disponibili per lo sviluppo di applicazioni standard. Ciò costringe alcuni sviluppatori ad adottare l'uso di messaggi di errore con elevata verbosità, ad abilitare le variabili dell'ambiente di debug e a volte a dimenticare di ripulire il codice quando lo si sposta nell'ambiente di produzione. I messaggi di errore particolarmente verbosi come lo stack trace o gli errori di sintassi, che vengono esposti agli utenti finali, possono rivelare dettagli sulla logica interna della funzione serverless, e a sua volta rivelare potenziali debolezze, difettosità o addirittura dare luogo a perdite di dati sensibili. Se l'ambiente serverless consente di definire messaggi di errori personalizzati, come nel caso degli API gateway, è opportuno generare messaggi di errore semplici che non rivelano dettagli implementativi o il contenuto di eventuali variabili di ambiente.

1.1.1.3. Best practice di secure design per le architetture basate su registri distribuiti (DLT)

I registri distribuiti (DLT o più notoriamente Blockchain) sono sistemi informatici che gestiscono dati, transazioni o codici eseguibili (Smart Contracts) in modo il più possibile indipendente da un'autorità centrale attraverso l'utilizzo di data storage distribuito in correlazione con processi crittografici e sistemi decisionali decentralizzati. Le DLT sono in rapido sviluppo per le più svariate applicazioni, da nuovi sistemi di identità digitale (Self Sovereign Identity) a sistemi di certificazione garantita di dati e certificazioni (Verifiable Claims), alla creazione e gestione di nuovi mercati digitali (vedi i mercati peer to peer di scambio energetico) fino alla creazione e gestione di nuove entità (vedi ad esempio le DAO/DAC Distributed Autonomous Organizations/Corporations). In questo documento evitiamo volutamente di menzionare le criptovalute, da cui le DLT sono nate, ma che esulano dal contesto di questa analisi.
Queste caratteristiche delle DLT hanno permesso la gestione di applicazioni in cui differenti partecipanti, con obiettivi in conflitto, possono comunque operare di concetto attraverso processi predefiniti ed immodificabili.
Bisogna comunque tenere in considerazione che questa classe di tecnologie annovera strumenti con design alquanto differenti, sia per la gestione die permessi di lettura/scrittura (permissioned/unpermissioned DLT) che per la specifica implementazione del processo di maggiore criticità, cioè quello della gestione del consenso.
I concetti espressi in questo paragrafo prevedono una buona conoscenza della tecnologia.
È fondamentale prima di tutto valutare la tematica dei dati memorizzati in una DLT relativamente all'integrità, alla disponibilità ed alla riservatezza.
Affrontando per primo il tema dell'integrità dei dati, uno dei fattori fondamentali che assicurano tale caratteristica in una DLT è la struttura a blocchi concatenati attraverso la funzione di hash. Ma questa di per sé non basterebbe se non fosse affiancata da un adeguato algoritmo di consenso che garantisca l'immediata evidenza di una qualsivoglia modifica di questa struttura in confronto a strutture uguali esistenti nei vari nodi componenti la DLT.
Tenendo conto che la struttura a blocchi amplifica la sicurezza della funzionalita' di hash attraverso la concatenazione (cioe' la modifica di un blocco in posizione <n> implicherebbe la conseguente modifica di tutti i blocchi successivi fino alla testa della catena) il fattore critico da considerare relativamente all'integrità dei dati è chiaramente quello dato dall'algoritmo di consenso. Riportiamo quindi la tematica alla sezione relativa alla sicurezza di tale algoritmo.
In una DLT la disponibilità dei dati è intrinsecamente garantita dall'infrastruttura sottostante: difatti ogni nodo di una DLT (a meno di nodi specifici, normalmente chiamati 'light nodes') detiene una intera copia della struttura dati e quindi l'indisponibilita' degli stessi si potrebbe avere solo nel caso in cui tutti i nodi fossero allo stesso istante inattivi. Attacchi che mirino ad un eventuale centro non avrebbero senso in una struttura decentralizzata.

Un possibile attacco sulla disponibilità quindi dovrebbe essere mirato all'intera infrastruttura e quindi avere ad esempio come obiettivo il software di base, al fine di rendere indisponibili le informazioni negandone l'accesso attraverso la modifica della modalita' con cui tutti i nodi operano.
In una DLT la riservatezza dei dati varia ampiamente a seconda della specifica tecnologia utilizzata: alcune DLT sono intrinsecamente trasparenti (vedi bitcoin) in quanto chiunque ha accesso a tutte le transazioni fatte da qualsivoglia partecipante e la confidenzialità è demandata all'anonimità dei partecipanti che vengono identificati da un indirizzo generico. Il problema è che, una volta questo indirizzo venga correlato ad un utente, tutte le transazioni di questo divengono immediatamente visibili.
Alcune DLT sono invece orientate all'anonimità dei partecipanti e delle transazioni (es. zcash, monero) ed ad un osservatore esterno è praticamente impossibile risalire agli attori parte di una qualsiasi transazione proprio per le tecnologie utilizzate (es. ring transactions in monero).
In qualsiasi caso è necessario provvedere ad una adeguata gestione della riservatezza dei dati in modo disgiunto dalla gestione dell'infrastruttura: il livello di riservatezza da implementare sarà come al solito legato alla ripologia dei dati e le metodologie le stesse applicate a qualsiasi altra tipologia di data storage.

