DRAFT
V.1.0
GIUGNO
2023

# Caratteristiche degli Smart Contract

BANCA D'ITALIA,
UNIVERSITÀ CATTOLICA DEL SACRO CUORE,
UNIVERSITÀ ROMA TRE

## EXECUTIVE SUMMARY

In questo documento illustriamo i risultati di un lavoro condotto sugli smart contract, intesi nelle due accezioni di *smart contract code*, ovvero di un programma *software* che viene memorizzato, verificato ed eseguito su una blockchain[1] o, più in generale, su un registro distribuito (*Distributed Ledger Technology* o DLT[2]), e di *smart legal contract*, ovvero di uno strumento che insiste sulla tecnologia a registro distribuito per articolare, verificare e applicare un accordo tra le parti. Il lavoro si propone di evidenziare le questioni che gli smart contract pongono sul piano sia giuridico sia tecnico, con l'obiettivo di indicare, in una seconda fase, linee guida deducibili dalla migliore prassi. Il documento è strutturato in due parti, divise in Sezioni.

In primo luogo (PARTE I), illustriamo i principali profili giuridici affrontati dalla dottrina italiana e straniera in relazione all'utilizzo degli smart contract, qualificati secondo la classificazione comunemente utilizzata di smart contract e smart legal contract.

Sulla base delle fonti normative esistenti, nell'ambito dell'Unione europea e nazionale (Sezione I), esponiamo le analisi condotte dalla dottrina (Sezione II) e le soluzioni offerte. Vengono poi proposti alcuni riferimenti extra-europei nei termini ritenuti utili per la comprensione di temi aperti (Focus 1).

Un'analisi degli smart contract anche sul piano della tecnologia (PARTE II) completa la rappresentazione dei problemi sollevati dalla dottrina per individuare possibili soluzioni attraverso una configurazione specifica degli smart contract o degli smart legal contract, tenendo in debito conto, ove del caso, le caratteristiche rilevanti della tecnologia blockchain su cui insistono.

L'analisi tecnica si articola quindi in due sezioni. La prima, ha ad oggetto le blockchain, la seconda gli smart contract.

Quanto alla prima (Sezione I), partendo dall'analisi dello stato dell'arte, il lavoro elenca e descrive le caratteristiche principali della tecnologia blockchain. Inoltre, in conclusione, si propone un approccio metodologico che potrebbe essere utilizzato per l'acquisizione delle informazioni necessarie per descrivere e analizzare le diverse blockchain in relazione alle caratteristiche individuate.

Quanto alla seconda (Sezione II), abbiamo analizzato, sul piano tecnico, le principali componenti degli smart contract, e abbiamo condotto una prima analisi su temi quali la sicurezza e la vulnerabilità degli smart contract.

[1] La blockchain è un particolare tipo di DLT in cui le transazioni del registro memorizzate sono raggruppate in una struttura dati a "blocchi". Tali blocchi sono collegati tra loro per via crittografica, creando così una registrazione in ordine cronologico e non modificabile di tutte le transazioni effettuate fino a quel momento [1]

[2] La Banca d'Italia [1, 2, 3] ha affrontato il tema sulla base della considerazione secondo cui "Lo sviluppo di tecnologie decentralizzate nel campo dei servizi finanziari poggia sul ruolo centrale della crittografia e della tecnologia dei registri distribuiti (Distributed Ledger Technology – DLT/blockchain). I due paradigmi tecnologici, crittografia sono fortemente complementari. Il primo consente di proteggere le informazioni relative alle transazioni e la loro non ripudiabilità; esso garantisce l'integrità e, se previsto, la confidenzialità delle medesime informazioni ed è alla base del meccanismo di autorizzazione delle transazioni. Il secondo (DLT/blockchain) consiste in un registro elettronico condiviso in cui i dati sono protetti sia tramite tecniche crittografiche sia attraverso la "ridondanza" (copie delle stesse informazioni possono essere validate e archiviate presso tutti i partecipanti attivi al registro)" [1]

## PARTE I - GLI SMART CONTRACT TRA TECNOLOGIA E DIRITTO

### Introduzione

Gli smart contract rappresentano ad oggi l'applicazione più nota della tecnologia a registro distribuito, unitamente alle criptovalute e alla c.d. tokenizzazione di asset, intesa come rappresentazione di asset reali sotto forma di token digitali emessi sulla blockchain, che ne rappresentano il valore economico intrinseco e i diritti di possesso [4, 5].

Il primo a teorizzare la progettazione di smart contract, prima ancora dell'avvento della tecnologia blockchain, è stato l'informatico e crittografo statunitense Nick Szabo[3] agli inizi degli anni Novanta, con lo scopo di: soddisfare condizioni contrattuali comuni (es: termini di pagamento, diritti, riservatezza), minimizzare il rischio di inadempimento, limitare il ricorso a intermediari fidati o a meccanismi di *enforcement* tradizionali [6, 7, 8, 9]. Secondo questo schema originario, gli smart contract consentono di ridurre, o addirittura eliminare, i costi eventualmente connessi [10].

Questa idea è stata tradotta in concreto per la prima volta nel sistema basato su un registro distribuito [11]. Questo ha permesso agli sviluppatori di interagire con smart contract per realizzare applicazioni eseguite in maniera decentralizzata, direttamente sulla tecnologia blockchain.

In questo lavoro ci focalizziamo su smart contract come programmi eseguiti su piattaforme blockchain. Inoltre, l'analisi si concentra sulle forme di smart contract (nelle due accezioni) utilizzate nei settori assicurativo, bancario e finanziario, che presentano esigenze specifiche e richiedono caratteristiche particolari, imposte (anche) dalla normativa di settore.

### 1. "Smart contract code" e "Smart legal contract"

L'espressione "smart contract", ormai associata prevalentemente alle piattaforme blockchain[4], non ha una definizione univoca [12, 13, 14]. Alcuni richiamano il concetto di "macchine autonome", altri quello di "contratti tra parti memorizzati su una blockchain", altri ancora associano lo smart contract più in generale a "qualsiasi calcolo che avviene su una blockchain". I tentativi definitori possono essere ricondotti principalmente a due macro-categorie [15, 16, 17, 18, 19]: a) *smart contract code* ("codice di contratto intelligente"), per identificare una tecnologia specifica o un codice che viene memorizzato, verificato ed eseguito su una blockchain; e b) *smart legal contract* ("contratti giuridici intelligenti"), quale applicazione specifica di questa tecnologia come complemento, o sostituto, dei contratti tradizionali.

[3] La definizione fornita da Nick Szabo è la seguente: "A smart contract is a computerized transaction protocol that executes the terms of a contract. The general objectives of smart contract design are to satisfy common contractual conditions (such as payment terms, liens, confidentiality, and even enforcement), minimize exceptions both malicious and accidental, and minimize the need for trusted intermediaries. Related economic goals include lowering fraud loss, arbitration and enforcement costs, and other transaction costs". Szabo aveva anche descritto un sistema decentralizzato di generazione e scambio di moneta digitale denominata "Bit gold", precorritrice dell'odierno e più famoso Bitcoin. Szabo ha altresì auspicato l'utilizzo dello smart contract in ulteriori ambiti, come l'acquisto di un bene a rate, quale l'automobile, ipotizzando un sistema per cui all'inadempimento del compratore consegua un automatico blocco del veicolo attraverso, appunto, l'interazione di un software e un hardware capaci di riconoscere l'avverarsi di una condizione prestabilita (ad esempio, il mancato o ritardato pagamento della rata di periodo), senza che sia necessario, né possibile, un ulteriore intervento umano perché si realizzino le relative conseguenze.

[4] A distanza di quasi vent'anni, Vitalik Buterin nel White Paper di Ethereum (Pubblicato in rete il 6 aprile del 2014: V. Buterin, A Next-Generation Smart Contract and Decentralized Application Platform), descrive gli smart contract dal punto di vista tecnico-informatico

La distinzione rileva al fine di individuare le funzioni di uno smart contract [20, 21]. **Con riferimento allo smart contract code, l'indagine verte sulle caratteristiche tecniche sia del linguaggio-codice utilizzato sia della blockchain su cui opera.**

**Con riferimento allo smart legal contract, invece, l'indagine ha per oggetto l'idoneità dello smart contract a consentire alle parti di un contratto di negoziarlo, stipularlo ed eseguirlo sulla blockchain.**

### 1.1 Smart contract come "Smart contract code"

La nozione di "smart contract code" [5] viene utilizzata abitualmente dagli sviluppatori che operano sulla tecnologia blockchain[6]. Gli smart contract hanno caratteristiche uniche rispetto ad altri tipi di software perché: (i) il programma è registrato sulla blockchain e ne acquisisce quindi le caratteristiche di immutabilità, sicurezza e trasparenza[7]; (ii) l'esecuzione del programma è deterministica e il risultato dell'esecuzione è memorizzato sulla blockchain; (iii) il programma può controllare le attività della blockchain e quindi può funzionare da deposito, nonché trasferire asset digitali (tra cui, le criptovalute); (iv) il programma viene eseguito dalla blockchain e – presupposte determinate caratteristiche della blockchain – è impermeabile a interferenze circa il suo funzionamento.

Il codice degli smart contract non ha le caratteristiche tipiche di un contratto. Inoltre, in molti casi gli smart contract non hanno una funzione autonoma, ma sono strumentali al successo di un'applicazione più ampia[8], eseguita sulla blockchain - e quindi decentralizzata[9][13].

### 1.2 Smart contract come "Smart legal contract"

Tra i giuristi il termine "smart contract" viene spesso inteso come uno strumento che insiste sulla tecnologia blockchain per articolare, verificare e applicare un accordo tra le parti, con l'obiettivo di integrare, o in alcuni casi sostituire, i contratti tradizionali. È il cosiddetto "Smart legal contract". Si tratta, in definitiva, di una combinazione tra codice di programmazione e linguaggio giuridico[10] [15].

---

[5] Il riferimento a "Smart contract code" è stato inizialmente utilizzato nella documentazione di Ethereum, su stackexchange e in articoli di carattere tecnico. Oggi, però, il termine è usato genericamente per riferirsi a qualsiasi programma complesso che viene memorizzato ed eseguito su una blockchain.

[6] Mentre le prime blockchain sono state progettate per eseguire un piccolo insieme di operazioni semplici - principalmente, transazioni di un token simile a una valuta - sono state poi sviluppate tecniche che consentono alle blockchain di eseguire operazioni più complesse, definite in veri e propri linguaggi di programmazione.

[7] Il codice dello smart contract è salvato sul registro condiviso da tutti i partecipanti della rete, quindi facilmente consultabile e verificabile.

[8] Il riferimento è alle DApp (Decentralized Applications), definite come applicazioni che possono operare in modo autonomo, tipicamente attraverso l'uso di smart contracts, che vengono eseguiti su un sistema informatico decentralizzato, una blockchain o un altro sistema di libro mastro distribuito. Ogni DApp o altra applicazione basata su blockchain è costruita utilizzando il codice degli smart contract per eseguire operazioni sulla blockchain scelta. Ciò che le differenzia, dunque, dalla maggior parte delle comuni applicazioni è che il proprio codice di back-end è in esecuzione su una rete peer-to-peer decentralizzata.

[9] La terminologia "smart contract" è dibattuta in dottrina perché enfatizza un singolo caso d'uso ristretto. I programmi di smart contract possono detenere essi stessi saldi di criptovaluta o addirittura controllare altri programmi di smart contract. Una volta creati, possono agire autonomamente quando vengono chiamati a compiere un'azione. Per questo motivo, molti preferiscono il termine "smart agent", analogo al concetto più generale di agente software.

[10] I contratti commerciali contengono spesso clausole che proteggono le parti da vari casi limite e che non sempre si prestano a essere rappresentate ed eseguite tramite codice. Immaginiamo che un fornitore di beni stipuli uno smart contract con un rivenditore. I termini di pagamento potrebbero essere definiti in codice ed eseguiti automaticamente al momento della consegna. Ma il rivenditore probabilmente insisterà affinché il contratto includa una clausola di manleva, in base alla quale il fornitore si impegna a manlevare e mantenere indenne il rivenditore da azioni di risarcimento derivanti da un prodotto difettoso. Non avrebbe senso rappresentare questa clausola nel codice, poiché si tratta di una clausola che non deve essere auto-eseguita, ma interpretata e applicata dalle parti e, in caso di controversia, dal giudice competente.

## Sezione I – Le fonti normative

### 2. Disciplina europea degli smart contract

Il legislatore dell'Unione europea ambisce a regolare gli smart contract sin dall'invito nel 2020 del Parlamento Europeo alla Commissione "a valutare lo sviluppo e l'utilizzo delle tecnologie di registro distribuito, comprese le blockchain e, in particolare, gli smart contract ".[11] Il Parlamento europeo ha riconosciuto in quell'occasione l'utilizzo degli smart contract e la mancanza, allo stato, di un quadro giuridico; ha quindi presentato proposte, tra cui la definizione di norme riguardo il loro utilizzo, la possibilità di intervenire nelle transazioni in caso di operazioni finanziarie sospette e specifiche misure di tutela per piccole e medie imprese che decidano di utilizzare tali strumenti. La Risoluzione adottata dal Parlamento rientra nella Blockchain Strategy[12] [22]: l'Unione Europea non solo intende regolamentare i contratti automatizzati ma, riconoscendone il potenziale innovativo anche nelle transazioni online, intende supportare le imprese e le tecnologie europee nel settore.

La Commissione europea ha istituito, altresì, il Progetto pilota "European Blockchain Observatory and Forum", gestito dalla Direzione generale per le reti di comunicazione, i contenuti e la tecnologia (DG CONNECT), con la finalità di: (i) accelerare lo sviluppo dell'innovazione blockchain in Europa; (ii) monitorare le iniziative blockchain in Europa; (iii) formulare raccomandazioni sul ruolo che l'UE potrebbe svolgere nella blockchain. Tra i Report prodotti dal Forum, ve n'è anche uno specifico sugli smart contract del 1° novembre 2022 [23]. Il documento elenca i benefici per l'adozione su larga scala degli smart legal contract rispetto ai normali contratti; al contempo, ne evidenzia i limiti e propone soluzioni.

Per poter inserire uno smart legal contract nella blockchain, il Report "Smart contracts" dello European Blockchain Observatory and Forum sottolinea la necessità che il linguaggio giuridico sia interamente tradotto in codice informatico. Ciò può costituire un problema se si considera il fatto che i giuristi di norma non hanno le competenze tecniche degli sviluppatori di codice e viceversa. È tuttavia "necessario mantenere un certo livello di fiducia e di competenza per garantire che tutte le parti possano fidarsi del fatto che il codice dello smart contract rifletta davvero il contenuto e lo scopo legale [dello stesso]" (p. 29 del Report). Tra le criticità, emerge la difficoltà di valutare gli eventi del mondo reale rispetto alla quale nel Report si individua l'oracolo[13] quale strumento per collegare dati affidabili alla blockchain.

Altre criticità si rinvengono in relazione alla cybersicurezza e alla vulnerabilità agli attacchi, nonché nel rischio di frodi e nell'ambito della tutela della privacy: rispetto a queste, il Report propone specifiche tecniche per prevenire rischi operativi, a partire da strumenti di auditing di sicurezza preventivi e test di finzione di attacco ("penetration tests"). Con specifico riferimento al settore finanziario, anche gli strumenti individuati in generale nel Regolamento DORA possono rivelarsi utili. DORA uniforma e armonizza le regole sulla sicurezza delle reti e dei sistemi informativi che sostengono i processi commerciali delle entità finanziarie. Tra gli obiettivi di DORA vi sono da un lato il raggiungimento di standard elevati in materia di Cyber Security e di governance dei rischi ICT

[11] Risoluzione del Parlamento europeo del 20 ottobre 2020 recante raccomandazioni alla Commissione sulla legge sui servizi digitali: adeguare le norme di diritto commerciale e civile per i soggetti commerciali che operano online (2020/2019(INL)).
[12] Per maggiori informazioni si consulti il sito della Commissione Europea https://digital-strategy.ec.europa.eu/en/policies/blockchain-strategy
[13] L'oracolo ha lo scopo di garantire la connessione tra quello che avviene sulla blockchain e quanto avviene fuori, 'certificando' l'originalità e la correttezza dei dati immessi sulla blockchain. In particolare, nei casi di oracolo c.d. oggettivo, l'input fornito dall'oracolo proviene da un software o da uno strumento informatico, che postula transazioni standard, semplici e automatiche, dipendenti dalla verificazione di un inopinabile fatto oggettivo (se effettivamente si sia verificato quel fatto A, cui consegue l'evento B).

e cyber, e dall'altro l'onere, per gli enti finanziari, di definire un quadro organizzativo e procedurale di governance, integrata nel più ampio quadro dei rischi operativi[14].

Da un punto di vista normativo, per l'integrazione degli smart legal contract su larga scala, il Report sottolinea le sfide poste per i consumatori dal linguaggio utilizzato, per cui sembra necessario (i) assicurare la fruibilità delle informazioni, quale che sia il linguaggio applicato e individuare meccanismi che consentano di tenere conto della posizione giuridica del consumatore; (ii) applicare rigorose procedure di KYC e controlli AML/CFT.

Inoltre, considerata la natura di semi-irreversibilità dei dati registrati sulla blockchain, qualsiasi errore nel codice può richiedere tempo e costi elevati per essere corretto. Dal punto di vista giuridico, è necessario, pertanto, portare a conoscenza del contraente il significato contrattuale di nozioni giuridiche specifiche (es: buona fede) e lasciare spazio alla flessibilità.

[14] Nello specifico, DORA si fonda su cinque pilastri fondamentali:

**A) ICT Risk Management:** per acquisire un elevato livello di resilienza operativa digitale, gli enti finanziari predispongono un quadro di gestione e di controllo interno che "garantisce una gestione efficace e prudente di tutti i rischi informatici" attribuendo all'organo di gestione interno il compito di definire e di approvare l'attuazione di tutte le disposizioni sul quadro per la gestione dei rischi informatici, nonché di vigilare sulla loro attuazione, assumendone la piena responsabilità. .

**B) ICT - Related Incident Management**: l'Incident Management è un processo necessario per evitare o minimizzare impatti di tipo economico e reputazionale, dovuti a un incidente cyber, e poter quindi ripristinare nel più breve tempo possibile la normale erogazione dei servizi. Esso si basa essenzialmente sulla standardizzazione delle attività di classificazione e segnalazione degli incidenti ICT.

**C) Digital Operational Resilience Testing**: per garantire un'adeguata gestione dei rischi ICT, tra le priorità degli enti finanziari vi è la definizione di un programma di test di resilienza operativa digitale. Per il corretto monitoraggio dell'efficacia della strategia di resilienza, i test di resilienza operativa digitale devono essere condotti tenendo in considerazione l'evoluzione delle minacce informatiche.

**D) ICT third-party Risks Management**: la gestione dei rischi di terze parti nel settore ICT ha l'obiettivo di fornire tutti i requisiti per le istituzioni finanziarie e i fornitori di servizi ICT per garantire un solido monitoraggio dei rischi ad essi associati. In questo ambito, le Autorità di vigilanza europee avranno il compito di: condurre ispezioni off-site e on-site, richiedere informazioni, rilasciare raccomandazioni e richieste, nonché imporre sanzioni.

**E) Information Sharing**: uno degli obiettivi di DORA consiste nell'incoraggiare lo scambio delle informazioni sulle minacce in ambito finanziario, tramite l'istituzione di un programma su base volontaria per consentire agli enti finanziari di stabilire accordi per la condivisione e lo scambio di informazioni sulla cyber-threat intelligence.

**FOCUS 1 – Smart contract e la proposta di Regolamento sull'Intelligenza Artificiale**

Come recentemente ricordato dall'OECD, nel settore finanziario sono sempre più frequenti i casi di applicazioni di blockchain integrate con sistemi di intelligenza artificiale. Questo è anche il caso degli smart contract. Da ciò deriva la rilevanza, anche ai fini di regolazione degli smart contract, della proposta di regolamento sull'intelligenza artificiale (AI Act), nel testo attuale, tuttora in discussione, anche se l'ambito di applicazione nel settore finanziario sia, alla luce dell'ultima versione, alquanto limitato.

Nella versione inizialmente proposta, tuttora oggetto di discussione[15] e potenzialmente soggetta a modifiche anche rilevanti, le uniche applicazioni finanziarie espressamente regolate dalla proposta sono i sistemi di *credit scoring,* annoverati tra i sistemi ad alto rischio di cui all'Allegato III.

Seppur a livello di mero considerando, la proposta di Regolamento attribuisce la disciplina e la vigilanza dei sistemi di intelligenza artificiale usati nel settore finanziario alla normativa di settore e alle rispettive autorità di vigilanza.

Seppur in larga misura non applicabili al settore finanziario, i requisiti dettati dalla proposta dell'AI Act, in assenza di requisiti specifici per la disciplina dell'intelligenza artificiale finanziaria, appaiono benchmark utili ai fini della elaborazione sia della disciplina di secondo livello, sia delle migliori prassi in materia di tecnologia finanziaria e dunque anche di smart contract. Tra i requisiti che l'AI Act pone per i sistemi ad alto rischio si ricordano i requisiti di data governance, di *logging* automatico, di gestione del rischio, di trasparenza e di supervisione umana.

Significativo, nel contesto degli smart contract, è il requisito di trasparenza, funzionalmente connesso al requisito di supervisione umana.

Quanto alla trasparenza, l'AI Act richiede che i sistemi siano strutturati in modo da consentire agli utenti la "comprensione e l'uso appropriato del sistema". Pertanto, l'art. 13 dell'AI Act richiede alle imprese l'adozione di sistemi che consentano di "comprendere" la logica alla base di un output rilasciato, come avviene per i modelli XAI.

Di conseguenza, il requisito della trasparenza impone alle imprese l'onere di scegliere il sistema di IA da integrare negli smart contracts sulla base di un'analisi costi-benefici, che soppesi i costi dell'"oscurità" del sistema rispetto alla efficienza.

Il riferimento alla "comprensibilità" per l'utente contenuto nella proposta di regolamento richiede la conoscibilità del sistema da parte di un soggetto/utente comune. Ciò implica il collegamento del requisito della comprensibilità con il contesto in cui il modello viene utilizzato. Tuttavia, l'assenza, nell'AI Act, di un diritto di informazione direttamente azionabile da parte dell'utente, qualificato ad esempio come un diritto di accesso alle informazioni relative al sistema impiegato, impedisce un effettivo controllo dell'utente sulle informazioni rese dal modello al fine della sua "comprensibilità". Ciò rischia di compromettere in parte il raggiungimento dell'obiettivo di trasparenza dichiarato dall'AI Act.

L'assenza, nella proposta di regolamento sull'IA, di diritti individuali di trasparenza può essere colmata, nei casi di trattamento di dati personali, dai diritti riconosciuti ai sensi del GDPR. Gli artt. 12-15 GDPR, nonché l'art. 22 GDPR prevedono infatti specifici diritti di accesso a "informazioni significative sulla logica utilizzata" nei sistemi automatizzati di trattamento dei dati personali. Questi diritti si applicano, ad esempio, ai modelli di *social scoring* utilizzati nel settore finanziario, che si basano in larga misura sul trattamento di dati personali.

Quanto ai modelli di consumer *credit scoring*, la nuova proposta di direttiva in materia di credito al consumo estende i diritti soggettivi di cui al GDPR al caso dei sistemi di IA utilizzati per valutare il punteggio o il merito creditizio dei consumatori, nella forma di un diritto di "ottenere informazioni significative sulla valutazione effettuata e sul funzionamento del trattamento automatizzato utilizzato, comprese, tra l'altro, le principali variabili, la logica e i rischi coinvolti, nonché il diritto di esprimere il proprio punto di vista e di contestare la valutazione del merito creditizio e la decisione", insieme a un diritto di "intervento umano".

Oltre al riferimento ai diritti di informazione di cui al GDPR e alla proposta di direttiva in materia di credito al consumo, la trasparenza dei modelli di IA utilizzati in materia finanziaria trova fondamento nei principi generali del miglior interesse e dell'adeguatezza informativa rispetto al cliente.[16] Come noto, questi principi richiedono che le informazioni fornite dagli enti finanziari siano adeguate al cliente che le riceve e dunque in linea con le sue esigenze informative.

Le informazioni "adeguate" sui sistemi di IA impiegati dovrebbero essere funzionali a fornire al cliente una comprensione delle modalità con cui l'IA determina la fornitura del prodotto o del servizio finanziario finale, aumentando così la consapevolezza dei clienti sulle tecnologie utilizzate, ad esempio, per la determinazione del tasso di credito o la formulazione di una consulenza in materia di investimenti.

Di seguito una breve rassegna della disciplina primaria e secondaria già adottata (2.1) e ancora in discussione (2.2).

### 2.1 Quanto alla DLT[17] (de iure condito)

Il Regolamento (UE) DLT Pilot 2022/858 prevede un regime pilota temporaneo (con decorrenza dal 23 marzo 2023) comune all'Unione europea per i servizi finanziari basati sulla tecnologia DLT. L'obiettivo è rimuovere gli ostacoli normativi all'emissione, alla negoziazione e al regolamento di strumenti finanziari emessi in forma digitale e supportare le autorità di regolamentazione ad acquisire esperienza nell'uso della DLT.

Il Regolamento si occupa, per quanto qui interessa:

- della concessione, della revoca e della modifica delle autorizzazioni a operare, comprese le esenzioni e le misure compensative o correttive, per la gestione delle infrastrutture di mercato DLT (identificate dal Regolamento come "sistema multilaterale di negoziazione DLT, sistema di regolamento DLT o sistema di negoziazione e regolamento DLT", art. 2, p.to 5);
- del funzionamento e della supervisione dell'infrastruttura di mercato DLT da parte delle autorità competenti;
- della cooperazione tra i gestori di infrastrutture di mercato DLT, le autorità nazionali e l'ESMA.

Per operare sulla base del Regolamento, i gestori di infrastrutture di mercato DLT devono soddisfare specifici requisiti e fornire idonee garanzie. Queste condizioni sono volte a preservare: (i) la tutela degli investitori, (ii) l'integrità del mercato, (iii) la stabilità finanziaria del sistema.

[15] Sebbene il 14 giugno il Parlamento europeo abbia espresso una posizione negoziale favorevole sul testo di compromesso, non è ancora chiaro se si possa attendere l'adozione finale dell'atto entro la fine del 2023 [24]

[16] Ai sensi dell'art. 5, comma 3 Codice del Consumo: "Le informazioni al consumatore, da chiunque provengano, devono essere adeguate alla tecnica di comunicazione impiegata ed espresse in modo chiaro e comprensibile, tenuto anche conto delle modalità di conclusione del contratto o delle caratteristiche del settore, tali da assicurare la consapevolezza del consumatore".

[17] Sulla definizione di DLT si vedano le note nn. 1 e 2.

Sulla conformità a tali requisiti e condizioni nonché sull'adeguatezza del tipo di tecnologia utilizzata, l'ESMA si esprime in sede di concessione dell'autorizzazione con parere non vincolante (art. 8, par. 7; art. 9, par. 7; art. 10, par. 7).

I gestori delle infrastrutture che utilizzano la DLT[18] devono, tra l'altro:

- stabilire regole sull'uso della tecnologia utilizzata, attraverso piani aziendali chiari e dettagliati e una documentazione scritta disponibile al pubblico, aggiornata e dettagliata (art. 7, par. 1[19]). L'adeguatezza della tecnologia DLT a soddisfare la normativa europea sembrerebbe costituire l'elemento principale sulla cui base il gestore è autorizzato a operare;
- fornire ai partecipanti, agli emittenti e ai clienti informazioni chiare e non ambigue;
- garantire sicurezza, continuità e costante trasparenza, disponibilità, affidabilità di tutti i dispositivi informatici e cibernetici relativi all'uso della tecnologia DLT, "compresa l'affidabilità degli smart contract utilizzati nell'infrastruttura di mercato DLT. Tali dispositivi garantiscono inoltre l'integrità, la sicurezza e la riservatezza di tutti i dati memorizzati dai gestori in questione, nonché che tali dati siano disponibili e accessibili" (art. 7, co 4)[20];
- predisporre specifiche procedure di gestione del rischio operativo (art. 7, par. 4, p.to 2);
- separare i fondi e fornire garanzie per gli strumenti finanziari DLT che li detengono. Sulle eventuali perdite dei fondi e delle garanzie grava la responsabilità sullo stesso gestore DLT (art. 7, par. 6);
- pianificare una tempestiva e chiara "strategia di transizione": si ha una strategia di transizione o di riconversione delle operazioni di tecnologia a registro distribuito in infrastrutture di mercato tradizionali nel caso in cui: (i) il valore totale degli strumenti finanziari DLT raggiunga i 9 miliardi di euro (art. 3, par. 3); (ii) si verifichi una cessazione volontaria o involontaria dell'attività dell'infrastruttura di mercato DLT; (iii) sia revocata o altrimenti sospesa un'autorizzazione specifica o l'esenzione concessa a norma del DLT Pilot (art. 7, par. 7).
  In materia di revoca o sospensione dell'autorizzazione, il caso più frequentemente indicato nel regolamento è quello in cui nel "funzionamento della tecnologia a registro distribuito utilizzata o nei servizi e nelle attività forniti dal gestore [..] è stato rilevato un vizio che rappresenta un rischio per la tutela degli investitori, l'integrità del mercato o la stabilità finanziaria e il vizio ha

[18] Questi i tre soggetti individuati dal DLT Pilot tra i gestori delle infrastrutture DLT:
(i) il gestore di un MTF DLT, ossia di un sistema multilaterale di negoziazione, che ammette alla negoziazione solo strumenti finanziari DLT. È soggetto ai requisiti che si applicano a un sistema multilaterale di negoziazione a norma del regolamento (UE) n. 600/2014 del Parlamento europeo e del Consiglio del 15 maggio 2014, relativo ai mercati degli strumenti finanziari e della direttiva 2014/65/UE del Parlamento europeo e del Consiglio, del 15 maggio 2014, relativa ai mercati degli strumenti finanziari;
(ii) il depositario centrale titoli (CSD) di un SS DLT, un «sistema di regolamento DLT»: regola operazioni in strumenti finanziari DLT contro pagamento o consegna. Il CSD di un SS DLT è soggetto ai requisiti che si applicano a un CSD che gestisce un sistema di regolamento titoli a norma del regolamento (UE) n. 909/2014;
(iii) il gestore di un TSS DLT, un «sistema di negoziazione e regolamento DLT»: un TSS DLT o un SS DLT che combina i servizi prestati da un MTF DLT e da un SS DLT. È soggetto ai requisiti che si applicano a un sistema multilaterale di negoziazione a norma del regolamento (UE) n. 600/2014 e della direttiva 2014/65/UE.

