Law, Science and Technology
MSCA ITN EJD n. 814177

RIoE
Rights of Internet of Everything

Law and Smart contracts

Mirko Zichichi

Outline

- Approccio “Law + Technology”
  - Lo Smart Contract e le sue Varietà
  - Oracoli
- Smart Legal Contract
  - Dalla fiducia nella parte contrattuale alla fiducia nel codice nell'esecuzione del contratto
  - Problematiche
- Intelligible Contract
  - Contratto Ricardiano

RIDE
Rights of Internet of Everything

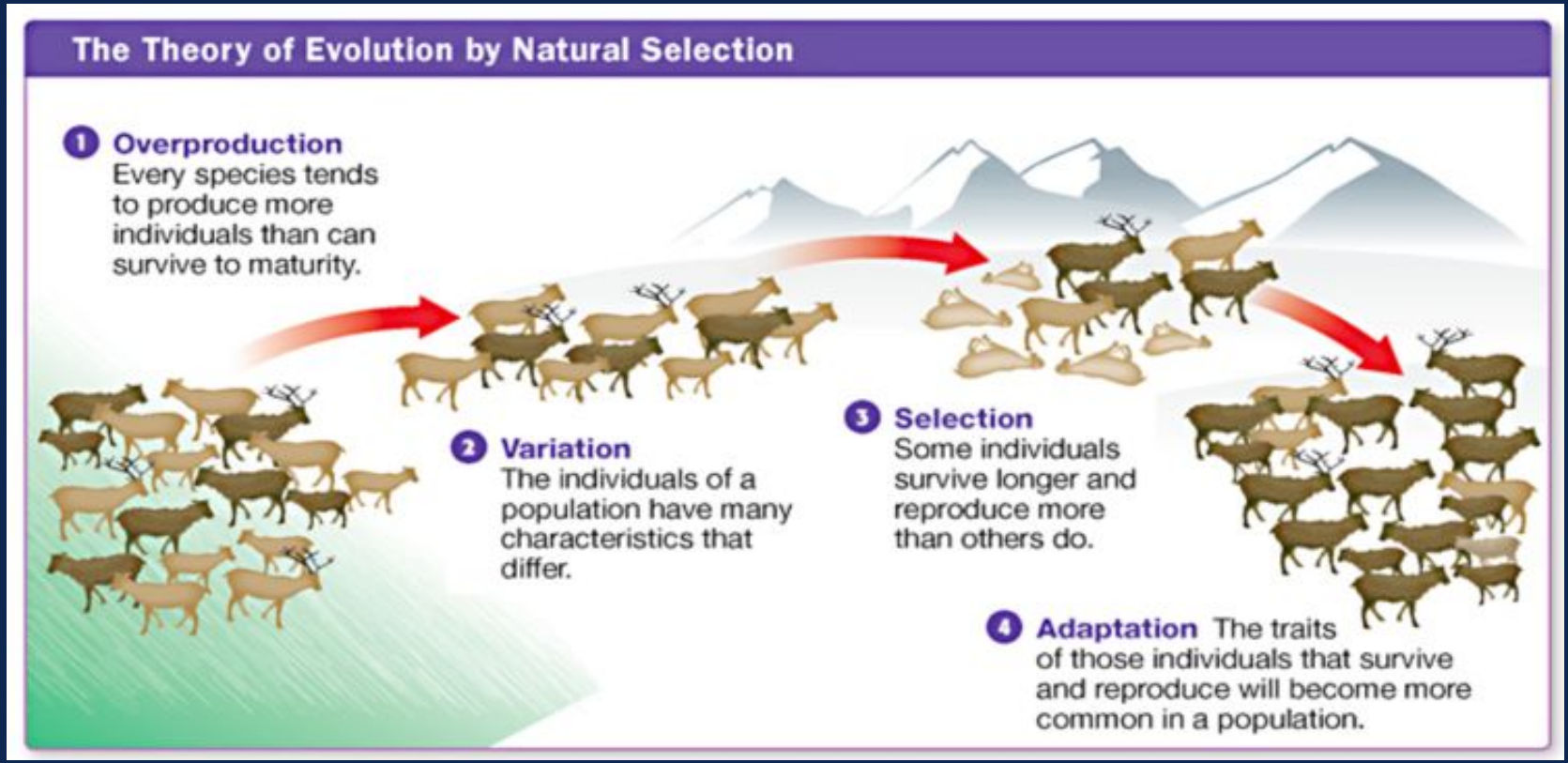
Law + technology

Approccio

“Smart Contracts and the Digital Single Market Through the Lens of a ‘Law + Technology’ Approach”, DR. THIBAULT SCHREPEL, LL.M. https://papers.ssrn.com/sol3/papers.cfm?abstract_id=3947174

RIOE
Rights of Internet of Everything

Un punto di vista evolutivo

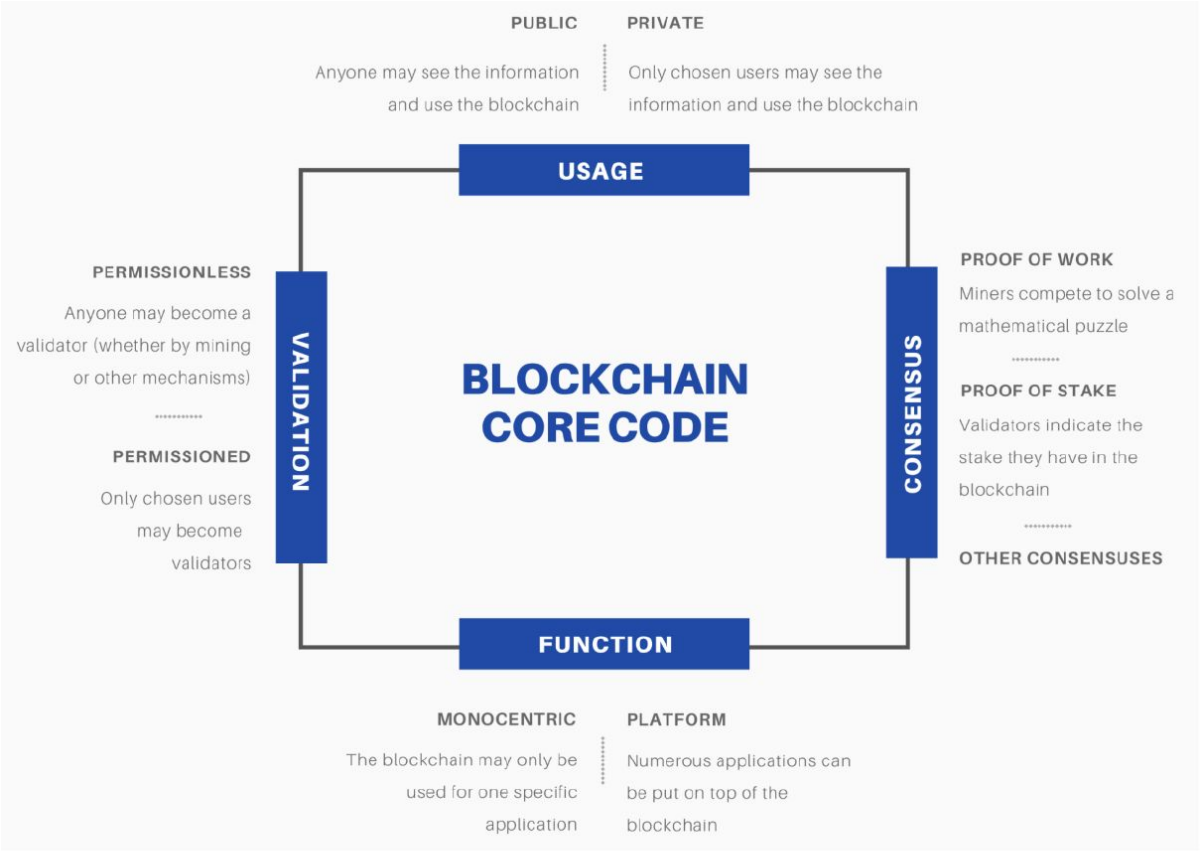


Un punto di vista evolutivo

- Bitcoin si riferisce e fa uso di ricerche, concetti e tecniche del passato, combinando questi elementi preesistenti per dare origine alla blockchain

- **Una volta che una nuova classe di tecnologia è emersa, segue un processo darwiniano di selezione naturale**
  - La technologia → **specie**
  - si muove in diverse direzioni simultaneamente, portando all'emergere di diverse → **varietà**
  - Le varietà che sopravvivono si moltiplicano e cercano di espandere il loro territorio, entrano in contatto con altre specie e iniziano a competere con loro.

