Blockchain for Enterprise:
A Security & Privacy Perspective through Hyperledger/fabric

Elli Androulaki
Staff member, IBM Research, Zurich

Workshop on cryptocurrencies
Athens, 06.03.2016

Blockchain systems

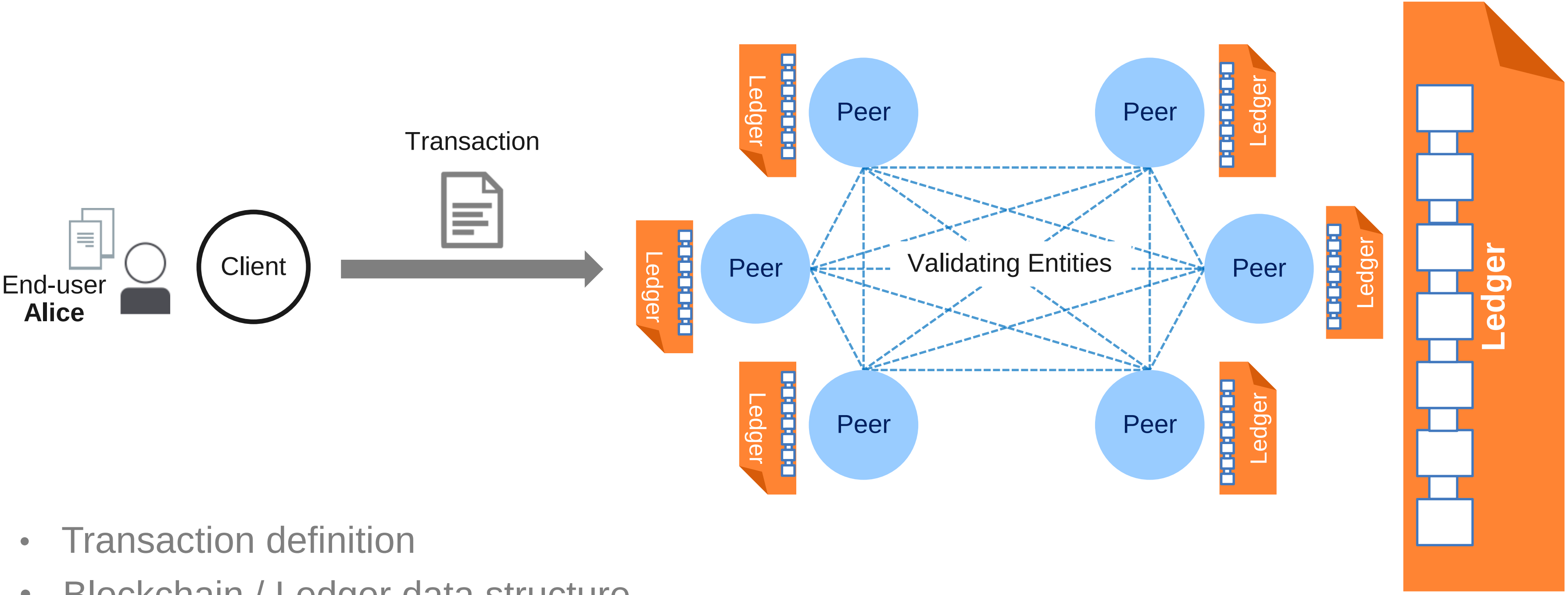
- Introduced in 2008 [Bitcoin08]
- Open to be used by anyone
- Decentralized networks to decide on the order & **validity** of **transactions** that are announced in it
  - Blockchain/Ledger of announced & validated transactions
  - Mechanism/protocols to extend the ledger
- Occasionally, with their own currency (e.g., BTC, ETHER)
- Emerging:
  – Integrated in multiple businesses around the globe
  – Market sizes of Billions USD
  – An ecosystem established around them