[19] "[i] gestori [..] stabiliscono o documentano, come appropriato, regole di funzionamento della tecnologia a registro distribuito che utilizzano, comprese le regole sull'accesso al registro distribuito, sulla partecipazione dei nodi di validazione, sulla risoluzione dei potenziali conflitti di interessi e sulla gestione del rischio, tra cui eventuali misure di attenuazione volte a garantire la tutela degli investitori, l'integrità del mercato e la stabilità finanziaria".

[20] Il Considerando 41 precisa che: "Le infrastrutture di mercato DLT dovrebbero disporre di specifici dispositivi informatici e cibernetici efficaci riguardanti l'uso della tecnologia a registro distribuito. Tali strumenti dovrebbero essere proporzionati alla natura, alla portata e alla complessità del piano aziendale del gestore dell'infrastruttura di mercato DLT. Tali strumenti dovrebbero inoltre garantire la continuità e la costante trasparenza, disponibilità, affidabilità e sicurezza dei servizi forniti, compresa l'affidabilità di eventuali smart contract utilizzati, indipendentemente dal fatto che tali smart contract siano creati dalla stessa infrastruttura di mercato DLT o da terzi in seguito a procedure di esternalizzazione. Le infrastrutture di mercato DLT dovrebbero inoltre assicurare l'integrità, la sicurezza, la riservatezza, la disponibilità e l'accessibilità dei dati memorizzati nel registro distribuito. L'autorità competente di un'infrastruttura di mercato DLT dovrebbe essere autorizzata a chiedere una verifica volta a garantire che gli strumenti informatici e cibernetici generali dell'infrastruttura di mercato DLT siano adatti allo scopo."

un peso maggiore rispetto ai vantaggi offerti dai servizi e dalle attività in fase di sperimentazione" (cfr. artt. 8, par 12; 9, par. 12; 10, par. 12);

- cooperare strettamente con le autorità competenti designate dagli Stati membri dell'Unione e presentare una relazione semestrale.

Inoltre, i gestori, per essere autorizzati ai sensi del DLT Pilot, devono dimostrare di:

- rispettare requisiti prudenziali sufficienti a far fronte alle passività e risarcire i clienti;
- aver adottato disposizioni per la custodia delle attività DLT dei clienti;
- aver predisposto e implementato misure per garantire la protezione degli investitori e per gestire i reclami e i ricorsi dei clienti, anche mediante mezzi digitali. Lo IOSCO (Organizzazione Internazionale delle Autorità di controllo dei mercati finanziari) ha pubblicato a gennaio 2021 un rapporto [25] che descrive buone pratiche per lo sviluppo e il miglioramento delle procedure e dei meccanismi di gestione dei reclami per gli investitori al dettaglio. Lo IOSCO individua come prioritaria la protezione degli investitori attraverso l'accesso a meccanismi di ricorso indipendenti, convenienti, equi, responsabili, tempestivi ed efficienti.

### 2.1.1 Quanto alla disciplina AML/CFT (Anti-Money Laundering/Countering the Financing of Terrorism)

In ragione delle caratteristiche tipiche della tecnologia blockchain di cui si avvalgono, gli smart contract, sotto specifiche condizioni, potrebbero concorrere all'implementazione di misure di mitigazione dei rischi di riciclaggio e di finanziamento del terrorismo. Si tratta di una tematica che può essere ulteriormente approfondita tenuto conto delle interrelazioni esistenti tra gli smart contract e le criptovalute e l'evoluzione della disciplina antiriciclaggio applicabile a queste ultime.

La quinta Direttiva antiriciclaggio (Direttiva 2018/843, cd. 'AMLD5') ha incluso tra i soggetti obbligati alcuni prestatori di servizi in valute virtuali (i.e. prestatori di servizi di exchange e di portafoglio digitale). Il legislatore italiano aveva già introdotto una disciplina simile – in ottica anticipatoria – con il D. lgs. 90/2017 (di recepimento della quarta Direttiva antiriciclaggio). Il d.lgs. 125/2019 (di recepimento della AMLD5) ha poi esteso l'ambito delle attività dei VASP sottoposte a obblighi antiriciclaggio[21] includendovi ogni servizio funzionale all'utilizzo, scambio e conservazione di valute virtuali [22].

Nel 2021 la Commissione europea ha presentato un pacchetto di proposte legislative in materia (c.d. "AML Package"), che estende e definisce in maniera precisa il novero dei soggetti destinatari degli obblighi AML/CFT - includendovi i cd. i crypto-asset service providers (CASPs) nell'accezione che ne fornisce il Regolamento MICA - e introduce regole specifiche per le informazioni che accompagnano i trasferimenti di cripto-attività, al fine di accrescerne la trasparenza e la tracciabilità, in coerenza con gli standard del GAFI.

Nello specifico, la norma impone ai prestatori di servizi relativi alle cripto-attività di raccogliere e rendere accessibili alle Autorità determinate informazioni relative all'ordinante e al beneficiario dei trasferimenti garantendone così la tracciabilità agevolando l'individuazione delle operazioni sospette[23].

[21] In entrambe le occasioni gli interventi normativi nazionali hanno delineato una disciplina più ampia di quella prevista nelle direttive, per l'esigenza di ricomprendere da un lato, i rischi connessi all'utilizzo delle valute virtuali e dall'altro, le Raccomandazione formulate dal GAFI (cfr. la normativa in tema di prevenzione del riciclaggio: autorità, regole e controlli – Quaderni dell'antiriciclaggio – Analisi e Studi n. 20, 2023).

[22] Cfr. art. 1, comma 2, lett. ff e ff-bis del D. Lgs. 231/2007.

[23] È in fase di finalizzazione la proposta di regolamento del Parlamento Europeo e del Consiglio riguardante i dati informativi che accompagnano i trasferimenti di fondi e determinate cripto-attività (rifusione) - COM (2021) 422 final. Uno

Cionondimeno, questo approccio non esaurisce del tutto il rischio connesso all'anonimato delle transazioni, poiché i trasferimenti di valuta virtuale possono realizzarsi anche senza il coinvolgimento di un prestatore di servizi destinatario degli obblighi AML/CFT. Tale rischio può assumere caratteri ancora più accentuati nell'ambito della c.d. *decentralized finance*, nelle cui forme più estreme, sarebbe possibile realizzare[24] piattaforme di servizi in valute virtuali "decentralizzate", che potrebbero non essere facilmente riconducibili a soggetti destinatari degli obblighi AML/CFT. Attualmente alcune piattaforme non adottano tali strumenti di controllo (i.e. procedure di KYC), e risultano quindi più attrattive, per scopi illeciti[25], rispetto ai prestatori di servizi in valute virtuali "centralizzati".

Di ciò è consapevole il legislatore europeo che, nel considerando 9 della quinta Direttiva antiriciclaggio[26], afferma: "*l'inclusione dei prestatori di servizi la cui attività consiste nella fornitura di servizi di cambio tra valute virtuali e valute reali e dei prestatori di servizi di portafoglio digitale non risolve completamente il problema dell'anonimato delle operazioni in valuta virtuale: infatti, poiché gli utenti possono effettuare operazioni anche senza ricorrere a tali prestatori, gran parte dell'ambiente delle valute virtuali rimarrà caratterizzato dall'anonimato*". In altre parole, la normativa vigente non si applica alle transazioni effettuate in assenza di intermediari [85] per le quali emerge chiaramente il problema dell'anonimato; inoltre, non considera le differenze esistenti tra diverse tecnologie blockchain (i.e. private e pubbliche, permissioned e permissionless, ecc.).

L'indagine sugli aspetti tecnici delle blockchain e degli smart contract porta dunque a prendere in considerazione le potenzialità di queste tecnologie ai fini della conservazione delle informazioni e della tracciabilità delle transazioni. Infatti, una transazione eseguita attraverso piattaforme blockchain che presentano caratteristiche tecniche e di governance adeguate al raggiungimento di questi obiettivi, potrebbe essere trasparente e immodificabile nonché sicura e tracciabile. In questo panorama, gli smart contract utilizzati nel settore finanziario su blockchain con le caratteristiche che consentano l'identificazione delle parti della transazione e la tracciabilità delle stesse, possono costituire un'opportunità per banche e intermediari finanziari anche per agevolare l'adempimento degli obblighi AML/CFT. Approfondimenti degli studi in questa direzione potrebbero consentire di individuare sinergie a beneficio non solo dei soggetti obbligati ma anche delle attività delle autorità competenti per il controllo e il monitoraggio delle transazioni.

### 2.1.2 Quanto a iniziative sperimentali: la "European Blockchain Regulatory Sandbox"

Il 14 febbraio 2023 la Commissione europea ha lanciato una "Sandbox" regolamentare (European Blockchain Regulatory Sandbox) per i casi d'uso innovativi che coinvolgono le Distributed Ledger Technologies e/o le Blockchain[27].

L'iniziativa muove dall'esigenza di superare l'attuale incertezza giuridica, determinata da una governance del processo complessa. La nuova metodologia intende semplificare e rafforzare il dialogo tra le autorità di regolamentazione e gli innovatori. Nello specifico, la Commissione individua

---

degli elementi innovativi della proposta di Regolamento prevede che l'insieme delle informazioni sul cedente (proprietario di asset) "viaggi" in maniera contestuale al trasferimento della cripto-attività (la c.d. *travel rule*), indipendentemente dall'importo di cripto-attività oggetto della transazione.

[24] La questione è all'attenzione di diverse autorità; in un recente report del Dipartimento del Tesoro Americano viene riferito che il livello di decentralizzazione perseguito da tali piattaforme sarebbe da verificare caso per caso. Resta infatti un certo grado di centralizzazione, riferibile ad esempio al gruppo di soggetti che manutengono il codice di tali applicazioni o che ne detengono le chiavi amministrative.

[25] Si veda il report dell'Organizzazione Internazionale delle Autorità di controllo dei mercati finanziari citato [27].

[26] Direttiva (UE) 2018/843 del Parlamento europeo e del Consiglio, del 30 maggio 2018, che modifica la direttiva (UE) 2015/849 relativa alla prevenzione dell'uso del sistema finanziario a fini di riciclaggio o finanziamento del terrorismo e che modifica le direttive 2009/138/CE e 2013/36/UE.

[27] https://digital-strategy.ec.europa.eu/en/news/commission-launches-european-regulatory-sandbox-blockchain.

gli ostacoli normativi all'introduzione delle soluzioni, e fornisce consulenza, esperienza e orientamenti normativi in un contesto sicuro e protetto per i partecipanti.

In sostanza, la Commissione si avvarrà degli operatori per approfondire gli aspetti tecnici di queste tecnologie, mentre gli operatori contribuiranno a individuare le migliori pratiche per il mercato[28], secondo il processo della regolazione partecipata (*infra*, **Focus 3 – La regolazione partecipata**).

### 2.2 Quanto agli smart contract (de iure condendo)

La proposta di Regolamento del Parlamento europeo e del Consiglio riguardante norme armonizzate sull'accesso equo ai dati e sul loro utilizzo, pubblicata il 23 febbraio 2022 (Data Act - 'Normativa sui dati') mira a istituire un quadro di misure, anche procedurali, in materia di interoperabilità per la creazione di un mercato interno unico dei dati, in accordo con la Strategia europea sui dati. La governance dei dati costituisce infatti il perno su cui ruota il Programma europeo[29] per la transizione digitale e il Piano d'azione verde.

Tra gli obiettivi della proposta, vi è la definizione di norme per lo "smart contract", inteso come un "programma informatico conservato in un sistema di registro elettronico in cui l'esito dell'esecuzione del programma è registrato nel registro elettronico" (art. 2, punto 16 - "Definizioni"), considerato strumento potenzialmente in grado di "fornire ai titolari e ai destinatari dei dati garanzie del rispetto delle condizioni per la condivisione dei dati" (par. 1 della Relazione – "Contesto della proposta - Motivi e obiettivi della proposta").

Il Data Act mira a regolare tre aspetti centrali tra loro connessi: (i) l'accesso, (ii) l'utilizzo e (iii) l'interoperabilità dei dati. In tutti e tre gli ambiti, la proposta di Regolamento richiama lo smart contract quale strumento a disposizione dell'operatore dello spazio di dati[30].

---

[28] A chiarire la natura non meramente esplorativa, ma di forte impatto concreto dell'iniziativa sono i criteri di selezione dei candidati: la priorità verrà data a casi d'uso più maturi per il cui sviluppo sorgono questioni giuridiche di più ampia rilevanza e già all'attenzione dei regolatori europei. Nello specifico:

- quanto ai soggetti partecipanti al bando (lett. A - "Identity and eligibility of the applicant"), la Sandbox è aperta alle aziende di tutti i settori industriali e agli enti pubblici per progetti che vanno oltre la fase di proof-of-concept e sono già prossimi al mercato o in una fase iniziale di operatività;
- quanto proprio all'idoneità del caso d'uso (lett. B - "Elibigility of the use case"), il bando richiede che questo sia stato validato in un "relevant environment"; inoltre, deve essere determinato nel contesto dei progetti finanziati dall'UE e in particolare dei progetti finanziati nell'ambito dei programmi quadro Horizon 2020 e Horizon Europe;
- quanto al livello di sviluppo del caso d'uso (lett. D – "Maturity of the use case"), il bando specifica che i casi d'uso più vicini alla commercializzazione otterranno un punteggio più alto in base alla valutazione della maturità delle soluzioni tecnologiche nel contesto dei progetti finanziati dall'UE e in particolare dei progetti finanziati nell'ambito dei programmi quadro Horizon 2020 e Horizon Europe;
- quanto al collegamento con questioni normative in tutti i settori industriali (lett. E – "Link with novel regulatory issues across industry sectors") e alla pertinenza con le priorità politiche UE (lett. F - "Relevance with the EU's wider Policy Priorities"), la Commissione verifica in maniera oggettiva la continuità del caso d'uso presentato con le tematiche già sui tavoli di dibattito europei e ancora in corso di sviluppo;
- la Commissione è interessata a sapere se il partecipante è assistito da uno o più regolatori europei e/o nazionali e di indicare quali nello specifico (lett. H - "Regulator support").

[29] Si veda: Commissione europea, Allegati del Programma di lavoro della Commissione 2020 – "Un'Unione più ambiziosa" COM (2020) 37, 29 gennaio 2020, p. 4 (" 2.2. Un'Europa pronta per l'era digitale: Una nuova strategia europea in materia di dati ci consentirà di sfruttare al massimo l'enorme valore dei dati non personali, una risorsa in continua espansione e riutilizzabile nell'economia digitale. [...]
La Commissione inoltre intende presentare "[u]na nuova legge sui servizi digitali [che] rafforzerà il mercato unico dei servizi digitali e contribuirà a fornire alle imprese più piccole la chiarezza giuridica e le condizioni di parità di cui hanno bisogno." […], "riesamin[are] la direttiva sulla sicurezza delle reti e dei sistemi informativi e [p]roporre iniziative volte a rendere la finanza digitale più solida contro gli attacchi informatici, tra cui una proposta concernente le cripto attività".

[30] Cfr: 2018, Libro bianco recante "Raccomandazioni per adottare standard comuni in Europa sulla blockchain e sulle DLT" a cura del CEN (Comitato Europeo di Normazione) e del CENELEC (Comitato europeo per la normazione elettronica); 2018, Parlamento europeo, "How blockchain technology could change our lives" (in cui viene analizzato l'impatto economico e sociale delle blockchain ed evidenziati i benefici); 2018, EBP (European Blockchain Partnership),

(i) Nella proposta di Regolamento, al capo III ("Obblighi per i titolari dei dati tenuti per legge a mettere a disposizione i dati"), l'art. 11 classifica lo smart contract tra le misure tecniche di protezione cui il titolare dei dati[31] può ricorrere per impedire l'accesso non autorizzato ai dati e garantire il rispetto degli articoli 5, 6, 9 e 10[32], nonché delle clausole contrattuali concordate per la messa a disposizione dei dati.
Lo smart contract – unitamente alle altre misure tecniche di protezione eventualmente adottate – non deve tuttavia "ostacolare il diritto dell'utente di fornire efficacemente dati a terzi a norma dell'articolo 5 o qualsiasi diritto di terzi a norma del diritto dell'Unione o della legislazione nazionale di attuazione del diritto dell'Unione di cui all'articolo 8, paragrafo 1" (art. 11, 1, secondo par.).
Il Data Act, letto insieme ai documenti di lavoro che ne hanno preceduto la pubblicazione[33], conferma che nel bilanciamento tra le diverse esigenze, quella della condivisione dei dati – contenuto di un diritto per l'utente, da una parte, e di un obbligo per il titolare, dall'altra – prevale sulla protezione dei dati da un utilizzo improprio (cyber security).

(ii) Il capo VIII del Data Act ("Interoperabilità"[34]) stabilisce condizioni minime per promuovere l'interoperabilità nell'ambito degli smart contract e individua alcune prescrizioni essenziali che gli operatori sono tenuti a osservare.

In particolare:

- **in materia di interoperabilità**, l'art. 28 impone agli operatori che redigono gli smart contract di fornire "i mezzi per consentire l'interoperabilità degli smart contracts nell'ambito dei loro servizi e delle loro attività" (art. 28, par. 1, lett. d). La disposizione prevede una presunzione di conformità per gli smart contract che soddisfano le condizioni di cui alle norme armonizzate adottate da organizzazioni europee di normazione su richiesta della Commissione, conformemente al regolamento sulla normazione europea (Regolamento (UE) n. 1025/2012). Peraltro, in assenza di tali norme armonizzate, è previsto che la Commissione possa adottare, mediante atti di esecuzione, specifiche comuni relative a ogni prescrizione di cui al paragrafo 1;
- **in materia di condivisione dei dati**, l'art. 30 si rivolge al "venditore di applicazioni che utilizzano smart contracts o, in sua assenza, [al]la persona la cui attività commerciale, imprenditoriale o professionale comporti l'implementazione di smart contracts per altri nel contesto di un accordo di messa a disposizione dei dati". Questi soggetti devono garantire che lo smart contract rispetti quattro caratteristiche chiave:
  **a) robustezza**: deve essere stato progettato in modo da offrire un grado di robustezza elevato per evitare errori funzionali e resistere alla manipolazione di terzi (art. 30, par. 1, lett a);

Dichiarazione firmata da 22 Paesi per la creazione dell'EBSI (Infrastruttura europea dei servizi blockchain) con lo scopo di garantire fornitura di servizi pubblici digitali transfrontalieri con i più elevati standard di sicurezza e privacy,.

[31] La Proposta di Regolamento di Normativa sui Dati definisce il titolare dei dati come la "[…] persona fisica o giuridica che ha il diritto, l'obbligo […] o, nel caso di dati non personali e attraverso il controllo della progettazione tecnica del prodotto e dei servizi correlati, la capacità di mettere a disposizione determinati dati" (art. 2, punto 6)).

[32] Gli articoli citati sono contenuti sia nel Capo II ("Condivisione dei dati da impresa a consumatore e da impresa a impresa") in cui attengono all'obbligo del titolare dei dati di mettere a disposizione di terzi i dati generati dall'uso di un prodotto o servizio correlato, su richiesta dell'utente (art. 5), ovvero all'obbligo del terzo che riceve i dati di utilizzarli solo per le finalità e alle condizioni concordate con l'utente (art. 6); sia nel Capo III, con riferimento alla giusta compensazione pattuita per la messa a disposizione dei dati tra titolare e destinatario (art. 9) e alla risoluzione delle controversie mediante organismi stragiudiziali (art. 10).

[33] Risoluzione del Parlamento europeo del 3 ottobre 2018 sulle tecnologie a registro distribuito e blockchain 'creare fiducia attraverso la disintermediazione' (2017/2772 (RSP)); Risoluzione del Parlamento europeo del 13 dicembre sulla blockchain 'una politica commerciale lungimirante' (2018/2085 (INI)); Risoluzione Parlamento europeo del 25 marzo 2021 su 'una strategia europea per i dati' (esorta la Commissione a presentare legge sui dati) (2020/2217/INI)).

[34] "Interoperabilità: la capacità di due o più spazi di dati o reti di comunicazione, sistemi, prodotti, applicazioni o componenti di scambiare e utilizzare dati per svolgere le loro funzioni" (art. 2, punto *19)*).

**b) cessazione e interruzione sicure**: deve prevedere un meccanismo per interrompere l'esecuzione continua delle transazioni[35]. In particolare "[lo smart contract deve] comprende[re] funzioni interne che possono reimpostarlo o trasmettergli l'istruzione di fermare o interrompere il proprio funzionamento per evitare esecuzioni (accidentali) future" (art. 30, par. 1, lett b);

**c) archiviazione e continuità dei dati**: nel caso in cui sia necessario procedere alla risoluzione o alla disattivazione di uno smart contract, è necessario "prevedere la possibilità di archiviare i dati relativi alle transazioni e la logica e il codice dello smart contract per tenere traccia delle operazioni effettuate sui dati in passato (verificabilità)" (art. 30, par. 1, lett c);

**d) controllo dell'accesso**: uno smart contract deve essere protetto mediante meccanismi rigorosi di controllo dell'accesso a livello della governance e dello smart contract stesso (art. 30, par. 1, lett d).

(iii) L'art. 30 prevede che i soggetti obbligati – venditore e/o imprenditore o professionista – effettuino una valutazione della conformità degli smart contract a questi requisiti e che rilascino una dichiarazione UE di conformità.
Venditore e/o imprenditore o professionista sono responsabili della corrispondenza della dichiarazione di conformità UE con le prescrizioni essenziali dell'art. 30, par.1.
La norma introduce, peraltro, una presunzione di conformità per gli smart contract che soddisfano le norme armonizzate adottate conformemente alle norme del regolamento sulla normazione europea che impongono prescrizioni analoghe a quelle dell'art. 30, par. 1 (art. 30, par. 4).

La potestà regolamentare in materia è ripartita tra la Commissione e le organizzazioni europee di normazione (CEN, CENELEC, ETSI). Queste ultime elaborano norme europee su richiesta della Commissione. Le norme "tengono conto dell'interesse pubblico e degli obiettivi politici chiaramente specificati nella richiesta della Commissione e sono fondati sul consenso. La Commissione stabilisce i requisiti relativi al contenuto che il documento deve rispettare e un termine per la sua adozione" (art. 30, par. 5). La Commissione interviene con propri atti di esecuzione contenenti specifiche comuni solo nel caso in cui manchino norme armonizzate o queste siano insufficienti (art. 30, par. 6).

### 3. La disciplina italiana sugli smart contract (de iure condito)

In Italia, il legislatore si è dimostrato attento al fenomeno ed è intervenuto tempestivamente rispetto ad altri Paesi, avendo disciplinato parzialmente, già nel 2018, gli smart contract.

Il "Decreto Semplificazioni" n. 135/2018, convertito con modificazioni dalla L. 11 febbraio 2019, n. 12 ha infatti consentito l'utilizzo delle tecnologie a registro distribuito e delle loro applicazioni – tra cui gli smart contract – e ha configurato un framework per una regolamentazione degli effetti giuridici relativi al loro utilizzo.

L'art. 8 ter[36], comma 2 D.L. 135/2018 stabilisce che:

[35] Questa è una prescrizione avversata da parte della comunità scientifica sul presupposto che un'interruzione del meccanismo dello smart contract imposto *via* regolatoria intaccherebbe un requisito indispensabile della tecnologia blockchain: "l'immutabilità". La proposta di Data Act (..) "would put smart contract immutability in check, thus challenging technology's survival. Immutability indeed differentiates smart contract from other contractual methods; it creates value" [26]

[36] Inoltre il legislatore, al comma 1, definisce le tecnologie a registro distribuito come tecnologie e protocolli informatici "che usano un registro condiviso, distribuito, replicabile, accessibile simultaneamente, architetturalmente decentralizzato su basi crittografiche, tali da consentire la registrazione, la convalida, l'aggiornamento e l'archiviazione di dati sia in chiaro che ulteriormente protetti da crittografia verificabili da ciascun partecipante, non alterabili e non modificabili".

> "Si definisce smart contract un programma per elaboratore che opera su tecnologie basate su registri distribuiti e la cui esecuzione vincola automaticamente due o più parti sulla base di effetti predefiniti dalle stesse."

Quanto agli effetti giuridici il comma 2 dell'art. 8 ter prevede che gli smart contract "soddisfano il requisito della forma scritta", sempre che vi sia stata la previa identificazione informatica delle parti interessate. La norma attribuisce all'Agenzia per l'Italia Digitale (AgID) il compito di definire – mediante linee guida - il processo relativo alla individuazione dei requisiti per effettuare l'identificazione informatica. Per quanto ad oggi non sia ancora in vigore un intervento strutturato sul piano applicativo, un primo passo in questa direzione è stato proposto dall'AgID con la stesura di "Linee Guida per la Modellazione delle Minacce ed Individuazione delle Azioni di Mitigazione Conformi ai Principi del Secure/Privacy By Design", e in particolare in riferimento alle "Best Practice di Secure Design per le Architetture Basate su Registri Distribuiti (DLT)". L'AgID propone un'analisi di alto livello rispetto ai requisiti di integrità, disponibilità e riservatezza di un sistema DLT, con particolare attenzione alle componenti infrastrutturali quali la rete, la struttura dati e gli algoritmi di consenso. Infine, le Linee guida AgID identificano con gli smart contract la componente maggiormente critica nell'ambito DLT; tuttavia si limitano a suggerire un approccio di secure-coding tradizionale nello sviluppo software, senza approfondire minacce e vulnerabilità specifiche degli smart contract. Infatti, nel recentissimo intervento del 1° giugno 2023 nell'ambito delle garanzie fideiussorie,[37] in attuazione di quanto disposto dall'art 26 del Codice dei contratti (D.lgs. n. 36/2023), l'AgID ha pubblicato il provvedimento che definisce requisiti tecnici e le modalità di certificazione delle piattaforme di approvvigionamento digitale.[38] In questo provvedimento l'AgID riconosce tra le condizioni che le piattaforme di fideiussione *devono* utilizzare la scrittura della garanzia fideiussoria per mezzo di uno smart contract. Anche in questa occasione l'AgID si limita a prevedere l'utilizzo di questo strumento con un'unica condizione che riguarda le caratteristiche che il soggetto deve possedere per rilasciare garanzie fideiussorie, non anche quelle che riguardano la tecnologia utilizzata.[39]

l'unica condizione che tale operazione sia possibile solo ad opera di un soggetto cui è consentito rilasciare garanzie fideiussorie ai sensi dell'articolo 106, comma 3 del Codice è autorizzato a scrivere nel registro distribuito, previa identificazione elettronica con un livello di garanzia significativo o elevato con riferimento al Regolamento eIDAS.

Il comma 3 dell'art. 8 ter prevede, inoltre, che - qualora i registri distribuiti siano conformi agli standard tecnici individuati dall'Agenzia per l'Italia Digitale - la memorizzazione di un documento informatico attraverso l'uso della tecnologia dei registri distribuiti "produce gli effetti giuridici della validazione temporale elettronica di cui all'articolo 41 del Regolamento (UE) n. 910/2014 del Parlamento europeo e del Consiglio, del 23 luglio 2014" in materia di identificazione elettronica e servizi fiduciari per le transazioni elettroniche nel mercato interno. Il rinvio al Regolamento comporta che ai dati registrati nella blockchain: a) viene attribuita una data certa e b) vengono riconosciuti gli effetti giuridici e l'ammissibilità come prova in giudizio.

L'art. 8 ter presenta qualche lacuna e ambiguità. La definizione di smart contract è generale, ma in parte anche generica [17]. Si riferisce all'esecuzione del programma, diversa dall'esecuzione

[37] Capitolo 6 "Piattaforme di gestione delle garanzie fideiussorie"

[38] Con determinazione n. 137, d'intesa con ANAC e Presidenza del Consiglio dei Ministri - Dipartimento per la Trasformazione Digitale

[39] Il par. 6.2-5.2 specifica che: "la scrittura della garanzia fideiussoria emessa nei registri distribuiti è effettuata per mezzo di uno smart contract che deve garantire che tale operazione sia possibile solo ad opera di un soggetto cui è consentito rilasciare garanzie fideiussorie ai sensi dell'articolo 106, comma 3 del Codice è autorizzato a scrivere nel registro distribuito, previa identificazione elettronica con un livello di garanzia significativo o elevato con riferimento al Regolamento eIDAS".

contrattuale, e presuppone quindi implicitamente una fase preventiva di formazione dell'accordo. La norma, inoltre, qualifica lo smart contract come vincolante le parti, inducendo molti a ritenere che la fonte giuridica del vincolo sia lo stesso smart contract.

Proprio queste criticità inducono a condurre l'indagine sulle caratteristiche degli smart contract in maniera olistica, considerando sia il dibattito scientifico nazionale e internazionale sia le scelte normative operate da legislatori di altri Paesi.

A questo proposito, particolarmente rilevante è il c.d. "Decreto Fintech"[40] con cui il legislatore italiano ha recepito il Regolamento (UE) 2022/858 (DLT Pilot Regime), che stabilisce un regime pilota per le infrastrutture di mercato basate sulla "Distributed Ledger Technology" e la semplificazione della sperimentazione Fintech. Le disposizioni del DLT Pilot principalmente[41] introducono la disciplina necessaria per l'emissione e la negoziazione di strumenti finanziari tokenizzati. La possibilità di tokenizzare diversi tipi di beni, prodotti o servizi e quindi di generare un token nel mondo virtuale e collegarlo a un bene reale tramite uno "smart contract" potrebbe avere un impatto significativo in termini di aumento della velocità e della sicurezza, ma anche di riduzione dei costi delle transazioni.

Nello specifico, il DLT Pilot prende in considerazione gli smart contract come uno degli elementi di cui può avvalersi l'infrastruttura di mercato DLT nello svolgimento delle attività, e la cui affidabilità deve essere garantita tanto quanto la continuità, la trasparenza, la disponibilità, l'affidabilità e la sicurezza dei servizi e delle attività che i gestori delle infrastrutture offrono attraverso dispositivi informatici e cibernetici relativi all'uso della loro tecnologia a registro distribuito[42] [27, 28, 29].

In questa prospettiva volta a "catturare" il fenomeno degli smart contract in tutta la loro complessità, nel lavoro si attribuisce rilievo centrale alla dottrina giuridica che ha affrontato il tema per definire lo stato dell'arte, cui si aggiunge una ricognizione sul piano normativo anche in chiave comparatistica. **Su questo secondo aspetto, per un approfondimento, si rinvia al *Focus* 2 – Analisi comparata di primo e di secondo livello.**

Nel lavoro abbiamo scelto, infine, di assumere una prospettiva interdisciplinare. Diritto e tecnologia richiedono una visione d'insieme. Si considereranno, dunque, anche aspetti tecnici confermati dalla migliore letteratura scientifica, nella misura in cui essi possano aiutare nell'analisi anche giuridica dei presupposti e delle conseguenze dell'utilizzo di smart contract.