La Specie Blockchain e le sue Varietà



A look at blockchain varieties
© Thibault Schrepel[7]

La competizione della blockchain

- Blockchain sta appena iniziando a competere con i mezzi transazionali centralizzati
  - cryptovalute vs. denaro fiat

- La competizione che è inizialmente forte tra le varietà di blockchain sta reggiungendo una competizione tra specie
  - blockchain vs. ecosistemi centralizzati

- Blockchain **sopravviverà** solo se manterrà forti elementi di **differenziazione per ottenere un vantaggio competitivo** sulle altre specie in un dato ambiente

La specie Smart Contracts

- La tecnologia Smart Contract sfrutta le blockchains così come una specie dipende da un’altra

- L'ambiente degli smart contracts ha
  - dimensioni legali, cioè soft law, regolamenti, case law, ecc.
  - dimensioni tecniche, la blockchain

- Devono essere combinati → in assenza di **cooperazione tra legge e tecnologia**, questi due aspetti lotterebbero per prendere il sopravvento

- Un approccio più **cooperativo e armonizzato** è quindi preferibile in modo che gli smart contract possano crescere in un ambiente coeso e duraturo

Possibili Approcci

Assolutista

- Law perspective:
Creare leggi senza cercare il modo di avvicinarsi alla tecnologia
- Technology perspective:
Il fondamentalismo tecnico consiste nel progettare la tecnologia senza fare affidamento su leggi, portando alla creazione di "zone temporaneamente autonome" (TAZ).

Svantaggi:
→comporta l'applicazione di regole e standard legali senza cercare di preservare gli elementi di differenziazione necessari per la sopravvivenza della tecnologia
→non appena la tecnologia estende il suo territorio e lascia la TAZ, l'applicazione della legge può portare all'estinzione della tecnologia.

Cooperativo

- legge e la tecnologia si completano a vicenda cercando di preservare la loro sfera d'influenza e costruendo sui punti di forza dell'altro
- mantenere le caratteristiche distintive della blockchain mentre viene permessa l'applicazione della legge

Vantaggi:
→si possono usare gli smart contracts dove il diritto contrattuale è difficile da far rispettare, per esempio, perché le giurisdizioni sono poco amichevoli
→usati dove la legge non può raggiungere un obiettivo da sola, come prevenire la corruzione

Caratteristiche principali della specie Smart Contract

1. Funzionamento
2. Immutabilità
3. Varietà della specie
4. Interazioni tra varietà e con il mondo esterno

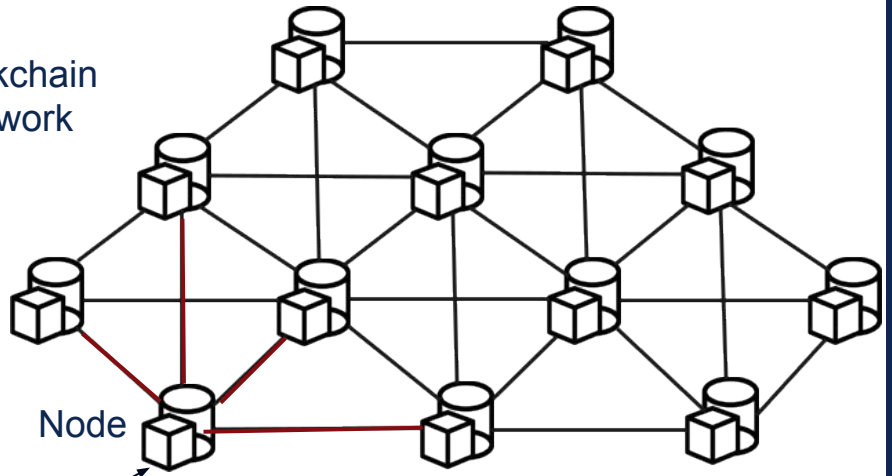
1. Funzionamento

Ethereum Smart Contracts

- Permette di mantenere facilmente delle strutture dati nella blockchain

- Una nuova transazione si riferisce ad una precedente e aggiorna lo stato del sistema
  - In questo caso lo stato del sistema considera non solo le transazioni monetarie, ma anche le strutture dei dati negli smart contracts
  - La transazione precedente si riferisce ad una che mantiene il codice e lo stato dello smart contract
  - La nuova transazione indica un insieme di istruzioni da eseguire nel contratto

Esempio: Smart Contract Voto



execute(

```
function _vote(uint256 challenge, bool inFavour) public {
    if (inFavour) {
        challenge.inFavour.add(msg.sender); // Alice
    } else {
        challenge.against.add(msg.sender); // Alice
    }
}
```

)

2. Immutabilità

2. Immutabilità

Gli smart contract inseriti in una blockchain si dice che siano immutabili di default.

Il codice sorgente (bytecode) di uno smart contract è infatti registrato in una transazione che viene “minata” in un blocco insieme ad altre transazioni:

TX 1 : 5 btc |--> Charlie Pub
TX 2 : TX 1 |--> Bob Pub
TX 3 : TX 2 |--> Alice Pub

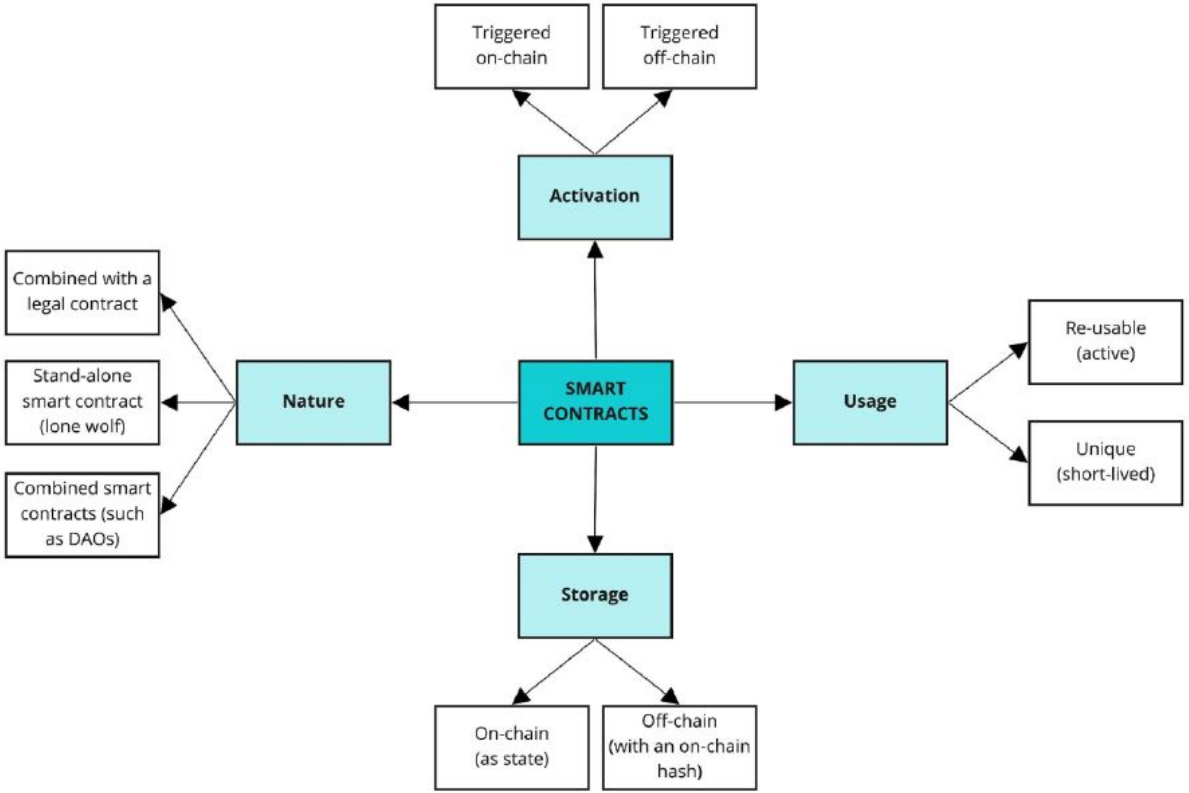
\+ **TX 4** : **Bytecode** |--> Indirizzo Contratto