ripple

MaidSafe

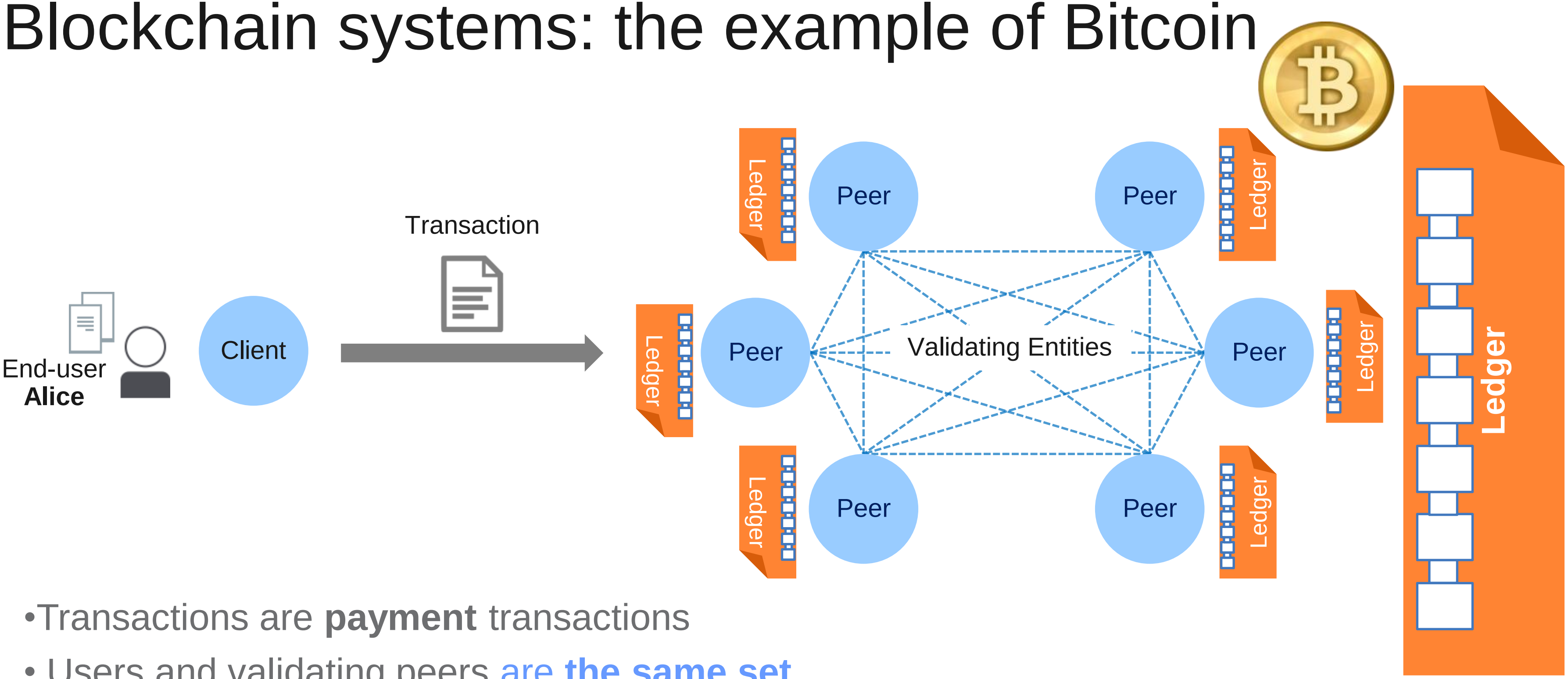
Digital Asset Holdings

Blockchain systems: concepts of interest



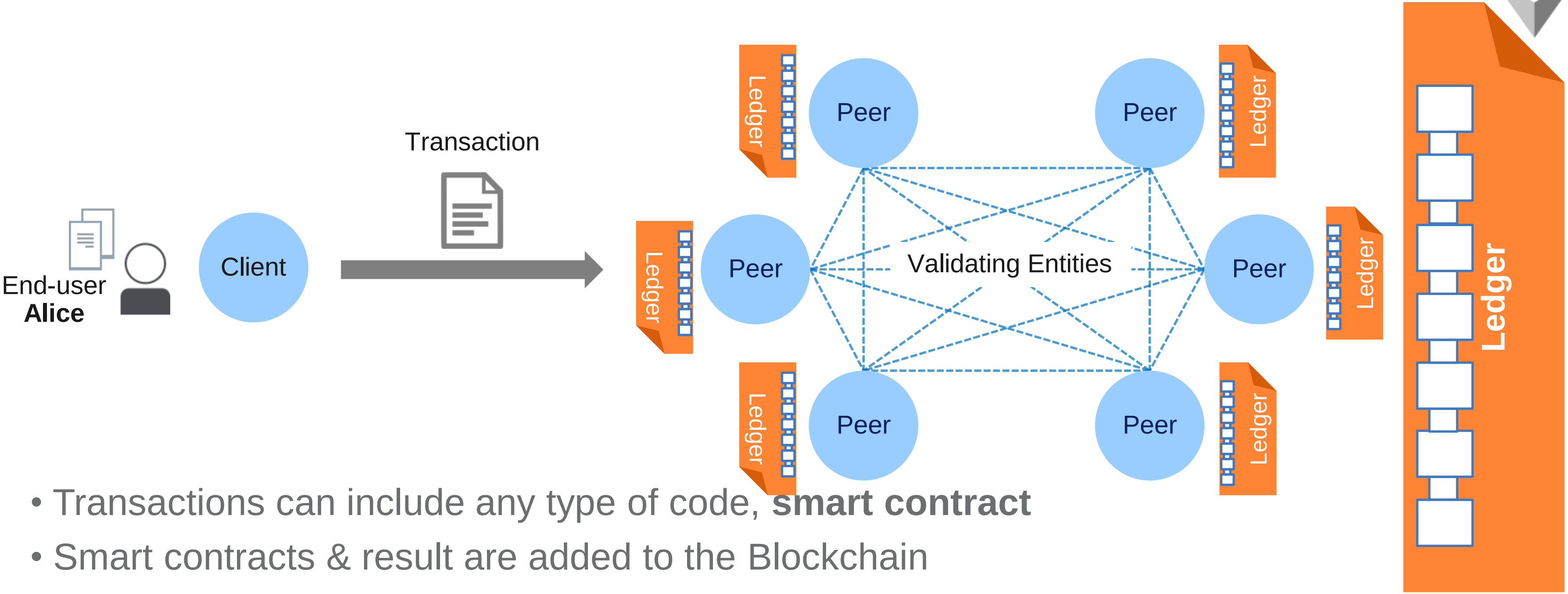
- Transaction definition
- Blockchain / Ledger data structure
- Participant identities
- Underlying agreement (aka consensus) protocol
- Motivation mechanisms for proper functionality of the system

Blockchain systems: the example of Bitcoin



- Transactions are **payment** transactions
- Users and validating peers are **the same set**
- Clients use **self-generated** pseudonyms
- Miners “vote” with their computing power the executed result
- Motivation for good behavior through the generation of BTC coins