Ciò premesso, è possibile operare una scelta metodologica proprio muovendo criticamente dalla norma italiana che, come è stato rilevato, non distingue in modo chiaro il piano dell'esecuzione dell'accordo contrattuale da quello dell'esecuzione del programma informatico. Pertanto, si procederà nell'analisi tenendo ferma la distinzione tra gli "smart contract" e gli "smart legal contract", per valutarne i vantaggi e i limiti sul piano giuridico.

[40] Decreto Legge 17 marzo 2023, convertito con Legge 10 maggio 2023, n. 52 recante "Disposizioni urgenti in materia di emissioni e circolazione di determinati strumenti finanziari in forma digitale e di semplificazione della sperimentazione Fintech".
[41] Più in generale, il DLT Pilot mira a consentire l'utilizzo di nuove tecnologie, in linea con le esigenze del mercato, e a rendere le infrastrutture di mercato DLT interoperabili con quelle del sistema finanziario tradizionale.
[42] Art. 7, par. 4 del DLT Pilot "Requisiti supplementari per le infrastrutture di mercato DLT".

**FOCUS 2 – Analisi comparata della normativa di primo e di secondo livello**

Nell'analisi comparativa della legislazione di diverse giurisdizioni in materia di Distributed Ledger Technologies e/o blockchain e delle relative applicazioni software comunemente denominate smart contract, l'attenzione è stata posta prevalentemente su giurisdizioni dei paesi membri dell'Unione europea.

Le giurisdizioni sono state selezionate secondo una metodologia di ricerca c.d. 'a palla di neve' o 'a valanga' (*snowball*), sulla base delle informazioni che sono state rinvenute on-line.

Il campionamento finale conta circa cinquanta giurisdizioni. In ragione di questo, i risultati della presente indagine non hanno pretesa di esaustività.

Per ogni giurisdizione, le aree tematiche DLT/blockchain e smart contract sono state analizzate separatamente secondo le seguenti variabili.

Quanto all'area DLT/blockchain:

- legislazione;
- definizione legislativa di DLT/blockchain;
- eventuali ulteriori condizioni;
- presenza di atti regolamentari (i.e. norme secondarie).

Quanto all'area smart contract:

- legislazione;
- definizione legislativa di smart contract;
- qualificazione degli smart contract come mero software;
- qualificazione degli smart contract anche come contratto;
- eventuale efficacia giuridica degli smart contract.

Ciascuna variabile è stata analizzata secondo un'impostazione binaria, verificando per ciascuna la presenza di un dato (Y) ovvero la sua assenza (N). Per ciascuna area tematica, le informazioni circa la presenza o l'assenza di legislazioni sono state considerate "preclusive" di tutte le altre. In altre parole, la ricerca circa le altre variabili si è presupposta sulla risultanza positiva relativa alla variabile "legislazione".

**Risultati dell'analisi**

L'analisi evidenzia l'assenza di legislazioni aventi a oggetto specificamente DLT/blockchain nella maggioranza delle giurisdizioni considerate. Le giurisdizioni per le quali si è verificata un'attività legislativa sono per lo più europee, con qualche sporadica eccezione (es: Israele dove è in corso di approvazione una normativa in materia di *digital asset*). Anche alcuni Stati degli Stati Uniti d'America hanno legiferato a tal riguardo (es: Wyoming).

Quanto agli smart contract, non si è riscontrata alcuna legislazione specifica nelle giurisdizioni considerate, fatta eccezione per la normativa del Wyoming. Quest 'ultima concerne i digital asset in generale e contiene una definizione di smart contract, come "an automated transaction, as

defined in W.S. 40-21-102(a)(ii), or any substantially similar analogue, which is comprised of code, script or programming language that executes the terms of an agreement, and which may include taking custody of and transferring an asset, or issuing executable instructions for these actions, based on the occurrence or nonoccurrence of specified conditions".

Per il resto, nel panorama degli ordinamenti considerati, solo Israele sembrerebbe in procinto di approvare una legge generale in materia di smart contract.

In Europa, nel Blockchain Act del Lichtenstein del gennaio 2020 compare una definizione di 'trustworthy technology', intesa come 'technologies through which the integrity of Tokens, the clear assignment of Tokens to TT Identifiers and the disposal over Tokens is ensured' e 'trustworthy technologies systems' definiti come 'transaction systems which allow for the secure transfer and storage of Tokens and the rendering of services based on this by means of trustworthy technology'. Senza entrare nella materia degli smart contract, il legislatore del Lichtenstein pone attenzione all'infrastruttura di mercato sottostante, stabilendo requisiti generali relativi alla: i) certezza dell'attribuzione dei token; ii) sicurezza del trasferimento dei token.

Dal quadro sopra esposto, può concludersi che il caso italiano resta in Europa, ad oggi, un *unicum* legislativo in materia.

La maggior parte degli ordinamenti europei (in particolare Francia, Germania, Lussemburgo, Polonia, Svizzera) ha predisposto una disciplina in materia di *securities token*, con orientamenti molto diversi relativi alla definizione della materia regolata e dunque all'individuazione di quali *securities token* far ricadere nella nuova normativa. La legge tedesca, ad esempio, disciplina unicamente i token rappresentativi di titoli al portatore, quella polacca solo azioni non quotate, quella francese tutti i titoli non soggetti a regime di dematerializzazione, mentre Lussemburgo e Svizzera sembrano aprire (pur per vie diverse data la soggezione della prima alla legislazione sovranazionale, in particolare la CSDR) alla tokenizzazione della maggior parte di *securities*. In questa prospettiva, gli ordinamenti citati istituiscono obblighi per i fornitori dei servizi DLT solo nella misura in cui questi servono per la circolazione delle securities rilevanti.

## Sezione II - Dottrina: temi e problemi

### 4. I diversi orientamenti sulla natura degli smart legal contract

Comprendere la natura giuridica dello smart legal contract e procedere alla sua qualificazione è ritenuto essenziale dalla dottrina sia per stabilire i casi d'uso sia per individuare la disciplina in concreto applicabile. Valutare dunque, se lo smart legal contract ha la sola funzione di dare esecuzione automatica a certe obbligazioni nascenti da contratto, o se possa essere qualificato come vero e proprio contratto non ha, evidentemente, rilievo solo teorico ma determina rilevanti conseguenze pratiche in relazione alle caratteristiche tecniche che devono connotare la tecnologia per assicurare il rispetto di requisiti normativi e per consentire la produzione degli effetti giuridici voluti dalle parti.

La qualificazione comporta la necessità di individuare i vincoli che regolano l'intera fase di vita del rapporto contrattuale, tenendo conto delle diverse regole operanti nei diversi settori e della diversa tipologia di clienti coinvolti.

Ciò premesso, tre sono le principali ricostruzioni proposte dalla dottrina che ha affrontato il tema dello "smart legal contract".

### 4.1 Smart legal contract come mero mezzo di adempimento di obbligazioni assunte altrove

Secondo questa impostazione, lo smart legal contract è un programma per elaboratore utilizzato solo per dare esecuzione, in tutto o in parte, a un contratto, alla stregua di un protocollo di transazione informatizzato che esegue ordini (in questo caso i termini di un contratto esterno alla blockchain) in modo automatico al verificarsi di condizioni predeterminate.

Uno smart legal contract non corrisponde, quindi, secondo questa tesi, a un vero e proprio contratto, ma al software (o protocollo informativo) sviluppato per l'esecuzione dello stesso [30]. In questo senso, uno smart legal contract è "programmato" per non avere bisogno di regole ulteriori rispetto a quelle incorporate nel codice ed è pertanto (auto)sufficiente [31, 32, 33, 34, 35]. I termini e le condizioni del contratto, concordati tra le parti, scritti sotto forma di codice e salvati nella blockchain, diventano così verificabili, immutabili e irrevocabili [36]. Quando i termini dell'accordo vengono soddisfatti, lo smart legal contract valuta i termini imposti dall'accordo (secondo una logica "if-then") ed eventualmente applica in modo automatico gli effetti previsti, come ad esempio approvare lo scambio di un token tra le parti.
L'automatizzazione dell'esecuzione dello smart legal contract garantisce l'adempimento. L'architettura blockchain, infatti, non consente violazioni volontarie delle condizioni stabilite. In questo modo il carattere più o meno vincolante dell'accordo deriva dalle code *layer* su cui lo smart legal contract è eseguito e non da una fonte esogena (normativa) [37]. Nell'applicazione degli smart legal contract, questa ricostruzione comporta un passaggio nella pratica contrattuale dal giudizio autoritativo *ex post* – tipico dei contratti tradizionali – alla valutazione automatizzata *ex ante* [38].

### 4.2 Smart legal contract come espressione tecnologica dell'accordo negoziale

Secondo questa impostazione [39, 40, 41, 42, 43, 44, 45] lo smart legal contract è un programma per elaboratore utilizzato anche per formare (in linguaggio informatico), in tutto [46] o in parte [47], il contenuto del contratto che viene poi eseguito automaticamente. Il programma, quindi, seppur non senza criticità nell'applicazione di alcuni istituti [48, 49, 50], è in grado di integrare alcune fasi contrattuali (l'accordo e l'esecuzione) come peraltro già ipotizzato in dottrina in relazione ai contratti standardizzati [51]; in definitiva, esprime il rapporto giuridico che vincola le parti così come tra queste concordato. Inoltre, svincolando dal fattore umano l'esecuzione delle prestazioni dedotte in contratto in ragione della 'notarizzazione' delle clausole sulla *chain*, lo smart contract automaticamente scongiurerebbe – o comunque attenuerebbe significativamente - il rischio di inadempimento di uno dei contraenti [17].

### 4.3 Lo smart legal contract come "contratto rafforzato"

Vi è in dottrina [52, 53, 54, 55] chi ritiene che lo smart legal contract abbia tutte le caratteristiche di un contratto, cui si aggiungono caratteristiche ulteriori, tipiche della blockchain, che ne rafforzano l'efficacia sul piano della negoziazione, dell'auto-esecuzione, della risoluzione delle controversie, del collegamento negoziale.
La tesi muove dal presupposto che il contratto, in linguaggio naturale, abbia solo talune clausole inserite su una blockchain.

Sulla base di questa ricostruzione, è possibile:

- notarizzare sulla blockchain la fase precontrattuale, certificando sia la fase delle trattative sia la fase di conclusione del contratto, con il vantaggio di prevenire quindi una parte significativa del contenzioso relativo a questa fase e all'interpretazione del contratto;

- consentire l'auto-esecuzione di alcune clausole che possono anche modificarsi nel tempo (es: i tassi di interesse). Più in generale, si può anche ipotizzare di rendere lo smart contract in parte flessibile così da poter, ad esempio, consentire a una banca l'esercizio dello *ius variandi* rispettando i presupposti di legge, oppure modificare i termini dell'accordo. Resta fermo che se lo smart contract è su blockchain, su questa resterà in ogni caso registrata anche la versione precedente dell'accordo. Questo consente peraltro di registrare e cristallizzare lo sviluppo temporale degli eventi;
- prevenire le controversie, da un lato registrando su blockchain le fasi negoziali, dall'altro prevedendo un oracolo, ovvero una terza parte fidata cui le parti demandano la soluzione della controversia;
- consentire il collegamento negoziale, subordinando la produzione degli effetti di una regola contrattuale o dell'intero contratto al verificarsi delle condizioni previste in altro contratto o documento (es: accesso a un contratto di investimento solo previa compilazione del questionario MIFID e conformità ai requisiti) [56]. La peculiare modalità con cui agisce lo smart legal contract può, più in generale, permettere una efficace gestione dei contratti collegati (si pensi ad esempio a un contratto di credito ai consumatori; oppure a un contratto di finanziamento collegato a polizza assicurativa), con un potenziale beneficio indiretto anche sotto il profilo della tutela della trasparenza della clientela.

Questa tesi va distinta dalle posizioni più radicali di chi ritiene che lo smart legal contract costituisca un fenomeno che, per il suo grado di autonomia, è destinato a sostituire con gli algoritmi (Rule of Code) le norme giuridiche fondate sul principio dello Stato di Diritto (Rule of Law) [57].

### 4.4 Limiti tecnici degli smart legal contract

La qualificazione in un senso o nell'altro non ha rilievo solo teorico ma incide anche sulla valutazione dello smart legal contract in termini di opportunità e limiti.

Provando a operare una sintesi sulle diverse letture della dottrina in questi termini, si può rilevare che diversi autori [58, 59, 60, 61, 62, 63, 64] si concentrano sui limiti tecnici degli smart legal contract in ragione del fatto che le garanzie tecnologiche di esecuzione dei contratti non sarebbero sufficienti per prevenire risultati ingiusti o non voluti [65]. Inoltre, pur ammettendo la capacità degli smart legal contract di creare fiducia in ambienti che ne sono privi, questa dottrina non riconosce loro un ruolo determinante per la risoluzione delle problematiche della contrattazione [66]. In generale, le criticità più diffusamente individuate sono:

- il linguaggio: gli smart legal contract sarebbero intrinsecamente limitati. Il computer, applicando la logica *booleana* e il linguaggio di codifica formale, cerca distinzioni categoriali nette nei fenomeni, diversamente dal mondo reale, molto più indeterminato e ambiguo [13, 20, 67, 68, 69]. L'esigenza di predeterminare un orizzonte semantico chiaro e ben definito (che è un vantaggio sotto il profilo della trasparenza sul contenuto dell'accordo e comporta un maggior grado di certezza sulle future prestazioni da compiersi) determinerebbe tuttavia una mancanza di flessibilità rispetto a quanto è possibile definire in via negoziale nella realtà;
- la capacità di giudizio: alcune decisioni richiedono una valutazione di molteplici fattori, come gli standard industriali, il *framework* normativo e il rapporto commerciale tra le parti. Non è semplice esprimere in un codice, standard e principi come, ad esempio, la "buona fede" [57];

- la limitazione della capacità di autodeterminazione: le parti contraenti non hanno la possibilità di compiere scelte razionali successivamente alla firma del contratto, come, ad esempio, valutare l'opportunità di risarcire i danni invece di procedere con l'esecuzione forzata. Vengono meno le opportunità di "'violazione efficiente"' [15];
- la vulnerabilità agli attacchi, il rischio di frodi e violazione della privacy, legati a problemi di cybersicurezza. Sul punto, il Parlamento europeo [23] propone specifiche tecniche per prevenire tali rischi operativi da parte dei gestori;
- non sempre l'adempimento delle obbligazioni è il fine ultimo di un contratto: non tutti gli accordi sono concepiti in funzione di una precisa esecuzione. Molte clausole aperte sono intenzionalmente redatte per garantire alle parti un'ampia discrezionalità, per favorire rapporti a lungo termine nei contratti di durata e consentire alle parti di rispondere in modo flessibile a circostanze impreviste senza la necessità di riformulare l'accordo [70, 71];
- la capacità dello smart contract nella rappresentazione della realtà è limitata. Si tratta del c.d. Digital Twin Problem, secondo cui la creazione tramite smart contract di una copia digitale di un bene esistente (tokenizzazione dell'asset) non sempre è autosufficiente rispetto alla realtà esterna. I token emessi cioè esistono sulla chain ("*digital twin*"), mentre i beni reali continuano ad esistere nel mondo "off-chain", determinando una difficile "convivenza". Diversa è l'ipotesi dei beni nativi digitali che non esistono fuori dalla chain: in questo secondo caso, lo smart contract consente l'emissione di token 'nativi' della blockchain, costruiti direttamente on-chain e che vivono esclusivamente sul libro mastro distribuito[43].

### 4.5 Smart legal contract e inclusione finanziaria

Una parte della dottrina osserva il fenomeno dello smart legal contract nella prospettiva di idoneità dello strumento alla realizzazione anche di obiettivi di giustizia sociale ed equità [72].
Secondo alcuni autori [70], gli smart legal contract non sono progettati per tenere conto delle complessità sociali implicate nella contrattualistica tradizionale, tanto da risultare inadeguati per gli usi sociali. Gli sviluppatori pongono attenzione alla forma tecnica dello smart contract e trascurano sia i contesti sociali in cui i contratti operano sia i modi in cui i consociati li utilizzano. Questa osservazione critica è particolarmente diffusa con riferimento al settore finanziario [73]. In questo ambito, il previo accertamento della capacità finanziaria dell'investitore - intesa come insieme delle competenze che consentono di comprendere i mercati finanziari e i rischi che derivano - è fondamentale per la tutela dei clienti finanziariamente meno informati o più fragili. In assenza degli strumenti di diritto contrattuale tradizionale, l'applicazione di questi contratti potrebbe generare risultati indesiderati per gli investitori [74]. A ben vedere questo rilievo critico può essere superato se si adotta uno smart contract in linguaggio naturale e se si accompagna l'uso di smart legal contract in campo finanziario con adeguati presidi informativi e di educazione finanziaria (**vedi infra Focus 4-Trasparenza).**
Inoltre, una sensibile riduzione dei tempi di lavorazione e dei costi necessari per l'elaborazione delle transazioni, considerate "*the essence of economic activity*", potrebbe concorrere a migliorare l'inclusione finanziaria [71].

## 5. Problemi giuridici posti in dottrina

Le differenze sul piano giuridico tra gli accordi elettronici tradizionali e gli smart legal contract su blockchain derivano dal modo in cui i contratti vengono eseguiti e fatti rispettare.

[43] Per un approfondimento si veda Consob, Tokenizzazione di azioni e azioni tokens, Quaderni giuridici, gennaio 2023; si veda anche OCSE, Crypto-Asset Reporting Framework and Amendments to the Common Reporting Standard, 2022. [75, 76].

### 5.1 Con riferimento allo smart contract code

Nell'analisi degli smart contract intesi come codice, sorgono questioni "traduttologiche" [13] implicate nel processo di creazione dei programmi per elaboratore. Le principali criticità si concentrano nel passaggio dal linguaggio naturale ad altri due linguaggi comprensibili per le macchine: il linguaggio di programmazione e il linguaggio macchina (solitamente, questi sono anche distinti come linguaggi rispettivamente di alto e di basso livello).

Per "linguaggio di programmazione" si intende il linguaggio che si esprime attraverso parole, numeri, simboli di puntuazione e altri simboli grafici[44] e che si estrinseca in una pluralità (centinaia) di lingue e perfino di "dialetti" variamente diversificati in termini sintattici e semantici. Questo linguaggio è preordinato all'elaborazione di istruzioni da tradurre nel linguaggio macchina che serve a trasmetterle agli elaboratori per la loro esecuzione. Per "linguaggio-macchina" si intende, invece, il linguaggio composto di *bit* convenzionalmente rappresentati con i numeri 0 e 1 (c.d. alfabeto o codice binario di *bit*). Anch'esso si manifesta in una pluralità di "lingue" o codici o linguaggi, diversificati in funzione delle caratteristiche degli elaboratori (architettura *hardware*)[45].

Per trasferire all'elaboratore le istruzioni originariamente concepite e poi espresse in linguaggio naturale è necessario tradurle in una 'lingua' del linguaggio di programmazione (prima traduzione) e successivamente dalla 'lingua' del linguaggio di programmazione a una 'lingua' del linguaggio macchina (seconda traduzione). Della prima fase si occupano i programmatori, mentre per la seconda agisce automaticamente l'elaboratore attraverso appositi programmi progettati con questo scopo (c.d. compilatori)[46].

Si pone dunque un problema di adattabilità, derivante dal bisogno di trasporre la semantica contrattuale in chiave algoritmica: il linguaggio umano viene "convertito" in codice di programmazione, che sostituisce la normale comprensibilità, flessibilità e duttilità del linguaggio naturale con la rigidità dialettica binaria 0 e 1 [17, 77].

### 5.2 Con riferimento agli smart legal contract

I giuristi considerano lo smart contract uno strumento innovativo per articolare, verificare e applicare un accordo tra le parti. Tuttavia, in mancanza di una definizione chiara e univoca del fenomeno, gli studiosi hanno addotto argomentazioni logico-giuridiche varie per legittimare l'utilizzo degli smart contract nella pratica degli scambi tradizionali.

Secondo parte della dottrina sarebbe superfluo indagare la disciplina applicabile agli smart legal contract, in ragione della loro capacità intrinseca di sopravvivere al di fuori di qualsiasi ordinamento: costituirebbero una vera e propria alternativa al diritto dei contratti [77]. Ad esempio, in virtù del loro collegamento funzionale con i sistemi a registro distribuito, la previsione di ipotesi problematiche di mancata esecuzione dell'accordo non avrebbe alcuna rilevanza sul piano applicativo. Inoltre, se anche si riscontrassero dolo o violenza nella formazione del contratto, o ipotesi di invalidità dello stesso, sarebbe comunque impossibile modificare *ex post* il database della blockchain. Sulla base di tale ricostruzione, eventuali azioni di risarcimento e/o restituzioni: a) sarebbero improbabili, data la difficoltà di individuare le parti, e b) non potrebbero mai incidere sul funzionamento della blockchain.

[44] Il linguaggio di programmazione si distingue in linguaggio di programmazione c.d. di alto livello, che è normalmente il primo linguaggio utilizzato dai programmatori, e linguaggio di programmazione c.d. di basso livello chiamato assembly, la cui sintassi e semantica si avvicinano di più al linguaggio macchina.

[45] Il programma scritto in linguaggio di programmazione viene definito 'codice sorgente', mentre quello scritto in linguaggio macchina, che viene eseguito dal computer, viene chiamato 'codice macchina' (o anche 'codice oggetto').

[46] Nei fatti, e nella quasi totalità dei casi, le traduzioni sono tre, perché vengono utilizzati due livelli di linguaggio di programmazione prima di passare alla traduzione nel linguaggio macchina.

Altra dottrina considera essenziale e al contempo sufficiente l'accordo tra le parti perché possa ritenersi concluso un contratto valido. Questa ricostruzione deriva, da un lato, dall'analisi comparata che individua il minimo comune denominatore del contratto nell'accordo, dall'altro (quanto al diritto italiano) dalla lettura dell'art. 1321 c.c. che definisce il contratto come "l'accordo di due o più parti per costituire regolare o estinguere tra loro un rapporto giuridico patrimoniale". In quest'ottica, gli elementi individuati nell'art. 1325 c.c.[47] devono considerarsi ulteriori rispetto a quello essenziale: l'accordo [13]. In altre parole, lo smart legal contract potrà essere concettualmente assimilato al contratto tradizionale se sarà idoneo a realizzare gli effetti voluti dalle parti.

All'interno di questa impostazione, c'è chi ha valorizzato l'accordo non solo dal punto di vista contenutistico – con specifico riguardo alle pattuizioni oggetto del contratto – ma anche funzionale. In questa prospettiva, l'azione materiale di "dare avvio" al programma sarà la prova della manifestazione della volontà di una parte di accettare le istruzioni in esso contenuto; sarà dunque l'avvio congiunto del programma, ad opera delle parti interessate, a documentarne l'accordo [77].

Secondo altri [30, 78] invece, lo smart legal contract designa in maniera più generica una struttura contrattuale, la cui natura risiede non nei contenuti, ma in una particolare architettura del contratto. Ciò che legittima questa tecnica di costruzione del contratto è l'accordo di chi la programma, la sceglie e decide di servirsene: "è l'accordo sulla tecnica (cioè sulla struttura peculiare che si vuole conferire al contratto) che salva la natura negoziale del "prodotto" della tecnica stessa. In definitiva, un "contratto sul contratto" [79] e "la valida conclusione del rapporto, pertanto, si determina in base alla teoria dell'affidamento sullo strumento informatico [80]

Di conseguenza, tali autori riconoscono allo smart legal contract la funzione di produrre effetti giuridici apprezzabili per l'ordinamento in ragione della sua capacità di manifestare la volontà delle parti, fosse anche soltanto sulla scelta dell'architettura contrattuale di cui servirsi per vincolarsi all'adempimento di determinate prestazioni. In questa prospettiva, l'utilizzo dell'espressione del legislatore italiano "effetti predefiniti dalle parti" all'art 8 ter del Decreto Semplificazioni, non dimostra l'assenza dell'accordo, ma – al contrario – l'esistenza di una volontà delle parti tesa all'esecuzione (automatizzata) di un accordo negoziale, stabilendone una sorta di realità che non si configura mediante la *traditio*, bensì attraverso l'esecuzione dei termini contrattuali pattuiti.

Un'ulteriore parte della dottrina [48], invece, muove dall'assunto per il quale il mondo digitale non può ritenersi avulso da quello reale e occorre, dunque, verificare quali criticità possono sorgere nell'applicazione delle regole pensate per gli scambi tradizionali al nuovo fenomeno degli smart legal contract intesi come contratti tra due o più parti. In altre parole, affinché uno smart legal contract produca effetti giuridici rilevanti per l'ordinamento, e dunque vincolanti per le parti, è necessario che gli elementi essenziali di questo nuovo strumento contrattuale e la relativa disciplina applicabile siano compatibili, per quanto possibile, con il quadro normativo civilistico che regola i contratti tradizionali [20, 36, 81, 82, 83]. A questo proposito, le questioni maggiormente interessate dal dibattito dottrinale riguardano:

1. il **riconoscimento dei soggetti** coinvolti nell'accordo negoziale: la questione si pone con riferimento alla necessità di verificare, in primo luogo, la capacità giuridica e di agire dei contraenti, come richiesto dall'art. 2 c.c.; in secondo luogo, di identificare con certezza la parte che si vuole convenire in giudizio in caso di controversia[48];

[47] Art. 1325 c.c.: "I requisiti del contratto sono: 1) l'accordo delle parti; 2) la causa; 3) l'oggetto; 4) la forma, quando risulta che è prescritta dalla legge sotto pena di nullità."

[48] Si tratta di un tema già affrontato dalla dottrina con la diffusione dei contratti conclusi per via telematica, per la cui sicurezza e validità si richiedevano tecniche di identificazione elettronica. Tuttavia, le soluzioni in quella sede individuate, basate su sistemi centralizzati, non sono compatibili con i sistemi DLT, il cui valore aggiunto è rappresentato proprio dalla decentralizzazione.

2. la **conclusione del contratto**, rilevante ai fini dell'applicazione dell'art. 1326 c.c. ("Conclusione del contratto"), sulla base del quale il contratto può dirsi concluso nel momento in cui "chi ha fatto la proposta ha conoscenza dell'accettazione dell'altra parte". La complessità qui deriva dalla natura della proposta e dell'accettazione: in quanto dichiarazioni di volontà – espressa nello smart contract tramite linguaggio di programmazione – è necessario che l'intento sia percepibile e comprensibile da tutti i contraenti. La questione è trattata in maniera diversa con riferimento a un accordo negoziato oppure a un contratto di adesione[49].
3. la **forma del contratto**, rilevante nei casi in cui la legge richiede sotto pena di nullità la forma scritta ai sensi dell'art. 1350 c.c. o comunque quando si richiede la forma "di protezione", ad esempio, nell'ipotesi di contratti di investimento o bancari: qui il problema non attiene all'idoneità dello smart legal contract a garantire la provenienza delle dichiarazioni, quanto all'eventuale necessità di ricostruire la volontà delle parti *ex post*, alla conoscibilità e opponibilità ai terzi, o ancora alle ipotesi in cui la forma è stabilita per tutelare il contraente debole, al fine di garantire una corretta e adeguata informazione per assicurare la consapevolezza di questo [84, 85];
4. l'**applicazione di principi generali dell'ordinamento** come il dovere di dare esecuzione al contratto secondo buona fede ai sensi art. 1375 c.c. ("esecuzione di buona fede"), il cui rispetto può essere oggetto di valutazione successivamente alla conclusione dell'accordo, poiché legato al comportamento tenuto dalle parti nel corso del suo svolgimento;
5. la **nullità delle clausole** inserite nel contratto per contrarietà a norma imperativa oppure per l'ipotesi in cui, pur in presenza di una clausola valida, il codice dello smart legal contract ne determini in concreto un'applicazione contraria alla legge;
6. le **c.d. sopravvenienze**, che assumono rilevanza fondamentale nei rapporti di durata: la dottrina a questo proposito si interroga sulla capacità dello smart legal contract di garantire certezza nell'esecuzione della prestazione dedotta in contratto anche al verificarsi di accadimenti originariamente non previsti dalle parti;
7. in un'ottica di più ampio respiro, la riflessione sulla realizzazione del **concetto di giustizia sostanziale** che si esplica nel dovere inderogabile di solidarietà ai sensi dell'art. 2 della Costituzione che, proprio con riferimento al diritto dei contratti, (i) ne integra il contenuto o gli effetti (art. 1174 cc); (ii) ne orienta l'interpretazione (art. 1366 cc); (iii) ne orienta l'esecuzione (art. 1375 cc);
8. la **soluzione delle controversie**, che si riducono in modo significativo in ragione della caratteristica dello smart contract code utilizzato e che possono essere risolte all'interno dello smart legal contract mediante un insieme di strumenti (riconducibili alla nozione e alla funzione di 'oracolo') più o meno automatici in funzione dello smart contract code utilizzato. Alcuni di questi profili sono peraltro connessi al tema della trasparenza bancaria, intesa come comprensibilità del contenuto e degli effetti del contratto, anche nella prospettiva della valutazione della correttezza dell'intermediario, che si lega anche all'individuazione dei rimedi riconosciuti al cliente e/o consumatore per la violazione dei loro diritti nell'esecuzione del contratto. **Su questo aspetto si veda Focus 4 - Trasparenza**

Più in generale, alcuni autori [15] fanno notare come l'utilizzo del codice determini incertezze in ordine alla salvaguardia dell'integrità del volere delle parti [72] nonché all'intelligibilità dell'accordo da parte dei contraenti e, conseguentemente, alla validità del consenso da essi prestato, specie se sono sprovvisti di particolari competenze in campo informatico e/o appartengono a categorie tutelate.

[49] Con riferimento specifico alle modalità di conclusione del contratto, c'è chi sostiene che uno smart contract non sia concettualmente diverso da un annuncio pubblicitario [86]

Questo processo di adattamento è affidato al giudizio del programmatore che, ove non coadiuvato dal giurista, può produrre un risultato negoziale che non corrisponde alla volontà delle parti contraenti come da loro intesa ed espressa, nonché può dare luogo a esiti opinabili oppure distorti[50].

Inoltre, nell'ambito di mercati regolati (e in questo studio sono stati valutati principalmente quelli dei settori bancario, assicurativo, dei mercati finanziari), il rapporto tra il programmatore e il giurista può non essere sufficiente. Occorre infatti in molti casi prevedere la partecipazione delle autorità di regolazione e/o supervisione, sin dall'origine, per collaborare alla verifica della effettiva presenza delle tutele minime previste dalle norme. Il rapporto tra questi soggetti è definito in dottrina 'regolazione partecipata' [52, 53, 54, 55].