sign(**TX 4**, Alice Priv)

Alice’s Wallet

3. Varietà della specie

Varietà



Title: An overview of blockchain smart contracts
© Thibault Schrepel (2021)

Varietà: **Natura**

1. **Combinare uno smart contract con un "contratto legale"**
   a. es., un contratto di affitto potrebbe essere scritto in prosa tra il proprietario di un appartamento e un inquilino, mentre lo smart contract potrebbe automatizzare il pagamento.

2. **Smart contract senza il supporto di un contratto legale**
   a. maggior parte degli smart contract in circolazione oggi
   b. "lupi solitari"→perché intendono essere autosufficienti.

3. **Smart contracts combinati con altri smart contracts**
   a. creano le condizioni per la governance decentralizzata di ecosistemi
   b. Decentralized Autonomous Organizations (**DAO**s).

Varietà: **Uso**

1. Smart contract condizionato a eventi della vita reale che **solo una delle due (o più) parti del contratto può invocare** -> *intuitu personae*
   a. le condizioni per invocare uno smart contract sono specifiche per una singola parte

2. Smart contract invocato regolarmente, sia da una sola parte che da una **moltitudine di parti**
   a. qualsiasi utente può invocarli
   b. detto “attivo”

Varietà: **Attivazione**

1. Attivati **on-chain**: vengono invocati in seguito ad un evento della blockchain
   a. es., uno smart contract può essere progettato per essere invocato solo quando il valore di uno asset presente nella blockchain supera un certo livello

2. Attivati **off-chain**: vengono invocati in seguito a un evento esterno alla blockchain
   a. oracoli

Varietà: **Conservazione**

1. **On-chain:** il bytecode di uno smart contract è memorizzato su una transazione messa on-chain
   a. facendo così si assicura l'immutabilità ma anche una mancanza di segretezza

2. **Off-chain**: i dati (compreso il bytecode) possono anche essere memorizzati off-chain, con solo solo l'hash che viene registrato sulla blockchain
   a. l'immutabilità dello smart contract rimane di fatto garantita perché cambiandolo si genera automaticamente un nuovo valore di hash che non corrisponde a quello originale registrato sulla blockchain.

4. Interazioni tra varietà e con il mondo esterno

Interazioni tra varietà

Inter-blockchain

- gli smart contracts interagiscono tra loro, sia per competere che per cooperare

- diverse blockchain sono in competizione per gli smart contracts e, a seconda della tecnologia su cui sono costruiti, hanno caratteristiche uniche

Esempi:
→ Gli smart contracts di *Polkadot*, *Cardano* e *EOS* sono, in media, convalidati più rapidamente di *Ethereum*
→ *Tezos* permette più segretezza
→ *Polkadot* utilizza dei bridge per consentire il trasferimento di token o dati da una blockchain ad un'altra

Intra-blockchain

- c'è anche concorrenza e cooperazione tra gli smart contract costruiti sulla stessa blockchain

- alcuni diventano più appetibili di altri perché sono meglio progettati, introducono nuove funzioni, o sono più supportati

Esempi:
→Uniswap 1, 2, 3
→Gli smart contracts cooperano quando sono tecnicamente collegati tra loro.
Ad esempio molti smart contracts trasferiscono automaticamente lo stesso tipo di ERC20 Token

Interazioni con il mondo esterno

Gli oracoli permettono agli smart contracts di **interagire con il mondo esterno**

- In origine, un oracolo era una persona incaricata di riferire la profezia sussurrata da fonti divine
- Per quanto riguarda la blockchain, generalmente, designa l'intermediario che riporta le informazioni dal mondo reale alla blockchain o viceversa
- In alternativa, l'oracolo può avere una funzione computazionale quando esegue calcoli off-chain

Varietà degli Oracoli: **Direzione, Raccolta Dati, Fonti**

1. Le informazioni possono prendere due **direzioni**:
   1.1. *outbound*, le informazioni dalla blockchain sono portate al mondo esterno
   1.2. *inbound*, si portano informazioni all'interno della blockchain.

2. Quando *inbound*, si distinguono diversi modi di **raccogliere informazioni**:
   2.1. *software*, interagisce con (esistenti) informazioni online e poi le trasmette
   2.2. *hardware*, trasforma le misure del mondo reale in informazioni digitali
   2.3. *human*, terza parte fidata che fornisce informazioni del mondo reale.

3. L'oracolo può usare una sola **fonte** o diverse di esse:
   3.1. *singola fonte*, ``ricentralizza'' la blockchain introducendo un **singolo punto di fallimento** e richiedendo la fiducia in un solo punto di ingresso
   3.2. *combinazione di diverse fonti*, è preferibile ma richiede tuttavia regole di governance ben progettate.

Varietà degli Oracoli: **Validazione, Integrazione, Uso**

4. Si deve poi **convalidare** l'informazione una volta trasmessa:
   4.1. *automatica*, se l'utente decide di fidarsi dell'oracolo
   4.2. *voto*, oggetto di un voto sottoposto agli utenti della blockchain (DAO).

5. Le informazioni devono essere **integrate**:
   5.1. *senza intermediari*, direttamente distribuite alla rete blockchain
   5.2. *interfaccia di uno smart contract personalizzata*, e.g. dApp
   5.3. *modulo software per il data pre-processing*
   5.4. *soluzione personalizzata*, per prevenire la falsificazione e.g. fingerprint

6. Una volta che l'informazione è integrata, i suoi **usi** possono essere:
   6.1. *contract-specific*, impiego in un singolo smart contract
   6.2. *multiple smart contracts use,* come un database es. dati finanziari

Smart Legal Contract

Dalla fiducia nella parte contrattuale alla fiducia nel codice nell'esecuzione del contratto

"From Trust in the Contracting Party to Trust in the Code in Contract Performance", Chantal Bomprezzi https://kluwerlawonline.com/api/Product/CitationPDFURL?file=Journals%5CEuCML%5CEuCML2021032.pdf

"Smart legal contracts: advice to Government", UK's Law Commission
https://www.lawcom.gov.uk/project/smart-contracts/

RIDE
Rights of Internet of Everything

Contratti Inviolabili?

L'uso degli smart contract può essere realizzato sia per i contratti commerciali business-to-business, sia per quelli peer-to-peer o anche per quelli business-to-consumer (B2C).