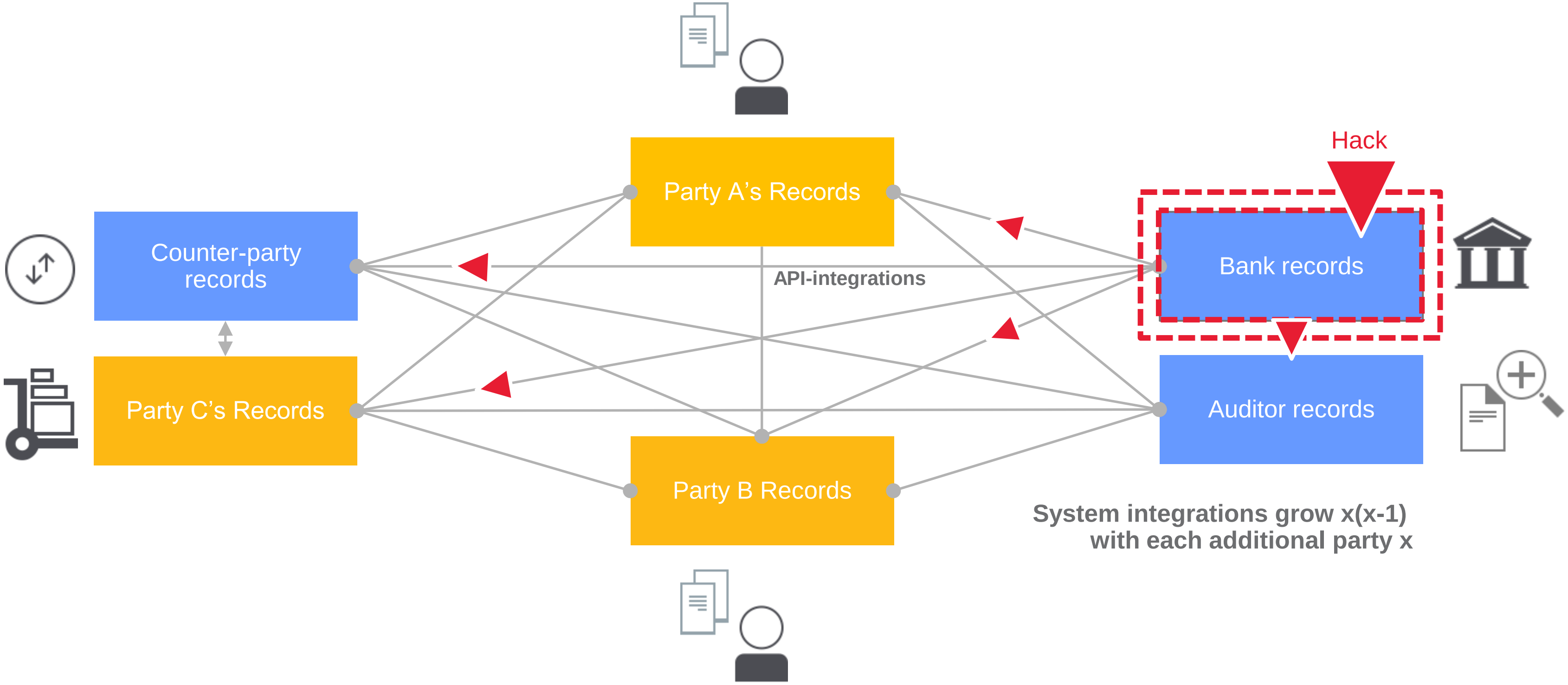
Blockchain systems: the example of Ethereum



- Transactions can include any type of code, **smart contract**
- Smart contracts & result are added to the Blockchain
- Users and validating peers are **the same set**
- Clients use self generated pseudonyms
- Miners “vote” with their computing power
- Motivation for good behavior through the generation of ETHER coins

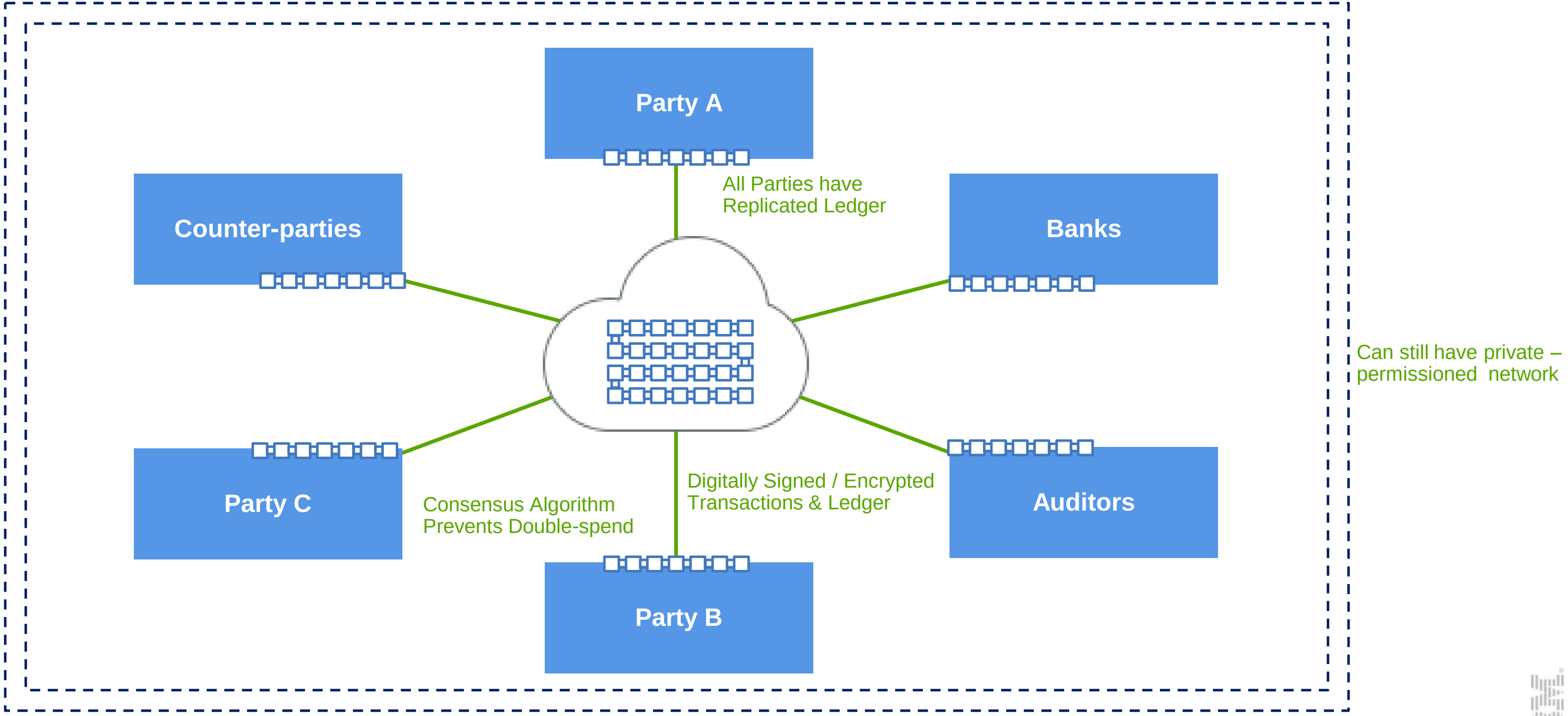
Blockchain for enterprise?