**FOCUS 3 – La regolazione partecipata**

Nella DLT e nella blockchain le regole sono incorporate nella tecnologia. Lo smart contract può orientare flussi economici, comprimere diritti fondamentali, ecc. È quindi impossibile per l'autorità di vigilanza o di regolazione intervenire ex post, imponendo non solo obblighi di trasparenza e non discriminazione, ma anche requisiti tecnici che garantiscano un livello minimo di diritti e tutele previsti dall'ordinamento. Occorre pertanto che le autorità intervengano sin dall'inizio, cooperando con gli operatori, consentendo lo sviluppo della tecnologia secondo condizioni e requisiti minimi condivisi, conformi e coerenti con il welfare che corrisponde alla tradizione giuridica che s'intende difendere.

Questa procedura regolatoria viene definita in dottrina 'regolazione partecipata' [52, 53, 54, 55] per differenziarla dalle altre forme di collaborazione tra autorità e mercato, che si estrinsecano ad esempio nelle procedure di consultazione sui provvedimenti delle autorità, o nella definizione degli impegni comportamentali o strutturali delle imprese in posizione dominante (diritto della concorrenza) o detentrici di una posizione di mercato significativa (regolazione delle comunicazioni elettroniche).

### 5.3 Rischi da traduzione tra smart code e smart legal code

Riprendendo il tema del linguaggio a valle di questa analisi, i 'rischi da traduzione' sono maggiori quando il testo originario del linguaggio naturale da tradurre in linguaggio di programmazione è tecnicamente connotato [87]. In questo caso, al programmatore viene richiesto di comprendere il linguaggio tecnico impiegato nel testo di lingua naturale che deve tradurre, o di avere o fruire dei mezzi di un'organizzazione che abbia sufficienti risorse per assumere, gestire e assimilare una consulenza che sia a sua volta qualificata in modo sufficiente a colmare la lacuna di conoscenza del programmatore per i fini dell'interpretazione del testo e della sua traduzione nella lingua di programmazione prescelta. Questo aspetto rappresenta anche il profilo maggiormente critico in prospettiva di tutela della trasparenza.

Secondo altri [88, 89] invece, il linguaggio informatico ha il potere di eliminare la vaghezza e ambiguità tipiche del linguaggio umano. Il codice strutturato secondo lo schema "se X allora Y" non conosce

[50] E' possibile infatti che si verifichi: a) un'ingiustificata esclusione dal programma per elaboratore di alcune istruzioni implicitamente o esplicitamente contenute nel testo da tradurre (in quanto, ad esempio, alcune istruzioni non sono riconosciute come tali dal programmatore o sono da questi ritenute irriducibili alla logica del se → allora, oppure in quanto il programmatore, pur riconoscendo che una parte di testo contiene un'istruzione riducibile alla logica del se → allora, ritenga nondimeno che uno o più degli elementi di tale istruzione sia troppo ambiguo e non sia di conseguenza possibile procedere con certezza alla sua individuazione), o b) un'inesatta traduzione in linguaggio di programmazione per un colpevole fraintendimento o per una volontaria deviazione dal significato testualmente riconoscibile o comunque conosciuto dal programmatore e riconoscibile sulla base di elementi extra testuali.

la polisemia, ma si caratterizza per un'unicità finalizzata a realizzare un messaggio per la macchina. Le variabili e le condizioni possiedono un solo e unico significato, e consegnano alle parti un contratto privo di equivocità e ambivalenza, sia interpretative sia esecutive. Lo smart contract assicura pertanto certezza: (i) nel suo significato semantico intrinseco; (ii) nel suo significato formale estrinseco; (iii) nell'esecuzione delle prestazioni in esso dedotte. Lo smart contract viene così sottratto a possibili letture discordanti del suo contenuto. Diversamente, si potrebbe verificare una "endemica situazione di precarietà", posto che ciascuna delle parti potrebbe, nei fatti, esimersi dall'adempiere fin tanto che l'esecuzione del contratto non le venga imposta dalle autorità, laddove e quando ciò effettivamente accada [89, 90].

**FOCUS 4 – La trasparenza**

Indipendentemente dalla qualificazione giuridica dello *smart legal contract* che si assume come corretta, quest'ultimo, se utilizzato per operazioni e servizi bancari e finanziari, dovrà assicurare anche il rispetto delle disposizioni in materia di trasparenza delle condizioni contrattuali e dei rapporti con i clienti, previste, quanto al settore specifico, - che qui si assume come ipotesi paradigmatica anche se certamente non esaustiva - nel Titolo VI del Testo unico delle leggi in materia bancaria e creditizia (T.U.B, oltre che in normative ad hoc, come ad esempio quelle in tema di diritti e obblighi delle parti nei servizi di pagamento ex d.lgs. 11/2010) e nelle Disposizioni in materia di trasparenza delle operazioni e dei servizi bancari e finanziari"[51], che disciplinano compiutamente le relazioni tra intermediari e clienti. Occorre quindi che lo *smart legal contract* sia formulato in modo coerente con il sistema normativo vigente anche quanto ai profili di tutela della trasparenza, ambito regolato da un corpus trasversale che interessa diversi settori del nostro ordinamento (ad es. Banca d'Italia, ma anche Consob e IVASS, per restare nell'ambito finanziario e assicurativo).

Trasparenza e informazione sono strumenti imprescindibili di tutela della clientela, specie *retail*, che deve essere messa in condizione di operare scelte contrattuali consapevoli e funzionali alle proprie esigenze. In ogni fase del rapporto contrattuale (pubblicità, precontrattuale, contrattuale, in corso di rapporto e in occasione della sua cessazione) la normativa, anche di matrice europea, detta regole di condotta per gli intermediari, improntate al rispetto di prescrizioni puntuali oltreché a un dovere generale di correttezza, che fanno perno, in estrema sintesi su: (i) obblighi di assistenza e informativi per consentire al cliente di comprendere le caratteristiche e i costi del servizio e/o del prodotto, confrontarli con facilità con quelli offerti sul mercato, adottare decisioni ponderate; (ii) standardizzazione di alcuni documenti informativi; (iii) requisiti di forma e contenuto dei contratti; (iv) rimedi per l'inosservanza delle regole. Il regime previsto è poi "a geometria variabile", in considerazione del principio di proporzionalità: la disciplina, infatti, si articola secondo modalità differenziate in relazione alle esigenze delle diverse fasce di clientela e alle caratteristiche dei servizi.

Il ricorso allo strumento dello *smart legal contract* per prestare servizi finanziari o per concludere contratti bancari non sembra in astratto pregiudicare gli obiettivi di trasparenza potendo anzi, ove in grado di rispettare i requisiti normativi e gestiti da clienti consapevoli, per alcuni profili, favorire relazioni chiare, certe e trasparenti. La tecnologia blockchain consente al mercato l'utilizzo di strumenti nuovi per offrire in modo efficiente servizi finanziari e bancari, in conformità agli obiettivi di tutela posti dalla normativa. Si tratta tuttavia di un'ipotesi che dovrà essere verificata, tenuto conto che ad oggi non esiste ancora una sperimentazione tale da garantire la tenuta del meccanismo rispetto alle regole vigenti.

Sul tema, assume rilievo il fatto che le autorità di vigilanza devono poter verificare nel concreto che non ci siano pregiudizi per i clienti e possibili lacune sul fronte del rispetto delle regole di trasparenza; a ciò va aggiunto che, accanto al rispetto tassativo delle regole, le autorità devono poter accertare anche la correttezza delle condotte, che non dipende solamente dalla modalità con cui le regole stesse sono state tradotte in codice per la creazione dello smart legal contract, ma anche dalla rappresentazione delle condizioni contrattuali al cliente.

Come in precedenza accennato, il legislatore – che ha riconosciuto il valore giuridico degli *smart contract* con il Decreto semplificazioni n. 135/2018, convertito con modificazioni dalla L. 11 febbraio 2019, n. 12 – ha espressamente attribuito agli smart contract la stessa efficacia probatoria della forma scritta purché vi sia stata l'identificazione informatica delle parti interessate.

L'art. 8 ter, comma 2, stabilisce che «Gli smart contract soddisfano il requisito della forma scritta previa identificazione informatica delle parti interessate, attraverso un processo avente i requisiti

fissati dall'Agenzia per l'Italia digitale con linee guida da adottare entro novanta giorni dalla data di entrata in vigore della legge di conversione del presente decreto».

La norma non rinvia in modo espresso al Regolamento (UE) n. 919/2014, noto come eIDAS e al d.lgs. 82/2005, istitutivo del Codice dell'Amministrazione Digitale (CAD) che, all'art. 20, comma 1-bis nel disciplinare il documento informatico prevede che questo "soddisfa il requisito della forma scritta e ha l'efficacia prevista dall'articolo 2702 del Codice civile quando vi è apposta una firma digitale, altro tipo di firma elettronica qualificata o una firma elettronica avanzata o, comunque, è formato, previa identificazione informatica del suo autore, attraverso un processo avente i requisiti fissati dall'AgID ai sensi dell'art. 71 del codice stesso, con modalità tali da garantire la sicurezza, integrità ed immodificabilità del documento, nonché la sua riconducibilità all'autore. In tutti gli altri casi, l'idoneità del documento informatico a soddisfare il requisito della forma scritta e il suo valore probatorio sono liberamente valutabili in giudizio, in relazione alle caratteristiche di sicurezza, integrità e immodificabilità".

Questa regola, dettata per il documento informatico, può essere utile per un'interpretazione anche dell'art. 8 *ter* in chiave funzionale agli interessi protetti. Ai fini della valenza probatoria prevista dall'art. 2702 c.c. rilevano le caratteristiche di sicurezza, integrità e immodificabilità. Queste prerogative possono essere garantite dall'uso di smart legal contract su blockchain.

Sulla base di questa considerazione, anche nel settore bancario si può ritenere che lo smart contract possa soddisfare il requisito di forma dettato dall'art. 117, comma 1, del T.U.B., secondo cui "i contratti sono redatti per iscritto e un esemplare è consegnato ai clienti", a pena di nullità[52]. Anche in questa prospettiva, l'immutabilità e tracciabilità delle azioni a rilevanza contrattuale consentono di stabilire con certezza quale sia il momento di perfezionamento dell'accordo e di riconoscere allo smart contract la stessa efficacia di un contratto redatto in forma scritta. L'impianto che contraddistingue il regime di trasparenza prevede, inoltre, un regime di nullità formali che riguardano il contenuto necessario del contratto, le condizioni contrattuali e, più in generale, la completezza e coerenza delle clausole contrattuali, anche in relazione alle informazioni pubblicizzate e fornite in sede precontrattuale (ad esempio quelle che prevedono oneri non pubblicizzati). Lo smart contract deve quindi essere allineato ai requisiti previsti dalle norme o dagli stessi operatori, incluso dare la possibilità al cliente di esercitare tutti i diritti di legge, specie ai fini del recesso dal rapporto negoziale. Resta fermo che queste potenzialità dello smart contract sono allo stato da verificare in fase di sperimentazione e presuppongono che sia risolto il problema del linguaggio.

Più in generale, l'impatto, auspicabilmente positivo, sul grado di trasparenza dell'uso di questa tecnologia dipenderebbe dalle caratteristiche dello *smart legal contract* di tracciabilità e immutabilità, che potrebbe garantire alla relazione tra intermediari e clienti maggiore certezza, a beneficio delle parti stesse del rapporto e, in prospettiva, delle Autorità di vigilanza nello svolgimento dei compiti di supervisione. Se ben strutturato, lo *smart legal contract* potrebbe anche contribuire a gestire i reclami e le controversie che possano insorgere tra le parti.

Assume rilievo centrale, ancora una volta, il problema del linguaggio e della traduzione, da più parti sottolineato come l'aspetto di maggiore complessità, e che si riflette anche sul modo in cui l'Autorità può svolgere i compiti di supervisione cui è tenuta. In prospettiva sembra si possa

[51] Ci si riferisce al provvedimento 18 giugno 2019 di Banca d'Italia, recante "Disposizioni in materia di trasparenza delle operazioni e dei servizi bancari e finanziari" [91] che verte sulla correttezza delle relazioni tra intermediari e clienti. L'intervento dà attuazione alla direttiva 2014/92/UE (Payment Account Directive, o PAD) e al capo II-ter, titolo VI, del Testo Unico Bancario.

[52] Per completezza, il comma 2 rimette al CICR la facoltà di prevedere che, per motivate ragioni tecniche, particolari contratti possano essere stipulati in altra forma.

superare queste criticità mediante la possibilità (da verificare tecnicamente) di rendere lo smart contract leggibile in linguaggio naturale.

Un aspetto delicato, che rileva ai fini di una più ampia riflessione sulla coerenza degli *smart legal contract* con la disciplina di tutela, è quello dell'accezione di trasparenza in senso sostanziale, come *comprensibilità* del contenuto e degli effetti del contratto, a garanzia soprattutto del consumatore. Il tema, per sua natura complesso, non sembra tuttavia porsi in termini troppo diversi per il contratto *on chain* e per quello tradizionale *off chain.*

Occorre, peraltro, che l'applicativo consenta di garantire al cliente, ma anche alle citate Autorità, non solo la comprensione del contratto espresso in linguaggio naturale, ma anche che la traduzione in linguaggio informatico rifletta puntualmente quanto espresso in linguaggio naturale.

O meglio, il problema della scarsa comprensibilità si pone se nello smart contract il contratto è "tradotto" in linguaggio informatico non leggibile né comprensibile, se non dai crittografi. A questo riguardo tuttavia parte della dottrina ha obiettato che lo smart contract è sintesi dell'accordo e che le regole sono "preimpostate" dalle parti e dunque "precomprese" prima di essere codificate. Questa conclusione muove però da alcuni presupposti, non necessariamente veri. Il primo è fare riferimento allo smart legal contract come strumento che non consente anche la negoziazione on chain. Il secondo è presupporre che il linguaggio informatico limiti (anche se non impedisca) la comprensibilità dell'accordo una volta caricato sulla chain. Una diversa via, astrattamente percorribile, è invece quella di articolare il processo di creazione dello smart legal contract prevedendo che, alla fine, esso sia, in qualche modo, leggibile in linguaggio naturale, o ne sia assicurata la coerenza con il testo in linguaggio naturale, in modo da assicurare alla clientela chiarezza e comprensibilità dei termini dell'accordo.

In ogni caso deve essere garantito alla parte contraente il diritto alla conformità della versione in linguaggio informatico con quella in linguaggio naturale, e dovrebbe essere fornita dall'intermediario un'informazione adeguata sulle modalità di esercizio di questo diritto e sulle caratteristiche del servizio o prodotto offerto. Più in generale, gli intermediari che vorranno ricorrere allo strumento dello smart legal contract dovranno agire sul piano della corretta individuazione delle fasce di clientela aperte a un utilizzo di questo strumento (anche in modo da valutare una certa gradualità nella sua diffusione) e dell'informazione adeguata. Si pone cioè un tema di valutazione della correttezza dell'intermediario ai fini della tutela della trasparenza.

Inoltre, sempre ai fini della garanzia di una trasparenza sostanziale oltre che formale, si può ipotizzare il ricorso a sistemi di automazione (**per il cui approfondimento si rinvia al Focus 1 – Smart contract e la proposta di Regolamento sull'Intelligenza Artificiale**) nella esecuzione dello smart contract, ad esempio, prevedendo, in fase di conclusione dell'accordo, la sottoposizione alla clientela di questionari finalizzati a verificare il loro effettivo livello di consapevolezza tecnica sull'accordo che stanno negoziando, stipulando e/o eseguendo. Si verrebbe così a garantire uno strumento di tutela addirittura ulteriore rispetto a quelli imposti dalla normativa vigente. Inoltre, lo strumento dello *smart legal contract* ha l'obiettivo e la capacità di ridurre in modo sostanziale le fattispecie di inadempimento contrattuale[53].

[53] Si veda già Banca d'Italia, *Fintech in Italia. Indagine conoscitiva sull'adozione delle innovazioni tecnologiche applicate ai servizi finanziari,* 2017, www.bancaditalia.it., [92] secondo cui: si tratta di contratti scritti in un linguaggio informatico intellegibile da appositi *software,* in grado di entrare in esecuzione e far rispettare le clausole in essi contenute in modo automatico, una volta soddisfatte le condizioni predefinite.

È quindi fortemente auspicabile che lo *smart legal contract*, a tal fine, preveda *by design* sul piano non soltanto giuridico, ma anche tecnologico, le regole idonee ad assicurare le tutele minime previste dalla normativa settoriale [93]

Ancora, come si è visto, lo *smart legal contract* non pone un problema di inosservanza della forma scritta, se presenta i requisiti sopra indicati, fermo restando anche il rispetto della disciplina in materia di clausole vessatorie che andrebbero sottoscritte autonomamente.

Inoltre, il recesso potrà essere esercitato con una funzione cd. *Kill* al verificarsi dei presupposti normativi.

Infine, anche la disciplina dei rimedi attivabili dal cliente/consumatore al verificarsi di accadimenti che impediscono la corretta esecuzione dell'accordo stipulato può essere rispettata: la tutela di natura rimediale è riproducibile nelle modalità di "creazione" dello smart contract e applicabile. A questo standard il mercato dovrebbe guardare come benchmark.

Resta fermo, e assume rilievo centrale anche ai fini qui considerati della tutela della trasparenza, che se la clausola contrattuale scritta in linguaggio naturale è ambigua o poco chiara, lo *smart legal contract* potrebbe prevedere la possibilità di rivolgersi a un soggetto terzo (un oracolo); ipotesi che potrebbe essere già "programmata" in fase di creazione dello smart legal contract. I profili legati alla gestione (eventualmente automatizzata) interna e/o esterna (anche con eventuale aggancio alla competenza di ADR già esistenti, e.g. Arbitro Bancario Finanziario) è un fronte aperto dal punto di vista concettuale e operativo, pertanto richiederà ulteriore monitoraggio e approfondimento.

**Bibliografia**

[1] Banca d'Italia, "Comunicazione della Banca d'Italia in materia di tecnologie decentralizzate nella finanza e cripto-attività" (2022)
[2] Banca d'Italia, "No. 26 - Integrating DLTs with market infrastructures: analysis and proof-of-concept for secure DvP between TIPS and DLT platforms" (2022)
[3] Cipollone, P., intervento alla conferenza dal titolo "Conference o Digital Platforms and Global Law", URL: https://www.bancaditalia.it/pubblicazioni/interventi-direttorio/int-dir-2022/CIPOLLONE_29_aprile_2022.pdf (2022)
[4] OECD, "The Tokenisation of Assets and Potential Implications for Financial Markets" (2020). URL: https://www.oecd.org/finance/The-Tokenisation-of-Assets-and-Potential-Implications-for-Financial-Markets.pdf
[5] European Law Institute "Principles on Blockchain Technology, Smart Contract and Consumer Protection" (2022)
[6] Szabo, N. "Smart Contract" (1994)
[7] Szabo, N. "Smart Contract Glossary" (1995)
[8] Szabo, N. "Smart Contract: Building Blocks for Digital Markets" (1996)
[9] Szabo, N. "The Idea of Smart Contract; Formalizing Securing Relationships on Public Networks" (1997)
[10] Amato, C. "La computerizzazione del contratto (Smart, data oriented, computable e self-driving contract. Una panoramica)", Europa e Diritto Privato, fasc.4, 1° dicembre 2020, p. 1259 ss.
[11] Buterin, V., "A Next-Generation Smart Contract and Decentralized Application Platform" (2014) consultabile su http://ethereum.org
[12] Belloni, D e Vasoli, F. "Blockchain, Smart Contract e Decreto Semplificazioni, in Cammino Diritto" (2020)
[13] Orlando, S. "Gli smart contract come prodotti software", Annuario 2021 - Osservatorio Giuridico sulla Innovazione Digitale (2021)
[14] Stark, J. "Making Sense of Blockchain Smart Contract", in Coindesk (2016)
[15] Patti, F. P. e Janssen A. U. "Demistificare gli smart contract", ODCC (2020), pp.31-50
[16] Carron, B. and Botteron, V. "How smart can a contract be?", in D. Kraus, T. Obrist, O. Hari (eds.), Blockchains, Smart Contract, Decentralised Autonomous Organisations and the Law, Cheltenham, UK-Northampton, MA, USA (2019) p. 101 ss., spec. pp. 111-114
[17] Rinaldi, G. "Smart contract: meccanizzazione del contratto nel paradigma della blockchain", in G. Alpa (cur.), Diritto e intelligenza artificiale (2020), pp. 353-354
[18] Davola, A. "Blockchain e Smart Contract as a Service: prospettive di mercato a criticità normative delle prestazioni BaaS e SCaaS alla luce di un'incerta qualificazione giuridica", Il Diritto industriale (2020)
[19] Ante, L. "Smart Contract on the Blockchain—A Bibliometric Analysis and Review", BRL Working Paper Series No. 10 (2020)
[20] Gentili, A. "La volontà nel contesto digitale: interessi del mercato e diritti delle persone", Rivista Trimestrale di Diritto e Procedura Civile, fasc.3 (2022), pag. 701
[21] Pardolesi, R. – Davola, A. "What is wrong in the debate about smart contract", Book of Short papaers, Società Italiana Statistica (2019), p. 481
[22] Cappai, M. "The role of private and public regulation in the case study of crypto-assets: The Italian move towards participatory regulation", Computer Law & Security Review, 49 (2023)
[23] EU Blockchain Observatory Forum, Report "Smart Contract" (2022), pp. 10-13, URL: https://www.eublockchainforum.eu/sites/default/files/reports/SmartContractsReport_Final.pdf
[24] Bertuzzi, L. "AI Act Enters Final Phase of EU Legislative Process" (2023) https://www.euractiv.com/section/artificial-intelligence/news/ai-act-enters-final-phase-of-eu-legislative-process/.
[25] IOSCO "Compliant Handling and Redress System for Retail Investor" Final Report (2021)

[26] Schrepel, T "Law+Technology", Stanford CodeX Working Paper, p. 6, (2022) http://dx.doi.org/10.2139/ssrn.4115666
[27] Malvagna, U. "Digital securities: prime note sul decreto di attuazione del DLT Pilot", Rivista di Diritto Bancario (2023)
[28] Annunziata, F. "Il nuovo Regolamento UE in materia di Distributed Ledger", Rivista di Diritto Bancario (2022)
[29] Annunziata, F., Chisari A. C. e Amenndola, P. R. "DLT-Based Trading Venues and EU Capital Markets Legislation: State of the Art and Perspectives under the DLT Pilot Regime", Bocconi Legal Studies Research Paper No. 4344803 (2023)
[30] Di Ciommo, F. "Smart contract e (non-)diritto. il caso dei mercati finanziari", Nuovo Diritto Civile, IV Anno (2019), p. 257 ss.
[31] Cuccuru, P. "Beyond bitcoin: an early overview on smart contract", International Journal of Law and Information Technology, vol. XXV (2017), 179 ss.
[32] Cutts, T. "Smart Contract and Consumers", West Virginia University (2019)
[33] Karamanlioğlu, A. "Concept of Smart Contract. A Legal Perspective", Kocaeli Üniversitesi Sosyal Bilimler Dergisi (2018), p. 29 ss
[34] Kasprzyk, K. "The Concept Of Smart Contract From The Legal Perspective", Review of Comparative Law Vol. XXXIV (2018)
[35] Raskin, M. "The Law and Legality of Smart Contract", Geo. L. Tech. Rev.305 (2017), p. 312
[36] Parola, L., Merati, P. e Gavotti G. "Blockchain e smart contract: questioni giuridiche aperte", Contr. (2018), p. 681 ss., spec. 685.
[37] Pellegrini, T. "Prestazioni auto-esecutive: smart contract e dintorni", Comp. dir. civ. (2019), pp. 27-28.
[38] Jerry, I. – Hsiao, H. "Smart Contract on the Blockchain – Paradigm Shift for Contract Law?", US – China Contract Law Review, Vol. 14 (2017), p. 686 ss.
[39] Giaccaglia, M. "Considerazioni su blockchain e smart contract", Contratto e impresa (2019), p. 951
[40] Giuliano M., "La blockchain e gli smart contract nell'innovazione del diritto nel terzo millenio", Diritto dell'Informazione e dell'Informatica (Il), fasc.6 (2018), p. 989 ss.
[41] Catchlove, P. "Smart Contract: A New Era of Contract Use", disponibile su SSRN (3090226), p. 15
[42] Clack, C. D. et al., "Smart contract templates: foundations, design landscape and research directions 2" (last revised 2017)
[43] Durovic, M. and Lech, F. "The Enforceability of Smart Contract", Italian Law Journal, 2 (2019), p. 504 ss.
[44] Eenmaa-Dimitrieva, H. and Schimdt – Kessen, M. J. "Creating Markets in No-trust Environments: the Law and Economics of Smart Contract", 35 C.L.S. Rev. (2019), pp. 69-88
[45] Stazi, A. "Smart Contract: Elements, Pathologies and Remedies", European University of Rome, National University of Singapore (Forthcoming in: Law and Change: An Asian Perspective, edited by J. Loo & N. Remolina Leon, SMU) 2022
[46] Werbach, K. and Cornell, N. "Contract ex machina", Duke Law Journal (2017) p. 338 ss.
[47] Caggiano, I. A. "Il contratto nel mondo digitale", (a cura di Lucilla Gatt) Il Contratto del Terzo Millennio - Dialogando con Guido Alpa, Editoriale Scientifica Napoli (2018), p. 61 ss.
[48] Maugeri, M. "Smart contract e disciplina dei contratti", Il Mulino, (2021)
[49] Finocchiaro, G. "Il contratto nell'era dell'intelligenza artificiale", Rivista Trimestrale di Diritto e Procedura Civile, fasc.2 (2018), p. 441 ss.
[50] Carriero, V. "Smart Contract In The Blockchain Context: What Happens?", European Journal of Privacy Law & Technologies (2019)
[51] Irti, N. "Scambi senza accordo", in Riv. trim. dir. proc. civ. (1998), p. 350 ss.

[52] Bassan, F. "Digital Platforms and Blockchains: The Age of Participatory Regulation", European Business Law Review (2023)
[53] Bassan, F. "Digital Platforms and Global Law, Edward Elgar Publishing" (2021)
[54] Bassan, F. "Potere dell'algoritmo e resistenza dei mercati in Italia. La sovranità perduta sui servizi", Rubbettino (2019)
[55] Bassan, F. "Web 3 in Transition", CPI-Tech Croniche (2023)
[56] Rabitti, M. "Prodotti finanziari tra regole di Condotta e di organizzazione. I limiti di MiFID II" Rivista di Diritto Bancario, fasc. I (2020), pp. 145-177
[57] De Filippi, P. e Wright, A. "Blockchain and the Law. The Rule of Code", Harvard University Press (2018), p. 77 ss.
[58] Giancaspro, M. "Is a smart contract really a smart idea? Insights from a Legal Perspective", Computer Law & Security Review (2017), pp. 7-8
[59] Lauslahti, K., Mattila, J. and Seppälä, T. "Smart Contract - How will Blockchain Technology Affect Contractual Practices?", ETLA Report, No 68, Elinkeinoelämän Tutkimuslaitos (2017)
[60] Levi, S.D. and Lipton, A. B., "An Introduction to Smart Contract and Their Potential and Inherent Limitations" (2018)
[61] Howell, B. E. and Potgieter, P. H. "Uncertainty and dispute resolution for blockchain and smart contract institutions", Cambridge University Press (2021)
[62] Lim, A. "502 Bad Gateway: Rebooting Smart Contract, Legal Information Management", Volume 20, Issue 2 (2020), pp. 106 – 107
[63] Mik, E. "Smart contract: Terminology, Technical Limitations and Real World Complexity", Law, Innovation and Technology, 9 (2017) pp. 269-300
[64] Savelyev, A. "Contract law 2.0: Smart Contract as the Beginning of the End of Classic Contract Law", Information & Communications Technology Law (2017), p. 124 ss.
[65] Faini, F. "Blockchain e diritto: la «catena del valore» tra documenti informatici, smart contract e data protection", Responsabilità Civile e Previdenza, fasc.1 (2020), p. 307 ss.
[66] Fauceglia, D. "Il problema dell'integrazione dello smart contract", I Contratti (2020)
[67] Crisci, S. "Intelligenza artificiale ed etica dell'algoritmo", Foro amm. (2018), p. 1803 ss.
[68] Gentili, A. "Crisi delle categorie e crisi degli interpreti", Rivista di diritto civile (2021)
[69] Caldana, F. G. e Colosio, C. "Smart Contract: problemi interpretativi. Cenni di carattere sistematico" (2021)
[70] Levy, K.E.C. "Book Smart, not Street-Smart: Blockchain-Based Smart Contract and the Social Workings of Law", Engaging Science, Technology and Society (2017)
[71] Casey, M. "Could Blockchain Technology Help the World's Poor?", World Economic Forum Agenda (2016)
[72] Gitti, G. "Tecnologie digitali, persona e istituzioni", Rivista di diritto civile (2020)
[73] Lusardi, A – Mitechell, O. "Financial Literacy and Retirement Preparedness: Evidence and Implications for Financial Education", Business Economics (2007)
[74] BIS Working Papers, No. 1061 "Cryptocurrencies and Decentralized Finance (2022), pp. 11-19
[75] Consob, "Tokenizzazione di azioni e azioni tokens", Quaderni giuridici (2023)
[76] OCSE, "Crypto-asset Reporting Framework and Amendments to the Common Reporting Standard" (2022)
[77] Sklaroff, J. M. "Smart Contract and the Cost of Inflexibility, University of Pennsylvania Law Review, Vol. 166, (2017), p. 287 ss.
[78] Manente, M. "L. 12/2019 – Smart contract e tecnologie basate su registri distribuiti – Prime note", Consiglio Nazionale del Notariato, approvato dalla Commissione informatica il 4 aprile 1029, pp. 6-7
[79] Benedetti, A. M. "Contratto, algoritmi e diritto civile transnazionale: cinque questioni e due scenari", Rivista di diritto civile, fasc. 3 (2021), pp. 415-417

[80] Lemme, G. "Gli smart contracts e le tre leggi della robotica", Analisi Giuridica dell'Economia, fasc. 1 (2019)
[81] Nuzzo, G. "Gli smart contract tra esigenze di calcolabilità e gestione delle sopravvenienze" Contenuto e limiti dell'autonomia privata in Germania e in Italia, a cura di Bordiga e Wais, Torino (2021)
[82] Cerrato, S. A. "Appunti su smart contract e diritto dei contratti", Banca Borsa Titoli di Credito, fasc. 3 (2020) p. 370 ss
[83] Sirena, P. e Patti, F. P. "Smart Contract and Automation of Private Relationships, Bocconi Legal Studies Research Paper Series, No. 3662402 (2020)
[84] Cappiello, B. "Dallo smart contract computer code allo smart (legal) contract. I nuovi strumenti (para) giuridici alla luce della normativa nazionale e del diritto internazionale privato europeo: prospettive de jure condendo", Diritto del Commercio Internazionale, fasc. 2 (2020), p. 477 ss.
[85] Rabitti, M. e Paglietti M. C. "A matter of Time. Digital-Financial Consumers Vulnerability in the Retail Payments Market", European Business Law Review, 33, no. 4 (2022), pp. 581-606
[86] Durovic, M. e Janssen, A. "The Formation of Blockchain-based Smart Contracts in the Light of Contract Law, European Review of Private Law, 6, pp. 753-772 (2019)
[87] Musio, A. "La storia non finita dell'evoluzione del contratto tra novità tecnologiche e conseguenti esigenze di regolazione", NGCC, fasc. 1 (2021), p. 234 ss
[88] Remotti, "Blockchain smart contract. Un primo inquadramento", Osservatorio del diritto civile e commerciale, fasc. 1 (2020), p. 200 ss
[89] Cuccuru, P. "Blockchain ed automazione contrattuale. Riflessioni sugli smart contract", NGCC 2017, II, p. 111 ss
[90] Werbach, K. "Trust, but Verify: Why the Blockchain Needs the Law", 33 Berkeley Tech. L.J. 489 (2018), pp. 545-546
[91] Banca d'Italia, "Disposizioni in materia di trasparenza delle operazioni e dei servizi bancari e finaziari" (2019)
[92] Banca d'Italia, "Fintech in Italia. Indagine conoscitiva sull'adozione delle innovazioni tecologiche applicate ai servizi finanziari" (2017)
[93] Sirgiovanni, B. "Lo smart contract e la tutela del consumatore: la traduzione del linguaggio naturale in linguaggio informatico attraverso il legal design", Le Nuove Leggi Civili Commentate, p. 214 ss. (2023)

# PARTE II – SMART CONTRACT: PROFILI TECNOLOGICI

## Premessa

Gli smart contract sono definiti dalla particolare tecnologia blockchain (o, più in generale, dalla DLT), quindi le loro caratteristiche cambiano in funzione della specifica tecnologia che utilizzano. La tecnologia consente di svolgere in via automatica (auto-esecutiva) molte attività e consente altresì di "registrare" in modo sicuro e immutabile non solo il contenuto dei contratti ma anche la loro "vita", dalla fase negoziale a quella di esecuzione.