Nel caso B2C, ci possono essere diverse situazioni in cui l'auto-esecuzione di uno smart contract porta a la violazione di tale contratto:

1. Il contenuto del codice non corrisponde alla volontà delle parti, determinando così che l'esecuzione del contratto non soddisfi il consumatore.

2. Problematiche di natura tecnica che impattano sull'esecuzione del contratto.

3. Altre problematiche dovute alla chiusura della blockchain verso l’esterno, ovvero quando vi è la necessità di collegare lo smart contract con il mondo off-chain.

Contratti Inviolabili? -> (1) Contenuto del codice

Quando il codice non funziona come previsto dal consumatore e concordato nel contratto, il contratto viene violato.

Contratti Inviolabili? -> (2) Problematiche di natura tecnica

Le applicazioni basate su blockchain sono costituite da più componenti e diversi problemi di natura tecnica possono influire negativamente su tali componenti:

- Lo smart contract può essere soggetto a **bug**, come qualsiasi altro programma informatico.
- I problemi possono derivare anche dalla blockchain sottostante, ad esempio da **attacchi** che possono dare spazio alla manipolazione dell'esecuzione di uno smart contract.
- Inoltre, gli oracoli possono essere compromessi, in quanto la fonte di dati esterna potrebbe non funzionare o diventare inattiva.

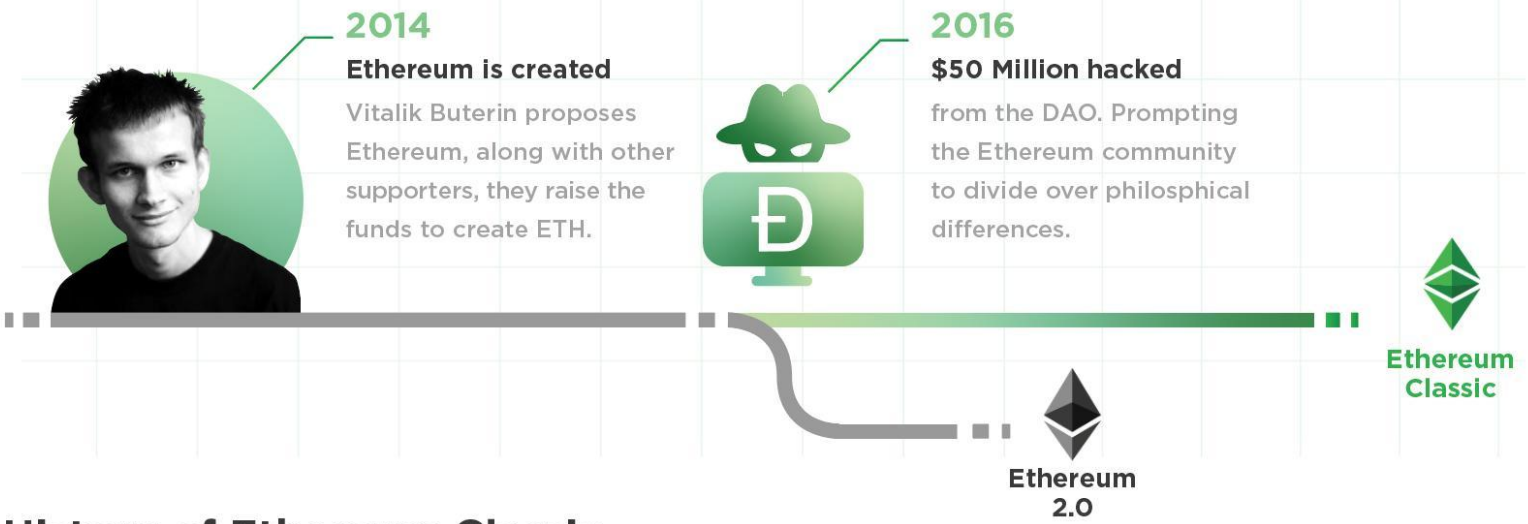
Contratti Inviolabili? -> (2) Problematiche di natura tecnica
Bugs negli Smart Contracts

- L'attacco **Reentrancy** in Solidity.
  Si verifica quando una funzione effettua una chiamata esterna a un altro contratto non attendibile. Poi il contratto non attendibile effettua una chiamata ricorsiva alla funzione originale nel tentativo di prosciugare dei fondi.

- Sebbene l'attacco di rientranza sia considerato piuttosto vecchio, negli ultimi due anni si sono verificati casi come:
  - *Uniswap/Lendf.Me hacks (April 2020) – $25 million.*
  - *The BurgerSwap hack (May 2021) – $7.2 million.*
  - *The SURGEBNB hack (August 2021) – $4 million.*
  - *CREAM FINANCE hack (August 2021) – $18.8 million.*
  - *Siren protocol hack (September 2021) – $3.5 million.*
  - *Fei Protocol hack (April 2022) – $80 million.*

Contratti Inviolabili? -> (2) Problematiche di natura tecnica

Bugs negli Smart Contracts → The **DAO hack**



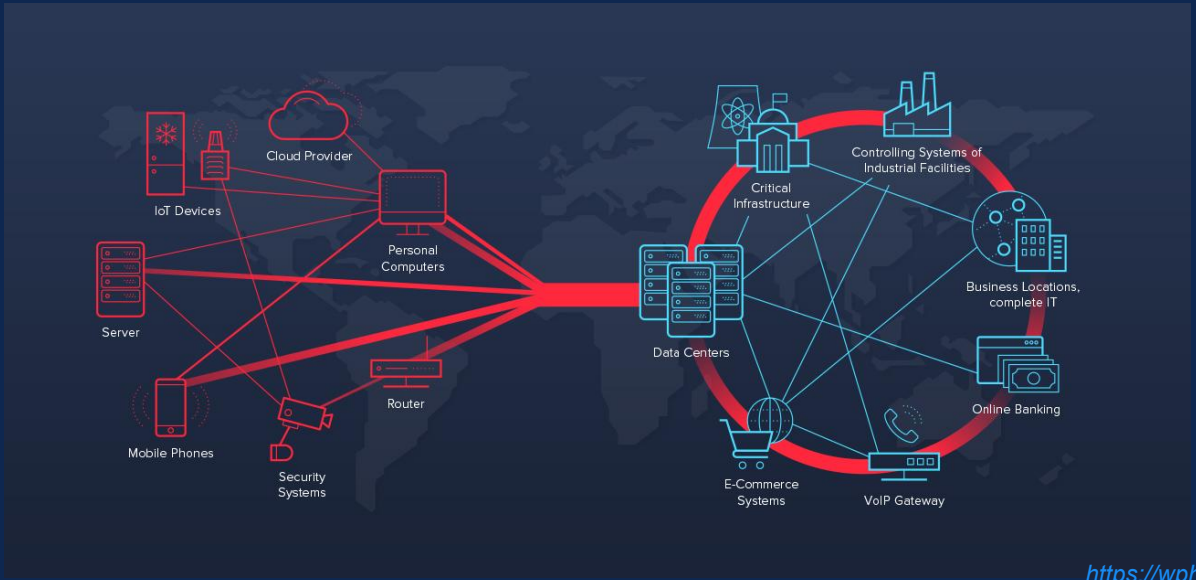
History of Ethereum Classic

The DAO Hack in 2016, one of the earliest projects on Ethereum called “The Dao”, was hacked for $50 million. The community was faced with a controversial dilemma: “Bailout investors like traditional bank systems have done for years or keep our values". The investors would receive their stolen funds, but at the cost of a foundational principle: immutability. This decision split the community, the majority supported “Bailout”, while the minority supported “Code is Law”. The chain forked, and the original Ethereum was labelled Classic.

https://twitter.com/eth_classic/status/1471329806671237121

Contratti Inviolabili? -> (2) Problematiche di natura tecnica
Problemi nelle Blockchain

- **Solana Went Offline for Four Hours ->** https://finance.yahoo.com/news/solana-latest-ddos-attack-leads-120022342.html
- Distributed Denial-of-Service (DDoS) attack



https://wphostinggeeks.com/cloud-vs-vps/

Contratti Inviolabili? -> (3) Chiusura verso l’esterno

- Se le informazioni di un oracolo non vengono fornite affatto o sono errate, il contratto non viene eseguito o non viene eseguito correttamente.
- Questo non può accadere per malfunzionamenti tecnici, ma anche **per errori o azioni umane**.