Problem: Electronic networks that transfer the ownership of assets between parties according to business rules are **inefficient**, **expensive** and **vulnerable**.



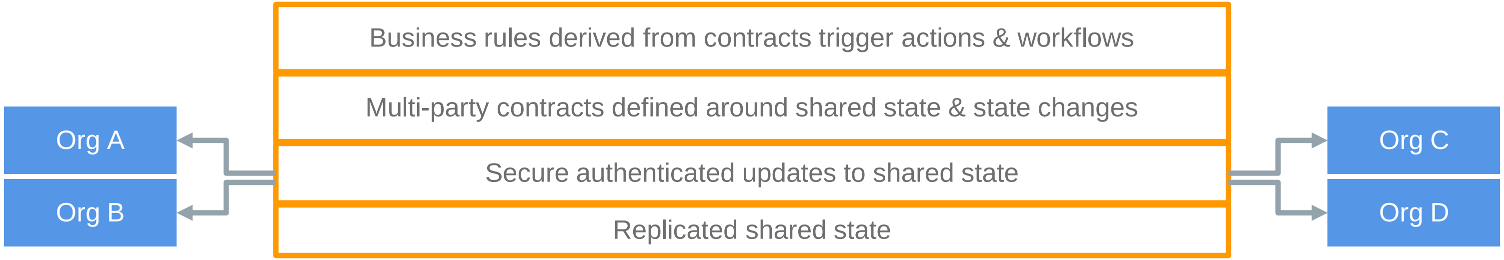
System integrations grow x(x-1) with each additional party x

Solution: Blockchain networks — simpler; no centralized control points; spread risk = lowered costs; hardened inside (not just at the perimeter).



Blockchain: all we need!

- **Shared replicated ledger**: a peer-to-peer append-only transaction database that is replicated and shared across organizational boundaries/legal entities
- **Embedded crypto layer**: supporting secure authenticated verifiable multi-party transactions via tokenization, digital identity, digital signatures, and other
- **Business rules** (evolving to Smart Contracts): ability to specify business logic, embed it in the transaction database, and couple execution of the logic with transaction processing


Business rules derived from contracts trigger actions & workflows
Multi-party contracts defined around shared state & state changes
Secure authenticated updates to shared state
Replicated shared state
Org A
Org B
Org C
Org D

Benefits of Blockchain

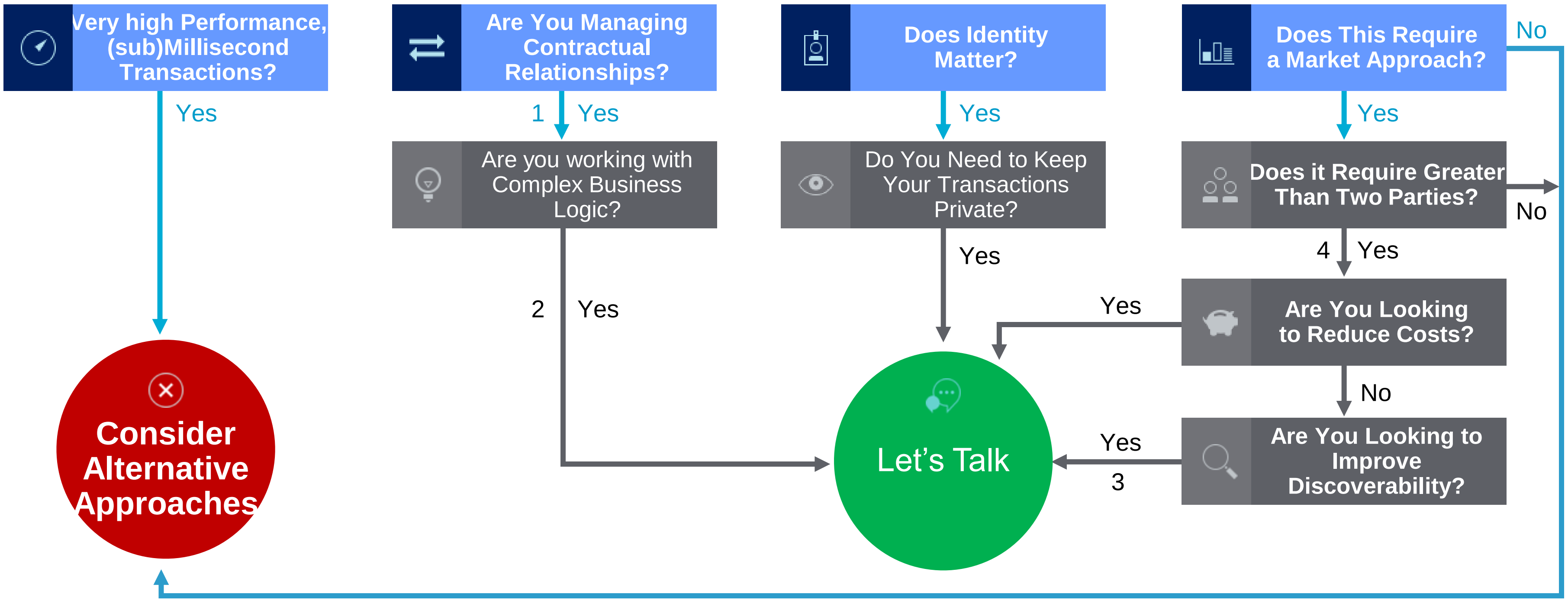
- Reduce costs and complexity
- Improve discoverability
- Automate trusted processes
- Ensure trusted record-keeping