Analizziamo qui di seguito la sicurezza delle DLT, approcciando i vari elementi fondamentali che le compongono:

1. Rete infrastrutturale peer-to-peer
2. Struttura dati (concatenazione di blocchi)
3. Algoritmi di consenso (e.g. PoW, PoS, etc)
4. Smart contracts e logiche dinamiche (e.g. token, DAC/DAO, etc)

Riguardo l'infrastruttura, un possibile attacco di Distributed Denial of Service potrebbe provenire dall'utilizzo di wallet/accounts per generare grandi numeri di transazioni al fine di rallentare la rete. Un evento del genere, anche se non rappresentante un attacco, è avvenuto nella rete ethereum all'apparire dello smart contract CryptoKitties: questo smart contract ha raggiunto volumi superiori al 10% dell'intero volume di transazioni della rete, causando un indesiderato rallentamento dell'intera rete.
Nel mese di Marzo 2016 la rete Bitcoin si è quasi arrestata a causa di un wallet che generava larghi volumi di transazioni con un costo di transazione piu' elevato della media: i minatori, prioritizzando questo wallet piu' remunerativo, tendevano ad ignorare altre transazioni nella generazione di nuovi blocchi.
La struttura dati è correlata all'algoritmo di consenso che decide l'introduzione di nuove transazioni in coda alla catena, studiamo quindi queste due tematiche in modo congiunto.
Riguardo l'algoritmo di consenso, dobbiamo tenere prima conto del teorema CAP, che indica che è impossibile per una DLT garantire contemporaneamente piu' di due dei seguenti elementi:

- *Consistenza:* una DLT è consistente quando i fork sono evitati, caratteristica detta anche '*finalizzazione del consenso'* . In pratica i nodi devono tutti avere la stessa copia del ledger nello stesso momento
- *Disponibilita':* una DLT è disponibile se le transazioni generate dai client sono gestite e quindi committate (cioe' aggiunte alla catena di blocchi)
- *Tolleranza alle partizioni*: quando una partizione della rete avviene, i nodi autoritativi sono divisi in gruppi disgiunti affinche' i nodi di un gruppo non possono comunicare con i nodi di un altro gruppo

L'imposizione delle proprieta' suddette è responsabilita' dell'algoritmo di consenso, tale da garantire il funzionamento dell'intero sistema ed è compito di chi definisce l'architettura di decidere quali caratteristiche siano fondamentali per lo specifico caso d'uso.

Due forme di inconsistenza possono avvenire nell'algoritmo di consenso lasciando blocchi validi al di fuori di una DLT: la prima forma è quella dello "stale block", in cui un blocco minato con successo non viene accettato dall'attuale bestBlockchain (cioè la chain più complessa da ricreare): gli staleblock avvengono più frequentemente nelle blockchain pubbliche a causa delle consizioni di competizione tra i minatori (raceconditions).
L'altra forma di inconsistenza è basata sugli "orphaned block", cioè blocchi in cui il blocco padre ha un hash che punta ad un blocco non autentico.

Alcuni attacchi di cui possono essere vittime le DLT:

1. Il block withholding attack (BWH) ha come obiettivo le mining pools e fa in modo che il Sistema di reward delle stesse pool non sia più corretto, gratificando alcuni partecipanti al pool (che normalmente sono anche gli attaccanti) ricevere dei premi non proporzionali con il lavoro di mining effettuato.
2. L'attacco denominato 'selfish mining' prevede che un attaccante possa guadagnare dei premi non proporzionali con il lavoro di mining effettuato generando deliberatamente dei fork.
3. L'attacco denominato 'Fork after withholding' (FAW), similarmente al selfish mining, utilizza false fork: a differenza però il reward di un attaccante FAW è sempre maggiore od uguale a quello di un attaccante BWH ma è utilizzabile fino a 4 volte più spesso.
4. Il famoso attacco del 51% (principalmente orientato al consenso PoW) prevede che un gruppo di nodi con più del 51% della capacità di minare possa decidere in ogni caso qual è la prossima transazione indipendentemente dalla realtà (da considerare che un simile attacco su altri algoritmi quali il PBFT potrebbe essere portato anche con il 33% dei nodi).
5. L'attacco denominato 'eclipse attack' prevede un attacco non a tutta la rete ma solo ad uno o pochi nodi. L'attaccante isola il nodo e poi introduce transazioni malevole facendo in modo che lo stesso nodo non si accorga della non-sincronizzazione con i suoi peer.
6. L'attacco denominato 'spatial partitioning' prevede l'isolamento di un autonomous system che gestisce uno o più sistemi di mining e ridurre il mining power globale: questo attacco è portato in generale per facilitare altri attacchi attraverso proprio la riduzione del mining powrer
7. L'attacco denominato 'consensus delay' è associato con la natura P2P delle blockchain: in questo attacco si iniettano blocchi falsi con lo scopo di incrementare la altenza e quindi prevenire i nodi dal raggiungere un consenso sullo stato della DLT.
8. L'attacco denominato 'Finney Attack' prevede che il minatore possa minare un blocco che contenga una delle proprie transazioni e mantenerlo nascosto (stealth): c'è la possibilità di un double-spending nel cao in cui merchant accetti la transazione non confermata. A seguire il blocco nascosto viene pubblicato prima che la transazione venisse confermata dalla rete.

Non si vuole dare qui un resoconto esaustivo di tutti gli attacchi ma sottolineare le tipologie degli stessi e quindi le azioni preventive da tenere in considerazione a riguardo.
Bisogna in ogni caso tenere conto che la maggioranza degli attacchi è orientata verso le applicazioni (smart contracts) che girano sulla DLT e non sulla chain stessa: ci si deve quindi concentrare sulla qualità delle applicazioni che operano sulla DLT e prevedere per le stesse lo stesso ciclo di controllo di un qualsiasi software sviluppato come da linee guida AGID.