- Ad esempio, un corriere che segnala di aver consegnato il pacco all'indirizzo specificato, mentre il pacco non è stato spedito, oppure il contenuto del pacco differisce da quanto concordato dalle parti nel contratto.

- L'inserimento dei dati di input nella blockchain è sotto il controllo diretto di qualcuno e non beneficia del carattere decentralizzato della blockchain.
- “**Garbage *In* → Garbage *Out***”

La discrepanza tra la tecnologia decentralizzata e l'assenza di controllo sull'esecuzione di un contratto.

- La blockchain è una tecnologia decentralizzata.
  C'è molta confusione sul significato del termine "decentralizzazione".

- Quest'ultimo quest'ultimo potrebbe riferirsi sia alla tecnologia in quanto tale sia alla governance dell'applicazione che gira su una blockchain.

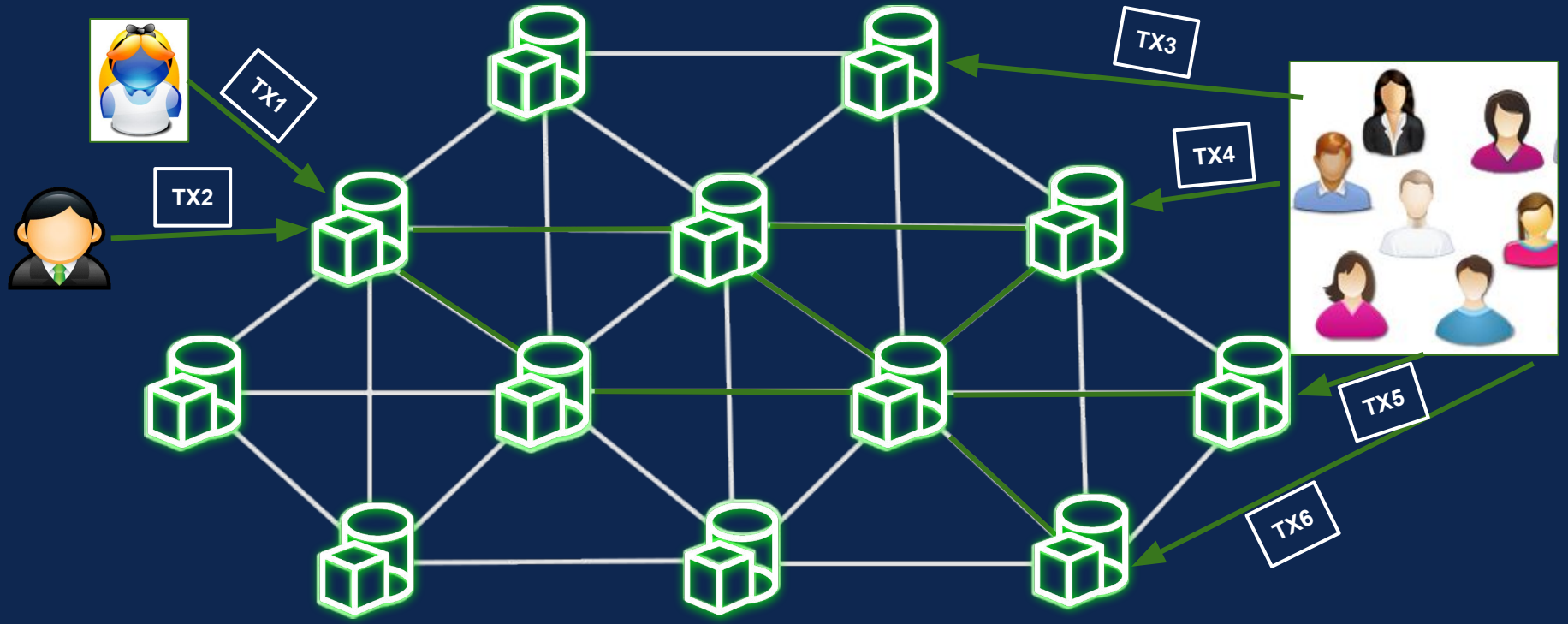
- **Governance** de/centralizzata ->
  Meccanismo di consenso + Sviluppo software