Per dare risposta agli interrogativi giuridici sugli smart contract e sugli smart legal contract occorre verificare quali soluzioni la tecnologia mette a disposizione e quali problemi queste risolvono. Su questa base potranno individuarsi standard minimi per garantire i presidi imposti dalla normativa vigente nonché le tutele che anzi, proprio la tecnologia consente di rafforzare, sul piano ad esempio, della certezza e della trasparenza. La tecnologia infatti può svolgere alcune funzioni di regolazione (ex ante) o di controllo e supervisione (on going e ex post) integrando l'attività del regolatore. Queste funzioni sono definite in dottrina *regulation by technology* [74, 75] o *rule of code* [76, 77]

Questa seconda parte del documento si concentra sui profili tecnologici, rispettivamente della blockchain (Sezione I) e degli smart contract (Sezione II).

Nello specifico, nella Sezione I, assumendo il punto di vista delle blockchain, vengono illustrati gli elementi (o proprietà) della tecnologia ritenuti rilevanti per lo sviluppo e per l'esecuzione di smart contract.

## Sezione I – Tassonomia per l'analisi delle piattaforme blockchain

### 1. Introduzione

La tecnologia blockchain consente la decentralizzazione di sistemi transazionali, tramite l'applicazione di un protocollo decentralizzato eseguito su una rete peer-to-peer[54]. Il protocollo associato alla blockchain definisce le regole di aggiornamento dello *shared ledger* (o registro condiviso) e garantisce la sicurezza e l'integrità dei dati memorizzati senza la necessità di intervento di una terza parte fidata.

I dati memorizzati nel registro possono rappresentare un qualsiasi cambiamento di stato, pertanto sono organizzati in transazioni. Le transazioni, ordinate cronologicamente, sono raggruppate e memorizzate in blocchi[55]. Ogni blocco contiene un gruppo di transazioni firmate crittograficamente da uno o più partecipanti al protocollo. L'organizzazione del blocco e la struttura della transazione dipendono dalla particolare tecnologia blockchain. Una transazione consiste genericamente nella registrazione di un evento (ad esempio, un trasferimento di valore tra due o più utenti, l'invocazione di uno smart contract) che determina un cambiamento dello stato registrato nel *ledger*. Per garantire la proprietà di immutabilità, ogni blocco è connesso al precedente in maniera crittograficamente sicura. Questa operazione avviene tecnicamente per mezzo di un puntatore al blocco precedente

[54] Tipologia di networking in cui ogni nodo può comunicare direttamente con un altro senza passare attraverso un punto centralizzato Source NISTIR 8202.

[55] L'organizzazione del registro delle transazioni sotto forma di blocchi concatenati crittograficamente tra loro caratterizza le blockchain all'interno del più ampio contesto delle c.d. "Distributed Ledger Technologies" (DLT). Sebbene in questo documento ci si concentri sulle blockchain e si faccia riferimento esplicito principalmente ad esse, al netto di tale differenza quanto descritto ha valenza in generale anche per le DLT. Eventuali eccezioni, ove non agevolmente desumibili dal contesto, vengono segnalate.

realizzato grazie al calcolo di una funzione di *hash*[56] applicata al contenuto del blocco corrente (che include il puntatore (*hash*) al blocco precedente); il risultato della funzione di *hash* è quindi parte integrante del blocco. Dato che ogni blocco è crittograficamente concatenato al precedente grazie al puntatore *hash*, ne consegue che una modifica ad un blocco invalida l'integrità di tutti i blocchi successivi.

La blockchain sostituisce la gestione centralizzata delle informazioni e dell'infrastruttura ed è pertanto fondamentale che il protocollo adottato sia in grado di assicurare proprietà di sicurezza, integrità ed efficienza. Ove sia richiesto a una blockchain di operare su larga scala e processare un numero elevato di transazioni, è necessario che ciò avvenga senza compromettere la disponibilità e l'integrità delle informazioni.

Negli ultimi anni, la blockchain ha trovato applicazione in numerosi settori [1, 2, 3] e l'interesse dell'industria e dell'accademia ha alimentato l'offerta di nuovi protocolli e piattaforme c.d. "layer 1"[57]. Ad oggi esistono decine di blockchain, ognuna con le proprie caratteristiche e *tradeoff* [8]: alcune favoriscono la scalabilità, altre puntano a ottimizzare lo sviluppo di applicazioni decentralizzate o a fornire meccanismi di privacy avanzati.

Nonostante le potenzialità di miglioramento dei processi che la tecnologia blockchain e gli smart contract hanno dimostrato di garantire in diversi domini (ad esempio, gestione delle *supply chain* [62], finanza [63], *health care* [64]), la loro applicazione in contesti regolamentati è ancora limitata e presenta ampi margini di miglioramento, sul piano sia tecnico (per gli smart contract code) sia della trasparenza (per gli smart legal contract). La regolamentazione risulta infatti frammentata e rende difficoltosa l'adozione della tecnologia su larga scala, in particolare per applicazioni governative e istituzionali [4].

La caratteristica nuova introdotta da queste tecnologie, in particolare per quelle che supportano forme avanzate di programmabilità, è che esse consentono di incorporare le regole: chi sviluppa uno smart contract ha la possibilità, ad esempio, di orientare flussi finanziari, di comprimere diritti fondamentali, di garantire un'esecuzione certa e immutabile del contratto nei termini previsti, di offrire una esecuzione contestuale di prestazione e contro-prestazione, di permettere il ricorso a una giurisdizione nazionale o anche a preventive forme di conciliazione sfruttando ad esempio degli oracoli [5]. Il supervisore non può quindi limitarsi a intervenire a posteriori e deve poter verificare il processo nel momento in cui viene disegnato. Inoltre le caratteristiche degli smart contract possono anche essere influenzate dalla particolare tecnologia blockchain che utilizzano; a tal fine, si ritiene utile far precedere l'analisi della tassonomia degli smart contract da una analisi di alcune caratteristiche delle blockchain.

Del resto la tecnologia blockchain resta a tutt'oggi uno strumento di complessa comprensione; come discusso nel Focus 5 sullo stato dell'arte (cfr. infra), definire strategie di analisi e comparazione tra le diverse piattaforme è un tema aperto nell'ambito scientifico.

[56] Una funzione di hash equivale all'applicazione di un algoritmo matematico a dati di input (di qualsiasi tipo e dimensione: file, testi, immagini, etc.) in grado di produrre in output una rappresentazione sintetica e relativamente unica del dato in input. Source NISTIR 8202

[57] Si intende una rete blockchain di base che definisce le funzionalità essenziale di un protocollo blockchain: eseguire le transazioni sulla blockchain e aggiornare un ledger condiviso. Diverse reti layer-1 differiscono per protocollo di consenso, grado di decentralizzazione, proprietà di sicurezza, efficienza, interoperabilità, funzionalità tecniche, struttura del ledger e scalabilità. Si parla invece di framework o protocollo secondario "layer 2" nel caso di soluzioni tecnologiche costruite al di sopra di una blockchain esistente; ciò avviene spesso al fine di per variarne alcune caratteristiche (ad esempio, scalabilità, decentralizzazione, visibilità delle informazioni).

**FOCUS 5 - Lo stato dell'arte**

Ad oggi, a seguito dell'analisi che abbiamo svolto sullo stato dell'arte sembrerebbe non essere disponibile uno strumento unico condiviso per comparare diverse blockchain e analizzarne le peculiarità in termini, ad esempio, di sicurezza e efficienza delle piattaforme. Infatti, molti degli studi condotti fino ad oggi comparano caratteristiche specifiche e tendono ad adottare analisi che puntano a evidenziare vantaggi e svantaggi di una blockchain rispetto ad altre o a mostrare in maniera sperimentale casi d'uso specifici. Ad esempio alcuni [8] propongono una classificazione delle proprietà fondamentali della blockchain e dei loro *tradeoff* focalizzandosi, su caratteristiche generali di sicurezza e *performance*. Un approccio qualitativo è stato utilizzato anche da chi, per la valutazione dei protocolli di consenso delle blockchain, si è concentrato su aspetti di sicurezza e *performance* che questi possono garantire, senza però valutare gli impatti sull'infrastruttura, come ad esempio la decentralizzazione e la *privacy* [9]. Altri propongono [10] un'analisi qualitativa dei protocolli di consenso utilizzati dalle blockchain *permissioned*[58]. Altri ancora [11, 12] forniscono esclusivamente un'analisi di correttezza/adeguatezza rispetto alle proprietà di *safety* e *liveness*, focalizzando lo studio su un set ridotto di blockchain permissioned. Un'analisi quantitativa è stata invece proposta da chi [13] offre analisi sperimentali di *benchmarking* delle performance delle blockchain Ethereum e Hyperledger Fabric, senza valutarne altre proprietà fondamentali come la sicurezza e la decentralizzazione. Infine, altri ancora [14] hanno analizzato l'efficienza delle blockchain *permissioned* in casi d'uso finanziari, senza però valutarne i requisiti in termini, ad esempio, di sicurezza e affidabilità.

L'approfondimento che segue ha lo scopo di individuare una tassonomia di parametri utili a rispondere alla seguente domanda:

***Quali sono i parametri o requisiti principali che andrebbero presi in considerazione per analizzare le caratteristiche di una piattaforma blockchain?***

L'individuazione di un set di parametri principali consente di analizzare le piattaforme blockchain; essi potrebbero rappresentare uno strumento utile a supporto dello sviluppo di regolamentazione di questa tecnologia. Su queste basi sarà possibile costruire una tassonomia degli smart contract sul piano della tecnologia (Parte II, Sezione II), che sarà a sua volta presupposto per un'eventuale successiva definizione delle linee guida.

## 2. Metodologia

Il funzionamento di una blockchain presuppone molteplici ambiti dell'informatica e della matematica, quali la crittografia, i sistemi distribuiti e le telecomunicazioni. L'obiettivo perseguito dalla tecnologia blockchain è quello di assicurare la validità e il tracciamento delle transazioni pur in assenza di fiducia reciproca tra gli attori coinvolti, attraverso un protocollo condiviso e una infrastruttura tecnologica decentralizzata.

Nonostante i benefici promessi dalla tecnologia, l'implementazione di applicazioni specifiche ha rivelato dipendenze tra le proprietà tipiche della tecnologia blockchain che portano a dover scendere a compromessi [8, 15]. In sintesi, il miglioramento di una proprietà può determinare il peggioramento di un'altra proprietà. Ad esempio, esiste un compromesso tra la disponibilità delle informazioni memorizzate e la consistenza garantita nell'ambito dei registri distribuiti [8]. Di conseguenza, risulta

[58] Le blockchain permissioned sono un particolare tipo di blockchain in cui i nodi che possono aggiornare la blockchain sono limitati, noti e autorizzati. Si rinvia al paragrafo 3.1 "Architettura di rete", per un approfondimento.

cruciale nell'ambito dello sviluppo di applicazioni basate su blockchain (o su smart contract) tener presente tali *tradeoff*.

A tale scopo, nel presente documento, vengono raccolte le principali caratteristiche (o parametri) qualitative ritenute fondamentali per l'analisi delle blockchain. La scelta dei parametri è stata condotta partendo dallo stato dell'arte (ad esempio, [8, 9, 10, 12, 16, 17, 18, 19]) e raccogliendo gli aspetti qualitativi ritenuti utili per rappresentare la tecnologia blockchain. Nello specifico, nel presente documento, l'analisi viene condotta su tre verticali principali:

1. caratteristiche tecniche;
2. modello economico;
3. ecosistema e dati on-chain.

Nelle sezioni seguenti vengono approfondite le singole verticali, motivando le scelte effettuate per la selezione dei parametri qualitativi. Inoltre, in conclusione, viene proposto un approccio metodologico che potrebbe essere utilizzato per l'acquisizione delle informazioni necessarie per analizzare una specifica tecnologia blockchain in relazione ai parametri individuati.

## 3. Caratteristiche tecniche

In questa sezione vengono considerati gli aspetti architetturali delle reti blockchain, i parametri di efficienza – intesa come scalabilità e decentralizzazione - e sicurezza, nonché la flessibilità – intesa come capacità computazionale (programmabilità) ottenuta tramite utilizzo di smart contract, la configurabilità del sistema e l'interoperabilità con altre piattaforme.

### 3.1 Architettura di rete

Una rete blockchain è un sistema decentralizzato di nodi connessi attraverso la rete al fine di:

- mantenere un ledger condiviso su ogni nodi della rete;
- creare e validare le transazioni e i blocchi;
- aggiornare il ledger tramite specifiche regole (o protocollo).

Per analizzare la robustezza del modello di rete che una blockchain propone occorre ipotizzare canali di comunicazione non sicuri, che possono essere corrotti o soggetti a guasti. Si considera una rete parzialmente sincrona [20] che prevede comunicazioni asincrone in cui i messaggi vengono consegnati entro un certo limite temporale. Questo modello è riconosciuto dalla teoria dei sistemi distribuiti come modello di riferimento per simulare una rete Internet.

I nodi della rete vanno distinti tra nodi *client* e nodi *validatori*. I nodi *validatori* sono incaricati di verificare la validità delle transazioni e dei blocchi e partecipare al meccanismo di selezione del prossimo blocco da aggiungere alla catena (protocollo di consenso). I nodi *client* mantengono lo stato della blockchain ed espongono agli utenti le interfacce per interagire con le funzionalità del protocollo della blockchain, quali l'invio di transazioni o l'invocazione di smart contract.

*I Nodi client* e *validatori* possono operare su reti con diversi livelli di visibilità [21, 22, 23, 24]:

1. rete pubblica: non c'è nessuna restrizione sulla visibilità delle informazioni memorizzate sul *ledger*;
2. rete privata: soltanto un gruppo ristretto di partecipanti autenticati può accedere alle informazioni memorizzate nel ledger.

Ogni modello di rete è inoltre caratterizzato da due livelli di permessi distinti:

1. *permissionless*: ai nodi non è richiesto alcun permesso per entrare a far parte della rete e partecipare al consenso;
2. *permissioned*: solo i nodi autorizzati sono abilitati, partecipano al consenso e quindi alla validazione e alla scrittura delle informazioni sul ledger.

L'architettura di rete ha riflessi sulle attività che una blockchain può garantire: blockchain pubbliche rendono le transazioni e i dati "trasparenti" e quindi non confidenziali "*by design*", anche se esistono soluzioni tecniche come il mascheramento dei dati funzionali alla riservatezza; diversamente, le blockchain permissioned sono caratterizzate da un consorzio ristretto di attori che ne gestiscono l'infrastruttura ed il ledger sottostante; nelle blockchain permissioned è necessaria una componente di fiducia rispetto i gestori del sistema che ne determinano la sicurezza: in caso il gruppo ristretto sia corrotto, l'intera blockchain risulterebbe corrotta [8, 15, 32].

### 3.2 Sicurezza, scalabilità, decentralizzazione

Le blockchain vengono comunemente descritte attraverso tre proprietà fondamentali: *sicurezza*, *scalabilità* e *decentralizzazione* sulle quali Vitalik Buterin, che ha creato Ethereum, ha elaborato il cosiddetto "*Scalability Trilemma*" [25] ovvero la necessità di bilanciare le condizioni di *tradeoff* di sicurezza, scalabilità e decentralizzazione in una rete blockchain. Il Trilemma afferma che è impossibile avere un pari livello di priorità di tutte e tre le proprietà nello stesso tempo: occorre raggiungere un compromesso a discapito di una delle tre dimensioni.

Nelle sezioni che seguono le tre proprietà, e le caratteristiche delle blockchain che le influenzano, vengono presentate in dettaglio.

#### 3.2.1 Parametri di sicurezza

In questa sezione vengono dettagliati i principali profili da analizzare per valutare gli aspetti di sicurezza di una blockchain. Si fa riferimento alle proprietà di sicurezza fondamentali, quali la confidenzialità, l'integrità, la disponibilità e la consistenza [8, 26] La confidenzialità è intesa come la capacità di un sistema di non rivelare informazioni ad utenti non autorizzati; l'integrità è intesa come la capacità di un sistema di mantenere inalterate le transazioni e le informazioni contenute nel ledger; la disponibilità è intesa come la probabilità che un sistema sia in grado di ricevere richieste di aggiornamento (e quindi di processare transazioni) ad ogni istante di tempo; la consistenza è intesa come l'assenza di stati del ledger divergenti tra i nodi della blockchain. Nelle sezioni successive vengono analizzate in dettaglio le caratteristiche delle blockchain che hanno impatto sulle proprietà di sicurezza.

##### 3.2.1.1 *Confidenzialità*

Secondo la narrazione dominante [37] la blockchain, per sua natura immutabile e decentralizzata, è in conflitto con i requisiti di confidenzialità e con la disciplina europea in materia di protezione dei dati personali (GDPR). Infatti, il GDPR richiede che: (i) i dati siano sotto il controllo di un singolo che ne è responsabile e, (ii) in ogni momento questi dati possano essere modificati o rimossi dal sistema informativo di riferimento. Queste proprietà non sono garantite nativamente su tutte le tipologie di blockchain. Tuttavia, esistono *tradeoff* che vanno analizzati quando si affronta il tema della protezione dei dati personali nella blockchain.

In particolare, blockchain *permissioned* offrono un sistema controllato in cui solo le parti autorizzate e autenticate hanno il controllo delle informazioni. Questo tipo di rete è caratterizzato da un insieme definito e limitato di validatori che hanno il controllo sulla rete, sul *ledger* condiviso e sulle

informazioni in esso contenute. In sistemi permissioned, i validatori hanno il potere di modificare e, sotto certe ipotesi, addirittura cancellare transazioni dal ledger in quanto hanno sostanzialmente il controllo dell'infrastruttura [15]. Questo ha un impatto sulla sicurezza del sistema: da un lato, permette ad un gruppo ristretto di partecipanti di invalidare la proprietà di immutabilità dei dati nella blockchain; dall'altro, consente una più agevole correzione di eventuali errori materiali. Pertanto, il modello di fiducia nelle blockchain *permissioned* deve tenere conto di tali caratteristiche e, se del caso, deve essere più stringente rispetto alle reti permissionless, in cui la fiducia viene distribuita sulla totalità della rete decentralizzata [15].

Nelle reti pubbliche (*permissionless* o *permissioned*) chiunque si colleghi alla rete può leggere le informazioni scritte nel *ledger*. In questo caso, la conformità alle normative in tema di protezione dei dati personali deve essere garantita tramite meccanismi di "*data-masking*" (offuscamento dei dati) e di anonimizzazione delle informazioni. Tuttavia, alcune piattaforme offrono soluzioni alternative come l'utilizzo di tecniche crittografiche per il diritto all'oblio on-chain (*Right to Be Forgotten - RTBF*), meccanismi di *zero-knowledge* [38], utilizzo di "piattaforme ad accesso controllato" o di sotto-reti private.

3.2.1.2 *Integrità*

La proprietà di integrità di una blockchain indica che le transazioni e i blocchi seguono le regole di validità definite dal protocollo della blockchain. Ad esempio, il protocollo può richiedere che una transazione che scambia denaro da un utente ad un altro, per essere valida, deve spendere un quantitativo di denaro disponibile (in una transazione precedente o in un bilancio) e deve essere firmata dall'utente che è in grado di spendere quella specifica quantità di denaro.

In generale, le regole di validità definite dal protocollo blockchain possono fare uso di funzioni crittografiche, quali le funzioni *hash* e le firme digitali. Le funzioni *hash* permettono di ottenere valori di controllo a partire da dati (generalmente un intero blocco), semplificando così la rilevazione di eventuali modifiche non autorizzate agli stessi. Le firme digitali, invece, consentono di verificare l'autenticità dei dati provenienti da un mittente, con la garanzia di non ripudio. La preservazione dell'integrità svolge un ruolo fondamentale per garantire la sicurezza di una blockchain [8].

3.2.1.3 *Disponibilità*

In generale, la disponibilità di un sistema viene definita in termini di probabilità che il sistema operi correttamente ad ogni istante di tempo. Poiché la blockchain è decentralizzata, in quanto ogni nodo detiene una copia dell'intero *ledger*, viene considerata altamente disponibile. In altre parole, la disponibilità del *ledger* aumenta all'aumentare del numero di nodi che ne mantengono una copia.

3.2.1.4 *Consistenza*

Tutti i nodi di un registro distribuito (nel caso specifico, di una blockchain), mantengono una replica locale del registro; di conseguenza, tutti i nodi devono essere sincronizzati e devono concordare sulle modifiche da applicare ai dati memorizzati (ovvero, sullo stato) al fine di garantire la consistenza del registro. Per tale scopo, viene utilizzato un protocollo di consenso.

In generale, un protocollo di consenso consente di gestire la negoziazione tra i nodi e di raggiungere, entro un certo lasso di tempo, un accordo su come aggiornare lo stato condiviso, anche in presenza di guasti (o fallimenti) dei nodi e della rete [10, 18, 27]. Nello specifico, nelle blockchain, il consenso viene utilizzato per definire l'ordine delle transazioni da aggiungere al registro e garantire che i nodi della rete convergano sullo stesso stato risolvendo (in maniera probabilistica o deterministica) problemi come il *double spending* [10].

In letteratura [11, 27, 29, 30], la correttezza di un protocollo di consenso viene misurata in termini di *safety* e *liveness*. La *safety* rappresenta la corretta esecuzione del protocollo anche in presenza di condizioni avverse (fallimenti); mentre la *liveness* rappresenta la capacità del protocollo di progredire e terminare anche in presenza di fallimenti.

Alcuni autori [59] discutono come le proprietà di *safety* e *liveness* siano legate al teorema CAP[59] definito nell'ambito dei sistemi distribuiti [28]. Per gli scopi del presente documento, il teorema CAP può essere enunciato in termini di impossibilità da parte di un registro distribuito, condiviso, replicato e sincronizzato tra molteplici nodi di garantire, in presenza di errori di comunicazione tra i nodi, contemporaneamente operazioni di lettura e scrittura dei dati atomiche e rendere sempre disponibili le ultime scritture sui dati. Nel teorema CAP, la consistenza è legata alla proprietà di *safety*: ogni replica del dato sarà sempre corretta (ovvero aggiornata). La disponibilità, invece, è legata alla proprietà di *liveness*: alla fine, ogni richiesta di aggiornamento o lettura dei dati riceverà prima o poi una risposta. Di conseguenza, il teorema CAP afferma che è impossibile per qualsiasi protocollo che implementa un registro con operazioni di lettura e scrittura atomiche garantire contemporaneamente *safety* e *liveness* in una rete soggetta a partizioni.

Di seguito vengono trattati due concetti legati alla consistenza nell'ambito della tecnologia blockchain, quali la finalità e le *fork*.

3.2.1.4.1 Finalità

La finalità (o *finality)* è una proprietà fondamentale introdotta per determinare le differenze tra algoritmi di consenso delle blockchain [31]. Questa proprietà viene applicata ai blocchi del *ledger* (e alle transazioni) e definisce la capacità di un sistema di considerare una transazione immutabile e irreversibile all'interno del *ledger*. Si distingue tra due tipologie di *finality* [23]:

> ***Finalità probabilistica***: indica che la garanzia che una transazione sia valida, e quindi parte del ledger in maniera irreversibile, cresce con l'aumentare di nuovi blocchi successivi al blocco contenente la transazione stessa. Tra gli algoritmi di consenso con finalità probabilistica si cita Proof-of-Work (PoW) in cui il lavoro computazionale necessario per modificare o rimuovere una transazione dal ledger è tanto più elevato quanto più sono i numeri di blocchi che seguono la transazione stessa. In alcune versioni di Proof of Stake (PoS), invece, la finalità viene perseguita disincentivando economicamente il comportamento di attori malevoli. La finalità probabilistica non offre forti garanzie di consistenza in quanto le transazioni di un ledger potrebbero essere rimpiazzate con altre in caso di conflitti.
>
> ***Finalità deterministica***: assicura che una transazione sia valida subito dopo essere stata inserita in un blocco della blockchain. Fanno parte degli algoritmi di consenso con finalità deterministica il protocollo Practical Byzantine Fault-Tolerant (PBFT) e le sue varianti [18, 33]. In tali protocolli, i validatori votano sul blocco successivo da aggiungere alla catena; quando un certo numero di voti è raggiunto (quorum), ogni nodo della rete aggiorna il proprio ledger in maniera simultanea. Una transazione, se approvata da un quorum di validatori, risulta immediatamente parte del ledger in maniera irreversibile. Assumendo un sistema avversario in cui i nodi possono essere guasti o corrotti (ovvero soggetti a guasti Bizantini), la teoria dei sistemi distribuiti fissa il valore minimo del quorum di approvazioni a ⅔ del numero totale di nodi validatori della rete [33].

[59] Il teorema CAP è stato introdotto come un *trade-off* tra consistenza (*consistency*), disponibilità (*availability*) e tolleranza alle partizioni (*partition tolerance*) nell'ambito dei sistemi distribuiti dove un servizio viene implementato e fornito da molteplici nodi. Per consistenza si intende la capacità da parte di ogni nodo di fornire una risposta corretta a ogni richiesta formulata. Per disponibilità si intende la capacità del servizio di fornire sempre una risposta per ogni richiesta formulata. Per tolleranza alle partizioni di rete si intende che il servizio continua a operare anche a fronte di fallimenti nella comunicazione tra i nodi.

Si precisa che ulteriori dimensioni vengono adottate in letteratura per classificare gli algoritmi di consenso. Ad esempio, in [24], si fa una distinzione tra algoritmi di consenso del tipo *lottery-based* e *vote-based*. I primi prevedono la selezione, in base a un criterio specifico del protocollo stesso, di un nodo validatore per proporre il prossimo blocco da aggiungere alla catena; i secondi, invece, utilizzano un approccio più tradizionale basato su *round* di voti. PoW e PoS sono un tipico esempio di protocolli del tipo *lottery-based* mentre PBFT rientra nella categoria *vote-based*.

3.2.1.4.2 *Fork*

La proprietà di consistenza viene meno in un sistema blockchain nell'eventualità in cui diverse versioni del *ledger* condiviso esistono contemporaneamente; di conseguenza, i diversi nodi della rete blockchain non sono sincronizzati sullo stesso stato del *ledger* in un certo istante di tempo. Questo scenario determina il fenomeno delle *fork*. In generale, le fork possono essere accidentali oppure intenzionali.

Le fork accidentali sono tipicamente conseguenza del meccanismo di consenso adottato dalla particolare tecnologia blockchain. Se due o più nuovi blocchi *validi* vengono creati (quasi) contemporaneamente dai nodi validatori, la catena di blocchi crescerà dividendosi in due o più rami *validi*. In questo caso, la consistenza della blockchain risulta (in genere, solo temporaneamente) violata [31] rendendo la blockchain vulnerabile a problemi come il *double-spending* [10, 31]. Tipicamente infatti, i conflitti tra le diverse versioni della catena si risolvono entro una certa finestra temporale, attraverso la selezione e la conseguente estensione da parte dei nodi della rete di un solo ramo formatosi della catena di blocchi [32]. La selezione del ramo avviene secondo criteri tipici della particolare tecnologia blockchain. Ad esempio, in Bitcoin, ogni nodo deve sempre selezionare ed estendere il ramo che porta alla definizione della catena di blocchi per cui è stato compiuto un lavoro maggiore; tale criterio è noto come il criterio della catena più lunga [24]. In questo scenario, la consistenza si dice finale (o *eventual consistency*) in quanto, con il passare del tempo e con il verificarsi di aggiornamenti, tutte le repliche del registro convergeranno verso lo stato del registro definito nella catena più lunga della blockchain. Se una blockchain non è in grado di risolvere possibili eventi di fork, la sicurezza della blockchain è compromessa [10, 32, 60].

I protocolli di consenso in grado di preservare, a ogni istante di tempo, la proprietà di *safety* garantiscono l'assenza di *fork* accidentali anche durante periodi di rete avversi. Questa proprietà viene assicurata al costo di periodi di stallo della rete [33, 34]. Al contrario, esistono protocolli di consenso che privilegiano *liveness,* e quindi evitano periodi di stallo adottando, contemporaneamente, modelli di consistenza finale*,* al costo di possibili eventi di *fork* accidentali [16, 31, 35].

Le fork intenzionali, invece, si verificano in occasione di aggiornamenti pianificati del protocollo (software) della blockchain. Nello specifico, il processo di aggiornamento prevede che i nodi della rete, in maniera asincrona e in autonomia, aggiornino il software e ciò può causare una fork della rete (ad esempio, in caso di modifica di un parametro di protocollo o della struttura dei blocchi). In questo caso si distinguono due tipologie di fork, quali *soft fork* e *hard fork* [36]. Nel caso di *soft fork*, l'aggiornamento di protocollo non rompe la retro-compatibilità (pertanto non viene considerato un aggiornamento obbligatorio); i nodi della rete continuano a interagire tra essi, anche con versioni differenti di protocollo. Nel caso di *hard fork*, invece, l'aggiornamento del protocollo rompe la retro-compatibilità (quindi l'aggiornamento viene considerato obbligatorio); i nodi che rifiutano di aggiornare alla nuova versione non potranno interagire con il resto della rete, ma continueranno a operare su un *ledger* obsoleto.

