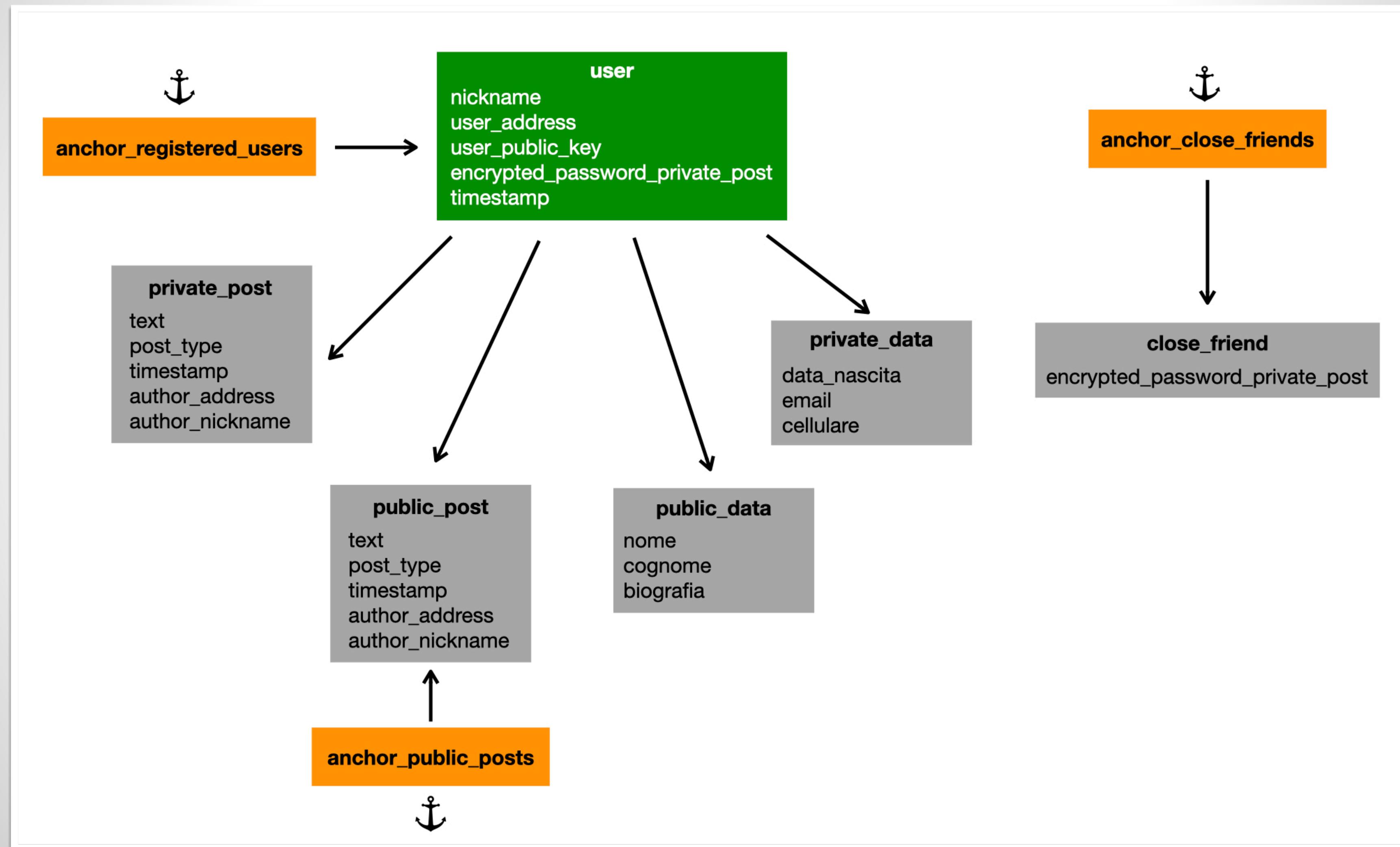
The image displays three screenshots of the holobook application interface:

- Login Screen:** Shows a blue header with the "holobook" logo. Below it is a form with a "Password" input field and a blue "Accedi" button.
- Home Screen:** Shows a blue header with the "holobook" logo. Below it is a "Pubblica post" (Public post) section with a text input field and a "Pubblica" (Publish) button. It also features sections for "Utenti registrati" (Registered users) showing "Carl", "Bob", and "Alice", and "Post pubblici" (Public posts) showing three posts from Carl, Bob, and Alice.
- User Profile Screen:** Shows a blue header with the "holobook" logo and a "Il mio profilo (Bob)" link. Below it is a "Profilo di Alice" (Profile of Alice) section with a "Torna alla Home" (Return to Home) link. It shows Alice's information: Nome: Alice, Cognome: Dawson, Biografia: *Mi piacciono i film e la musica!*, and Data di nascita: 13/04/1995. To the right is a "Post" section showing three posts from Alice with timestamps and content.