- Tecnologia de/centralizzata ->
  registro distribuito oppure centralizzato

+ Public **Permissionless** Blockchain

Meccanismo di Consenso

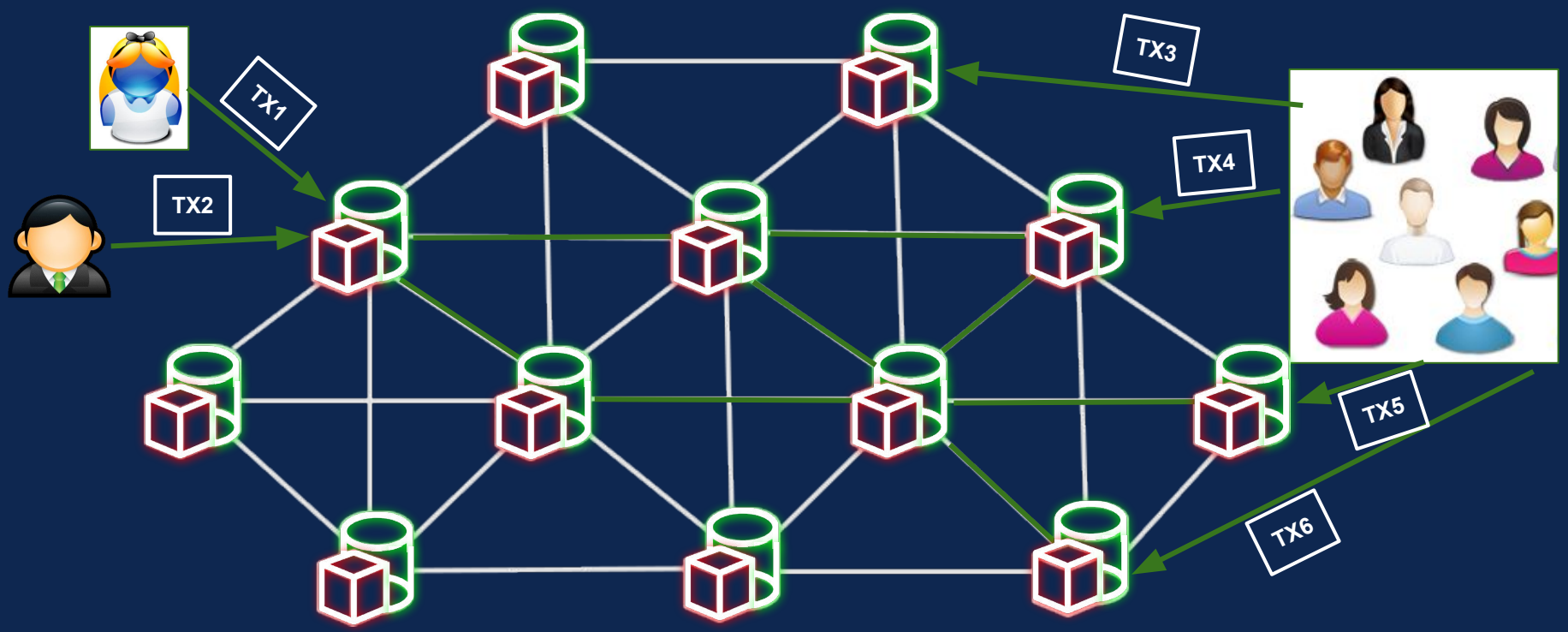
Registro



+ Public **Permissioned** Blockchain

Meccanismo di Consenso

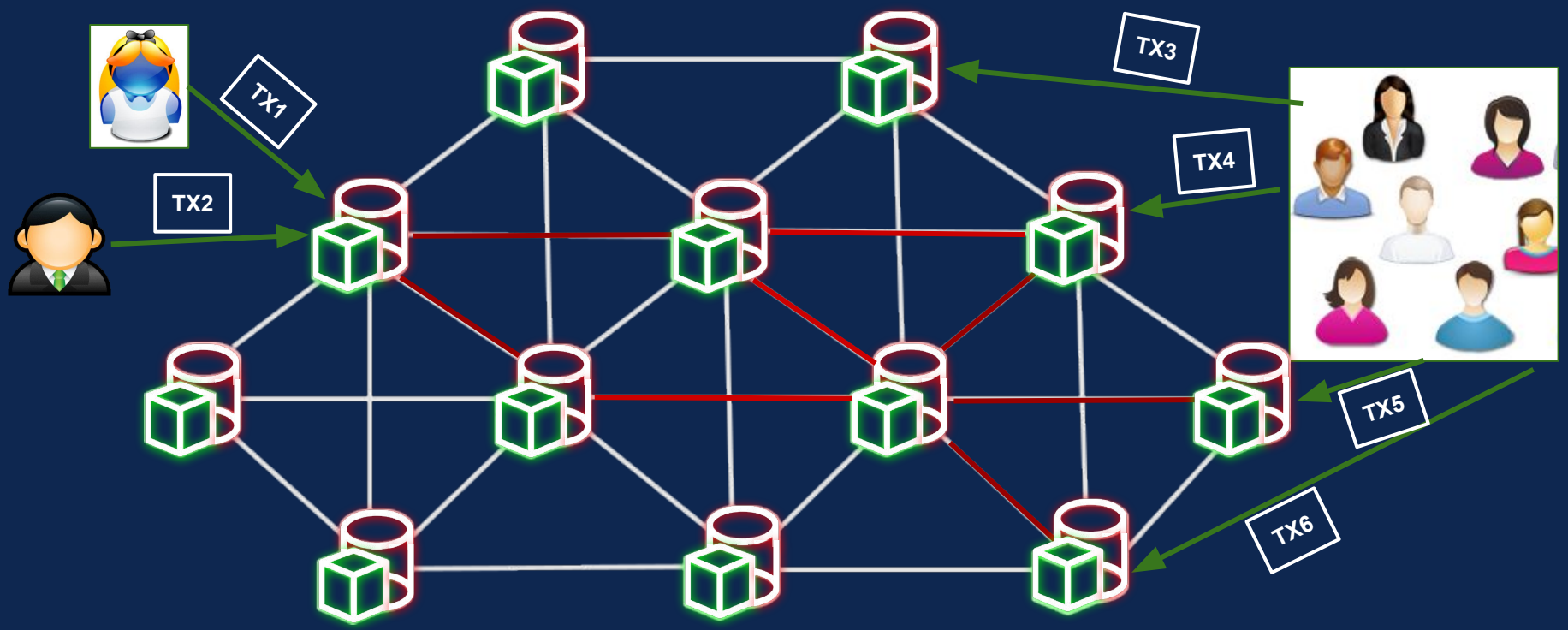
Registro



+ Private **Permissionless** Blockchain

Meccanismo di Consenso

Registro



+ Private **Permissioned** Blockchain

Meccanismo di Consenso

Registro

La discrepanza tra la tecnologia decentralizzata e l'assenza di controllo sull'esecuzione di un contratto

- *Private Permissioned*: non coinvolge i consumatori, è un esempio di governance centralizzata della blockchain.
- Di conseguenza, agisce come un'architettura standard client-server (es. il Cloud). L'esecuzione del contratto è sotto un controllo centralizzato, senza ulteriori vantaggi per i consumatori.

- *Public Permissioned*: il problema è simile, ma se ben progettata può garantire un meccanismo di controllo decentralizzato quasi simile a permissionless.

- Permissionless: tuttavia, anche in questo caso può esserci un controllo centralizzato sull'esecuzione del contratto da parte dell'azienda. Quest'ultimo *non dipende dalla governance centralizzata* dell'applicazione, ma piuttosto dall'oggetto dell'obbligazione contrattuale.

La discrepanza tra la tecnologia decentralizzata e l'assenza di controllo sull'esecuzione di un contratto (codice sorgente)

- Lo smart contract viene eseguito in base alle disposizioni contrattuali.

- Per proteggere i consumatori è di vitale importanza assicurarsi che siano consapevoli del contenuto del contratto
  - che il contratto non sia troppo sbilanciato a favore dell'azienda
  - che non contenga clausole abusive.

- Se la conclusione del contratto è nelle mani dell'azienda:
  - non importa che i contratti intelligenti siano in grado di auto-eseguirsi;
  - l'azienda può influenzare indirettamente l'esecuzione del contratto.

La discrepanza tra la tecnologia decentralizzata e l'assenza di controllo sull'esecuzione di un contratto (codice sorgente)

- Il codice dello smart contract è una creazione umana.
- Pertanto, si potrebbe affermare che tali creatori, o chi li ha ingaggiati, dovrebbero essere ritenuti responsabili per i malfunzionamenti del codice che hanno causato la violazione del contratto.
- Se il codice viene prodotto dall’azienda, i consumatori devono ancora avere fiducia nell'azienda e sono applicabili i tradizionali rimedi legali per l'inadempimento.

The **code is law**?

- I non programmatori si devono affidare completamente agli esperti (programmatori di Smart Contracts) per spiegare il contratto, il che comporta ulteriori sfide e pone un'enfasi ancora maggiore sulla responsabilità.

- *"Utilizzando l'analogia con gli avvocati, i programmatori di smart contracts potrebbero diventare una professione regolamentata e, analogamente agli avvocati, potrebbero essere obbligati a sottoscrivere un'assicurazione di responsabilità civile".*

- Tuttavia, gli avvocati possono *" aiutare le parti a determinare quale sarebbe la migliore struttura contrattuale per una particolare transazione e spiegare loro i potenziali rischi, come quelli legati alla sicurezza o alla natura oggettiva degli smart contracts, che lasciano meno spazio alle negoziazioni."*

- *Dora Kadar, https://tech.eu/2022/05/09/can-smart-contracts-replace-lawyers-in-europe/*

UK's Law Commission **Smart Legal Contract**

- E’ un contratto vincolante in cui alcuni o tutti gli obblighi contrattuali sono definiti e/o eseguiti automaticamente da un programma informatico

- Sottoinsieme degli smart contract

- Possono assumere una varietà di forme con diversi gradi di automazione:
  - *Contratto in linguaggio naturale con esecuzione automatica tramite codice*
  - *Contratto ibrido*
  - *Contratto redatto esclusivamente tramite codice*

Contratto in linguaggio naturale con esecuzione automatica tramite codice

- Contratto “classico” in linguaggio naturale, in cui alcuni o tutti gli obblighi contrattuali sono eseguiti automaticamente dal codice.
- Il codice stesso non definisce alcun obbligo, ma è solo uno strumento utilizzato per adempiere agli obblighi contrattuali.
- E’ la forma di smart contract legale attualmente più utilizzata. Questa forma non solleva problemi giuridici nuovi (o ne solleva pochi) nel contesto della creazione e interpretazione dei contratti.
- Il problema più grande rimane capire la giusta traduzione da linguaggio naturale a codice.