3.2.1.5 *Quantum Resistance*

I computer quantici saranno in grado di risolvere problemi di calcolo estremamente complessi in un tempo drasticamente inferiore rispetto ai computer tradizionali. Questo rappresenta una minaccia per la crittografia moderna e per la sicurezza delle blockchain. Ad esempio, un computer quantistico potrebbe essere usato per ricavare la chiave privata di un utente o addirittura forgiare blocchi a suo piacimento per manipolare il *ledger* condiviso.

Gli schemi di crittografia asimmetrica odierni, utilizzati per il calcolo delle firme digitali, non riescono a garantire sufficiente sicurezza per resistere a eventuali attacchi provenienti da computer quantistici. Tuttavia, ad oggi, il NIST sta lavorando a nuovi standard per la definizione di algoritmi crittografici post-quantum, che consentiranno di definire firme digitali resistenti ad attacchi provenienti da computer quantistici [65]. La maggior parte delle proposte attualmente in corso di valutazione da parte del NIST si basano su reticoli, un'astrazione matematica che sembrerebbe essere in grado di definire problemi computazionalmente complessi anche per i computer quantistici [39].

I computer quantistici sono una tecnologia pionieristica e lontana dall'essere utilizzabile su larga scala; non rappresentano una minaccia reale al tempo di scrittura di questo documento. Tuttavia, la *quantum resistance* è un parametro utile a valutare la robustezza e la sicurezza di una piattaforma blockchain a largo spettro e nel medio-lungo periodo[60].

### 3.2.2 Parametri di scalabilità

Nella teoria dei sistemi distribuiti, la scalabilità di un sistema indica la capacità del sistema di mantenere un adeguato livello di performance all'aumentare della complessità della rete (numero di nodi) o del carico di lavoro (numero di operazioni richieste) [40]. Diverse sono le metriche che possono essere utilizzate per misurare le performance di un sistema; tra queste, si considerano il *throughput* e la *latenza* [41].

Il throughput misura il numero di operazioni completate dal sistema in un determinato periodo di tempo. In particolare, per le blockchain, il throughput è misurato come il numero massimo di transazioni al secondo (TPS) che la blockchain è in grado di confermare[61].

La latenza misura l'intervallo di tempo richiesto dal sistema per completare un'operazione. Per le blockchain, la latenza si misura come il tempo medio richiesto dal protocollo per processare e finalizzare le transazioni. Solitamente questo viene misurato come la differenza tra il tempo in cui una transazione è considerata finalizzata e il tempo in cui la transazione era stata inviata da un utente al nodo *client* (c.d. "*time to finality*").

### *3.2.3* Parametri di decentralizzazione

Un sistema decentralizzato viene definito in letteratura come un sistema in cui i nodi della rete non sono gestiti da un'unica autorità ma da molteplici soggetti indipendenti [15].

[60] Si parla spesso di "*crypto agility*" come di un possibile approccio di mitigazione alle minacce rappresentate dal quantum computing. Si indica con tale espressione la capacità di un sistema informatico di implementare metodi di crittografia alternativi, rendendoli così in grado di reagire rapidamente alle eventuali minacce crittografiche. In tal senso, una blockchain "*crypto-agile*" dovrebbe rendere possibile la modifica degli algoritmi di firma e/o di hash in maniera agevole, delineando anche un percorso di change management che consente di gestire periodi transitori in cui differenti categorie di algoritmi coesistono all'interno del medesimo protocollo.

[61] Una transazione si considera confermata quando 'viene raggiunta finalità', cioè la proprietà che ne garantisce immutabilità e irreversibilità sul ledger condiviso. Si rinvia al paragrafo 3.5 per un approfondimento sulla finalità.

Il numero di nodi validatori che partecipano al consenso concorre a determinare il livello di decentralizzazione della rete. Le reti *permissioned* hanno tipicamente un livello di decentralizzazione inferiore rispetto a quelle *permissionless* perché utilizzano un insieme di validatori predefinito e limitato. In una rete permissionless invece, tutti i nodi della rete possono ricoprire il ruolo di validatore, aumentando così il livello di decentralizzazione della rete [8, 15].

Come descritto nello *"Scalability Trilemma"* [25], garantire la scalabilità e la decentralizzazione in una rete blockchain è complicato, in quanto aumentando il numero di validatori diventa più complesso raggiungere il consenso e garantire tolleranza a guasti Bizantini simultaneamente [10, 18, 33].

Nel consenso *PoW* il consenso viene raggiunto in maniera probabilistica, utilizzando il potere computazionale messo a disposizione dai nodi della rete per la risoluzione di un *puzzle* matematico complesso; i validatori con più potere computazionale hanno più possibilità di essere selezionati come proponenti del prossimo blocco [31, 42]. Altri sistemi, basati su PoS, richiedono ai validatori un deposito in denaro (*stake*) per poter essere selezionati per proporre un blocco. In questo caso, l'elezione del proponente del blocco può avvenire in due modalità:

1. approccio deterministico: sceglie il validatore proponente in base alla proporzione di *stake* depositato;
2. approccio probabilistico: sceglie il validatore proponente con un approccio *random* basato sullo *stake*.

In entrambe le soluzioni, lo *stake* depositato gioca un ruolo fondamentale in quanto aumenta le possibilità di essere selezionato come validatore proponente. Il grado di decentralizzazione di un sistema blockchain viene quindi valutato secondo due dimensioni [8]: (i) il numero di nodi validatori che partecipano al consenso, (ii) la distribuzione del potere di validazione, cioè la distribuzione delle risorse (ad esempio, computazionali come in PoW). La dimensione (ii) è rilevante nell'ambito dei protocolli del tipo *lottery-based* di cui PoW e PoS fanno parte. Nel proseguo della trattazione, si assume tale punto di vista.

3.2.3.1 *Numero di nodi validatori*

In reti blockchain permissioned il numero di nodi validatori è un insieme predefinito di nodi, mentre nelle reti permissionless tutti i nodi devono poter aggiornare il *ledger* condiviso partecipando (esplicitamente o meno) al protocollo di consenso. In PoW sono considerati validatori quei nodi che partecipano al processo di *mining*, cioè di soluzione del puzzle crittografico. In PoS, i validatori sono tutti quei nodi che sono registrati per partecipare al consenso. Quanti più validatori ha una rete blockchain tanto più il sistema può considerarsi decentralizzato.

3.2.3.2 *Distribuzione del potere di validazione*

È fondamentale individuare all'interno del sistema come il potere di validazione viene distribuito tra nodi validatori. In PoW, per esempio, i *miners* con maggiore potenza computazionale hanno maggiori possibilità di risolvere il puzzle matematico e quindi maggiore possibilità di essere proponenti di un blocco. Per questo motivo, la PoW è condizionata da un meccanismo detto *pooling* in cui più *miners* tendono a unire la propria potenza computazionale per aumentare le possibilità di produrre nuovi blocchi. Questo fenomeno rende la rete più centralizzata, affidando la produzione dei blocchi nelle mani di pochi *miners* ed esponendo la rete all'attacco del 51%, in cui la maggioranza del potere computazionale della rete si trova nelle mani di un gruppo di miners [43].

Un processo simile può avvenire con la PoS, in cui i validatori "stakeholders" possono unire il proprio stake per aumentare le possibilità di produrre blocchi - fenomeno del *pooled staking* [6]. In PoS alcune

regole di penalizzazione possono essere definite, in cui parte dello stake depositato dal validatore (o l'intero importo) viene detratto in caso di malfunzionamenti o di comportamento malevolo. Questo meccanismo è noto come *slashing*. È quindi importante per una *staking pool* minimizzare il rischio di *slashing*: offrire un servizio efficiente e affidabile di validazione in modo da attirare più stake e quindi massimizzare le possibilità di produrre blocchi. Come nel mining, il fenomeno del *pooling* introduce rischi di centralizzazione: avendo lo stake distribuito su pochi gruppi di validatori, aumentano i rischi di sicurezza come l'attacco del 51%.

**FOCUS 6 – I sistemi PoS**

Non tutti i sistemi PoS sono uguali e in alcune implementazioni, il rischio di attacco al 51% dello stake viene mitigato. In particolare, blockchain basate su PoS possono adottare diversi approcci di elezione:

Bonded Proof of Stake: esiste un insieme di nodi che possono essere "eletti" come validatori; gli stakeholders depositano il proprio *stake* nel protocollo per acquisire diritto di voto e indicare uno o più validatori; i nodi con più voti vengono eletti come validatori e quindi autorizzati a produrre nuovi blocchi; la distribuzione dello stake viene concentrata su un gruppo di validatori che è disposto a depositare lo stake richiesto;

Delegated Proof of Stake: i validatori vengono eletti tramite un sistema di votazione basato sullo stake; gli stakeholders esprimono la propria preferenza delegando lo stake presso un validatore; i validatori con più stake delegato vengono eletti; solitamente viene scelto un numero relativamente contenuto di validatori per mantenere un *tradeoff* equilibrato tra sicurezza e scalabilità; la distribuzione dello stake è limitata al numero di validatori eletti [60];

Pure Proof of Stake: ogni stakeholder ha l'autorità di partecipare al consenso come validatore e quindi essere eletto come proponente di un nuovo blocco; i validatori vengono eletti in maniera casuale tramite l'utilizzo di un meccanismo crittografico che non richiede elevati costi computazionali. Questa tipologia di PoS non prevede meccanismi di aggregazione dello stake per incrementare le chance di essere eletti come proponenti di un blocco, e non pone barriere economiche o computazionali per partecipare al consenso; la distribuzione dello stake è ampia sulla totalità degli stakeholder [60].

3.2.3.3 *Fairness*

In molte piattaforme blockchain esistono barriere di accesso per i validatori che ne potrebbero limitare la *fairness*. Ad esempio, in sistemi basati su PoW, la barriera di accesso è rappresentata dal costo dell'hardware e dell'energia richiesti per partecipare al consenso. In alcuni protocolli PoS, invece, le barriere vengono rappresentate dall'ammontare minimo di *stake* da depositare, affinché un nodo possa essere considerato un validatore. Minori sono le barriere di accesso, maggiore è la possibilità di avere una rete altamente decentralizzata.

### 3.3. Flessibilità

Nelle sezioni precedenti abbiamo presentato le caratteristiche di infrastruttura e i parametri di sicurezza, scalabilità e decentralizzazione che differenziano le blockchain. In questa sezione analizziamo le loro differenze, da un punto di vista applicativo e di usabilità: applicativo in termini di possibilità di costruire applicazioni sulla blockchain, e usabilità in termini di facilità di utilizzo e

interazione della blockchain con altri sistemi. In particolare, identifichiamo con la flessibilità di una blockchain (i) la programmabilità, ovvero la capacità di poter implementare (ed eseguire) applicazioni decentralizzate (tramite lo sviluppo di smart contracts), (ii) la possibilità di configurare la rete in base a regole "custom", e infine (iii) le tecniche di interoperabilità disponibili per facilitare l'interazione tra diversi sistemi.

### *3.3.1 Programmabilità*

La programmabilità è intesa come la capacità di una blockchain di interpretare ed eseguire in maniera decentralizzata programmi (*smart contract*). L'esecuzione di uno smart contract è deterministica e il risultato dell'esecuzione, memorizzato in maniera immutabile sulla blockchain, dipende dai parametri dati in input e dallo stato della blockchain. Gli smart contract permettono la definizione di logiche di business per l'approvazione di transazioni o per l'aggiornamento di stato del ledger stesso, e vengono adottati per la realizzazione di applicazioni decentralizzate. Nella sezione II gli smart contract vengono trattati in maggiore dettaglio.

### *3.3.2 Configurabilità*

Ogni sistema blockchain definisce un insieme di parametri che ne impattano il funzionamento in termini di efficienza, usabilità e sicurezza. Ad esempio, può essere necessario per un nodo validatore poter determinare lo spazio di memoria messo a disposizione per gestire le code di transazioni ricevute in ingresso (ad esempio in caso di memoria limitata del nodo), o addirittura configurare parametri di protocollo come la dimensione dei blocchi prodotti, e la frequenza. In questo caso, la flessibilità viene impattata, e bisogna distinguere i tradeoff tra blockchain che forniscono un ambiente di esecuzione configurabile, e quindi maggiormente flessibile, rispetto a sistemi che non consentono configurazioni "custom" del nodo, in favore di un sistema più stabile e consolidato [8].

### *3.3.3 Interoperabilità*

Numerose piattaforme blockchain si stanno affermando e stanno acquisendo un'importante presenza sul mercato, attirando utenti e nuove applicazioni decentralizzate. Diventa determinante abilitare meccanismi di interoperabilità che consentano agli utilizzatori di poter operare su diverse piattaforme blockchain (ad esempio inviare asset da una blockchain A ad una blockchain B, o invocare uno smart contract di una blockchain C) [49].

**FOCUS 7 - Interoperabilità e Bridges**

Ad oggi, l'obiettivo di interoperabilità è solitamente garantito tramite l'utilizzo di intermediari (detti *notary*) che operano scambi atomici *cross-chain*. Gli intermediari sono rappresentati tipicamente da smart contract, o coppie di smart contract definite sulle blockchain interessate [50]. Queste parti agiscono come degli oracoli centralizzati o semi-centralizzati e vengono comunemente denominati "*Bridges*". Essendo le blockchain usualmente tra loro incompatibili[62], scambiare un asset o un'informazione da una piattaforma all'altra risulta problema di non semplice soluzione. Per abilitare questo tipo di operazioni i Bridges si incaricano di "eliminare" (*burn*) un'informazione dalla blockchain mittente e di "creare" (*mint*) la stessa informazione nella blockchain di destinazione. Solitamente questo processo avviene tramite l'utilizzo di smart contract per le operazioni di *burning* e *minting*, e di una componente software (*middleware*) tra le due blockchain per certificare la correttezza del processo stesso. Data la loro complessità, i Bridge rappresentano uno dei punti maggiormente critici per la sicurezza degli scambi cross-chain: sia gli smart contract del Bridge, sia la componente *middleware*, possono presentare vulnerabilità ed essere compromessi. Nel 2022 i Bridge sono stati tra le principali vittime di attacchi alle blockchain che hanno prodotto il furto di centinaia di milioni di dollari [51].

Una alternativa ai Bridge *notary* sono le soluzioni di interoperabilità basate su relay, cioè sistemi eseguiti direttamente all'interno di una blockchain, in grado di validarne lo stato tramite la verifica delle firme dei validatori, e comunicare quest'ultimo ad una seconda blockchain [52]. Meccanismi basati su relay sono convenienti in reti estremamente efficienti, ma la verifica può diventare un collo di bottiglia nel caso in cui questa coinvolga una blockchain con tempi di finalità più lunghi rispetto a un'altra. Questo risulta un limite in caso di scambi di mercato cross-chain dove il tempo di validazione delle transazioni è un elemento fondamentale [50]. Tuttavia, esistono in letteratura soluzioni alternative che mitigano il problema della verifica delle firme di validazione di una blockchain tramite l'utilizzo di certificati crittografati compatti [53]. Questi rappresentano degli oggetti crittografici facilmente esportabili da una blockchain ad un'altra, e che permettono di verificare lo stato direttamente sulla blockchain (ad esempio tramite uno smart contract che interpreti il certificato) in maniera efficiente e senza doversi affidare a bridges esterni.

### 3.4 Impatto Energetico

Le blockchain, come ogni infrastruttura informatica, richiedono un determinato consumo energetico per operare in maniera efficiente e sicura. Il sistema esegue le operazioni di processamento di transazioni ed esecuzione di smart contract, richiedendo ai nodi della rete di sincronizzarsi tramite il protocollo di consenso caratteristico della blockchain. Queste operazioni richiedono evidentemente risorse computazionali che generano un conseguente dispendio energetico.

L'associazione *Cripto Carbon Ratings Institute* ha definito un modello per la valutazione dei consumi energetici delle blockchain. Dallo studio emerge come reti basate su protocollo PoW (e.g. Bitcoin) che consumano in media 120000 [Gwh/anno], richiedano circa 2000 [kWh] per la validazione di una transazione[63]. Per sua natura, infatti, il protocollo di consenso PoW richiede di dimostrare la risoluzione di un problema computazionalmente oneroso, che si traduce in notevole consumo

[62] Esistono blockchain che condividono protocolli nativi e rendono l'interoperabilità cross-chain più agevole; si tratta tuttavia al momento di eccezioni nel panorama del mercato. Un esempio in tal senso è rappresentato da Polkadot, una blockchain multichain nativa che consente di effettuare scambi cross-chain di dati e asset tra le chain "specializzate" che costituiscono l'ecosistema Polkadot.

[63] 2000 kWh rappresentano circa il fabbisogno energetico medio annuale di una famiglia italiana composta da 2-3 persone.

energetico. Dallo studio si evince anche che altri protocolli, in particolare quelli basati su PoS, possano essere definiti più "green" in quanto per loro natura non comportano un carico computazionale elevato da parte dei validatori per eseguire il consenso. Lo studio mostra ad esempio come una rete PoS (e.g. Ethereum) consumi circa 2,7 [Gwh/anno] [54].

## 4. Modello economico

Per garantire il corretto funzionamento delle blockchain di tipo *permissionless* è necessario l'utilizzo di un token nativo che consenta di definire incentivi finanziari affinché i validatori agiscano onestamente e mantengano la rete funzionante. Il token nativo inoltre viene utilizzato per ricoprire i costi di esecuzione delle transazioni (dette commissioni o *fees*). In altri termini, mentre per le blockchain di tipo *permissioned* il modello economico sottostante è di tipo tradizionale ed estrinseco rispetto alla blockchain[64], la natura decentralizzata delle blockchain *permissionless* richiede l'esistenza di un meccanismo di incentivazione intrinseco, che garantisca a chi accetta il ruolo di validatore una remunerazione che renda economicamente conveniente continuare a svolgerlo. Tale incentivo economico è costituito proprio dai token nativi, che remunerano le attività di validazione.

### 4.1 Distribuzione del Token nativo

La tecnica di distribuzione del token nativo (ove tale distribuzione avvenga) determina il valore iniziale del token, il ruolo e gli incentivi dei principali attori coinvolti nel progetto, le modalità scelte per continuare a incentivare la crescita dell'ecosistema nel tempo. L'allocazione iniziale dei token va valutata in rapporto ad alcune grandezze fondamentali, come il numero massimo teorico di token nativi e il numero di token effettivamente in circolazione ad un dato momento[65].

Nelle blockchain basate su PoS, l'allocazione dei token nativi determina anche i maggiori stakeholders e quindi gli individui o gli enti in grado di esercitare maggiore controllo sulla rete. Per valutare una blockchain è quindi necessario analizzare le modalità adottate per il rilascio iniziale del token, il modo in cui sono stati selezionati gli stakeholders e l'allocazione risultante. Ad esempio, una blockchain che ha affidato la maggior parte dei token a un'entità o ad un gruppo ristretto di beneficiari, si espone a maggiori rischi di concentrazione dello stake su poche entità, che possono pregiudicarne la sicurezza e l'affidabilità [55].

Le più comuni tecniche di distribuzione iniziale – di norma combinate tra loro con pesi diversi – sono[66]:

- *Allocazione agli insider*: distribuzione a titolo gratuito a beneficiari interni (es. sviluppatori) che avviene prima della distribuzione o vendita pubblica, può essere associata a vincoli che limitano la cessione dei token ricevuti per un certo periodo di tempo;
- *Fondazioni no-profit*: rilascio dei token a fondazioni incaricate di promuovere la piattaforma e incentivare nuovi progetti nell'ecosistema;
- *Vendita privata:* cessione a investitori di norma istituzionali (es. venture capital) sulla base di accordi bilaterali;
- *Airdrop*: distribuzione gratuita al pubblico (es. agli utenti che hanno interagito con la blockchain nella fase di testing);

[64] Ad esempio, un insieme di soggetti o organizzazioni realizzano il sistema per supportare i propri processi di business e accettano di sostenerne in contropartita i costi di realizzazione e gestione.
[65] In caso di totale emissione del token al momento di lancio, le due dimensioni possono coincidere.
[66] Le tecniche descritte sono usualmente utilizzate per token non-nativi. Nel caso dei token nativi, rileva anche la casistica in cui essi non vengono distribuiti affatto, in quanto assegnati esclusivamente in esito al processo di *mining/minting* (è il caso di Bitcoin).

- *Vendita pubblica*: vendita tramite aste pubbliche o ICO;

### 4.2 Capitalizzazione

La capitalizzazione indica il valore totale di mercato di tutti i token nativi che sono stati immessi in circolazione, valutati al prezzo di mercato in un dato momento. Questo parametro può ricoprire un ruolo nell'analisi di robustezza e sicurezza di una blockchain, in quanto può avere impatto sullo sforzo economico necessario a controllare la rete. Esistono tuttavia modelli di incentivi diversi, pertanto la capitalizzazione del token nativo ha effetti differenti a seconda del caso specifico.

Ad esempio, in blockchain basate su PoW, un attaccante è meno incentivato ad attaccare la rete se il valore di mercato del token risulta basso. In tal caso infatti, i costi di esecuzione dell'attacco (che non dipendono dal valore del token ma ad esempio dal costo dell'energia elettrica) potrebbero risultare superiori agli eventuali guadagni. D'altro canto, tale linea di pensiero rischia di essere semplicistica se non è accompagnata anche da riflessioni di senso opposto sull'appetibilità di una blockchain per potenziali miners; infatti, in presenza di una barriera di ingresso al "mercato del mining" costituita dagli investimenti in hardware necessari per poter partecipare al meccanismo di validazione, un alto valore del token costituisce un incentivo ad investire e quindi tendenzialmente contribuisce positivamente alla crescita del numero di validatori, fenomeno che rende l'esecuzione di attacchi più difficoltosa. In caso di meccanismi di consenso PoS, un attaccante potrebbe invece giovarsi di bassi prezzi di mercato del token per accrescere la propria *stake* e quindi acquisire potere decisionale sulla rete, consentendogli di agire in maniera malevola senza subire elevate penalizzazioni economiche (ad esempio in caso di *slashing*) [43].

### 4.3 Costi di Transazione

Le piattaforme blockchain richiedono agli utilizzatori il pagamento di costi di esecuzione delle transazioni essenzialmente al fine di riconoscere fondi ai validatori come ricompensa per l'esecuzione del processo di validazione, contribuendo così a rendere la rete sicura (e.g. prevenendo possibili attacchi o fenomeni di *spam*).

È possibile individuare due diverse tipologie di costi transazionali:

- *Costo flat*: si applica un costo fisso alle transazioni, indipendentemente dalla tipologia (es: pagamento, esecuzione di smart contract);
- *Costo dinamico*: si applica un costo variabile che viene determinato dalla tipologia di transazione da eseguire o dalla congestione della rete; in caso di alto utilizzo della rete, i costi possono crescere esponenzialmente.

Alcune blockchain consentono inoltre agli utenti di incentivare il livello di priorità con cui richiedere la registrazione delle proprie transazioni, aggiungendo una ricompensa aggiuntiva ai validatori. Questa funzionalità, pensata per accrescere l'efficienza complessiva del sistema rendendo possibile l'esecuzione di transazioni a costi inferiori (i.e. ove l'utente non abbia particolare urgenza nella sua esecuzione), potrebbe d'altro canto rendere le piattaforme meno inclusive [29], poiché disincentiva i validatori ad approvare transazioni a bassa ricompensa.

## 5. Ecosistema e dati *on-chain*

### 5.1 Governance

La governance di una piattaforma blockchain è un altro parametro utile a valutarne la sostenibilità e la durabilità. Infatti, i sistemi blockchain, in quanto decentralizzati, possono non presentare autorità

centrali che prendono decisioni di governance rispetto, ad esempio, agli aggiornamenti del protocollo o alla gestione dei token nativi non distribuiti; è pertanto necessario individuare quali meccanismi di *governance* possano essere adottati a seconda del contesto, in modo da abilitare gli attori interessati a poter partecipare al processo decisionale.

In blockchain *permissioned*, dove i partecipanti possono operare solo a seguito di specifiche autorizzazioni e con ruoli ben definiti, risultano più agevolmente applicabili meccanismi di governance tradizionali, ovvero centralizzata, ad esempio delegandola direttamente al consorzio che gestisce l'infrastruttura. Sono infatti gli enti controllanti che hanno il potere di modificare il software dei nodi della rete, o prendere decisioni sui permessi e i ruoli dei partecipanti. Al contrario, in una blockchain *permissionless*, non essendo possibile individuare puntualmente soggetti responsabili per l'erogazione dei servizi del sistema nel suo complesso, a causa della natura eminentemente decentralizzata dello stesso, possono essere applicati modelli di *governance* differenti.

In linea generale l'ideazione del protocollo, lo sviluppo dell'infrastruttura e dei *tool* per il suo utilizzo richiedono necessariamente il coinvolgimento di un gruppo di persone. Il team di sviluppo in alcuni casi si incarica della manutenzione del protocollo e della gestione degli aggiornamenti. Questo potrebbe generare dubbi circa il grado di effettiva decentralizzazione del sistema stesso. In un contesto *permissionless* il team deve necessariamente condividere le scelte in merito alla gestione della rete con una comunità di utilizzatori e collaboratori esterni, ad esempio per eventuali decisioni sull'*upgrade* del protocollo.

A questi meccanismi di *governance*, che possiamo definire "*off-chain*" e che talvolta si avvalgono anche di meccanismi di coordinamento non convenzionali (ad esempio, forum), si affiancano poi i protocolli informatici nativi della piattaforma, che regolano ad esempio il processo di validazione delle transazioni e di esecuzione degli smart contract; tali meccanismi, assieme ad altri più sofisticati (ad esempio, basati su smart contracts e/o sui c.d. "*governance token*"), sono definiti "*on-chain*". Essi si prefiggono l'obiettivo di automatizzare anche i processi con cui i partecipanti cooperano per prendere decisioni, ad esempio sugli aggiornamenti del protocollo e la definizione di nuove funzionalità, ma anche sulle politiche di incentivazione dell'ecosistema o sull'organizzazione interna (es. definizione di nuovi team di sviluppo o gestionali) [57, 58, 66, 67, 68, 69, 70, 71, 72, 73].

### 5.2 Utilizzo della piattaforma

Un altro parametro utile per la caratterizzazione di una blockchain è l'utilizzo effettivo della piattaforma. Un sistema scarsamente utilizzato difficilmente potrà acquisire valore. Un parametro per valutare l'utilizzo di una piattaforma è il volume di transazioni eseguite on-chain. Se il TPS *massimo* identifica una metrica fondamentale per misurare la scalabilità del sistema, il TPS *effettivo* è un indicatore per misurare quanto la piattaforma blockchain viene effettivamente utilizzata ad ogni istante di tempo. Il TPS effettivo viene determinato da due fattori; il numero di transazioni confermate all'interno dei blocchi del ledger durante un determinato periodo di tempo e il TPS *massimo*.

## 6. Applicazione della tassonomia

In questa sezione viene proposto un possibile processo di analisi di una piattaforma blockchain rispetto ai parametri presentati nelle sezioni precedenti. In particolare, il diagramma sottostante (Diagramma 1) presenta un approccio metodologico per l'acquisizione delle informazioni necessarie a mappare i parametri qualitativi con la soluzione blockchain in analisi. Quindi, data una blockchain, si procede eseguendo tre fasi di raccolta dati e l'analisi dei risultati.

La prima fase richiede la valutazione degli indicatori qualitativi tecnici:

1. valutare pubblicazioni *peer-reviewed*. Queste devono essere considerate come fonte principale di informazione per i dati relativi alle caratteristiche tecniche della piattaforma;
2. integrare le informazioni tecniche con il *white paper* della piattaforma (se esistente);
3. in caso di valutazioni tecniche mancanti (ad esempio, non si hanno informazioni relative alla tipologia e funzionamento dello specifico meccanismo di consenso), consultare la documentazione *on-line* della piattaforma.

La seconda fase è caratterizzata da un'analisi economica rispetto alla *tokenomics* del progetto e ai dati relativi al mercato di riferimento. Queste informazioni possono essere acquisite tramite data providers specializzati; sebbene nel contesto attuale non sia sempre facile valutare la credibilità e affidabilità delle fonti.

Infine, la terza e ultima fase della metodologia consiste nella raccolta dei dati sull'ecosistema e sull'utilizzo diretto *on-chain*.

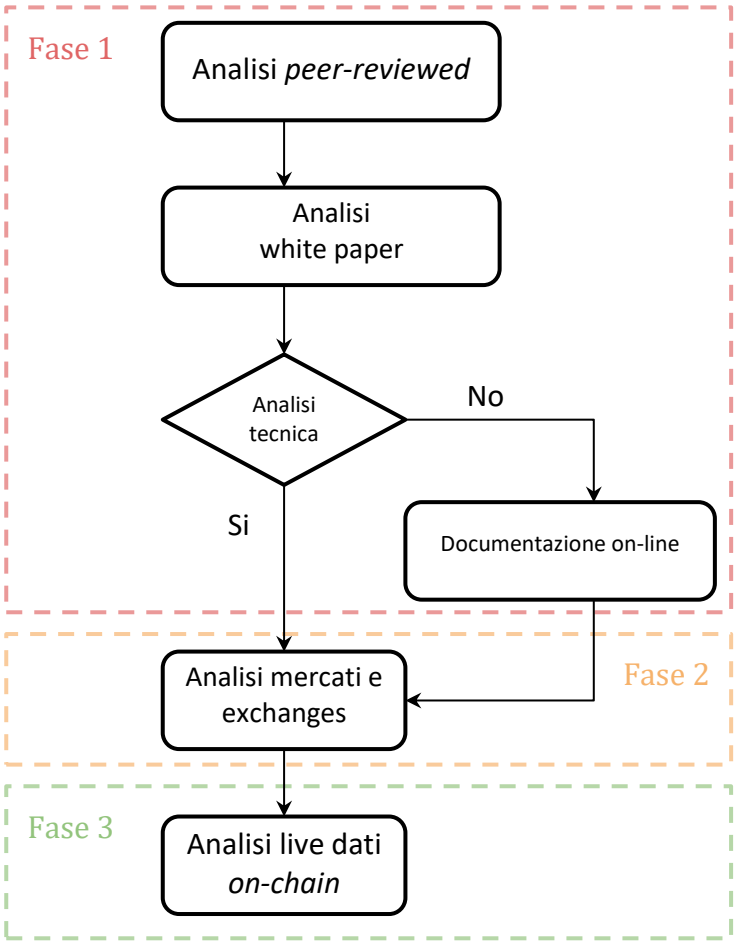


Diagramma 1. Applicazione del processo di analisi ad una nuova blockchain. Fase di acquisizione dei dati.