"Over decades banks and other firms have built systems for themselves ... and then a collection of processes has emerged between the banks ... to make sure these systems are kept synchronized and are reconciled with each other."

"With shared or distributed ledgers perhaps we can imagine a world where participants share this infrastructure, so rather than everyone running their own systems that have to be reconciled, we can have ... an open platform that multiple firms can connect to.“

R. G. Brown

Blockchain: How to decide whether to use it?



Very high Performance, (sub)Millisecond Transactions?

Yes

Consider Alternative Approaches

Are You Managing Contractual Relationships?

1 Yes

Are you working with Complex Business Logic?

2 Yes

Does Identity Matter?

Yes

Do You Need to Keep Your Transactions Private?

Yes

Let's Talk

Does This Require a Market Approach?

No

Yes

Does it Require Greater Than Two Parties?

No

4 Yes

Are You Looking to Reduce Costs?

Yes

No

Are You Looking to Improve Discoverability?

Yes

3

1 By design, no one party can modify, delete, or even append any record without consensus, making the system useful for ensuring the immutability of contracts and other legal documents.

2 Smart contracts aim to provide security superior to traditional contract law and to reduce other transaction costs associated with contracting.

3 When everyone on an exchange can view the same ledger, it is easy to broadcast an intention (or offer) by appending it. For example, in a trading network, all ask and bids would be visible to every network participant.

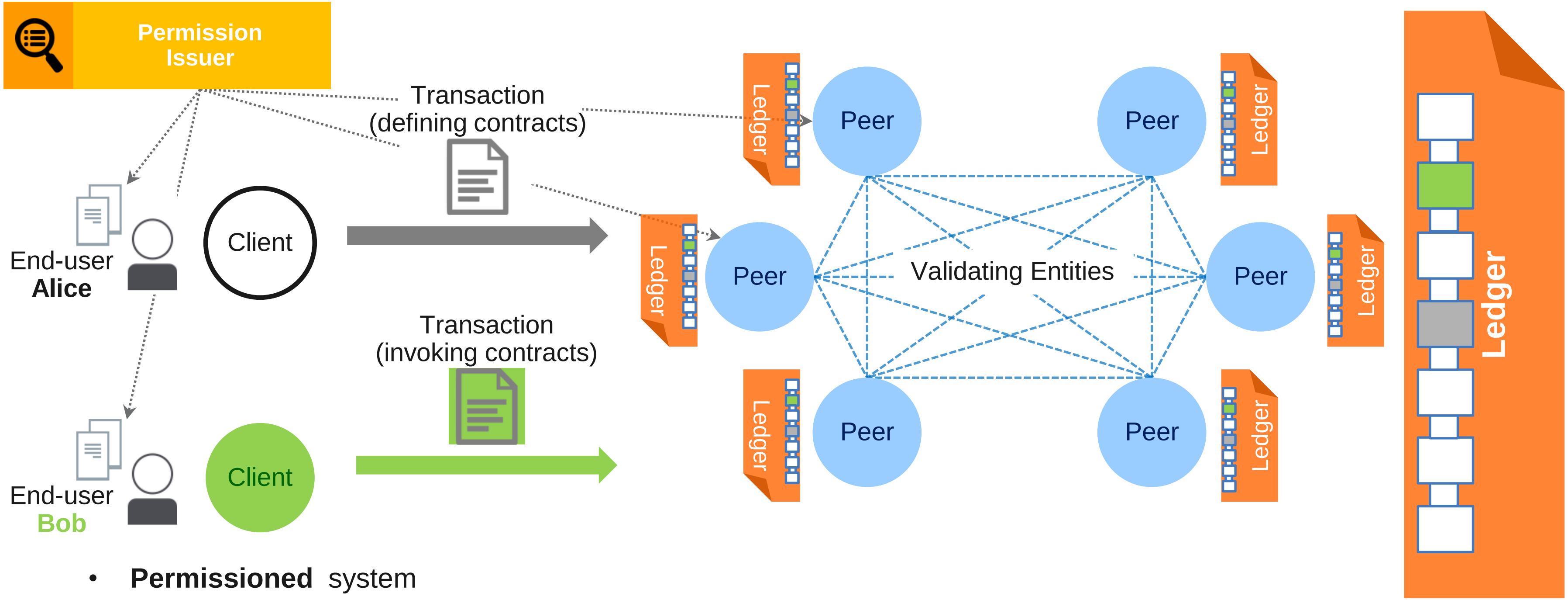
4 Blockchain networks allow each participant to create customized solutions using their own proprietary business logic while running on the same common ledger.