Contratto ibrido

- Un contratto legale intelligente ibrido è un contratto in cui alcuni obblighi contrattuali sono definiti in linguaggio naturale e altri sono definiti nel codice di un programma informatico.

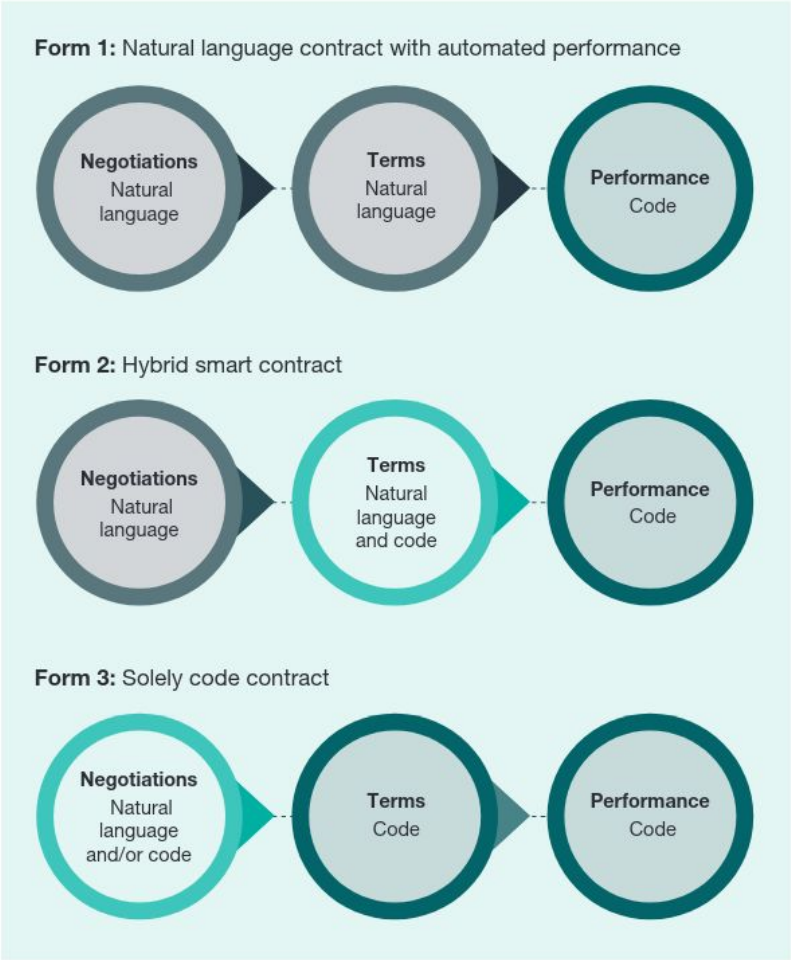
- Alcuni o tutti gli obblighi contrattuali sono eseguiti automaticamente:
  - principalmente scritti in codice con alcuni termini in linguaggio naturale che stabiliscono, ad esempio, la legge applicabile e la giurisdizione.
  - principalmente scritti in linguaggio naturale e includono solo uno o due termini scritti in codice.

- I termini in linguaggio naturale possono essere in un documento separato oppure trasposti in commenti in linguaggio naturale inclusi nel codice.

+ Contratto redatto esclusivamente tramite codice

- Tutti i termini contrattuali sono definiti nel codice ed eseguiti automaticamente. Non esiste una versione in linguaggio naturale.
- Questo tipo di smart legal contract presenta le maggiori sfide dal punto di vista del diritto contrattuale, in termini di determinazione di se e quando si forma un contratto legale e di come tale contratto possa essere interpretato.
- I contratti commerciali sono tipicamente troppo ricchi di sfumature per essere ridotti esclusivamente a codice.

Tre tipi di Smart Legal Contract



Smart legal contracts: Summary

Alcune problematiche

- Dal momento che gli atti hanno diversi requisiti di formalità (ad esempio, devono essere testimoniati e attestati), allo stato attuale delle cose **è difficile** utilizzare *contratti ibridi* o *esclusivamente in codice* per creare un atto.

- Potrebbero sorgere difficoltà in relazione alla **determinazione della giurisdizione e della legge applicabile** per alcuni smart legal contracts.
  - in particolare quando sono unilaterali e unicamente in codice, o formati dall'interazione autonoma di programmi informatici, es. altri smart contract

- Anche la **localizzazione digitale**, ossia la necessità di attribuire luoghi reali ai beni digitali e alle azioni che "hanno luogo" su un registro distribuito, rappresenta una sfida significativa.

+ Altre problematiche legate al trattamento dei dati personali

- **Protezione e ubicazione dei dati personali**.
- Gli smart contract possono fare uso di dati personali e il GDPR può essere applicato a loro a seconda dei dati che utilizzano e generano.

- Tensioni principali:
  - Come gestire immutabilità e diritto all’oblio?
  - Come far valere la responsabilità dei data controllers in una permissionless blockchain sono identificati da indirizzi pseudonimi?
  - Obblighi di archiviazione dei dati all'interno dell'Unione Europea o di uno Stato membro dell'UE -> non applicabile in una permissionless blockchain

+ Possibili Soluzioni

- *UK's Law Commission* **“Reasonable coder”**:
  - l'interpretazione di un termine del contratto sotto forma di codice dovrebbe essere determinata chiedendo cosa significherebbe il termine ad una *persona ragionevole* con conoscenza e comprensione del codice.
  - La risposta a questa domanda sarà ciò che il codice sembrava istruire il computer a fare, secondo l'opinione ragionata di tale persona.

- Sviluppare pratiche consolidate e modelli di contratti che le parti possono utilizzare per negoziare e redigere i loro smart contracts

- Tecnologie e metodi per la protezione dei dati personali e segreti industriali
  - es. crittografia (hash, zero knowledge proof)
  - es. multi-layered DLTs

Intelligible Contract

Luca Cervone, Monica Palmirani, Fabio Vitali

http://hdl.handle.net/10125/63959

Slides originali di Luca Cervone

http://lbl.cirsfid.unibo.it/wp-content/uploads/2020/01/The_Intelligible_Contract-v4.pdf

RIDE
Rights of Internet of Everything

Contratti Ricardiani

- I contratti ricardiani cercano di colmare il divario tra la prosa legale (linguaggio naturale) e il codice eseguibile.
- Lo sviluppatore descrive una tripla <***P, C, M***> dove:
  - ***P*** descrive la semantica denotativa dei contratti (la prosa legale);
  - ***C*** descrive la semantica operativa dei contratti (il codice codice sorgente);
  - ***M*** è una mappatura tra le operazioni espresse in ***C*** e la prosa legale in ***P***.

Smart Contract Templates

- Gli Smart Contract Templates sono un'implementazione dei *contratti ricardiani* il cui codice operativo è standardizzato e il cui comportamento è controllato da parametri contenuti in uno smart contract.
- Strumenti di redazione legale permettono a sviluppatori e ad esperti legali di creare modelli di smart contracts insieme
- La prosa legale è serializzata tramite vocabolari standard e flessibili
- Un mark-up dei documenti collega gli elementi di un contratto ad ontologie standard
- Alcune “features” collegano la prosa legale al codice operativo

Intelligible Contracts

- Gli ***Intelligible Contracts*** sono smart legal contracts scritti in linguaggio naturale che possono essere mappati, interamente o parzialmente, in codice smart contract basato su blockchain.

- Gli ***Intelligible Contracts*** estendono i Contratti Ricardiani e gli Smart Contract Templates fornendo specifiche per l'intelligibilità dei contratti contratti digitali.