### 6.1 Conclusioni

Lo sviluppo di applicazioni basate su blockchain (o su smart contract) è fortemente condizionato dalle caratteristiche della specifica tecnologia utilizzata. A tale scopo, in questo documento, partendo da un'analisi dello stato dell'arte, vengono elencate e descritte le caratteristiche principali della tecnologia blockchain. Il documento propone inoltre un approccio metodologico che potrebbe essere

utilizzato per l'acquisizione delle informazioni necessarie per descrivere e analizzare le piattaforme blockchain in relazione ai parametri individuati. In tale ottica, il processo di analisi proposto può essere applicato, in via di esercizio sperimentale, ad un gruppo di blockchain per valutarne le caratteristiche fondamentali e analizzarne le similitudini e le differenze.

**Bibliografia**

[1] Italian Blockchain Service Infrastructure (IBSI). URL: https://progettoibsi.org/; Accesso: 30/01/2023
[2] ABI Lab, "Spunta Banca DLT", (2020). URL: https://www.abilab.it/aree-ricerca/blockchain-dlt/spunta-banca-dlt; Accesso: 30/01/2023
[3] European Blockchain Service Infrastructure (EBSI). URL: https://ec.europa.eu/digital-building-blocks/wikis/display/EBSI/Home; Accesso: 30/01/2023
[4] World Economic Forum, "Blockchain Toolkit - Legal and Regulatory Compliance". URL: https://widgets.weforum.org/blockchain-toolkit/legal-and-regulatory-compliance/index.html; Accesso: 20/12/2022
[5] Bassan, F. *"*Digital Platforms and Global Law*"*, Edward Elgar Publishing (2021)
[6] Bassan, F. "Digital Platforms and Blockchains: the AGE of Participated Regulation", European Business Law Review, 2022
[7] Bassan, F. *"*Potere dell'algoritmo e resistenza dei mercati in Italia: la sovranità perduta sui servizi*"*, Rubbettino editore, 2018
[8] Kannengießer N. *et al.*, *"*Trade-offs between Distributed Ledger Technology Characteristics*"*, ACM Comput. Surv. 53.2, (May 2020)
[9] Mingxiao, D. *et al.*, *"*A Review on Consensus Algorithm of Blockchain*", IEEE International Conference on Systems, Man, and Cybernetics (SMC), Banff, AB, Canada, 2017, pp. 2567-2572, doi: 10.1109/SMC.2017.8123011*
[10] Cachin, C. and Vukolic, M. *"*Blockchain Consensus Protocols in the Wild*"*, CoRR, 2017
[11] Lamport, L. "Proving the Correctness of Multiprocess Programs, IEEE Transactions on Software Engineering SE-3.2, 1977, pp. 125–143
[12] Dinh, A. *et al.*, "BLOCKBENCH: A Framework for Analyzing Private Blockchains", SIGMOD. ACM. 2017, pp. 1085–1100
[13] Schäffer, M., Di Angelo M. and Salzer, G. "Performance and Scalability of Private Ethereum Blockchains", Business Process Management: Blockchain and Central and Eastern Europe Forum. BPM, 2019. Lecture Notes in Business Information Processing, vol 361. Springer, Cham. https://doi.org/10.1007/978-3-030-30429-4_8
[14] Mazzoni, M., Corradi A. and Di Nicola, V. "Performance evaluation of permissioned blockchains for financial applications: The ConsenSys Quorum case study", Blockchain: Research and Applications, 2021
[15] Troncoso C. *et al.*, "Systematizing Decentralization and Privacy: Lessons from 15 Years of Research and Deployments", Proceedings on Privacy Enhancing Technologies, 2017
[16] Pass, R. Seeman L, and Shelat, A. "Analysis of the Blockchain Protocol in Asynchronous Networks", Coron, JS., Nielsen, J. (eds) Advances in Cryptology – EUROCRYPT 2017. EUROCRYPT 2017. Lecture Notes in Computer Science, vol 10211. Springer, Cham. https://doi.org/10.1007/978-3-319-56614-6_22
[17] Reis Furtado F. et al., "Towards characterising architecture and performance in blockchain: a survey", International Journal of Blockchains and Cryptocurrencies, 2020 Vol.1 No.2, pp.121 - 153
[18] The Performance of Byzantine Fault Tolerant Blockchains". In: 2020 IEEE 19th International Symposium on Network Computing and Applications (NCA), IEEE 19th International Symposium on Network Computing and Applications (NCA), Cambridge, MA, USA, 2020, pp. 1-8, doi: 10.1109/NCA51143.2020.9306742
[19] Sorensen, D. "Establishing Standards for Consensus on Blockchains", Blockchain Second International Conference
[20] Dwork, C., Lynch N. and Stockmeyer, L. "Consensus in the presence of partial synchrony", Journal of the ACM, 1988

[21] BitFury Group and Garzik, J. "Public versus Private Blockchains Part 1: Permissioned Blockchains", URL: https://bitfury.com/content/downloads/public-vs-private-pt1-1.pdf
[22] BitFury Group and Garzik, J. "Public versus Private Blockchains Part 2: Permissionless Blockchains", URL: https://bitfury.com/content/downloads/public-vs-private-pt2-1.pdf
[23] Lorne, L. and Cawrey, D. "Mastering Blockchain". O'Reilly Media, 2020
[24] Bashir, I. "Mastering blockchain". Packt Publishing Ltd, 2017
[25] Buterin, V. "The Scalability Trilemma" URL: https://vitalik.ca/general/2021/04/07/sharding.html; Accesso: 20/12/2022
[26] Avizienis A. et al., "Basic Concepts and Taxonomy of Dependable and Secure Computing". In: IEEE Trans. Dependable Secur. Comput
[27] Cachin, C., Guerraoui R. and Rodrigues, L. "Introduction to Reliable and Secure Distributed Programming", Springer, 2011, XIX, 320 pages
[28] Gilbert S. and Lynch, N. "Brewer's Conjecture and the Feasibility of Consistent, Available, Partition-tolerant Web Services", Available, Partition-tolerant Web Services" SIGACT News 33, 2 (June 2002), 51–59. https://doi.org/10.1145/564585.564601
[29] De Angelis, S. *et al.*, "Evaluating Blockchain Systems: A Comprehensive Study of Security and Dependability Attributes" Proceedings http://ceur-ws. org ISSN, (2022)
[30] Van Steen, Maarten, and A. Tanenbaum. "Distributed systems principles and paradigms." Network 2 (2002), URL: https://www.distributed-systems.net/index.php/books/ds3/
[31] Vukolić, M. "The Quest for Scalable Blockchain Fabric: Proof-of-Work vs. BFTReplication". In: Open Problems in Network Security
[32] Vukolic, M. "Eventually Returning to Strong Consistency", IEEE Data Eng. Bull. 39, 2016
[33] Castro M. and Liskov, B. "Practical Byzantine Fault Tolerance", Proceedings of the Third Symposium on Operating Systems Design and Implementation
[34] Gilad, Y., Hemo, R., Micali, S., Vlachos, G. and Zeldovich, N. "Algorand: Scaling Byzantine Agreements for Cryptocurrencies". In Proceedings of the 26th Symposium on Operating Systems Principles (SOSP '17). Association for Computing Machinery, New York, NY, USA, 51–68. https://doi.org/10.1145/3132747.3132757
[35] Buterin, V. & Griffith, V. "Casper the Friendly Finality Gadget", 2017
[36] Schär, F. "Blockchain Forks: A Formal Classification Framework and Persistency Analysis", The Singapore Economic Review, 2020
[37] European Parliamentary Research Service, "Can distributed ledgers be squared with European data protection law?". Blockchain and the General Data Protection Regulation
[38] Goldwasser, S., Micali, S. and Rackoff, C. "The knowledge complexity of interactive proof-systems". In Proceedings of the seventeenth annual ACM symposium on Theory of computing (STOC '85). Association for Computing Machinery, New York, NY, USA, 291–304. https://doi.org/10.1145/22145.22178
[39] Micciancio D. & Regev, O. "Lattice-based Cryptography", Bernstein, D.J., Buchmann, J., Dahmen, E. (eds) Post-Quantum Cryptography. Springer, Berlin, Heidelberg. https://doi.org/10.1007/978-3-540-88702-7_5
[40] Bondi, A. B. "Characteristics of Scalability and Their Impact on Performance", Proceedings of the 2Nd International Workshop on Software and Performance
[41] Braddock, R. L., Claunch M. R and Walter Rainbolt, J. "Operational Performance Metrics in a Distributed System. Part II.: Metrics and Interpretation", Proceedings of the 1992 ACM/SIGAPP Symposium on Applied Computing: Technological Challenges of the 1990's
[42] Nakamoto, S. "Bitcoin: A peer-to-peer electronic cash system". 2008. URL: https://bitcoin.org/bitcoin.pdf.
[43] Eyal, I. and Emin Gün Sirer. 2018. Majority is not enough: bitcoin mining is vulnerable. Commun. ACM 61, 7 (July 2018), 95–102. https://doi.org/10.1145/3212998
[44] Micali, S., Vadhan, S. and Rabin, M. "Verifiable Random Functions", Proceedings of the 40th Annual Symposium on Foundations of Computer Science (FOCS '99). IEEE Computer Society, USA, 120.
[45] Marc, J. & Farouk, H. & Ramy, G. & Ziyaad, Q. "Do Smart Contract Languages Need to be Turing Complete?", Prieto, J., Das, A., Ferretti, S., Pinto, A., Corchado, J. (eds) Blockchain and Applications. BLOCKCHAIN 2019. Advances in Intelligent Systems and Computing, vol 1010. Springer, Cham. https://doi.org/10.1007/978-3-030-23813-1_3

[46] Zheng, Z., Xie, S., Dai, H.N., Chen, W., Chen, X., Weng, J. Imran, M. "An overview on smart contracts: Challenges, advances and platforms" FGCS, 2020
[47] Atzei, N., Bartoletti, M., Cimoli, T. (2017). A Survey of Attacks on Ethereum Smart Contracts (SoK). In: Maffei, M., Ryan, M. (eds) Principles of Security and Trust. POST 2017. Lecture Notes in Computer Science, vol 10204
[48] Destefanis, G., Marchesi, M., Ortu, M., Tonelli, R., Bracciali A. and Hierons, R. "Smart contracts vulnerabilities: a call for blockchain software engineering?" 2018 International Workshop on Blockchain Oriented Software Engineering (IWBOSE), Campobasso, Italy, 2018, pp. 19-25, doi: 10.1109/IWBOSE.2018.8327567
[49] Johnson, S., Robinson, P., and Brainard, J. "Sidechains and interoperability", eprint arXiv:1903.04077, 2019
[50] Buterin, V. "Chain Interoperability" (2016), URL: https://allquantor.at/blockchainbib/pdf/vitalik2016chain.pdf
[51] Chainalysis, "2022 Crypto Crime Report.". URL: https://go.chainalysis.com/2023-crypto-crime-report.htmAccesso: 02/03/2023
[52] Goes, C. "The Interblockchain Communication Protocol: An Overview" (2020), eprint arXiv:2006.15918
[53] Micali, S., Reyzin, L., Vlachos, G., Wahby, R. S. and Zeldovich, N. "Compact Certificates of Collective Knowledge" (2021), IEEE Symposium on Security and Privacy (SP), San Francisco, CA, USA, 2021, pp. 626-641, doi: 10.1109/SP40001.2021.00096
[[54] CCRI, "Crypto Sustainability Indices". URL:https://indices.carbon-ratings.com/
[55] Bucko, J. & Palová, D. & Vejačka, M., "Security and Trust in Cryptocurrencies", Central European Conference in Finance and Economics, 2015
[56] Sayeed, Sarwar & Marco-Gisbert, Hector, "Assessing Blockchain Consensus and Security Mechanisms against the 51% Attack". Applied Sciences. 9, (2019)
[57] Reijers, W., O'Brolcháin, F., & Haynes, P. "Governance in Blockchain Technologies & Social Contract Theories. Ledger", 1, 134–151. https://doi.org/10.5195/ledger.2016.62, Ledger, 1, 134–151. https://doi.org/10.5195/ledger.2016.62, 2016
[58] Atzori, M. "Blockchain Technology and Decentralized Governance: Is the State Still Necessary?" Available at SSRN: https://ssrn.com/abstract=2709713 or http://dx.doi.org/10.2139/ssrn.2709713, 2015
[59] Gilbert S. and Lynch, N. "Perspectives on the CAP Theorem," in *Computer*, vol. 45, no. 2, pp. 30-36, Feb. 2012, doi: 10.1109/MC.2011.389
[60] Xiao, Y., Zhang, N. Lou W. and Hou. Y. T. *"A Survey of Distributed Consensus Protocols for Blockchain Networks"* in IEEE Communications Surveys & Tutorials, vol. 22, no. 2, pp. 1432-1465, (2020), doi: 10.1109/COMST.2020.2969706
[61] He, Ping and Tang, Dunzhe and Wang, Jingwen. "Staking Pool Centralization in Proof-of-Stake Blockchain Network" (May 25, 2020). Available at SSRN: https://ssrn.com/abstract=3609817
[62] Feng Tian. 2016. An agri-food supply chain traceability system for China based on RFID & blockchain technology. In Proceedings of the 13th International Conference on Service Systems and Service Management. 1–6.
[63] Filip Caron. 2018. The evolving payments landscape: Technological innovation in payment systems. IT Profess. 20, 2 (2018), 53–61.
[64] Gaby G. Dagher, Jordan Mohler, Matea Milojkovic, and Praneeth Babu Marella. 2018. Ancile: Privacy-preserving framework for access control and interoperability of electronic health records using blockchain technology. Sustain. Cities Soc. 39 (2018), 283–297.
[65] NIST Computer Security Resource Center, Post-Quantum Cryptography https://csrc.nist.gov/projects/post-quantum-cryptography
[66] Accenture. 2019. Governing DLT Networks. Distributed Ledger Technology Governance for Permissioned Networks.
[67] Allen, D. W. E., Berg, C. 2020. Blockchain Governance: what we can learn from the Economics of Corporate governance, The JBBA, Volume 3, Issue 1.
[68] ASTRI. 2016. Whitepaper on Distributed Ledger Technology, Hong Kong Applied Science and Technology Research Institute and Hong Kong Monetary Authority.
[69] Hofman, D., et al. 2021. Blockchain Governance: De Facto (x)or Designed?, in Lemieux, V.L., Feng, C. (eds.), Building Decentralized Trust, Chapter 2.

[70] Liu, Y., et al. 2022. Defining Blockchain Governance Principles: A Comprehensive Framework, University of New South Wales, Australia.
[71] Naudts, E., et al. 2022. Governance in systems based on distributed ledger technology (DLT): A comparative study, AFM.
[72] van Pelt, R. et al. 2020. Defining Blockchain Governance: A Framework for Analysis and Comparison, Information Systems Management, 38:1, 21-41.
[73] Wang S. et al. 2019. Decentralized Autonomous Organizations: Concept, Model, and Applications, IEEE Transactions on Computational Social Systems, Vol. 6, n. 5.
[74] Bassan, F. "Web 3 Regulation in Transition, Competition Policy International", in TechREG CHRONICLE, 2023
[75] Bassan, F. "Digital Platforms and Blockchains: The Age of Participated Regulation", in European Business Law Review, 2022
[76] De Filippi, P., Mannan, M. e Reijers, W. "Blockchain Technology and the Rule of Code: Regulation via Governance", disponibile su SSRN: https://ssrn.com/abstract=4292265 (2022)
[77] De Filippi, P., Mannan, M., Henderson, J., Merk, T., Cossar, S. e Nabben, K., Report on blockchain technology & legitimacy, European University Institute (2022)

## Sezione II - Tassonomia delle caratteristiche tecniche degli smart contract

### 1. Introduzione

Lo scopo di questa sezione è descrivere, da un punto di vista tecnico, il fenomeno degli smart contract e di identificarne le peculiarità (si veda il box sottostante che descrive l'approccio adottato – Nota Metodologica 1). La sezione è organizzata come segue. Nel paragrafo 2 vengono introdotte la nozione tecnica di smart contract, le sue principali funzionalità e il ciclo di vita. Il paragrafo 3 presenta una classificazione delle caratteristiche fondamentali, facendo distinzione tra caratteristiche tecnologiche e caratteristiche di alto livello. Nel paragrafo 4 vengono affrontati, invece, aspetti di sicurezza degli smart contract, con un'introduzione delle principali sfide per lo sviluppo sicuro di applicazioni decentralizzate (Decentralized Application – DApp) e una prima analisi delle possibili vulnerabilità che la tecnologia introduce.

**Nota metodologica 1**

Nella sezione vengono descritte le caratteristiche tecnologiche e di alto livello degli smart contract e le loro principali funzionalità. In particolare, il contributo della sezione deriva dall'analisi della letteratura e da una ricerca selettiva dei lavori di taglio metodologico. Sono stati analizzati i lavori esistenti in letteratura che affrontano temi quali la definizione, la comparazione e la regolamentazione degli smart contract. Per fornire il modello generico presentato nel paragrafo 3 ("*Smart Contract Overview*"), sono stati esaminati lavori di sistematizzazione e *survey* sugli smart contract [1, 2, 3, 4, 5]. I lavori di analisi tecnica e benchmark [2, 5, 6, 7, 8] sono stati utilizzati per (i) fornire una sistematizzazione delle caratteristiche tecniche degli smart contract; (ii) analizzare le differenze e i tradeoff tra diversi ambienti di esecuzione; (iii) indagare i linguaggi di programmazione esistenti.

Infine, viene proposta una valutazione dei parametri di sicurezza da considerare nello sviluppo di applicazioni decentralizzate e dei relativi smart contract. Nello specifico, vengono riportate l'insieme di vulnerabilità note nel settore [8, 9, 10, 11]. Il risultato ottenuto è stato parzialmente ispirato dai lavori noti in letteratura quali la DASP - TOP 10 [12], lo "*Smart Contract Security Verification Standard*" [13] e il registro SWC [14] di vulnerabilità note.

### 2. Smart Contract Overview

In informatica, la nozione di smart contract è stata introdotta per la prima volta nel 1990 come protocollo di computazione digitale in grado di eseguire in maniera autonoma i termini di un contratto definiti in un sistema transazionale [15]. Con l'avvento della blockchain, gli smart contract sono stati oggetto di rinnovato interesse anche grazie all'aumento significativo del numero di applicazioni decentralizzate (o "DApp") sviluppate [57].

In generale, l'espressione DApp viene frequentemente utilizzata per indicare qualunque applicazione decentralizzata basata su blockchain e che include, oltre alla tecnologia blockchain sottostante, l'interfaccia utente (di solito, un'interfaccia web) e gli smart contract [19, 20]. Le DApp, essendo decentralizzate, non necessitano di un fornitore di servizi o, in generale, di una parte fidata che ne gestisca l'infrastruttura. Uno dei principali vantaggi derivanti dall'utilizzo delle applicazioni decentralizzate è quello di ottimizzare flussi di esecuzione inefficienti.

Considerando solo il livello degli smart contract, in questa sezione si fornisce una loro analisi da un punto di vista tecnologico.

## 2.1 Cosa sono gli smart contract?

In generale, gli smart contract sono dei programmi software sviluppati in uno specifico linguaggio di programmazione. Bitcoin, ad esempio, consente, attraverso l'utilizzo di un linguaggio di *scripting* Turing non completo, lo sviluppo di smart contract per gestire la trasferibilità degli *asset* [16, 17]. Ethereum [18], invece, attraverso l'astrazione computazionale dell'*Ethereum Virtual Machine* (EVM) consente l'esecuzione di smart contract sviluppati in linguaggi di programmazione Turing completi[67].

Essendo gli smart contract programmi che risiedono all'interno della blockchain e che quindi vengono eseguiti in maniera collettiva e decentralizzata dai nodi della rete [1, 18], la loro esecuzione viene validata dalla rete blockchain sottostante e la loro affidabilità è connessa anche a quella della blockchain [8, 11]. L'esecuzione di uno smart contract è deterministica e si basa esclusivamente su dati disponibili on-chain. Questo garantisce che, durante l'esecuzione distribuita del codice dello smart contract, ogni nodo della rete ottenga lo stesso risultato (o *output*) dato un set di parametri in ingresso (o *input*) e un determinato stato della blockchain[68] [21, 22]. Esempi di esecuzione di uno smart contract sono: l'approvazione condizionale di un pagamento tra due utenti (ad esempio, è possibile approvare la transazione di pagamento verso un utente beneficiario se e solo se è passato un determinato lasso temporale); lo scambio di un *asset* (ad esempio, uno smart contract che implementa un *market place* di *asset* collezionabili che possono essere scambiati tra utenti) [3, 6].

L'esecuzione di uno smart contract è tipicamente attivata tramite una transazione detta "invocante". Ad esempio, è possibile invocare attraverso una transazione uno smart contract che implementa l'operazione di *somma* passando come argomenti della transazione stessa (ovvero come parametri di *input*) gli addendi dell'operazione. Le transazioni invocanti uno smart contract vengono approvate e il codice dello smart contract eseguito soltanto se le regole del protocollo della blockchain sottostante che determinano la validità di una transazione sono soddisfatte.

Gli smart contract ereditano le proprietà di tracciabilità e immutabilità della blockchain sottostante. In particolare, il codice degli smart contract è registrato all'interno della blockchain stessa e quindi per sua natura non può essere modificato. Questa caratteristica risulta cruciale per definire modelli di sicurezza basati su applicazioni decentralizzate in cui gli utenti hanno la certezza che il codice non venga alterato. A tal proposito, la blockchain mantiene una traccia unica sia della creazione, sia di tutte le invocazioni a uno smart contract. Inoltre, anche con *pattern* computazionali complessi che ne alterano l'esecuzione e consentono di realizzare nuovi comportamenti dello smart contract stesso (ad esempio, questo avviene nel meccanismo di aggiornamento "*Proxy Pattern*" [23, 24]) è sempre possibile ricostruire la traccia dello smart contract osservando lo storico del *ledger* [8, 23].

Gli smart contract che, come anticipato, vengono scritti in uno specifico linguaggio di programmazione di alto livello (ad esempio, *Solidity* per la blockchain di Ethereum [25]) o in un linguaggio di scripting (ad esempio *Bitcoin Scripting* [16, 17] per la blockchain Bitcoin), vengono compilati in un set di istruzioni direttamente eseguibili dal modello computazionale implementato dalla specifica blockchain (c.d. *bytecode*). Il *bytecode* compilato di uno smart contract viene installato e eseguito all'interno di un'ambiente di esecuzione.

Esistono diversi ambienti di esecuzione che generalmente si distinguono in due categorie [8, 26]: (i) interpreti basati su stack di memoria (ad esempio, *Bitcoin Script interpreter* [16, 17]); (ii) interpreti basati su macchina virtuale (ad esempio, la *Ethereum Virtual Machine* - EVM [6, 18]).

[67] Capacità di eseguire codice ricorsivo e loop all'interno del programma. Si rinvia al paragrafo 4.1.4 per approfondire le differenze tra linguaggi Turing completi e Turing non completi.
[68] Per stato della blockchain si intende l'ultima versione del ledger - l'insieme di blocchi e transazioni - validata e approvata dalla maggioranza della rete blockchain ad ogni istante temporale [18].

Gli ambienti di esecuzione si differenziano per il linguaggio di programmazione che supportano, la tipologia di comandi interpretabili (ad esempio istruzioni, funzioni, *loop* computazionali, salti) e la tipologia di memoria utilizzata [2, 7, 8, 27]. Nel paragrafo 3 ("Caratteristiche fondamentali") presenteremo le diverse caratteristiche degli ambienti di esecuzione e cui seguirà una valutazione sui rispettivi tradeoff.

## 2.2 Interagire con gli smart contract

Per interagire con uno smart contract, generalmente gli utenti della blockchain inviano una transazione firmata crittograficamente con la propria chiave privata. La blockchain mantiene traccia sia della creazione sia di tutte le invocazioni a uno smart contract memorizzandole nello stato mantenuto dal *ledger*.

Esistono due modelli di stato principali su cui si basano le blockchain: *account-based* e *token-based*.

Per *account* si intende, in generale, un'entità in grado di inviare transazioni alla blockchain e dotato di un bilancio. Nel modello *account-based*, ad ogni account viene dedicato uno spazio di memoria per salvare le informazioni di stato associate all'*account* stesso (ad esempio, il saldo di *token* da esso posseduti) [6, 18]. Lo stato locale di ciascun *account* contribuisce alla definizione dello stato globale della blockchain [6, 8, 18]. Generalmente, questo modello di stato delle blockchain distingue due tipologie di account: (i) *account standard*, a cui è associata una coppia di chiavi pubblica/privata e controllati da chiunque possiede la chiave privata; (ii) *contract account* controllato dalla logica dello specifico smart contract associato [6, 18].

Alternativo al modello *account-based* è il modello *token-based*, di cui l'*Unspent Transaction Output* (UTXO, in breve) utilizzato da Bitcoin [16, 17, 28] e l'*Extended* UTXO [29] sono un particolare esempio. In questo modello, la blockchain traccia attraverso le transazioni memorizzate nel *ledger* i *token* disponibili. Di conseguenza, il bilancio di *token* di un utente può essere sparso tra centinaia di transazioni e blocchi. In un modello *token-based*, non c'è infatti un concetto di *account* e di bilancio associato; ci sono solo *token* bloccati per essere utilizzati da specifici utenti. Di conseguenza, in un determinato momento, lo stato del *ledger* è definito da tutti i *token* ancora disponibili.

In questo documento, senza pretesa di esaustività, si considerano principalmente tecnologie blockchain che adottano un modello di stato del tipo *account-based* - che sono al momento quelle maggiormente diffuse per la realizzazione di applicazioni in ambito finanziario.

Quindi, in una tecnologia blockchain avente un modello di stato del tipo *account-based*, gli smart contract sono un particolare tipo di *account*; di conseguenza, hanno un bilancio associato e possono essere invocati tramite transazioni. Tuttavia non sono controllati da un utente e vengono eseguiti dalla rete di nodi della blockchain. Ogni transazione invocante produce l'esecuzione della logica in esso contenuta e l'aggiornamento dello stato globale della blockchain. In particolare, dato un utente di uno smart contract, si possono sintetizzare gli step di esecuzione come segue [2, 31, 32]:

1. l'utente firma e invia sulla blockchain una transazione invocante specificando una funzione dello smart contract;
2. il software della blockchain valuta la validità della transazione, verifica l'esistenza dello smart contract richiesto e ne estrae i parametri di input (se specificati);
3. viene invocata la funzione dello smart contract richiesta ed eseguita la logica codificata passando i parametri di input (se specificati);
4. in seguito all'esecuzione, le transazioni risultanti e le nuove informazioni di stato vengono confermate e memorizzate nel *ledger* in accordo con le regole del protocollo della blockchain sottostante.

Questi step definiscono il ciclo di esecuzione di uno smart contract.

## 2.3 Ciclo di vita

In questo documento viene descritto un modello di stato delle blockchain di tipo account-*based*. Inoltre, vengono considerati smart contract eseguiti in un ambiente basato su macchina virtuale (Virtual Machine, VM) (cfr. paragrafo 3.1.1). In tale contesto, è possibile articolare il ciclo di vita degli smart contract in sei fasi fondamentali [2, 32]:

- Sviluppo: traduzione di requisiti funzionali di un contratto o di un processo specifico nel linguaggio di programmazione dello smart contract. Questa fase segue gli stessi principi di programmazione tradizionali, quali lo sviluppo modulare e il test delle funzioni del programma;
- Compilazione: generazione del bytecode dello smart contract per l'esecuzione su VM; il codice sorgente viene processato da un compilatore;
- Deployment: il bytecode viene validato dal software della blockchain e istanziato all'interno del ledger; vengono generati nell'ambiente di esecuzione lo smart contract account e il relativo stato; gli utenti possono inviare transazioni invocanti per eseguire la logica dello smart contract;
- Esecuzione: a seguito di un'invocazione la VM esegue la logica dello smart contract applicando gli input e generando un output univoco; alla fine dell'esecuzione, la VM aggiorna lo stato dello smart contract con il risultato dell'output;
- Aggiornamento: modifica dello smart contract a una nuova versione del bytecode compilato. Ad esempio, gli sviluppatori potrebbero decidere di modificare le condizioni di approvazione, risolvere problemi al codice/*bugs* o aggiungere nuove funzionalità; la procedura di aggiornamento può essere resa arbitrariamente complessa (es: per gli smart contract di Ethereum esistono pattern di aggiornamento che gli sviluppatori possono eseguire (es. Contract Migration, Proxy pattern etc. [33]) [23, 24]. Tuttavia, questi meccanismi possono variare a seconda della blockchain sottostante; in generale, questa procedura richiede l'implementazione di permessi di governance per evitare manipolazioni inaspettate da parte di attori non autorizzati a cambiare le logiche di esecuzione [24];
- Cancellazione: disabilitazione delle funzioni del contratto; lo smart contract risulta inaccessibile e quindi non utilizzabile. Questa procedura richiede l'implementazione di permessi per evitare manipolazioni inaspettate.

## 2.4 Aggiornamento e Governance

Nella sezione precedente, sono state evidenziate le proprietà fondamentali di immutabilità e irreversibilità degli smart contract, le quali offrono garanzie di sicurezza sia sul contenuto del contratto, sia sulle operazioni previste durante l'esecuzione. Tuttavia è possibile prevedere un aggiornamento di uno smart contract nella circostanza in cui, ad esempio, si renda necessario aggiungere, cambiare o rimuovere una clausola, o eventualmente risolvere problemi tecnici relativi a pattern di esecuzione imprevisti, o vulnerabilità nel codice. L'aggiornamento di uno smart contract comporta la modifica della logica eseguita preservando, tuttavia, lo stato del contratto. È importante chiarire che mutabilità e aggiornabilità non sono sinonimi, soprattutto nel contesto degli smart contract. Non è infatti possibile cambiare il codice: tuttavia è possibile cambiare la logica eseguita quando si invoca uno smart contract, purché vengano adottati opportuni accorgimenti.

In questa sezione, si propone un focus sulle tecniche attuabili per gestire l'aggiornamento di uno smart contract. Esistono diversi pattern di aggiornamento di uno smart contract in letteratura [23, 24], che si affidano a differenti livelli di governance, ottimizzati sulle singole blockchain [33]. Di

seguito si fornisce una descrizione, ad alto livello, dei possibili pattern di aggiornamento nell'ambito di due categorie (i) aggiornamento di singole funzioni, (ii) aggiornamento dello smart contract. Infine, vengono evidenziati i meccanismi di governance che gli attori coinvolti possono adottare per assicurare un aggiornamento sicuro, efficiente e democratico del codice.