Enterprise blockchain? Not quiet there yet…

- **Strong identity management**
  - Auditable user-behavior
  - Accountability of individual users and validating entities
- **Transactional privacy of blockchain users**
  - Anonymity & unlinkability of transactions of the same user
  - Confidentiality of the contract to be executed w.r.t. validating entities
  - Access control in contract invocation
- **Scalability & performance**
  - Proof-of-work systems need to be substituted by something more “energy-efficient”
  - Need to sustain large number of transactions per time unit
  - Scale to large number of nodes
- **Support for auditing**

Open(-sourced) **Blockchain**:
Enterprise Blockchain born within IBM
One of Hyperledger candidates

Hyperledger-fabric model



- **Permissioned** system
- **Transactions** can implement **arbitrary** (business) **logic** via **chain-codes**
- Distinct roles of **users**, and **validators**
- Users **deploy** chaincodes and **invoke** them through **deploy** & **invoke** transactions
- Validators evaluate the effect of a transaction and reach consensus over the new version of the **ledger**
- **Ledger** = total order of transactions + hash (global state)
- Pluggable **consensus protocol**, currently PBFT & Sieve

Security & privacy features

Privacy of user-participation

Each user has control over the degree to which its transaction activity will be shared with its environment

Contract Privacy

Contract logic can be confidential, i.e., concealable to unauthorized entities

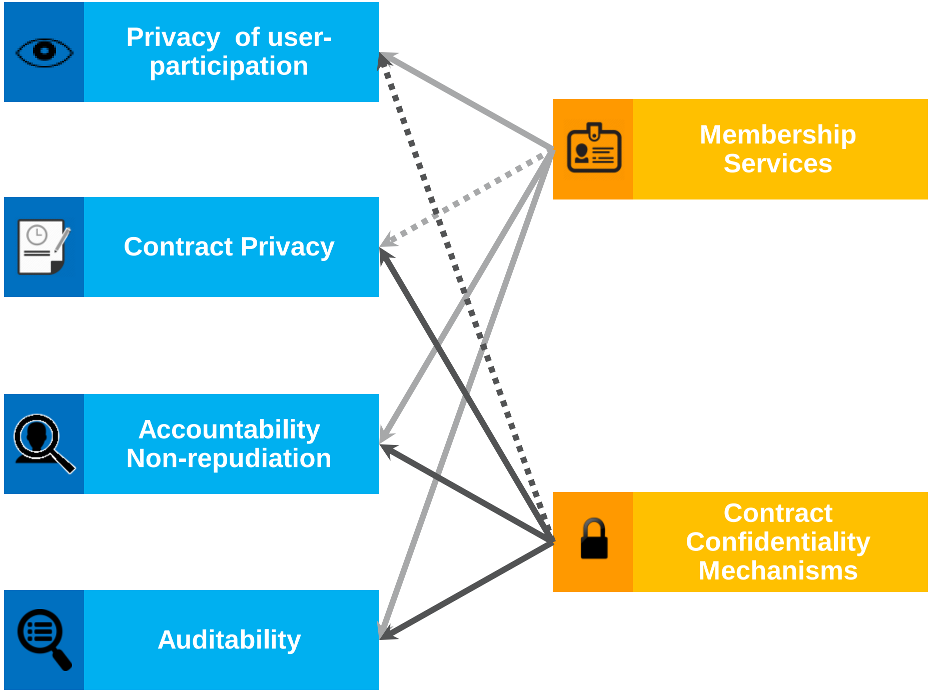
Accountability Non-repudiation

Users can be accounted for the transactions they create, cannot frame other users for their transactions, or forge other users’ transactions.

Auditability

Auditors are able to access & verify any transaction they are legally authorized to

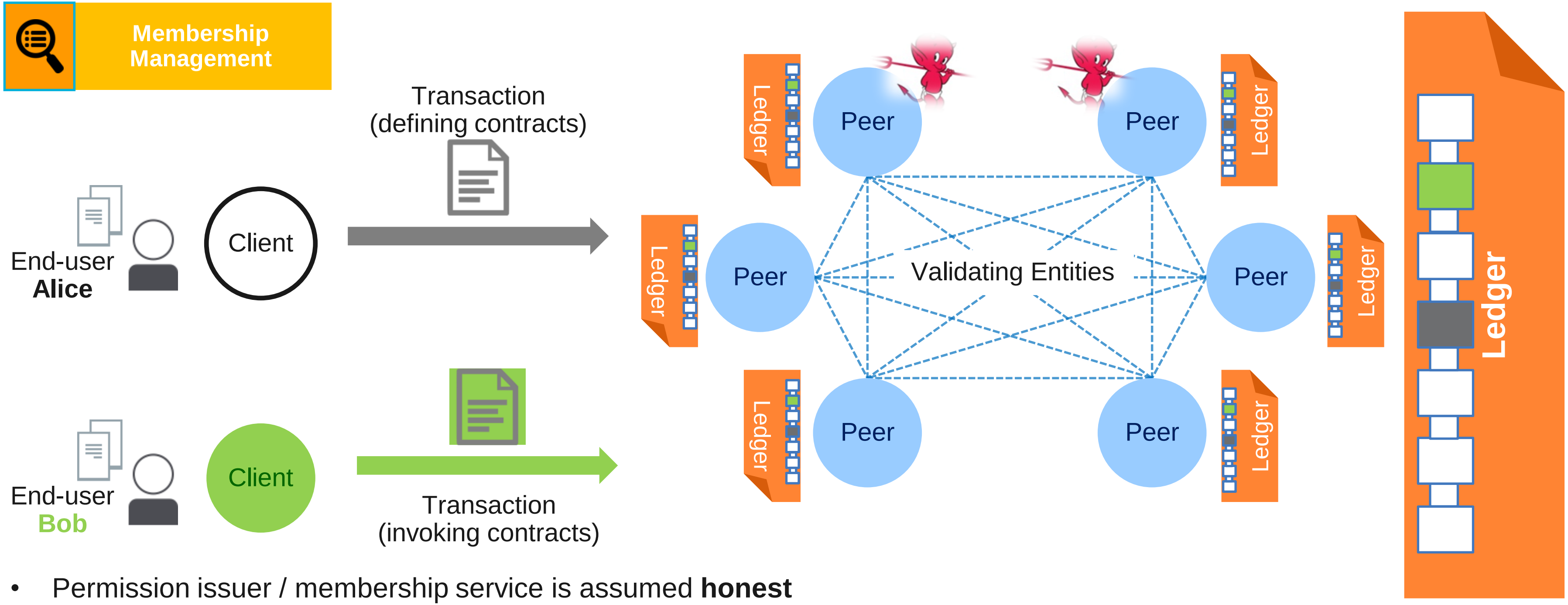
Security and privacy mechanisms


Privacy of user-participation
Contract Privacy
Accountability Non-repudiation
Auditability
Membership Services
Contract Confidentiality Mechanisms

Auditable mechanisms to grant users **credentials** to (**anonymously**) **authenticate their transactions** and secure their connection to the network.