- Colmano le lacune dei Contratti Ricardiani e degli Smart Contracts Templates

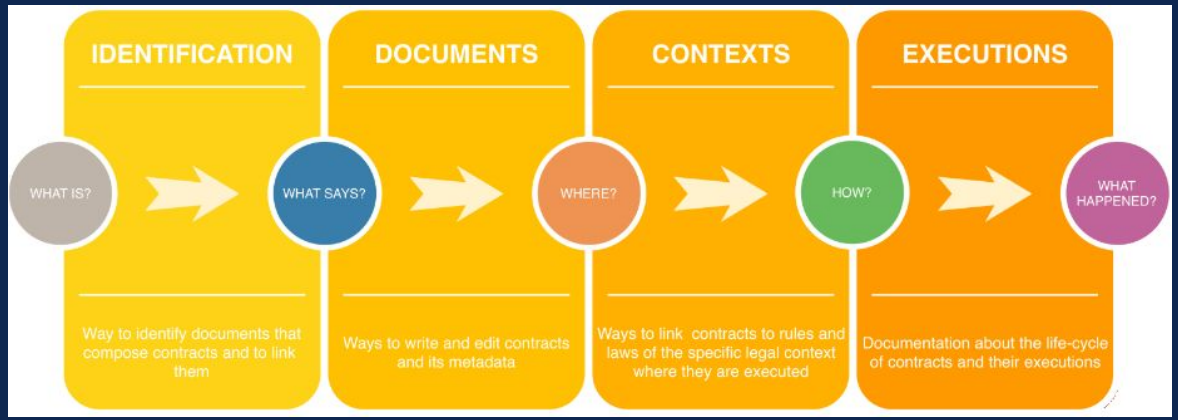
->

Intelligible Contracts - Features

- Forniscono collegamenti tra i contratti e altre risorse e documenti legali e non
  - ad esempio, riferimenti normativi nei contratti
- Forniscono una descrizione del contesto giuridico dei contratti
  - ad esempio, la giurisdizione dei fatti.
- Forniscono informazioni sul contesto operativo dei contratti
  - ad esempio, il tipo di blockchain.
- Riportano l'esecuzione automatica dei contratti.

Intelligible Contracts - Features 2

- collegano tutte le risorse che compongono i contratti o che ne definiscono i contesti giuridici
- collegano gli agenti coinvolti nel ciclo di vita dei contratti
- collegano le risorse digitali che descrivono le modalità di esecuzione del codice operativo
- collegano le risorse digitali che riportano ciò che accade durante le esecuzione dei contratti.

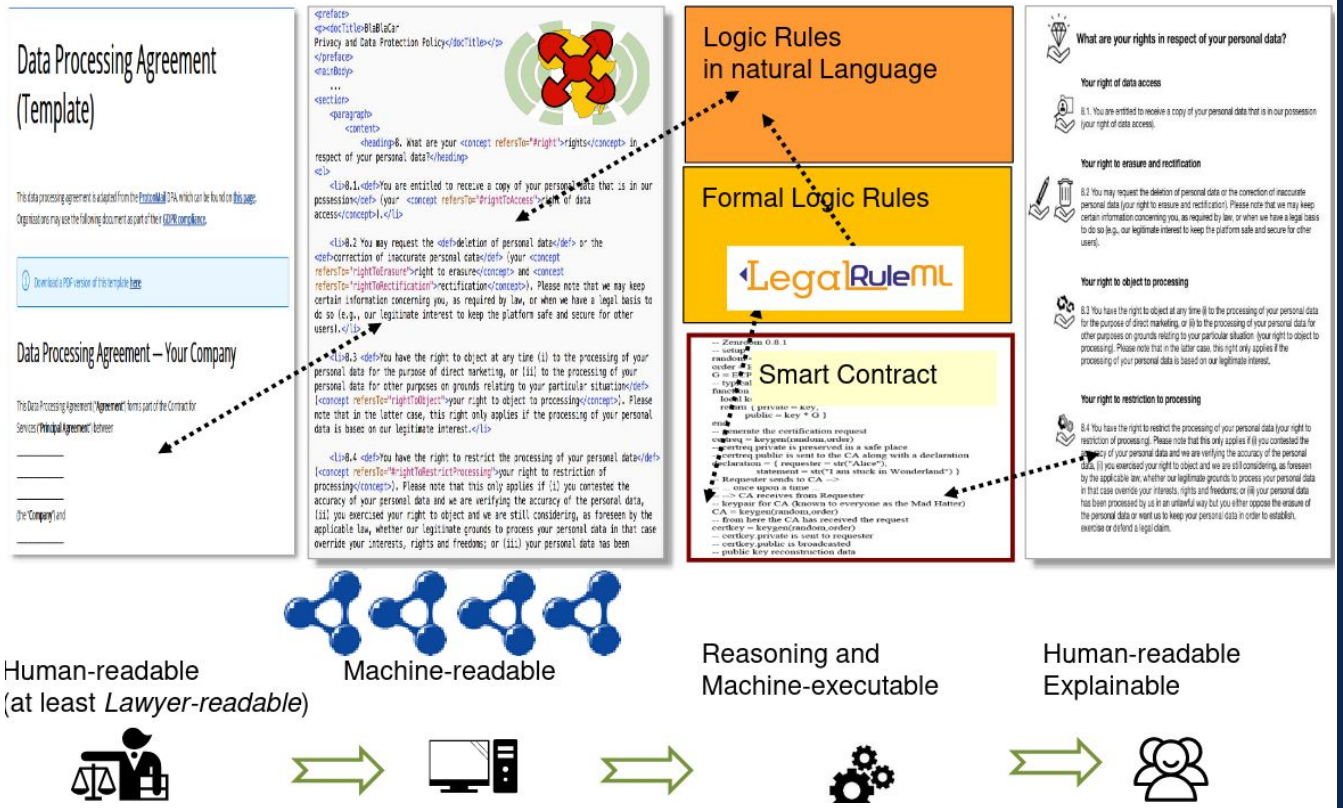


Definizione denotativa di Intelligible Contracts

```
Intelligible Contract::=
     UID and
     Context+ and,
     Document+ and,
     Execution Report+

UID::= URI => HASH

Context::= UID and Legal Context+ and Operational Context+
     Legal Context::= (Legal Document Ref  or Legal Document)+
     Operational Context::= Op Environment Ref+ and Op Code Ref+
          Operational Environment ::= URI
          Operational Code::= UID  and Bitcode+

Document::= UID and (Generic Document or Generic Document Ref)+
or (Legal Document or Legal Document Ref)+
     Generic Document::= Bitcode+
     Legal Document::= Legal Prose+ and Metadata+
          Legal Prose::= Human Natural Language Statement +
          Metadata::= <Legal Prose,Operational
Context,Descript.>

Execution Report::= UID and Document+
```

(::=) significa "è definito come";
(*) significa "zero o più occorrenze";
(+) significa "una o più occorrenze";
se non ci sono né (*) né (+), allora deve esserci "esattamente un'occorrenza";
(x and y) significa "sia x che y";
(x or y) significa "x o y o entrambi".
(A ⇒ B) per indicare che "A e B sono entrambi obbligatori e B è in funzione del contenuto di A".

Scenario


Data Processing Agreement (Template)
Data Processing Agreement – Your Company
Logic Rules in natural Language
Formal Logic Rules
LegalRuleML
Smart Contract
What are your rights in respect of your personal data?
Your right of data access
Your right to erasure and rectification
Your right to object to processing
Your right to restriction to processing
Human-readable (at least Lawyer-readable)
Machine-readable
Reasoning and Machine-executable
Human-readable Explainable