- Aggiornamento di singole funzioni dello smart contract: pattern di aggiornamento di una parte contenuta dello smart contract. In questo scenario, possono essere modificati o aggiornati alcuni parametri dello smart contract (e.g. tramite invio di nuovi input) che di conseguenza modificano l'esecuzione dello stesso (un esempio sono gli smart contract che fanno affidamento su un parametro passato come input per influenzare l'esecuzione della business logic); in scenari più complessi l'esecuzione di intere funzioni dello smart contract viene delegata a funzioni di altri smart contract [34] (ad esempio in Ethereum questo avviene tramite i pattern 'Proxy' e 'Diamond' [33] - Modifica tramite input/modifica tramite modularità);
- Aggiornamento dello smart contract: l'intero codice dello smart contract viene indirizzato su una nuova versione; questa procedura sostituisce totalmente il codice di uno smart contract con una sua nuova versione; in piattaforme come Ethereum questo può avvenire tramite la cancellazione dello smart contract [35] e il successivo deployment di una nuova versione, oppure tramite meccanismi di proxy che gestiscono l'indirizzamento delle richieste a nuove versioni del contratto [33, 34].

L'aggiornamento di uno smart contract può cambiare drasticamente il suo funzionamento e quindi le logiche di business eseguite. Risulta quindi fondamentale gestire i permessi degli attori abilitati ad approvare tali modifiche. In tal senso, esistono differenti approcci di gestione degli attori incaricati dell'aggiornamento [34] che si posizionano su tre direttive:

- singolo amministratore: entità nota allo smart contract (identificata con l'identità a chiave pubblica blockchain) incaricata di eseguire le funzionalità di aggiornamento; tale entità può essere scritta in maniera immutabile nel codice dello smart contract o può essere esplicitata programmaticamente (ad esempio dal creatore del contratto stesso);
- gruppo di amministratori: gruppo di soggetti autorizzati ad aggiornare lo smart contract, che sono solitamente identificati da una identità blockchain "multipla" (detta multi-signature wallet [36]) ove l'aggiornamento viene approvato se e solo se tutti gli appartenenti al gruppo di amministratori appongono la loro firma sull'operazione;
- organizzazione decentralizzata: un sistema decentralizzato (solitamente basato su smart contract) in cui un insieme di partecipanti - non necessariamente amministratori - si accordano tramite una votazione democratica sull'approvazione o il rifiuto dell'aggiornamento dello smart contract.

## 3. Caratteristiche fondamentali

In questa sezione vengono descritte le caratteristiche fondamentali degli smart contract. Vengono illustrate da un lato le caratteristiche tecnologiche, mostrando altresì i tradeoff tra le alternative implementative adottate ad oggi dalle piattaforme blockchain, e dall'altro le caratteristiche di alto livello che identificano parametri generali da considerare per la valutazione degli smart contract.

### 3.1 Caratteristiche tecnologiche

#### 3.1.1 Ambiente di esecuzione

Gli smart contract vengono eseguiti in un ambiente computazionale distribuito su tutti i nodi della blockchain In via generale, le istruzioni del bytecode di uno smart contract vengono eseguite da un interprete basato su allocazione dinamica della memoria (stack-based) [7, 8, 26]. La VM è un'implementazione di un modello astratto di calcolatore che virtualizza componenti come la CPU, la memoria e l'archiviazione, e viene spesso utilizzata per simulare un ambiente computazionale isolato all'interno di un computer fisico. Nell'ambito delle tecnologie blockchain, la VM utilizzata tiene traccia dello stato della blockchain, inclusi gli account, il bytecode degli smart contract e la memoria da essi utilizzati. L'esecuzione del codice avviene tramite la valutazione delle istruzioni del bytecode in maniera sequenziale qualora non si verifichino condizioni di errore.

L'architettura della VM può essere in via generale costituita dalle seguenti componenti [7, 18]:

- stack: porzione dinamica di memoria che esegue un set di istruzioni in maniera push-down (verso indirizzi di memoria decrescenti); solitamente ha una lunghezza limitata e compatibile con tipi di dati base come bytes e interi; lo stack viene resettato ad ogni esecuzione;
- memoria: spazio di memoria volatile (o temporaneo) dedicato a una singola esecuzione e che non viene salvato in archiviazione sullo stato globale;
- memoria di archiviazione: spazio di memoria salvato sullo stato globale VM state, solitamente rappresentato con una struttura dati "chiave valore". Alcune blockchain offrono spazio di archiviazione limitato per smart contract, mentre altre definiscono dei limiti di memoria in modo da ottimizzare l'esecuzione e la lettura delle informazioni [6, 11, 18, 37, 38].

#### 3.1.2 Tradeoff tra ambienti di esecuzione Stateful e Stateless

La complessità dell'ambiente di esecuzione e la gestione dello stato impattano in maniera diretta sulle performance del sistema.

Esistono, infatti, *tradeoff* tra la realizzazione di ambienti di esecuzione *stateless* e ambienti di esecuzione *stateful*. Nel primo caso, l'ambiente di esecuzione non memorizza informazioni o riferimenti dovute ad esecuzioni passate [39, 40, 41]. L'interprete utilizzato da Bitcoin per l'esecuzione di script è un esempio di ambiente di esecuzione *stateless* [17, 41]. Al contrario, ambienti di esecuzione in cui si tiene traccia delle informazioni derivanti da esecuzione passate vengono detti *stateful* [6, 18, 39]. L'ambiente di esecuzione basato su VM utilizzato in Ethereum è un esempio di ambiente di esecuzione *stateful*.

Quindi nel caso *stateless*, l'ambiente di esecuzione risulta ristretto e limitato in quanto questo non può fare affidamento su condizioni e informazioni salvate in memoria. Gli ambienti di esecuzione *stateless*, tuttavia, consentono un'esecuzione sicura, efficiente e a costi contenuti [7, 40, 41, 42]. Al contrario, in ambienti *stateful* è possibile fare affidamento sulla memoria per mantenere uno stato che può essere consultato anche in diverse esecuzioni dello smart contract. Questo rende il sistema più flessibile, in quanto permette lo sviluppo di applicazioni più complesse, a fronte però di costi di gestione più elevati. A tal proposito, bisogna infatti valutare i *tradeoff* sulla gestione della memoria. Nel caso in cui

non esistono limiti di memoria sono gli sviluppatori a dover gestire la memoria dello smart contract in maniera ottimale, eventualmente introducendo rischi legati a possibili bug o implementazioni non efficienti. Al contrario, disporre di una memoria limitata aiuta a mitigare possibili errori, al costo di una ridotta flessibilità [42].

### 3.1.3 Linguaggio di programmazione

In modelli di esecuzione *stateful* basati su VM, sullo stack della VM viene eseguito un set di istruzioni basiche (opcode) [43] relativo a operazioni logiche, aritmetiche e operazioni specifiche per interagire con lo stato della blockchain.

Il bytecode viene generato da un compilatore e da un codice sorgente solitamente scritto in un linguaggio di alto livello. Esistono diverse tipologie di linguaggi con cui è possibile sviluppare smart contract, solitamente dipendenti dalla blockchain e dalla VM di riferimento. Ad esempio, per Ethereum i due linguaggi più diffusi sono Solidity [25] e Vyper [44].

I linguaggi di programmazione offrono numerose funzionalità quali [7, 25, 40, 43]:

- estensione dei tipi supportati nativamente dalla VM (bytes e interi) per casi d'uso più complessi;
- utilizzo di librerie: definizione di funzioni riutilizzabili nello smart contract o invocabili da altri smart contracts;
- funzioni di supporto per la manipolazione dello stato della VM.

Queste funzionalità determinano la flessibilità di un linguaggio di programmazione, e in particolare la possibilità di sviluppare complesse applicazioni tramite gli strumenti offerti quali appunto i tipi, le librerie e le funzioni.

### 3.1.4 Tradeoff *Turing-Complete* vs *Non Turing-Complete*

I linguaggi di programmazione di smart contract si differenziano principalmente tra quelli Turing completi e non Turing completi. In generale, la letteratura definisce un linguaggio come Turing completo se è in grado di esprimere un problema computazionale complesso risolvibile con una macchina di Turing (un computer) [45, 46].

Nell'ambito degli smart contract, Ethereum è stata la prima piattaforma ad offrire una VM in grado di eseguire un linguaggio Turing completo (Solidity [25]). Il vantaggio della Turing *completeness* è che gli smart contract risultano estremamente flessibili, consentendo agli sviluppatori di riprodurre qualsiasi tipo di calcolo direttamente sulla blockchain [8]. Ad esempio, un linguaggio Turing completo permette lo sviluppo di programmi, anche complessi, che utilizzano loop o funzioni ricorsive. Tuttavia, questo introduce una maggiore complessità e quindi rende il sistema stesso più incline a errori e vulnerabilità (ad esempio, si può incorrere in esecuzioni infinite del codice) [7, 42, 47]. Al contrario, un linguaggio non Turing completo rinuncia alla flessibilità favorendo un linguaggio più semplice e meno soggetto a errori/bug. Ad esempio, il linguaggio di scripting adottato in Bitcoin è non Turing completo [17, 41, 47].

## 3.2 Caratteristiche di alto livello

In questa sezione vengono descritte le caratteristiche di alto livello per la valutazione degli smart contract, quali i costi di deployment e i costi di esecuzione. Questi si differenziano rispetto alla blockchain sottostante e vengono descritti come segue:

- Costi di deployment: il deployment di uno smart contract avviene tramite l'invio di una transazione di deployment sulla blockchain. Questa transazione deve contenere al suo interno il bytecode dello smart contract che verrà scritto sul ledger e istanziato nel contesto della VM. In particolare, la VM dovrà dedicare allo smart contract uno spazio di memoria per la gestione dello stack, della memoria, e della memoria di archiviazione. Per l'esecuzione di smart contract, i costi sono determinati dalla dimensione del bytecode e allo spazio di memoria che questo richiede per operare. A questi costi, si sommano inoltre quelli di transazione, cioè le commissioni che gli utenti devono pagare per eseguire il deployment di uno smart contract sulla blockchain stessa. [48, 49]
- Costi di esecuzione: oltre ai costi di deployment vi sono i costi di esecuzione. Ogni smart contract, infatti, viene invocato tramite una transazione che ne esegue la logica. Questo prevede il consumo delle risorse, che sulla blockchain rappresentano un bene comune e condiviso. Per eseguire le operazioni dello smart contract è quindi necessario coprire i costi di esecuzione. Ad esempio, in Ethereum, questo avviene tramite il concetto di gas [50].

## 4. Considerazioni di sicurezza

Illustriamo ora le considerazioni di sicurezza da valutare per lo sviluppo di DApp. Sebbene l'architettura di DApp possa coinvolgere una vasta superficie di attacco, determinata dall'intero stack applicativo (ad esempio, un backend che si interfaccia con lo smart contract, un frontend per la fruizione delle UI/UX, basi di dati, ecc.), in questa sezione si propone un focus esclusivo sul dominio di esecuzione dello smart contract. L'analisi proposta è, quindi, divisa in due parti. Nella prima si evidenziano le sfide relative allo sviluppo sicuro di applicazioni basate su smart contract. Nella seconda parte, si illustrano le componenti tecniche e tecnologiche degli smart contract che possono rendere il sistema vulnerabile ad attacchi e quindi non affidabile. In particolare, appare sin d'ora evidente la necessità di linee guida che indichino in maniera dettagliata modalità e strumenti per prevenire e per reagire a eventuali attacchi agli smart contract consentiti da vulnerabilità presenti: (i) nella VM, (ii) nel linguaggio di programmazione, (iii) nel codice.

### 4.1 Sfide per lo sviluppo di DApp sicure

I paradigmi di programmazione di smart contract per DApp richiedono un processo di sviluppo differente rispetto agli approcci tradizionali di ingegneria del software. La tecnologia da un lato garantisce opportunità, dall'altro produce rischi nuovi imputabili, ad esempio, ad attacchi informatici. In generale, un fallimento di uno smart contract ha un impatto significativo sul piano dei costi, in quanto il valore da esso controllato o che esso rappresenta potrebbe essere compromesso, perso, o rubato. Si definiscono di seguito le principali sfide che gli sviluppatori devono affrontare per approcciare lo sviluppo sicuro di applicazioni decentralizzate.

1. Accessibilità: gli smart contracts vengono eseguiti in un ambiente decentralizzato e ogni fase del ciclo di vita di uno smart contract comporta sfide di sicurezza.

2. Modularità: il supporto ad applicazioni comprensibili e robuste impone modelli architetturali che considerino tutte le possibili interazioni interne alla VM (e.g. da altri smart contracts) ed esterne allo smart contract (chiamate da attori malevoli tramite transazioni invocanti) [52, 53].

3. Complessità: la compilazione dello smart contract deve generare bytecode semplice per limitare possibili pattern di esecuzione inaspettati [8].

4. Test: Per eseguire il deployment in produzione, lo smart contract deve essere testato e validato da più fonti; una volta in esecuzione bisogna assicurarsi che tutte le funzionalità dello smart contract seguano le condizioni contrattualizzate [35, 54].

5. Recupero: poiché gli smart contract sono programmi scritti in linguaggio macchina ed eseguiti in maniera decentralizzata, potrebbero essere esposti a malfunzionamenti, errori o addirittura a manipolazioni esterne. In caso di logiche di business per la gestione di valore, è *fondamentale* quindi definire modelli di recupero/annullamento delle operazioni con cui gli sviluppatori (o entità individuate alla gestione) possano aggiornare lo smart contract e in casi estremi bloccarne l'esecuzione (o cancellarla) e mitigare il rischio di furti; ad esempio alcune soluzioni nello stato dell'arte propongono pattern di recupero di asset tokenizzati [9].

6. Aggiornamento sicuro: il paragrafo 3.4 ("Aggiornamento e Governance") ha introdotto i meccanismi di aggiornamento di uno smart contract e i relativi attori autorizzati; per garantire la sicurezza di uno smart contract è essenziale gestire propriamente la governance sui pattern di aggiornamento, definendo i permessi e i ruoli degli attori coinvolti in base al caso d'uso.

7. Interoperabilità: gli smart contract vengono installati ed eseguiti nel contesto di una singola piattaforma blockchain; questo pone dei limiti alle funzioni di interoperabilità tra blockchain [55]. Come evidenziato nel Focus 7 della Parte II, i *bridges* sono soluzioni applicative che prevedono la composizione di smart contract tra diverse blockchain; questi rappresentano punti di vulnerabilità per le DApp;

8. Input esterni: gli smart contract, e in generale le blockchain, non hanno accesso a dati presenti esternamente. In particolare, per accedere a informazioni specifiche vengono utilizzati Oracoli [10] che forniscono, in base al caso d'uso, le informazioni da utilizzare nella logica di business dello smart contract; è cruciale per la sicurezza delle DApp validare gli input provenienti da oracoli e limitarne l'utilizzo per evitare possibili manipolazioni esterne.

## 4.2 Possibili vulnerabilità

Gli smart contract vengono eseguiti sulla blockchain in maniera decentralizzata e in assenza di fiducia. Per garantire la correttezza, abbiamo visto nelle precedenti sezioni come sia necessario l'utilizzo di complessi sistemi, partendo dalla blockchain sottostante fino all'ambiente di esecuzione degli smart contract. In questa sezione si analizzano le componenti degli smart contract che possono nascondere minacce di sicurezza dovute a vulnerabilità o errori nel codice.

Le aree individuate sono state ottenute da una revisione di alcuni dei lavori esistenti in letteratura sulle vulnerabilità degli smart contract (ad esempio, [10, 11, 51, 56]):

1. l'ambiente di esecuzione: la VM dove vengono eseguite le istruzioni degli smart contract è una componente critica per la sicurezza. Occorrono strumenti adeguati a prevenire e reprimere problemi legati al contesto di esecuzione. Ad esempio: una VM con vincoli computazionali troppo stringenti potrebbe causare fallimenti imprevisti nell'esecuzione, oppure bug nell'implementazione degli opcode potrebbero generare risultati inattesi. Per sviluppare o utilizzare uno smart contract è quindi importante valutare le vulnerabilità della VM sottostante;
2. il linguaggio di programmazione: il bytecode degli smart contract eseguito sulla VM è compilato da un programma scritto in linguaggio di alto livello. In fase di compilazione, si deve garantire che il linguaggio stesso non generi errori o vulnerabilità. Ad esempio: è fondamentale che il linguaggio di alto livello offra funzionalità di scrittura di metodi ad

accesso controllato onde evitare manipolazioni da utenti non autorizzati, oppure che garantisca librerie di calcolo aritmetico sicure;

3. il codice: per quanto gli smart contract siano eseguiti in un contesto sicuro e affidabile garantito dalla tecnologia blockchain, si tratta di programmi che possono presentare errori dovuti a codice incorretto o con esecuzioni impreviste. Un codice è sicuro se tiene conto che, in un contesto di esecuzione come una blockchain, chiunque può accedere ai dati scritti nel ledger, o interagire con gli smart contract tramite transazioni invocanti. La blockchain è un ambiente, per sua natura, trustless, e su questa assunzione occorre strutturare gli smart contract.

## 5. Conclusioni

In questa sezione sono stati introdotti gli smart contract sul piano tecnico e tecnologico.

Nella prima parte sono state analizzate le principali componenti degli smart contract, presentando i modelli di stato account-based e token-based e discutendone le peculiarità e le differenze. Nel contesto di modelli *account-based*, è stato inoltre descritto il ciclo di vita degli smart contract e le metodologie note per l'aggiornamento degli stessi e per la governance. Successivamente, è stata proposta una tassonomia delle caratteristiche fondamentali degli smart contract, evidenziando i tradeoff tra gli ambienti di esecuzione con stato (*stateful*) e senza stato (*stateless*) e tra i linguaggi di programmazione Turing completi e non Turing completi. La sezione si conclude con le caratteristiche di alto livello focalizzando lo studio sui costi di deployment ed esecuzione degli smart contract.

Nella seconda parte è stato proposto un approfondimento sulla sicurezza, identificando le sfide di sicurezza che gli sviluppatori devono affrontare per la realizzazione di DApp sicure e affidabili e una classificazione delle possibili vulnerabilità che possono affliggere gli smart contract quali la VM, il linguaggio di programmazione e il codice.

## Bibliografia

[1] Mik, E. "Smart Contracts: Terminology, Technical Limitations and Real World Complexity" SSRN: https://ssrn.com/abstract=3038406 or http://dx.doi.org/10.2139/ssrn.3038406 (2017)
[2] Zheng, Z., Xie S., Dai H-N., Chen, W., Chen X., Weng J., Imran M. "An Overview in Smart Contracts: Challenges, advances and platforms", FGCS (2020)
[3] Mohanta B. K., Panda S. S. and Jena, D. "An Overview of Smart Contract and Use Cases in Blockchain Technology," 9th International Conference on Computing, Communication and Networking Technologies (ICCCNT 2018)
[4] Khan, S.N., Loukil, F., Ghedira-Guegan, C. et al. "Blockchain smart contracts: Applications, challenges, and future trends", Peer-to-Peer Netw. Appl. 14, 2901–2925 (2021)
[5] Wang, S., Ouyang, L., Yuan, Y., Ni, X. and Han, X. "Blockchain-Enabled Smart Contracts: Architecture, Applications, and Future Trends", IEEE Transactions on Systems, Man, and Cybernetics (2019)
[6] Buterin, V. "A Next Generation Smart Contract & Decentralized Application Platform" (2015)
[7] Zheng, S., Wang, H., Wu, L., Huang, G., & Liu, X, "VM Matters: A Comparison of WASM VMs and EVMs in the Performance of Blockchain Smart Contracts", ArXiv, abs/2012.01032 (2020)
[8] Kannengießer, N., Lins S., Sander C., Winter K., Frey H. and Sunyaev A."Challenges and Common Solutions in Smart Contract Development", IEEE Transactions on Software Engineering, vol. 48, no. 11, pp. 4291-4318, 1, doi: 10.1109/TSE.2021.3116808 (2022)
[9] López Vivar A, Castedo AT, Sandoval Orozco AL, García Villalba LJ. An Analysis of Smart Contracts Security Threats Alongside Existing Solutions. Entropy (Basel); 22(2):203 (2020)

[10] Atzei, N., Bartoletti, M., Cimoli, T. (2017). A Survey of Attacks on Ethereum Smart Contracts (SoK). In: Maffei, M., Ryan, M. (eds) Principles of Security and Trust. POST 2017. Lecture Notes in Computer Science, vol 10204.
[11] De Angelis, S. and Zanfino, G., Aniello, L., Lombardi, F., Sassone, V. "Evaluating Blockchain Systems: A Comprehensive Study of Security and Dependability Attributes", Proceedings http://ceur-ws. org ISSN, (2022)
[12] NCC Group, "Decentralized Application Security Project - DASP Top 10" (2018). URL: https://dasp.co/index.html
[13] Securing, "Smart Contract Security Verification Standard" (2020). URL: https://github.com/securing/SCSVS
[14] "Smart Contract Weakness Classification Registry" (2019), URL: https://swcregistry.io/
[15] Szabo, N. "Secure Property Titles with Owner Authority" (1998), https://web.archive.org/web/20140115142013/http://szabo.best.vwh.net/securetitle.html
[16] Nakamoto, S. "Bitcoin: A peer-to-peer electronic cash system" (2008) URL: https://bitcoin.org/bitcoin.pdf.
[17] Bistarelli, S. "Mercanti Ivan, Santini Francesco, "An Analysis of Non-standard Transactions"., Crypto Valley Conference on Blockchain Technology (CVCBT 2018), pp. 93-96, doi: 10.1109/CVCBT.2018.00016 (2018)
[18] Wood, G. "Ethereum: A secure Decentralised Generalised Transaction Ledger", EIP-150 REVISION (2014)
[19] Cai, W, Wang, Z., Ernst J. B., Hong, Z., Feng C. and Leung V. C. M. "Decentralized Applications: The Blockchain-Empowered Software System," *IEEE Access*, vol. 6, pp. 53019-53033, (2018), doi: 10.1109/ACCESS.2018.2870644
[20] Chibuzor, U., Anyanka, H. and Norta. A. "Evaluation of Approaches for Designing and Developing Decentralized Applications on Blockchain. In Proceedings of the 4th International Conference on Algorithms, Computing and Systems (ICACS '20)". Association for Computing Machinery, New York, NY, USA, 55–62. https://doi.org/10.1145/3423390.3426724 (2020)
[21] Morabito, V. "Smart Contracts and Licensing", Business Innovation Through Blockchain (pp.101-124) (2017)
[22] Alharby M., Aldweesh A. and Moorsel A. V., "Blockchain-based Smart Contracts: A Systematic Mapping Study of Academic Research" International Conference on Cloud Computing, Big Data and Blockchain (ICCBB 2018), pp. 1-6, doi: 10.1109/ICCBB.2018.875639
[23] Bloo F.W.C., "Towards updatable smart contracts", http://essay.utwente.nl/76769/(2018)
[24] Bui V. C., Wen S., Yu J., Xia X., Haghighi M. S. and Xiang Y., "Evaluating Upgradable Smart Contract," IEEE International Conference on Blockchain (Blockchain 2021) pp. 252-256, doi: 10.1109/Blockchain53845.2021.00041
[25] "Solidity: Ethereum Smart Contracts Programming Language", URL: https://soliditylang.org/
[26] Zou W. et al. "Smart Contract Development: Challenges and Opportunities,", IEEE Transactions on Software Engineering, vol. 47, no. 10, pp. 2084-2106, 1 (2021), doi: 10.1109/TSE.2019.2942301
[27] Connors, C. and Sarkar, D. "Comparative Study of Blockchain Development Platforms: Features and Applications", eprint: arXiv:2210.01913 (2022)
[28] Konrad, R. and Stephen P. "Bitcoin UTXO Lifespan Prediction.", Available at: https://cs229.stanford.edu/proj2015/225_poster.pdf (2015)
[29] Chakravarty, M., Chapman, J., MacKenzie, K., Melkonian, O., Jones, M. and Wadler, P. "The Extended UTXO Model", Workshop on Trusted Smart Contracts (Financial Cryptography (2020)
[30] Brünjes, L., Gabbay, M.J. "UTxO- vs Account-Based Smart Contract Blockchain Programming Paradigms", in Margaria, T., Steffen, B. (eds) Leveraging Applications of Formal Methods,

Verification and Validation: Applications. IsoLA, Lecture Notes in Computer Science, vol 12478. Springer, Cham. https://doi.org/10.1007/978-3-030-61467-6_6 (2020)
[31] EU Blockchain Observatory and Forum, "Smart Contracts" (2022)
[32] Sillaber, C., Waltl, B. "Life Cycle of Smart Contracts in Blockchain Ecosystems." Datenschutz Datensich 41, 497–500 (2017)
[33] Upgrading Ethereum Smart Contracts. URL:
https://ethereum.org/en/developers/docs/smart-contracts/upgrading/#proxy-patterns
[34] Salehi, M., Clark, J. and Mannan, M. "Not so immutable: Upgradeability of Smart Contracts on Ethereum" (2022) 10.48550/arXiv.2206.00716
[35] Chen, J., Xia, X. and Lo, D. and Grundy, J. "Why Do Smart Contracts Self-Destruct? Investigating the Selfdestruct Function on Ethereum", ACM Trans. Softw. Eng. Methodol., Vol. 1, No. 1, Article 1 (2020)
[36] Chatterjee A. and Hansdah R. C. "Deploying Transactional Smart Contracts using Multisignature Boolean Formulas. In Proceedings of the 23rd International Conference on Distributed Computing and Networking (ICDCN '22). Association for Computing Machinery 170–174. https://doi.org/10.1145/3491003.3491014 (2022)
[37] Chakravarty, M., Chapman, J., MacKenzie, K., Melkonian, O., Müller, J., Jones, M., Vinogradova, P. and Wadler, P. "Native Custom Tokens in the Extended UTXO Model", ISoLA (2020)
[38] Chaudhury, A. and Haney, B. "Smart Contracts on Algorand", IUP Journal of Knowledge Management, Available at SSRN: https://ssrn.com/abstract=3887719 or http://dx.doi.org/10.2139/ssrn.3887719 (2021)
[39] Johnson S, Hyland-Wood D., Madsen A.L. and Mengersen K., "Stateful to Stateless: Modelling Stateless Ethereum", Electronic Proceedings in Theoretical Computer Science (2022)
[40] Bartoletti, M., Bracciali, A., Lepore, C., Scalas, A. and Zunino, R. "A formal model of Algorand smart contracts", eprint: arXiv:2009.12140 (2020)
[41] Antonopoulos A.M., "Mastering Bitcoin", O'Reilly Media, Inc. (2014)
[42] Jansen, M., Hdhili, F., Gouiaa, R., Qasem, Z. (2020). Do Smart Contract Languages Need to Be Turing Complete?. In: Prieto, J., Das, A., Ferretti, S., Pinto, A., Corchado, J. (eds) Blockchain and Applications. BLOCKCHAIN 2019. Advances in Intelligent Systems and Computing, vol 1010. Springer, Cham. https://doi.org/10.1007/978-3-030-23813-1_3
[43] Bistarelli, S., Mazzante, G., Micheletti, M., Mostarda, L., Tiezzi, F. (2020). "Analysis of Ethereum Smart Contracts and Opcodes". In: Barolli, L., Takizawa, M., Xhafa, F., Enokido, T. (eds) Advanced Information Networking and Applications. AINA 2019. Advances in Intelligent Systems and Computing, vol 926. Springer, Cham. https://doi.org/10.1007/978-3-030-15032-7_46
[44] Vyper: A Smart Contract Programming Language for the EVM. URL https://docs.vyperlang.org/en/latest/
[45] Hopcroft, J., and Ullman, J. "Introduction to Automata Theory, Languages, and Computation", Addison-Wesley (1979).
[46] Teller, A. "Turing completeness in the language of genetic programming with indexed memory," Proceedings of the First IEEE Conference on Evolutionary Computation. IEEE World Congress on Computational Intelligence, Orlando, FL, USA, (1994)
[47] Lantz, L., Cawrey, D. "Mastering Blockchain: Unlocking the Power of Cryptocurrencies, Smart Contracts, and Decentralized Applications", O'Reilly Media 1st edition (2020)
[58] Suvitha, M. and Subha, R. "A Survey on Smart Contract Platforms and Features," 7th International Conference on Advanced Computing and Communication Systems (ICACCS), Coimbatore, India (2021) pp. 1536-1539, doi: 10.1109/ICACCS51430.2021.9441970.
[48] Baird, K., Jeong S., Kim, Y., Burgstaller, B. and Scholz, B. "The Economics of Smart Contracts", CoRR (2019)

[49] Chakravarty, M., Chapman, J., MacKenzie, K., Melkonian, O., Müller, J., Jones, M., Vinogradova, P. and Wadler, P. "Native Custom Tokens in the Extended UTXO Model", ISoLA (2020)
[50] Ethereum Gas. URL: https://ethereum.org/en/developers/docs/gas/
[51] Destefanis, G., Marchesi, M., Ortu, M., Tonelli, R., Bracciali A. and Hierons, R. "Smart contracts vulnerabilities: a call for blockchain software engineering?" 2018 International Workshop on Blockchain Oriented Software Engineering (IWBOSE 2018), pp. 19-25, doi: 10.1109/IWBOSE.2018.8327567
[52] Wöhrer M. and Zdun, U. "Design Patterns for Smart Contracts in the Ethereum Ecosystem", IEEE International Conference on Internet of Things (iThings) and IEEE Green Computing and Communications (GreenCom) and IEEE Cyber, Physical and Social Computing (CPSCom) and IEEE Smart Data (SmartData) (2018)
[53] Chen, J., Xia, X., Lo, D., Grundy, J., Luo X. and Chen, T. "Defining Smart Contract Defects on Ethereum," IEEE Transactions on Software Engineering, vol. 48, no. 1, pp. 327-345, 1 Jan. 2022, doi: 10.1109/TSE.2020.2989002
[54] Akca, S., Peng, C. and Rajan, A. "Testing Smart Contracts: Which Technique Performs Best?", Proceedings of the 15th ACM / IEEE International Symposium on Empirical Software Engineering and Measurement (ESEM) (ESEM 2021)
[55] Beniiche, A. "A Study of Blockchain Oracles", arxiv.org/pdf/2004.07140, (2020)
[56] Watanabe H., Fujimura S., Nakadaira A., Miyazaki Y., Akutsu A. and Kishigami J. "Blockchain contract: Securing a blockchain applied to smart contracts", IEEE International Conference on Consumer Electronics (ICCE 2016) pp. 467-468, doi:10.1109/ICCE.2016.7430693
[57] Kannengießer N. *et al.*, "Trade-offs between Distributed Ledger Technology Characteristics", ACM Comput. Surv. 53.2, (May 2020)