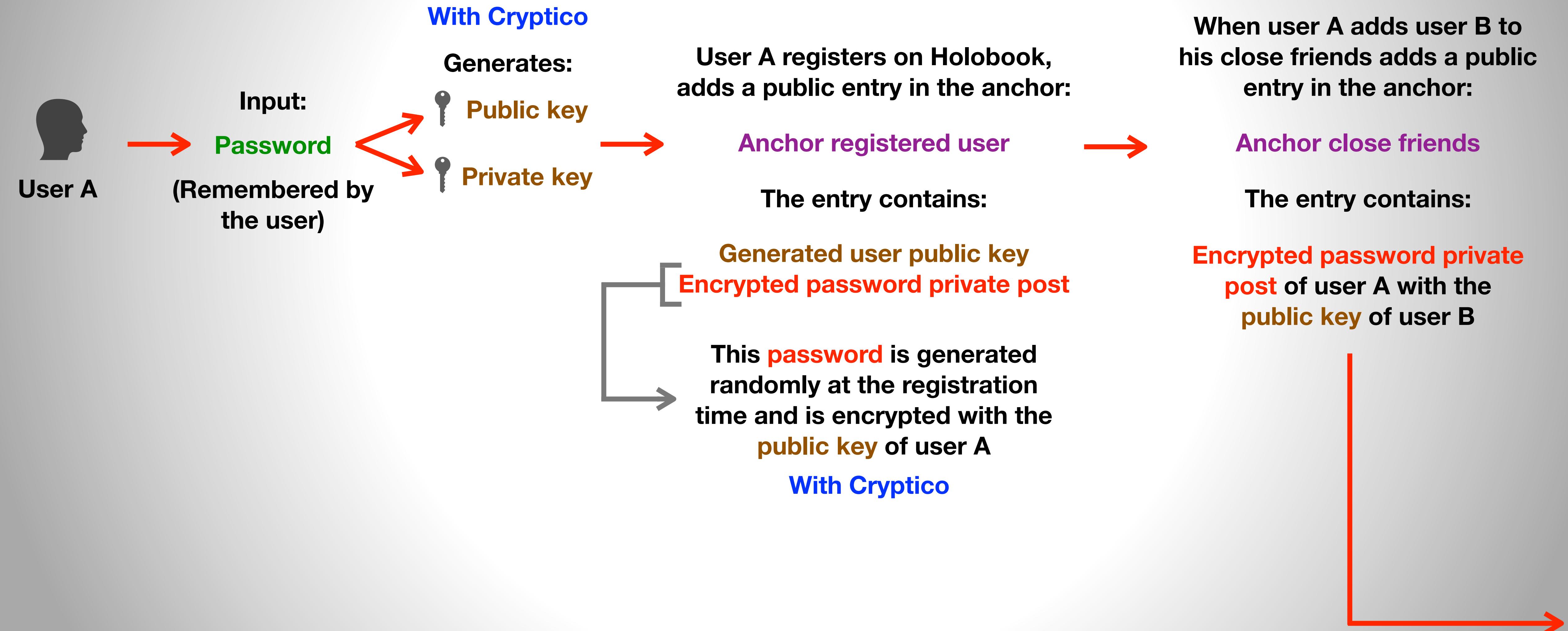
# Holobook: schema entry generale



# Holobook: schema entry dettagliato



# Holobook: schema crittografia



# Holobook: schema crittografia

When user A creates private post  
(or information) publishes it as a public  
entry in the DHT linked to his Agent ID

The entry contains:

Encrypted post with the  
**password private post**

With CryptoJS

Friendship relationship (link tag):

**Agent ID user A -> Agent ID user B**

When user B goes to the user's A  
profile checks if himself is a  
closer friend of user A