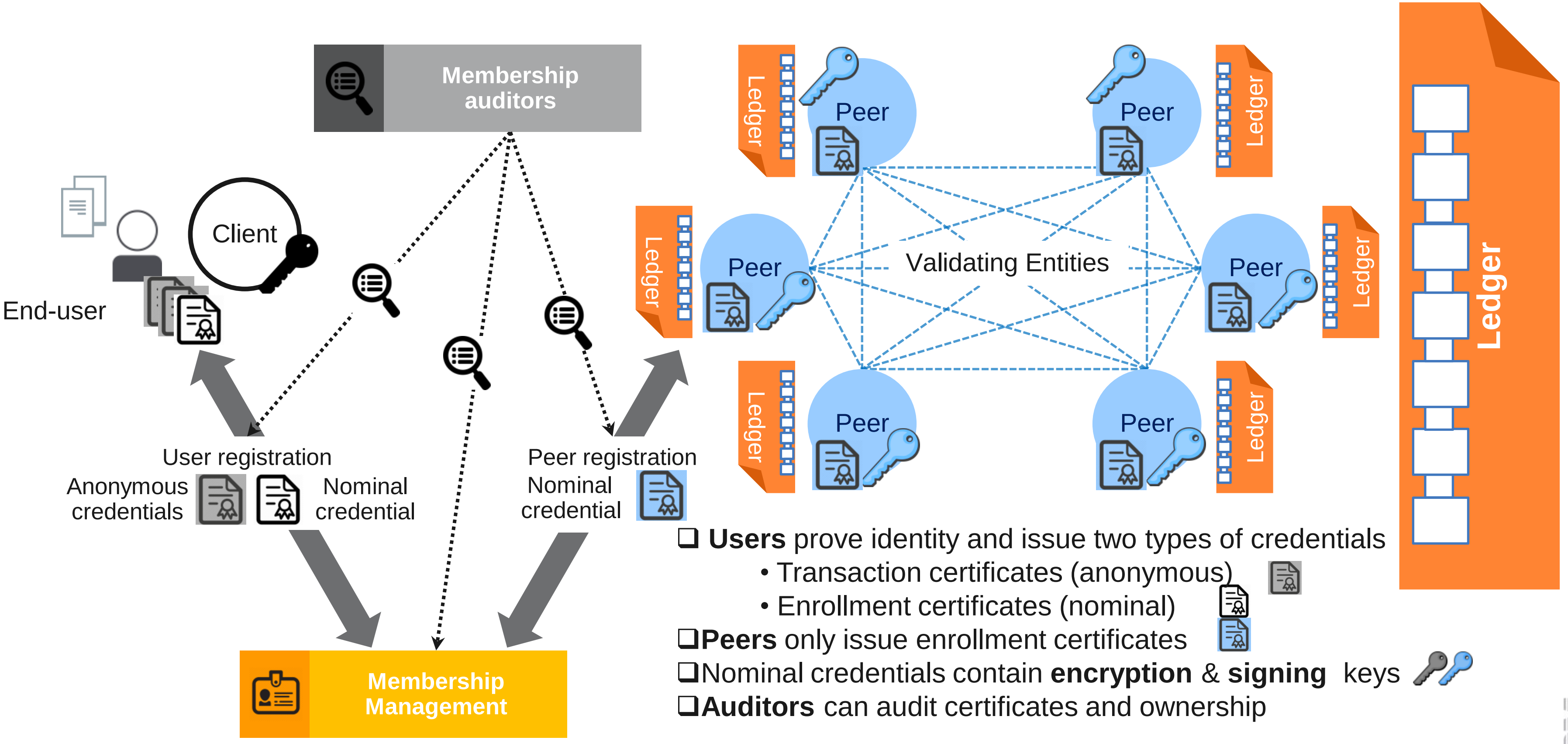
Auditable mechanisms to enable users to **conceal** their **contracts** and associated transactions to **unauthorized nodes**, while guaranteeing correctness.

Adversarial model



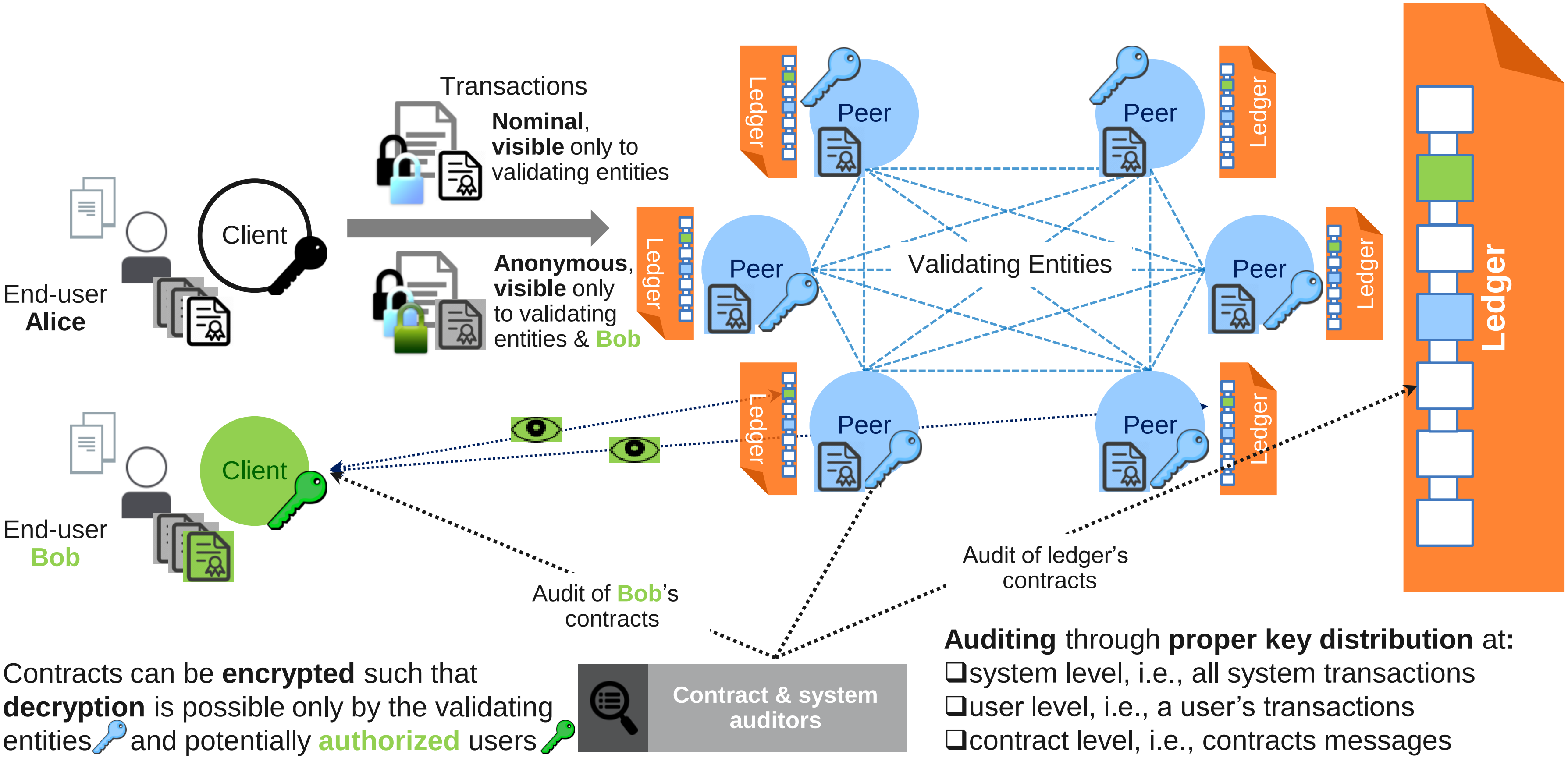
- Permission issuer / membership service is assumed **honest**
- **Users:**
  - have access to raw ledger data
  - may try to escalate their read/invoke access rights
- **Validators:** Up to **f** number of byzantine nodes
- **All entities assumed trusted to not reveal confidential information that are given access to.**

Membership



- □ **Users** prove identity and issue two types of credentials
  - Transaction certificates (anonymous)
  - Enrollment certificates (nominal)
- □ **Peers** only issue enrollment certificates
- □ Nominal credentials contain **encryption** & **signing** keys
- □ **Auditors** can audit certificates and ownership

Working towards user & contract privacy


Transactions
Nominal, **visible** only to validating entities
Anonymous, **visible** only to validating entities & **Bob**
End-user **Alice**
Client
End-user **Bob**
Client
Ledger
Peer
Validating Entities
Ledger
Audit of **Bob**'s contracts
Audit of ledger's contracts
Contract & system auditors

Contracts can be **encrypted** such that **decryption** is possible only by the validating entities🔑 and potentially **authorized** users🔑

Auditing through **proper key distribution** at:

□system level, i.e., all system transactions
□user level, i.e., a user's transactions
□contract level, i.e., contracts messages

Other contract security considerations

- Transaction unforgeability: an attacker should not be able to alter (forge) the content of other user transactions
  - Guaranteed through the unforgeability of digital signatures (membership services)

- Non-repudiation/impersonation attack: an attacker should not be able to claim ownership of other user transactions or frame other users for her transactions
  - through security of digital signatures (membership services)
  - through transaction “bindings” to bind application security to the platform

- Replay attack protection: an attacker should not be able to replay Blockchain transactions and affecting system state (replay attack protection)
  - through transaction nonces
  - optimized via the use of anonymous certificate expiration

Future directions / online discussions

- Not all pieces are there **yet**

- Hyperledger/fabric evolves as a community effort
  - https://github.com/hyperledger/fabric/

- Hot topics:
  - Separating chaincode execution & consensus
  - Extend confidentiality features to extend to validating entities/endorsers
  - Decentralization of membership services / high availability of membership services

Overview

- Blockchain systems
- Blockchain security requirements for enterprise
- Hyperledger/fabric: A security and privacy perspective

Thank you for your attention !
lli@zurich.ibm.com

Contract access management

- Contract resources are accessible only to authorized parties

- Contract resources:
  - Prototypes of contract-functions
  - Contract content
  - Contract state
  - Contract activity

  }

  - Contract invocation →

- “accessible”: **authorization** to…
  - **read** (read access)
    - ➢ requires **trust** to **not reveal** confidential info
    - ➢ granted to **users** or **validators**
    - ➢ fine-grained
  - **submit transactions** to (invocation access)
    - ➢ granted to **users**

- Privacy-preserving by leveraging our membership services infrastructure
  - Use of (anonymous) signing keys in (transaction) enrollment certificates for authentication
  - Use of (anonymous) encryption keys for read access