If no: view only the  
public post and  
informations of user A



If yes: get the private posts and  
informations of user A and decrypt  
them with the **password private post**  
decrypted using the **private key of**  
user B

With Cryptico and CryptoJS

# Estensioni future, osservazioni e conclusioni

- **Estensioni future:** altre funzionalità che potrebbero essere implementate sono i **mi piace** e i **commenti** ai post, collegandoli direttamente alla entry del post a cui si riferiscono tramite un link type specifico (es: like e comment).
- **Osservazioni:** la **chiave privata generata da Holochain** non è recuperabile, per questo si è utilizzata la libreria Crypto. Altro problema è stata l'**eliminazione automatica delle entry private** se il nodo va offline, per questo si sono utilizzate entry pubbliche crittografate. In entrambi i casi si tratta di problemi strutturali della versione di Holochain utilizzata.
- **Conclusioni:** Holochain è un framework peer-to-peer distribuito molto promettente, in particolare per la **privacy** dei dati, il ridotto **impatto energetico** e la **capacità computazionale** necessaria.



# Bibliografia e sitografia

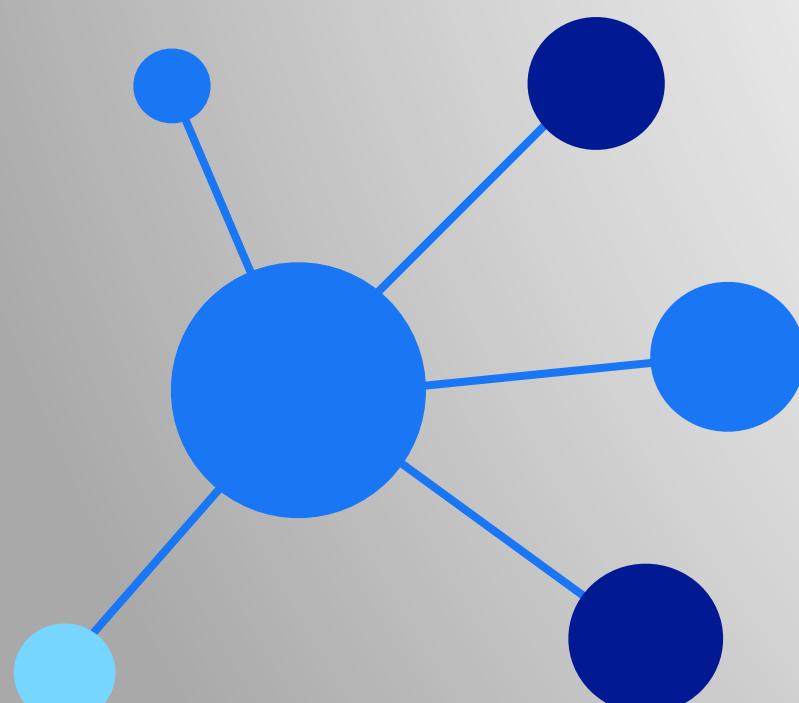
- **Holochain:** <https://holochain.org>
- **Holo:** <https://holo.host>
- **Documentazione Holochain:** <https://developer.holochain.org>
- **Wiki Holochain:** <https://github.com/holochain/holochain-proto/wiki>
- **White paper Holochain:** <https://github.com/holochain/holochain-proto/blob/whitepaper/holochain.pdf>
- **Green paper Holo:** <https://files.holo.host/2018/03/Holo-Green-Paper.pdf>
- **Hackernoon:** <https://hackernoon.com/wtf-is-holochain-35f9dd8e5908>
- **Social decentralizzati:** <https://cimoinfo.com/social-network-decentralizzati-overview/>
- **Barbara Guidi, “P2P Architectures for Distributed Online Social Networks”.** IEEE, 2013:  
<https://ieeexplore.ieee.org/document/6641493>

Alma Mater Studiorum - Università di Bologna  
Scuola di Scienze, Dipartimento di Informatica - Scienza e Ingegneria  
Corso di Laurea Magistrale in Informatica  
Curriculum in Informatica per il Management  
A.A. 2019/2020

---

# Grazie per l'attenzione

## Progetto di Sistemi Peer-to-Peer



Lorenzo Biagio Lanzarone

Cristian Romanello

